

Guía del Proyecto

-

HAIRBOOKER

REALIZADO POR:

SANTIAGO ROSADO RUIZ

ÍNDICE

1	INTRODUCCIÓN.....	4
1.1	PRESENTACIÓN DEL PROYECTO.....	4
1.2	OBJETIVOS DEL PROYECTO	4
1.3	JUSTIFICACIÓN DEL PROYECTO	5
2	ANÁLISIS DE REQUERIMIENTOS	5
2.1	IDENTIFICACIÓN DE NECESIDADES Y REQUERIMIENTOS	5
2.2	IDENTIFICACIÓN DE PÚBLICO	8
2.3	ESTUDIO DE MERCADO Y COMPETENCIA	8
2.3.1	<i>Tamaño del mercado</i>	8
2.3.2	<i>Competencia.....</i>	10
3	DISEÑO Y PLANIFICACIÓN	12
3.1	DEFINICIÓN DE LA ARQUITECTURA DEL PROYECTO.....	12
3.2	DISEÑO DE INTERFAZ DEL USUARIO.....	14
3.2.1	<i>Introducción.....</i>	14
3.2.2	<i>Elementos visuales.....</i>	14
3.3	PLANIFICACIÓN DE LAS TAREAS Y LOS RECURSOS NECESARIOS	18
3.3.1	<i>Introducción.....</i>	18
3.3.2	<i>Planificación de las tareas</i>	18
3.3.3.	<i>Recursos utilizados.....</i>	20
4	IMPLEMENTACIÓN Y PRUEBAS.....	21
4.1	DESARROLLO DE LAS FUNCIONALIDADES DEL PROYECTO	21
4.1.1	<i>Gestión de usuarios</i>	22
4.1.2	<i>Gestión de servicios</i>	25
4.1.3	<i>Gestión de citas.....</i>	28
4.1.4	<i>Otras funcionalidades.....</i>	34
4.2	PRUEBAS UNITARIAS Y DE INTEGRACIÓN	35
4.2.1	<i>Pruebas en lado servidor</i>	35
4.2.2	<i>Pruebas en lado cliente.....</i>	39
5	DOCUMENTACIÓN	49
5.1	DOCUMENTACIÓN TÉCNICA	49
5.1.1	<i>Descripción general</i>	49
5.1.2	<i>Estructura del proyecto.....</i>	50
5.1.3	<i>Dependencias y herramientas</i>	51
5.1.4	<i>Configuración</i>	51

5.1.5	<i>Base de Datos</i>	55
5.2	DOCUMENTACIÓN DE USUARIO	60
5.2.1	<i>Requisitos previos</i>	60
5.2.2	<i>Guía de usuario</i>	60
5.3	MANUAL DE INSTALACIÓN Y CONFIGURACIÓN.....	71
5.3.1	<i>Instalación y configuración en local</i>	74
6	MANTENIMIENTO Y EVOLUCIÓN	77
6.1	PLAN DE MANTENIMIENTO Y SOPORTE	77
6.2	ACTUALIZACIONES Y MEJoras FUTURAS	79
7	CONCLUSIONES	80
7.1	EVALUACIÓN DEL PROYECTO.....	80
7.1.1	<i>Fortalezas</i>	81
7.1.2	<i>Debilidades</i>	81
7.2	CUMPLIMIENTO DE OBJETIVOS Y REQUISITOS.....	81
7.3	LECCIONES APRENDIDAS Y RECOMENDACIONES PARA FUTUROS PROYECTOS	82
8	BIBLIOGRAFÍA Y REFERENCIAS	83

1 INTRODUCCIÓN

1.1 Presentación del proyecto

El nombre que recibe la aplicación web desarrollada es *Hairbooker*. Este nombre hace referencia al sector al que va dirigida, en este caso, peluquerías y barberías, por otro lado, “Booker” proviene del verbo en inglés book que significa reservar. De esta manera, se representa la finalidad de esta aplicación, que es ofrecer una herramienta eficiente para la gestión de citas y reservas en peluquerías de manera sencilla y accesible, tanto para los propios propietarios, como para los clientes.

1.2 Objetivos del proyecto

Los objetivos específicos que se quieren lograr con el desarrollo de este proyecto son los siguientes:

- Facilitar la gestión de las citas por parte de los propietarios y sus clientes.
- Mejorar la interactividad entre el propietario y el cliente.
- Integración de recordatorios.

Los usuarios que podrán acceder a la aplicación web son: los propietarios de los establecimientos y los clientes de estos. Los propietarios tendrán diferentes privilegios que los usuarios. Estos deben registrarse incluyendo sus datos personales y los datos fiscales de su establecimiento. Respecto a los clientes, tan solo deben introducir sus datos personales para poder acceder a la web.

La funcionalidad de la web variará en función del tipo de usuario que acceda a la plataforma. Si el usuario es propietario, tendrá acceso a varias herramientas clave para la gestión de su negocio. Entre ellas, podrá visualizar un calendario interactivo con todas las citas programadas, mover citas a diferentes días según la disponibilidad, gestionar los servicios que ofrece en su establecimiento, etc. Además, el propietario recibirá notificaciones automáticas cada vez que un cliente cancele una cita, lo que le permitirá reaccionar de manera oportuna y reorganizar su agenda de manera eficiente.

1.3 Justificación del proyecto

El objetivo general de este proyecto es simplificar y optimizar la gestión de citas para propietarios de pequeñas peluquerías y usuarios de peluquería. En la actualidad, muchos de estos establecimientos aún dependen de agendas en papel y de la comunicación por teléfono o WhatsApp para coordinar las citas. Este enfoque tradicional puede resultar ineficiente y propenso a errores. Por ello, *Hairbooker* se propone modernizar y agilizar este proceso mediante una solución digital que facilite la programación, modificación y cancelación de citas de manera más organizada y accesible. Con *Hairbooker*, buscamos mejorar la eficiencia operativa de estos establecimientos y ofrecer una experiencia más cómoda y fluida para sus clientes.

2 ANÁLISIS DE REQUERIMIENTOS

2.1 Identificación de necesidades y requerimientos

En la actualidad, un gran número de peluquerías o barberías todavía siguen utilizando una libreta, agenda, calendario, etc., para gestionar las citas de sus clientes. Este método conlleva a una mayor dedicación del tiempo por parte de los propietarios de estos establecimientos a la hora de organizar su jornada. La asignación de una cita puede verse afectada por un retraso, un adelanto o una cancelación de una cita. Este es un claro ejemplo de cómo, ante cualquier imprevisto, el peluquero o barbero debe notificar a todos sus clientes con citas ese mismo día para modificar sus horarios, ya sea mediante una llamada o vía WhatsApp, lo que complica significativamente la gestión de las citas.

Por ello, con el objeto de facilitar la gestión, se crea *Hairbooker*. Una agenda digital dedicada para este tipo de establecimientos que permite a los propietarios y a sus clientes organizar sus citas de manera óptima y orgánica, con una comunicación ininterrumpida por ambas partes. El cliente podrá modificar sus citas, y la aplicación avisará automáticamente al propietario. Asimismo, el cliente también recibirá una notificación si su cita ha sido modificada.

El propósito de este proyecto es garantizar que la comunicación entre el cliente y el propietario nunca se interrumpa, asegurando una interacción constante y fluida entre ambos.

Respecto a los requerimientos del proyecto, se van a clasificar en dos tipos: requisitos funcionales y no funcionales.

- **Requisitos funcionales:** los requisitos funcionales son aquellos que la aplicación debe tener para cumplir con las necesidades o expectativas del usuario (Consultado el 7 de octubre de 2024).
<https://www.pmoinformatica.com/2018/05/que-es-requerimiento-funcional.html>).

Registro y autenticación de usuarios:

- Los usuarios deben poder iniciar sesión utilizando su correo electrónico y su contraseña. Además, el sistema debe permitir la opción de restablecer contraseña en caso de que el usuario la olvide.

Gestión de usuarios.

- Los usuarios deben poder editar su perfil (nombre, contraseña, imagen de perfil, correo electrónico, etc.).

Creación y gestión de citas:

- La aplicación debe permitir al cliente darse de alta en una peluquería, seleccionar el servicio que desea recibir y la posibilidad de modificación de su cita, dar reseñas.
- La aplicación debe permitir al propietario dar de alta a su establecimiento, registrar los servicios que ofrece, posibilidad de modificación de citas de sus clientes, asignación de precios.
- La aplicación debe permitir a los propietarios ver el calendario completo del mes con sus respectivas citas y la funcionalidad integrada de poder gestionarlas desde el mismo.

Notificaciones:

- La aplicación tiene que enviar una notificación, ya sea para confirmar la cita o recordarle una cita cercana.
- Los propietarios deben recibir notificaciones cuando haya nuevas reservas, cancelaciones o modificaciones de citas.

Panel de administración:

- Los propietarios tendrán acceso a un panel de control donde puedan acceder al calendario, gestionar las citas, gestionar los servicios y modificar su perfil de usuario.
- Los clientes tendrán acceso a un panel de control donde puedan acceder al calendario de la peluquería donde se encuentran datos de alta, gestionar sus citas y modificar su perfil de usuario.
- **Requisitos no funcionales:** son las restricciones o los requisitos impuestos al sistema. Se ocupan de problemas como la escalabilidad, mantenibilidad y rendimiento.

Rendimiento:

- El tiempo de respuesta de la aplicación debe ser reducido.
- La aplicación debe ser capaz de operar con gran cantidad de usuarios.

Seguridad:

- Las contraseñas deben estar cifradas en nuestro base de datos.
- La aplicación debe controlar el acceso, tanto para clientes como para propietarios.

