

**Facultad de Tecnologías Libres**

**Desarrollo de un portal web para el salón de belleza TereStyle.**

Trabajo de diploma para optar por el título de   
Ingeniero en Ciencias Informáticas

**Autor:**

Yohan Manuel Travieso Alvarez

**Tutores:**

MsC. Daniel José Olazabal Guerra.

IngAlejandro Romero Martínez

Ing Adrián Santiago Nuevo Príncipe

La Habana, <mes> de 2025

Año 67 de la Revolución

**DECLARACIÓN DE AUTORÍA**

El(Los) autor(es) del trabajo de diploma con título <título de la tesis entre “comillas”, *en formato de letra Italic* y en **negrita**> concede(n) a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara(n) como único(s) autores de su contenido. Para que así conste firma(n) la presente a los <día> días del mes de <mes> del año <año>.

|  |  |
| --- | --- |
| **<nombre del autor>** | **<nombre del autor>** |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Firma del Autor | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Firma del Autor |
| **<nombre del tutor>** | **<nombre del tutor>** |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Firma del Tutor | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Firma del Tutor |

**Resumen**

***Abstract***

**Tabla de contenidos**

[Introducción 3](#_Toc191249837)

[Capítulo I: Fundamentos y referentes teórico-metodológicos sobre el objeto de estudio 9](#_Toc191249838)

[1.1 Conceptos fundamentales relacionados con el desarrollo de sitios web 9](#_Toc191249840)

[1.2 Análisis de sistemas homólogos 12](#_Toc191249844)

[1.2.1 Sistemas homólogos internacionales 12](#_Toc191249847)

[1.2.1.1 Fresha 12](#_Toc191249848)

[1.2.1.2 Booksy 13](#_Toc191249849)

[1.2.1.3 Mindbody 14](#_Toc191249850)

[1.2.2 Sistemas homólogos nacionales 14](#_Toc191249851)

[1.2.2.1 GesPel (Gestión de Peluquerías - Cuba) 14](#_Toc191249852)

[1.2.2.2 StyloCuba 15](#_Toc191249853)

[1.2.2.3 SalonSoft Cuba 15](#_Toc191249854)

[1.2.3 Valoración del estudio realizado 16](#_Toc191249855)

[1.3 Metodologías de desarrollo 17](#_Toc191249856)

[1.3 Herramientas y tecnologías de desarrollo 18](#_Toc191249858)

[Conclusiones del capítulo 27](#_Toc191249859)

[Capítulo II: Diseño de la solución propuesta al problema científico 28](#_Toc191249860)

[2.1 Descripción del negocio 28](#_Toc191249861)

[2.1.1 Actores del negocio 29](#_Toc191249862)

[2.2 Modelo Conceptual 30](#_Toc191249863)

[2.3 Propuesta de solución 31](#_Toc191249864)

[2.4 Especificación de los requisitos del sistema 32](#_Toc191249865)

[2.4.1 Requisitos funcionales 33](#_Toc191249866)

[2.4.2 Requisitos no funcionales 38](#_Toc191249867)

[2.5 Modelo de caso de uso del sistema 41](#_Toc191249876)

[2.5.1 Actores del sistema 41](#_Toc191249877)

[2.5.2 Diagrama de caso de uso del sistema 41](#_Toc191249878)

[2.5.3 Descripción de caso de uso del sistema 42](#_Toc191249879)

[2.6 Descripción de la arquitectura de software 49](#_Toc191249880)

[2.6.1 Patrón arquitectónico Modelo Vista Controlador 49](#_Toc191249881)

[2.7 Patrones de diseño 51](#_Toc191249882)

[2.7.1 Patrones GRASP 51](#_Toc191249883)

[2.7.2 Patrones GOF 52](#_Toc191249884)

[2.8 Modelo físico de datos 53](#_Toc191249885)

[2.9 Diagrama de clases del diseño 55](#_Toc191249886)

[2.10 Diagrama de secuencia 57](#_Toc191249888)

[2.11 Diagrama de despliegue 58](#_Toc191249889)

[Conclusiones del capítulo 60](#_Toc191249892)

[Capítulo III: Validación de la solución propuesta (nombre de la propuesta) 61](#_Toc191249894)

[III.1 Nombre del Epígrafe III.1 61](#_Toc191249895)

[III.2 Nombre del Epígrafe III.2 **Error! Bookmark not defined.**](#_Toc191249896)

[III.3 Nombre del Epígrafe III.3 67](#_Toc191249897)

[Conclusiones del capítulo 79](#_Toc191249898)

[Conclusiones 81](#_Toc191249899)

[Recomendaciones 82](#_Toc191249900)

[Referencias Bibliográficas 83](#_Toc191249901)

[ANEXOS 87](#_Toc191249902)

# Introducción

Con el constante avance de las Tecnologías de la Información y las Comunicaciones (TIC), es evidente el impacto positivo que tienen en diversos sectores, desde el académico hasta el comercial, al facilitar la digitalización y optimización de procesos que anteriormente requerían mucha intervención manual(Trujillo Sáez y Álvarez Jiménez 2021). En el ámbito empresarial, la adopción de herramientas digitales permite gestionar y organizar la información de manera más eficiente, y los sistemas de gestión de información se han consolidado como una solución integral que ayuda en la recopilación, procesamiento, almacenamiento y visualización de datos relevantes para apoyar la toma de decisiones y mejorar la interacción con los clientes(Tunque-Lizana, Quispe-Huaman y Tunque-Dueñas 2023).

Hoy en día, en el sector de la belleza y el cuidado personal, la presencia digital ha adquirido un papel fundamental en la visibilidad y crecimiento de los negocios. La implementación de plataformas digitales permite a los salones de belleza ofrecer una experiencia más cómoda y accesible para sus clientes, facilitando la consulta de servicios, precios y disponibilidad, además de posibilitar la interacción a través de distintos canales en línea. La creciente preferencia de los consumidores por realizar búsquedas y reservas en línea evidencia la necesidad de adaptación a las nuevas tendencias tecnológicas, ya que una presencia digital eficiente contribuye significativamente a la fidelización de clientes y al incremento en la demanda de los servicios ofrecidos.

El acceso a plataformas digitales también ha generado un cambio en el comportamiento de los clientes, quienes ahora esperan mayor comodidad, rapidez y flexibilidad en la prestación de servicios. Factores como la inmediatez en la respuesta a consultas, la disponibilidad de información clara y detallada, así como la posibilidad de realizar reservas de manera automatizada, se han convertido en aspectos clave en la experiencia del usuario(Guerra 2024). En este sentido, la falta de una estrategia digital puede representar una desventaja competitiva para los negocios que aún dependen de métodos tradicionales, limitando su crecimiento y reduciendo su alcance dentro del mercado.

A nivel global, el desarrollo y uso de herramientas digitales ha permitido a empresas del sector de la belleza expandir su mercado y mejorar su gestión interna. Grandes cadenas y pequeños emprendimientos han implementado sistemas de reserva en línea, catálogos digitales y estrategias de marketing digital para captar y retener clientes. Estos modelos han demostrado ser efectivos en la reducción de costos operativos, optimización del tiempo de trabajo y aumento de la satisfacción del cliente(Cyberclick 2025). En contraste, aquellos negocios que no han adoptado estas estrategias enfrentan desafíos como la baja visibilidad, dificultades en la gestión de citas y menor interacción con sus clientes potenciales.

En el caso específico de los salones de belleza en entornos urbanos altamente competitivos, contar con una presencia digital sólida es un factor determinante para su posicionamiento en el mercado. Negocios que han logrado integrar soluciones digitales han experimentado mejoras en la organización de su agenda, reducción de cancelaciones imprevistas y un incremento en la confianza de sus clientes, quienes valoran la transparencia y facilidad en la gestión de los servicios. La digitalización también permite una comunicación más fluida con los clientes, a través de recordatorios automatizados, notificaciones sobre promociones y atención personalizada mediante plataformas interactivas.

El entorno digital ofrece múltiples oportunidades para la expansión y crecimiento de los negocios, pero también implica desafíos en cuanto a su implementación y adaptación. La falta de conocimiento técnico, la inversión inicial y la resistencia al cambio son factores que pueden dificultar la adopción de tecnologías en algunos negocios tradicionales. Sin embargo, la tendencia global hacia la digitalización indica que aquellos emprendimientos que logren superar estas barreras podrán consolidarse en un mercado cada vez más orientado hacia lo digital. En este sentido, resulta fundamental analizar el impacto de la tecnología en el sector y comprender los beneficios que puede aportar la transformación digital en la gestión empresarial.

A partir de esta realidad, es esencial evaluar los desafíos que enfrentan los salones de belleza que aún no han implementado herramientas digitales en su gestión. Identificar las principales dificultades, las oportunidades de mejora y las estrategias que han resultado exitosas en negocios similares permitirá sentar las bases para la formulación de soluciones efectivas y adaptadas a las necesidades específicas de cada emprendimiento.

El salón de belleza TereStyle opera en un entorno altamente competitivo, donde la mayoría de los establecimientos de la zona han adoptado herramientas digitales para fortalecer su presencia en línea, mediante perfiles en redes sociales (Facebook, Instagram), grupos de WhatsApp y canales digitales. En contraste, TereStyle carece de una estrategia de visibilidad digital, lo que genera las siguientes limitaciones:

1. Gestión manual de reservas: La dependencia de métodos tradicionales (como agendas físicas o llamadas telefónicas) provoca errores recurrentes, como dobles reservas o ausencias no registradas, aumentando la carga administrativa y afectando la experiencia del cliente.
2. Limitada captación de nuevos clientes: Al no contar con una plataforma digital, el salón pierde oportunidades de atraer a un mercado cada vez más orientado a buscar servicios en línea, especialmente ante competidores que ofrecen promociones y comunicación ágil a través de medios digitales.
3. Dificultad para mostrar promociones de servicios: La falta de canales digitales integrados (como redes sociales, página web o boletines electrónicos) impide comunicar de manera efectiva promociones, descuentos especiales o lanzamientos de nuevos servicios. Esto limita la capacidad de atraer clientes interesados en ofertas temporales o personalizadas, reduciendo oportunidades para incrementar ventas y fidelizar a través de incentivos estratégicos.
4. Incapacidad para mostrar su oferta de manera efectiva: TereStyle no dispone de un espacio digital que permita exhibir su catálogo de servicios, productos utilizados (como tintes, tratamientos especializados) o una galería visual de estilos y cortes recientes, elementos clave para destacar en un sector basado en la estética y las tendencias.

Estas limitaciones no solo reducen su competitividad frente a otros salones, sino que también dificultan la optimización de procesos internos y la construcción de una relación moderna y dinámica con los clientes.

A partir de esta **situación problemática**, la presente investigación se estructura en torno al siguiente **problema de investigación**: ¿Cómo contribuir a mejorar la visibilidad del salón de belleza TereStyle en los medios digitales? De acuerdo a este planteamiento, se define el **objeto de estudio:** proceso de visibilidad web a través de medios digitales. El **campo de acción** se delimita comoproceso de visibilidad web del salón de belleza TereStyle a través de un portal web.

El **objetivo general** que guiará el desarrollo de la investigación es: Desarrollar un portal web que contribuya a mejorar la visibilidad el salón de belleza TereStyle en los medios digitales.

Para cumplir el objetivo, se plantean las siguientes **preguntas científicas**:

1. ¿Cuáles son los referentes teóricos fundamentales que sustentan la investigación relacionada con el proceso de desarrollo de un portal web para el salón de belleza TereStyle?
2. ¿Qué elementos deben tenerse en cuenta para el diseño del portal web que contribuya a mejorar la visibilidad del salón de belleza TereStyle?
3. ¿Qué componentes son necesarios para implementar un portal web que contribuya a mejorar la visibilidad del salón de belleza TereStyle?
4. ¿Qué resultados se obtendrán al validar el funcionamiento del portal web desarrollado para el salón de belleza TereStyle?

Para abordar efectivamente las preguntas cientificas planteadas y darle cumplimiento al objetivo de la investigación, se se plantean las **tareas de investigación** siguientes:

1. Analizar los referentes teóricos que sustentan el desarrollo de portales web como medio digital para mejorar la visibilidad web.
2. Diseñar un portal web que contribuya a mejorar la visibilidad web del salón de belleza TereStyle.
3. Implementar las funcionalidades del portal web para el salón de belleza TereStyle.
4. Validar el funcionamiento del portal web para el salón de belleza TereStyle.

Para cumplir las tareas propuestas se emplearán métodos de investigación teóricos, empíricos y el método matemático.

**Métodos teóricos:**

**Analítico-Sintético:** Este método permitira realizar un exhaustivo análisis de teorías y documentos relevantes al ámbito de los portales web y la interacción con los clientes en el sector de la belleza. A través de esta técnica, se identificaran las características y necesidades específicas del salón "TereStyle", lo que facilitará la consulta de los referentes teoricos necesarios para el desarrollo del portal web propuesto.

**Histórico-lógico**: Se utilizó para estudiar la evolución histórica y tendencias actuales de las soluciones digitales en el sector de la belleza, para la construcción de los referentes teóricos que sustentan la investigación, el estudio de soluciones similares, así como de las metodologías de desarrollo, las tecnologías, herramientas, lenguajes de programación y de modelado utilizados.

**Modelación:** Se utlizará un lenguaje de modelado unificado (UML) para mostrar los diferentes artefactos que se diseñarán como resultado del proceso de ingeniería de software. Estos permitirán describir las funcionalidades de los procesos, así como la interacción de los actores con el sistema. El modelado de estos artefactos está en correspondencia con los estándares de la metodología AUP-UCI escenario 2. Mediante diagramas de casos de uso, se representará la interacción de los actores con el sistema, los diagramas de secuencia detallarán el flujo de operaciones críticas, mientras que los diagramas de clases definirán la estructura de la base de datos y las relaciones entre entidades. Estos artefactos garantizarán una visión coherente del diseño, así como la alineación con los requisitos funcionales y no funcionales del proyecto.

**Métodos empíricos:**

**Análisis documental:** Para la revisión bibliográfica, la revisión de las fuentes de información, el estudio de documentos clasificados, elementos teóricos que sustentan la concepción sobre la creación de un portal web.

**Encuesta:** Se aplicarán al personal administrativo y a la gerencia del salón para identificar desafíos operativos específicos, como los errores recurrentes en la gestión manual de reservas o la falta de mecanismos para comunicar promociones. Estas conversaciones estructuradas permitirán profundizar en las causas raíz de los problemas y priorizar funcionalidades críticas**.**

**Método matemático:** Este método se empleará para el análisis y presentación de los resultados que se obtengan al aplicar los instrumentos, para lo cual se emplearán indicadores de la estadística descriptiva.

**Técnica de validación:**

**Técnica de IADOV:** La Técnica de Iadov se utilizará para medir la satisfacción de los usuarios finales del portal web "TereStyle". Esta técnica permitirá obtener una valoración de la aceptación y satisfacción de los clientes y administradores del salón respecto a la funcionalidad y facilidad de uso de la plataforma. Para llevar a cabo esta validación, se formularán una serie de preguntas en un cuestionario dirigido a usuarios del sistema (clientes y administradores), que evaluarán aspectos clave como la usabilidad del sistema de reservas, la claridad de la interfaz de usuario, y la funcionalidad de las diferentes secciones del portal (servicios, galería, blog, etc.).

Las respuestas a este cuestionario se analizarán utilizando el índice de satisfacción grupal de Iadov, permitiendo identificar el grado de aceptación del portal y las áreas de mejora. Este método de validación proporciona una forma estructurada de analizar las percepciones de los usuarios y ofrece una base para realizar ajustes en el sistema con el objetivo de optimizar la experiencia de usuario y la eficiencia operativa del portal(Mirabal Sarria y Torres Paez 2021).

# Capítulo I: Fundamentos y referentes teórico-metodológicos sobre el objeto de estudio

## En el presente capítulo se establecen los fundamentos teóricos y metodológicos que sustentan el desarrollo del portal web para el salón de belleza TereStyle. Se realiza un análisis de los conceptos clave relacionados con la visibilidad digital, el diseño y desarrollo de sitios web, y las metodologías empleadas en este tipo de proyectos. Además, se estudian sistemas similares en el sector de la belleza para identificar buenas prácticas y elementos esenciales a considerar en la propuesta. Se presentan las tecnologías seleccionadas para la implementación del portal, justificando su elección en función de los requisitos del proyecto. Este marco teórico proporciona la base necesaria para abordar el problema identificado y garantizar una solución alineada con las tendencias actuales del mercado digital.

## 1.1 Conceptos fundamentales relacionados con el desarrollo de sitios web

## Sitio web

Un sitio web es un conjunto de páginas interconectadas alojadas en un servidor y accesibles mediante un dominio en Internet. Su principal función es comunicar información sobre una entidad, producto o servicio, ofreciendo una experiencia clara y accesible para los usuarios(Cabezas Sánchez 2023). Aunque puede presentar contenidos estáticos o dinámicos, generalmente se enfoca en la difusión de información sin incorporar funcionalidades interactivas avanzadas. En resumen, un sitio web se orienta a facilitar la navegación y consulta de datos relevantes de manera sencilla.

**Portal web**

Un portal web es una plataforma digital que centraliza servicios, recursos e información en un solo lugar. Integra herramientas interactivas que permiten a los usuarios realizar acciones como consultas y reservas, facilitando la interacción en tiempo real(Gutierrez Arana y León Gutierrez 2023). Además, suele incluir secciones especializadas y conexiones con aplicaciones externas, convirtiéndose en un punto de encuentro para diversas funcionalidades. Su diseño busca ofrecer una experiencia de usuario dinámica y adaptada a necesidades específicas, destacando su rol como una herramienta centralizada para la gestión de servicios.

**Diferenciación Funcional: Por qué nuestra propuesta se clasifica como un portal web**

En cuanto a por qué se describe la propuesta como un portal y no simplemente como un sitio web, la distinción radica en la funcionalidad y el alcance de la solución. Mientras que un sitio web tradicional se limita a exhibir información, nuestra propuesta integra múltiples módulos interactivos (por ejemplo, gestión de reservas y visualización dinámica de servicios) que facilitan la comunicación directa y la administración en tiempo real. Esta centralización de funciones y la capacidad de ofrecer una experiencia personalizada y operativa hacen que la solución se configure como un portal web, con un enfoque integral en la optimización de procesos y en la interacción con el usuario.

## SEO (Search Engine Optimization)

## El SEO es un conjunto de estrategias y técnicas orientadas a mejorar la visibilidad de un sitio web en los motores de búsqueda. Esto se logra mediante la optimización del contenido, el uso adecuado de palabras clave, la mejora en la estructura del código HTML y la generación de enlaces de calidad(López 2024). En el caso de TereStyle, una correcta implementación de SEO permitirá que el portal web aparezca en los primeros resultados cuando los usuarios busquen servicios de belleza en su área geográfica, aumentando así la captación de clientes.

**Visibilidad digital**

La visibilidad digital se refiere al nivel de exposición que tiene un negocio en el entorno digital. En el contexto del portal web de TereStyle, esto implica no solo el posicionamiento en buscadores, sino también la integración con redes sociales, directorios locales y estrategias de marketing digital. Una alta visibilidad digital facilitará que potenciales clientes encuentren fácilmente los servicios del salón, mejorando su competitividad en el mercado(Aguilar 2023).

**Experiencia de usuario (UX - User Experience)**

El término UX se refiere a la percepción y sensación que tiene un usuario al interactuar con un sitio web. Un portal con una buena UX es aquel que ofrece navegación intuitiva, carga rápida, accesibilidad y un diseño atractivo. En TereStyle, se priorizará una interfaz limpia y sencilla que permita a los clientes explorar los servicios, conocer precios y reservar citas sin complicaciones.

**Interfaz de usuario (UI - User Interface)**

La UI hace referencia al diseño visual del sitio web, incluyendo los colores, tipografías, botones e imágenes utilizadas(Wood 2022). Un portal con una UI bien estructurada transmite confianza y profesionalismo. En el caso de TereStyle, la interfaz se diseñará con una estética acorde al sector de la belleza, utilizando colores armónicos y una presentación clara de los servicios.

**Accesibilidad web**

La accesibilidad web se enfoca en garantizar que cualquier usuario, independientemente de sus capacidades o limitaciones físicas, pueda interactuar con el sitio web(Domecq Babie y Niño Benitez 2021). Esto incluye adaptaciones como la compatibilidad con lectores de pantalla y la correcta disposición de elementos en dispositivos móviles. Dado que TereStyle busca llegar a un público amplio, se implementarán buenas prácticas de accesibilidad para mejorar la experiencia de todos los usuarios.

