

**DESARROLLO DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN
FINANCIERA DEL SALÓN DE BELLEZA TODO ESTILO**

YEISON FERNEY PINEDA ORTIZ

**INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO
FACULTAD DE INGENIERÍA
TECNOLOGÍA EN DESARROLLO DE SOFTWARE
MEDELLÍN
2025**

**DESARROLLO DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN
FINANCIERA DEL SALÓN DE BELLEZA TODO ESTILO**

YEISON FERNEY PINEDA ORTIZ

**Trabajo de grado para optar al título de
TECNÓLOGO EN DESARROLLO DE SOFTWARE**

Asesores

LILIANA MARIA GARCIA AGUIRRE

Magister en Ingeniería de Software

JAVIER ALBERTO SALDARRIAGA CANO

Magister en Entornos Virtuales de Educación

INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO

FACULTAD DE INGENIERÍA

TECNOLOGÍA EN DESARROLLO DE SOFTWARE

MEDELLÍN

2025

Contenido

	Pág.
Resumen.....	8
Abstract.....	9
Glosario.....	10
1. Planteamiento del problema.....	15
1.1 Descripción.....	15
1.2 Formulación.....	16
2. Justificación.....	17
3. Objetivos.....	19
3.1. Objetivo general.....	19
3.2. Objetivos específicos.....	19
4. Marco teórico.....	20
4.1. Bases teóricas.....	20
4.1.1. Gestión financiera.....	20
4.1.2. Mecanismos de control.....	22
4.1.3. Análisis financiero.....	22
4.1.4 Método horizontal.....	22
4.1.5. Método vertical.....	23
4.1.6. Ventajas de la Digitalización Empresarial.....	24
4.1.7. Computación en la nube.....	25
4.1.8. Metodologías ágiles.....	26
4.1.9. Herramientas para el desarrollo.....	27
4.1.9.1. Programación Orientada a Objetos (POO).....	27
4.1.9.2. Modelo Vista Controlador (MVC).....	29
4.1.9.3. HTML5.....	30
4.1.9.4. PHP.....	31
4.1.9.5. Laravel.....	32
4.2. Antecedentes.....	32
4.2.1 Soluciones comerciales existentes.....	32
4.2.2 Proyectos académicos y universitarios similares.....	35
5. Metodología.....	37
5.1 Tipo de proyecto.....	37
5.2 Método.....	38
5.2.1 Fase de inicio y levantamiento de requerimientos:.....	38
5.3 Instrumentos de recolección de información.....	39
5.3.1 Fuentes primarias:.....	39
5.3.2 Fuentes secundarias:.....	40
5.4 Plan de actividades o cronogram.....	41
6. Resultados.....	42

6.2 Fase de inicio y levantamiento de requerimientos.....	42
6.2.1 Requisitos de usuario.....	44
6.2.2 Requerimientos funcionales.....	45
6.2.3 Requerimientos NO funcionales.....	48
6.2.4 Diagramas de casos de uso.....	50
6.3 Fase de diseño conceptual.....	52
6.3.1 Modelo entidad relación (MER).....	52
6.3.2 Diagrama de clases modelo UML.....	53
6.4 Fase de desarrollo ágil por iteraciones (sprints).....	55
6.4.2 Backlog.....	62
6.5 Fase de pruebas continuas y evaluación de resultados.....	65
6.6 Fase de implementación y documentación final.....	81
7. Conclusiones.....	83
7.1. Síntesis de los resultados obtenidos y cumplimiento de objetivos.....	83
7.2. Aportes generados por el proyecto.....	84
8. Recomendaciones.....	87
8.1. Mejoras y optimizaciones a corto y mediano plazo.....	87
8.2. Complementos al proyecto actual (nuevos módulos).....	88
8.3. Nuevas líneas de trabajo y proyectos de mayor envergadura.....	89
9. Referencias bibliográficas.....	91
10. Anexos.....	94

Lista de ilustraciones

	Pág.
Ilustración. 1 Diagrama de causa-efecto oportunidades de mejora en Todo Estilo.....	16
Ilustración. 2 Tipos de servicios de computación en la nube.....	26
Ilustración. 3 Esquema de cómo se encuentran relacionadas las tres capas de aplicación.....	30
Ilustración. 4 Testimonio de éxito de la aplicación Quickbooks en Colombia.....	33
Ilustración. 5 Página principal del software contable alegre.....	34
Ilustración. 6 Diagrama de flujo de proceso actual.....	43
Ilustración. 7 Diagrama de caso de uso clientes.....	50
Ilustración. 8 Diagrama de caso de uso de usuarios.....	50
Ilustración. 9 Diagrama de caso de uso financiero.....	51
Ilustración. 10 Modelo entidad relación.....	52
Ilustración. 11 Diagrama de clases modelo UML.....	54
Ilustración. 12 Estructura de la vista usuarios.....	58
Ilustración. 13 Estructura de la vista ingresos	58
Ilustración. 14 Estructura del formulario para el registro de ingresos.....	59
Ilustración. 15 Estructura del formulario para la edición de ingresos.....	59

Lista de tablas

	Pág.
Tabla. 1 Ejemplo de análisis horizontal.....	23
Tabla. 2 Ejemplo de análisis vertical	24
Tabla. 3 Cronograma de actividades.....	41
Tabla. 4 Requisitos de usuario.....	44
Tabla. 5 Requerimientos funcionales.....	45
Tabla. 6 Requerimientos NO funcionales.....	48
Tabla. 7 Product Backlog del proyecto.....	62
Tabla. 8 Caso de prueba CP-USUARIOS-001.....	66
Tabla. 9 Caso de prueba CP-USUARIOS-002.....	67
Tabla. 10 Caso de prueba CP-USUARIOS-003.....	68
Tabla. 11 Caso de prueba CP-USUARIOS-004.....	69
Tabla. 12 Caso de prueba CP-SERVICIOS-001.....	70
Tabla. 13 Caso de prueba CP-SERVICIOS-002.....	72
Tabla. 14 Caso de prueba CP-SERVICIOS-003.....	73
Tabla. 15 Caso de prueba CP-SERVICIOS-004.....	73
Tabla. 16 Caso de prueba CP-SERVICIOS-005.....	74
Tabla. 17 Caso de prueba CP-SERVICIOS-006.....	75
Tabla. 18 Caso de prueba CP-SERVICIOS-007.....	75
Tabla. 19 Caso de prueba CP-SERVICIOS-008.....	76
Tabla. 20 Caso de prueba CP-FINANCIERA-001.....	77
Tabla. 21 Caso de prueba CP-FINANCIERA-002.....	78
Tabla. 22 Caso de prueba CP-FINANCIERA-003.....	78
Tabla. 23 Caso de prueba CP-FINANCIERA-004.....	79
Tabla. 24 Caso de prueba CP-FINANCIERA-005.....	80
Tabla. 25 Caso de prueba CP-FINANCIERA-006.....	81
Tabla. 26 Caso de prueba CP-FINANCIERA-007.....	81
Tabla. 27 Caso de prueba CP-FINANCIERA-008.....	82

Lista de anexos

	Pág.
Anexo A. Aplicación web desarrollada (todo-estilo.zip).....	96
Anexo B. Manual de usuario.....	97
Anexo C. Manual de instalación.....	98

Resumen

DESARROLLO DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN FINANCIERA DEL SALÓN DE BELLEZA TODO ESTILO

YEISON FERNEY PINEDA ORTIZ

Este proyecto buscó resolver una necesidad real del salón de belleza Todo Estilo: la falta de un sistema organizado para el control financiero que permitiera el registro de ingresos, egresos, pagos y cobros. La solución fue desarrollar una aplicación web sencilla, práctica y adaptada a las necesidades del negocio, utilizando tecnologías como Laravel, PHP, HTML y Tailwind CSS. El proyecto se basó en programación orientada a objetos y el patrón Modelo Vista Controlador (MVC), lo que permitió estructurar el sistema de forma clara y funcional.

Se ejecuta el proyecto empleando una metodología con un enfoque ágil, que permitió tener una buena adaptación al proyecto, enfocarnos en el desarrollo de los objetivos de una manera dinámica y responder de forma adecuada a los cambios. En el proceso se incluyeron etapas fundamentales como el levantamiento de requerimientos, diseño de base de datos, creación de interfaces y desarrollo de módulos para la gestión financiera. Se realizaron pruebas para verificar que todo funcionara correctamente y se elaboró un manual de usuario para facilitar su uso por parte del personal. Con esta aplicación se busca mejorar el control financiero, centralizando la información y reduciendo los errores manuales.

Además de beneficiar directamente al negocio, este proyecto permitió al estudiante aplicar de manera práctica los conocimientos adquiridos en todas las etapas del ciclo de vida del software, desde la planificación hasta el despliegue. Como trabajo futuro, se propone ampliar la herramienta con funciones estadísticas, alertas automáticas y un módulo para el control de inventario y ventas, con el fin de fortalecer la toma de decisiones del negocio y brindar un sistema aún más completo.

Palabras claves: aplicación web, gestión financiera, salón de belleza, Laravel, MVC

Abstract

DEVELOPMENT OF A WEB APPLICATION FOR THE FINANCIAL MANAGEMENT OF THE BEAUTY SALON TODO ESTILO

YEISON FERNEY PINEDA ORTIZ

This project addressed a real need of the "Todo Estilo" beauty salon: the lack of an organized financial control system to register income, expenses, payments, and collections. The solution was to develop a simple, practical web application tailored to the business's needs, using technologies such as Laravel, PHP, HTML, and Tailwind CSS. The project was based on object-oriented programming and the Model-View-Controller (MVC) pattern, which allowed for a clear and functional system structure.

The process included fundamental stages such as requirements gathering, database design, interface creation, and development of financial management modules. Tests were conducted to verify proper functionality, and a user manual was created to facilitate its use by staff. This application aims to improve financial control by centralizing information and reducing manual errors.

In addition to directly benefiting the business, this project allowed the student to practically apply the knowledge acquired in all stages of the software development life cycle, from planning to deployment. For future work, it is proposed to expand the tool with statistical functions, automatic alerts, and a module for inventory and sales control, in order to strengthen business decision-making and provide an even more comprehensive system.

Keywords: web application, financial management, beauty salon, Laravel, MVC

Glosario

Aplicación: es un programa o herramienta web para realizar una serie de tareas o actividades interactivas, que permite satisfacer unas necesidades específicas o cumplir con un objetivo puntual.

Back-End: es la parte no visible de un programa o aplicación, esta parte del programa se encarga de interactuar con bases de datos y otras funciones lógicas del sistema.

Belleza: conjunto de cualidades físicas y estéticas que se realzan o mejoran mediante servicios especializados, como maquillaje, peinados, cuidado de la piel, manicura, pedicura, etc.

Contenido: información, texto, imágenes o archivos multimedia que se encuentran presentes en una página web o aplicación.

CSS: es una herramienta utilizada para mejorar la presentación visual de un documento o página web.

Desarrollo de Software: disciplina que comprende métodos, herramientas y técnicas para la creación de programas informáticos.

Diseño: es el proceso de creación y estructuración de un sistema, página web o aplicación.

Front-End: al contrario de back-end esta es la parte visual y funcional de una aplicación o página web que los usuarios ven e interactúan directamente con ella.

Hosting: servicio que proporcionan los servidores, como un espacio en línea para almacenar y hacer accesibles los sitios web en internet.

HTML: es un lenguaje utilizado principalmente para la estructuración de páginas web, ayuda al adecuado manejo de elementos como encabezados, párrafos, imágenes, listas, entre otros.

Interfaz: es el espacio donde los usuarios interactúan con una aplicación, página web o sistema.

Investigación: proceso de búsqueda, recopilación y análisis de información para ampliar los conocimientos y con esto poder resolver un problema o dificultad específica.

Laravel: framework de desarrollo web para el manejo del lenguaje de programación PHP, que facilita la creación de aplicaciones web escalables y seguras.

Liquidación: es el proceso de calcular, cerrar y pagar un valor pendiente, como una cuenta, salario, comisión, impuesto, o cualquier obligación económica.

Manicura: operación o trabajo de cuidado y embellecimiento de las manos y, especialmente, de las uñas.

Maquillaje: es el servicio de aplicación de productos cosméticos en el rostro u otras zonas visibles del cuerpo, con el fin de embellecer, resaltar rasgos, cubrir imperfecciones o adecuar la imagen para un evento.

MySQL: es un sistema gestor de bases de datos relacionales, que permite la gestión de información, muy utilizado en aplicaciones web.

Mockup: es el modelo gráfico que representa cómo se verá un programa, página web o sistema, de forma gráfica pero sin funcionalidades interactivas.

Optimización: es el método por el cual se busca mejorar la eficiencia de un sistema o recurso utilizando la menor cantidad posible de recursos.

Página web: es un sistema que permite la visualización de información en un sitio web, accesible por medio de un navegador, dentro de esta información se puede incluir texto, imágenes, videos y en donde también se puede tener interacción con los usuarios.

pedicura: cuidado y embellecimiento de los pies y, especialmente, de las uñas.

PHP: es un lenguaje de programación, utilizado principalmente para la creación de páginas web.

Programación orientada a objetos (POO): es una metodología de la programación donde se busca crear objetos, donde los objetos representan cosas del mundo real y están compuestas principalmente por dos componentes que son, las características del objeto (atributos) y las acciones o funciones del objeto(Métodos).

Repositorio web: es una plataforma en línea donde se almacenan y comparten recursos y archivos, facilitando la colaboración en equipo, la consulta de información y la recuperación de datos.

Scripts: secuencia de comandos escritos normalmente en lenguajes de programación o sistemas de gestión de bases de datos, utilizados para manipular, personalizar o automatizar procesos en un sistema existente, que por lo general ayudan o facilitan la operación de tareas repetitivas o complejas.

Servicio: Conjunto de actividades intangibles que se realizan para satisfacer las necesidades del público o de una persona.

Software: conjunto de programas o algoritmos diseñados para cumplir funciones específicas en un sistema informático.

Introducción

Desde finales de los años ochenta, la revolución digital ha transformado la economía y la sociedad. Con el nacimiento de la Internet surgió una economía conectada, que se caracterizó por la masificación del uso de Internet y redes de banda ancha. Luego, evolucionó a una economía digital impulsada por la expansión de plataformas como modelo de negocio para la oferta de bienes y servicios, más conocido como comercio electrónico o e-commerce. En la actualidad se apunta hacia una economía mucho más digitalizada, basada en la integración de tecnologías digitales en todas las dimensiones económicas, sociales y ambientales, lo que conlleva a que la tecnología sea un motor fundamental en el crecimiento económico, soportados principalmente en el tema de la conectividad, información, los datos y nuevos modelos disruptivos que emergen para generar más impacto y competitividad en los diferentes sectores económicos (CEPAL, 2021).

La implementación de nuevas tecnologías por parte de las empresas de todos los sectores económicos, hace que los mercados y la economía en general sean cada vez más competitivos y dinámicos, las nuevas herramientas tecnológicas para el manejo empresarial son fundamentales para mejorar los principales indicadores de eficiencia empresarial. Según el informe “Tecnologías digitales para un nuevo futuro” (CEPAL, 2021), la optimización de los procesos administrativos y el control financiero, son esenciales para el crecimiento y sostenibilidad de las pequeñas y medianas empresas (Pymes). De esta forma, tecnologías como las redes móviles 5G, el Internet de las cosas (IoT), la computación en la nube, la inteligencia artificial y el Big Data se convierten en factores clave para transformar sus operaciones y mantenerse competitivos en el mercado.

Por otra parte, la administración financiera y el control de procesos son retos críticos para los negocios de servicio como los salones de belleza, que buscan equilibrar la calidad de servicio al cliente con la eficiencia operativa. Este proyecto está centrado en el salón de belleza Todo Estilo, ubicado en Itagüí (San Gabriel). El negocio destaca por la buena calidad de sus servicios y contar con muy buena clientela, pero enfrenta dificultades para el control administrativo y el manejo de su contabilidad de forma eficaz.

Con este panorama, el desarrollo de una aplicación web para Todo Estilo tiene como propósito optimizar la gestión administrativa y financiera del negocio, que garantice un adecuado manejo de su contabilidad. La aplicación será diseñada con un enfoque moderno, seguro, funcional y de fácil manejo, lo que garantizará su adopción y sostenibilidad a largo plazo. Esta solución no sólo atenderá las necesidades inmediatas del negocio, sino que también integrará las nuevas tecnologías, mejorando su competitividad en el mercado actual.

1. Planteamiento del problema

1.1 Descripción

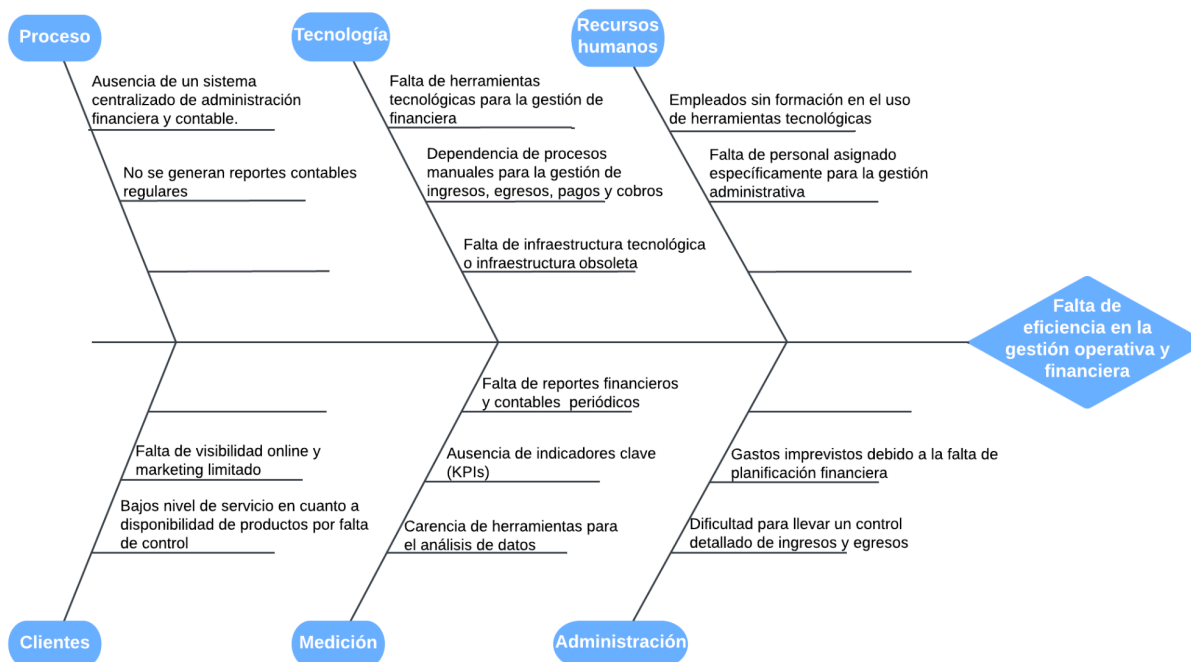
El salón de belleza Todo Estilo, está ubicado en el barrio San Gabriel del municipio de Itagüí, inició operaciones en el año 2016 en el corregimiento de San Antonio de Prado. Desde su apertura, se ha dedicado a ofrecer una variedad de servicios para el cuidado personal y la venta de productos de belleza. En los años de operación ha logrado consolidarse como un negocio de confianza para sus clientes, actualmente cuenta con cuatro empleados, dos especialistas en tratamientos capilares avanzados, como químicos, cortes y alisados, y otras dos especializadas en spa de uñas (manicura, pedicura), asegurando una atención integral para sus clientes.

A pesar de su crecimiento y de la calidad de los servicios que ofrece, Todo Estilo enfrenta serias dificultades en cuanto a la gestión administrativa y el manejo de su contabilidad. La falta de una herramienta tecnológica que permita gestionar eficientemente los procesos financieros y operativos está generando problemas en la toma de decisiones. La ausencia de un sistema organizado y centralizado para el control de ingresos, egresos, pagos y cobros está llevando a errores en la información financiera, lo que afecta la planificación y la capacidad del negocio para tomar decisiones informadas tanto a corto como a largo plazo y el poder gestionar de manera óptima sus recursos.

El desafío principal para Todo Estilo radica en la necesidad urgente de contar con un sistema que optimice tanto la gestión administrativa como la gestión contable, lo que permitirá una mejora en la eficiencia operativa y una mejor toma de decisiones. Esta solución también contribuiría a asegurar la sostenibilidad del negocio, brindando a los administradores las herramientas necesarias para mantener la competitividad en un mercado cada vez más digitalizado y demandante.