Usabilidad:

- El sistema debe proporcionar mensajes orientativos e informativos que ayuden al usuario.
- El sistema debe poseer interfaces gráficas bien formadas.
- La aplicación web tiene que tener un diseño *responsive* para que se adapte diferentes tipos de dispositivos.
- Los textos deben ser fáciles de leer.
- El sistema debe ser intuitivo y fácil de usar por parte del usuario.

2.2 Identificación de público

El público objetivo de nuestra aplicación se enfoca principalmente en los propietarios de peluquerías, barberías o salones de belleza, así como en sus clientes. La edad de los usuarios de estos establecimientos puede variar según el tipo de negocio (peluquerías unisex, barberías, peluquerías low-cost). No obstante, es importante destacar a la población joven y adulta, especialmente aquellos entre los 18 y 30 años y los 30 y 50 años, ya que son quienes manejan con mayor facilidad las tecnologías. El resto de población, sobre todo aquella superior a 60 años de edad, depende menos de la tecnología y suelen optar por métodos más tradicionales, ya sea vía llamada o acudiendo presencialmente al local.

2.3 Estudio de mercado y competencia

2.3.1 Tamaño del mercado

Para realizar el estudio del tamaño del mercado, en primer lugar, hablaremos del mercado al cual se dirige la aplicación, que en este caso se trata del sector de la estética y belleza. Por otro lado, se analizará el mercado de desarrollo de aplicaciones web.

En primer lugar, **la industria de la peluquería** ha experimentado un crecimiento constante, sobre todo después de la pandemia de COVID-19, que fue uno de los sectores donde este golpeo bastante fuerte. Actualmente, se estima que hay más de 50.000 salones de peluquería en España, generando un volumen de negocio que supera los 4 mil millones de euros anuales (Consultado el 7 de octubre de 2024. <https://www.interempresas.net/Peluqueria/Articulos/508772-El-negocio-de-las-peluquerias-un-sector-en-auge.html>). Por ello, como se detalla en el informe “*Radiografía del sector de la belleza y el cuidado personal en España*” (Square, 2024) es en este contexto donde la tecnología supone un importante aliado para ayudar a las PYMES del sector, mejorando su productividad y rentabilidad, creando mecanismo que les permitan elevar la experiencia de cliente y contribuyendo a su fidelización.

Este informe detalla los principales medios que los clientes utilizan para reservar sus citas: un 61,1% lo hace a través de WhatsApp, un 56% de manera presencial, un 54% por llamada telefónica, y solo un 38,4% mediante página web. Por ello, es fundamental ofrecer una aplicación web eficiente que permita duplicar este último porcentaje.

Ya analizado el mercado y la industria de la belleza, es importante investigar como es **el mercado del desarrollo web**. En 2023, el tamaño del mercado mundial de desarrollo web fue de 65.350 millones de dólares y se prevé que alcance los 130.900 millones de dólares en 2032. Este crecimiento se debe a varios factores: el aumento del uso de Internet, debido a su facilidad de acceso, y la llegada del *smartphone* (Business Research. Consultado el 8 de octubre de 2024. <https://www.businessresearchinsights.com/es/market-reports/web-development-market-109039>).

Según Business Research (2023), las últimas tendencias en el mundo del desarrollo web son las aplicaciones **PWA y SPA**. Las PWA se tratan de aplicaciones que se pueden acceder a través de un navegador web sin la necesidad de instalarlas desde una tienda de aplicaciones. Por otro lado, las aplicaciones SPA son aquellas

que cargan una única página HTML y se actualizan dinámicamente a medida que el usuario interactúa con la aplicación.

En conclusión, el sector de la estética y belleza en España ha mostrado una notable recuperación y crecimiento tras la pandemia, con más de 50.000 salones de peluquería generando un volumen de negocio que supera los 4.000 millones de euros anuales. Sin embargo, a pesar de este crecimiento, la digitalización en la gestión de citas todavía es limitada, ya que solo el 38,4% de los clientes utiliza páginas web para reservar sus servicios. Esto representa una gran oportunidad para mejorar la eficiencia del sector mediante el desarrollo de una aplicación web, que permita mejorar la gestión de citas online.

Además, el mercado de desarrollo web a nivel mundial se encuentra en plena expansión. Las tendencias emergentes como las PWA y SPA ofrecen soluciones más flexibles y eficientes, lo que sugiere que el desarrollo de una aplicación utilizando estas tecnologías podría posicionarse de manera competitiva en este mercado creciente. La combinación de un sector en expansión con la evolución tecnológica refuerza la importancia de implementar herramientas digitales que optimicen la productividad y mejoren la experiencia del cliente en el sector de la belleza.

2.3.2 Competencia

El importante crecimiento del mercado desarrollo web ha llevado a la creación de multitud de aplicaciones web, de ahí que existan varias que compitan directamente con *HairBooker*.

Algunas de estas aplicaciones son:

- *DunaSoft*: software de gestión de salones de belleza creada por la compañía con mismo nombre. Esta compañía ofrece varios tipos de software según la finalidad que se desea, desde la gestión de las citas,

venta online de productos y servicios, hasta la gestión de varios salones de belleza o franquiciados. Con un precio cada uno de 300 €.

- *Treatwell*: web de reservas de belleza online. Es una de las más grandes de Europa, por lo cual supone una importante competencia para *HairBooker*. Incorpora una serie de agentes que darán apoyo a los propietarios para aprovechar al máximo sus recursos y masificar sus beneficios.
- *Booksy*: app gratuita para la gestión de reservas dentro del sector de la belleza de manera gratuita.

¿Cómo *Hairbooker* se puede diferenciar de esta competencia tan fuerte? La respuesta es apostando por estrategias que lo diferencien de la competencia como las siguientes:

- *Hairbooker* se especializa en un solo nicho, en este caso peluquerías y barberías, ya que tanto *Treatwell* como *Booksy* se especializan en más servicios (estética, spa, etc.).
- Establecimiento de precios competitivos adoptando un modelo *freemium*, es decir, donde nuestros usuarios puedan acceder a la función básicas de manera gratuita y pagar por funcionalidades más avanzadas.
- Adopción de funcionalidades orientadas a pequeñas peluquerías, ofreciendo una interfaz sencilla y fácil de entender, con una fácil integración con las redes sociales.
- Integración de blog con noticias el sector y consejos para mejorar el uso de la plataforma.

3 DISEÑO Y PLANIFICACIÓN

3.1 Definición de la arquitectura del proyecto

La arquitectura adoptada para este proyecto sigue el patrón MVC (Modelo - Vista - Controlador). Este diseño organiza el desarrollo en tres capas bien definidas, lo que facilita la separación de la lógica de negocio de la capa de presentación destinada al cliente.

Para implementar este patrón, he optado por el framework Laravel, desarrollado en PHP, complementado con Blade, su motor de plantillas, para construir una interfaz intuitiva y funcional. Además, para dotar a estas interfaces de dinamismo, he empleado JavaScript junto con Tailwind CSS, un moderno framework de diseño basado en la utilización de clases, que permite aplicar estilos de manera eficiente y flexible a la web.

Los tres componentes principales en los cuales se organiza la aplicación son los siguientes:

a. Modelo

El modelo representa la estructura y organización de los datos en nuestro proyecto. Es responsable de gestionar la información almacenada en la base de datos, abarcando tanto la recuperación de datos existentes como el registro de nueva información.

Laravel ofrece una herramienta poderosa para desarrollar esta capa: los ORM (Object-Relational Mapping). Los ORM permiten mapear las tablas de la base de datos en objetos de programación, facilitando la persistencia y manipulación de los datos. Este enfoque mejora la interacción con la base de datos al proporcionar un manejo más intuitivo y eficiente de la información.

b. Vista

La vista es la encargada de presentar la información al usuario de manera clara y visualmente atractiva. Para este proyecto, se ha utilizado el motor de plantillas Blade, que destaca por su capacidad para generar interfaces

dinámicas y responsivas. Una de sus principales ventajas es la posibilidad de reutilizar código repetitivo, lo que resulta especialmente útil al crear componentes como formularios, botones o campos de entrada.

Además, Blade se integra con Vite, una herramienta moderna para la construcción del frontend. Vite simplifica la empaquetación y gestión de archivos CSS y JavaScript, optimizando tanto el rendimiento como el flujo de trabajo durante el desarrollo.

c. Controlador

El controlador actúa como el intermediario entre el modelo y la vista, ya que será el encargado de gestionar las solicitudes de los usuarios, a su vez, acceder a los datos de la base de datos mediante el uso de los modelos.

De forma resumida, el flujo de esta arquitectura sigue los siguientes pasos:

1. **Interacción del usuario:** El usuario interactúa con la vista desde el navegador, por ejemplo, al agendar una cita.
2. **Envío de la solicitud:** La solicitud del usuario se dirige a un controlador a través de una ruta previamente definida en Laravel.
3. **Validación:** El controlador valida los datos ingresados; si son correctos, los envía al modelo.
4. **Gestión de datos:** El modelo interactúa con la base de datos para realizar operaciones como crear, leer, actualizar o eliminar información.
5. **Respuesta:** Una vez completado el proceso, el modelo devuelve los datos al controlador, que los prepara y los envía a la vista.
6. **Presentación al usuario:** Finalmente, la vista muestra la respuesta al usuario de manera clara y comprensible.