**Responsive Design (Diseño adaptable)**

El diseño adaptable es una metodología que permite que un sitio web se visualice correctamente en distintos dispositivos, desde computadoras hasta teléfonos móviles(Suarez y Andrea 2024). Dado que muchos clientes de TereStyle acceden a internet desde sus smartphones, se aplicará un diseño responsive para garantizar una correcta visualización y funcionalidad en cualquier pantalla.

**Seguridad web**

La seguridad web es fundamental para proteger los datos de los usuarios y la integridad del portal. Esto incluye el uso de certificados SSL, protección contra ataques cibernéticos y almacenamiento seguro de información. En el caso de TereStyle, se implementarán medidas de seguridad para resguardar los datos de los clientes y garantizar una navegación confiable.

**Medios digitales y su impacto en la visibilidad**

Los medios digitales, como redes sociales, blogs y plataformas de video, juegan un papel clave en la promoción de un negocio(Avendaño Peña y Cardona Arcila 2023). Para TereStyle, el portal web funcionará como un punto central de información, complementado con estrategias en redes sociales para aumentar su alcance y fidelizar clientes.

## 1.2 Análisis de sistemas homólogos

## En la actualidad, el desarrollo de soluciones web para la gestión de salones de belleza ha experimentado un crecimiento significativo. Diversas plataformas internacionales y nacionales han abordado la problemática con diferentes enfoques, ofreciendo funcionalidades variadas que pueden servir de referencia para el desarrollo del portal web de TereStyle.

## El objetivo de este estudio es analizar sistemas homólogos existentes, tanto a nivel internacional como en Cuba, para identificar elementos de diseño, arquitectura, lenguajes de programación y funcionalidades comunes que puedan tomarse como referencia en la definición de los requisitos funcionales de la propuesta de solución.

### 1.2.1 Sistemas homólogos internacionales

#### 1.2.1.1 Fresha

Fresha es una de las plataformas líderes en la gestión de salones de belleza y spas. Su sistema permite la gestión integral de un negocio de belleza con herramientas avanzadas para la administración de citas, control de inventario, procesamiento de pagos y comunicación con clientes. Es utilizada en más de 120 países y ha revolucionado la forma en que los salones interactúan con sus clientes. Entre sus principales características destacan:

* Arquitectura: Basada en la nube con acceso web y aplicación móvil permitiendo la sincronización en tiempo real.
* Lenguajes y tecnologías: React.js para el frontend, Node.js en el backend, con bases de datos PostgreSQL.
* Funcionalidades relevantes:
* Reserva de citas en línea con recordatorios automáticos por correo y SMS.
* Integración con sistemas de pago como Stripe y PayPal.
* Gestión de clientes con historial de servicios y preferencias.
* Aplicación móvil para administradores y clientes.
* Herramientas de análisis y reportes detallados sobre el rendimiento del negocio.

#### 1.2.1.2 Booksy

Booksy es una plataforma global diseñada para facilitar la gestión de reservas en la industria de la belleza y bienestar. Se ha convertido en una herramienta esencial para negocios de este sector gracias a su enfoque en la automatización y la accesibilidad. Entre sus principales características destacan:

* **Arquitectura:** Basada en microservicios con backend en Node.js y frontend en Vue.js, permitiendo escalabilidad y modularidad.
* **Funcionalidades clave:**
* Integración con redes sociales para la captación de clientes y la programación de citas desde Facebook e Instagram.
* Generación de reportes detallados sobre ingresos, retención de clientes y tendencias de servicios.
* Aplicación móvil que permite a los clientes gestionar sus citas de manera sencilla.
* Notificaciones automáticas por SMS y correo electrónico.

#### 1.2.1.3 Mindbody

Mindbody es una plataforma integral diseñada para la industria del bienestar y la belleza, enfocándose en la fidelización de clientes y la automatización del marketing. Este software ayuda a los negocios a gestionar sus operaciones diarias, optimizando la programación de citas y mejorando la experiencia del cliente mediante herramientas de marketing integradas. Entre sus principales características destacan:

* **Arquitectura:** Plataforma SaaS (Software como Servicio) con infraestructura en la nube.
* **Lenguajes y tecnologías:** Backend en Java con Spring Boot y frontend en React.
* **Funcionalidades destacadas:**
* Automatización de marketing con envío de correos y promociones personalizadas.
* Gestión avanzada de empleados y comisiones.
* Aplicación móvil para clientes con opciones de pago, suscripciones y fidelización.
* Integración con software de contabilidad y finanzas.

### 1.2.2 Sistemas homólogos nacionales

#### 1.2.2.1 GesPel (Gestión de Peluquerías - Cuba)

GesPel es un software cubano desarrollado para la gestión de salones de belleza, enfocado en la optimización de recursos internos. A pesar de ser una aplicación de escritorio, su implementación en diversos salones cubanos ha demostrado su utilidad en la organización de negocios de este sector. Entre sus principales características destacan:

* **Arquitectura:** Aplicación de escritorio basada en SQLite.
* **Funcionalidades relevantes:**
  + Control de inventario de productos.
  + Gestión de turnos y clientes.
  + Reportes básicos sobre ingresos y servicios prestados.

#### 1.2.2.2 StyloCuba

StyloCuba es una plataforma web emergente que busca digitalizar la gestión de salones de belleza en Cuba. Su desarrollo responde a la creciente demanda de soluciones en línea para mejorar la eficiencia en la administración de turnos y servicios. Esta iniciativa se alinea con la transformación digital que está experimentando Cuba en los últimos años, como parte de la política integral para el perfeccionamiento de la informatización de la sociedad cubana(Fernández 2023). Entre sus principales características destacan:

* **Arquitectura:** Basada en PHP con MySQL.
* **Funcionalidades clave:**
  + Reserva en línea con confirmación por SMS.
  + Historial de clientes y preferencias.
  + Módulo de ventas de productos.

#### 1.2.2.3 SalonSoft Cuba

SalonSoft Cuba es una solución local enfocada en la gestión de citas y empleados. Su enfoque modular permite a los salones personalizar sus procesos según sus necesidades. Entre sus principales características destacan:

* **Arquitectura:** Aplicación web con backend en Laravel y frontend en Vue.js.
* **Funcionalidades destacadas:**
  + Gestión de empleados y horarios.
  + Registro de clientes y servicios.
  + Reportes de facturación.

#### 1.2.3 Valoración del estudio realizado

A continuación, se presenta una tabla con los criterios evaluados en cada sistema homólogo:

***Tabla 1.*** *Comparación de sistemas homólogos.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | **Gestión de citas** | **Integración con redes sociales** | **Gestión de clientes** | **Confirmación de vía SMS o correo electrónico** |
| Fresha | Si | No | Si | Si |
| Booksy | Si | Si | Si | Si |
| Mindbody | Si | No | Si | No |
| GesPel | Si | No | Si | No |
| StyloCuba | Si | No | Si | No |
| SalonSoft Cuba | Si | No | Si | No |

Tras el análisis de los sistemas homólogos internacionales (Fresha, Booksy, Mindbody) y nacionales (GesPel, StyloCuba, SalonSoft Cuba), se concluye que ninguno puede ser adoptado directamente para el salón TereStyle. Los sistemas internacionales, aunque cuentan con funcionalidades avanzadas como reservas en línea, integración con pasarelas de pago y gestión de clientes, están diseñados para mercados globales con infraestructuras tecnológicas y necesidades operativas distintas a las de Cuba. Por ejemplo, Fresha y Booksy dependen de servicios de pago internacionales (Stripe, PayPal) no accesibles en el contexto local, lo que limita su aplicabilidad.

En cuanto a las soluciones nacionales, GesPel y SalonSoft Cuba son aplicaciones de escritorio o web con enfoque en la gestión interna, carentes de integración con redes sociales y confirmación automatizada de citas mediante SMS o correo, elementos críticos para mejorar la visibilidad digital de TereStyle. StyloCuba, aunque emergente, no aborda la gestión de promociones ni la exhibición dinámica de servicios, aspectos clave para destacar en un sector basado en tendencias estéticas.

Estos elementos servirán como referencia para definir requisitos funcionales del portal web, priorizando la integración con redes sociales, la confirmación de citas vía SMS y la visualización interactiva de servicios. La ausencia de una solución que combine accesibilidad local, visibilidad digital y gestión eficiente justifica el desarrollo de un portal personalizado, adaptado a las necesidades específicas de TereStyle y al contexto tecnológico cubano.

## 1.3 Metodologías de desarrollo

En el desarrollo de software la necesidad de organizar o estructurar de forma correcta y disciplinada, es uno de los factores más importantes para evitar pérdidas de tiempo y recursos. Para evitar tales errores es preciso definir una estrategia para darle un orden a las tareas posibles a desarrollar, así como también llevar a cabo una guía de cómo efectuar las actividades, en fin, llevar a cabo un conjunto de procedimientos y pasos que se deben de seguir para el desarrollo de un software constituyendo los mismos una metodología de desarrollo de *software*.

Para guiar el desarrollo del portal web de TereStyle, se seleccionó la metodología *Proceso Unificado Ágil (AUP-UCI)*, una variante adaptativa que combina principios ágiles con un enfoque estructurado(Martell León 2021a). Esta elección se fundamenta en su capacidad para integrar iteraciones rápidas con una gestión disciplinada de requisitos, ideal para proyectos que requieren flexibilidad ante cambios y una documentación clara de los procesos. Si bien AUP-UCI es la metodología institucional en la Universidad de las Ciencias Informáticas, su pertinencia radica en su enfoque incremental, que permite ajustar el desarrollo según las necesidades emergentes del cliente, y su énfasis en la trazabilidad de los artefactos generados, esencial en un entorno académico-profesional.

La metodología AUP-UCI se estructura en **tres etapas principales**:

1. **Inicio:** En esta fase se definen los objetivos del proyecto, se identifican los riesgos clave y se establece el plan de trabajo, incluyendo la asignación de recursos y plazos.
2. **Ejecución:** Aquí se desarrollan las actividades técnicas, como el modelado de requisitos, el diseño arquitectónico y la implementación iterativa del sistema, priorizando funcionalidades críticas.
3. **Cierre:** Se evalúan los resultados finales contra los objetivos iniciales, se realizan pruebas integrales y se prepara la documentación para la entrega al cliente.

## Además, AUP-UCI integra siete disciplinas clave: modelado de negocio, requisitos, análisis y diseño, implementación, pruebas internas, pruebas de liberación y pruebas de aceptación, asegurando un ciclo de vida completo y coherente(Martell León 2021b).

Dentro de AUP-UCI, existen cuatro escenarios para abordar proyectos según su complejidad y alcance. Para este trabajo, se eligió el Escenario 2, el cual se centra en:

* El modelado conceptual del negocio, facilitando una representación visual de los procesos clave de TereStyle (reservas, gestión de servicios, interacción con clientes).
* La definición de casos de uso del sistema (CUS), que permiten especificar las funcionalidades desde la perspectiva de los usuarios finales.

Este escenario fue seleccionado sobre otros (como el Escenario 1, orientado a prototipos rápidos, o el Escenario 4, para sistemas críticos) debido a que prioriza la claridad en la especificación de requisitos y la alineación entre el modelo de negocio y las necesidades técnicas del portal web. Al basarse en casos de uso, garantiza que todas las interacciones (clientes, administradores, profesionales) queden documentadas de manera estructurada, reduciendo ambigüedades y facilitando la comunicación entre stakeholders. Además, su enfoque en el modelado conceptual resulta crucial para un proyecto donde la visibilidad digital y la experiencia del usuario son pilares fundamentales.

## 1.3 Herramientas y tecnologías de desarrollo

Para garantizar el desarrollo eficiente y escalable del portal web del salón TereStyle, se seleccionaron herramientas y tecnologías alineadas con los requisitos funcionales y no funcionales del proyecto. A continuación, se detallan las principales:

**Lenguaje de Modelado Unificado (UML 2.5.1)**

El Lenguaje de Modelado Unificado (UML 2.5.1) es una técnica de modelado visual empleada en la ingeniería de software, que incluye un conjunto cohesivo de diagramas. Fue creado para facilitar a los desarrolladores en la concepción, especificación, construcción y documentación de los componentes de sistemas de software. UML reúne las mejores prácticas en ingeniería que han demostrado eficacia en el modelado de sistemas amplios y complejos(Claus 2021). Es fundamental en el desarrollo de software orientado a objetos y se basa principalmente en notaciones gráficas para ilustrar el diseño de los proyectos.

Su selección se fundamenta en su estandarización en la industria, su capacidad para representar de manera clara los procesos del portal web (como reservas, gestión de clientes y comunicación), y su alineación con la metodología AUP-UCI Escenario 2, requerida para proyectos académicos en el contexto institucional.

**Visual Paradigm 17.0**

Visual Paradigm 17.0 es una herramienta de modelado multiplataforma que utiliza UML. Actualmente, hay una amplia gama de herramientas CASE (Ingeniería de Software Asistida por Computadora) disponibles para el desarrollo de software. Está diseñada para diversos usuarios, incluidos ingenieros de software, analistas de sistemas, analistas de negocios y arquitectos de sistemas, así como para cualquier persona interesada en desarrollar sistemas de software a gran escala de manera confiable mediante un enfoque orientado a objeto(Hernández Suárez 2022).

Esta herramienta facilita la colaboración en equipo y apoya el ciclo completo de vida del desarrollo de software. La versión 17.0 de Visual Paradigm incluye varias mejoras y nuevas características en comparación con versiones anteriores, como una interfaz de usuario optimizada y mayor capacidad para trabajar con modelos extensos. Se eligió frente a otras herramientas CASE por su integración nativa con UML 2.5.1, su soporte para metodologías ágiles, y su capacidad para generar documentación técnica automatizada, esencial en un proyecto académico que requiere trazabilidad y claridad en los entregables.

**Javascript 8.0**

JavaScript (JS) es un lenguaje de programación ligero e interpretado, orientado a objetos y con funciones de primera clase. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje más utilizado, aunque también se emplea en muchos entornos fuera del navegador, como Node.js. Es un lenguaje de script multiparadigma, basado en prototipos, dinámico y que admite estilos de programación funcional(Cassettai 2022). Funciona en tiempo de ejecución, lo que significa que el código fuente se interpreta y ejecuta al iniciar el programa. Su eficiencia se debe al uso de motores que convierten el código JavaScript en código de máquina, optimizando así las velocidades de ejecución.

Además, JavaScript incluye interfaces de programación de aplicaciones (APIs) en todos los navegadores para facilitar funciones básicas, siendo estas interfaces librerías integradas en los navegadores(Voronov 2022). Su adopción se justifica por su versatilidad para desarrollar tanto el frontend (interfaz interactiva) como el backend (mediante Node.js), su amplia comunidad de soporte, y su capacidad para implementar funcionalidades críticas como reservas en tiempo real y actualizaciones dinámicas sin recargar la página, clave para mejorar la experiencia del usuario en TereStyle.

**Lenguaje marcado**

**HTML5**

HTML5 es la versión más reciente de HTML y XHTML. Este estándar define un único lenguaje que puede ser escrito tanto en HTML como en XML. En esta versión se buscan resolver los problemas de versiones anteriores de HTML y se enfoca en las necesidades de las aplicaciones web, un ámbito que no estaba suficientemente abordado antes. HTML5 incluye una amplia variedad de funcionalidades que anteriormente requerían el uso de plugins de terceros, como Applets de Java o Flash(FERNANDO 2021).

Las mejoras en el manejo de multimedia son significativas; ahora, imágenes, videos, audios y fuentes de texto son mucho más fáciles de manipular, además de permitir la integración de gráficos vectoriales (SVG) y MathML para fórmulas matemáticas. Estas características permiten incluir y controlar contenido multimedia en la web sin necesidad de plugins ni APIs propietarias. El código HTML se compone de un archivo de texto delimitado por etiquetas (o "tags" en inglés), dentro de las cuales se inserta información diversa, así como objetos e imágenes para complementar el texto.

Se optó por HTML5 por ser el estándar actual para estructurar contenido web, su soporte nativo para elementos multimedia (esencial para la galería visual de estilos de TereStyle), y su compatibilidad con dispositivos móviles, garantizando una experiencia óptima en smartphones, donde la mayoría de los clientes accederán al portal.

**Hojas de estilo**

**CCS3**

CSS3 (Hojas de Estilo en Cascada) es el lenguaje utilizado para definir los estilos de los elementos HTML, incluyendo propiedades como tamaño, color, fondo, borde, entre otros(Sánchez 2024). La especificación de HTML5 se desarrolló teniendo en cuenta a CSS como responsable del diseño, lo que hace que la integración entre HTML y CSS sea fundamental para el desarrollo web. Para declarar estilos personalizados, CSS utiliza propiedades y valores, lo que se conoce como declaración. Su elección se debe a su papel indispensable en el diseño web moderno, permitiendo crear interfaces atractivas y adaptables, así como su sinergia con frameworks como Tailwind CSS, que extienden sus capacidades para agilizar el desarrollo.

**Marco de trabajo**

Un framework es un conjunto de herramientas, bibliotecas y convenciones de programación que facilita el desarrollo de aplicaciones. Se considera una arquitectura de software que modela las relaciones generales entre los componentes de un proyecto, ofreciendo una estructura y un enfoque de trabajo que las aplicaciones del mismo utilizan(García 2025a). La finalidad de los frameworks es asistir a los desarrolladores en la creación de software de manera eficiente y productiva, al integrar convenciones y patrones de diseño establecidos, lo que permite a diseñadores y programadores centrarse en los requisitos del proyecto.

**Framework Tailwind CSS**

Tailwind CSS es un framework de diseño de sitios web y aplicaciones web que se centra en la utilidad y la personalización. A diferencia de otros frameworks, como Bootstrap, Tailwind no proporciona componentes predefinidos, sino que permite a los desarrolladores construir diseños personalizados utilizando clases de utilidad. Esto le proporciona una flexibilidad considerable para crear interfaces adaptables y únicas(Zermeño 2024).

Tailwind se basa en una técnica de diseño que se ajusta automáticamente al tamaño del dispositivo en el que se visualiza, conocida como "responsive design" o diseño adaptativo, lo que lo hace muy popular entre los desarrolladores web. Proporciona un enfoque centrado en las clases que facilita el estilizado de elementos mediante la adición de clases HTML directamente en el marcado(Zermeño 2024).

Se seleccionó sobre alternativas como Bootstrap por su enfoque minimalista, que permite un diseño altamente personalizado acorde a la identidad visual de TereStyle, y su eficiencia para implementar diseños responsivos sin sobrecargar el código, crítico para optimizar el rendimiento del portal.

**React**

React es una biblioteca de JavaScript de código abierto diseñada para construir interfaces de usuario interactivas y dinámicas, especialmente en aplicaciones de una sola página (SPA). Desarrollada por Facebook, utiliza un enfoque basado en componentes que permite a los desarrolladores crear elementos reutilizables y gestionar el estado de la aplicación de manera eficiente mediante un DOM virtual, lo que optimiza el rendimiento al reducir las actualizaciones directas en el DOM real(Alvear Vaca 2023). Su arquitectura basada en el paradigma declarativo facilita la creación de vistas predecibles y su integración con otras bibliotecas o frameworks lo hace altamente versátil.

La elección de React para el desarrollo del portal web de TereStyle se basa en su capacidad para construir una interfaz de usuario moderna, interactiva y responsive, esencial para mostrar el catálogo de servicios, la galería de imágenes y el sistema de reservas en tiempo real. Además, su amplio ecosistema de soporte y su compatibilidad con herramientas como Tailwind CSS y Node.js garantizan un desarrollo ágil y alineado con las necesidades de usabilidad y estética del proyecto.

**Entorno de desarrollo integrado**

**Visual Code Studio 1.94.2**

Visual Studio Code es un editor de código fuente gratuito y de código abierto creado por Microsoft. Está disponible en plataformas Windows, macOS y Linux, y es compatible con diversos lenguajes de programación, como JavaScript, TypeScript, Python, C++, Java, PHP, Go y muchos más. Siendo el primer IDE de 64 bits, facilita el manejo de proyectos más grandes y cargas de trabajo complejas, respondiendo de manera más fluida y eficiente a las necesidades de los programadores(«Documentation for Visual Studio Code» 2025).

Ademas admite el control de versiones a través de Git, lo que permite clonar, crear y abrir repositorios de manera sencilla. Además, Visual Studio Code está integrado con sistemas de control de versiones como Git, lo que permite a los desarrolladores colaborar y mantener un historial de las versiones de su código. La versión 1.77 introduce varias mejoras respecto a versiones anteriores, incluidas optimizaciones en rendimiento y estabilidad, así como nuevas funcionalidades como la integración nativa con GitHub.