En el siguiente diagrama de Ishikawa se analiza a profundidad las posibles causas que pueden estar causando dichas dificultades.

Ilustración. 1
Diagrama de causa-efecto oportunidades de mejora en Todo Estilo



Nota: Diagrama de Ishikawa donde se plasman las causas y efectos inherentes al problema.
Fuente: Elaboración propia.

1.2 Formulación

¿Cómo desarrollar una aplicación web que optimice la gestión administrativa y contable del salón de belleza Todo Estilo, mejorando la eficiencia operativa y facilitando la toma de decisiones para asegurar su sostenibilidad en un mercado digitalizado y que garantice una experiencia funcional, segura y de fácil uso?

2. Justificación

Como se informa en la revista (CEPAL, 2021), la transformación digital ha redefinido la interacción y las operaciones empresariales desde la masificación de internet, justificando la necesidad que negocios como el salón de belleza Todo Estilo busquen trascender las limitaciones de la gestión tradicional. Teniendo en cuenta lo anterior, dentro del contexto empresarial hay ciertas habilidades o capacidades con las que las empresas deben contar para ser más competitivas en el mercado al que pertenecen, la eficiencia y la agilidad por ejemplo son las cartas de presentación de la competitividad empresarial, incorporar herramientas tecnológicas es de vital importancia, no solo se vuelven una ventaja competitiva sino una necesidad si desean alcanzar niveles de rendimiento mucho más altos.

Con base en la problemática identificada en el salón de belleza Todo Estilo, principalmente en las dificultades en el control administrativo y el manejo contable, se considera aprovechar las ventajas propias de una aplicación web. Esta solución tecnológica posibilitará la centralización automatizada de la información, proporcionando una visión integral y actualizada de la situación financiera y operativa del negocio. La aplicación web se materializa, buscando convertirse en una herramienta imprescindible para una gestión moderna, facilitando la supervisión de ingresos, egresos, control de pagos y gestión de las cuentas por cobrar, mejorando significativamente el control financiero y contable del negocio.

El presente proyecto se justifica en función de dos aspectos principales uno está basado los resultados directos que favorecerá a negocio Todo Estilo el segundo está basado en la perspectiva como estudiante de desarrollo de software, donde este proyecto conlleva una importancia crucial como ejercicio práctico de aplicación integral de los conocimientos adquiridos durante la formación académica. Con el objetivo de implementar las herramientas del ciclo de vida del software (planificar, diseñar, implementar, hacer pruebas, desplegar, realizar mantenimiento), dando una solución tecnológica real para la problemática identificada anteriormente en Todo Estilo.

Este proyecto permite la puesta en práctica de diversos conocimientos fundamentales en el

campo del desarrollo de software, esto incluye el análisis de requerimientos del cliente, diseño de la arquitectura de la aplicación web (frontend y backend), la creación e implementación de la base de datos para el almacenamiento y gestión de la información, el desarrollo de interfaces de usuario intuitivas y funcionales, la aplicación de principios de seguridad informática para proteger la información sensible del negocio y de los usuarios, y la implementación de metodologías de desarrollo de software para asegurar un proceso eficiente y la entrega de un producto de calidad basado en un diseño Modelo-Vista-Controlador (MVC) .

Con el desarrollo e implementación de este proyecto, se busca impulsar a Todo Estilo para que esté a la vanguardia tecnológica, favoreciendo la optimización de sus procesos, esto no solo mejorará la gestión interna, sino que también sentará las bases para futuras innovaciones y un crecimiento sostenible en el tiempo. Adicional representa una oportunidad inigualable para el estudiante, para aplicar y consolidar las habilidades en un escenario real, pasando por todas las etapas del desarrollo de software, generando desafíos en cada una de estas etapas, haciendo una preparación previa para el ámbito profesional.

3. Objetivos

3.1. Objetivo general

Desarrollar una aplicación web que integre herramientas para la administración financiera, que garantice la correcta funcionalidad, de fácil uso y que integre los procesos de diseño, desarrollo, pruebas y mantenimiento, basados en estándares de calidad de software, y así optimice la gestión operativa y financiera del salón de belleza Todo Estilo.

3.2. Objetivos específicos

Levantar y analizar los requerimientos de usuarios, identificando los requisitos funcionales y no funcionales necesarios para el desarrollo de la aplicación.

Diseñar la estructura del sistema, incluyendo el diagrama entidad-relación (ER) para la base de datos, los mockups de la interfaz de usuario y la organización funcional de los módulos.

Desarrollar e implementar la aplicación web, con énfasis en la integración de un módulo financiero facilitando la supervisión de ingresos, egresos, control de pagos y gestión de las cuentas por cobrar, buscando mejorar el control financiero y contable del negocio.

Realizar pruebas técnicas y funcionales para garantizar el rendimiento, la seguridad y el correcto funcionamiento del sistema antes de realizar el despliegue.

Elaborar la documentación técnica y el manual de usuario, asegurando una implementación adecuada y facilitando su uso por parte de los empleados y clientes de Todo Estilo.

4. Marco teórico

4.1. Bases teóricas

Las bases teóricas constituyen la teoría que según Hernández et al. (2014), es un conjunto de constructos (conceptos), definiciones y proposiciones relacionadas entre sí, que presentan un punto de vista sistemático de fenómenos especificando relaciones entre variables, con el objeto de explicar y predecir los fenómenos.

Desde la perspectiva cuantitativa para Hernández et al. (2014), una teoría desempeña un papel fundamental al describir, explicar y tratar de predecir un fenómeno, contexto, evento o hecho que se investiga. De igual forma, busca organizar sistemáticamente el conocimiento existente sobre este y guiar el curso de la investigación. Una teoría amplía indudablemente nuestra comprensión de la realidad que se investiga.

4.1.1. Gestión financiera

Según Cevallos, (2010), la gestión financiera es el proceso con el cual se busca establecer un análisis estratégico de las Fortalezas, Oportunidades, Debilidades y Amenazas (FODA) en un entorno empresarial dinámico y en constante evolución, el análisis exhaustivo de las fortalezas y debilidades organizacionales resulta fundamental para la definición de las estrategias más idóneas en la consecución de los objetivos establecidos. Este tipo de procesos tienen como objetivo principal que la organización alcance una posición competitiva frente a otras entidades del mismo sector. El punto más relevante de la planificación estratégica está fundamentado en el carácter esencial para la prosperidad y el desarrollo sostenible de la organización a largo plazo, que permite, de manera consecuente, una dirección y orientación óptima de sus actividades.

(Cabrera Bravo et al., 2017)“El análisis financiero se basa en el cálculo de indicadores financieros que expresan la liquidez, solvencia, eficiencia operativa, endeudamiento, rendimiento y rentabilidad de una empresa. Se considera que una empresa con liquidez es solvente pero no siempre una empresa solvente posee liquidez. La gestión financiera basada en cifras ajustadas por inflación proporciona información financiera válida, actual, veraz y

precisa. Este análisis se basa en la interpretación de los sucesos financieros ocurridos en el desarrollo de la actividad empresarial, para lo cual utiliza técnicas que una vez aplicadas llevan a una toma de decisiones acertadas.(Pág 226, 227)

Los autores Weston & Copeland.(2012) proponen que la planeación financiera incluye la creación de indicadores de ventas, ingresos y patrimonio, fundamentados en estrategias de producción y comercialización, así como la identificación de los recursos necesarios para alcanzar las proyecciones establecidas.

Perdomo (2012) indica que, actualmente, tomar decisiones financieras tales como la compra de acciones o de bonos en cualquiera de sus tipos, apertura de cuentas bancarias, financiamientos, seguros, o en general cualquier otro instrumento que brinde el mercado financiero no puede ser precipitada, invertir es una meta constante, y considerar que algo es una buena inversión dependerá muchas veces del punto de vista del inversionista, de su aversión al riesgo y del tiempo en el cual mantendrá su excedente en instrumentos en este mercado, así como también del entorno que lo rodea.

Según las definiciones expuestas por los diferentes autores, se puede concluir que la gestión financiera es el proceso clave que permite a las empresas proyectarse a futuro mediante un análisis detallado y estratégico, basado en información histórica bien registrada. Este proceso no solo involucra la planificación de objetivos financieros a largo plazo, sino también el análisis de factores como el entorno económico, las fortalezas y debilidades internas, así como las oportunidades y amenazas externas. Con las proyecciones de ventas y activos, y la evaluación de riesgos, las organizaciones pueden tomar decisiones informadas sobre sus inversiones, financiación y crecimiento esperado. La gestión financiera, por tanto, no se limita a la contabilidad, sino que abarca la toma de decisiones estratégicas que favorecen la sostenibilidad y competitividad de la empresa, garantizando su adaptabilidad a los cambios del entorno económico. En resumen, una adecuada gestión financiera, respaldada por una información precisa y oportuna, es fundamental para lograr los objetivos empresariales y asegurar su estabilidad y crecimiento en el futuro.

4.1.2. Mecanismos de control

Para Rondón(2012), el control financiero es un mecanismo que permite analizar los resultados de una empresa para compararlos con los objetivos de negocio. Esto permite detectar problemas y ajustar los planes de la empresa.

Los mecanismos de control financiero más comunes son el control presupuestario, control de efectivo y liquidez, control de riesgos financieros y control financiero permanente.

Según el autor mencion los objetivos del control financiero son:

- Controlar que se cumplan los objetivos de la empresa
- Detectar problemas o desajustes
- Garantizar que se están siguiendo los planes de negocio
- Modificar los planes de negocio en caso de desviaciones
- Maximizar el beneficio y el valor de la empresa
- Brindar confiabilidad en la información financiera

4.1.3. Análisis financiero

Moyer et al., (2005) explican que el Análisis Financiero es un proceso de selección, relación y evaluación de información financiera compuesto por técnicas y herramientas contables cuyo objetivo es proporcionar sólidas bases de conocimientos para tomar decisiones en el mundo de los negocios.

Gispert (2010), señala que la recopilación de datos, antecedentes, apuntes e informes descriptivos y numéricos significativos, son objeto de estudio y evaluación, por un lado, de las partes que integran el todo, y por otro, estudiando y evaluando conceptos y cifras económicas y financieras del pasado de la empresa.

Existen métodos y técnicas para el análisis económico - financiero de datos significativos para tomar decisiones. Entre las técnicas e instrumentos necesarios para el análisis financiero y la toma de decisiones, se encuentran el análisis horizontal y el análisis vertical.

4.1.4 Método horizontal

“El análisis horizontal compara los resultados obtenidos en un periodo específico y los

resultados alcanzados en períodos anteriores. Este se puede realizar tanto en términos absolutos como porcentuales, para conocer las variaciones que se presentaron en cada una de ellas, a diferencia del análisis vertical que realiza el comparativo con cifras del mismo año que se está analizando.” (Agudelo Aguirre, 2021,152).

Tabla. 1
Ejemplo de análisis horizontal

Detalle	Año ^{a, b}			
	2015	2016	2017	2018
Ingresos operacionales (\$)	7912003	8517382	8532913	8417604
Costo de ventas (\$)	6097927	6595353	6970156	6852288
Utilidad bruta	1814076	1922029	1562757	1565316
Gastos operacionales (\$)				
Administración	598662	699310	683835	625529
Ventas	264387	263373	254229	260317
Otros ingresos y gastos operativos	(11842)	23015	69556	145641
Utilidad operacional	939185	982361	694249	825111
Otros ingresos (gastos) no operativos (\$)				
Gastos financieros	274963	340828	406094	414638

Nota: Estado de resultados integral Cementos Argos año 2018 y anteriores
Fuente: Información tomada de (Agudelo Aguirre, 2021, 152)

4.1.5. Método vertical

El análisis vertical, según Agudelo Aguirre (2021), “Consiste en obtener relaciones entre los valores de cada una de las cuentas que componen el estado de resultados integral o el estado de situación financiera, con el valor de una cuenta que se toma como referente general. Cuando el análisis se enfoca en el estado de resultados, la cuenta que es tomada como referencia en la mayoría de los casos suele ser el valor total de los ingresos del periodo. Mientras que, si se analiza el estado de situación financiera, se acostumbra a tomar como referencia el valor del total de activos para el análisis de las cuentas del activo o el total de pasivos más patrimonio para cada una de las cuentas que componen el lado derecho del estado de situación financiera, es decir, el pasivo y el patrimonio. La importancia de este análisis radica en la determinación del porcentaje que representa cada una de las cuentas

examinadas, respecto al valor total, con lo que se obtiene una idea general de la importancia relativa de cada una de ellas.” (P. 150)

Tabla. 2

Ejemplo de análisis vertical

Estado de resultados integral Cementos Argos Entre el 01 de enero y el 31 de diciembre Cifras en millones de pesos y porcentaje		
	2018 (\$)	2018 (%)
Ingresos operacionales	8 417 604	100.00
Costo de ventas	6 852 288	81.40
Utilidad bruta	1 565 316	18.60
Gastos operacionales		
Administración	625 529	7.43
Ventas	260 317	3.09
Otros ingresos y gastos operativos	145 641	1.73
Utilidad operacional	825 111	9.80
Otros ingresos (gastos) no operativos		
Gastos financieros	414 638	4.93
Otros ingresos	16 193	0.19
Otros gastos	12 203	0.14
Utilidad antes de impuesto de renta	414 463	4.92
Provisión impuesto renta	123 029	1.46
Utilidad neta	291 434	3.46

Nota: Estado de resultados integral Cementos Argos año 2018

Fuente: Información tomada de (Agudelo Aguirre, 2021, 152)

4.1.6. Ventajas de la Digitalización Empresarial

La digitalización es una herramienta clave para organizar y transformar documentos, registros y facturas, facilitando el manejo de la información. Entre sus beneficios, destaca la rapidez en el acceso a los datos, ya que los documentos digitales son fácilmente localizables en una computadora, en la nube o en un programa, evitando largas búsquedas en archivos físicos. Además, reduce el espacio necesario para almacenar papeles y optimiza el tiempo destinado a organizar archivos en una empresa. Desde una perspectiva ambiental, la digitalización también disminuye el uso de papel, lo que contribuye a reducir los costos operativos de la empresa.

El mismo sistema de transformación digital que el mundo entero está desarrollando ha obligado a que muchas empresas tengan que adaptarse a esta nueva era de completa

optimización. La transformación digital implica integrar tecnologías y soluciones digitales en todas las áreas de un negocio. Este es tanto un cambio cultural como tecnológico, ya que requiere que las organizaciones hagan cambios fundamentales en la forma en que operan y en el modo en que brindan experiencias y beneficios al cliente. Las soluciones digitales también ayudan a aumentar la fuerza laboral y pueden conducir a la transformación de los procesos de negocio y del modelo de negocios. (SAP, 2021).

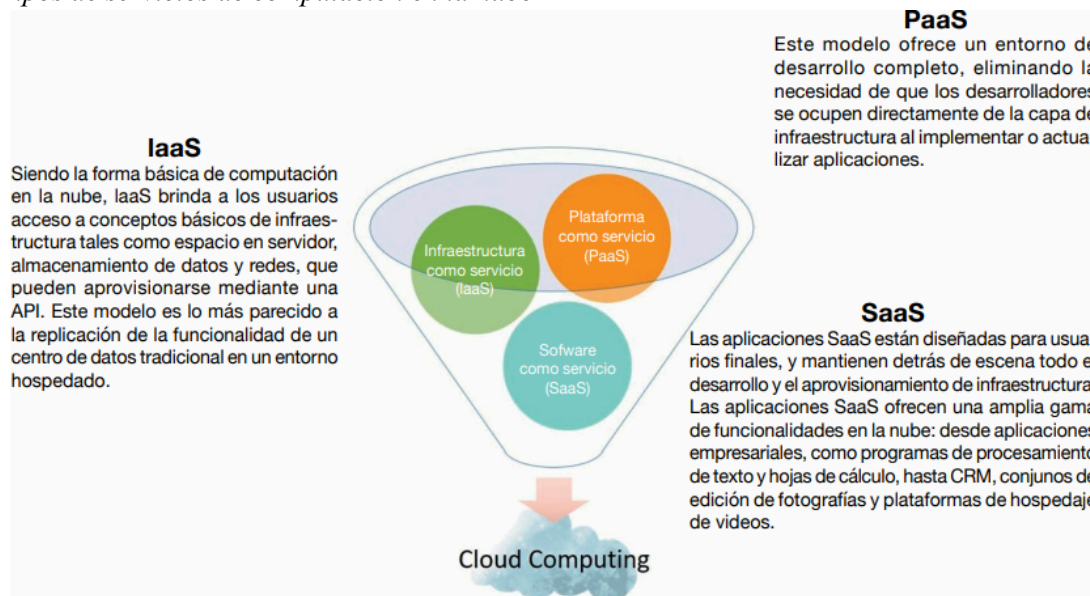
La integración de tecnologías digitales en las empresas, conocida como transformación digital, según (Páez Gabriunas et al., 2022) abarca desde el uso de sistemas para manejar información hasta la innovación en productos y servicios digitales, como ventas online o consultoría virtual. Este cambio representa un desafío para los líderes de empresas, quienes deben equilibrar el avance tecnológico con el bienestar organizacional. Este proceso comenzó con la digitalización cotidiana y se aceleró con la cuarta revolución industrial y la pandemia por el Covid 19, caracterizada por tecnologías como la inteligencia artificial, la robótica y el internet de las cosas. Hoy en día, es indispensable para que las organizaciones sean competitivas y sostenibles. De hecho, investigaciones muestran que la mayoría de las empresas, a nivel global y en Colombia, ya están trabajando en estrategias para transformarse digitalmente.

4.1.7. Computación en la nube

La computación en la nube o cloud computing es también una tecnología clave para la transformación digital y el desarrollo de la sociedad actual. Consiste en un modelo que permite acceder, de manera remota y bajo demanda, a recursos computacionales compartidos como redes, servidores, almacenamiento y software. Estos recursos, gestionados de forma virtual a través de proveedores, pueden ajustarse rápidamente según las necesidades de las empresas, facilitando la escalabilidad y optimización de costos. Esta tecnología ofrece flexibilidad, innovación más ágil y economías de escala, reemplazando la infraestructura local con soluciones accesibles en línea (Sanabria et al., 2022, 43-44).

Ilustración. 2

Tipos de servicios de computación en la nube



Nota: Los 3 tipos esenciales de servicios en los que se fundamenta la computación en la nube, tomado de (Sanabria et al., 2022, 44).

4.1.8. Metodologías ágiles.

Para Sanabria et al.(2022), las metodologías ágiles nacen como una evolución de los enfoques tradicionales para optimizar los procesos de desarrollo de software, especialmente en proyectos donde los requisitos no son estables y se comportan de una forma cambiante durante el proceso, esta metodología se enfoca en priorizar la flexibilidad, lo simple y la capacidad de adaptación, permitiendo a los equipos responder a las demandas del cliente. A diferencia de los métodos tradicionales, la metodología ágil reduce la carga administrativa al evitar tareas con valor dudoso a largo plazo y minimizar la documentación innecesaria. El objetivo principal es ir entregando funcionalidades de manera continua en tiempos cortos, promoviendo la colaboración activa entre los miembros del equipo y los clientes para ir incorporando mejoras en iteraciones posteriores.

Basadas en los principios del manifiesto ágil, estas metodologías destacan por:

- Individuos e interacciones por encima de procesos y herramientas.
- Software funcional por encima de documentación extensiva.
- Colaboración con el cliente por encima de negociación contractual.
- Respuesta ante el cambio por encima de seguir un plan.

En esencia, las metodologías ágiles representan un enfoque dinámico que simplifica procesos, promueve la interacción constante y garantiza soluciones efectivas frente a las demandas cambiantes de los proyectos (Sanabria et al., 2022, 207).

En este proyecto, el desarrollo se llevará a cabo mediante sprints semanales, donde cada semana se planificará, desarrollará y presentará una funcionalidad específica de la aplicación web. Al final de cada sprint, se realizará una revisión con el equipo y los interesados para evaluar el avance, recibir retroalimentación y ajustar el backlog si es necesario. Este enfoque permite un desarrollo incremental y adaptable, asegurando que las funcionalidades entregadas respondan a las necesidades reales del negocio y que el proyecto evolucione de forma ordenada y eficiente.