3.2 Diseño de interfaz del usuario

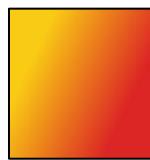
3.2.1 Introducción

El desarrollo de la interfaz ha seguido directrices orientadas a garantizar la usabilidad y accesibilidad. En el contexto de nuestro proyecto, nos hemos propuesto crear una experiencia visual y funcionalmente intuitiva, que facilite la gestión eficiente de las citas tanto para los propietarios como para sus clientes. Como se mencionó previamente, *Hairbooker* está diseñado para pequeñas peluquerías que buscan optimizar la organización de sus citas, proporcionando una interfaz amigable y de fácil uso, permitiendo así una gestión sencilla y rápida. Además, es crucial ofrecer al usuario mensajes de ayuda y retroalimentación para evitar que se sienta desorientado, sin embargo, desde una perspectiva crítica, la interfaz debe minimizar la necesidad de tales mensajes, evidenciando su facilidad de uso.

Por otro lado, es esencial que la interfaz se pueda adaptar a diferentes dispositivos (móviles, tabletas, ordenadores...), ya que nuestros usuarios accederán a la web desde diferentes entornos.

3.2.2 Elementos visuales

3.2.2.1 Colores



Gradiente de amarillo a rojo

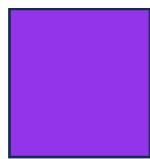
HEX: #FACC15

RGB: 250, 204, 21



HEX: #DC2626

RGB: 220, 38, 38



Color morado

HEX: #9333EA

RGB: 147, 51, 234



Color grisáceo

HEX: #374151

RGB: 55, 65, 81



Color azulado

HEX: #172554

RGB: 23, 37, 84



Color amarillo

HEX: #FFC107

RGB: 255, 193, 7



Color verde

HEX: #4CAF50

RGB: 76, 175, 80



Color rojo

HEX: #F44336

RGB: 244, 67, 54



Color azul

HEX: #4C65F3

RGB: 76, 101, 243

Estos son los colores principales utilizados en la web de Hairbooker y su propósito dentro del diseño. El color gradiente y el morado son los elementos centrales del estilo visual de la web, predominando en las partes más visibles, como el encabezado y los formularios. Estas tonalidades aportan coherencia y un aspecto moderno a la interfaz.

El color grisáceo se ha asignado a los modales que proporcionan retroalimentación al usuario, ya sea para mostrar errores o mensajes informativos, destacando su función sin romper la armonía del diseño.

En el panel de administración del dashboard se ha optado por un color azulado, seleccionado para mantener una conexión visual con la gama del morado, logrando una integración estética con el resto de la web.

Para el calendario, se han elegido colores específicos para indicar los estados de las citas:

- El amarillo representa citas no confirmadas por el propietario.
- El verde identifica citas confirmadas.
- El rojo indica citas canceladas por el propietario.
- El azul señala las citas de otros usuarios visibles en el calendario del cliente.

Esta paleta de colores no solo mejora la experiencia visual, sino que también facilita la identificación rápida y clara de los diferentes elementos y estados dentro de la plataforma.

3.2.2.2 Tipografía

La tipografía seleccionada para la web es Work Sans, una elección que garantiza legibilidad y estilo moderno en todas las secciones de la plataforma. Esta tipografía está presente de manera uniforme a lo largo del sitio, ajustando su tamaño dinámicamente para adaptarse a los diferentes dispositivos y garantizar una experiencia visual óptima en móviles, tabletas y ordenadores.

En los encabezados se emplea la variante en negrita, con el objetivo de resaltarlos y darles mayor jerarquía visual, facilitando la identificación de secciones importantes y mejorando la navegación para los usuarios.

Figura 1. Tipografía WorkSans

Whereas disregard

3.2.2.3 Iconografía

El logo diseñado para Hairbooker se encuentra ubicado estratégicamente en el encabezado de la página web, asegurando que sea el primer elemento que los usuarios visualicen al acceder. Este logo está compuesto únicamente por el

nombre "Hairbooker", destacando la marca de manera sencilla y directa para facilitar su reconocimiento e identificación. Su diseño minimalista se integra perfectamente con el estilo general de la web, reforzando la identidad visual de la plataforma.

Figura 2. Logo de *Hairbooker*



Otros iconos que se han utilizado para la página web son:



Iconos utilizados para acceder a las redes sociales de *Hairbooker*.



Iconos utilizados en los formularios para representar el tipo de cada campo.



3.3 Planificación de las tareas y los recursos necesarios

3.3.1 Introducción

La planificación de las tareas y de los recursos necesarios supone un elemento clave para conseguir los objetivos marcados. En este apartado, se detallarán las actividades previstas, los recursos utilizados y los plazos establecidos para la consecución de los objetivos.

3.3.2 Planificación de las tareas

FASE	ACTIVIDAD	DURACION
Análisis de requisitos	Establecimiento de los objetivos a nivel general y específico. Elaboración de los requisitos funcionales y no funcionales. Elaboración del modelo entidad-relación. Elaboración del diagrama de clases. Elaboración del diagrama de flujo.	1 semana
	Creación de las posibles vistas de la web.	

Diseño	<p>Selección de la paleta de colores y tipografía.</p> <p>Definición de la arquitectura del proyecto y elección de tecnología para su desarrollo.</p> <p>Creación de las tablas en la base de datos basadas en el modelo E/R.</p>	2 semanas
Desarrollo	<p>Creación de vistas con Blade.</p> <p>Creación de migraciones y de modelos en Laravel.</p> <p>Creación de controladores y rutas.</p> <p>Implementación de verificación de e/mail.</p> <p>Implementación de recuperación de contraseña vía e-mail.</p> <p>Desarrollo de multi-autenticación basada en roles.</p> <p>Implementación de formularios para creación de usuarios, peluquerías, citas y servicios.</p> <p>Integración de calendario interactivo con FullCalendar.</p> <p>Implementación de notificaciones.</p>	2 meses
	Comprobación de redirecccionamiento de rutas.	

Pruebas	Validación del sistema de autenticación. Pruebas de la web. Resolución de los errores detectados.	1 semana y media
Documentación	Redacción de la documentación	2 semanas

3.3.3. Recursos utilizados

Los recursos que se han utilizado para el desarrollo del proyecto han sido:

Hardware:

Portátil Asus Rog Strix con 16 GB de RAM y un AMD Ryzen 7 de procesador.

Software:

- Laravel: framework de PHP empleado para estructurar y desarrollar la lógica de la aplicación web, facilitando la implementación del patrón MVC y el manejo de rutas, controladores y modelos.
- MySQL: sistema de gestión de bases de datos relacional utilizado para almacenar y organizar la información de la aplicación de manera eficiente.
- Tailwind CSS: framework de CSS que simplifica el diseño visual de la web mediante un enfoque basado en clases HTML, permitiendo un desarrollo rápido y personalizado de los estilos.
- XAMPP: servidor local que integra Apache, MySQL, PHP y otras herramientas, utilizado para ejecutar y probar la aplicación durante el desarrollo.
- Visual Studio Code (VS Code): entorno de desarrollo integrado (IDE) elegido para la escritura y edición del código, gracias a su flexibilidad y amplia gama de extensiones.

- Mailtrap: herramienta que permite simular el envío de correos electrónicos durante el desarrollo, asegurando que las funcionalidades relacionadas se prueben sin afectar a usuarios reales.
- FullCalendar: plugin utilizado para integrar un calendario interactivo en la web, proporcionando una interfaz visual atractiva para la gestión de citas.
- Draw.io: aplicación empleada para crear y esquematizar los diagramas necesarios para la planificación y documentación del proyecto.
- Iconify Design: herramienta para la importación de diferentes iconos que aporten al estilo visual de la web.
- Docker: Herramienta de virtualización utilizada para crear contenedores que encapsulan las aplicaciones y sus dependencias, garantizando un entorno de desarrollo y producción uniforme. Docker permite desplegar la aplicación de manera rápida y eficiente,

Esta combinación de herramientas ha sido seleccionada para garantizar un desarrollo ágil, organizado y eficiente, cubriendo cada aspecto del proyecto de manera integral.

4 IMPLEMENTACIÓN Y PRUEBAS

4.1 Desarrollo de las funcionalidades del proyecto

El principal objetivo en el desarrollo de *Hairbooker* ha sido garantizar una experiencia sencilla y clara para el usuario, evitando que se sienta abrumado por un exceso de opciones al acceder a la web. Desde el primer momento, el usuario debe comprender que el propósito central de la plataforma es la gestión de citas, lo que constituye su funcionalidad principal.

Para cumplir con esta premisa, se hizo imprescindible separar las responsabilidades y funcionalidades en capas bien definidas, lo que llevó a la adopción del patrón de diseño MVC (Modelo-Vista-Controlador). Este enfoque

estructurado facilita el desarrollo, mantenimiento y escalabilidad del proyecto, al tiempo que asegura una interacción intuitiva para el usuario.

A continuación, se detallará el desarrollo de las funcionalidades de la web, comenzando desde sus elementos iniciales.

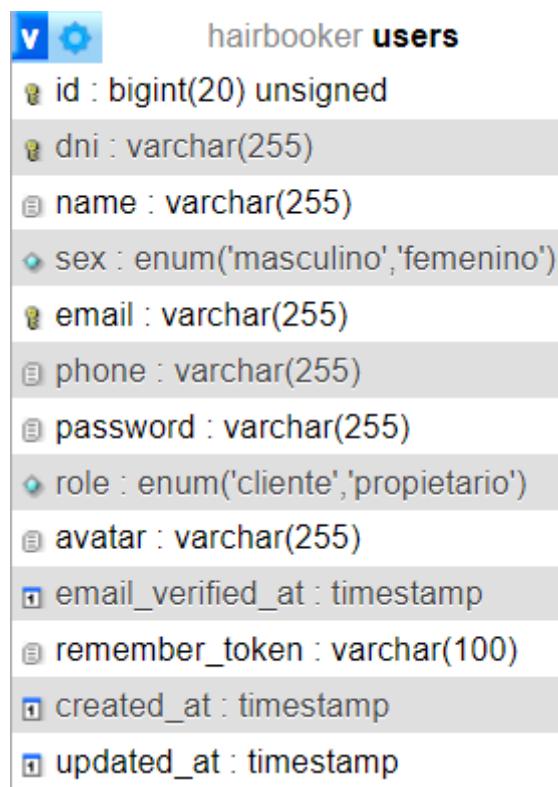
4.1.1 Gestión de usuarios

Desde el principio, se concibió Hairbooker como una plataforma diseñada para atender a dos tipos de usuarios con necesidades específicas:

- **El cliente:** Este usuario puede conectarse directamente con su peluquería, accediendo a un calendario actualizado para gestionar sus citas de manera eficiente. De esta forma, se elimina la posibilidad de pérdida de información y se facilita una experiencia más organizada y transparente.
- **El propietario:** Este usuario tiene a su disposición herramientas para gestionar tanto las citas como los servicios de su peluquería de forma visual y estructurada. Además, puede mantenerse en contacto con sus clientes, lo que le permite ofrecer una atención más personalizada y fortalecer la relación con ellos.