Su elección se basa en su extensibilidad mediante plugins (como ESLint para depuración de JavaScript), su integración con herramientas clave del proyecto (Git, Tailwind CSS) y su soporte multiplataforma, ideal para equipos de desarrollo que trabajan en distintos sistemas operativos.

**MariaDB**

MariaDB es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto, derivado de MySQL, diseñado para ofrecer alto rendimiento, escalabilidad y flexibilidad en entornos que requieren manejo de cargas de trabajo medias a grandes. A diferencia de las bases de datos sin servidor como SQLite, MariaDB opera bajo una arquitectura cliente-servidor, donde un servicio central gestiona las conexiones y operaciones, permitiendo un manejo eficiente de múltiples usuarios y transacciones concurrentes.

Es reconocido por su compatibilidad con SQL avanzado, incluyendo soporte completo para transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad) mediante motores de almacenamiento como InnoDB, lo que asegura integridad en escenarios de alta demanda. Su capacidad para escalar vertical y horizontalmente lo hace ideal para aplicaciones web dinámicas, plataformas de comercio electrónico o sistemas empresariales que requieren gestión de grandes volúmenes de datos. MariaDB incluye características modernas como tablas temporales, soporte nativo para JSON, optimización de consultas mediante columnStore y replicación avanzada, además de mantener una amplia compatibilidad con estándares y herramientas de ecosistemas como MySQL.

Aunque consume más recursos que alternativas ligeras como SQLite, su robustez y personalización (mediante motores de almacenamiento configurables) lo posicionan como una elección estratégica para proyectos que priorizan crecimiento a largo plazo, seguridad mediante roles granularizados y adaptabilidad a entornos cloud. La decisión de utilizar MariaDB en un portal como TereStyle, se justifica cuando se anticipan necesidades de escalabilidad inmediata, alta concurrencia de usuarios o integración con microservicios distribuidos, requiriendo una infraestructura más compleja pero potente para garantizar estabilidad y rendimiento sostenido.

**Node.js**

Node.js es un entorno de ejecución de JavaScript basado en el motor V8 de Google Chrome, que permite ejecutar código JavaScript en el lado del servidor. Su modelo de entrada/salida no bloqueante y orientado a eventos lo hace ideal para aplicaciones escalables y de alto rendimiento que requieren manejar múltiples solicitudes concurrentes(Catalán Calabuig 2023). Además, su sistema de módulos (NPM) proporciona acceso a una vasta biblioteca de paquetes que simplifican tareas como la gestión de rutas, la conexión con bases de datos y la implementación de APIs.

Se seleccionó Node.js para el backend del portal web de TereStyle por su capacidad para procesar solicitudes rápidas y gestionar la lógica de negocio, como la autenticación de usuarios, la gestión de reservas y la interacción con la base de datos SQLite3. Su compatibilidad con JavaScript y React favorece una integración fluida entre el frontend y el backend, reduciendo la complejidad del desarrollo y optimizando el rendimiento del sistema en un contexto de recursos limitados.

**Express**

Express es un framework minimalista y flexible para Node.js que simplifica la creación de aplicaciones web y APIs RESTful. Proporciona un conjunto de herramientas para gestionar rutas, middleware y solicitudes HTTP, permitiendo a los desarrolladores estructurar el backend de manera eficiente y robusta(Catalán Calabuig 2023). Su diseño ligero no impone restricciones excesivas, lo que lo hace adaptable a proyectos de diferentes escalas.

La inclusión de Express en el desarrollo del portal TereStyle se justifica por su capacidad para organizar la lógica del servidor de forma clara, facilitando la implementación de endpoints para funcionalidades críticas como la gestión de reservas, la autenticación de usuarios y la administración de servicios. Al combinarse con Node.js, Express ofrece una solución ágil y eficiente para manejar las interacciones entre el cliente y el servidor, asegurando una comunicación rápida y segura que mejora la experiencia del usuario.

**Git 2.42.0**

Git es un sistema de control de versiones distribuido diseñado para gestionar cambios en el código fuente durante el desarrollo de software. Permite el seguimiento de modificaciones, la colaboración en equipo mediante ramificaciones y la integración continua de funcionalidades(«Git - Docmentation» 2025). En este proyecto, Git se empleará para gestionar el versionado del código, facilitar pruebas de integración mediante la fusión controlada de ramas (como main y develop), y garantizar la trazabilidad de los cambios realizados. Su integración con Visual Studio Code y plataformas como GitHub optimiza la colaboración entre desarrolladores y tutores, asegurando un flujo de trabajo organizado y eficiente. La elección de Git se fundamenta en su estandarización en la industria, su capacidad para resolver conflictos de código y su compatibilidad con metodologías ágiles.

**Acunetix 14.7**

Acunetix es una herramienta automatizada de pruebas de seguridad especializada en la detección de vulnerabilidades en aplicaciones web, como inyecciones SQL, cross-site scripting (XSS) o configuraciones inseguras. En el contexto del portal web de TereStyle, Acunetix se utilizará para realizar escaneos profundos de seguridad, identificando y priorizando riesgos potenciales en el frontend y backend. Esto garantizará que el portal cumpla con estándares de protección de datos y reduzca la exposición a amenazas cibernéticas. Su selección se justifica por su precisión en la identificación de vulnerabilidades críticas, su capacidad para generar reportes detallados y su integración con entornos de desarrollo modernos.

## Conclusiones del capítulo

El estudio de los referentes teóricos y sistemas homólogos reveló que las soluciones existentes, tanto internacionales como nacionales, no satisfacen las necesidades específicas del salón TereStyle, especialmente en términos de gestión de clientes y confirmación automatizada de citas, lo que confirma la pertinencia de desarrollar un portal web a medida. Este hallazgo subraya la carencia de herramientas digitales adaptadas al contexto cubano, justificando la originalidad y necesidad de la propuesta. La elección de la metodología AUP-UCI Escenario 2 resultó idónea al permitir un modelado conceptual que alinea los procesos del salón con los requisitos técnicos, reduciendo ambigüedades y estableciendo una base sólida para el diseño. Asimismo, la selección de tecnologías como React, Node.js y MariaDB se valida por su capacidad para ofrecer un sistema ágil y escalable, adaptable a las limitaciones tecnológicas locales, lo que asegura viabilidad y eficiencia. Estos resultados sientan las bases teórico-metodológicas para abordar el problema de visibilidad digital, aportando un marco claro que orienta el desarrollo hacia una solución funcional y contextualizada.

# Capítulo II: Diseño de la solución propuesta al problema científico

En el presente capítulo se describe el negocio TereStyle, su funcionamiento actual y las características clave que definen su operación. Se analizan los procesos críticos que presentan limitaciones, así como los elementos esenciales para el diseño de la solución propuesta: el portal web. Se especifican los requisitos funcionales y no funcionales del sistema, se establece el modelo conceptual del negocio y se justifica la arquitectura tecnológica seleccionada. Además, se detalla la interacción entre los actores del sistema y las funcionalidades prioritarias para mejorar la visibilidad digital del salón.

## 2.1 Descripción del negocio

El salón de belleza TereStyle es un establecimiento dedicado a la prestación de servicios estéticos y de cuidado personal para el público en general. Entre los servicios que ofrece se encuentran cortes de cabello, aplicación de tintes, manicura y pedicura. Su equipo de trabajo está compuesto por estilistas especializados, técnicos en manicura y pedicura, así como personal administrativo encargado de la gestión del negocio y la atención al cliente.

El proceso de atención al cliente inicia cuando un usuario llega al salón o solicita información sobre los servicios disponibles. El personal encargado orienta a cada cliente en función de sus necesidades y preferencias, ofreciendo asesoramiento sobre los tratamientos y estilos más adecuados. Una vez seleccionado el servicio, se agenda una cita según la disponibilidad del salón y las preferencias del cliente. Durante la prestación del servicio, se emplean productos de alta calidad y técnicas actualizadas, con el objetivo de garantizar resultados satisfactorios y alineados con las tendencias del sector.

Actualmente, la gestión de citas y reservas se realiza de manera presencial o a través de llamadas telefónicas, lo que implica una administración manual del proceso. Esta metodología, aunque funcional, puede generar dificultades en la organización de la agenda y en la optimización del tiempo de trabajo del equipo.

La constante evolución del sector de la belleza y la creciente competitividad del mercado impulsan a TereStyle a buscar estrategias que le permitan mejorar la organización de sus procesos y fortalecer su relación con los clientes. La diferenciación en la calidad del servicio y la atención personalizada son aspectos fundamentales para mantener su posicionamiento en el sector y garantizar una experiencia única para su clientela.

#### 2.1.1 Actores del negocio

Un actor del negocio es cualquier individuo, grupo o entidad externa con la que TereStyle interactúa para facilitar la prestación de sus servicios y optimizar su funcionamiento. Cada actor asume un rol específico que contribuye a la calidad y organización de la atención brindada en el salón(Universidad Icesi et al. 2023). Se han identificado los siguientes actores del negocio:

***Tabla 1****. Actores del negocio*

|  |  |
| --- | --- |
| **Nombre del Actor** | **Descripción** |
| Cliente | Individuo que acude al salón para solicitar y disfrutar de los servicios ofrecidos, como cortes de cabello, tintes, manicura y pedicura. |
| Estilista | Profesional encargado de realizar cortes de cabello y aplicar tintes, brindando asesoramiento personalizado en función de las tendencias y preferencias. |
| Técnico en manicura y pedicura | Especialista responsable de llevar a cabo los tratamientos de manicura y pedicura, utilizando técnicas actualizadas y productos de alta calidad. |
| Personal administrativo | Encargado de gestionar citas, reservas y otros procesos administrativos, asegurando una coordinación efectiva y atención oportuna al cliente. |

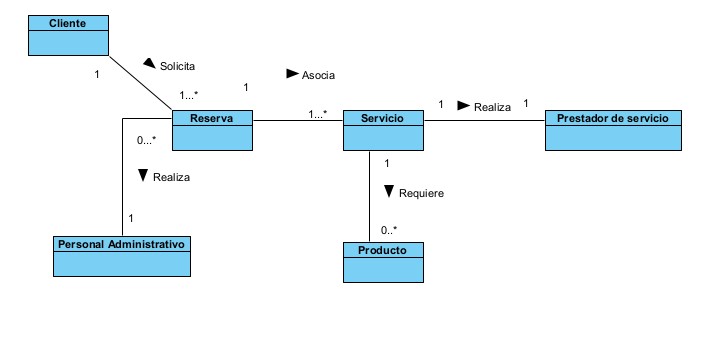
## 2.2 Modelo Conceptual

El modelo conceptual es una representación abstracta de los conceptos más significativos dentro del dominio del problema, en este caso, el funcionamiento del salón de belleza TereStyle(Al-Fedaghi 2021). Este modelo permite describir, de forma clara y comprensible para los usuarios y creadores del sistema, el conocimiento esencial relacionado con la prestación de servicios, las interacciones y los procesos internos del negocio. Desde el punto de vista del usuario, el modelo conceptual abarca:

* La información que debe conocer o adquirir sobre el funcionamiento del salón.
* La identificación de los elementos clave y las relaciones entre ellos.

En el siguiente modelo conceptual se representan los conceptos fundamentales que sustentan el negocio de TereStyle:

***Figura 1.*** *Modelo Conceptual*



En el siguiente modelo conceptual se representan los conceptos fundamentales que sustentan el negocio de TereStyle:

1. El concepto reserva representa la solicitud de un servicio por parte del cliente. Incluye información relevante como fecha, hora, servicio solicitado y la asignación del profesional correspondiente.
2. El concepto servicio abarca las distintas prestaciones ofrecidas por el salón, como cortes de cabello, tintes, manicura y pedicura.
3. El concepto cliente define a la persona que acude a TereStyle en busca de servicios de belleza, considerando sus preferencias y datos de contacto.
4. El concepto profesional (Estilista o Técnico en Manicura y Pedicura) representa al personal especializado encargado de realizar los servicios, cada uno con sus competencias específicas.
5. El concepto personal administrativo se encarga de gestionar citas, reservas y otros procesos administrativos, garantizando la correcta coordinación del negocio.
6. El concepto producto incluye los insumos y productos de alta calidad utilizados en la prestación de los servicios, fundamentales para asegurar resultados satisfactorios.

## 2.3 Propuesta de solución

Actualmente, TereStyle enfrenta desafíos derivados de la gestión manual de reservas y la atención al cliente, lo que ocasiona errores en la programación de citas y limita la capacidad de ofrecer una experiencia de servicio diferenciada. Con el objetivo de optimizar estos procesos y mejorar la interacción con la clientela, se plantea desarrollar un portal web para el salón de belleza TereStyle. Esta solución digital permitirá automatizar la gestión de reservas, facilitar el acceso a la información de los servicios y fortalecer la comunicación entre el salón y sus clientes.

El portal web se estructurará en diversos módulos, cada uno orientado a cubrir necesidades específicas del negocio:

* Gestión de Reservas: Permitirá a los clientes consultar disponibilidad y agendar citas reduciendo la dependencia de llamadas telefónicas y el manejo manual de agendas.
* Visualización de Servicios: Se presentará un catálogo interactivo de los servicios ofrecidos (cortes de cabello, tintes, manicura y pedicura), con descripciones, imágenes y promociones, facilitando que los clientes conozcan y seleccionen el servicio que mejor se adapte a sus necesidades.
* Gestión de Usuarios: permitirá la administración de cuentas de usuarios, tanto para clientes como para administradores, asegurando un control seguro y eficiente del acceso al sistema.
* Gestión de Galería: Se presentaran los trabajos realizados por el salón a través de una galería de imágenes, fortaleciendo su imagen de marca.
* Gestión de Promociones: Este módulo facilitará la creación y difusión de ofertas para atraer y fidelizar clientes.

En el contexto de esta propuesta, se definen los siguientes roles:

* Administrador del Portal: Encargado de gestionar la configuración y mantenimiento del sistema, actualizando la oferta de servicios y asegurando el correcto funcionamiento de la plataforma.
* Prestador de servicio: Profesionales (Estilistas y Técnicos) que, a través del portal, reciben y confirman las citas asignadas, gestionan su agenda y acceden al historial de reservas para planificar de manera eficiente su labor.
* Clientes: Usuarios finales que interactúan con el portal para consultar la información sobre los servicios, agendar citas y recibir confirmaciones automáticas, lo que contribuye a una experiencia de atención más fluida y satisfactoria.

## 2.4 Especificación de los requisitos del sistema

El sistema a desarrollar debe cumplir con una serie de requisitos que expresan de forma abstracta, general y formal los servicios que debe ofrecer y las limitaciones que debe respetar. Estos requisitos describen en detalle las funcionalidades del portal web, garantizando que la solución propuesta resuelva los problemas actuales en la gestión de reservas y atención al cliente, cumpla con los estándares de calidad requeridos y satisfaga las expectativas tanto del negocio como de sus usuarios.

Para definir y concretar los requisitos de la solución, se emplearon diversas técnicas de recopilación de información, fundamentales para el desarrollo exitoso de un portal que se ajuste a las necesidades de TereStyle. Las técnicas utilizadas fueron:

* **Entrevistas**: Se realizaron entrevistas con el personal administrativo, estilistas y técnicos para comprender los desafíos presentes en la gestión manual de citas, identificar necesidades específicas y recabar sugerencias sobre funcionalidades deseables en una plataforma digital.
* **Talleres de trabajo**: Se organizaron sesiones de intercambio de ideas entre el equipo de desarrollo y los responsables del salón. Estas reuniones permitieron compartir perspectivas, priorizar funcionalidades y definir los requisitos específicos, orientados a optimizar la experiencia del cliente y mejorar la organización interna.
* **Análisis de sistemas homólogos:** Se estudió el funcionamiento de portales web y aplicaciones digitales en el sector de la belleza, identificando buenas prácticas y requisitos funcionales y no funcionales que pueden ser adaptados al contexto de TereStyle.

#### 2.4.1 Requisitos funcionales

Los requisitos funcionales (RF) de un sistema son aquellos que especifican o definen una función que un sistema o componente debe cumplir(Alonso, 2024). En el contexto del portal web para el salón de belleza TereStyle, estos requisitos describen las acciones que el sistema debe permitir a los usuarios realizar.

Para la identificación de estos requisitos, se ha tomado en cuenta las limitaciones y necesidades del salón TereStyle, así como las funcionalidades comunes en portales web similares del sector. La prioridad de cada requisito fue determinada en base a su impacto en la mejora de la visibilidad del salón, la eficiencia en la gestión de reservas y la satisfacción del cliente(García-Peñalvo, García-Holgado y Vázquez-Ingelmo 2024). La complejidad se evaluó considerando el esfuerzo técnico necesario para implementar cada funcionalidad.

A continuación, la siguiente tabla muestra el resultado del levantamiento de los requisitos:

***Tabla 2.*** *Requisitos funcionales del sistema para el portal web TereStyle*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nombre** | **Descripción** | **Prioridad** | **Complejidad** |
| RF1 | Autenticar usuario | Permite la entrada del usuario al sistema utilizando su usuario y contraseña. | Alta | Baja |
| RF2 | Registrar usuario | Permite a los usuarios crear una cuenta proporcionando datos básicos como nombre, email y teléfono. | Alta | Media |
| RF3 | Editar nombre | Permite al usuario que este registrado cambiar su nombre. | Baja | Media |
| RF4 | Editar teléfono | Permite al usuario que este registrado cambiar su número de teléfono. | Baja | Media |
| RF5 | Editar email | Permite al usuario que este registrado cambiar su email. | Baja | Media |
| RF6 | Editar su nombre de usuario | Permite al usuario que este registrado cambiar su nombre de usuario. | Baja | Media |
| RF7 | Cambiar contraseña | Permite al usuario que este registrado cambiar su contraseña. | Baja | Media |
|  |  |  |  |  |
| RF8 | Crear usuario | Permite al usuario con rol Administrador crear usuarios | Alta | Media |
| RF9 | Modificar usuarios | Permite al usuario con rol Administrador Modificar los usuarios | Baja | Media |
| RF10 | Listar usuarios | Permite al usuario con rol Administrador listar los usuarios. | Baja | Baja |
| RF11 | Eliminar usuarios | Permite al usuario con rol Administrador Eliminar los usuarios. | Media | Media |
| RF12 | Asignar roles | Permite al usuario con rol Administrador de usuarios asignar los roles | Alta | Media |
| RF13 | Modificar rol de usuario | Permite al usuario con rol Administrador de usuarios modificar los roles. | Baja | Media |
| RF14 | Listar roles de usuarios | Permite al usuario con rol Administrador de usuarios Listar los roles. | Baja | Media |
|  |  |  |  |  |
| RF15 | Agregar reserva | Permite a los usuarios seleccionar un servicio, una fecha y hora disponible para reservar una cita en el salón. | Alta | Alta |
| RF16 | Modificar reserva | Permite a los usuarios modificar la fecha y hora de una cita antes de su confirmación por parte del personal. | Baja | Media |
| 1111 |  |  |  |  |
| RF17 | Listar reservas | Permite al usuario con rol cliente listar sus reservas. | Baja | Media |
| RF18 | Cancelar reserva | Permite a los usuarios cancelar una cita programada. | Media | Media |
|  |  |  |  |  |
| RF19 | Agregar servicio | Permite al administrador registrar un nuevo servicio en la plataforma con su respectiva descripción, precio e imagen. | Alta | Media |
| RF20 | Modificar servicio | Permite modificar los detalles de un servicio, como su nombre, precio, descripción e imagen. | Baja | Media |
| RF21 | Listar servicios | Muestra un catálogo con los servicios disponibles, incluyendo descripciones, precio e imágenes. | Baja | Baja |
| RF22 | Eliminar servicio | Permite al administrador eliminar un servicio que ya no esté disponible en el salón. | Media | Media |
|  |  |  |  |  |
| RF23 | Agregar imagen a galería | Permite subir fotos de trabajos realizados en el salón (peinados, cortes, etc.) para mostrar a los clientes. | Alta | Media |
| RF24 | Modificar imagen de la galería. | Permite modificar imágenes de la galería. | Media | Media |
| RF25 | Listar imágenes | Permite listar las imágenes. | Baja | Media |
| RF26 | Eliminar imágenes de la galería. | Permite eliminar imágenes de la galería. | Media | Media |
|  |  |  |  |  |
| RF27 | Crear promoción | Permite crear promociones especiales y descuentos para atraer nuevos clientes o fidelizar a los existentes. | Alta | Media |
| RF28 | Modificar promoción | Permite modificar las promociones | Baja | Baja |
| RF29 | Listar promociones | Permite listar las promociones que hayan sido creadas | Baja | Baja |
| RF30 | Eliminar promoción | Permite eliminar las promociones | Media | Baja |
| RF31 | Visualizar agenda | Permite que el prestador de servicio pueda visualizar la agenda de reservas que le han sido asignadas | Baja | media |
| RF32 | Actualizar estado de la reserva | Permite que el prestador de servicio pueda actualizar el estado de la reserva como “Realizada” | Baja | Media |

En la **tabla 2** anteriormente definida, la columna Número (No.) se refiere al identificador del requisito, por ejemplo, RF1 corresponde al requisito funcional número uno. La columna Nombre representa el título del requisito, mientras que la prioridad indica su importancia para el correcto funcionamiento del sistema.