4.1.9. Herramientas para el desarrollo.

En el desarrollo web moderno existen diversas herramientas, lenguajes y tecnologías que permiten construir aplicaciones eficientes, dinámicas y adaptadas a las necesidades de los usuarios y las empresas. Estas herramientas cumplen diferentes funciones dentro del proceso de creación de una aplicación web: desde la estructuración del contenido, hasta la programación del comportamiento y la interacción con bases de datos.

Entre las más utilizadas se encuentran HTML5, que se encarga de la estructura y presentación básica de las páginas; PHP, como lenguaje de programación del lado del servidor para gestionar la lógica de las aplicaciones; y Laravel, un framework basado en PHP que permite optimizar el desarrollo mediante buenas prácticas, seguridad y organización del código.

El uso adecuado y combinado de estas tecnologías permite crear aplicaciones web modernas, responsivas, funcionales y escalables, facilitando la experiencia del usuario y mejorando los procesos internos de las organizaciones. A continuación, se describen las características y funciones más relevantes de cada una de estas herramientas fundamentales para el desarrollo del proyecto.

4.1.9.1. Programación Orientada a Objetos (POO)

Los autores Velarde de Barraza et al., (2006) definen la Programación Orientada a

Objetos (POO) como un conjunto de técnicas que se pueden aplicar en el desarrollo de software, actualmente viene integrado a la mayoría de los lenguajes de programación, sus principales objetivos son incrementar la productividad en el desarrollo, permitir abordar proyectos de mayor complejidad, facilitar el desarrollo y mantenimiento de los programas mediante la reutilización de código y la organización en entidades llamadas objetos. No es un lenguaje de programación, sino una forma de conceptualizar y estructurar la programación.

La programación orientada a objetos se puede definir como una técnica o estilo de programación que utiliza objetos como bloques esenciales de construcción. Los elementos básicos de la POO son: objetos, mensajes, métodos y clases. (Velarde de Barraza et al., 2006, Pág 9).

Características de la programación orientada a objetos

Morero, (2000) define las siguientes características para la programación orientada a objetos:

- Basada en objetos: esta metodología se centra en la manipulación de objetos reales animados, cada objeto se compone de atributos y funciones (métodos) que operan sobre esos datos.
- Clases: Los objetos se crean a partir de clases, que son como plantillas o planos que definen la estructura y el comportamiento de los objetos según el tipo. La clase sirve como modelo para crear otros objetos con características similares, estos objetos comparten los mismos tipos de información (atributos o propiedades) y los mismos comportamientos (métodos o funciones).
- Encapsulación: Los datos y los métodos de un objeto están intrínsecamente ligados, formando una unidad conceptual y operacional. Esta es una característica que ayuda a proteger los datos internos del objeto y a controlar cómo se accede a ellos, para evitar que se tenga acceso directo e incontrolado desde otras partes del programa.
- Herencia: permite que una clase hija hereda características (atributos y/o métodos) de otra clase padre. Esto ayuda a fomentar la reutilización de código y la creación de jerarquías de clases.

- Polimorfismo: permite usar las mismas funciones o métodos por parte de diferentes objetos, pero con distintos comportamientos según el tipo de objeto que lo llame. Es una forma de escribir código más flexible y reutilizable.
- Reusabilidad: los objetos bien diseñados pueden ser utilizados en diferentes partes del código de un programa o incluso en otros proyectos, lo que ahorra tiempo y esfuerzo en el desarrollo.
- Abstracción: Consiste en ocultar los detalles internos de cómo funciona algo y mostrar solo lo necesario para su uso.

4.1.9.2. Modelo Vista Controlador (MVC)

(España León et al., 2016) “El Modelo Vista Controlador (MVC) fue definido por Trygve Reenskaug a finales de los 70 en laboratorios de investigación de Xerox e introducido como parte de la versión Smalltalk-80 del lenguaje de programación Smalltalk, con la finalidad de reducir el esfuerzo de programación en la implementación de sistemas múltiples y sincronización de los datos. Se caracteriza por tratar al Modelo, las Vistas y los Controladores como entidades separadas, lo que permite realizar cualquier cambio en el Modelo y este se refleja automáticamente en cada una de las vistas asociadas a él. Este modelo de arquitectura se puede emplear en software de representación gráfica de datos, donde se presentan partes del diseño con diferente escala de aumento, en ventanas separadas.

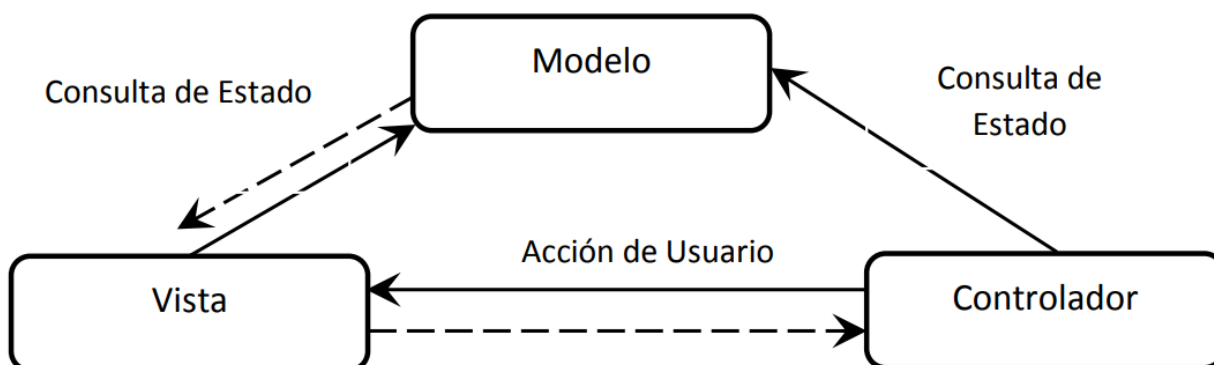
Entre las principales ventajas que presenta la utilización del patrón MVC al momento de desarrollar una aplicación bajo esta arquitectura se describen a continuación:

- Permitir la sustitución de las interfaces de usuario
- Generar componentes de las interfaces
- Diseñar vistas simultáneas del mismo modelo
- Aplicar fácilmente cambios de las interfaces

Se puede concluir que el MVC es uno de los patrones más utilizados por desarrolladores de software, porque permite organizar los componentes aplicando ciertas normas de diseño para fortalecer su utilidad y prepararlo para su evolución.” (Pág 564,565)

Ilustración. 3

Esquema de cómo se encuentran relacionadas las tres capas de aplicación.



Nota: Modelo Vista Controlador.

Fuente: Tomado de (España León et al., 2016, 564).

4.1.9.3. HTML5

HTML5 es un lenguaje de etiquetas que permite diseñar documentos web estáticos, con mejoras significativas en comparación con versiones anteriores. Introduce un conjunto de funciones avanzadas para ofrecer una experiencia más dinámica y eficiente en el diseño web.

Según (Torres Remon, 2014) sus principales funciones son:

- Semántica: Optimiza la distribución de elementos, lo que aporta precisión al contenido.
- Conectividad: Facilita una comunicación más ágil con servidores web sin consumir muchos recursos.
- Servicio Local: Permite navegar en un sitio sin conexión, gracias a la ejecución local desde el cliente.
- Multimedia: Simplifica la incorporación de vídeos y música mediante etiquetas específicas.
- Alto Rendimiento: Optimiza la velocidad de procesamiento en el hardware del cliente.
- Multiplataforma: Asegura una visualización uniforme en diversos dispositivos.

HTML5 introduce nuevas etiquetas estructurales que mejoran la organización y dinamismo de las páginas web, tales como:

- Header (encabezado)

- Nav (navegación)
- Article (contenido principal)
- Section (sección temática)
- Aside (información complementaria)
- Footer (pie de página)

Estas etiquetas rediseñan el flujo de los elementos para un desarrollo más intuitivo y dinámico, adaptándose a las demandas de dispositivos modernos.

4.1.9.4. PHP

PHP es un lenguaje de programación orientado a la creación de aplicaciones web dinámicas, que permite intercalar con HTML para simplificar la construcción de páginas web. Es un lenguaje interpretado (no compilado) que opera mediante servidores como Apache y es compatible con la mayoría de sistemas operativos y servidores web. Su sintaxis está influenciada por lenguajes como C, Java y Perl.

PHP es de código abierto y fue creado por Rasmus Lerdorf en 1994, inicialmente como "Personal Home Page" para gestionar información personal. En 1997, fue mejorado por Zeev Suraski y Andi Gutmans, transformándose en "HyperText Preprocessor" (PHP).

Evolución de PHP:

- PHP 3.0 (1998):
 - Popularidad global (instalado en más del 80% de los servidores web).
 - Introdujo soporte para programación orientada a objetos y extensibilidad.
 - Capacidad de integrarse con bases de datos y protocolos diversos.
- PHP 4.0 (2000):
 - Mejoras en velocidad y confiabilidad.
 - Introdujo el motor Zend Engine 1.0, soporte COM, manejo de sesiones HTTP, entre otras características.
- PHP 5.0 (2004):
 - Nueva versión del Zend Engine 2.0.
 - Incorporación de SQLite, soporte avanzado para MySQL, manejo de excepciones (try-catch) y soporte SOAP.
- PHP 6.0 (2006 - en desarrollo):
 - Introducción del soporte nativo para Unicode, ampliando su alcance más allá del desarrollo web hacia scripting en línea de comandos y aplicaciones gráficas mediante GTK+.

PHP sigue siendo una herramienta fundamental en el desarrollo web moderno debido a su

flexibilidad y amplio soporte en servidores y plataformas (Flórez Fernández & Hernández Rodríguez, 2021, 3-4).

4.1.9.5. Laravel

Laravel es un framework de código abierto para el desarrollo de aplicaciones web, basado en el lenguaje de programación PHP. Fue creado por Taylor Otwell en 2011 con el objetivo de ofrecer una sintaxis limpia, moderna y expresiva que facilitara las tareas comunes del desarrollo web, como el enrutamiento, la autenticación, la gestión de sesiones y el acceso a bases de datos. Su enfoque se centra en mejorar la experiencia del desarrollador mediante una arquitectura sólida y herramientas integradas que permiten construir aplicaciones robustas en menor tiempo (Moreno & García, 2019).

Laravel se fundamenta en el patrón arquitectónico Modelo-Vista-Controlador (MVC), lo cual permite separar claramente la lógica de negocio, la interfaz del usuario y la gestión de datos. Esta estructura facilita la escalabilidad, el mantenimiento y la reutilización del código. Además, incluye componentes destacados como Blade (su motor de plantillas), Eloquent ORM (para interactuar con bases de datos), migraciones de esquema, rutas personalizables, y Artisan, una herramienta de línea de comandos para agilizar el desarrollo de tareas repetitivas (Hernández & Rodríguez, 2020).

4.2. Antecedentes

4.2.1 Soluciones comerciales existentes

Para el manejo financiero y contable existen muchas herramientas que pueden servir como guía o antecedente para el desarrollo de del presente proyecto, entre todos estos proyectos destacan algunos como por ejemplo intuit quickbooks y alegra, herramientas que ofrecen una solución amigable y de fácil manejo para el control financiero y contable.

Quickbooks: es un software contable para pymes que tiene operación a nivel mundial, en su sitio oficial <https://quickbooks.intuit.com/> se puede obtener información de sus funcionalidades, ofertas y servicios.

Quickbooks nació hace más de 28 años en California, EE.UU, pero cuenta con clientes a nivel mundial, incluso tiene testimonios de empresarios colombianos que han tenido éxito

con el manejo de su plataforma.

Ilustración. 4

Testimonio de éxito de la aplicación Quickbooks en Colombia.



Juliana es una creativa que decidió emprender un negocio de arreglos florales con la idea de que todos los seres humanos necesitamos flores en nuestra vida. Actualmente, trabaja como fundadora y directora creativa de su empresa con un equipo de 12 personas a su cargo. Al ahorrar tiempo en la gestión de gastos y facturación, ayudamos a clientes como Juliana a trabajar de forma más eficiente para que puedan centrarse en sus clientes. Ora ofrece servicios de arreglos florales para eventos que van desde bodas hasta sesiones de fotos. Esta marca evolucionó a partir de una pequeña idea y, desde entonces, se ha asociado con MasterCard y ha aparecido en Domestika. Descubre más sobre Ora, la Agencia Floral, [aquí](#).

Nota: captura de pantalla tomada de la página oficial de Quickbooks(<https://quickbooks.intuit.com>).

Dentro de las principales propuestas de valor que ofrece esta aplicación están las siguientes, tomadas del sitio oficial (<https://quickbooks.intuit.com/>):

- Monitorea los egresos, personaliza las facturas, genera reportes y más, todo en un solo lugar.
- Administra tus facturas en movimiento con la aplicación móvil de QuickBooks: realiza un seguimiento de las facturas para saber quién ha pagado. Envía recordatorios automáticos para pagos tardíos. Cambia rápidamente de presupuestos a facturas, envía facturas recurrentes y ejecuta informes instantáneos en cualquier lugar y desde cualquier dispositivo.
- Para ahorrar tiempo y reducir errores, conecta tu banco a QuickBooks. No tiene costo

adicional, y puedes sincronizar cuentas en minutos.

- Ve cómo le está yendo a tu empresa con una variedad de reportes populares. O personaliza uno para incluir detalles que te interesen.

Alegra: este es otro gran ejemplo de una plataforma de contabilidad y facturación, pero un poco más local ya que es una herramienta que fue desarrollada hace aproximadamente 12 años por Jorge Soto y Santiago Villegas en Medellín, Colombia y ha extendido sus servicios a otros países latinos como Perú, Ecuador y México.

Ilustración. 5

Página principal del software contable alegra.



Nota: captura de pantalla tomada de la página oficial de Alegra (www.alegra.com).

Las propuestas de valor que ofrece este software se centra principalmente en dos ideas que resaltan en la página principal www.alegra.com, estas dos ideas son:

- El poder de hacer más en menos tiempo.
- El poder de hacer tareas en automático.

En la página oficial enseñan las características que tiene cada una de estas, con 5 características principales para cada idea, a continuación se describen como los enseñan en su página:

El poder de hacer más en menos tiempo:

1. Facturación electrónica sin esfuerzo: Crea facturas y genera automáticamente los eventos de facturación electrónica con IA. Cumple con la DIAN sin complicaciones,

sin importar el monto de las facturas.

2. Documentos y reportes al instante: Emite Documentos de Soporte Electrónico sin límites y genera reportes de Información Exógena al instante.
3. Gestión de inventarios centralizada: Controla tu stock y bodegas en un solo lugar.
4. Automatiza la creación y gestión de contactos: Crea contactos en segundos desde imágenes o PDFs con IA o sincronizar información con la DIAN de forma automática y precisa.
5. Protege tu información: Autoguarda tus facturas de venta y evita pérdidas ante cualquier falla (Alegra,2024) recuperado de la página oficial(<https://www.alegra.com>)

El poder de hacer tareas en automático:

1. Contabilidad y conciliación inteligente: Automatiza el registro contable de tus facturas y realiza conciliación bancaria automática cruzando tus movimientos con la contabilidad, todo impulsado por IA.
2. Reportes y análisis en tiempo real: Accede a reportes contables y de ventas con análisis y tendencias inteligentes.
3. Cobranza y recordatorios automáticos: Configura notificaciones de saldos pendientes y optimiza tu flujo de caja.
4. Recepción y registro automático de facturas: Recibe y registra facturas electrónicas de proveedores con nuestro Buzón Inteligente.
5. Atiende y factura desde WhatsApp: Gestiona clientes y genera facturas al instante con nuestro asistente virtual impulsado por IA (Alegra,2024) recuperado de la página oficial (<https://www.alegra.com>).

4.2.2 Proyectos académicos y universitarios similares

Se realiza la investigación proyectos académicos donde centran en la solución de problemas basados principalmente en la implementación y desarrollo de herramientas web, como por ejemplo el proyecto titulado DESARROLLO DE UN SISTEMA WEB DE GESTIÓN FINANCIERA PARA LA FUNDACIÓN “JASPE”, donde sus autores Morocho Guashpa William Patricio y Carranza Lucas Jerson Kenny, desarrollan el proyecto como trabajo de grado para obtener el título de ingeniero en sistemas informáticos en la Escuela Superior Politécnica de Chimborazo y sirve como pilar para el desarrollo del presente proyecto ya que brinda luces en

cuanto a las etapas de desarrollo al llevarse a cabo en una metodología ágil que a pesar de ser diferente maneja etapas para el desarrollo del proyecto similares.

Como lo describen sus autores (Orocho Guashpa & Carranza Lucas, 2017) el proyecto consistió en el desarrollo de un sistema web de gestión financiera para la Fundación "JASPE" utilizando la tecnología de base de datos No-SQL, El objetivo principal fue automatizar los procesos de gestión financiera de la fundación, lo que se esperaba que ayudara a la toma de decisiones y solucionar la demora en la entrega de información que existía previamente, cuando los procesos se manejaban de forma manual utilizando software como Word y Excel.

El alcance del sistema incluyó el control de procesos administrativos como el cobro de matrícula y pensiones de estudiantes, y el control de pagos a profesores, estos procesos están directamente relacionados con los módulos de contabilidad, que abarcan el libro diario, libro mayor, flujo de caja y balance de comprobación.

Desarrollo:

El proceso de desarrollo siguió las seis fases de la metodología XP: Exploración, Planificación, Diseño, Codificación, Pruebas y Finalización.

La fase de exploración incluyó la recopilación de información mediante observación y entrevistas para diagnosticar la situación actual de la fundación y gestionar posibles riesgos, se identificaron riesgos como poca explicación del usuario, mal diseño de la base de datos, cambios continuos en requisitos, interfaces complejas, incumplimiento de entregas, cambio de autoridades, falta de apoyo y no usar un estándar de programación.

La fase de planificación implicó la definición de los requisitos del usuario como historias de usuario, escritas en lenguaje no técnico junto al cliente, y la creación de un plan de lanzamiento (release planning) que incluía fechas para cada historia.

La fase de diseño se enfocó en el diseño simple y la creación de metáforas del sistema, que permitieran una fácil implementación y desarrollo. Estas metáforas incluyeron la Arquitectura del sistema, el estándar de codificación, el diseño de la base de datos y el diseño de la interfaz de usuario.

La fase de codificación dividió el desarrollo en seis iteraciones. Cada iteración se centró en desarrollar un conjunto de historias de usuario, documentadas con fichas técnicas que detallan el ID, nombre, puntos estimados (horas) y tareas de ingeniería.

La Fase de Pruebas es uno de los pilares de la metodología XP. Se realizaron pruebas de aceptación para verificar que las historias de usuario cumplieran con la funcionalidad especificada.

La Fase de Finalización consistió en la capacitación de los usuarios y la presentación de la documentación, incluyendo un manual de usuario (OROCHO GUASHPA & CARRANZA LUCAS, 2017).

5. Metodología

5.1 Tipo de proyecto

El presente proyecto se clasifica como una investigación aplicada, ya que su objetivo principal es abordar un problema práctico y específico identificado en el contexto real del salón de belleza Todo Estilo. A diferencia de la investigación pura, que busca la generación de nuevo conocimiento teórico, este proyecto se enfoca en la aplicación de conocimientos existentes y metodologías de desarrollo de software para crear una solución tangible y funcional.

El enfoque principal del proyecto es el desarrollo tecnológico. Esto implica un proceso sistemático que abarca el análisis de requerimientos, el diseño de la arquitectura de la aplicación web (tanto frontend como backend), la implementación de las funcionalidades identificadas, las pruebas exhaustivas para garantizar su correcto funcionamiento y la documentación del sistema desarrollado. El resultado final esperado es una aplicación web operativa y adaptable a las necesidades específicas de Todo Estilo.

No obstante, el proceso de desarrollo tecnológico estará precedido y acompañado de un componente descriptivo. En la fase inicial del proyecto, se llevará a cabo una investigación para comprender en profundidad los procesos administrativos y contables actuales del salón, las herramientas que utilizan, los desafíos que enfrentan y las necesidades específicas que la aplicación web debe abordar. Esta descripción detallada de la situación actual, obtenida a través de la observación y las entrevistas, será fundamental para definir los

requerimientos funcionales y no funcionales de la aplicación, asegurando que la solución desarrollada sea pertinente y responda eficazmente a las problemáticas identificadas.

En resumen, el proyecto se concibe como una investigación aplicada que utiliza el desarrollo tecnológico como estrategia principal para crear una solución práctica. Este proceso estará informado por una fase descriptiva inicial que permitirá comprender el contexto y las necesidades específicas del caso de estudio.