Esta dualidad de roles asegura que la plataforma sea útil y funcional para ambas partes, promoviendo una interacción fluida y beneficiosa entre clientes y propietarios.

Para lograr esta dualidad, en nuestra base de datos contamos con una tabla de usuarios, los cuales dividiremos por roles: cliente y propietario. Gracias a esta división, aprovechamos mejor las herramientas que dispone Laravel para mostrar contenido a un usuario u otro.

Figura 3. Tabla Usuarios.

The screenshot shows the 'users' table from the 'hairbooker' database. The table has 11 columns:

	Column Name	Type
v	id	bigint(20) unsigned
v	dni	varchar(255)
v	name	varchar(255)
v	sex	enum('masculino','femenino')
v	email	varchar(255)
v	phone	varchar(255)
v	password	varchar(255)
v	role	enum('cliente','propietario')
v	avatar	varchar(255)
v	email_verified_at	timestamp
v	remember_token	varchar(100)
v	created_at	timestamp
v	updated_at	timestamp

Gracias a la directiva proporcionada por Laravel con el uso de las anotaciones @auth, para verificar que el usuario esta verificado, podremos acceder al rol del usuario y mostrar u ocultar contenido según el rol que este tenga. En las siguientes imágenes, se muestra un ejemplo de ello.

Figura 4. Código para implementar la multiautenticación.

```

@auth
@if (Auth::user()->role == 'cliente')
- <a id="appointments"></a>>Mis citas</a>
- <a id="signup"></a>>Darse de alta en peluquería</a>

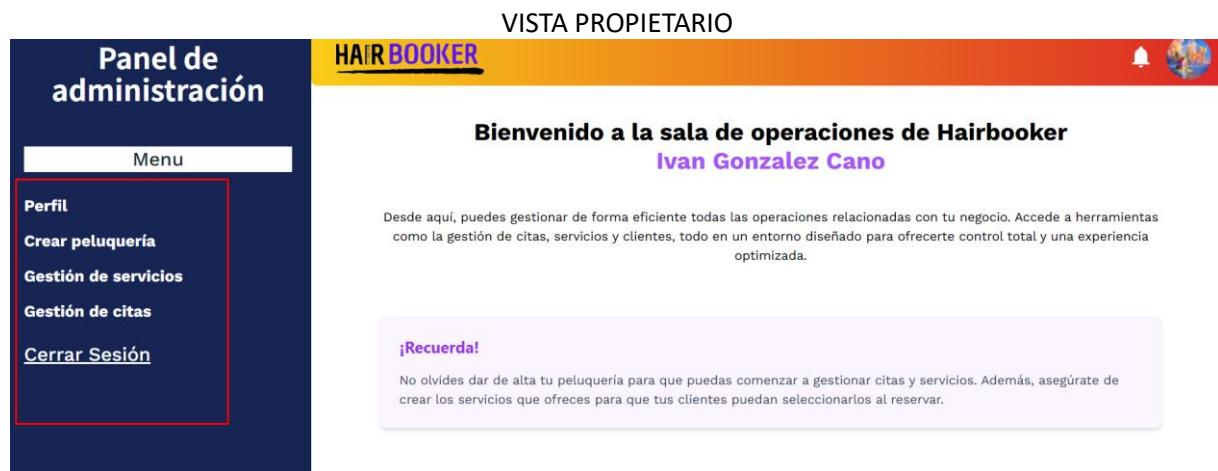
@endif
@endauth

@auth
@if (Auth::user()->role == 'propietario')
- >Crear peluquería</a>
- >Gestión de servicios</a>
- >Gestión de citas</a>


@endif
@endauth

```

Figura 5. Resultado de la multiautenticación.





4.1.2 Gestión de servicios

Antes de gestionar las citas, es fundamental que la peluquería cuente con una lista de servicios disponibles para ofrecer a sus clientes. Por ello, se consideró esencial incluir una sección dedicada exclusivamente al propietario, donde pueda administrar los servicios de su peluquería.

Esta funcionalidad permite al propietario agregar, editar o eliminar servicios de manera sencilla y eficiente, asegurando que la oferta esté siempre actualizada y alineada con las necesidades de los clientes.

Esta sección está diseñada con una tabla que incluye botones para crear, actualizar y eliminar servicios, tal como se mostró en el apartado de Diseño de Interfaz de Usuario. Para utilizarla, el propietario debe seleccionar la peluquería en la que desea gestionar sus servicios. Al hacer clic, se abrirá un modal que permitirá realizar las acciones correspondientes:

- Creación de servicios:** En este modal, los campos del formulario aparecerán vacíos. Una vez que el propietario complete los datos necesarios y envíe el formulario, la información será enviada al servidor. La página se actualizará dinámicamente utilizando la API Fetch, evitando la necesidad de recargar completamente la página.
- Actualización de servicios:** Este modal presenta un formulario similar, pero los campos estarán prellenados con los datos del servicio que se desea

modificar. Esto facilita al propietario realizar cambios sin perder la información existente.

- **Eliminación de servicios:** En este caso, el modal muestra un mensaje de confirmación antes de proceder con el borrado definitivo del servicio en la base de datos.

Esta estructura mejora la experiencia del usuario, ofreciendo una gestión intuitiva y eficiente de los servicios sin interrupciones innecesarias en el flujo de trabajo.

Figura 6. Modal para creación de servicio.

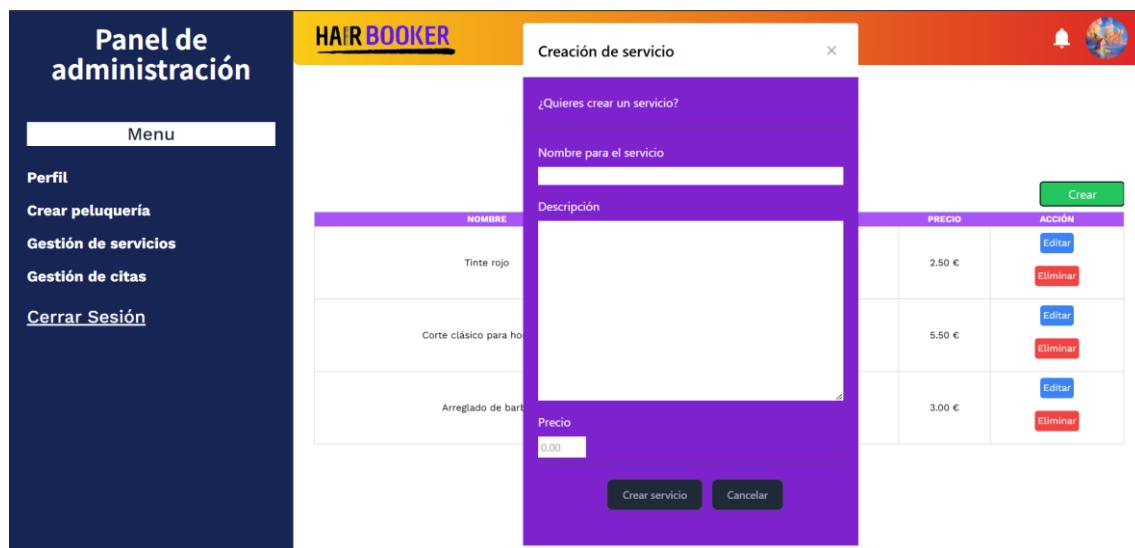


Figura 7. Modal para modificación de servicio.