Una vez definidos los requisitos funcionales, se realizó el proceso de refinamiento, agrupando los requisitos en casos de uso, de la siguiente manera:

* Gestionar usuarios (Crear usuario, listar usuarios, modificar usuarios, eliminar usuarios, asignar rol, modificar rol de usuario y listar roles de usuario).
* Gestionar reservas (Agregar reserva, modificar reserva, listar las reservas y cancelar reserva).
* Gestionar servicios (Agregar servicio, modificar servicio, ver listado de servicios y eliminar servicio).
* Gestionar galería (Agregar imagen a la galería, modificar imagen de la galería, listar imágenes y eliminar imagen de la galería).
* Gestión de promociones (Crear promoción, modificar promoción, listar promociones y eliminar promoción,).
* Gestionar perfil de usuario (Editar nombre, editar teléfono, editar email, editar nombre de usuario y editar contraseña).
* Autenticar Usuario
* Registrar usuario

#### 2.4.2 Requisitos no funcionales

Los requisitos no funcionales representan características generales y restricciones del sistema que no están directamente relacionadas con funciones específicas, pero que influyen en su comportamiento y desempeño(García-Peñalvo, García-Holgado y Vázquez-Ingelmo 2021). Estos incluyen aspectos como usabilidad, rendimiento, seguridad, mantenibilidad y portabilidad.

El modelo de calidad es una referencia clave para la evaluación del sistema, estableciendo las características necesarias para garantizar un producto de software que cumpla con las expectativas de los usuarios y aporte valor al negocio. Según la norma ISO/IEC 25010, los atributos de calidad del software se pueden categorizar en diferentes características y subcaracterísticas(ISO 25010 2022), las cuales guían la construcción y evaluación del portal web de TereStyle(Tamayo Espinosa et al. 2021).

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las nueve características de calidad que se muestran en la siguiente figura:

**Figura x.** ISO 25010(Normas ISO).

A continuación, se presentan los requisitos no funcionales definidos para la solución:

#### Usabilidad

Subcaracterística: Operabilidad

* **RnF1:** El sistema debe presentar una interfaz intuitiva y fácil de usar para los clientes y el personal del salón.
* **RnF2:** El 100% de los botones, formularios y secciones del portal deben contener mensajes y etiquetas claras con un máximo de 10 palabras redactados en lenguaje simple para guiar al usuario en su navegación.
* **RnF3:** El sistema debe utilizar títulos y metadatos adecuados para mejorar la visibilidad en motores de búsqueda y la comprensión del contenido.
* **RnF4:** El sistema debe emplear URLs claras y descriptivas (por ejemplo, "/servicios/corte-de-cabello") para facilitar la navegación y optimizar el SEO.

Subcaracterística: Estética

* **RnF5:** La interfaz del sistema debe seguir una estructura visual moderna y atractiva, utilizando una paleta de colores y tipografía alineada con la identidad del salón de belleza.

#### Eficiencia del desempeño

#### Subcaracterística: Tiempo de respuesta.

* **RnF6:** El sistema debe cargar las páginas en menos de 5 segundos bajo condiciones normales de red.

#### Portabilidad

Subcaracterística: Adaptabilidad

* **RnF7:** El sistema debe garantizar una visualización correcta (sin errores de diseño o funcionalidad) en el 100% de las pruebas realizadas en dispositivos con resoluciones entre 320px y 1920px, utilizando un diseño responsive.

#### Compatibilidad

Subcaracterística: Interoperabilidad

* **RnF8:** El sistema debe funcionar sin errores críticos en Chrome 90.0 o superior y Firefox 94.0 o superior.

#### Seguridad

Subcaracterística: Confidencialidad

* **RnF9:** Los datos de los clientes y del salón deben estar protegidos contra accesos no autorizados mediante encriptación y medidas de autenticación.
* **RnF10:** No se deben mostrar detalles técnicos específicos en caso de errores, para evitar exposición de vulnerabilidades.

Subcaracterística: Integridad

* **RnF11:** El sistema debe implementar roles y niveles de acceso, asegurando que cada usuario solo pueda acceder a la información y funciones que le correspondan.
* **RnF12:** Se debe garantizar que los datos no puedan ser alterados o eliminados sin la debida autorización.

#### Mantenibilidad

Subcaracterística: Analizabilidad

* **RnF13:** El código fuente debe estar bien estructurado y documentado para facilitar su mantenimiento y futuras modificaciones.

Subcaracterística: Modificabilidad

* **RnF14:** La plataforma debe permitir la adición de nuevas funcionalidades sin afectar el funcionamiento del sistema.

#### Confiabilidad

Subcaracterística: disponibilidad

#### RnF15: El sistema debe estar operativo al menos el 99% del tiempo, excluyendo mantenimientos programados.

## 2.5 Modelo de caso de uso del sistema

#### 2.5.1 Actores del sistema

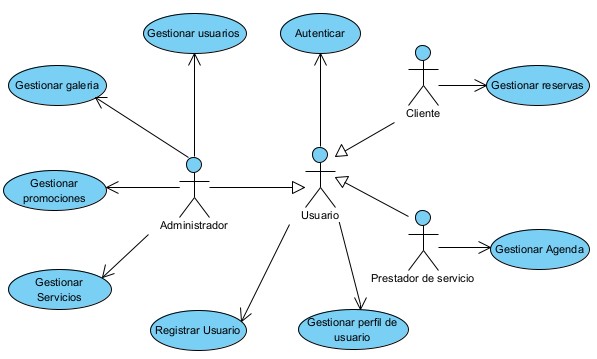
***Tabla 3.*** *Actores del sistema*

|  |  |
| --- | --- |
| Nombre del Actor | Descripción |
| Cliente | Usuario que accede al sistema para explorar los servicios disponibles, realizar reservas, modificar o cancelar citas, así como gestionar su perfil (cambiar contraseña, nombre, teléfono). |
| Administrador | Usuario con privilegios para gestionar la plataforma, incluyendo la administración de usuarios, servicios, promociones y la galería de imágenes. |
| Prestador de servicio (Estilista/Técnico en manicura y pedicura) | Usuario que interactúa con el sistema para visualizar la agenda de citas asignadas. |

#### 2.5.2 Diagrama de caso de uso del sistema

Los Diagramas de Casos de Uso son una técnica fundamental para capturar y visualizar cómo interactúan los actores con el sistema. Estos diagramas identifican los casos de uso (funciones o servicios que el sistema debe ofrecer), los actores (roles de usuario, que pueden ser personas, sistemas o entidades) y las relaciones entre ellos, proporcionando una visión clara de la operatividad del portal web de TereStyle.

***Figura 2.*** *Diagrama de casos de uso del sistema.*



## 2.5.3 Descripción de caso de uso del sistema

Para realizar la especificación de requerimientos del sistema se hace uso de las descripciones textuales de los casos de uso, técnica que propone el escenario 2 de la metodología AUP-UCI para encapsular los requerimientos funcionales del sistema.

***Tabla 4.*** *Descripción del caso de uso Gestionar Reservas.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Objetivo** | | El sistema debe permitir adicionar nuevas reservas, modificar reservas existentes y cancelar reservas. | |
| **Actores** | | Cliente: Usuario que accede al portal para solicitar servicios y reservar citas. | |
| **Resumen** | | El caso de uso se inicia cuando el actor decide adicionar, modificar, buscar, o cancelar una reserva. | |
| **Complejidad** | | Alta | |
| **Prioridad** | | Alta | |
| **Precondiciones** | | • El usuario debe estar autenticado en el sistema.  • Los datos del cliente y del profesional deben estar previamente registrados.  • El sistema debe contar con disponibilidad de horarios para el servicio solicitado. | |
| **Postcondiciones** | | • La reserva se registra, modifica o cancela en el sistema  • En caso de modificación o cancelación, se refleja la actualización en tiempo real. | |
| **Flujo de eventos** | | | |
| **Flujo básico < Gestionar Reserva >** | | | |
| **Actor** | | | **Sistema** |
| **1.** | El usuario ingresa a la opción “Reservas” desde la página principal del portal. | |  |
| **2.** |  | | Se muestra una pantalla con las opciones: Adicionar Reserva, Modificar Reserva y Cancelar Reserva. |
| **3.** | El usuario selecciona la acción que desea realizar. | |  |
| **Flujo básico < Adicionar Reserva >** | | | |
| **Actor** | | | **Sistema** |
| **1.** | Selecciona la opción “Adicionar Reserva” desde la pantalla de reservas. | |  |
| **2.** |  | | El sistema valida los datos ingresados:  a. Si la información es correcta y hay disponibilidad, se registra la reserva y se muestra el mensaje “Reserva realizada con éxito”.  b. Si falta información o hay conflicto de horario, se muestra un mensaje de error indicando “Por favor, corrija los errores señalados”. |
| **3.** |  | | La reserva se almacena en la base de datos y se actualiza la agenda. |
| **Flujo básico < Modificar Reserva >** | | | |
| **Actor** | | | **Sistema** |
| **1.** | Selecciona la opción “Modificar Reserva” en la pantalla de reservas. | |  |
| **2.** |  | | Se presenta una lista de las reservas activas asociadas al cliente. |
| **3.** | El usuario selecciona la reserva que desea modificar. | |  |
| **4.** |  | | Se abre un formulario prellenado con los datos actuales de la reserva. |
| **5.** | El usuario modifica los campos permitidos (cambiar fecha u hora, servicio y prestador de servicio) y presiona “Guardar”. | |  |
| **6.** |  | | El sistema valida la información actualizada; si es correcta, se actualiza la reserva y se muestra el mensaje “Reserva modificada correctamente”. |
| **7.** |  | | Se actualiza la agenda y se notifica el cambio al profesional asignado. |
| **Flujo básico < Cancelar Reserva >** | | | |
| **Actor** | | | **Sistema** |
| **1.** | Selecciona la opción “Cancelar Reserva” desde la pantalla de reservas. | |  |
| **2.** |  | | Se muestra una lista de las reservas activas. |
| **3.** | El usuario selecciona la reserva a cancelar. | |  |
| **4.** |  | | El sistema solicita confirmación con el mensaje “¿Está seguro de cancelar la reserva?”. |
| **4.1** | Al confirmar, la reserva se cancela y se muestra el mensaje “Reserva cancelada exitosamente” | |  |
| **4.2** |  | | La agenda se actualiza y se notifica al profesional asignado. |
| **Flujo alternativo Campos Vacíos** | | | |
| **Actor** | | | **Sistema** |
| **1.** |  | | Si el usuario intenta guardar una reserva sin completar todos los campos obligatorios, el sistema resalta los campos faltantes en rojo y muestra un tooltip “Este campo es obligatorio”. |
| **Flujo alternativo Datos inválidos** | | | |
| **Actor** | | | **Sistema** |
| **1.** |  | | Si se ingresan datos en formato incorrecto, se despliega un mensaje “Entrada inválida, por favor verifique la información” y no se permite guardar hasta corregir los errores. |

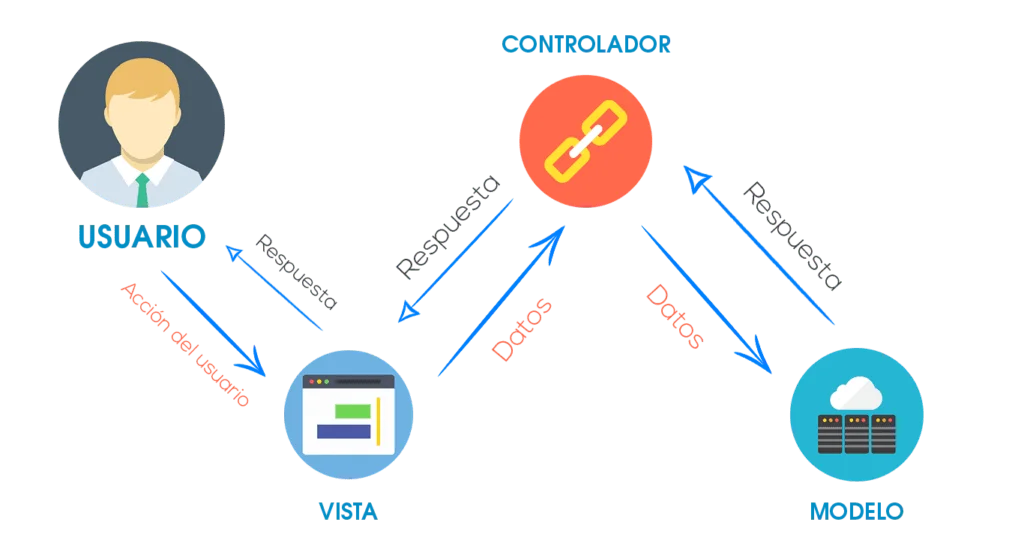
***Tabla 5.*** *Descripción del caso de uso Gestionar Servicio.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Objetivo** | | El sistema debe permitir al Administrador gestionar de forma integral la oferta de servicios del salón TereStyle, lo que incluye la capacidad de adicionar, modificar, eliminar y listar los servicios disponibles. Esto garantiza que el catálogo mostrado en el portal esté siempre actualizado y refleje de manera precisa la oferta del salón. | |
| **Actores** | | Administrador: Usuario con privilegios para agregar, modificar y eliminar los servicios ofrecidos por el salón. | |
| **Resumen** | | El caso de uso se inicia cuando el actor decide agregar, modificar, eliminar o listar los servicios disponibles. | |
| **Complejidad** | | Alta | |
| **Prioridad** | | Alta | |
| **Precondiciones** | | El usuario debe estar autenticado en el sistema como Administrador para poder realizar operaciones de alta, modificación o eliminación. | |
| **Postcondiciones** | | • Los cambios realizados se guardan en la base de datos y se reflejan de inmediato en el catálogo de servicios.  • Los clientes verán la información actualizada, garantizando coherencia en la oferta del salón. | |
| **Flujo de eventos** | | | |
| **Flujo básico < Gestionar Servicios>** | | | |
| **Actor** | | | **Sistema** |
| **1.** | El usuario ingresa a la sección “Servicios” desde la página principal del portal. | |  |
| **2.** |  | | Se muestra una pantalla con las siguientes opciones: Adicionar Servicio, Modificar Servicio, Eliminar Servicio y Listar Servicios. |
| **3.** | El Administrador selecciona la acción deseada (Agregar, Modificar o Eliminar), mientras que el Cliente puede acceder únicamente a la opción “Listar Servicios”. | |  |
| **Flujo básico < Adicionar Servicio>** | | | |
| **Actor** | | | **Sistema** |
| **1.** | Selecciona la opción “Agregar Servicio” desde la pantalla de servicios. | |  |
| **2.** |  | | Se despliega un formulario con los campos obligatorios: Nombre del Servicio, Descripción, Precio e Imagen. |
| **3.** | El Administrador completa el formulario y presiona el botón “Guardar”. | |  |
| **4.** |  | | El sistema valida la información ingresada:  a. Si todos los datos son correctos y se completaron los campos obligatorios, se registra el nuevo servicio y se muestra el mensaje “Servicio agregado exitosamente”.  b. Si falta información o los datos no cumplen con el formato esperado, se muestra un mensaje de error indicando “Por favor, corrija los errores señalados”. |
| **5.** |  | | El servicio se almacena en la base de datos y queda disponible en el catálogo del portal para que los clientes lo consulten. |
| **Flujo básico < Modificar Servicio>** | | | |
| **Actor** | | | **Sistema** |
| **1.** | Selecciona la opción “Modificar Servicio” en la pantalla de servicios. | |  |
| **2.** |  | | Se despliega una lista de los servicios registrados. |
| **3.** | El Administrador selecciona el servicio que desea modificar. | |  |
| **4.** |  | | El Administrador selecciona el servicio que desea modificar. |
| **5.** | El Administrador realiza las modificaciones necesarias y presiona “Guardar”. | |  |
| **6.** |  | | El sistema valida la nueva información; de estar correcta, actualiza los datos y muestra el mensaje “Servicio modificado correctamente”. |
| **Flujo básico < Eliminar Servicio>** | | | |
| **Actor** | | | **Sistema** |
| **1.** | Selecciona la opción “Eliminar Servicio” en la lista de servicios. | |  |
| **2.** |  | | El sistema solicita confirmación mediante un mensaje “¿Está seguro de eliminar este servicio?”. |
| **3.** |  | | Al confirmar, el sistema elimina el servicio de la base de datos y muestra el mensaje “Servicio eliminado con éxito”. |
| **4.** |  | | Al confirmar, el sistema elimina el servicio de la base de datos y muestra el mensaje “Servicio eliminado con éxito”. |
| **Flujo básico < Listar Servicios>** | | | |
| **Actor** | | | **Sistema** |
| **1.** | Selecciona la opción “Listar Servicios” desde la pantalla principal. | |  |
| **2.** |  | | El sistema despliega un catálogo con todos los servicios disponibles, incluyendo detalles como nombre, descripción, precio e imagen. |
| **3.** |  | | Se ofrece la posibilidad de filtrar y ordenar los servicios por categorías, precio u otros criterios relevantes |
| **Flujo alternativo Campos Vacíos** | | | |
| **Actor** | | | **Sistema** |
| **1.** |  | | Si el Administrador intenta guardar un servicio sin completar los campos obligatorios, el sistema resalta dichos campos en rojo y muestra un tooltip “Este campo es obligatorio”. |
| **Flujo alternativo Datos inválidos** | | | |
| **Actor** | | | **Sistema** |
| **1.** |  | | Si se ingresan datos en formato inválido (por ejemplo, un precio no numérico), el sistema muestra un mensaje “Entrada inválida, por favor verifique la información” y no permite guardar hasta corregir la entrada. |

## 2.6 Descripción de la arquitectura de software

La arquitectura de software es la base fundamental que garantiza la escalabilidad, mantenibilidad y eficiencia del portal web de TereStyle. Para cumplir con estos objetivos, se ha optado por el patrón de diseño Modelo Vista Controlador (MVC), que facilita la separación de responsabilidades en la aplicación y mejora la claridad en el desarrollo.

## 2.6.1 Patrón arquitectónico Modelo Vista Controlador

****

***Figura 3.*** *Arquitectura MVC*

A continuación, se analiza las diversas partes o conceptos en los que debemos separar el código de nuestras aplicaciones:

**Modelo:**  
Esta capa es responsable de la gestión de la información y la lógica de negocio. En el contexto de TereStyle, el Modelo se encarga de manejar la gestión de datos relacionados con las reservas, los servicios ofrecidos y el perfil de los usuarios. Se implementa utilizando técnicas de acceso a datos (por ejemplo, mediante una base de datos relacional como MariaDB) y se diseña para asegurar la integridad y consistencia de la información, encapsulando las operaciones CRUD (Crear, Leer, Actualizar y Eliminar).

**Vista:**  
La Vista constituye la interfaz de usuario del portal. Se encarga de presentar la información al usuario de forma clara y atractiva, garantizando una experiencia de usuario óptima y responsiva. En este proyecto, la capa de Vista se implementa utilizando HTML5, CSS3 (apoyado en frameworks como Tailwind CSS para asegurar un diseño moderno y adaptable) y JavaScript, lo que permite generar interfaces interactivas y dinámicas que facilitan la navegación y el acceso a los servicios del salón.

**Controlador:**  
El Controlador actúa como intermediario entre el Modelo y la Vista. Su función es interpretar las acciones del usuario, procesar la información recibida y coordinar las respuestas que deben reflejarse en la Vista. De este modo, el Controlador se encarga de validar las solicitudes, invocar las operaciones correspondientes en el Modelo y actualizar la interfaz de usuario de acuerdo con el resultado de estas operaciones.

## 2.7 Patrones de diseño

Los patrones de diseño son soluciones reutilizables a problemas comunes que surgen durante el desarrollo de software, proporcionando una estructura sólida y eficiente para organizar el código. Estos patrones facilitan la creación de sistemas más flexibles, mantenibles y escalables(Guerra Chaverra 2024). En el contexto del portal web TereStyle, se aplicaron diversos patrones de diseño GRASP y GOF con el objetivo de garantizar un desarrollo robusto y coherente.