5.2 Método

La realización del presente proyecto de grado se abordará mediante una estrategia metodológica que integra elementos del análisis y síntesis, la deducción, la observación, y un enfoque de desarrollo iterativo y ágil. Estos métodos operarán de la siguiente manera para dar respuesta a la pregunta de investigación sobre ¿cómo desarrollar una aplicación web que optimice la gestión administrativa y contable del salón de belleza Todo Estilo?.

5.2.1 Fase de inicio y levantamiento de requerimientos:

Se iniciará con una observación detallada de los procesos administrativos y contables actuales del salón de belleza Todo Estilo. Esto implicará la inmersión en las operaciones diarias para identificar los flujos de trabajo, las herramientas utilizadas, los puntos de ineficiencia y las áreas donde la tecnología podría aportar valor.

Simultáneamente, se llevará a cabo un análisis exhaustivo de la información recopilada a través de la observación, las entrevistas con el personal operativo y, directamente al área administrativa. Este análisis permitirá descomponer el problema de la gestión en sus componentes clave, identificar las causas subyacentes y definir los requerimientos funcionales y no funcionales de la aplicación web. Se utilizarán herramientas como diagramas de flujo y modelos de procesos para visualizar y comprender la información.

5.2.2 Fase de diseño conceptual:

A partir de los requerimientos identificados y los principios generales del diseño de software, la gestión de bases de datos, la seguridad web y la usabilidad (que resultan de la apropiación del saber teórico [Marco teórico]), se procederá a la síntesis de una arquitectura

conceptual para la aplicación web. Esto incluirá la definición de los módulos principales, la estructura de la base de datos y la creación de wireframes y mockups de la interfaz de usuario (UI) para visualizar la disposición de los elementos y el flujo de navegación. Se utilizarán herramientas de modelado UML o similares para representar el diseño de la arquitectura.

5.2.3 Fase de desarrollo ágil por iteraciones (sprints):

La construcción de la aplicación web se realizará mediante un enfoque de desarrollo iterativo y ágil. Esto implica dividir el proyecto en ciclos cortos (iteraciones o sprints) donde se diseñarán, desarrollarán, probarán y revisarán funcionalidades específicas de la aplicación.

En cada iteración, se aplicarán los principios de síntesis para integrar los diferentes componentes de la aplicación (frontend, backend, base de datos) en una solución funcional parcial. La retroalimentación obtenida de las pruebas internas y, potencialmente, de los usuarios en cada iteración, se utilizará para realizar ajustes y mejoras en las siguientes iteraciones.

5.2.4 Fase de pruebas continuas y evaluación de resultados:

A lo largo del proceso de desarrollo, se realizarán pruebas continuas para asegurar la calidad, funcionalidad y usabilidad de la aplicación. Esto incluirá pruebas unitarias, pruebas de integración y pruebas de usuario con la muestra seleccionada de empleados.

El análisis de los resultados de las pruebas permitirá identificar errores, áreas de mejora y la efectividad de la aplicación para abordar los requerimientos definidos inicialmente.

5.2.5 Fase de implementación y documentación final:

Una vez que la aplicación alcance un nivel de funcionalidad y estabilidad adecuado, se procederá a su implementación en el entorno del salón de belleza.

Se realizará la síntesis final de toda la información relevante del proyecto, incluyendo la documentación técnica de la aplicación, manuales de usuario y un informe final que detalle el proceso de desarrollo, los resultados obtenidos y las conclusiones.

5.3 Instrumentos de recolección de información

5.3.1 Fuentes primarias:

Observación directa: Diagnóstico del proceso actual mediante la participación y análisis

detallado de la operación.

Entrevistas: Aplicadas a empleados para validar el funcionamiento del sistema actual y recoger ideas sobre mejoras.

Encuestas: Instrumento clave para recopilar información sobre la experiencia y percepción del sistema desde múltiples perspectivas.

5.3.2 Fuentes secundarias:

Revisión de literatura: consulta de bases de datos, publicaciones y estudios sobre metodologías de diseño y optimización de aplicaciones web.

Casos de éxito: análisis de tecnologías y procesos implementados en proyectos similares dentro del sector de belleza.

Tendencias tecnológicas: identificación de innovaciones en el diseño de soluciones digitales y experiencia del usuario.

5.4 Plan de actividades o cronogram

Tabla. 3
Cronograma de actividades.

DESARROLLO DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN FINANCIERA DEL SALÓN DE BELLEZA TODO ESTILO.																	RESPONSABLE	
OBJETIVO GENERAL: Desarrollar una aplicación web que integre herramientas para la administración financiera, asegurando su correcta funcionalidad y facilidad de uso mediante procesos de diseño, desarrollo y mantenimiento basados en estándares de calidad, con el propósito de optimizar la gestión operativa y financiera del salón de belleza Todo Estilo.																		
OBJETIVOS ESPECIFICOS	ACTIVIDADES	MARZO		ABRIL				MAYO				JUNIO						
		SEMANAS																
		1	2	3	4	5	6	7	8	9	10	11	12	13				
1. Levantar y analizar los requerimientos del sistema, identificando los requisitos funcionales y no funcionales necesarios para el desarrollo de la aplicación web	1.1 - Revisión y ajustes del anteproyecto, subir anteproyecto y definir cronograma	█	█														Yeison Pineda	Pendiente
	1.2 - Estructurar documento final			█	█	█	█	█	█	█	█	█					Yeison Pineda	Terminada
	1.3 - Recopilación de información			█													Yeison Pineda	En proceso
	1.4 - Definir requerimientos funcionales y no funcionales				█												Yeison Pineda	Atrasada
	1.5 - Análisis y aprobación de requerimientos				█												Yeison Pineda	
2. Diseñar la estructura del sistema, incluyendo el diagrama entidad-relación (ER) para la base de datos, los mockups de la interfaz de usuario y la organización funcional de los módulos	2.1 - Diagramas de casos de uso				█												Yeison Pineda	
	2.2 - Modelo relacional y diagrama de clases				█												Yeison Pineda	
	2.3 - Mockups				█	█											Yeison Pineda	
3. Desarrollar e implementar la aplicación web, con énfasis en la integración de un módulo financiero que gestione ingresos, egresos, ventas y reportes contables con seguridad y precisión.	3.1 - Creación y conexión de base de datos						█	█									Yeison Pineda	
	3.2 - Desarrollo de informes								█	█							Yeison Pineda	
	3.3 - Desarrollo del código fuente									█	█						Yeison Pineda	
4. Realizar pruebas técnicas y funcionales para garantizar el rendimiento, la seguridad y el correcto funcionamiento del sistema antes de su lanzamiento.	3.1 - Pruebas técnicas y de software											█	█				Yeison Pineda	
5. Elaborar la documentación técnica y el manual de usuario, asegurando una implementación adecuada y facilitando su uso por parte de los empleados y clientes de Todo Estilo.	3.1. Actividad 7								█	█	█	█					Yeison Pineda	

Nota: Cronograma de actividades en base a los objetivos planteados.
Fuente: Elaboración propia.

6. Resultados

6.2 Fase de inicio y levantamiento de requerimientos

Dentro de la fase inicial lo primero que se realizó fue la observación de cómo se realizan los procesos actualmente, donde se puede evidenciar que el registros de los servicios, ingresos y salidas se realiza de una forma manual, llenando unas planillas o minuta donde se lleva el registro de los servicios prestados por cada estilista, en la planilla se registra el servicio prestado, la fecha, el valor y el porcentaje que corresponde a la estilista.

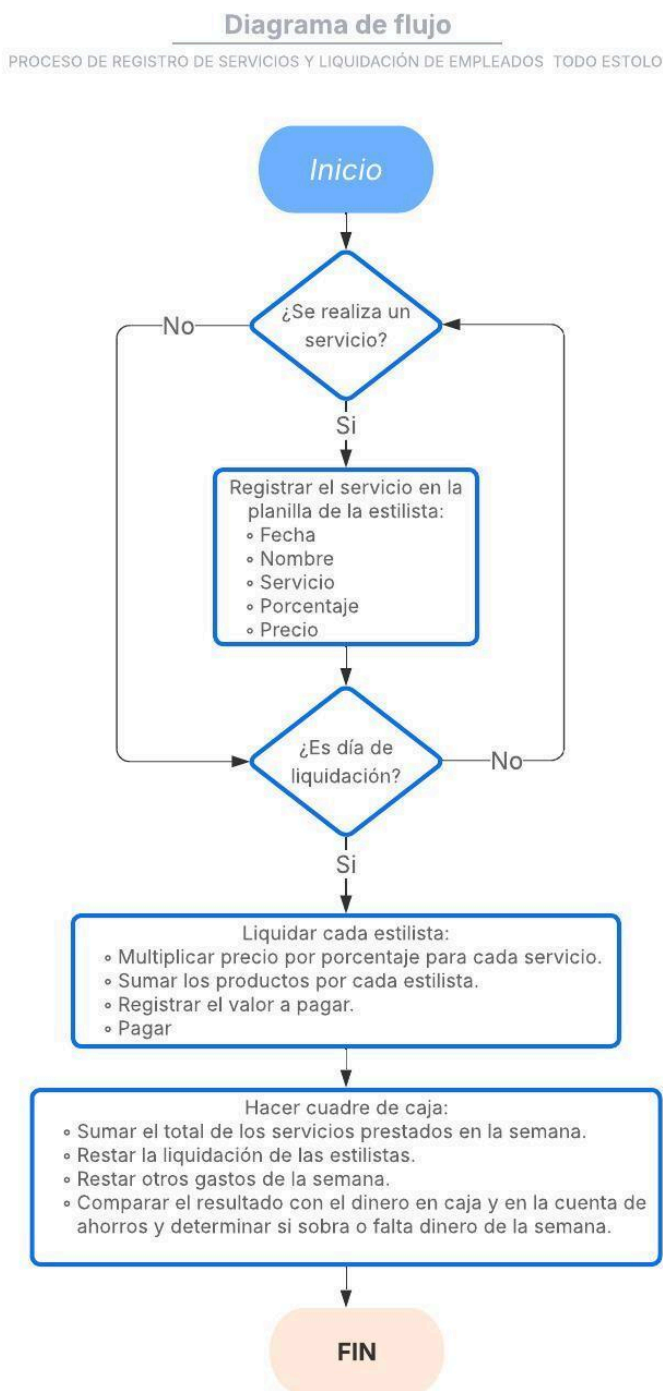
Con estos registros de forma semanal se hace la liquidación de empleados, realizando la sumatoria correspondiente a los resultados de multiplicar el valor de cada servicio por el porcentaje que corresponde y este es el valor a cancelar a la estilista. Posteriormente se realiza el cuadro de caja, que se hace con las mismas planillas, se suma el total de los servicios prestados y se restan tanto los pagos realizados a las empleadas como otros gastos adicionales que se hayan tenido en esa semana.

Como se puede observar es un proceso que se lleva a cabo de forma muy manual, presto a presentar errores e inconsistencias como:

- Errores en cálculos y registros: la dependencia de planillas manuales aumenta el riesgo de errores en sumas, multiplicaciones y registros incorrectos de datos. Estos errores pueden llevar a pagos incorrectos a las estilistas y discrepancias en la información financiera.
- Pérdida de información: la información almacenada en planillas puede perderse o corromperse, dificultando la recuperación de datos históricos y la generación de informes precisos.
- Demora en el proceso: al realizar todos los cálculos de forma manual, conlleva a demoras en los procesos de liquidación y de cuadro de caja.
- No hay trazabilidad: se dificulta el poder hacer seguimiento a la parte financiera del negocio, ingresos, egresos y otros.

Ilustración. 6

Diagrama de flujo de proceso actual.



Nota: Diagrama de flujo del proceso actual de registro de servicios, liquidación de empleados y cuadro de caja.

Fuente: Elaboración propia.

6.2.1 Requisitos de usuario

Tabla. 4

Requisitos de usuario.

ID Requisito	Nombre del requisito	Descripción
RU-001	Registro de empleados	El sistema debe permitir registrar a los empleados con su nombre, teléfono, dirección y correo.
RU-002	Registro de ingresos por trabajadora	El sistema debe permitir registrar los ingresos generados por cada estilista (empleada).
RU-003	Cálculo de porcentaje de pago de empleada	El sistema debe calcular automáticamente el porcentaje de pago de cada trabajadora basado en los ingresos generados.
RU-004	Registro de egresos	El sistema debe permitir registrar los egresos o salidas de dinero, incluyendo pagos de empleadas.
RU-005	Registro de pagos de empleadas	En el módulo de egresos, el sistema debe permitir registrar el pago a las empleadas.
RU-006	Liquidación semanal de trabajadora	El sistema debe calcular la liquidación semanal de cada trabajadora, considerando sus ingresos y pagos.
RU-007	Módulo de reportes de contabilidad mensual	El sistema debe generar un reporte mensual que sume los ingresos y egresos, y calcule el efectivo y el saldo en la cuenta.
RU-008	Registro de pagos pendientes	El sistema debe permitir registrar los pagos pendientes de los clientes, con su nombre, teléfono y dirección.
RU-009	Módulo de reporte mensual por cliente	El sistema debe generar reportes mensuales de los pagos realizados y pendientes por cada cliente.
RU-010	Visualización de saldo final mensual	El sistema debe mostrar el saldo final mensual, que sume ingresos, egresos y el saldo en efectivo y en cuenta.

RU-011	Gestión de pagos a proveedores y empleados	El sistema debe permitir registrar los pagos realizados a proveedores y empleados de manera detallada.
RU-012	Validación de registros de ingresos	El sistema debe validar que los registros de ingresos por trabajadora sean correctos antes de realizar cálculos de pagos.
RU-013	Acceso a historial de liquidaciones	El sistema debe permitir ver el historial de liquidaciones anteriores de cada trabajadora.
RU-014	Modificación de registros de pagos pendientes	El sistema debe permitir editar o eliminar los registros de pagos pendientes de los clientes.
RU-015	Control de pagos pendientes (arriendo, servicios, proveedores)	El sistema debe permitir registrar los pagos pendientes, especificando la empresa a la que se le adeuda, el monto, el concepto (arriendo, servicios, proveedores, facturas pendientes, etc.) y las fechas de pago.
RU-016	Almacenamiento de servicios y porcentajes para empleadas	El sistema debe permitir almacenar los servicios prestados y los porcentajes asignados a cada empleada, para agilizar el registro de los ingresos generados por cada trabajadora, mediante una lista desplegable.

Nota: Requisitos definidos por el usuario.

Fuente: Elaboración propia.

6.2.2 Requerimientos funcionales

Tabla. 5

Requerimientos funcionales.

Id requisitos funcionales	Nombre	Descripción	Información necesaria	Actor	Sesión
RF01	Registro de usuario	El usuario debe registrar los campos requeridos para crear su cuenta en la aplicación.	Nombre Apellido Celular Email Contraseña	Usuario	Gestión de usuarios
RF02	Inicio de sesión(Login)	El usuario debe validar los campos de usuario y	Usuario Contraseña	Usuario	Gestión de

		contraseña para ingresar a su cuenta dentro de la aplicación.			usuarios
RF03	Eliminación de cuenta	El sistema debe permitir eliminar los registros de los usuarios inactivos.	Usuario Contraseña	Usuario Admin	Gestión de usuarios
RF04	Registro de presupuesto	El sistema manejará un saldo inicial ingresado por el usuario al comenzar a usar la aplicación.	Saldo inicial	Usuario Admin	Gestión financiera
RF05	Registro de egresos	El sistema permitirá al usuario registrar sus egresos en el presupuesto y clasificarlos si son en transferencia o en efectivo.	Usuario Servicio o producto Nota Valor monetario Fecha	Usuario	Gestión financiera
RF06	Actualización de saldos	El sistema actualizará los ingresos y egresos registrados por los diferentes usuarios, y debe mostrar el saldo disponible en efectivo (caja) y en el banco (cuenta de ahorros).	Tipo de cuenta (Caja o Banco)	Sistema	Gestión financiera
RF087	Actualización de saldos	El sistema actualizará los ingresos y egresos registrados por los diferentes usuarios, y debe mostrar el saldo disponible en efectivo (caja) y en el banco (cuenta de ahorros).	Tipo de cuenta (Caja o Banco)	Sistema	Gestión financiera
RF08	Historial de movimientos	El sistema mostrará un registro de todas las transacciones realizadas, con fecha y	Fecha Tipo de movimiento Categoría	Sistema	Gestión financiera

		categoría (Ingresos, egresos y ventas).	Usuario		
RF09	Consultas y búsquedas	El sistema permitirá filtrar movimientos por categoría o rango de fechas.	Categoría Fecha Tipo movimiento Usuario	Usuario	Gestión financiera
RF10	Cálculo de la liquidación de empleados	El sistema debe calcular el valor a liquidar de cada empleado de forma semanal, con el porcentaje que le toca según los servicios prestados.	Rango de fechas Valor a liquidar Usuario	Usuario Admin	Gestión financiera
RF11	Registro de servicios	El sistema debe permitir registrar los servicios prestados, clasificándolos en tres categorías, servicios de peluquería, maquillaje y servicios de manicura y pedicure, cada uno con el porcentaje de liquidación y el tipo de pago, si es en efectivo o en transferencia.	Rango de fechas Valor a liquidar Usuario	Usuario Admin	Gestión financiera
RF12	Registro de clientes	El sistema debe permitir registrar a clientes.	Nombre de cliente Documento de identidad Celular Dirección	Usuario	Gestión de clientes
RF13	Registro de cuentas por cobrar de clientes	El sistema debe permitir el registro de cuentas por cobrar de clientes que pagan a crédito.	Cedula Fecha Valor Servicios	Usuario	Gestión de clientes

Nota: Requerimientos funcionales solicitados para el funcionamiento del sistema.

Fuente: Elaboración propia.

6.2.3 Requerimientos NO funcionales

Tabla. 6
Requerimientos NO funcionales.

ID REQUISITO NO FUNCIONAL	NOMBRE	DESCRIPCIÓN
Usabilidad (Usability)		
RNF 01	Tiempo de adaptación	La aplicación debe tener una interfaz de usuario intuitiva y fácil de aprender para el personal del salón de belleza, permitiendo a un nuevo usuario realizar el registro de un servicio en menos de 5 minutos después de una breve capacitación.
RNF 02	Navegabilidad	La navegación dentro de la aplicación debe ser coherente y consistente en todas las pantallas y módulos, facilitando la localización de funcionalidades clave.
RNF 03	Errores	Los mensajes de error y las notificaciones del sistema deben ser claros, concisos y orientados a la solución, permitiendo al usuario entender y corregir problemas fácilmente.
RNF 04	Diseño	La interfaz debe ser responsive, adaptándose correctamente a diferentes tamaños de pantalla (escritorio, tablet) para una experiencia de usuario óptima.
Rendimiento (Performance)		
RNF 05	Capacidad de respuesta	Las operaciones de consulta y registro de datos (ej. registro de un servicio, visualización de ingresos/egresos diarios) deben completarse en menos de 3 segundos para el 90% de las solicitudes, bajo una carga normal de 5 usuarios concurrentes.
RNF 06	Capacidad de usuarios	La aplicación debe soportar un mínimo de 5 usuarios concurrentes sin una degradación significativa del rendimiento
RNF 07	Tiempo de respuesta	El tiempo de carga inicial de cualquier página principal (dashboard, listado de servicios) no debe exceder los 2 segundos.