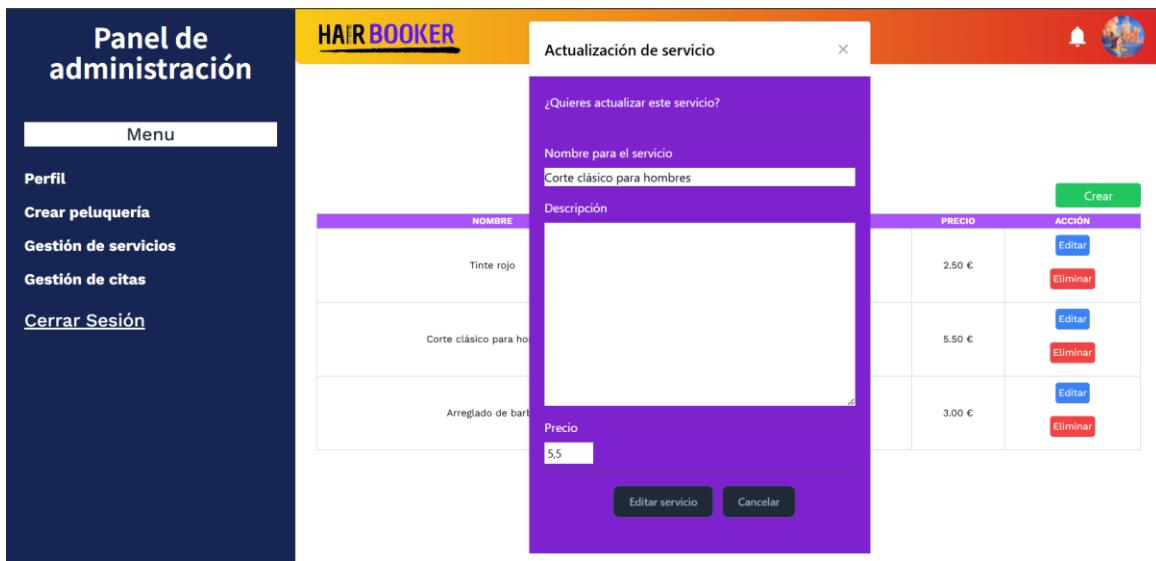
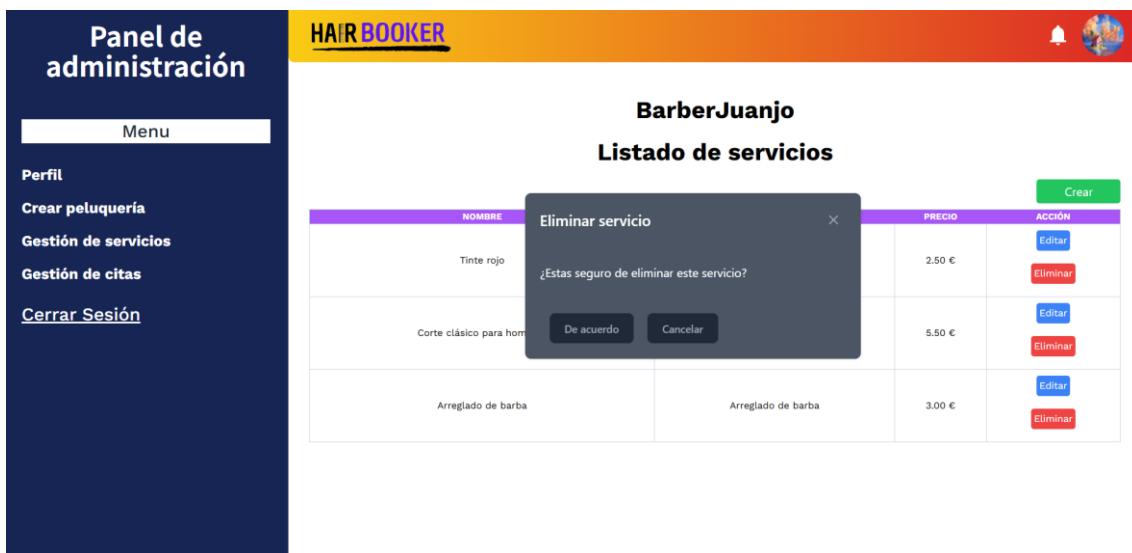


Figura 8. Modal para eliminación de servicio.



4.1.3 Gestión de citas

La gestión de citas supone la parte más importante del proyecto. Esta sección se compone de un calendario organizado en días y en horas. Para realizar el calendario, se ha optado por utilizar la Librería de *FullCalendar*, librería de JavaScript que permite renderizar fácilmente un calendario interactivo. Esta librería dispone de varias herramientas para lograr el registro de eventos en el calendario, lo cual ha facilitado el desarrollo de la funcionalidad de gestión de citas.

La vista del calendario ya se ha mostrado en apartados anteriores. En este apartado, detallaremos la funcionalidad y el cómo el usuario podrá gestionar sus citas. Esta gestión será diferente según el usuario sea cliente o propietario.

De manera general, los usuarios podrán crear sus citas haciendo click en cualquier hora del calendario. En el momento que realicen click, aparecerá un modal, compuesto por un formulario con los datos a introducir de la cita. A continuación, explicaremos las diferencias entre ambos usuarios a la hora de ver, crear, actualizar y eliminar una cita.

- **Cliente:** el usuario cliente podrá ver sus citas en el calendario y también la de otro cliente, aunque estas últimas, por privacidad a los otros clientes, no se mostrará su nombre, como se observa en la siguiente imagen:

Figura 9. Vista de citas del usuario cliente.

Para crear una cita, el cliente hará click a la hora que el estime y aparecerá el modal con el formulario. El formulario de creación de las citas tendrá tres campos, el servicio que desea el cliente y la fecha de inicio y de final de la cita, la cual se encuentra predefinida en 1 hora de duración. El estado de la cita siempre será pendiente, ya que el usuario no podrá confirmar su cita, solo tiene esta posibilidad el propietario.

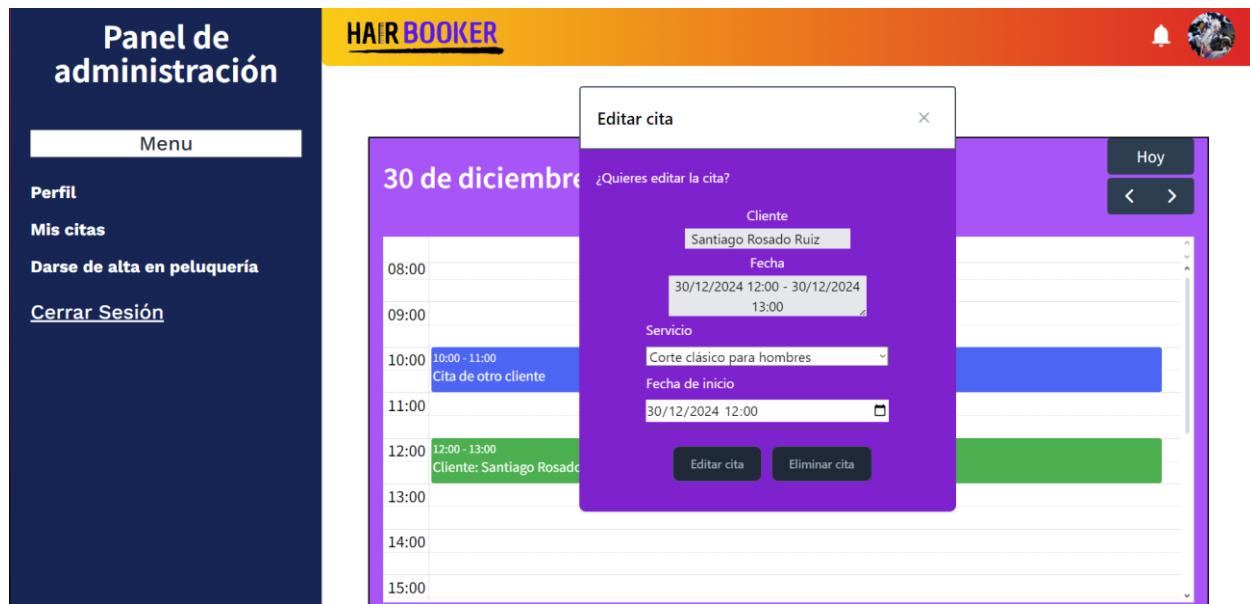
Figura 10. Vista de modal para creación de una cita del usuario cliente.

En el proceso de actualización de citas, el usuario podrá interactuar directamente con el calendario. Al hacer clic sobre una de sus citas, se abrirá un modal que le permitirá realizar modificaciones. Este modal incluirá un formulario en el que el usuario podrá cambiar el servicio seleccionando entre los disponibles ofrecidos por la peluquería y modificar la fecha de inicio eligiendo una nueva fecha y hora para su cita, siempre dentro de los horarios disponibles.

Es importante destacar que el sistema está diseñado para garantizar la privacidad y seguridad de los datos de otros usuarios, por lo que un cliente solo podrá visualizar y actualizar sus propias citas sin posibilidad de editar las citas de otros clientes.

Este enfoque asegura que el usuario tenga control sobre sus reservas de forma sencilla y eficiente mientras se respetan las restricciones necesarias para proteger la integridad del sistema.

Figura 11. Vista de modal para la actualización de una cita del usuario cliente.



Para eliminar una cita, el procedimiento es similar al de actualizar. El usuario deberá hacer clic sobre una de sus citas en el calendario, lo que abrirá un modal. En este modal, además de las opciones de actualización, estará disponible un botón para eliminar la cita.

Al pulsar este botón, el sistema mostrará un mensaje de aviso solicitando confirmación. Este mensaje permitirá al usuario decidir si desea proceder con la eliminación o cancelar la acción. Si el usuario confirma, la cita se eliminará de la base de datos y el calendario se actualizará dinámicamente para reflejar los cambios sin necesidad de recargar la página.