## 2.7.1 Patrones GRASP

Estos patrones se utilizan en la programación orientada a objetos, con lo que el punto de partida para aplicarlos siempre será usar POO. Cómo definir estos objetos/clases será la aplicación de cada patrón. Con el objetivo de alcanzar una solución flexible y reusable, se pusieron en práctica los siguientes patrones GRASP que a continuación se describen:

* **Experto:** El patrón Experto asigna responsabilidades a las clases o módulos que poseen la información necesaria para realizar una tarea específica(Guerra Chaverra 2024). En este caso, se puede observar cómo la clase Reserva es responsable de gestionar toda la información relacionada con las reservas, como su creación, modificación y eliminación. Esta especialización asegura que cada entidad maneje únicamente los datos que le corresponden, mejorando así la organización del sistema.
* **Bajo Acoplamiento:** El principio de Bajo Acoplamiento busca reducir la dependencia entre los componentes del sistema(García 2025b). En el portal TereStyle, este patrón se refleja en la separación clara entre las capas del modelo MVC. Por ejemplo, la comunicación entre la capa de presentación (Vista) y la lógica de negocio (Modelo) se realiza mediante el Controlador, minimizando las interdependencias directas entre ellas. Esto facilita futuros cambios sin afectar significativamente otras partes del sistema.
* **Controlador:** El patrón Controlador define quién debe manejar las solicitudes provenientes del usuario y coordinar las respuestas del sistema(Reina [sin fecha]). En este proyecto, las clases controladoras como ReservaController.js y ServicioController.js actúan como intermediarios entre la interfaz de usuario y el modelo de datos. Estos controladores reciben las acciones realizadas por los usuarios (como agregar una reserva o modificar un servicio) y delegan las tareas correspondientes a las clases adecuadas, asegurando una gestión ordenada de las funcionalidades.
* **Creador:** El patrón Creador se utiliza para determinar qué clase es responsable de crear instancias de objetos. En TereStyle, este patrón se implementa principalmente en las clases controladoras y modelos. Por ejemplo, cuando un cliente solicita una nueva reserva, el ReservaController.js crea una instancia de la clase Reserva para registrarla en la base de datos. Este enfoque centraliza la creación de objetos y simplifica el proceso de mantenimiento del sistema.

## 2.7.2 Patrones GOF

Los patrones GoF, conocidos como el grupo de los cuatro, se clasifican en tres grupos dependiendo de sus propósitos, Creacional, Estructural y Comportamiento. Estos describen soluciones simples a problemas específicos en el diseño de software orientado a objetos y otros ámbitos referentes al diseño de interfaces(Velazco 2023).

**Patrón Decorador:** El patrón Decorador permite añadir funcionalidades adicionales a un objeto dinámicamente, sin alterar su estructura base. En el portal TereStyle, este patrón se aplica en la generación de vistas utilizando plantillas padre e hijas. Todas las páginas del portal heredan de una plantilla principal (layout.jsx), lo que facilita la consistencia visual y la reutilización de elementos comunes, como encabezados, menús y pies de página. Por ejemplo:

import React from 'react';

import Header from './Header';

import Footer from './Footer';

import Navigation from './Navigation';

function **Layout**({ children }) {

  return (

    <div>

      <Header />

      <Navigation />

      <main>

        {children}

      </main>

      <Footer />

    </div>

  );

}

export default **Layout**;

**Patrón Factory:** El patrón Factory se emplea para encapsular la lógica de creación de objetos, ocultando la complejidad detrás de su inicialización. En TereStyle, este patrón se implementa mediante el uso de interfaces y métodos estáticos que facilitan la creación de instancias de diferentes entidades. Por ejemplo, cuando se necesita crear una nueva reserva, el método create() del modelo Reserva gestiona todos los detalles técnicos relacionados con su construcción. De esta manera, cualquier cambio en la forma de crear reservas se limita a este único punto de acceso, promoviendo la cohesión y reduciendo errores.

## 2.8 Modelo físico de datos

Un modelo de datos físicos define todos los componentes y servicios lógicos necesarios para construir una base de datos operativa. Este modelo incluye la estructura de las tablas, los nombres y tipos de las columnas, las claves primarias y foráneas, así como las relaciones entre las diferentes tablas(ANTONIO 2022).

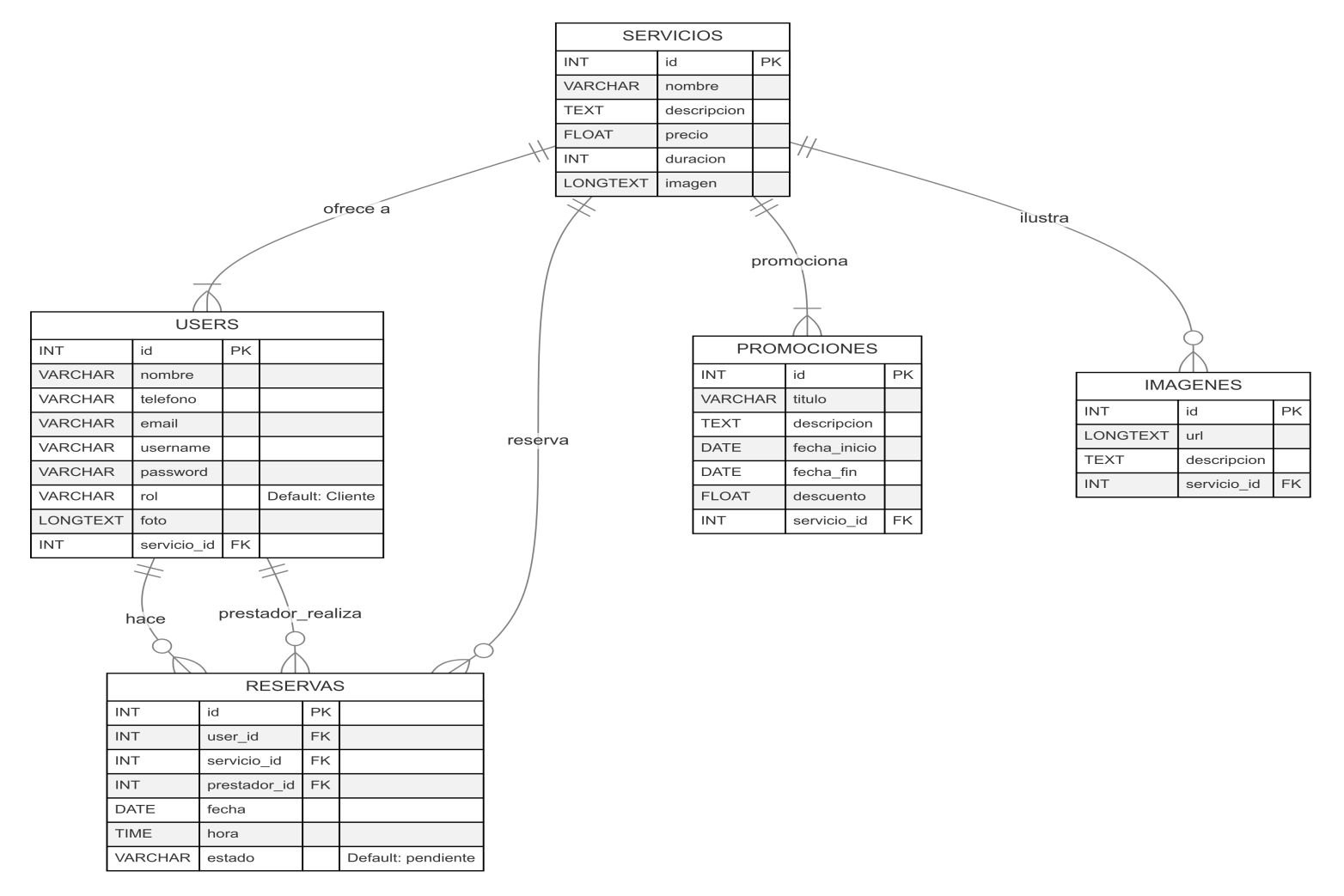
El diagrama entidad relación de la solución propuesta *Figura 4* está conformado por la entidad *Usuario*, la cual centraliza la gestión de todos los usuarios del sistema. Esta entidad se distingue por el atributo “rol”, que permite identificar si se trata de un Cliente, de un Prestador de servicio o de un Administrador.

La entidad *Users* se relaciona con la entidad *Reserva* en dos contextos: por un lado, un usuario con rol “Cliente” puede generar muchas reservas (relación de uno a muchos), y por otro, un usuario con rol “Prestador de servicio” atiende múltiples reservas a través del atributo *prestador\_id*, estableciendo también una relación de uno a muchos. La entidad *Reservas* contiene atributos propios como fecha, hora y estado (Pendiente o Cancelada) y se vincula con la entidad *Servicios* mediante una relación de uno a muchos, lo que permite que un mismo servicio pueda ser reservado en diversas ocasiones.

Por su parte, la entidad *Servicios*, que agrupa atributos tales como nombre, descripción y precio, se asocia opcionalmente con la entidad *Promociones* en una relación de uno a muchos, donde un servicio puede tener múltiples promociones, pero cada promoción está asociada únicamente a un solo servicio.

Finalmente, la entidad *Servicios* se relaciona con la entidad *Imagenes* mediante una cardinalidad de uno a muchos, permitiendo que cada servicio disponga de múltiples imágenes que lo representen visualmente (con atributos como url y descripción).

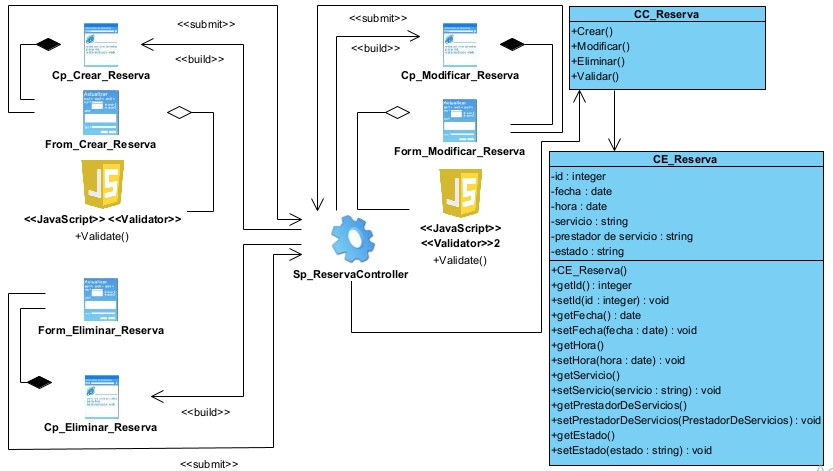
***Figura 4.*** *Modelo Entidad Relación.*

**

## 2.9 Diagrama de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en la aplicación para la gestión de reservas. Ilustra las relaciones entre los componentes del sistema y se utiliza durante el proceso de análisis y diseño, permitiendo una visión detallada de cómo se organiza la lógica de negocio y cómo interactúan los distintos elementos entre sí(Estrada [sin fecha]). A continuación, la *figura 5* representa el diagrama de clases del diseño para **gestionar reservas** en el salón de belleza.

***Figura 5.*** *Diagrama de clases de diseño gestionar reservas.*

****

**Paquete Vista** agrupa los componentes encargados de la interacción directa con el usuario. Las clases *Cp\_Crear\_Reserva, Cp\_Modificar\_Reserva y Cp\_Eliminar\_Reserva* representan las interfaces donde el usuario inicia acciones para gestionar reservas, mostrando formularios y recibiendo datos iniciales como fecha, servicio y prestador. Estas clases se complementan con los formularios *Form\_Crear\_Reserva, Form\_Modificar\_Reserva y Form\_Eliminar\_Reserva*, los cuales validan campos obligatorios y formatos mediante scripts JavaScript, asegurando que la información enviada al controlador sea correcta.

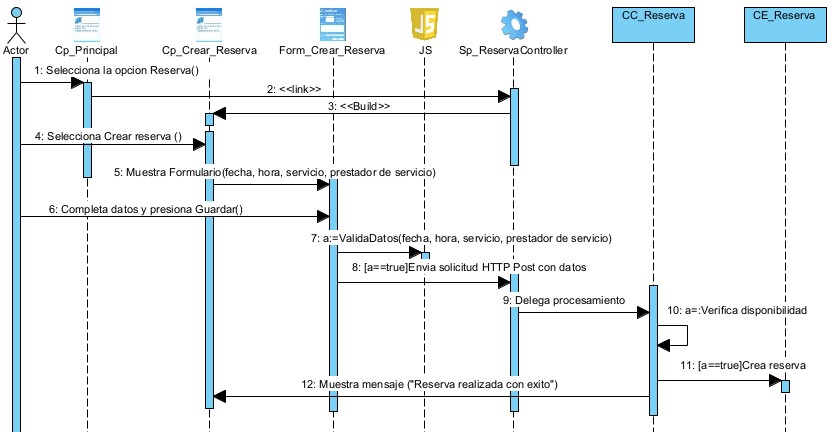
### Paquete Controlador centraliza la coordinación de operaciones de reserva a través de la clase *Sp\_ReservaController*. Este componente recibe solicitudes de la Vista, valida la información (como disponibilidad de cupos) y decide qué métodos del Modelo invocar. Además, gestiona el flujo de la aplicación: tras procesar una solicitud (crear, modificar o eliminar), notifica el resultado a la Vista, actuando como intermediario entre la lógica de negocio y la interacción del usuario.

**Paquete Modelo** contiene las clases que gestionan datos y reglas de negocio. La clase *CE\_Reserva* representa la entidad reserva, almacenando atributos como id, fecha y estado, y proporcionando métodos de acceso *(getters/setters).* Por otro lado, *CC\_Reserva* implementa la lógica de negocio con métodos como *crear(), modificar() y validar(),* aplicando reglas como disponibilidad de horarios o límites para modificaciones. El Controlador utiliza estas clases para garantizar que las operaciones cumplan las políticas del sistema.

## 2.10 Diagrama de secuencia

En contraste con los diagramas de clase y los diagramas de implementación, que representan la estructura estática de un sistema de software, el diagrama de secuencia muestra las comunicaciones dinámicas entre los objetos durante la ejecución de un proceso. Este tipo de diagrama expone el orden temporal de los mensajes que se intercambian para llevar a cabo una tarea específica, ofreciendo una visión clara de cómo interactúan los componentes en tiempo de ejecución(Solano et al. 2023).

***Figura 6.*** *Diagrama de secuencia para el caso de uso gestionar reserva escenario crear reserva.*

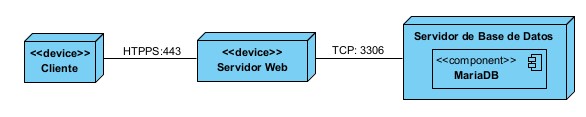
****

## 2.11 Diagrama de despliegue

Un modelo de despliegue consiste en una representación estructural de la arquitectura del sistema desde el punto de vista de la distribución de los artefactos de software en los nodos de ejecución; definiendo a dichos artefactos como representaciones de elementos concretos en el mundo físico que son el resultado de un proceso de desarrollo. En la *Figura 7* se muestra el diagrama de despliegue correspondiente a la aplicación web propuesta para el salón de belleza TereStyle.

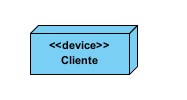
En este diagrama se observa cómo el cliente, mediante el protocolo HTTPS a través del puerto 443, se conecta al servidor web. A su vez, el servidor web, que contiene la lógica de la aplicación, accede a la base de datos en este caso implementada con MariaDB para el almacenamiento y recuperación de la información necesaria, tal como se detalla a continuación.

***Figura 7.*** *Diagrama de despliegue*



El dispositivo **cliente** corresponde a cualquier equipo (computadora, tableta o teléfono inteligente) que posea un navegador web con soporte para el protocolo HTTP/HTTPS. Desde este navegador, el usuario accede a la dirección o dominio donde está hospedado el sistema, pudiendo realizar las operaciones de reserva de citas, consulta de servicios, visualización de promociones, entre otras funcionalidades ofrecidas por la aplicación.

***Figura 8.*** *Dispositivo Cliente*

****

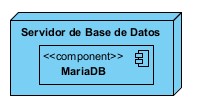
**Dispositivo Cliente:** requiere de un navegador web para interactuar con el sistema y consumir los servicios que brinda el servidor. Su conexión se realiza a través del protocolo **HTTPS**, usando el puerto **443**, lo que garantiza la seguridad en la comunicación de datos.

***Figura 9.*** *Nodo Servidor Web*

******

## Nodo Servidor Web: en este nodo *Figura 9* se instala la aplicación desde la cual se brindarán los servicios.

***Figura 10.*** *Nodo Base de Datos*

******

## No Nodo Servidor de Base de Datos: en este nodo (Figura 10) se almacenan todos los datos referentes al portal web para TereStyle.

## Conclusiones del capítulo

## El diseño de la solución propuesta logró integrar los fundamentos teóricos con las necesidades específicas de TereStyle, traduciendo las limitaciones identificadas en requisitos funcionales y no funcionales que aseguran un portal alineado con el objetivo de mejorar la visibilidad digital. La elaboración del modelo conceptual y los casos de uso permitió formalizar los procesos clave del salón, revelando la criticidad de automatizar la gestión de citas para reducir errores operativos, lo que representa un avance significativo respecto al estado actual. La adopción del patrón MVC demostró ser acertada al estructurar el sistema en capas independientes, facilitando la escalabilidad y el mantenimiento futuro, mientras que los diagramas UML (clases, secuencia, despliegue) aportaron precisión al diseño, garantizando coherencia entre las funcionalidades y la arquitectura técnica. Estos resultados consolidan un diseño robusto que no solo resuelve las deficiencias actuales, sino que posiciona a TereStyle para competir digitalmente en un mercado exigente.

.

## Capítulo III: Validación de la solución propuesta: Desarrollo de un portal web para el salón de belleza TereStyle.

En el presente capítulo se describe el proceso de implementación del portal web para el salón de belleza TereStyle, además, se realiza la validación del sistema a través una estrategia de pruebas.

## Introducción

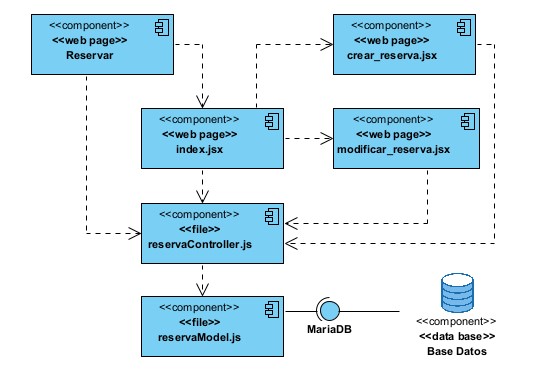
El objetivo de este flujo de trabajo es construir el sistema y su arquitectura a partir de los componentes y subsistemas definidos en las clases, objetos y subsistemas. Para ello, se establece la organización del código, se desarrollan y prueban los componentes, y se integran en un sistema operativo. El flujo de trabajo de prueba se encarga de comprobar que el sistema cumple con la funcionalidad especificada en los casos de uso y con los requisitos del sistema, validando el resultado de la implementación.

## 3.1 Diagrama de componentes

Los diagramas de componentes son una representación visual que detalla la estructura interna de un sistema software, mostrando sus componentes, las interfaces que los conectan y las dependencias entre ellos. Su objetivo principal es ilustrar la organización física de los componentes en el hardware, su distribución en nodos o procesos, y cómo se estructuran modularmente según el entorno de implementación y el lenguaje de programación utilizado. Además, estos diagramas destacan las interfaces críticas y las relaciones de dependencia, facilitando la comprensión de la arquitectura del sistema.