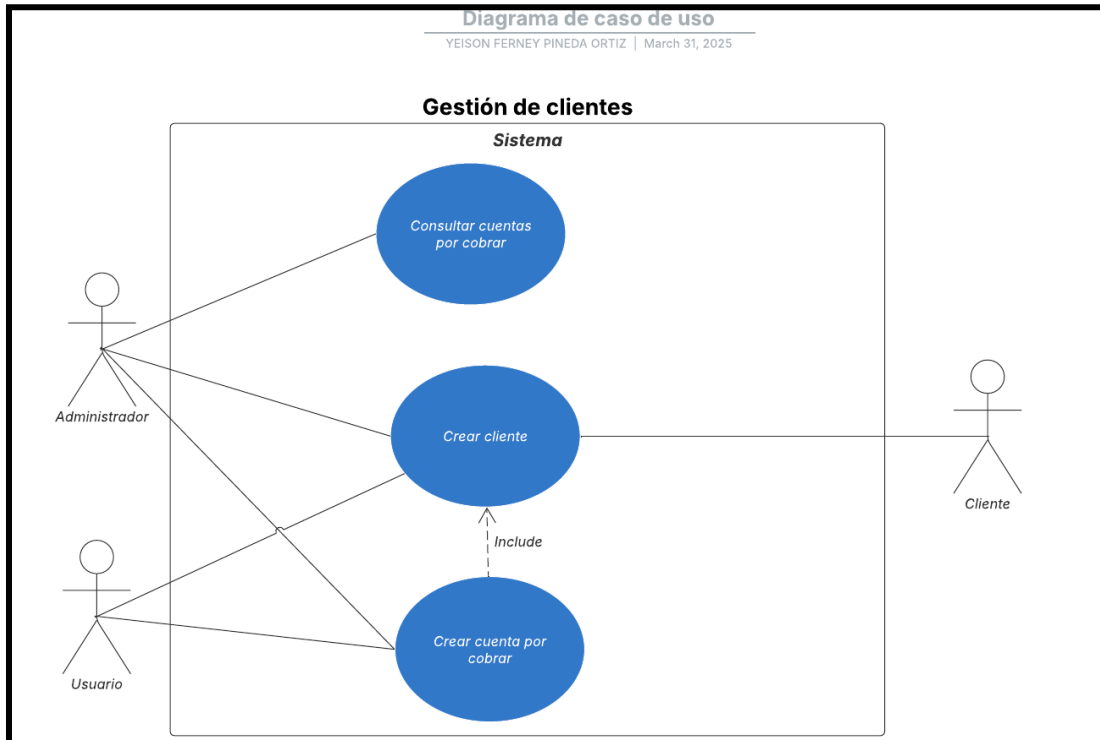
Seguridad (Security)		
RNF 08	Seguridad	La Aplicación debe ser confiable y solo el cliente obtendrá la información necesaria para acceder al Login, también tratar los datos personales de forma confidencial.
Mantenibilidad (Maintainability)		
RNF 09	Código	El código fuente de la aplicación debe estar bien documentado, utilizando comentarios claros y siguiendo estándares de codificación
RNF 10	Sistema modular	La arquitectura del sistema debe ser modular, permitiendo la fácil adición, modificación o eliminación de funcionalidades sin afectar el resto del sistema (favorecido por MVC).
RNF 11	Compatibilidad	La aplicación debe ser compatible con los navegadores web modernos más utilizados (ej. Chrome, Firefox, Edge, Safari en sus últimas 3 versiones estables).
RNF 12	Base de datos	La aplicación debe ser compatible con la base de datos MySQL o MariaDB en sus versiones estables actuales.
Escalabilidad (Scalability)		
RNF 13	Arquitectura	La arquitectura de la aplicación debe permitir un incremento futuro en el volumen de datos (ej. servicios, ingresos, egresos) sin requerir una reestructuración significativa del sistema.
RNF 14	Número de usuarios	El sistema debe ser capaz de soportar un aumento del 50% en el número de usuarios concurrentes (hasta 7-8) con una configuración mínima de recursos de hardware.

*Nota: Requerimientos NO funcionales solicitados para el funcionamiento del sistema.
Fuente: Elaboración propia.*

6.2.4 Diagramas de casos de uso

Ilustración. 7

Diagrama de caso de uso clientes

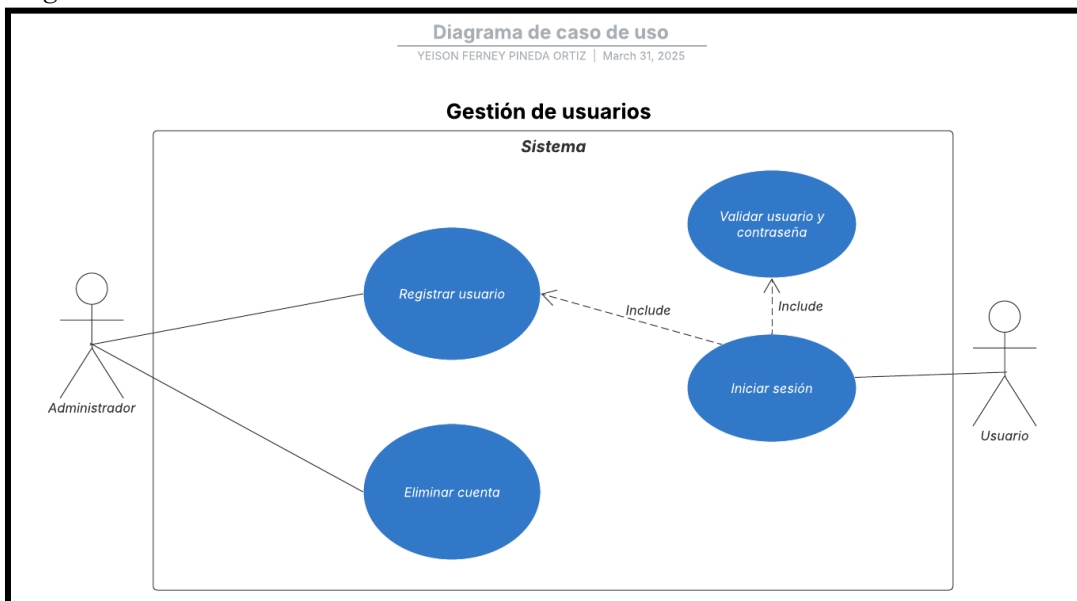


Nota: Diagrama de caso de uso gestión de clientes .

Fuente: Elaboración propia.

Ilustración. 8

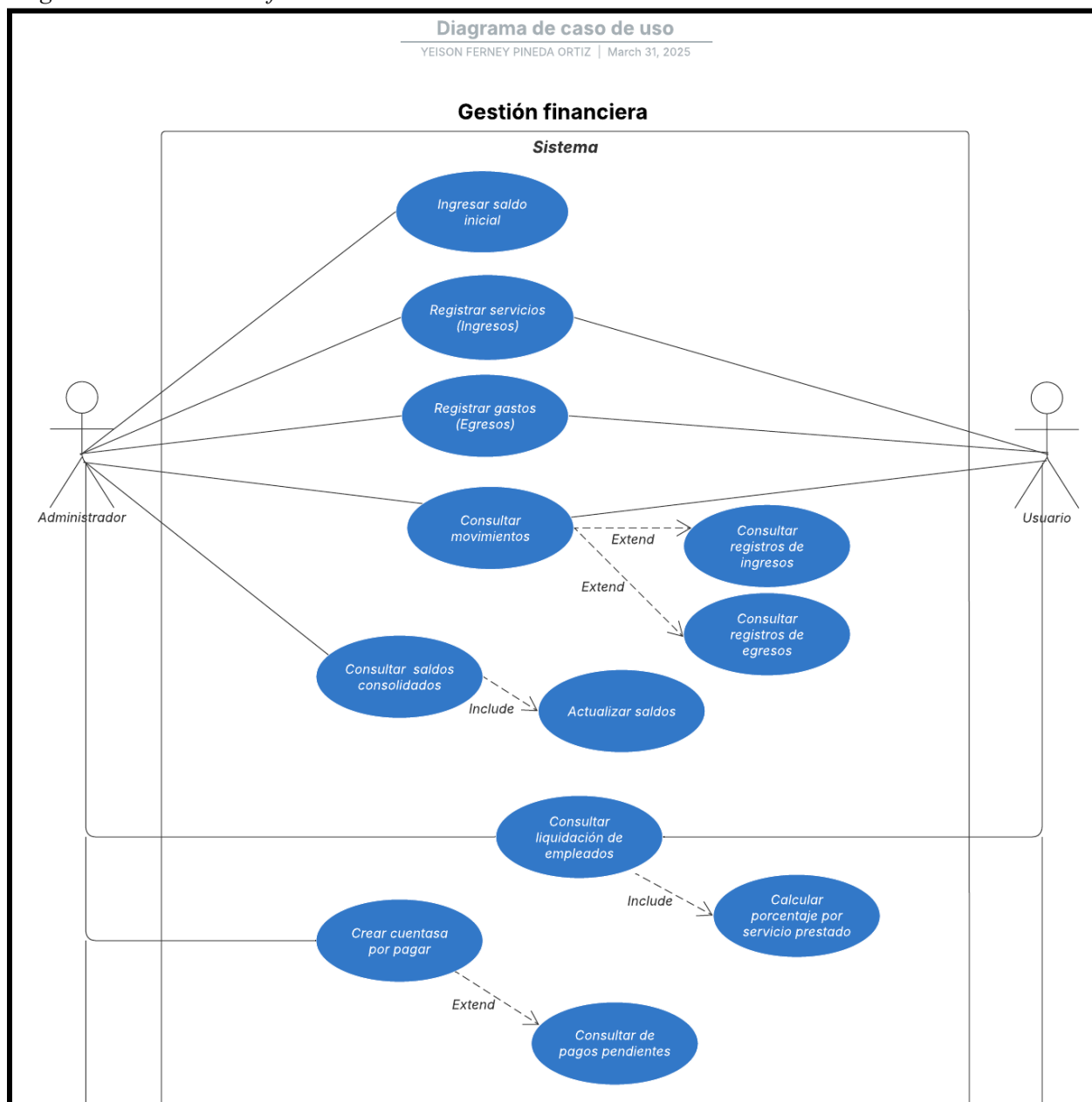
Diagrama de caso de uso de usuarios.



Nota: Diagrama de caso de uso gestión de clientes .

Fuente: Elaboración propia.

Ilustración. 9
Diagrama de caso de uso financiero.



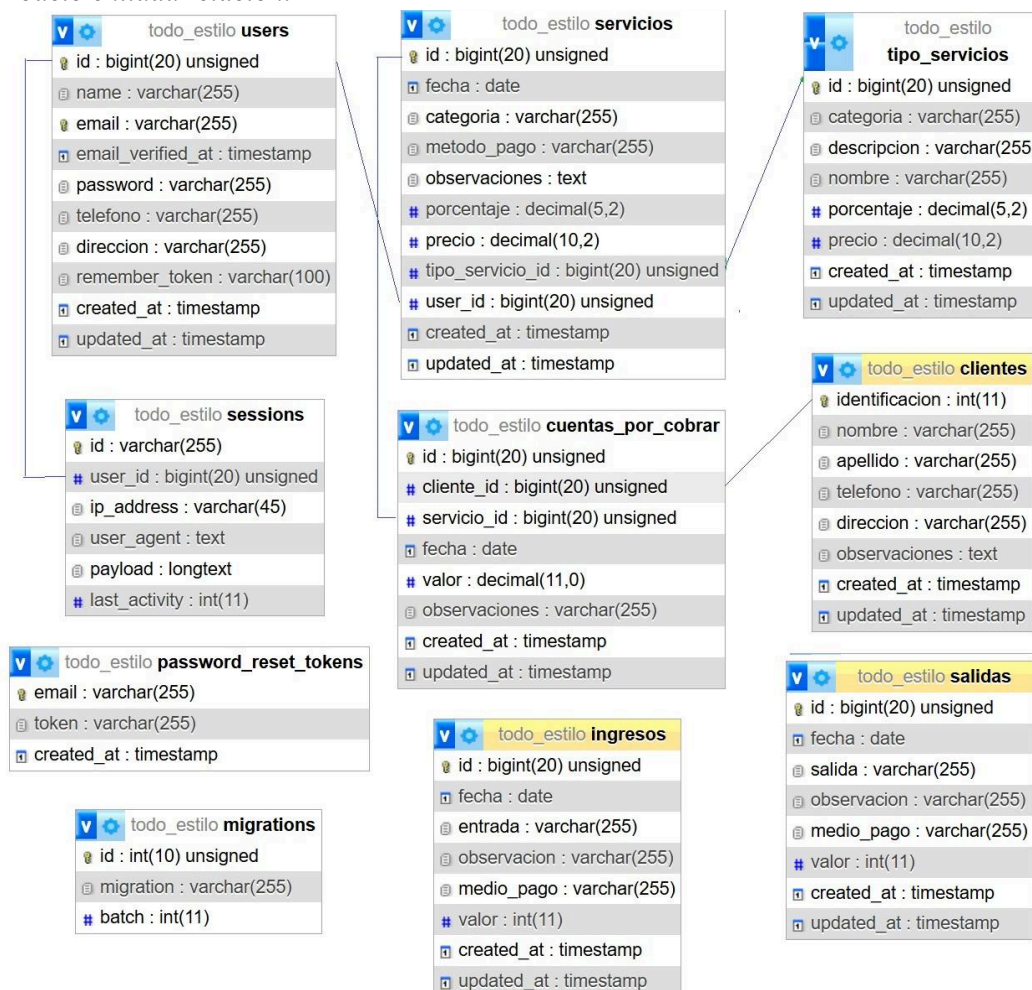
Nota: Diagrama de caso de uso gestión financiera
Fuente: Elaboración propia.

6.3 Fase de diseño conceptual

6.3.1 Modelo entidad relación (MER)

Para el desarrollo del presente proyecto se diseñó una base de datos relacional utilizando el sistema de administración de bases de datos MySQL (phpMyAdmin), por su robustez, eficiencia y compatibilidad con entornos web. El modelo entidad relación (MER) se presenta a continuación, en donde se puede visualizar de manera clara la estructura lógica de la información necesaria para gestionar los procesos administrativos y operativos del salón de belleza Todo Estilo. Este diseño es fundamental para garantizar la integridad de los datos, optimizar las consultas y facilitar el desarrollo de funcionalidades clave dentro de la aplicación.

Ilustración. 10
Modelo entidad relación.



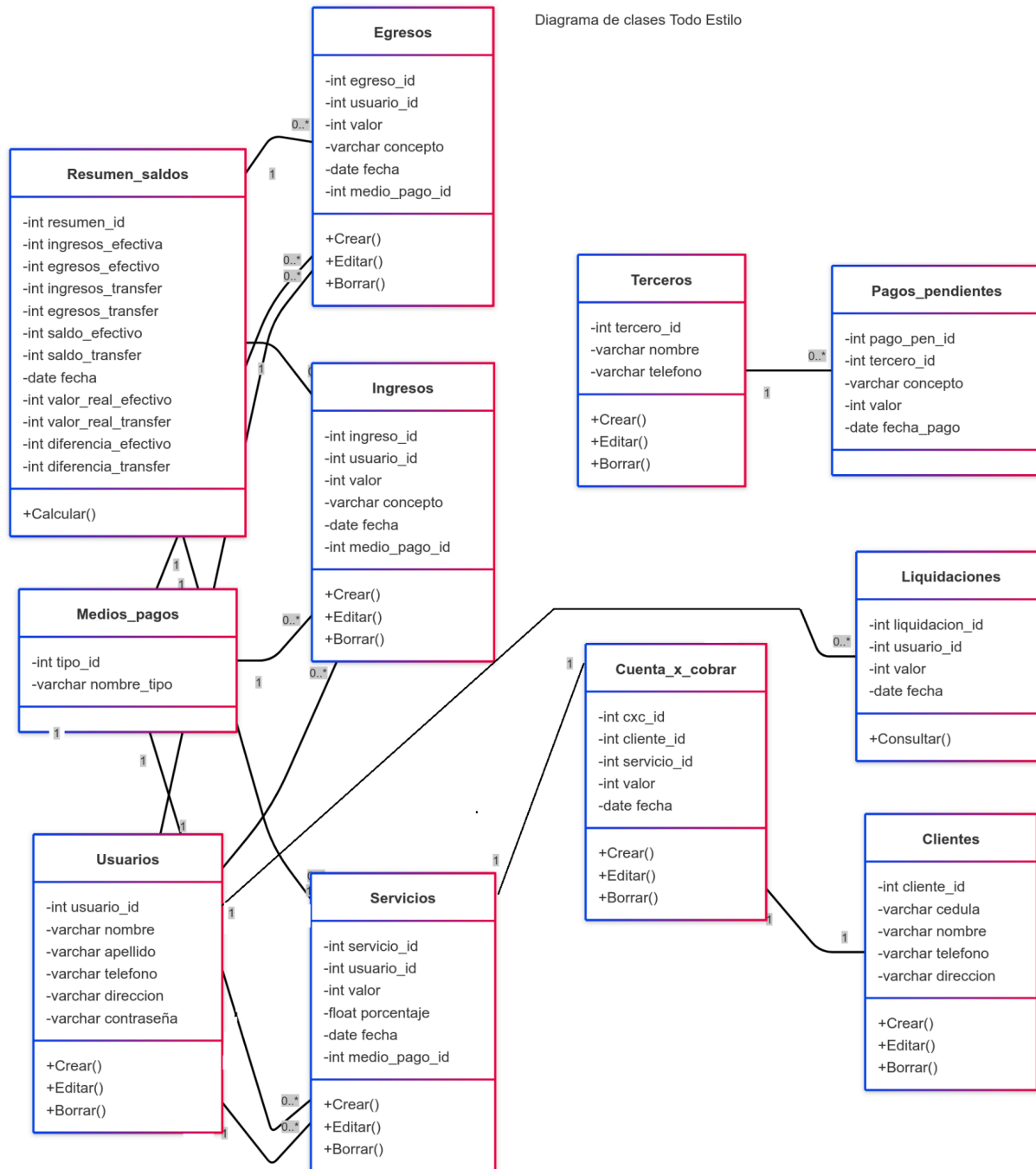
Nota: Diseño de la base de datos (Modelo ER)

Fuente: Elaboración propia.

6.3.2 Diagrama de clases modelo UML

El diagrama de clases UML representa la estructura lógica del modelo dentro de la arquitectura Modelo Vista Controlador (MVC) utilizada en el desarrollo del proyecto. Este componente es fundamental ya que define las clases, los atributos, métodos y las relaciones que rigen la lógica de negocio y la gestión de los datos del sistema. Al definir el modelo de esta manera, el diagrama no solo ofrece una vista clara de la estructura de la base de datos requerida, sino que también establece las bases para la lógica de negocio y las interacciones de datos que luego serán gestionadas por los controladores y presentadas en las vistas. Este diagrama actúa como un plano conceptual del sistema general a desarrollar, definiendo qué datos existen, cómo se deben de organizar y cómo se relacionan entre sí, permitiendo así una implementación coherente, ordenada y escalable.

Ilustración. 11
Diagrama de clases modelo UML



Nota: Diagrama de clases Todo Estilo
Fuente: Elaboración propia.

6.4 Fase de desarrollo ágil por iteraciones (sprints)

Funcionalidades clave desarrolladas

La aplicación desarrollada para Todo Estilo en resumen incorpora las siguientes funcionalidades esenciales, que se desarrollaron y se entregaron en cada uno de los Sprint como se puede ampliar en la tabla. 7 Product Backlog del proyecto:

- Gestión de usuarios
- Registro de servicios
- Control de ingresos y egresos
- Liquidación de empleados
- Gestión de cuentas por cobrar
- Cuadre de caja
- Control de obligaciones y pagos pendientes

Metodología de desarrollo

El proyecto se desarrolló bajo una metodología ágil, lo que permitió un enfoque iterativo y flexible. En el proceso se incluyeron:

- Fase de preparación y definición del producto: observación y análisis de procesos, seguido de la creación de un product backlog priorizado.
- Sprints de desarrollo: ciclos cortos y fijos donde se planificaron, diseñaron, implementaron y probaron funcionalidades específicas de forma incremental.
- Revisión y adaptación: sesiones periódicas de sprint review con los stakeholders y sprint retrospective para la mejora continua del producto y del proceso.

Estructura de carpetas y módulos de la aplicación Todo Estilo.

Como ya ha sido mencionado la aplicación web desarrollada está basada en el framework Laravel, el cual emplea la arquitectura modelo vista controlador (MVC). Esta arquitectura permite una separación clara de responsabilidades entre la lógica de negocio, la presentación de datos y el manejo de solicitudes del usuario, facilitando el desarrollo, mantenimiento y escalabilidad del sistema.

Además de la estructura estándar que Laravel ofrece, el proyecto ha sido organizado de manera modular, agrupando componentes relacionados por funcionalidad (Controller, Models, Views) dentro del núcleo de la aplicación. Esta organización modular mejora la legibilidad del código, permite un desarrollo más ordenado y facilita futuras expansiones o mantenimientos en áreas específicas del sistema.

Justificación de la estructura

- Claridad y organización: agrupa la lógica relacionada, haciendo que el código sea más fácil de encontrar y comprender.
- Mantenibilidad: los cambios en un módulo tienen menos probabilidades de afectar a otros, facilitando la depuración y las actualizaciones.
- Escalabilidad: permite añadir nuevas funcionalidades como módulos independientes sin saturar los directorios principales de laravel.
- Separación de responsabilidades (srp): refuerza el principio de que cada módulo o clase debe tener una única responsabilidad.
- Reusabilidad: facilita la reutilización de componentes dentro de un módulo o entre diferentes módulos si es necesario.