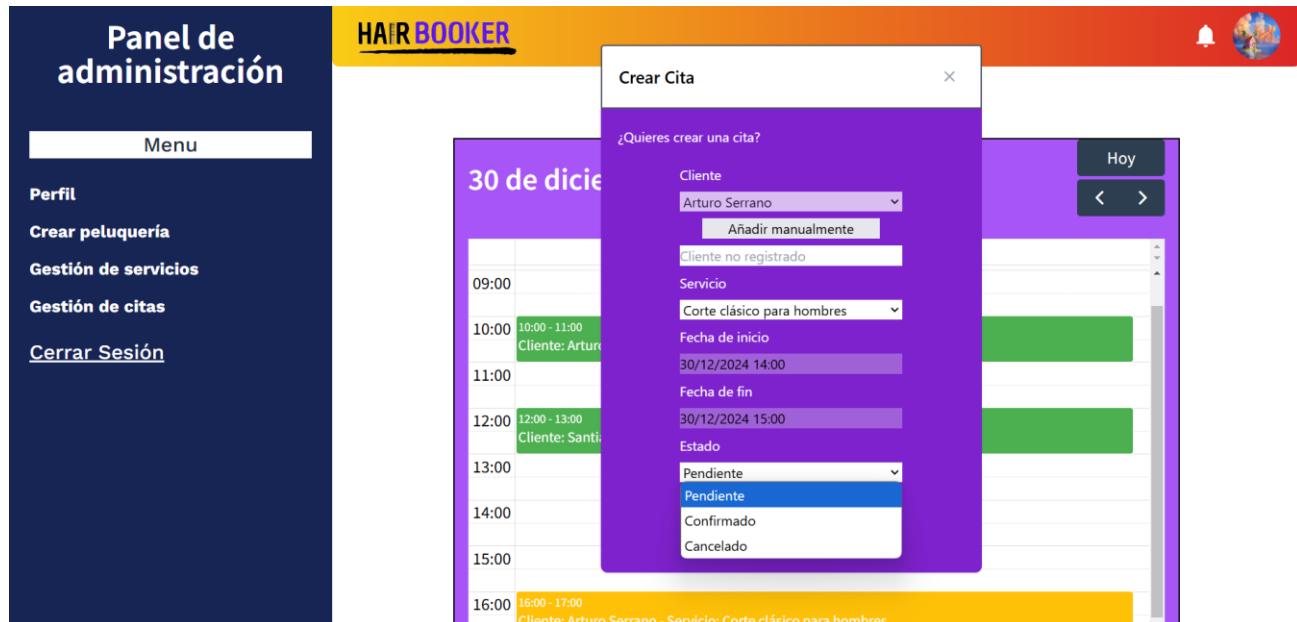
Figura 12. Vista de modal para la eliminación de una cita del usuario cliente.



- **Propietario:** el usuario propietario, a diferencia del cliente, tendrá la capacidad de visualizar todas las citas relacionadas con su peluquería. El funcionamiento para crear, actualizar y eliminar citas sigue un procedimiento similar al descrito para el cliente. Sin embargo, el propietario cuenta con funcionalidades adicionales.

El propietario puede crear citas para cualquier cliente, independientemente de si está dado de alta en la aplicación o no. Además, puede gestionar los estados de las citas, permitiendo mayor control sobre el flujo de trabajo en su peluquería. Por este motivo, el formulario de creación de citas presenta diferencias respecto al del cliente, incluyendo campos adicionales para especificar datos del cliente y seleccionar el estado de la cita, como se puede observar en la siguiente imagen.

Figura 13. Vista de modal para la creación de una cita del usuario propietario.



El modal para la actualización y eliminación de la cita será bastante similar, diferenciándose en la posibilidad de seleccionar estado por parte del propietario.

Cualquier cambio en las citas será informado al cliente o al propietario, mediante notificaciones como se observa en la siguiente imagen.

Figura 14. Vista de notificación.



4.1.4 Otras funcionalidades

Hairbooker también incluye otras funcionalidades, como permitir que los usuarios clientes se den de alta en una peluquería. Para ello, se dispone de un desplegable donde se muestran las peluquerías registradas en la plataforma. El cliente podrá seleccionar la peluquería en la que desea darse de alta y completar el proceso de forma sencilla.

Por otro lado, el usuario propietario podrá registrar en la plataforma su peluquería a través de un sencillo formulario.

4.2 Pruebas unitarias y de integración

4.2.1 Pruebas en lado servidor

Para la realización de las pruebas unitarias en servidor utilizaremos las herramientas que nos proporciona Laravel para la realización de test PHPUnit. Para ello comenzaremos con las pruebas de los controladores y las rutas.

- **HomeController:** este controlador dispone de dos métodos que se encargan de mostrar la vista principal.

Figura 15. Test del controlador HomeController.

```
PASS Tests\Feature\HomeControllerTest
✓ it loads the index view
✓ it loads the about me view
```

- **UserController:** este controlador se encarga de gestionar todas las acciones relacionadas con el usuario, tales como el registro, inicio de sesión, cambio de contraseña, entre otras. En los siguientes tests, se evaluarán diferentes aspectos de estas funcionalidades, incluyendo la inserción de nuevos registros de usuarios, el manejo de errores cuando un usuario ya existe, el proceso de inicio de sesión y la validación de credenciales incorrectas, el restablecimiento de contraseña, y el cierre de sesión (logout) del usuario.

Figura 16. Test del controlador UserController.

```
PASS Tests\Feature\UserControllerTest
✓ it loads the login view
✓ it loads the register view
✓ it loads the login view
✓ it loads the register view
✓ it loads the forgot password view
✓ it loads the register view
✓ it loads the forgot password view
✓ it loads the forgot password view
✓ it registers a new user
✓ it fails to register a user with existing email
✓ it authenticates an existing user
✓ it fails to authenticate with incorrect credentials
✓ it resets password
✓ it logs out the user

Tests: 9 passed (24 assertions)
Duration: 1.41s
```

- **DashboardController:** este controlador se encarga de gestionar todas las acciones dentro del dashboard de *Hairbooker*, abarcando desde la actualización del perfil del usuario hasta la creación de servicios y la carga del calendario de citas.

Figura 17. Test del controlador DashboardController.

```
PASS Tests\Feature\DashboardControllerTest
✓ it loads the profile view
✓ it loads the hairdresser view
✓ show services returns 401 if user not authenticated
✓ show services returns 404 if hairdresser not found
✓ show services displays view with hairdresser and services
✓ select hairdresser returns view with hairdressers
✓ select hairdresser redirects if user not authenticated
✓ show hairdressers returns view with hairdressers and signed up hairdressers
✓ show hairdressers redirects if user not authenticated
✓ update profile success
✓ update profile validation error
✓ store hairdresser success
✓ store hairdresser validation error
✓ store hairdresser authorization error
✓ create service success
✓ create service authorization error
✓ create service validation error
✓ update service success
✓ update service not found
✓ update service authorization error
✓ update service validation error
✓ delete service success
✓ delete service not found
✓ delete service authorization error
✓ signup hairdresser success
✓ signup hairdresser authorization error
✓ signup hairdresser validation error
✓ signup hairdresser not found error
✓ signup hairdresser already signed up error

Tests: 29 passed (68 assertions)
Duration: 1.83s
```

- **CalendarController:** se encarga de mostrar el calendario a los usuarios.

Figura 18. Test del controlador CalendarController.

```
PASS Tests\Feature\CalendarControllerTest
✓ it find signup redirects to login if user is not authenticated
✓ display hairdressers for authenticated client
✓ display hairdressers for owners
✓ show calendar with valid hairdresser id
✓ show calendar with invalid hairdresser id
✓ show calendar hairdresser not found

Tests: 6 passed (28 assertions)
Duration: 1.14s
```

- **AppointmentController:** controlador encarga de mostrar las citas en el calendario, y de la gestión de las citas.

Figura 19. Test del controlador AppointmentController.

```
PASS Tests\Feature\AppointmentControllerTest
✓ create appointment success
✓ create appointment error time taken
✓ create appointment validation error
✓ update appointment success
✓ update appointment validation error
✓ delete appointment success
✓ delete appointment not found

Tests: 10 passed (20 assertions)
Duration: 1.20s
```

4.2.2 Pruebas en lado cliente

En este apartado se describen las pruebas realizadas en el lado cliente para garantizar la correcta funcionalidad de las interfaces y la interacción con el usuario. Estas pruebas se enfocan en verificar la validación de formularios, el comportamiento dinámico de los componentes y la comunicación con el servidor, asegurando que la experiencia del usuario sea fluida, intuitiva y libre de errores. A través de estas evaluaciones, se busca detectar y corregir posibles fallos antes del despliegue final del proyecto.

4.2.2.1 Formulario de registro de usuario

En el formulario de registro de usuario se realiza una validación en tiempo real al usuario. Si el usuario hace caso omiso de las validaciones en tiempo real y envía el formulario al servidor, se realiza una doble validación en el servidor y se muestra un modal informando del error al usuario.

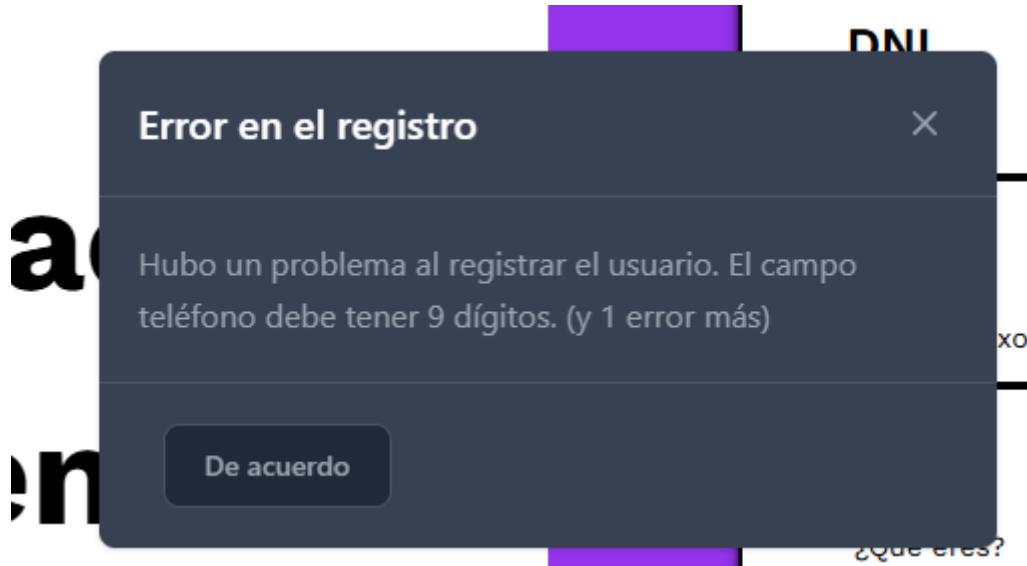
Figura 20. Validación en el registro de usuario.