Para el portal web del salón de belleza TereStyle, se diseñó un diagrama de componentes alineado con el patrón arquitectónico Modelo-Vista-Controlador (MVC) y las tecnologías seleccionadas: React (frontend), Node.js/Express (backend) y MariaDB (base de datos). A continuación, se describe la estructura:

## *Figura X. Diagrama de Componente del Portal Web de TereStyle (Gestionar Reservas).*

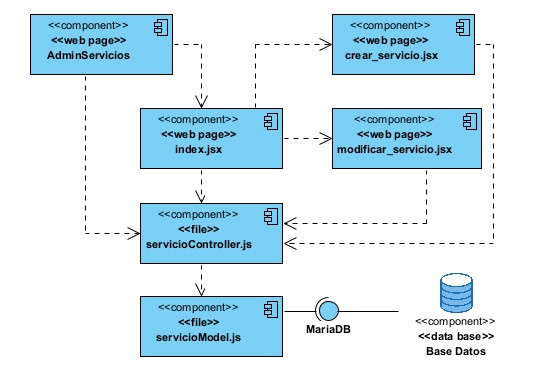


## 

## *Tabla X. Descripción de Diagrama de Componente (Gestionar Reservas).*

|  |  |  |
| --- | --- | --- |
|  | Componentes | Descripción |
| Modelo | reservaModel.js | Clase entidad que almacena las reservas en el sistema. |
| Vistas | Reservar | Este componente se encarga de redireccionar desde la página principal del sistema hacia la vista donde se gestionan las Reservas. |
| index.jsx | Este componente se encarga de mostrar un listado de las Reservas insertadas en la base de datos, así como mostrar las opciones necesarias para gestionar las Reservas |
| crear\_reserva.jsx | Este componente se encarga de registrar una nueva reserva en la base de datos con los datos necesarios de la misma. |
| modificar\_reserva.jsx | Este componente se encarga de modificar una reserva en la base de datos con los datos necesarios de la misma. |
| Controlador | reservaController.js | Este componente se encarga de realizar todos los métodos relacionados a la gestión de las Reservas en el sistema. |

## *Figura X. Diagrama de Componente del Portal Web de TereStyle (Gestionar Servicios).*



## *Tabla X. Descripción de Diagrama de Componente (Gestionar Servicios).*

|  |  |  |
| --- | --- | --- |
|  | Componentes | Descripción |
| Modelo | reservaModel.js | Clase entidad que almacena los servicios en el sistema. |
| Vistas | AdminServicios | Este componente se encarga de redireccionar desde la página principal del sistema hacia la vista donde se gestionan los Servicios. |
| index.jsx | Este componente se encarga de mostrar un listado de los Servicios insertadas en la base de datos, así como mostrar las opciones necesarias para gestionar los Servicios. |
| crear\_servicio.jsx | Este componente se encarga de registrar un nuevo servicio en la base de datos con los datos necesarios de la misma. |
| modificar\_servicio.jsx | Este componente se encarga de modificar un servicio en la base de datos con los datos necesarios de la misma. |
| Controlador | servicioController.js | Este componente se encarga de realizar todos los métodos relacionados a la gestión de los Servicios en el sistema. |

## 3.2 Estándares de codificación

Los estándares de codificación son un conjunto de pautas que, más allá de la lógica interna del programa, buscan uniformar la estructura y la apariencia del código fuente. Estas convenciones facilitan la lectura, comprensión y posterior mantenimiento del software, lo cual es esencial para un proyecto como el portal web de TereStyle. Un código consistente y bien estructurado no solo mejora la productividad del equipo de desarrollo, sino que también asegura que la aplicación pueda evolucionar de manera controlada y sin introducir errores o inconsistencias.  
Dado que el proyecto utiliza JavaScript como lenguaje principal tanto en el frontend (React) como en el backend (Node.js/Express), las siguientes pautas se han diseñado para ser aplicables a ambos entornos, promoviendo un estilo armonioso que refleje la colaboración y garantice un código de alta calidad.

#### 3.2.1 Nomenclatura de las Clases y Componentes

Para mantener coherencia en el nombramiento de clases y componentes a lo largo del proyecto, se adoptan las siguientes convenciones:

* **Nombres de clases y componentes:** Los nombres de clases y componentes deben comenzar con la primera letra en mayúscula y usar la notación PascalCasing para nombres compuestos, asegurando que el propósito sea claro con solo leer el nombre.  
  **Ejemplos:**
* En el backend: Reserva, Servicio.
* En el frontend (React): GestionarServicios, GestionarPromociones.
* **Clases Controladora (Backend):** Las clases que gestionan la lógica de negocio en el backend incluirán el sufijo "Controller" al final de su nombre.
  + **Ejemplo:** ReservaController.js
* **Clases de Modelos (Backend):** Representan las entidades del dominio y se nombrarán de forma descriptiva sin sufijos adicionales, reflejando su propósito en la lógica de negocio.
  + **Ejemplo**: ReservaModel.js
* **Componentes de React (Frontend):** Los componentes de React seguirán la convención de PascalCasing y se almacenarán en archivos con extensión “.jsx” para indicar que contienen JSX.
  + **Ejemplo:** Reservar.jsx

Estas reglas aseguran que los nombres sean intuitivos y consistentes, facilitando la identificación de responsabilidades en el patrón MVC y la distinción entre frontend y backend.

#### 3.2.2 Estándares para Comentarios

Los comentarios son esenciales para documentar el código y mejorar su legibilidad. Se establecen las siguientes pautas:

* Los comentarios deben ser oraciones completas.
* Si un comentario es una frase u oración, su primera palabra debe comenzar con mayúscula, a menos que sea un identificador que comience con minúscula.
* Si un comentario es corto, el punto final puede omitirse.
* Los comentarios de una línea para aclaraciones del código aparecerán seguidos de los caracteres // y deben ubicarse en la misma línea que se desea comentar.
* Los comentarios de varias líneas para organización del código o documentación aparecerán dentro de los caracteres /\*\* … \*/.

Estas convenciones son directamente aplicables tanto en el frontend como en el backend, aprovechando las características de JavaScript.

#### 3.2.3 Estándares para Espacios en Blanco en Expresiones y Sentencias

El uso adecuado de espacios en blanco mejora la legibilidad del código. Se definen las siguientes reglas:

* **Evitar utilizar espacios en blanco en las siguientes situaciones:**
* Inmediatamente dentro de paréntesis, corchetes y llaves: function(x), [1,2], {key:value}.
* Inmediatamente antes de una coma, punto y coma o dos puntos: let x=1,y=2; .
* Inmediatamente antes del paréntesis que inicia la lista de argumentos en la llamada a una función: sum(a,b).
* Rodear con exactamente un espacio los siguientes operadores binarios:
* Asignación: let x = 5;.
* Asignación de aumentación: x += 2; .
* Comparación: if (x == y), x < 10.
* Expresiones lógicas: x && y, !z .

#### 3.2.4 Otros Estándares

Se incluyen prácticas adicionales para mejorar la calidad y evitar errores comunes:

* **Arrays Multilínea:** Añadir una coma al final de cada elemento en arrays multilínea para facilitar la adición o eliminación de elementos sin errores de sintaxis.
* **Saltos de línea:** Añadir un salto de línea antes de una sentencia return, a menos que el return esté solo en un bloque de sentencias, y un salto después de cada llave de cierre de sentencia, excepto después de la llave de cierre de clase.

Estas prácticas aseguran que el código sea legible, mantenible y coherente en todas sus partes, facilitando el trabajo colaborativo y la evolución futura del portal web de TereStyle.

## 3.3 Validación de la solución (Prueba de software)

Las pruebas de software son un conjunto de actividades diseñadas para verificar el comportamiento de un programa frente a un grupo finito de casos de prueba, seleccionados cuidadosamente de un dominio de ejecuciones potencialmente infinito, comparándolo con el comportamiento esperado. Estas pruebas buscan identificar fallos en la implementación, evaluar la calidad y garantizar la usabilidad del sistema.

#### 3.3.1 Niveles de prueba

Los niveles de prueba representan etapas progresivas para verificar y validar el software, abarcando desde componentes individuales hasta el sistema completo. Para el portal TereStyle, se aplicaron cuatro niveles de prueba: unitarias, de integración, de sistema y de aceptación, adaptados al alcance del proyecto.

**Pruebas unitarias**:Se enfocan en los componentes más pequeños del software, verificando que cada uno funcione correctamente de manera aislada. Estas pruebas constituyen la primera fase de validación, asegurando que los flujos de control y datos estén correctamente implementados según las especificaciones de diseño.

**Pruebas de integración**: Evalúan la interacción entre diferentes componentes del sistema para garantizar que operen adecuadamente al combinarse. Este nivel busca detectar errores en las interfaces y asegurar que las especificaciones de diseño se cumplan al integrar los módulos, verificando la comunicación entre entidades internas y externas.

**Pruebas del sistema**: Se realizan cuando el software opera como un todo, integrando todos los componentes de hardware y software. Este nivel verifica que el sistema cumpla con los requisitos establecidos, evaluando la navegación, el ingreso de datos, el procesamiento y la recuperación de información en un entorno completo.

**Pruebas de aceptación**: Representan la validación final antes del despliegue, con el objetivo de confirmar que el software está listo para ser utilizado por los usuarios finales, ejecutando las funciones y tareas para las que fue diseñado. Estas pruebas se dividen en dos subniveles:

* **Pruebas alfa**: Se llevan a cabo en un entorno controlado, con la participación del equipo de desarrollo y usuarios representativos, para identificar y corregir problemas en presencia de los desarrolladores.
* **Pruebas beta**: Se ejecutan por usuarios finales en condiciones reales, sin intervención directa del equipo de desarrollo, recopilando retroalimentación sobre el uso del sistema en su entorno operativo.

**Métodos de prueba**

**Pruebas de caja blanca (estructurales)**: Este método utiliza el conocimiento de la estructura interna del software para derivar casos de prueba que cubran los caminos lógicos del código. Se enfocan en verificar las condiciones, bucles y flujos de control, basándose en las especificaciones internas de diseño o el código fuente.

**Pruebas de caja negra (funcionales)**: Este enfoque se centra en los requisitos funcionales del software, sin considerar su estructura interna. Los casos de prueba se diseñan para ejercitar las funcionalidades especificadas, detectando errores en interfaces, rendimiento, inicialización, terminación o acceso a datos externos. Dentro de este método, se aplicó la técnica de **Partición Equivalente**, que divide el dominio de entrada en clases de datos para generar casos representativos que cubran valores válidos e inválidos.

#### 3.3.2 Estrategias de prueba

Una estrategia de pruebas de software proporciona una guía que describe los pasos que deben realizarse como parte de la prueba, cuándo se planean y se llevan a cabo dichos pasos, y cuánto esfuerzo, tiempo y recursos requieren. Por tanto, cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de la prueba y la recolección y evaluación de los resultados.

Para validar el portal web desarrollado para el salón de belleza TereStyle, se propone una estrategia de pruebas que contempla la realización de pruebas funcionales, de usabilidad, de seguridad y de aceptación. Esta estrategia asegura que el sistema cumpla con los requisitos establecidos, garantizando su calidad y funcionalidad.

A continuación, se presenta la **Tabla X. Validación de la solución**, que detalla los tipos de prueba, los métodos o técnicas utilizados y los objetivos de validación específicos para el portal web de TereStyle.

***Tabla 6.*** *Validación de la solución.*

|  |  |  |
| --- | --- | --- |
| **Tipo de Prueba** | **Método (técnica) de prueba** | **Validación** |
| **Pruebas de integración** | Integración de módulos GitHup | Consiste en la comprobación de que los elementos del software que interactúan entre sí, funcionen de manera correcta y sin inconsistencias en el flujo de datos. |
| **Pruebas de sistema (Funcionales)** | Casos de prueba (Caja Negra) | Valida las funcionalidades diseñadas para el portal. |
| **Pruebas de sistema (Usabilidad)** | Pruebas con usuarios representativos | Valida la facilidad de uso, la intuitividad de la interfaz y la experiencia del usuario. |
| **Prueba de seguridad** | Software Acunetix v14.7 | Valida la confidencialidad, integridad y disponibilidad de los datos en el portal, identificando y mitigando vulnerabilidades |
| **Rendimiento, accesibilidad, mejores prácticas y SEO.** | Herramienta Lighthouse | Se generan informes detallados que identifican áreas de mejora en estos ámbitos |
| **Pruebas de aceptación** | Evaluar el cumplimiento de los requisitos pactados con el cliente | Verifica que el software está listo y puede ser usado por usuarios finales |

#### 3.3.3 Prueba de Integración

La prueba de integración es un tipo de prueba funcional ampliamente utilizada que se llevó a cabo de manera automatizada para verificar el correcto funcionamiento de los módulos individuales del sistema al ser integrados. En este proyecto, los módulos principales incluyen el backend (desarrollado en Node.js con Express y conectado a una base de datos MariaDB) y el frontend (desarrollado en React). Las pruebas de integración se enfocaron principalmente en validar la comunicación efectiva entre estos componentes y asegurar que los datos fluyeran correctamente a través de la API.

Para la gestión del proyecto, se utilizó GitHub como plataforma de alojamiento, aprovechando el sistema de control de versiones Git para organizar el código fuente. El entorno de desarrollo principal fue Visual Studio Code, desde donde se ejecutaron los comandos necesarios para clonar, modificar y actualizar el repositorio.

Procedimiento para la integración:

* El proyecto se clonó desde GitHub utilizando el comando “git clone https://github.com/traviesodev/1-PW-para-Teretyle.git” en la terminal de Visual Studio Code.
* Se creó una rama específica llamada pruebas-integracion para trabajar en las pruebas de manera aislada con el comando “git checkout -b pruebas-integracion”
* Se configuraron las credenciales de acceso a GitHub en el entorno local para autenticar las operaciones de Git.
* Las pruebas automatizadas del backend se ejecutaron con Jest y Supertest mediante el comando “set NODE\_ENV=test && npx jest”, verificando la integración entre la API y la base de datos.
* . Los cambios realizados se subieron al repositorio con commits específicos, como git commit -m "Agregar pruebas de integración para backend" y git push origin pruebas-integracion.

En la primera iteración se identificaron varias no conformidades relacionadas con la integración del backend y la base de dato, e**rror de conexión a la base de datos** donde el pool de conexiones no podía acceder a mydb\_test (timeout de 10 segundos), se solucionó creando manualmente mydb\_test en HeidiSQL y ajustando db.js para inicializar tablas automáticamente, un error impidió ejecutar las pruebas, este se corrigió exportando la instancia de Express en server.js sin iniciar el servidor en modo prueba y las tablas no se creaban en mydb\_test debido a una condición en db.js. Se resolvió eliminando la restricción condicional y corrigiendo el orden de la función initializeDatabase.

***Figura X.*** *Resumen de la prueba de integración*

Tras resolver estas no conformidades, la prueba de integración se ejecutó con éxito, confirmando que el backend podía conectarse a la base de datos, insertar datos y responder adecuadamente a las solicitudes HTTP. Este proceso aseguró que los componentes del backend, incluyendo el servidor Express y la base de datos, estuvieran completamente integrados y operaran sin inconvenientes, sentando una base sólida para la funcionalidad del sistema.

#### 3.3.3 Prueba de Sistema

Las pruebas de sistema constituyen un conjunto de actividades diseñadas para evaluar el comportamiento integral del sistema informático desarrollado. Su propósito principal es verificar que el sistema cumpla con los requisitos establecidos, tanto funcionales como no funcionales. Estas pruebas se llevan a cabo después de las pruebas de integración y antes de las pruebas de aceptación, con el objetivo de garantizar que todos los componentes del sistema hayan sido correctamente integrados y que ejecuten las funciones esperadas de manera adecuada. Las pruebas pueden realizarse de forma manual o automatizada, dependiendo de las necesidades y recursos disponibles.

En este caso, las pruebas de sistema se enfocaron en evaluar aspectos funcionales del software, asegurando que las funcionalidades implementadas respondan a lo especificado en los requisitos. A continuación, se describen las pruebas realizadas, los casos diseñados y los resultados obtenidos.

##### **3.3.4.1 Pruebas Funcionales (Caja Negra)**

Las pruebas funcionales se diseñaron tomando como base las especificaciones funcionales del sistema, con el objetivo de comprobar si el software cumple con las funciones esperadas por los usuarios. Estas pruebas, conocidas como de "caja negra", se aplicaron a la interfaz del sistema, analizando su comportamiento sin considerar la estructura lógica interna del código. El enfoque se centró en detectar errores relacionados con:

* Funciones incorrectas o ausentes.
* Errores en la interfaz de usuario.
* Fallos en el manejo de estructuras de datos o accesos a bases de datos externas.
* Problemas de comportamiento o rendimiento.

Los casos de prueba se diseñaron para validar que:

* Las funciones del software sean operativas.
* Las entradas sean procesadas correctamente.
* Las salidas generadas(mensajes de confirmación, actualizaciones en pantalla) sean las esperadas.
* La integridad de la información externa se mantenga intacta.

Para el diseño de estos casos, se empleó la **Técnica de Partición de Equivalencia**, que segmenta el dominio de entrada en clases de datos equivalentes, permitiendo generar casos de prueba representativos que cubran tanto valores válidos como inválidos. Esta técnica reduce el número total de casos necesarios al enfocarse en detectar clases específicas de errores, optimizando el proceso de prueba.

A continuación, se presenta un ejemplo del diseño de un caso de prueba para el requisito funcional RF "Adicionar Reserva":

***Tabla X****. Descripción de las variables del caso de prueba 1 - Adicionar Reserva*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nombre de campo** | **Clasificación** | **Valor Nulo** | **Descripción** |
| 1 | Fecha | Campo de fecha | No | Se debe ingresar una fecha válida (formato AAAA-MM-DD). |
| 2 | Hora | Campo de selección | No | Se debe ingresar una hora válida (formato HH:MM). |
| 3 | Servicio | Campo de selección | No | Se debe seleccionar un servicio del catálogo disponible. |
| 4 | Prestador de servicio | Campo de selección | No | Se debe seleccionar un profesional registrado. |

Con el propósito de evaluar la calidad del portal web y garantizar que las funcionalidades implementadas cumplan con los requisitos establecidos, se elaboraron casos de prueba basados en la técnica de caja negra. A continuación, se muestra un caso de prueba específico para el escenario "Adicionar Reserva":

***Tabla X.*** *Caso de prueba RF - Adicionar Reserva*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Escenario** | **Descripción** | **Fecha** | **Hora** | **Servicio** | **Prestador** | **Respuesta del sistema** | **Resultado Esperado** |
| EC 1 - Todos los datos válidos | Interfaz con el formulario para llenar los datos de la reserva, si todos son  correctos, se  adiciona la  reserva | V | V | V | V | Agrega los datos Muestra un  mensaje de  notificación.  color verde  “Reserva creada exitosamente” | Muestra mensaje: "Reserva realizada con éxito" |
| (2025-04-01) | (10:00) | (Corte de cabello) | (Prestador 1) |
| EC 2 - Fecha vacía | Interfaz con el formulario para llenar los datos de la reserva, si existe  algún campo vacío,  se muestra un mensaje pidiendo llenar el campo. | I | V | V | V | Comprueba  si la fecha esta  vacía, sí lo está,  muestra un mensaje  bajo el campo en color rojo  “La fecha es requerida “ | Muestra mensaje: "La fecha es requerida" |
| (vacío) | (10:00) | (Corte de cabello) | (Estilista 1) |
| EC 3 - Servicio no seleccionado | Interfaz con el formulario para llenar los datos de la reserva, si existe  algún campo vacío,  se muestra un mensaje pidiendo llenar el campo. | V | V | I | V | Comprueba  si se seleccionó el servicio, si no lo está, muestra un mensaje bajo el campo en color rojo “Debe seleccionar un servicio“ | Muestra mensaje: "Debe seleccionar un servicio" |
| (2025-04-01) | (10:00) | (vacío) | (Estilista 1) |

Las pruebas se ejecutaron en múltiples iteraciones hasta eliminar todas las no conformidades detectadas. En cada iteración, se corrigieron los errores identificados para garantizar que todo sea completamente funcional. A continuación, se presentan los resultados obtenidos tras dos iteraciones de pruebas funcionales:

Descripción de las no conformidades y soluciones:

1-Campos de texto aceptaban caracteres no válidos: Por ejemplo, el campo "Descuento" permitía ingresar caracteres como "#@!", lo que fue corregido restringiendo la entrada a formatos numéricos válidos mediante validaciones en el frontend y backend.

**2-Campos obligatorios no validados:** En la primera iteración, el sistema permitía guardar reservas sin seleccionar un servicio, lo que se solucionó añadiendo validaciones en el formulario correspondiente.

**3-Mensajes de error imprecisos:** Algunos mensajes no describían adecuadamente el problema ("Error desconocido" en lugar de "Este campo es obligatorio"). Esto se corrigió ajustando las respuestas del servidor en Express para reflejar el error específico.

**4-Errores en la integración con la base de datos:** En un caso, la reserva no se almacenaba correctamente debido a un error en la consulta SQL, solucionado ajustando el script para asegurar la correcta inserción en MariaDB.

**En la segunda iteración, se resolvió la no conformidad pendiente:**

5-El campo "Fecha" se guardaba en formatos inválidos; se implementó una validación estricta en el frontend y backend para aceptar solo fechas válidas según el estándar ISO (AAAA-MM-DD).