Componentes de las vistas (frontend) de la aplicación

La interfaz de usuario de la aplicación se construye utilizando las plantillas .blade de Laravel, en conjunto con Tailwind CSS para el estilizado y JavaScript para la interactividad. La organización de las vistas se ha diseñado para promover la modularidad, la reutilización de código manteniendo siempre una coherencia visual.

El directorio resources/views/ organiza las plantillas blade en subcarpetas que reflejan la modularidad del backend, además de incluir directorios para componentes reutilizables y layouts generales.

Los layouts definen la estructura HTML base de las páginas, incluyendo la cabecera (<head>), el menú de navegación principal, el pie de página y los contenedores para el contenido dinámico.

layouts/app.blade.php:

- Propósito: define el layout principal para la mayoría de las páginas autenticadas de la aplicación (ej. dashboard, módulos de gestión).
- Contenido: incluye la barra de navegación principal (sidebar y/o topbar), el área de contenido (`@yield('content')`), el pie de página, y enlaces a los estilos CSS (Tailwind CSS) y scripts JavaScript globales.
- Uso: todas las vistas principales (ej. `dashboard/index.blade.php`, `empleados/index.blade.php`) extenderán este layout (`@extends('layouts.app')`).

layouts/auth.blade.php:

- Propósito: define el layout específico para las páginas de autenticación (login, registro, recuperación de contraseña).
- Contenido: suele ser más simple, con un contenedor centrado para el formulario de autenticación y sin la navegación interna de la aplicación.
- Uso: vistas como `auth/login.blade.php`, `auth/register.blade.php` lo extenderán.

Dentro de cada directorio de cada módulo (ej. `usuarios/`, `ingresos/`), se encuentran las vistas que construyen las interfaces de usuario para las funcionalidades específicas de ese módulo.

Típicamente, incluyen:

- `index.blade.php`: lista o tabla de registros (ej. listado de usuarios, listado de ingresos).

Ilustración. 12
Estructura de la vista usuarios



Todo Estilo Cerrar sesión Regresar

Crear usuario

Lista de usuarios

NOMBRE	EMAIL	TELÉFONO	DIRECCIÓN		
Admin	yeis.pineda@gmail.com	3128845647	calle 57 19-30	Editar	Borrar
Sara Henao A.	sara@gmail.com	3117324230	CR 79 37S-58 Int 212	Editar	Borrar
Paula	paula@gmail.com	3117324230	CR 79 37S-58 Int 212	Editar	Borrar

Nota: CRUD para la administración de usuarios.
Fuente: Elaboración propia.

Ilustración. 13
Estructura de la vista ingresos



Todo Estilo Cerrar sesión Regresar

Registrar nuevo ingreso

Lista de ingresos

FECHA	ENTRADA	OBSERVACIÓN	MEDIO DE PAGO	VALOR		
2025-05-31	Saldo Inicial	Prueba 5	Efectivo	\$ 24,000	Editar	Borrar
2025-06-01	Insumos	Prueba 5	Transferencia	\$ 10,000	Editar	Borrar

Nota: CRUD para la administración y gestión de ingresos.
Fuente: Elaboración propia.

- create.blade.php: formulario para crear un nuevo registro.

Ilustración. 14

Estructura del formulario para el registro de ingresos.



The screenshot shows a web form titled "Registrar ingreso" for a business named "Todo Estilo". At the top is a logo featuring a woman's profile with flowing hair and a pair of scissors. Below the logo, the form contains several input fields: "Fecha" with a date picker showing "dd/mm/aaaa"; "Entrada" with a dropdown menu labeled "Seleccione el tipo de entrada"; "Observación" with a text input field; "Medio de pago" with a dropdown menu labeled "Seleccione el medio de pago"; and "Valor" with a text input field containing "Ej: 50000". At the bottom of the form are two buttons: "Registrar ingreso" and "Cancelar".

Nota: Vista del formulario para el registro de un nuevo ingreso.

Fuente: Elaboración propia.

- edit.blade.php: formulario para editar un registro existente.

Ilustración. 15

Estructura del formulario para la edición de ingresos.



The screenshot shows a web form titled "Editar ingreso" for the same business "Todo Estilo". It features the same logo as the previous form. The form fields are: "Fecha" with a date picker showing "31/05/2025"; "Entrada" with a dropdown menu showing "Saldo Inicial"; "Observación" with a text input field containing "Prueba 5"; "Medio de Pago" with a dropdown menu showing "Efectivo"; and "Valor" with a text input field containing "24000". At the bottom are two buttons: "Guardar cambios" and "Cancelar".

Nota: Vista del formulario para el registro de un nuevo ingreso.

Fuente: Elaboración propia.

- `show.blade.php`: vista para mostrar los detalles de un registro específico.

Configuración del Entorno

La aplicación web de Todo Estilo gestiona su configuración del entorno mediante variables definidas en el archivo `.env`. Este archivo es crucial para adaptar el comportamiento de la aplicación a diferentes entornos (desarrollo, pruebas, producción) sin modificar el código fuente.

Los parámetros de configuración clave gestionados por el archivo `.env` incluyen:

- Configuración general de la aplicación: nombre de la aplicación, el entorno de despliegue (`APP_ENV`), la clave de cifrado (`APP_KEY`), el estado de depuración (`APP_DEBUG`), y la URL base de la aplicación (`APP_URL`).
- Conexión a la base de datos: credenciales y detalles de conexión para la base de datos MySQL/phpMyAdmin (host, puerto, nombre de la base de datos, usuario y contraseña).
- Servicios adicionales: configuración para el envío de correo electrónico y otros servicios externos que la aplicación pueda requerir.

Procedimiento de despliegue

El despliegue de la aplicación web se realizó en un entorno de hosting compartido, utilizando la plataforma Hostinger. El proceso se llevó a cabo siguiendo los siguientes pasos:

1. Configuración del Entorno de Hosting:

- Contratación de un plan de hosting en Hostinger con soporte para PHP y MySQL.
- Acceso a hPanel para la administración del hosting.
- Asociación del dominio/subdominio del proyecto.
- Creación de una nueva base de datos MySQL, incluyendo su nombre, usuario y contraseña.

2. Preparación del Proyecto Laravel:

- Generación local de las dependencias PHP mediante `composer install` (incluyendo la

carpeta vendor).

- Configuración del archivo .env del proyecto con los parámetros del entorno de producción de Hostinger:
- APP_ENV=production
- APP_DEBUG=false
- APP_URL configurado con el dominio del proyecto.
- DB_HOST, DB_DATABASE, DB_USERNAME, DB_PASSWORD actualizados con las credenciales de la base de datos de Hostinger.
- APP_KEY asegurado con un valor único.

3. Transferencia de archivos del proyecto:

- Utilización del administrador de archivos de hPanel o FTP para la transferencia.
- Los archivos principales del proyecto (excepto el contenido de public/) se subieron a un directorio seguro fuera de public_html.
- El contenido de la carpeta public/ de Laravel se subió directamente al directorio public_html (o la raíz del dominio/subdominio).
- Ajuste del archivo public_html/index.php para corregir las rutas de carga (vendor/autoload.php y bootstrap/app.php) apuntando al directorio seguro.

4. Configuración de Base de Datos y Permisos

- Importación del esquema de la base de datos y los datos iniciales a través de phpMyAdmin en hPanel.
- Ajuste de permisos de escritura (generalmente 775) para los directorios storage/ y bootstrap/cache/ del proyecto en el servidor.

5. Verificación y Optimización

- Acceso a la URL del dominio para verificar la carga y funcionalidad de la aplicación.
- Se realizaron pruebas funcionales básicas para asegurar la correcta operación de los módulos.

6.4.2 Backlog.

Tabla. 7

Product Backlog del proyecto.

ID	Historia de Usuario	Prioridad	Aprobado Sí/No	Observaciones
SPRINT 1 - 08 de mayo de 2025				
HU-1	Como administrador, quiero registrar a los empleados con su nombre, celular, dirección y correo, para hacerlos usuarios del sistema.	Alta	SI	Funcional
HU-2	Como usuario, quiero validar los campos de usuario y contraseña para ingresar a mi cuenta dentro de la aplicación, para acceder de forma segura.	Alta	SI	Login Ok, funcional
HU-3	Como usuario, quiero registrar los campos requeridos para crear mi cuenta en la aplicación, para poder acceder al sistema.	Alta	SI	Funcional
HU-4	Como administrador, quiero eliminar los registros de los usuarios inactivos, para mantener la base de datos limpia.	Baja	SI	Funcional
SPRINT 2 - 15 de mayo de 2025				
HU-5	Como empleado, quiero registrar los ingresos generados por cada trabajadora, para llevar un control de los ingresos individuales.	Alta	SI	Funcional
HU-6	Como administrador, quiero validar que los registros de ingresos por trabajadora sean correctos antes de realizar cálculos de pagos, para asegurar la precisión de los pagos.	Media	SI	Tabla principal de servicios, se puede validar el general por día y el detallado por empleado
HU-7	Como administrador, quiero registrar los egresos o salidas de dinero, incluyendo pagos de empleadas, para llevar un control de los gastos.	Alta	SI	Funcional
HU-8	Como administrador, quiero editar o eliminar los registros de los ingresos y salidas de dinero.	Alta	SI	Ok. Funcional
HU-9	Como administrador, quiero registrar los pagos realizados a proveedores y empleados de manera detallada, para mantener un registro preciso de los pagos.	Media	SI	Formulario de salidas

HU-10	Como administrador, quiero manejar un saldo inicial ingresado por el usuario al comenzar a usar la aplicación, para establecer el presupuesto inicial.	Baja	SI	Se maneja en el formulario de ingresos
HU-11	Como usuario, quiero ver un registro de todas las transacciones realizadas, con fecha y categoría, para tener un historial de movimientos.	Baja	SI	Formulario de servicios
HU-12	Como usuario, quiero filtrar movimientos por categoría o rango de fechas, para analizar las transacciones.	Baja	SI	Formulario de servicios
HU-13	Como administrador, quiero validar que los registros de ingresos por trabajadora sean correctos antes de realizar cálculos de pagos, para asegurar la precisión de los pagos.	Media	SI	Tabla principal de servicios, se puede validar el general por día y el detallado por empleado
SPRINT 3 - 22 de mayo de 2025				
HU-14	Como administrador, quiero registrar los pagos pendientes de los clientes, con su nombre, teléfono y dirección, para llevar un control de las deudas de los clientes.	Media	SI	OK
HU-15	Como administrador, quiero generar reportes mensuales de los pagos realizados y pendientes por cada cliente, para llevar un control de las cuentas por cobrar.	Media	SI	Funcional
HU-16	Como administrador, quiero editar o eliminar los registros de pagos pendientes de los clientes, para mantener la información actualizada.	Media	SI	Funcional
HU-17	Como administrador, quiero ver el historial de liquidaciones anteriores de cada trabajadora, para llevar un registro de los pagos realizados.	Media	SI	Reporte de liquidaciones
HU-18	Como administrador, quiero que el sistema calcule automáticamente el porcentaje de pago de cada trabajadora basado en los ingresos generados, para agilizar el cálculo de la liquidación.	Alta	SI	Liquidaciones
HU-19	Como administrador, quiero registrar el pago a las empleadas, para llevar un registro de los pagos al personal.	Media	SI	Registro como salidas

HU-20	Como administrador, quiero calcular la liquidación semanal de cada trabajadora, considerando sus ingresos y pagos, para realizar los pagos de manera oportuna.	Media	SI	Se puede filtrar por fecha	
HU-21	Como administrador, quiero permitir registrar los pagos pendientes, especificando la empresa a la que se le adeuda, el monto, el concepto y las fechas de pago, para llevar un control de las deudas.	Media	SI	OK	
HU-22	Como administrador, quiero almacenar los servicios prestados y los porcentajes asignados a cada empleada, para agilizar el registro de los ingresos generados por cada trabajadora, mediante una lista desplegable.	Baja	SI	Funcional	
HU-23	Como usuario, quiero ingresar mis ingresos en el presupuesto y clasificarlos si son en transferencia o en efectivo, para llevar un control de los ingresos.	Baja	SI	En el registro de servicios, ingresos y egresos se puede indicar el medio de pago	
HU-24	Como usuario, quiero que el sistema actualice los ingresos y egresos registrados por los diferentes usuarios, y muestre el saldo disponible en efectivo (caja) y en el banco (cuenta de ahorros), para conocer el estado actual de las finanzas.	Baja	SI	En el registro de servicios - ingresos y egresos se puede indicar el medio de pago	
HU-25	Como administrador, quiero generar un reporte mensual que sume los ingresos y egresos, y calcule el efectivo y el saldo en la cuenta, para obtener una visión general de la situación financiera.	Alta	SI	Reporte de cuadro de caja	
SPRINT 4 - 28 de mayo de 2025					
	Entrega total: . Elaborar la documentación técnica y el manual de usuario, asegurando una implementación adecuada y facilitando su uso por parte de los empleados y clientes de Todo Estilo.	Alta	SI	Entrega del manual de instalación y manual de usuario.	
			Aprobados:	SI	26
				NO	0

Nota: Product Backlog e HU entregadas en cada cada sprint.
Fuente: Elaboración propia.

6.5 Fase de pruebas continuas y evaluación de resultados

Plan de pruebas:

El plan de pruebas de la aplicación web "Todo Estilo" tiene como objetivo garantizar la calidad, funcionalidad, usabilidad y seguridad del sistema. Se aplicará un enfoque de pruebas por módulos, combinando pruebas unitarias, de integración y funcionales (end-to-end), para asegurar que cada componente y el sistema en su conjunto operen según los requisitos definidos.

Las pruebas se llevan a cabo divididos en tres módulos donde cada uno de estos se enfocara en una sesión en específico del proyecto, los módulos son usuarios y login (CP-USUARIOS), gestión de servicios (CP-SERVICIOS) y gestión financiera (CP-FINANCIERA), con los cuales se llevan a cabo los siguientes tipos de pruebas:

- Pruebas unitarias: verificación individual de componentes de código (clases, funciones) para asegurar que cada unidad funcione correctamente de forma aislada.
- Pruebas de integración: verificación de la interacción entre diferentes módulos o componentes del sistema.
- Pruebas funcionales : simulación de escenarios de usuario reales para verificar que las funcionalidades cumplan con los requisitos del negocio.
- Pruebas de usabilidad: evaluación de la facilidad de uso, aprendizaje y satisfacción del usuario con la interfaz.
- Pruebas de rendimiento : verificación de la velocidad de respuesta de las operaciones clave.
- Pruebas de Seguridad: verificación de la protección contra vulnerabilidades comunes.
- Enfoque: pruebas incrementales durante el desarrollo (dentro de los Sprints) y pruebas de regresión al finalizar cada iteración importante.

Módulo 1: autenticación (login) y gestión de usuarios.

Objetivo: asegurar el correcto funcionamiento de los procesos de acceso y gestión de usuarios.

Tabla. 8
Caso de prueba CP-USUARIOS-001.

Caso de prueba #01	Código del CP	1
	¿Prueba de despliegue?	Si
Descripción	Inicio de sesión.	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Iniciar sesión con credenciales de usuario existentes y válidas.	
Prerrequisitos	* El sistema web debe estar operativo y accesible.	
	* Debe existir al menos un usuario con credenciales válidas en la base de datos (Se cargan por migración a la base de datos).	
Datos de entrada	Email: yeis.pineda@gmail.com	
	Contraseña: password	
Pasos	1. Navegar a la URL de inicio de sesión (/login), se accede a esta por el botón "Inicio sesión" ubicado en la parte superior derecha de la página de inicio.	
	2. Ingresar el email y la contraseña correctos en los campos correspondientes.	
	3. Hacer clic en el botón de "Iniciar Sesión".	
	4. Dar clic en el botón "Regresar".	
	5. Tratar de acceder con credenciales inválidas.	
Resultado esperado	La sesión debe iniciarse exitosamente.	
	El usuario debe ser redirigido al dashboard principal de la aplicación.	
	El sistema debe reconocer al usuario como autenticado.	
	Al dar en el botón "Regresar" debe cancelar el inicio de sesión y regresar a la pantalla de Inicio.	
	Al tratar de ingresar con credenciales inválidas tanto el correo como la contraseña, el sistema no debe permitir ingresar y muestra el mensaje "Estas credenciales no coinciden"	
Resultado obtenido	El proceso de login funciona según las especificaciones indicadas, hace el proceso y redirige a la vista principal o dashboard.	

	Al dar en el botón "Regresar" cancela la operación de login exitosamente y redirige a la página de inicio.
	Al tratar de iniciar sesión con las credenciales inválidas el sistema genera la alerta de “credenciales no coinciden” y no permite el acceso.

Nota: Caso de prueba para el login.

Fuente: Elaboración propia.

Tabla. 9

Caso de prueba CP-USUARIOS-002.

Caso de prueba #02	Código del CP	2
	¿Prueba de despliegue?	Si
Descripción	Registrar usuario.	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Registrar un nuevo usuario con credenciales válidas y todos los datos requeridos.	
Prerrequisitos	El sistema web debe estar operativo y accesible.	
	La función y el formulario de registro de usuarios debe estar habilitada.	
Datos de entrada	Nombre	
	Teléfono	
	Dirección	
	Correo electrónico	
	Contraseña	
Pasos	1. Navegar a la URL de registro (/register), ubicada en el panel lateral izquierdo de la vista principal o dashboard Usuarios>>Crear usuario.	
	2. Llenar los campos correspondientes sin dejar datos nulos o vacíos.	
	3. Hacer clic en el botón de "Crear usuario".	
	4. Dar clic en el botón "Cancelar".	
Resultado esperado	El nuevo usuario debe ser creado exitosamente en la base de datos.	
	El usuario debe ser redirigido a la vista de usuarios.	
	Se debe visualizar el mensaje de confirmación "Usuario creado exitosamente".	

	La información del usuario creado se debe visualizar en el CRUD de usuarios.
	Al dar clic en el botón cancelar debe cerrar el formulario de registro y redirigir a la vista de usuarios.
Resultado obtenido	<p>se genera el siguiente error: “Target class App/Http/Controllers/usuarioController, does not exist” faltaba la sentencia <?php en el controlador</p> <p>El proceso de creación se realiza exitosamente, se genera el mensaje de confirmación y se visualiza la información en el CRUD.</p> <p>Al dar en el botón "Cancelar" cancela la operación y redirige a la página usuarios.</p>

Nota: Caso de prueba para el registro de un usuario nuevo.

Fuente: Elaboración propia.

Tabla. 10

Caso de prueba CP-USUARIOS-003.

Caso de prueba #03	Código del CP	3
	¿Prueba de despliegue?	Si
Descripción	Editar la información de un usuario.	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Modificar la información de un usuario existente.	
Prerrequisitos	El sistema web debe estar operativo y accesible.	
	Debe existir al menos un usuario (empleado) registrado en el sistema para editar.	
	La función y el formulario de editar usuarios debe estar habilitada.	
Datos de entrada	Nombre	
	Teléfono	
	Dirección	
	Correo electrónico	
	Contraseña	
Pasos	1. Navegar a la URL de usuarios , ubicada en el panel lateral izquierdo de la vista principal o dashboard Usuarios.	
	2. Dar clic en el botón de editar del CRUD.	

	3. Realizar los cambios requeridos.
	4. Dar clic en el botón "Guardar cambios".
	4. Dar clic en el botón "Cancelar".
Resultado esperado	La información del usuario debe ser actualizada exitosamente en la base de datos.
	El usuario debe ser redirigido a la vista de usuarios.
	Debe mostrarse un mensaje de éxito confirmando la actualización.
	Los cambios deben reflejarse inmediatamente en la lista de usuarios.
Resultado obtenido	Add {_token} to fillable property to allow mass assignment on {App/Models/usuario}
	Faltaba agregar la cláusula de protected \$fillable = [] al modelo de usuarios.
	El proceso de edición se realiza exitosamente, se genera el mensaje de confirmación y se visualiza la información en el CRUD.
	Al dar en el botón "Cancelar" cancela la operación y redirige a la página usuarios.

Nota: Caso de prueba para la edición de un usuario existente.

Fuente: Elaboración propia.

Tabla. 11

Caso de prueba CP-USUARIOS-004.