The screenshot shows a user registration form with the following fields and their current values:

- Nombre:** Santiago Rosado Ruiz (highlighted with a red border)
- DNI:** 06292184M
- Sexo:** Hombre
- Rol:** Propietario
- E-mail:** santiago.rosadooutlook.es (highlighted with a red border)
- Teléfono:** 2312312

Validation messages displayed below the fields are:

- For Name: **Correcto**
- For Email: **Introduce un email válido**
- For Phone: **El teléfono debe empezar por 6 o 9 y tener 9 dígitos.**

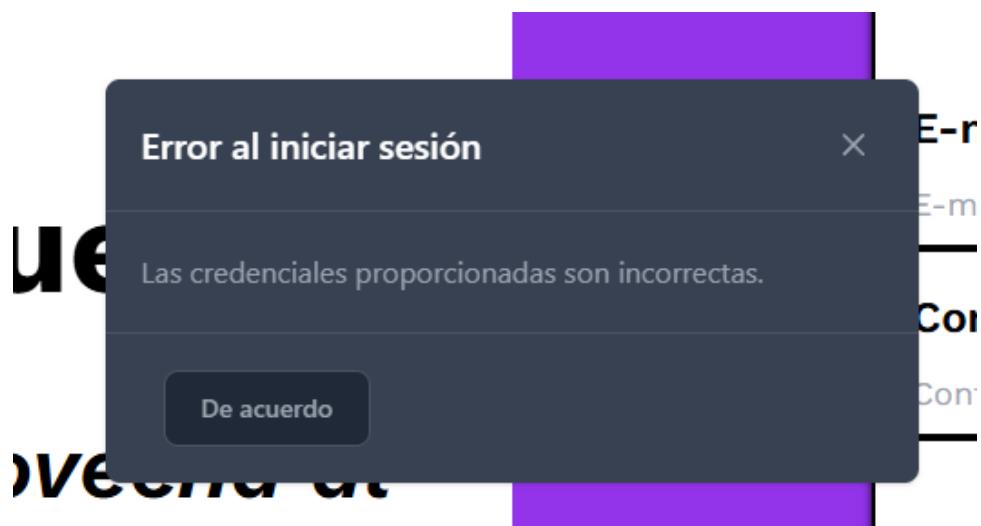


Si el registro se realiza de forma correcta, se mostrará en pantalla el mensaje de que se ha enviado un e-mail a la dirección de correo electrónico introducida para la verificación del e-mail. Una vez verificado el e-mail se redirige al nuevo usuario al dashboard.

4.2.2.2 Formulario para iniciar sesión

En este formulario, si el usuario introduce las credenciales incorrectas se le mostrará un modal de error donde se le indicará que los datos introducidos son incorrectos. Si el usuario inicia sesión correctamente, se le redirigirá a la página del dashboard.

Figura 21. Modal de error por credenciales incorrectas.



Respecto al cambio de contraseña, el usuario debe introducir en un formulario su dirección de e-mail registrada en la plataforma. Si la dirección no se encuentra, se mostrará un mensaje de aviso al usuario. Si existe, se enviará al correo electrónico un enlace el cual mostrará un formulario donde debe introducir su nueva contraseña.

Figura 22. Modal de error por e-mail no existente al reestablecer contraseña.



4.2.2.3 Dashboard – actualización de perfil

En la sección de actualización de perfil, el formulario cuenta con un mecanismo de doble validación: en el lado cliente y en el servidor. A través de este formulario, los usuarios tienen la posibilidad de actualizar su información personal de manera sencilla y segura. En caso de que ocurra algún inconveniente durante el proceso, se mostrará un modal de error que proporcionará información clara sobre el problema, permitiendo al usuario corregirlo fácilmente.

Figura 23. Validación del formulario para actualización del perfil en lado cliente y servidor.

Perfil del usuario



Hola, Santiago Rosado Ruiz

Nombre	Santiago Rosado Ruiz	
DNI	06292184	
Teléfono	665613319	
Nueva Contraseña	Contraseña...	
Repite la contraseña	Contraseña...	



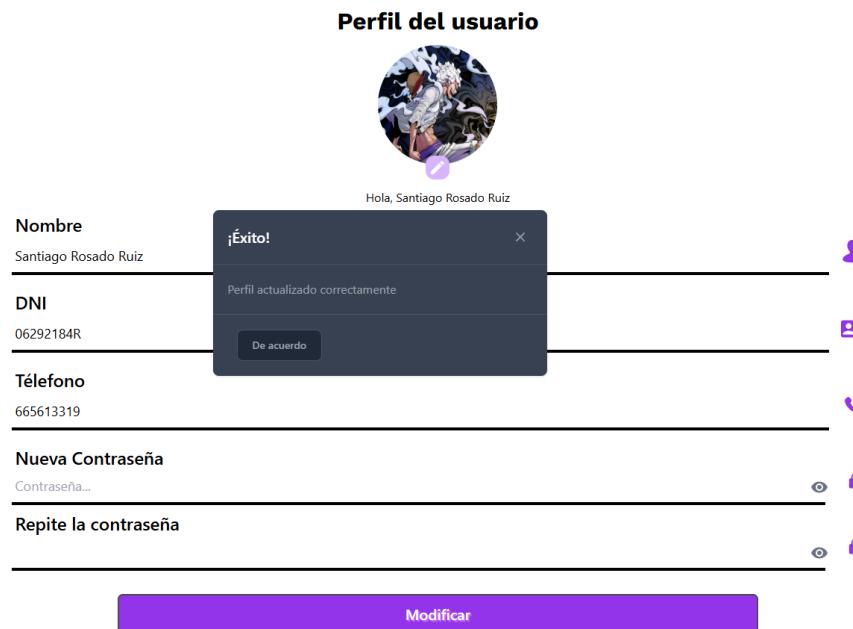
Hola, Santiago Rosado Ruiz

Nombre	Santiago Rosado Ruiz	
DNI	¡Éxito!	
Teléfono	06292184	
Nueva Contraseña	Contraseña...	
Repite la contraseña	Contraseña...	

Error: El formato del campo dni no es válido.

De acuerdo

Figura 24. Mensaje de éxito en la actualización del perfil.



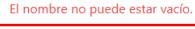
4.2.2.4 Dashboard – crear peluquería

En la sección destinada al alta de una peluquería, el formulario implementa un sistema de doble validación: en el lado cliente y en el servidor. Este enfoque garantiza que los datos proporcionados por el usuario cumplan con los requisitos establecidos antes de ser procesados. A través de este formulario, los usuarios tienen la posibilidad de registrar su peluquería de manera eficiente y segura. En caso de que se detecte algún error durante el proceso, se mostrará un modal informativo que indicará al usuario la naturaleza del problema y las acciones necesarias para resolverlo.

Figura 25. Validación del formulario para registro de peluquería en lado cliente y servidor.

Añade tu peluqueria

Nombre 

El nombre no puede estar vacío. 

CIF 

A12345544 

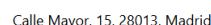
Correcto 

Teléfono 

665613319 

Correcto 

Dirección 

Calle Mayor, 15, 28013, Madrid 

Correcto 

Añadir peluqueria

Nombre 

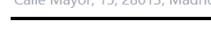
CIF 

A12345678... 

Teléfono 

665612234... 

Dirección 

Calle Mayor, 15, 28013, Madrid 

Error al enviar el formulario 

Error: El campo nombre es obligatorio. 

De acuerdo 

Añadir peluquería

Figura 26. Mensaje de éxito en el registro de la peluquería.

Añade tu peluqueria

Nombre	Avatar
CIF	Avatar
A12345678...	
Teléfono	Icono teléfono
665612234...	
Dirección	Icono casa
Calle Mayor, 15, 28013, Madrid	

¡Éxito!

Peluquería creada correctamente.

De acuerdo

Añadir peluquería

4.2.2.5 Dashboard – gestión de servicios

Las pruebas que se han realizado, en primer lugar, es detectar que la peluquería tiene servicios, si no tiene servicios se muestra en pantalla que no tiene servicios y el botón de crear. Si tiene se muestra en pantalla la lista de servicios con los botones de crear, actualizar y eliminar.

En el caso de la creación de servicios se trata de un formulario, Si los datos introducidos son erróneos, se muestra un modal con el mensaje del error en la validación de los datos. En caso de un registro con éxito, se actualiza la tabla con el nuevo servicio gracias a la API Fetch sin tener que recargar la página.

Figura 27. Mensaje de error al crear servicio por error en validación de campos del formulario.

The screenshot shows two parts of a user interface related to service creation. The top part is a modal dialog titled 'Creación de servicio' with a purple header. It contains fields for 'Nombre para el servicio' (Service name), which is highlighted with a red border indicating an error, and 'Descripción' (Description). Below these is a 'Precio' (Price) field with '0.00' entered. At the bottom are 'Crear servicio' (Create service) and 'Cancelar' (Cancel) buttons. The bottom part is a table with columns 'NOMBRE', 'DESCRIPCIÓN', and 'PRECIO'. It lists three services: 'Corte de pelo normal' (Normal haircut), 'Arreglado de barba' (Shave trim), and 'Corte para niños' (Children's haircut). A dark blue modal dialog titled 'Error al crear el servicio' (Error creating service) is overlaid on the table, displaying the message 'Error en la solicitud: El campo nombre es obligatorio. (y 1 error más)' (Request error: The name field is mandatory. (and 1 more error)). At the bottom of this modal is a 'De acuerdo' (Agree) button.