La criticidad de los errores se clasificó en cuatro niveles: Alto, Medio, Bajo e Informativo. En la primera iteración, se detectaron 2 errores de nivel alto (integridad de datos y campos obligatorios), 2 de nivel medio (validación de formatos y mensajes), y 1 de nivel bajo (caracteres no válidos en campos secundarios). Todos fueron solucionados manualmente y validados en la segunda iteración.

**3.3.5 Pruebas de sistema (Usabilidad)**

La prueba de sistema aplicada a la solución implementada fue la Prueba de Usabilidad, un método esencial para evaluar la experiencia del usuario al interactuar con un producto o servicio digital. Esta prueba permite identificar patrones de comportamiento y percepción de los usuarios al realizar tareas específicas, ayudando a mejorar la funcionalidad y la satisfacción del usuario.

Se aplicó una prueba de usabilidad cualitativa, centrada en recopilar información sobre el uso del producto desde la perspectiva del usuario. Para llevar a cabo esta prueba, se utilizó el método de prueba de usabilidad en laboratorio, en el cual un moderador guía a los participantes mientras realizan tareas específicas. El moderador también escucha preguntas y comentarios en tiempo real, facilitando ajustes inmediatos si fuera necesario.

La ISO/IEC 25000 define un conjunto de subcaracterísticas de usabilidad que fueron consideradas durante la evaluación. Estas subcaracterísticas son:

1. Reconocimiento apropiado: Capacidad del producto para permitir al usuario comprender rápidamente si el software es adecuado para sus necesidades.
2. Aprendizaje: Facilidad con la que el usuario puede aprender a utilizar el sistema.
3. Operatividad: Capacidad del sistema para ser operado y controlado fácilmente.
4. Protección contra errores de usuario: Capacidad del sistema para prevenir y manejar errores cometidos por los usuarios.
5. Estética de la interfaz de usuario: Atractivo visual y capacidad de la interfaz para generar satisfacción en el usuario.
6. Accesibilidad: Capacidad del sistema para ser usado por personas con diferentes características y discapacidades.

Resultados de la Prueba de Usabilidad

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Categoría** | **Total de indicadores evaluados** | **Resultados positivos** | **Resultados negativos** | **No Procede** |
| **Visibilidad del sistema** | 17 | 14 | 2 | 1 |
| **Lenguaje común entre sistema y usuario** | 11 | 9 | 0 | 2 |
| **Libertad y control por parte del usuario** | 29 | 21 | 1 | 7 |
| **Consistencia y estándares** | 33 | 21 | 0 | 12 |
| **Estética y diseño minimalista** | 18 | 17 | 0 | 1 |
| **Prevención de errors** | 8 | 7 | 1 | 0 |
| **Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores** | 11 | 1 | 0 | 10 |
| **Ayuda y documentación** | 11 | 0 | 1 | 9 |
| **Flexibilidad y eficiencia de uso** | 6 | 4 | 0 | 2 |

**Negativos identificados**

* Falta de retroalimentación visual durante retrasos.
  + **Solución**: Implementar un indicador de carga (espinner o mensaje) para acciones que tarden en procesarse.
* No hay "breadcrumbs" para orientar al usuario.
  + **Solución**: Añadir una barra de navegación de tipo "migas de pan" en todas las páginas internas.
* No hay opción para ajustar el tamaño del texto.
  + **Solución**: Implementar controles de accesibilidad (botones +/-) o asegurar que el sitio sea responsivo a los ajustes del navegador.
* Uso de pop-ups para contenido crítico.
  + **Solución**: Integrar el contenido importante directamente en la página principal o usar modales no intrusivos.

**3.3.6 Rendimiento, accesibilidad, mejores prácticas y SEO.**

Las pruebas de rendimiento, accesibilidad, buenas prácticas y SEO son fundamentales para garantizar que la aplicación web ofrezca una experiencia óptima a todos los usuarios, independientemente de su dispositivo o condiciones de red. Estas evaluaciones se centran en medir la velocidad de carga, la eficiencia en la utilización de recursos, la facilidad de uso para personas con diversas capacidades y la adherencia a estándares recomendados por los motores de búsqueda.

Para la realización de estas pruebas se empleó la herramienta Lighthouse, la cual analiza de forma automatizada la calidad de las páginas web en términos de rendimiento, accesibilidad, buenas prácticas y optimización para motores de búsqueda. Lighthouse proporciona un reporte detallado que identifica áreas de mejora, asignando puntuaciones en cada categoría para facilitar la priorización de ajustes técnicos.

Durante la primera iteración, se obtuvieron los siguientes resultados:  
***Tabla 12.*** *Resumen de la prueba con Lighthouse.*

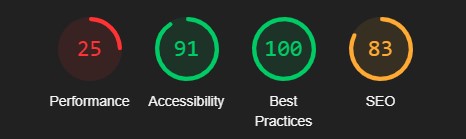
|  |  |  |
| --- | --- | --- |
| **Categoría** | **Mobile** | **Desktop** |
| **Rendimiento** | 25 | 25 |
| **Accesibilidad** | 91 | 100 |
| **Mejores Prácticas** | 100 | 100 |
| **SEO** | 83 | 83 |

Los resultados obtenidos reflejan el desempeño del portal web en cada una de las categorías evaluadas:

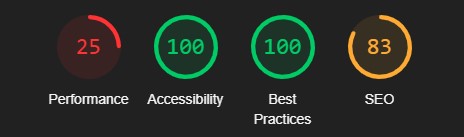
* **Mejores Prácticas**: El portal obtuvo una puntuación perfecta de 100 en ambas plataformas, lo que indica que cumple plenamente con las recomendaciones actuales de desarrollo web, como el uso de HTTPS, la ausencia de bibliotecas vulnerables y otras prácticas de seguridad y eficiencia.
* **Accesibilidad**: Con una puntuación de 91 en mobile y 100 en desktop, el sitio demuestra ser altamente accesible para usuarios con discapacidades. La diferencia entre plataformas sugiere que, aunque el portal es funcional en términos de contraste de color y navegabilidad con teclado, existen pequeñas oportunidades de mejora en la versión móvil para alcanzar el estándar perfecto logrado en desktop.
* **SEO**: La puntuación de 83 en ambas versiones indica un buen nivel de optimización para motores de búsqueda. Sin embargo, este resultado señala que hay aspectos como metadatos, estructura de encabezados o enlaces descriptivos que podrían ajustarse para mejorar aún más la visibilidad en buscadores.
* **Rendimiento**: Con una puntuación de 25 tanto en mobile como en desktop, esta categoría destaca como el área más crítica. Este bajo desempeño refleja problemas significativos en los tiempos de carga y la interactividad del sitio, lo que podría impactar negativamente la experiencia del usuario, especialmente en dispositivos móviles donde la rapidez es clave.

A continuación, en las siguientes figuras se pueden observar todos los resultados antes descritos:

***Figura X.*** *Resumen de la prueba con la herramienta Lighthouse para mobile.*



***Figura X.*** *Resumen de la prueba con la herramienta Lighthouse para desktop*

*.*

**3.3.7 Prueba de Aceptación**

Las pruebas de aceptación son un componente clave en el desarrollo del portal web para el salón de belleza TereStyle, ya que representan la validación final de la satisfacción del cliente con el producto desarrollado. Estas pruebas marcan el cierre de una iteración y el inicio de la siguiente, asegurando que el sistema cumpla con las expectativas del usuario final, en este caso, el personal y los clientes del salón TereStyle. Se diseñaron y ejecutaron en paralelo con el desarrollo del sistema, adaptándose a los cambios surgidos durante el proceso de implementación.

Para la realización de las pruebas de aceptación, se definieron los siguientes elementos, siguiendo la estructura establecida en el ejemplo proporcionado:

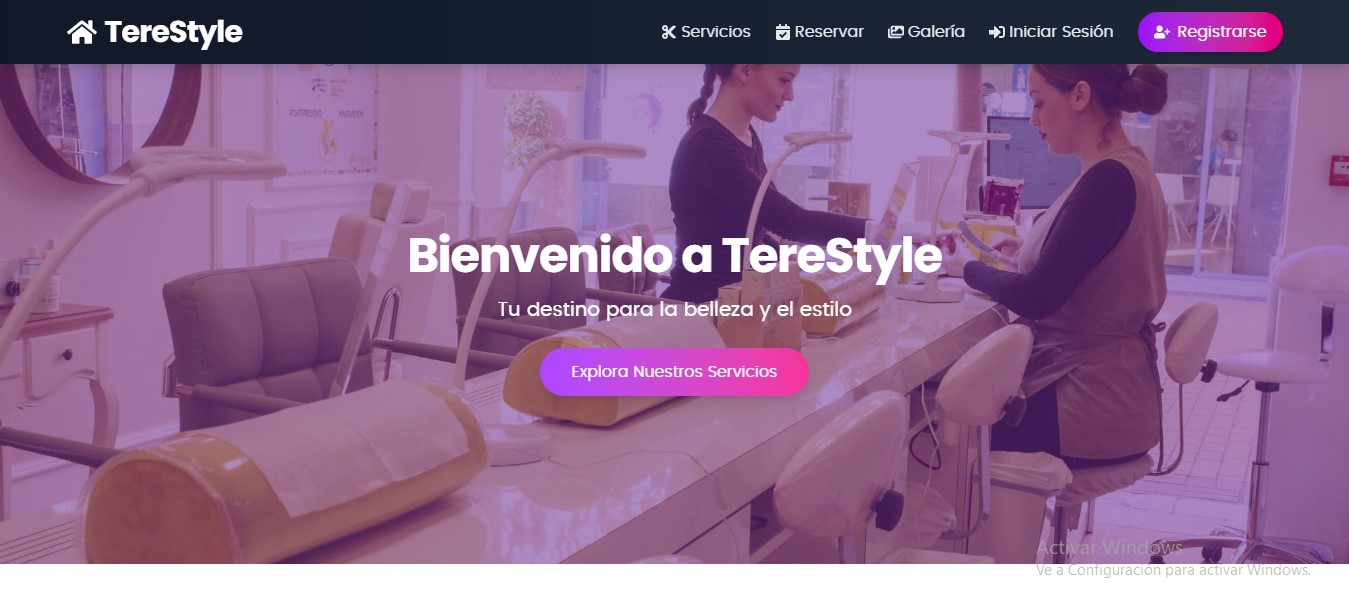
* **CU:** Número de la CUS (Caso de Uso del Sistema) a la cual pertenece la prueba.
* **Nombre:** Identificador único del caso de prueba, compuesto por el código y un nombre descriptivo.
* **Descripción:** Acción específica que el sistema debe realizar.
* **Condiciones de ejecución:** Características y elementos necesarios en el sistema para llevar a cabo la prueba.
* **Entrada/Pasos de Ejecución:** Datos de entrada requeridos y pasos detallados para ejecutar el caso de prueba.
* **Resultados Esperados:** Respuesta esperada del sistema tras la ejecución de la prueba.
* **Evaluación de la prueba:** Clasificación del resultado como "satisfactoria" o "insatisfactoria".

## 3.4 Validación de los resultados de la investigación.

## 3.5 Interfaces principales.

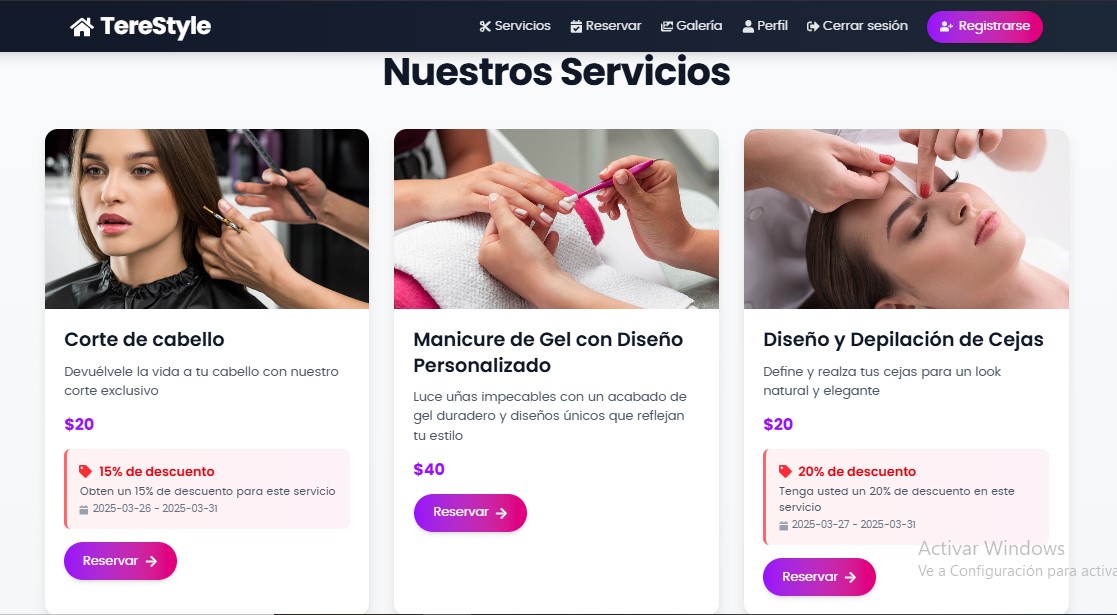
Una vez desarrollado el portal web para el salón de belleza TereStyle, es posible visualizar sus pantallas principales, donde se observa el resultado obtenido durante la implementación de los requisitos funcionales descritos en el capítulo 2.

***Figura X.*** *Captura de pantalla de la página inicio.*

**

En la figura se muestra la pantalla inicio del portal web para TereStyle, donde se tiene acceso a distintas funcionalidades.

***Figura X.*** *Captura de pantalla de la página Servicios.*



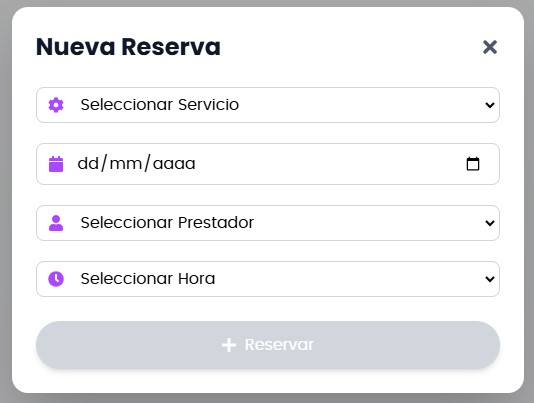
En la figura se muestra la pantalla de servicios del portal web para TereStyle, donde se pueden visualizar los servicios actuales que posee el salón con todos sus detalles y se tiene acceso a reservar si así lo desea.

***Figura X.*** *Captura de pantalla de la página Reservar.*

## 

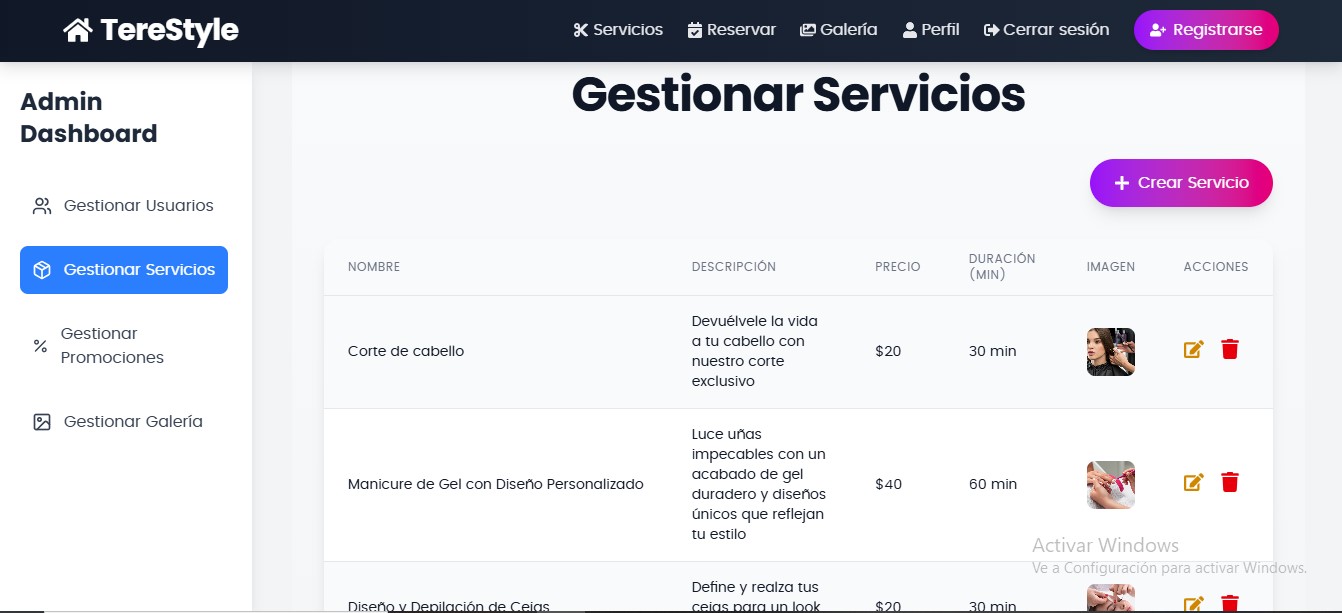
En la figura se muestra la pantalla de reservar del portal web para TereStyle, donde se puede visualizar las reservas creadas y la opción de crear una nueva reserva.

***Figura X.*** *Captura de pantalla del formulario para reservar.*



## En la figura se muestra el formulario correspondiente para realizar una reserva en el salón.

***Figura X.*** *Captura de pantalla de Gestionar Servicios*

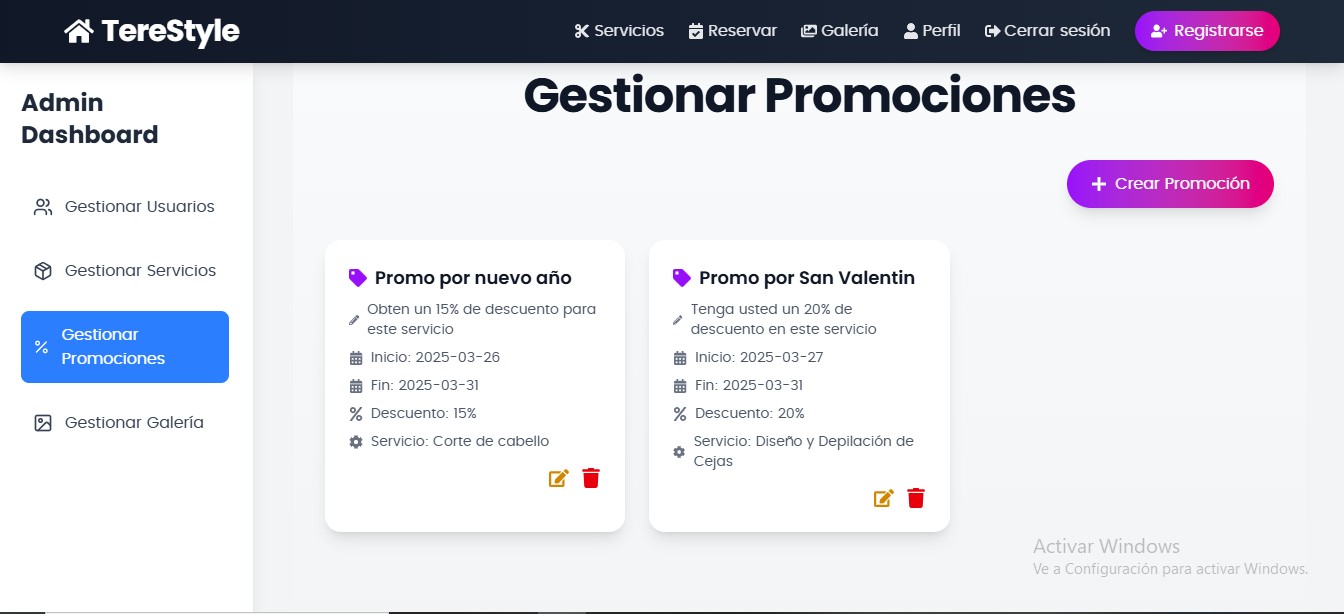


En la figura se muestra el listado de los servicios actuales del portal y diferentes opciones que se pueden realizar.