Caso de prueba #04	Código del CP	4
	¿Prueba de despliegue?	Si
Descripción	Borrar el registro de un usuario del sistema.	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Borrar el registro de un usuario del sistema.	
Prerrequisitos	El sistema web debe estar operativo y accesible.	
	Debe existir al menos un usuario (empleado) registrado en el sistema que pueda ser eliminado.	
	La función de borrar usuarios debe estar habilitada.	
Datos de entrada	ID de usuario a eliminar: (obtenido de la lista de usuarios)	
Pasos	1. Navegar a la URL de usuarios , ubicada en el panel	

	lateral izquierdo de la vista principal o dashboard usuarios.
	2. Dar clic en el botón de borrar del CRUD.
	3. Confirmar que se desea borrar el registro del usuario.
Resultado esperado	El usuario debe ser eliminado exitosamente de la base de datos y de la lista de usuarios.
	Debe mostrarse un mensaje de éxito confirmando la eliminación.
	El usuario eliminado no debe poder iniciar sesión en el sistema.
Resultado obtenido	El proceso de eliminación se realiza exitosamente, se genera el mensaje de confirmación, no se visualiza la información en el CRUD usuarios y no permite ingresar con las credenciales de este usuario.

Nota: Caso de prueba para borrar el registro de un usuario existente.

Fuente: Elaboración propia.

Módulo 2: gestión de servicios (tipos de servicio y servicios realizados)

Objetivo: asegurar el correcto registro y consulta de los servicios ofrecidos y realizados.

Casos de prueba (CP):

Tabla. 12

Caso de prueba CP-SERVICIOS-001.

Caso de prueba #05	Código del CP	5
	¿Prueba de despliegue?	Si
Descripción	Crear un nuevo tipo de servicio	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Añadir un nuevo tipo de servicio diligenciando todos los campos.	
Prerrequisitos	El sistema web debe estar operativo y accesible.	
	El formulario de creación de tipo de servicio debe estar disponible.	
	La función y el formulario de editar usuarios debe estar habilitada.	
Datos de entrada	Nombre del servicio	
	Categoría	

	Descripción
	Porcentaje de liquidación
	Precio
Pasos	1. Iniciar sesión en el sistema.
	2. Navegar a la sección de gestión de tipos de servicio.
	3. Hacer clic en el botón "Nuevo tipo de servicio".
	4. Llenar todos los campos del formulario con datos válidos.
	5. Hacer clic en el botón "Crear servicio".
	6. Hacer clic en el botón "Cancelar" en lugar de "Crear servicio".
Resultado esperado	Si se crea el servicio: El nuevo tipo de servicio debe ser creado y visible en el listado de tipos de servicio..
	Debe mostrarse un mensaje de éxito.
	Si se da clic en cancelar, el proceso de creación debe cancelarse y la vista debe regresar a la lista de tipos de servicio sin guardar ningún cambio.
Resultado obtenido	Se presenta error: “No se puede declarar la clase App/Http/Controllers/TipoServicioController” se revisa el controlador y se encuentra que no estaban definidos los métodos create() y store() en tu controlador.
	Se vuelve a probar y ahora se genera el error “Ver {crear_tipo_servicios} no encontrado”, se encuentra mal definida la ruta en el controlador, estaba como crear_tipo_servicios y realmente era formulario_crear_tipo_servicios.
	El tipo de servicio se creó exitosamente y aparece en el listado. Se mostró un mensaje de éxito. El proceso fue cancelado y la vista regresó al listado de tipos de servicio sin guardar el servicio.

Nota: Caso de prueba para crear un nuevo tipo de servicio.

Fuente: Elaboración propia.

Tabla. 13

Caso de prueba CP-SERVICIOS-002.

Caso de prueba #06	Código del CP	6
	¿Prueba de despliegue?	Si
Descripción	Registrar un servicio prestado	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Registrar un servicio completado, asignando un empleado, tipo de servicio, fecha, precio cobrado y porcentaje.	
Prerrequisitos	El sistema debe estar operativo y accesible.	
	Debe existir al menos un empleado y uno o más tipos de servicios registrados.	
	Debe estar habilitada la función de registrar servicios.	
	Debe de estar listo y funcional el formulario de registrar servicio.	
Datos de entrada	Fecha, estilista, categoría, servicio, método de pago, observaciones, porcentaje y precio.	
Pasos	1. Iniciar sesión en el sistema.	
	2. Navegar a la sección de registro de servicios.	
	3. Seleccionar el estilista.	
	4. Seleccionar la categoría y el servicio.	
	5. Ingresar la fecha y hora del servicio.	
	6. Ingresar el precio cobrado y el porcentaje (si aplica).	
	7. Ingresar observaciones y método de pago.	
	8. Hacer clic en "Guardar".	
Resultado esperado	El servicio debe registrarse correctamente en la base de datos. Se muestra el mensaje de éxito y aparece en el historial de servicios.	
Resultado obtenido	<p>Se genera el siguiente error al guardar un registro nuevo “SQLSTATE{42S22}: Colum ‘user_id’ ‘field list’ (Connection: mysql, SQL:insert into ‘servicios’ (‘fecha’,‘user_id’, ‘categoria’, ‘tipo_servicio_id’, ‘observaciones’, ‘porcentaje’, ‘precio’, ‘updated_at’,‘created_at’))</p> <p>Error en la ruta definida en el controlador la cual es corregida. Después, el registro se realiza exitosamente. Se genera mensaje de confirmación y el nuevo servicio aparece en el historial, con los datos guardados correctamente.</p>	

Nota: Caso de prueba para crear un servicio prestado.

Fuente: Elaboración propia.

Tabla. 14

Caso de prueba CP-SERVICIOS-003.

Caso de prueba #07	Código del CP	7
	¿Prueba de despliegue?	Si
Descripción	Cálculo de comisión del empleado.	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Verificar que la comisión del empleado se calcule correctamente al registrar un servicio.	
Prerrequisitos	Tener el registro de al menos un servicio prestado por un usuario en diferentes fechas.	
Datos de entrada	Servicio registrado con usuario, precio, porcentaje asignado y fechas de prestación del servicio.	
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en el sistema. 2. Registrar un servicio con precio y porcentaje de comisión. 3. Navegar a la sección de liquidación. 4. Filtrar por fechas. 5. Consultar el valor calculado de la comisión. 	
Resultado esperado	La comisión calculada para el empleado corresponde al valor esperado: Precio del servicio × porcentaje, según las fechas filtradas.	
Resultado obtenido	La comisión se calcula correctamente y coincide con el valor esperado para el servicio registrado.	

Nota: Caso de prueba para validar el cálculo de la liquidación.

Fuente: Elaboración propia.

Tabla. 15

Caso de prueba CP-SERVICIOS-004.

Caso de prueba #08	Código del CP	8
	¿Prueba de despliegue?	Si
Descripción	Visualización de historial de servicios realizados.	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Consultar el historial de servicios realizados por fecha y por empleado.	
Prerrequisitos	Tener registros de servicios prestados en diferentes fechas.	
Datos de entrada	Fecha del servicio, categoría, nombre del servicio, precio, porcentaje, método de pago, observaciones y usuario	

Pasos	1. Iniciar sesión en el sistema. 2. Navegar a la sección de historial de servicios. 3. Aplicar filtros por rango de fechas y/o empleado específico.
Resultado esperado	La lista de servicios muestra solo los registros que coinciden con los filtros aplicados, correctamente ordenados.
Resultado obtenido	El historial se muestra correctamente según los filtros aplicados, sin errores.

Nota: Caso de prueba para validar la visualización del historial de servicios.

Fuente: Elaboración propia

Tabla. 16

Caso de prueba CP-SERVICIOS-005.

Caso de prueba #09	Código del CP	9
	¿Prueba de despliegue?	Si
Descripción	Editar tipo de servicio.	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Modificar la información de un tipo de servicio existente.	
Prerrequisitos	Tener registros de tipos de servicios creados previamente.	
Datos de entrada	Nombre del servicio, categoría, descripción, precio y porcentaje.	
Pasos	1. Iniciar sesión. 2. Navegar a la sección de gestión de tipos de servicio. 3. Seleccionar un tipo de servicio. 4. Hacer clic en "Editar". 5. Modificar alguno o varios de los campos. 6. Guardar los cambios.	
Resultado esperado	El tipo de servicio se actualiza correctamente en el listado, se muestra un mensaje de éxito, y los cambios se reflejan tanto en la lista como al registrar un nuevo servicio.	
Resultado obtenido	El tipo de servicio se actualizó exitosamente y se reflejan los cambios sin errores.	

Nota: Caso de prueba para editar un tipo de servicio.

Fuente: Elaboración propia

Tabla. 17

Caso de prueba CP-SERVICIOS-006.

Caso de prueba #10	Código del CP	10
	¿Prueba de despliegue?	Si
Descripción	Editar servicio realizado.	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Modificar los detalles de un servicio ya registrado.	
Prerrequisitos	Tener registros de servicios realizados previamente.	
Datos de entrada	Fecha, servicio, precio, porcentaje, medio de pago, observaciones, estilista.	
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión. 2. Navegar al historial de servicios. 3. Seleccionar un servicio realizado. 4. Hacer clic en "Editar". 5. Modificar los campos necesarios. 6. Guardar los cambios. 	
Resultado esperado	El servicio se actualiza correctamente, se muestra un mensaje de éxito, y si se modifican precio o porcentaje, la comisión también se recalcula.	
Resultado obtenido	El servicio fue editado correctamente y los valores se actualizaron sin errores.	

Nota: Caso de prueba modificando un servicio ya registrado.

Fuente: Elaboración propia

Tabla. 18

Caso de prueba CP-SERVICIOS-007.

Caso de prueba #11	Código del CP	11
	¿Prueba de despliegue?	Si
Descripción	Eliminar un tipo de servicio del sistema.	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Verificar que se pueda eliminar correctamente un tipo de servicio existente.	
Prerrequisitos	Tener registrados al menos uno o más tipos de servicio.	
Datos de entrada	Identificador del tipo de servicio a eliminar.	

Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión. 2. Navegar a la gestión de tipos de servicio. 3. Seleccionar un tipo de servicio. 4. Hacer clic en "Borrar". 5. Confirmar la eliminación.
Resultado esperado	El tipo de servicio es eliminado del sistema, se muestra un mensaje de confirmación y desaparece de la lista.
Resultado obtenido	<p>se presenta el error “Call to undefined method App/Http/Controllers/TipoServicioController::destroy”</p> <p>Faltaba la ruta de borrar tipo servicio en el controlador.</p> <p>El tipo de servicio fue eliminado correctamente y ya no aparece en el listado.</p>

Nota: Caso de prueba para eliminar un tipo de servicio.

Fuente: Elaboración propia

Tabla. 19

Caso de prueba CP-SERVICIOS-008.

Caso de prueba #12	Código del CP	12
	¿Prueba de despliegue?	Si
Descripción	Eliminar un registro de servicio prestado.	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Asegurar que un servicio registrado pueda ser eliminado de forma segura.	
Prerrequisitos	Tener servicios registrados previamente.	
Datos de entrada	Identificador del servicio prestado.	
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión. 2. Ir al historial de servicios. 3. Seleccionar un servicio registrado. 4. Hacer clic en "Borrar". 5. Confirmar la acción. 	
Resultado esperado	El servicio desaparece del historial, se muestra mensaje de confirmación, y no se considera en cálculos posteriores como comisiones.	
Resultado obtenido	<p>se presenta el error “Call to undefined method App/Http/Controllers/ServicioController::destroy”</p> <p>Faltaba la ruta de borrar en el controlador.</p>	

	<p>Luego sale otro error “View {servicios.index} not found”</p> <p>Se encuentra que esta redireccionando a la vista que no era, debía redireccionar a admin.blade.php no a servicios.index.</p> <p>El servicio fue eliminado exitosamente y ya no aparece en el historial</p>
--	---

Nota: Caso de prueba para eliminar un servicio ya registrado.

Fuente: Elaboración propia

Módulo 3: gestión financiera (ingresos, egresos, cuentas por cobrar, obligaciones, cuadro de caja)

Objetivo: Validar la precisión y funcionalidad de todos los procesos financieros.

Casos de prueba (CP):

Tabla. 20

Caso de prueba CP-FINANCIERA-001.

Caso de prueba #13	Código del CP	13
	¿Prueba de despliegue?	Si
Descripción	Registrar un ingreso.	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Verificar que se pueda registrar correctamente un ingreso financiero en el sistema.	
Prerrequisitos	Tener el proceso de registros de ingresos y el formulario para registrar un nuevo ingreso funcional.	
Datos de entrada	Fecha, tipo de entrada, observación, medio de pago, valor.	
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión. 2. Navegar a la sección de "ingresos". 3. Dar clic en registrar nuevo ingreso. 4. Diligenciar el formulario con los datos del ingreso. 5. Hacer clic en "Registrar ingreso". 	
Resultado esperado	Que el ingreso se pueda registrar correctamente, se muestra un mensaje de éxito y aparezca en el listado de ingresos .	
Resultado obtenido	<p>view {ingresos.create} not found.</p> <p>No se encuentra el archivo de la vista, se valida y la ruta no</p>	

	estaba bien definida.
	El ingreso fue registrado satisfactoriamente y se visualiza en el historial.

Nota: Caso de prueba para registrar un ingreso.

Fuente: Elaboración propia

Tabla. 21

Caso de prueba CP-FINANCIERA-002.

Caso de prueba #14	Código del CP	14
	¿Prueba de despliegue?	Si
Descripción	Registrar una salida.	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Verificar que se registre correctamente un egreso financiero en el sistema.	
Datos de entrada	Fecha, tipo de salida, observación, medio de pago, valor.	
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión. 2. Navegar a la sección de "Salidas". 3. Dar clic en "Registrar nueva salida" 3. Llenar los campos requeridos. 4. Hacer clic en "Registrar salida". 	
Resultado esperado	Que la salida se registre correctamente, que aparezca en el listado de salidas y se muestra un mensaje de éxito.	
Resultado obtenido	El egreso fue registrado correctamente y es visible en el historial.	

Nota: Caso de prueba para registrar una salida.

Fuente: Elaboración propia

Tabla. 22

Caso de prueba CP-FINANCIERA-003.

Caso de prueba #15	Código del CP	15
	¿Prueba de despliegue?	Si
Descripción	Registrar un nuevo cliente.	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Verificar que el sistema permita registrar correctamente un nuevo cliente.	
Prerrequisitos	Sistema operativo funcionando, acceso al sistema, formulario de registro de cliente habilitado.	

Datos de entrada	Identificación, nombre, apellido, teléfono, dirección, observaciones.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión. 2. Navegar a la sección "Cuentas por cobrar". 3. Hacer clic en "Crear cliente". 4. Diligenciar el formulario con los datos del cliente. 5. Hacer clic en "Crear cliente".
Resultado esperado	El cliente es registrado correctamente, aparece en la tabla de clientes y se muestra un mensaje de confirmación.
Resultado obtenido	<p>Al tratar de guardar un registro se genera el siguiente error: The cliente id field is required"</p> <p>La novedad se presenta porque en el controlador se define como PK el campo id del cliente y en la base de datos se define como cliente_id, se ajusta el controlador y se hace de nuevo la prueba.</p> <p>El cliente fue registrado exitosamente y se visualiza en la tabla.</p>

Nota: Caso de prueba para registrar un nuevo cliente.

Fuente: Elaboración propia

Tabla. 23

Caso de prueba CP-FINANCIERA-004.

Caso de prueba #16	Código del CP	16
	¿Prueba de despliegue?	Si
Descripción	Registrar una cuenta por cobrar.	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Verificar que el sistema permita registrar correctamente una cuenta por cobrar.	
Prerrequisitos	Tener al menos un cliente registrado. Formulario de cuentas por cobrar funcional.	
Datos de entrada	Cliente, tipo de servicio, fecha, valor, observaciones, estado .	
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión. 2. Navegar a la sección "Cuentas por cobrar". 3. Hacer clic en "Nueva cuenta por cobrar". 4. Completar los campos requeridos. 5. Hacer clic en "Registrar cuenta por cobrar". 	
Resultado esperado	La cuenta por cobrar queda registrada con estado "pendiente", aparece en el listado y se muestra un mensaje de éxito.	
Resultado obtenido	Al abrir el formulario se genera el siguiente error: 'unclosed '{' on line 22'	

	<p>Se encuentra que faltaba por cerrar una llave }, se ajusta y permite ingresar.</p> <p>La cuenta por cobrar fue registrada exitosamente y se visualiza correctamente en el sistema.</p>
--	---

Nota: Caso de prueba para registrar una cuenta por cobrar.

Fuente: Elaboración propia

Tabla. 24

Caso de prueba CP-FINANCIERA-005.

Caso de prueba #17	Código del CP	17
	¿Prueba de despliegue?	Si
Descripción	Registrar un pago de una cuenta por cobrar previamente registrada.	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Verificar que el sistema registre correctamente un pago, que quede registrado como un servicio prestado y actualizando el estado a "Pagada".	
Prerrequisitos	Tener una cuenta por cobrar en estado pendiente. Formulario de pago funcional.	
Datos de entrada	Monto del pago, estilista, servicio, método de pago, porcentaje.	
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión. 2. Navegar a "Cuentas por cobrar". 3. Seleccionar una cuenta pendiente. 4. Hacer clic en "Pagar". 5. Ingresar los datos en el formulario. 6. Hacer clic en "Confirmar pago". 	
Resultado esperado	<p>El estado cambia a "Pagada".</p> <p>Se registra el pago como un servicio prestado.</p> <p>Mensaje de éxito mostrado.</p>	
Resultado obtenido	El pago fue registrado correctamente, el estado de la cuenta se actualizó a "pagada", y el mensaje de confirmación se mostró.	

Nota: Caso de prueba para registrar un pago de una cuenta por cobrar

Fuente: Elaboración propia

Tabla. 25

Caso de prueba CP-FINANCIERA-006.

Caso de prueba #18	Código del CP	18
	¿Prueba de despliegue?	Si
Descripción	Registrar una nueva cuenta por pagar.	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Verificar que el sistema permita registrar correctamente una cuenta por pagar, estableciendo su descripción, valor, fecha de vencimiento y estado.	
Prerrequisitos	El módulo de cuentas por pagar debe estar habilitado y funcional.	
Datos de entrada	Descripción de la deuda, valor, fecha de vencimiento, estado (por defecto: pendiente).	
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión. 2. Navegar a “Cuentas por pagar”. 3. Hacer clic en “Registrar nueva cuenta por pagar”. 4. Ingresar la información requerida. 5. Hacer clic en “Registrar cuenta”. 6. Luego de creada se da en el botón editar y se valida que se pueda cambiar el estado a “paga”. 	
Resultado esperado	<p>La cuenta por pagar queda registrada correctamente con estado “Pendiente”, visible en el listado general, y se muestra un mensaje de confirmación de éxito.</p> <p>Se da en el botón editar y se cambia el estado del cuenta por pagar de forma exitosa.</p>	
Resultado obtenido	Cuenta por pagar registrada exitosamente, visible en el listado con estado pendiente.	

Nota: Caso de prueba para registrar una nueva cuenta por pagar.

Fuente: Elaboración propia

Tabla. 26

Caso de prueba CP-FINANCIERA-007.

Caso de prueba #19	Código del CP	19
	¿Prueba de despliegue?	Si
Descripción	Ejecutar el proceso de liquidación para las estilistas en un rango de fechas determinadas.	
Autor de la prueba	Yeison Ferney Pineda Ortiz	

Objetivo de la prueba	Verificar que el sistema calcule correctamente el monto total a pagar en comisiones a cada estilista, con base en los servicios prestados durante el periodo.
Prerrequisitos	Tener servicios registrados con sus respectivos precios y porcentajes de comisión. El módulo de liquidación debe estar funcional.
Datos de entrada	Fecha inicial y fecha final del periodo de liquidación.
Pasos	1. Iniciar sesión. 2. Navegar a la sección de “Liquidación de empleados”. 3. Seleccionar la fecha inicial y final. 4. Hacer clic en “Filtrar”.
Resultado esperado	El sistema muestra por cada estilista el total acumulado en comisiones para el periodo filtrado. Los cálculos corresponden al porcentaje aplicado sobre el precio de cada servicio.
Resultado obtenido	View {liquidaciones.index} not found. Se crea el archivo lidaciones/index.blade.php y permite ingresar. El sistema calcula correctamente las comisiones por estilista, visibles en el listado, con los valores esperados.

Nota: Caso de prueba para ejecutar el proceso de liquidación.