***Figura X.*** *Captura de pantalla de formulario para crear un servicio*

**

***Figura X.*** *Captura de pantalla de Gestionar Promociones*

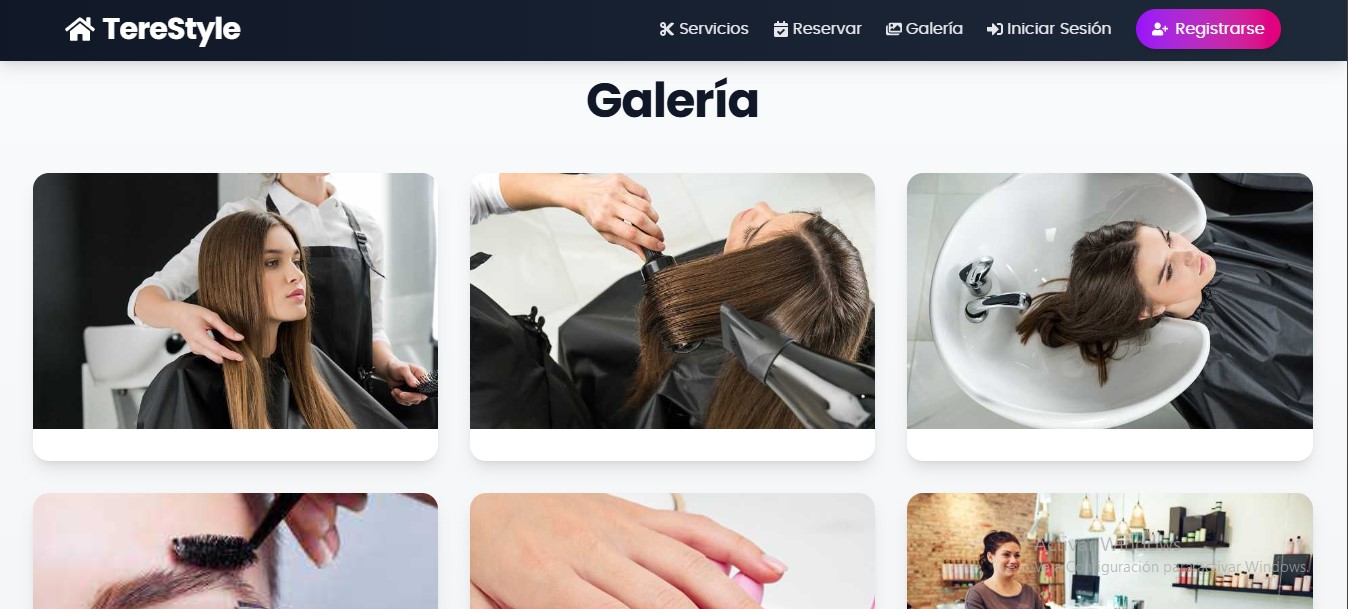


En la figura se muestra la pantalla para la gestión de promociones donde se visualiza las promociones creadas y diferentes opciones.

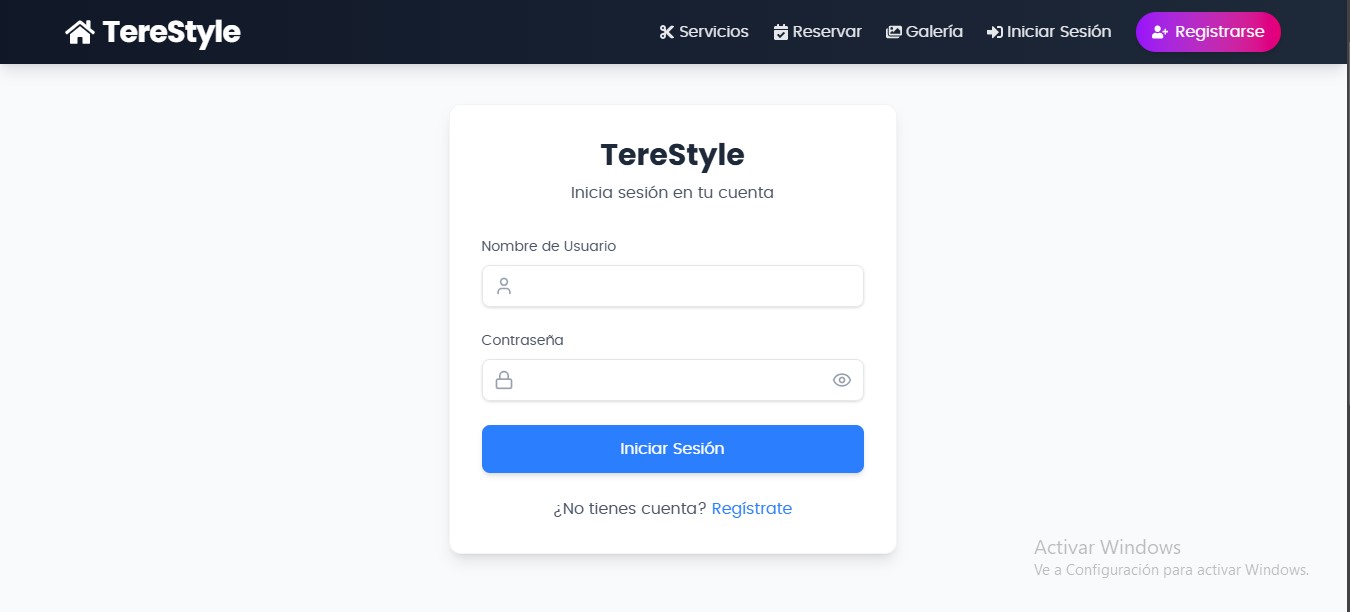
***Figura X.*** *Captura de pantalla de formulario para crear las promociones*



***Figura X.*** *Captura de pantalla de la página Galería*

**

## En la figura se muestra la pantalla de Galería donde se muestran las imágenes de trabajos realizados por parte del salón.

***Figura X.*** *Captura de pantalla de Iniciar Sesión.*

## En la figura se muestra el formulario para iniciar sesión en el portal

## Conclusiones del capítulo

<La lista de conclusiones en este capítulo por lo general van dirigidas a establecer los argumentos y resultados que demuestran la veracidad, factibilidad y fiabilidad de la solución propuesta en términos de los datos obtenidos al aplicar técnicas y métodos de verificación y validación de software, técnicas y métodos de validación científica de la transformación o impacto sobre el objeto de estudio; así como la factibilidad económica de la solución propuesta>

<EJEMPLO DE ESTRUCTURA Y CONTENIDO DE ESTE CAPÍTULO:

Título del Capítulo: Validación de SoftCalTest como aplicación web para la evaluación de la calidad de software en la empresa SOFT-PLUS+

I.1 Verificación y validación de la aplicación web SoftCalTest

I.2 Impacto de la aplicación web SoftCalTest en la evaluación de la calidad de software en la empresa SOFT-PLUS+

I.3 Estudio de factibilidad del desarrollo de software de de la aplicación web SoftCalTest

Conclusiones del capítulo>

# Conclusiones

<Las conclusiones, por lo general van dirigidas a establecer los argumentos y resultados a los que se arribó en los siguientes aspectos: (1) sistematización del estado del arte referido al objeto de estudio y el campo de acción, (2) diagnóstico del estado actual del objeto de estudio, (3) principales aspectos del análisis, diseño e implementación de la solución, (4) principales resultados de la validación de la solución propuesta. Deben apoyarse en los resultados obtenidos y descritos en la memoria y no en datos que no aparezcan en este documento. No pueden exceder una cuartilla en su extensión>

# Recomendaciones

<Las recomendaciones, por lo general van dirigidas a establecer aquellos aspectos en los cuales la investigación puede continuar para su perfeccionamiento, mantenimiento o evolución en el tiempo. No deben constituir acciones no realizadas por omisión de etapas del proceso investigativo o ingenieril; ni ser demasiadas en número que cuestionen la completitud y pertinencia de la investigación realizada. No pueden exceder una cuartilla en su extensión>

# Referencias Bibliográficas

AGUILAR, M.V.G., 2023. Estrategias de marketing digital para la empresa Mozé centro de eventos. En: Accepted: 2025-01-29T17:44:08Z [en línea], [consulta: 24 febrero 2025]. Disponible en: https://repositorio.unach.mx/jspui/handle/123456789/4290.

AL-FEDAGHI, S., 2021. Conceptual Data Modeling: Entity-Relationship Models as Thinging Machines. En: arXiv:2109.14717 [cs], *International Journal of Computer Science and Network Security*, vol. 21, no. 9, DOI 10.22937/IJCSNS.2021.21.9.33.

ALONSO, G.R., [sin fecha]. RoomieFinder: aplicación iOS para la Búsqueda de Compañeros de Piso en el Entorno Estudiantil. ,

ALVEAR VACA, P.A., 2023. Arquitectura micro-frontend para optimizar el desarrollo de aplicaciones web tipo spa. [en línea], [consulta: 23 febrero 2025]. Disponible en: https://repositorio.uta.edu.ec/handle/123456789/39527.

ANTONIO, P.P., 2022. *Gestión de bases de datos*. S.l.: Ediciones Paraninfo, S.A. ISBN 978-84-13-66530-6.

AVENDAÑO PEÑA, A.S. y CARDONA ARCILA, T., 2023. *Estrategias de marketing digital utilizadas por influencers a través de las redes sociales para los restaurantes gourmet del barrio Laureles en la ciudad de Medellín en el periodo 2019-2022* [en línea]. bachelorThesis. S.l.: Escuela de Economía, Administración y Negocios. [consulta: 23 febrero 2025]. Disponible en: https://repository.upb.edu.co/handle/20.500.11912/11785.

CABEZAS SÁNCHEZ, E., 2023. Aplicación web para la gestión y venta de productos 3D. En: Accepted: 2023-07-14T20:57:44Z [en línea], [consulta: 23 febrero 2025]. Disponible en: https://openaccess.uoc.edu/handle/10609/148341.

CASSETTAI, D.L., [sin fecha]. Trabajo final de Grado. ,

CATALÁN CALABUIG, C., 2023. *Aplicación web para la gestión de los comerciales en una empresa* [en línea]. Proyecto/Trabajo fin de carrera/grado. S.l.: Universitat Politècnica de València. [consulta: 23 febrero 2025]. Disponible en: https://riunet.upv.es/handle/10251/199925.

CLAUS, P.E., 2021. *Klein: Proceso de desarrollo de software con trazabilidad de requerimientos, documentación mínima y aplicable con equipos pequeños y distribuidos* [en línea]. Tesis. S.l.: Universidad Nacional de La Plata. [consulta: 24 febrero 2025]. Disponible en: http://sedici.unlp.edu.ar/handle/10915/121741.

CYBERCLICK, [sin fecha]. Informe: 20 Tendencias de Marketing para el Sector Beauty en 2025. [en línea]. [consulta: 19 febrero 2025]. Disponible en: https://www.cyberclick.es/informe-tendencias-digitales-en-el-sector-beauty-para-2025.

Documentation for Visual Studio Code. [en línea], [sin fecha]. [consulta: 19 febrero 2025]. Disponible en: https://code.visualstudio.com/docs.

DOMECQ BABIE, D. y NIÑO BENITEZ, Y., 2021. Pruebas y herramientas para comprobar la accesibilidad de sistemas y páginas web. [en línea], [consulta: 19 febrero 2025]. DOI 10.5281/ZENODO.5545985. Disponible en: https://zenodo.org/record/5545985.

ESTRADA, A.D.L., [sin fecha]. Trabajo de Integración Curricular. ,

FERNÁNDEZ, M.M., 2023. Transformación digital en Cuba. ¿Dónde estamos? *Cubahora* [en línea]. [consulta: 19 febrero 2025]. Disponible en: https://www.cubahora.cu/ciencia-y-tecnologia/transformacion-digital-en-cuba-donde-estamosquestion.

FERNANDO, P.M., 2021. *Lenguajes de marcas y sistemas de gestión de información*. S.l.: Ediciones Paraninfo, S.A. ISBN 978-84-283-4059-5.

GARCÍA, B.C., 2025a. *Diseño de elementos software con tecnologías basadas en componentes. UF1289*. S.l.: Tutor Formación. ISBN 9791387566265.

GARCÍA, B.C., 2025b. *Diseño de elementos software con tecnologías basadas en componentes. UF1289*. S.l.: Tutor Formación. ISBN 9791387566265.

GARCÍA-PEÑALVO, F.J., GARCÍA-HOLGADO, A. y VÁZQUEZ-INGELMO, A., 2021. Requisitos no funcionales. [en línea], [consulta: 18 febrero 2025]. Disponible en: http://repositorio.grial.eu/handle/grial/2457.

GARCÍA-PEÑALVO, F.J., GARCÍA-HOLGADO, A. y VÁZQUEZ-INGELMO, A., 2024. Ingeniería de requisitos. [en línea], [consulta: 18 febrero 2025]. Disponible en: http://repositorio.grial.eu/handle/grial/3022.

Git - Documentation. [en línea], [sin fecha]. [consulta: 19 febrero 2025]. Disponible en: https://git-scm.com/doc.

GUERRA CHAVERRA, S., 2024. Rediseño de la arquitectura de un personalizador para ecommerce: una revisión del proceso de migración de tecnología e integración de patrones de diseño de software. Práctica empresarial. En: Accepted: 2024-11-19T19:44:49Z [en línea], [consulta: 21 febrero 2025]. Disponible en: https://bibliotecadigital.udea.edu.co/handle/10495/43608.

GUERRA, G., 2024. Tendencias de Consumo 2025: Cómo será el comportamiento de los consumidores - La Caja Company. [en línea]. [consulta: 19 febrero 2025]. Disponible en: https://lacaja.company/blog/tendencias-de-consumo-2025-como-sera-el-comportamiento-de-los-consumidores/.

GUTIERREZ ARANA, E.J. y LEÓN GUTIERREZ, J.B., 2023. Sistema de gestión de secretos para el acceso privilegiado a sistemas de información en organizaciones del sector financiero. En: Accepted: 2024-01-04T01:12:35Z, *Universidad Peruana de Ciencias Aplicadas (UPC)* [en línea], [consulta: 23 febrero 2025]. Disponible en: https://repositorioacademico.upc.edu.pe/handle/10757/670548.

HERNÁNDEZ SUÁREZ, S., 2022. *Módulo de Evaluación integrada del Sistema de gestión de trabajo educativo de la facultad 1* [en línea]. bachelorThesis. S.l.: Universidad de las Ciencias Informáticas. Facultad 1. [consulta: 24 febrero 2025]. Disponible en: https://repositorio.uci.cu/jspui/handle/123456789/10565.

ISO 25010. [en línea], [sin fecha]. [consulta: 23 febrero 2025]. Disponible en: https://iso25000.com/index.php/normas-iso-25000/iso-25010.

LÓPEZ, I.C., [sin fecha]. Análisis y estrategia SEO en U4IMPACT: un TFG con impacto. ,

MARTELL LEÓN, M., 2021a. *Herramienta para el diseño de casos de prueba funcionales aplicando técnicas de reutilización* [en línea]. bachelorThesis. S.l.: Universidad de las Ciencias Informáticas. Facultad 1. [consulta: 23 febrero 2025]. Disponible en: https://repositorio.uci.cu/jspui/handle/123456789/10450.

MARTELL LEÓN, M., 2021b. *Herramienta para el diseño de casos de prueba funcionales aplicando técnicas de reutilización* [en línea]. bachelorThesis. S.l.: Universidad de las Ciencias Informáticas. Facultad 1. [consulta: 23 febrero 2025]. Disponible en: https://repositorio.uci.cu/jspui/handle/123456789/10450.

MIRABAL SARRIA, Y. y TORRES PAEZ, C.C., 2021. Validación del modelo de gestión pública de la calidad de vida en Cuba. *Cooperativismo y Desarrollo*, vol. 9, no. 3, ISSN 2310-340X.

REINA, S.E.F., [sin fecha]. Aplicación móvil multiplataforma de gestión de torneos con asistente de respuesta inteligente integrado. ,

SÁNCHEZ, M.B.P., 2024. *Guía para las prácticas de Comunicación Digital e Internet*. S.l.: Universidad Miguel Hernández. ISBN 978-84-18177-70-5.

SOLANO, R., VÉLEZ, J., OJEDA, K. y HERRERA, A., 2023. Diseño de un mapa interactivo de contaminantes potenciales generados por empresas de la ciudad de Cartagena. *Revista Ing-Nova*, vol. 2, no. 2, ISSN 2805-9182. DOI 10.32997/rin-2023-4344.

SQLite Documentation. [en línea], [sin fecha]. [consulta: 19 febrero 2025]. Disponible en: https://www.sqlite.org/docs.html.

SUAREZ, P. y ANDREA, Y., 2024. Diseño de un sitio web educativo basado en una arquitectura modular y sostenible para la gestión académica en la institución educativa de llano grande sede principal del municipio de nuevo colon boyacá. En: Accepted: 2024-05-09T21:57:10Z [en línea], [consulta: 23 febrero 2025]. Disponible en: http://repository.unad.edu.co/handle/10596/60912.

TAMAYO ESPINOSA, L., SILEGA MARTÍNEZ, N., TAMAYO ESPINOSA, L. y SILEGA MARTÍNEZ, N., 2021. Gestión de la mantenibilidad desde etapas tempranas en el desarrollo de software. *Revista Cubana de Ciencias Informáticas*, vol. 15, no. 1, ISSN 2227-1899.

TRUJILLO SÁEZ, F. y ÁLVAREZ JIMÉNEZ, D., 2021. Transformación digital de la administración pública: ¿Qué competencias necesitan los empleados públicos? *Gestión y Análisis de Políticas Públicas*, ISSN 1989-8991, 1134-6035. DOI 10.24965/gapp.i27.10923.

TUNQUE-LIZANA, M., QUISPE-HUAMAN, L. y TUNQUE-DUEÑAS, T., 2023. *Sistemas de información y toma de decisiones en una organización de salud: un análisis integral* [en línea]. S.l.: Instituto Universitario de Innovación Ciencia y Tecnología Inudi Perú. [consulta: 19 febrero 2025]. ISBN 978-612-51-3008-2. Disponible en: https://editorial.inudi.edu.pe/index.php/editorialinudi/catalog/book/137.

UNIVERSIDAD ICESI, MARTÍNEZ ROMERO, A.C., MORENO BARRAGÁN, J.A., UNIVERSIDAD ICESI, PEREIRA LAVERDE, F., PONTIFICIA UNIVERSIDAD JAVERIANA CALI, OSORIO TINOCO, F.F., PONTIFICIA UNIVERSIDAD JAVERIANA CALI, LÓPEZ LAMBRAÑO, M., UNIVERSIDAD DEL NORTE, SCHMUTZLER, J., UNIVERSIDAD DEL NORTE, GÓMEZ NÚÑEZ, L., CORPORACIÓN UNIVERSITARIA AMERICANA, SANTIAGO MARTÍNEZ, V., UNIVERSIDAD AUTÓNOMA DEL CARIBE, PARRA BERNAL, L.D., UNIVERSIDAD EAN, OROZCO TRIANA, J.A., y UNIVERSIDAD EAN, 2023. *Actividad empresarial en Colombia 2022-2023: en camino hacia la reactivación* [en línea]. 1. S.l.: Universidad Icesi. [consulta: 18 febrero 2025]. ISBN 9786287630086. Disponible en: https://repository.icesi.edu.co/biblioteca\_digital/handle/10906/101355.

VELAZCO, A.A.E., [sin fecha]. Universidad de las Ciencias Informáticas. ,

VORONOV, A., 2022. *Migración a Javascript de funciones desarrolladas en Matlab para su ejecución sobre el cliente web* [en línea]. Proyecto/Trabajo fin de carrera/grado. S.l.: Universitat Politècnica de València. [consulta: 24 febrero 2025]. Disponible en: https://riunet.upv.es/handle/10251/187869.

WOOD, D., 2022. *Diseño de interfaces: Introducción a la comunicación visual en el diseño de interfaces de usuario*. S.l.: Parramón Paidotribo. ISBN 978-84-342-6247-8.

ZERMEÑO, E.V., 2024. Criterios de implementación de Tailwind CSS en desarrollos frontend. *Cuadernos Técnicos Universitarios de la DGTIC* [en línea], vol. 2, no. 3, [consulta: 19 febrero 2025]. DOI 10.22201/dgtic.ctud.2024.2.3.63. Disponible en: https://cuadernos.tic.unam.mx/index.php/cua/article/view/63.

# ANEXOS

<Contenido de los anexos con igual tipo de fuente Arial, pero a tamaño 11 puntos e interlineado 1.0 puntos. Debe tratar de sólo utilizarse aquellos anexos imprescindibles para complementar lo presentado en la memoria escrita y que no excedan las ocho (8) o diez (10 páginas). Deben aparecer uno a continuación del otro sin necesidad de saltos de página entre estos>