Fuente: Elaboración propia

Tabla. 27

Caso de prueba CP-FINANCIERA-008.

Caso de prueba #20	Código del CP	20
	¿Prueba de despliegue?	Si
Descripción	Realizar un cierre de caja.	
Autor de la prueba	Yeison Ferney Pineda Ortiz	
Objetivo de la prueba	Validar que el sistema calcule correctamente los saldos en efectivo y transferencia, y que detecte si hay faltantes, sobrantes o si el cuadro es correcto.	
Prerrequisitos	Tener operaciones registradas (ingresos, egresos, servicios) para el periodo seleccionado. El módulo de cuadro de caja debe estar funcional.	
Datos de entrada	Fecha de corte, saldo en efectivo, saldo en transferencia.	

Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión. 2. Navegar a la sección “Cuadre de caja”. 3. Seleccionar el mes y año a cerrar. 4. Hacer clic en “Filtrar”. 5. Diligenciar los valores reales en efectivo y transferencia. 6. Hacer clic en “Comparar saldos”.
Resultado esperado	El sistema muestra el total acumulado por método de pago (efectivo y banco). Calcula correctamente la diferencia entre lo registrado y lo real. Genera alerta si hay sobrante, faltante o si está cuadrado.
Resultado obtenido	El sistema presenta los saldos correctamente, identifica el estado del cuadro (cuadrado, sobra, falta) y guarda el cierre de caja con éxito.

Nota: Caso de prueba para realizar el cuadro de caja.

Fuente: Elaboración propia

6.6 Fase de implementación y documentación final

Durante esta fase del proyecto, se llevó a cabo la implementación definitiva de la aplicación web, junto con la elaboración y entrega de la documentación técnica final. Esta fase no sólo marca el cierre del desarrollo funcional del sistema, sino que también asegura la transferencia de conocimiento necesaria para su instalación, uso y mantenimiento por parte de usuarios y administradores.

Como productos principales de esta etapa se destacan los siguientes documentos:

Manual de instalación

El Manual de Instalación fue desarrollado con el fin de proporcionar una guía detallada para técnicos y administradores de sistemas sobre el proceso de despliegue de la aplicación. Incluye los requisitos de hardware y software, la configuración del entorno (Laravel, PHP, MySQL, Tailwind CSS), así como los comandos necesarios para instalar dependencias, ejecutar migraciones, compilar estilos y poner en marcha el sistema en un entorno local o productivo. También detalla los pasos de verificación inicial para confirmar una instalación exitosa y funcional.

Manual de usuario

El manual de usuario está orientado a capacitar a los diferentes perfiles que interactúan con el sistema, incluyendo empleados y administradores del salón de belleza. Contiene una introducción clara del propósito de la aplicación y una guía paso a paso para utilizar cada módulo: registro de servicios, gestión de clientes, ingresos, egresos, cuentas por pagar y cobrar, liquidaciones y cuadro de caja. También se explican funciones administrativas como la creación y edición de usuarios, además de ofrecer secciones de preguntas frecuentes y solución de problemas comunes, lo cual promueve la autonomía de los usuarios.

Ambos manuales representan un pilar fundamental para la operación y continuidad del sistema, ya que permiten que tanto la infraestructura técnica como las operaciones diarias del negocio sean gestionadas eficientemente por los usuarios finales. Esta documentación garantiza que el proyecto pueda mantenerse, escalarse y ser aprovechado al máximo, incluso por nuevos integrantes o personal sin conocimientos técnicos profundos.

7. Conclusiones

Las conclusiones del presente proyecto se derivan directamente del cumplimiento de los objetivos específicos y del objetivo general planteados inicialmente, así como de los alcances propuestos al inicio de este. Estos dan reflejo a los resultados tangibles y los aprendizajes obtenidos durante el transcurso del desarrollo, generando un aporte significativo tanto a nivel local para la entidad como a nivel personal y profesional para el estudiante.

7.1. Síntesis de los resultados obtenidos y cumplimiento de objetivos

El desarrollo e implementación de la aplicación web para Todo Estilo ha culminado con la entrega de un sistema funcional que cumple a cabalidad con el objetivo general de optimizar la gestión administrativa y financiera del salón de belleza. Este logro se sustenta en la consecución exitosa de cada uno de los objetivos específicos:

- **Centralización y automatización de procesos:** Se ha logrado consolidar la información financiera dispersa en un único sistema digital. La aplicación centraliza el registro de servicios, ingresos, egresos, cuentas por cobrar y obligaciones, eliminando la dependencia de métodos manuales y reduciendo drásticamente los errores asociados. Esto se traduce en una mayor precisión y confiabilidad de los datos financieros.
- **Eficiencia en la liquidación de estilistas:** La funcionalidad de liquidación de comisiones ha transformado un proceso que era tedioso y propenso a errores en una tarea automatizada y precisa. Al basarse en los servicios registrados, el sistema calcula automáticamente las comisiones correspondientes a cada estilista, ahorrando tiempo significativo y garantizando pagos justos y transparentes.
- **Mejora en el cuadro de caja y control de flujo:** La herramienta de cuadro de caja proporciona una visión clara y en tiempo real del flujo tanto del efectivo como el saldo en la cuenta bancaria. Al integrar ingresos, egresos y pagos de estilistas, el sistema permite obtener un saldo de caja exacto al final de cada periodo, lo cual era una de las principales debilidades de la gestión anterior. Esto habilita un control financiero mucho más robusto.
- **Toma de decisiones basada en datos:** Al disponer de información financiera organizada y accesible, los administradores del salón ahora cuentan con la capacidad de generar

reportes de ingresos y egresos de manera ágil. Esta visibilidad proporciona la base para la toma de decisiones estratégicas informadas, permitiendo identificar tendencias, optimizar gastos y planificar inversiones con mayor certeza.

- **Interfaz intuitiva y adaptable:** Se ha entregado una interfaz de usuario limpia, intuitiva y responsiva, diseñada para ser accesible y fácil de usar por el personal del salón, independientemente de su nivel de experiencia técnica. La implementación de componentes Blade y la adhesión a los principios de diseño de Tailwind CSS aseguran una experiencia de usuario consistente y agradable en diferentes dispositivos.
- **Plataforma sólida y escalable:** La elección de Laravel como framework, junto con la implementación de una arquitectura MVC y la modularización del código, ha resultado en una base de desarrollo robusta, mantenible y escalable. Esto asegura que la aplicación pueda crecer y adaptarse a futuras necesidades del salón, como la adición de nuevos módulos (ej. inventario, citas, CRM) sin requerir una reingeniería completa.

7.2. Aportes generados por el proyecto

El proyecto "aplicación web para la administración financiera del salón de belleza todo estilo" ha generado múltiples aportes en diversos niveles:

A nivel local (al salón de belleza Todo Estilo):

- **Optimización operativa directa:** Se ha brindado al salón una solución tecnológica a medida que resuelve su problemática central de gestión financiera, liberando recursos humanos y monetarios que antes se perdían en procesos manuales y errores.
- **Incremento de la rentabilidad y control:** Al tener una visión clara de ingresos y egresos, y un control riguroso de la caja, el salón está en mejor posición para maximizar su rentabilidad y minimizar las pérdidas por fugas de dinero o malas decisiones.
- **Modernización y competitividad:** La adopción de este sistema posiciona el negocio a la vanguardia tecnológica en su segmento de negocio en Itagüí, mejorando su imagen y su capacidad de adaptación a un mercado cada vez más digitalizado.
- **Mejora del clima laboral:** La automatización de la liquidación de estilistas reduce posibles conflictos y malentendidos sobre las comisiones, fomentando un ambiente de

trabajo más transparente y equitativo.

A nivel académico, profesional y personal (aporte como desarrollador):

- **Aplicación práctica de conocimientos:** El proyecto ha sido una oportunidad invaluable para aplicar y consolidar de manera integral los conocimientos adquiridos en el programa de la tecnología en desarrollo de software, abarcando desde el análisis de requisitos y el diseño de bases de datos hasta el desarrollo frontend/backend y el despliegue.
- **Dominio de metodologías ágiles:** Esto permitió experimentar de primera mano la flexibilidad y eficiencia de las metodologías ágiles, aprendiendo a gestionar un proyecto de forma iterativa y a adaptarse a los cambios de requisitos.
- **Experiencia en desarrollo de software a escala real:** Enfrentar un problema de negocio real y construir una solución funcional desde cero ha proporcionado una experiencia comparable a la de un proyecto en un entorno profesional, preparando al desarrollador para futuros desafíos.
- **Validación de habilidades técnicas:** El proyecto sirve como una sólida validación de las habilidades en PHP, Laravel, MySQL, y el desarrollo de interfaces de usuario modernas, lo que representa un activo significativo para la inserción laboral.

A nivel nacional e internacional:

- **Modelo de digitalización para PYMES:** La solución desarrollada puede servir como un modelo replicable y adaptable para otros salones de belleza o pequeñas y medianas empresas (PYMES) en Colombia (y potencialmente en otros países de la región) que enfrentan desafíos administrativos similares y buscan digitalizar sus operaciones.
- **Fomento de la cultura de datos:** El proyecto contribuye a fomentar una cultura de toma de decisiones basada en datos entre los pequeños negocios, lo cual es fundamental para su crecimiento y sostenibilidad en la economía digital.
- **Base para investigaciones futuras:** La aplicación, como caso de estudio, puede servir como punto de partida para futuras investigaciones en el campo de software para PYMES, la optimización de procesos de negocio o la aplicación de tecnologías emergentes en el sector de servicios.

Como conclusión general, la aplicación web desarrollada para Todo Estilo no solo resuelve una necesidad crítica y específica del salón de belleza, sino que también representa un logro significativo en la aplicación de principios de ingeniería de software para crear soluciones prácticas y eficientes, demostrando la capacidad de transformar desafíos de negocio en oportunidades de mejora a través de la tecnología.

8. Recomendaciones

Las siguientes recomendaciones surgen de los aprendizajes y las oportunidades de mejora identificadas durante el desarrollo de la aplicación. Están encaminadas a proporcionar una visión clara para futuros trabajos de grado, la evolución del sistema y la exploración de nuevas líneas de trabajo que puedan potenciar aún más la gestión del salón de belleza.

8.1. Mejoras y optimizaciones a corto y mediano plazo

Optimización del rendimiento y la carga de datos:

- **Visión:** Aunque el sistema actual es eficiente para el volumen de datos manejado, se recomienda implementar técnicas de optimización de consultas a base de datos como índices avanzados, optimización de relaciones “Eloquent” y estrategias de caché más agresivas para mejorar la velocidad de respuesta a medida que el volumen de datos y usuarios crezca.
- **Trabajos futuros:** Un proyecto de grado podría enfocarse específicamente en la auditoría y mejora del rendimiento de la base de datos y la aplicación.

Reportes y análisis financieros avanzados:

- **Visión:** Expandir la capacidad de generación de reportes para incluir gráficos interactivos, reportes de rentabilidad por estilista o tipo de servicio, análisis de flujo de efectivo proyectado y balances sencillos. Esto permitiría una comprensión más profunda del desempeño financiero.
- **Trabajos futuros:** El desarrollo de un módulo de Business Intelligence o analítica financiera sería un proyecto de grado con un enfoque fuerte en visualización de datos y minería de información transaccional.

Sistema de notificaciones personalizadas:

- **Visión:** Implementar notificaciones automáticas para eventos clave, como recordatorios

de pagos de cuentas por cobrar/obligaciones pendientes, alertas de bajo inventario de productos (si se integra ese módulo), o resúmenes diarios/semanales de ingresos/egresos.

- **Trabajos futuros:** Un proyecto centrado en la integración de servicios de mensajería (SMS, WhatsApp) o correo electrónico para notificaciones automatizadas.

Refinamiento de la interfaz de usuario y experiencia:

Visión: Aunque la interfaz es funcional, se puede realizar un estudio de usabilidad más profundo con usuarios reales para identificar puntos de fricción y aplicar mejoras en el flujo de trabajo y la navegación. Esto podría incluir animaciones sutiles o retroalimentación más explícita.

Trabajos futuros: Un proyecto de grado con enfoque en diseño de interacción y usabilidad, aplicando pruebas de usuario y rediseño de interfaces.

8.2. Complementos al proyecto actual (nuevos módulos)

Módulo de inventario de productos:

- **Visión:** Integrar un módulo para la gestión de productos de venta (shampoos, tintes, etc.), incluyendo registro de entradas y salidas, control de stock, alertas de mínimos y gestión de proveedores. Esto permitiría una visión holística del negocio.
- **Trabajos futuros:** El diseño e implementación de un módulo de inventario completo, que se integre con los ingresos (ventas de productos).

Módulo de gestión de citas y agendamiento:

- **Visión:** Desarrollar un sistema de agendamiento de citas online para clientes, con visualización de la disponibilidad de estilistas, confirmaciones automáticas y recordatorios. Esto optimizaría la programación y el servicio al cliente.
- **Trabajos futuros:** Un proyecto de grado enfocado en sistemas de agendamiento y sus complejidades (conurrencia, notificaciones, integración con calendarios).

Módulo de clientes (CRM básico):

- **Visión:** Crear un módulo para gestionar la información de los clientes, incluyendo historial de servicios, preferencias, datos de contacto y notas personalizadas. Esto permitiría un marketing más dirigido y un servicio al cliente mejorado.
- **Trabajos futuros:** Un proyecto enfocado en la implementación de un pequeño CRM con funcionalidades de seguimiento y comunicación con clientes.

8.3. Nuevas líneas de trabajo y proyectos de mayor envergadura

Integración con pasarelas de pago:

- **Visión:** Explorar la integración con pasarelas de pago electrónicas para permitir el registro de pagos online de servicios o productos, facilitando la contabilidad automatizada.
- **Trabajos futuros:** Un proyecto de grado centrado en la seguridad transaccional y la integración con APIs de pagos, abordando los desafíos técnicos y normativos.

Implementación de inteligencia artificial (IA) para análisis predictivo:

- **Visión:** Utilizar técnicas de machine learning para analizar los datos históricos de servicios y finanzas, permitiendo predecir tendencias de demanda de servicios, optimizar horarios de estilistas o identificar patrones de gasto.
- **Trabajos futuros:** Un proyecto de grado interdisciplinario que combine el desarrollo web con la ciencia de datos y la inteligencia artificial, utilizando los datos de la aplicación de Todo Estilo como caso de estudio.

Desarrollo de aplicación móvil nativa (Android/iOS):

- **Visión:** Complementar la aplicación web con una aplicación móvil nativa para estilistas (ej. para registrar servicios rápidamente desde el celular) o para clientes (agendamiento de citas y consultas).
- **Trabajos futuros:** Un proyecto de grado que aborde el desarrollo móvil nativo y la integración con una API REST desarrollada sobre la base del backend existente.

Estas recomendaciones son la visión a futuro que queda después del desarrollo del presente

proyecto, y también pueden inspirar a futuros estudiantes del programa de desarrollo de software a continuar desarrollando sobre la base del proyecto que se deja, lo cual es un gran aporte para la comunidad académica.

9. Referencias bibliográficas

- Agudelo Aguirre, A. A. (2021). *Análisis financiero corporativo* (Primera edición ed.). Editorial Universidad Nacional de Colombia.
- ANDI. (2024, Junio 25). Asociación Nacional de Empresarios de Colombia. *Reportes de la ANDI*.
<https://www.andi.com.co/Home/Noticia/17696-cada-colombiano-gasta-un-promedio-de-44>
- Amaya Balaguera, Y. D. (2013, 11 5). Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual. *Journal Technology*, 12(2), 111-124.
<https://dialnet.unirioja.es/descarga/articulo/6041502.pdf>
- Cabrera Bravo, C. C., Fuentes Zurita, M. P., & Cerezo Segovia, G. W. (2017, Octubre). La gestión financiera aplicada a las organizaciones. *Revista científica Dominio de las ciencias*, 3(4).
<https://dominiodelasciencias.com/ojs/index.php/es/index>
- Cevallos, V. A. (2010). El presupuesto como herramienta de gestión y planificación en una institución financiera pública. Tesis de grado previo a la obtención del título de licenciada en contabilidad y auditoría. Universidad Tecnológica Equinoccial. Recuperado de http://repositorio.ute.edu.ec/bitstream/123456789/11041/1/27132_1.pdf.
- Comisión Económica para América Latina y el Caribe (CEPAL). (2021). *Tecnologías digitales para un nuevo futuro*. Santiago.
<https://repositorio.cepal.org/server/api/core/bitstreams/879779be-c0a0-4e11-8e08-cf80b41a4fd9/content>
- España León, Á. R., Gonzáles Valero, M. I., Mejía Viteri, J. T., & Campi Mayorga, J. A. (2016). Patrón MVC. *Revista de Ciencia, Tecnología e Innovación*, 3(4).
<https://revista.uniandes.edu.ec/ojs/index.php/EPISTEME>

- Flórez Fernández, H., & Hernández Rodríguez, J. (2021). *Aplicaciones web con PHP*. Ediciones de la U.
- Grand View Research. (2023). *Beauty And Personal Care Products Market Size, Share & Trend Analysis Report*. <https://www.grandviewresearch.com/>. Retrieved 03 25, 2025, from <https://www.grandviewresearch.com/industry-analysis/beauty-personal-care-products-market>
- Lara, C., & Figueroa, L. (2020). Metodología ágil para el desarrollo de aplicaciones móviles educativas. *Experiencias Docentes*. https://sedici.unlp.edu.ar/bitstream/handle/10915/103770/Documento_completo.pdf-PDF_A.pdf?sequence=1&isAllowed=y
- López, M., Hernández, A., y Marulanda, C., Procesos y prácticas de gestión del conocimiento en cadenas productivas de Colombia, doi: 10.4067/S0718-07642014000300015, *Información Tecnológica*, 25(3), 125-134 (2014)
- Morero, F. (2000). *Introducción a la OOP*. (1st ed.). Grupo EIDOS. <https://kataix.umag.cl/~ruribe/Utilidades/Introduccion%20a%20la%20Programacion%20Orientada%20a%20Objetos.pdf>
- MOYER, R., McGuigan, J., & Kretlow, W. (2005). *Administración Financiera Contemporánea*. Editorial Thomson.
- Perdomo, A. (2012). *Administración financiera del capital de trabajo*. Editorial Internacional Thomson. México.
- Rodríguez, D. (2023, Agosto 28). El mercado de las peluquerías. *La republica*. <https://www.larepublica.co/empresas/el-mercado-de-las-peluquerias-genera-500-000-millones-al-ano-solamente-en-bogota-3688254>
- Rondón, F. (2012). *Contabilidad superior*. Distribuidora Lexis, C. A., Caracas.,

Sanabria, M., Gauthier Umaña, V., Méndez Romero, R. A., & Rivera Virgüez, L. (2022).

Transformación digital en las organizaciones (I. Páez Gabriunas, Ed.). Editorial

Universidad del Rosario.

https://www.researchgate.net/profile/Rafael-Mendez-Romero/publication/361167298_Transformacion_digital_en_las_organizaciones/links/62a095df6886635d5cc89026/Transformacion-digital-en-las-organizaciones.pdf#page=154

Torres Remon, M. Á. (2014). *Desarrollo de aplicaciones web con PHP*. Editorial Macro.

https://ebooks.arnoi.com/media/eb_0104/samples/9786123042486cap1-07.pdf

OROCHO GUASHPA, W. P., & CARRANZA LUCAS, J. K. (2017). *DESARROLLO DE UN SISTEMA*

WEB DE GESTIÓN FINANCIERA PARA LA FUNDACIÓN “JASPE” [Proyecto de grado para ingeniero en sistemas]. Escuela Superior Politecnica de Chimborazo.

<https://dspace.esPOCH.edu.ec/items/5d972a69-0f05-4f5d-850a-c326509c8e06>

Velarde de Barraza, O., Murillo de Velásquez, M., Gómez de Meléndez, L., & Castillo de Krol, F. (2006).

Introducción a la programación orientada a objetos (PRIMERA EDICIÓN ed.). PEARSON EDUCACIÓN.

<https://travezurasdeltraviezo.wordpress.com/wp-content/uploads/2014/02/introduccion-a-la-programacion-orientada-a-objetos.pdf>

Weston, J., & Copeland, T. (2012). *Finanzas en Administración*. Mc Graw-Hill. México.

10.Anexos

Anexo A. Aplicación web desarrollada (todo-estilo.zip)

Anexo B. Manual de usuario

Anexo C. Manual de instalación