NOMBRE	DESCRIPCIÓN	PRECIO
Corte de pelo normal		9.00 €
Arreglado de barba		6.00 €
Corte para niños		3.00 €

El proceso de actualización de servicios sigue la misma lógica que el de creación. Si el formulario no contiene datos válidos, se muestra un mensaje de error al usuario. Por el contrario, si los datos ingresados son correctos, la tabla de servicios se actualiza automáticamente para reflejar los cambios realizados.

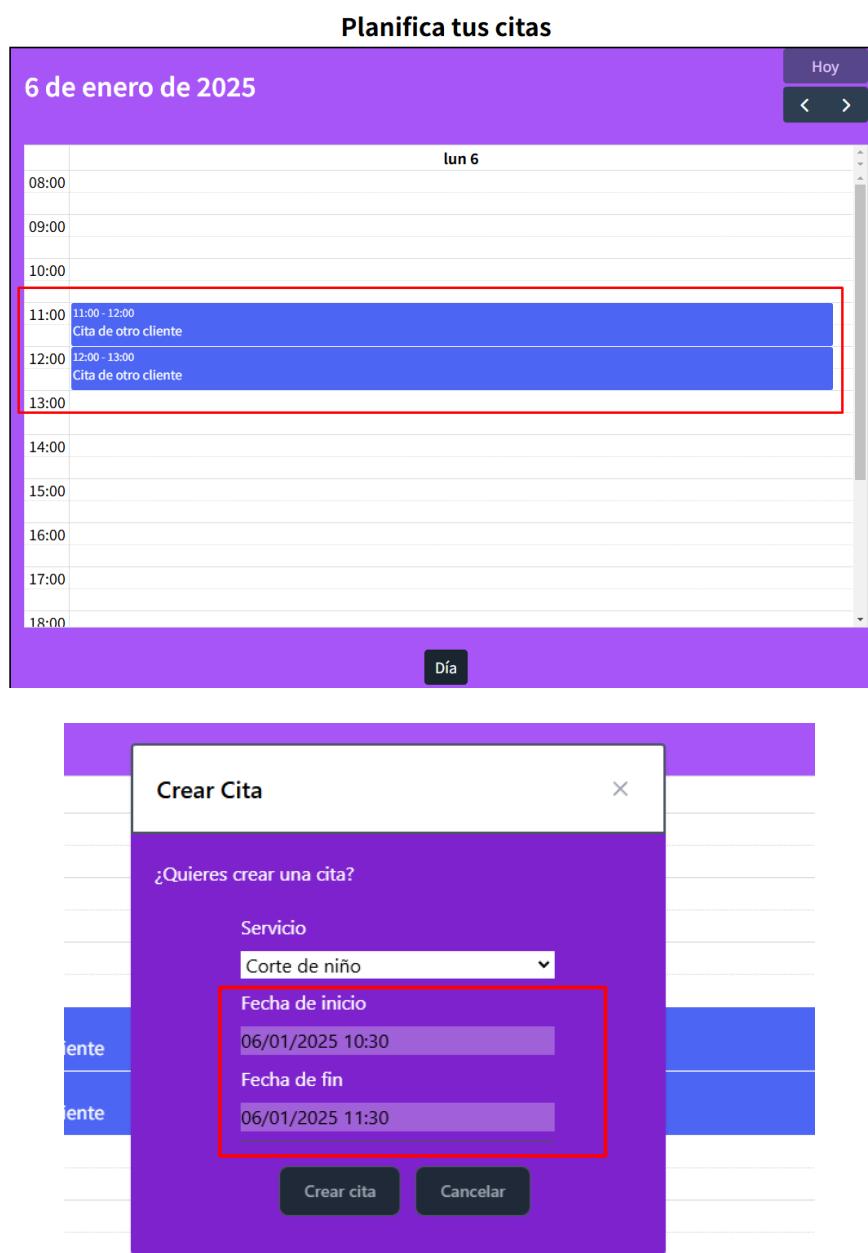
En el caso de la eliminación de servicios, el procedimiento es similar: si la operación se completa con éxito, la tabla se actualiza inmediatamente. Este enfoque asegura una experiencia de usuario coherente y dinámica, manteniendo siempre los datos en pantalla sincronizados con las operaciones realizadas.

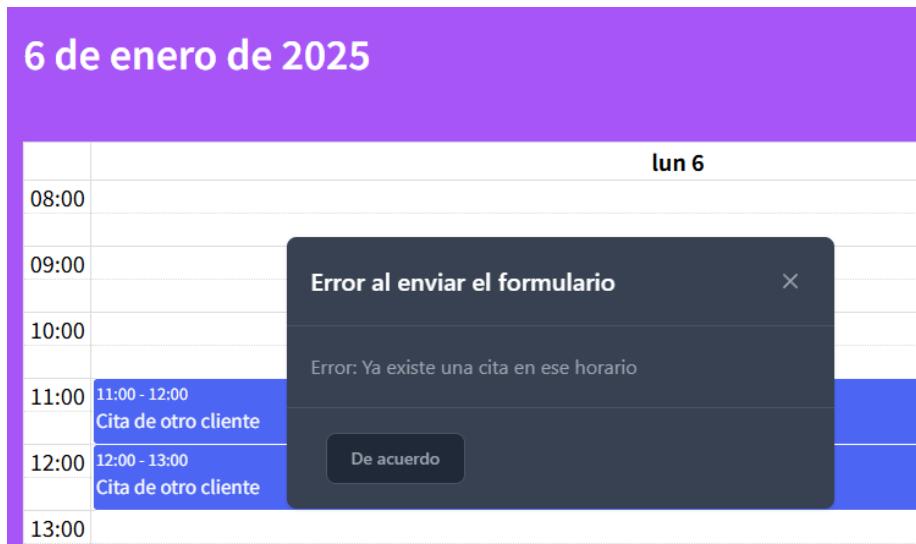
4.2.2.6 Dashboard – Gestión de citas

En la gestión de citas, el principal error a evitar es el solapamiento de horarios.

Para prevenirlo, se ha implementado un control en el servidor. Si un usuario intenta insertar una cita en un horario que ya está ocupado, el sistema verifica el estado de la cita existente. En caso de que la cita en conflicto esté cancelada, no habrá inconvenientes. Sin embargo, si la cita está activa, se mostrará un modal de aviso informando al usuario de que ya existe una cita programada en ese horario.

Figura 28. Mensaje de error al crear cita en un horario ya reservado.





En cuanto a las notificaciones enviadas por los usuarios, se ha verificado que el sistema genera y muestra correctamente notificaciones en la sección correspondiente cada vez que se registra, modifica, elimina o cancela una cita. Estas notificaciones reflejan los cambios realizados, como se ilustra en la siguiente imagen. Además, se ha comprobado que una vez que el usuario marca una notificación como leída, esta no vuelve a aparecer.

Figura 29. Mensaje de éxito en el registro de la peluquería.



5 DOCUMENTACIÓN

5.1 Documentación técnica

5.1.1 Descripción general

La web *Hairbooker* sigue el patrón MVC (Modelo – Vista - Controlador), que es utilizado para aplicaciones web basadas en PHP y Laravel. Este patrón separa la lógica de negocio, la presentación de la interfaz y el control de las interacciones de los usuarios en partes independientes.

- **Modelo:** se encarga de la gestión de la lógica de negocio, representando las entidades (tablas) de la base de datos.
- **Vista:** se trata de la interfaz que ve el usuario al utilizar la web.
- **Controlador:** se encarga de procesar las solicitudes del usuario y coordinar la comunicación entre el modelo y la vista.

Figura 30. Esquema de la estructura de *Hairbooker*



Principalmente, la plataforma se compone de los siguientes componentes:

- **Frontend:** desarrollado con el motor de plantillas Blade de Laravel, Tailwind CSS para los estilos y JavaScript con la API Fetch.
- **Backend:** desarrollado completamente con las herramientas proporcionadas por Laravel (rutas, controladores y modelos).

- **Base de datos:** se ha utilizado MySQL para el almacenamiento de los datos.

5.1.2 Estructura del proyecto

La estructura del proyecto sigue las convenciones recomendadas por Laravel, organizando el código de manera clara y modular. Los directorios principales son los siguientes:

- **app:** contiene la lógica de negocio de la aplicación, como son los controladores, modelos, servicios y middlewares. Dentro de este directorio podemos destacar:
 - **Http:** en este directorio se encuentran los controladores y middlewares creados.
 - **Models:** contiene las entidades o modelos que representan a las tablas de la base de datos.
 - **Services:** contiene los servicios utilizados en la lógica de negocio,
 - **View:** contiene los componentes creados para su uso en nuestras vistas o plantillas de Blade.
- **config:** contiene los archivos de configuración para la base de datos, el mail de la aplicación, y las sesiones.
- **database:** contiene las migraciones creadas para el desarrollo de la base de datos.
- **routes:** en este directorio se encuentra las rutas definidas para la web. Estas rutas serán las que comuniquen a los controladores con la parte cliente. En este caso, las masas utilizados son web.php y auth.php.

- **storage:** directorio destinado a almacenar archivos generados por la aplicación, como logs, chaché y archivos subidos por los usuarios, como es nuestro caso con la imagen del avatar.
- **vendor:** almacena las dependencias instaladas a través de Composer.

5.1.3 Dependencias y herramientas

- **Laravel:** framework PHP para el desarrollo de la estructura del proyecto y del backend.
- **JavaScript:** lenguaje para el desarrollo en lado cliente.
- **Tailwind CSS:** framework CSS para estilizar las vistas.
- **Vite:** herramienta para la gestión y compilación de rutas y assets.

Para la instalación de las dependencias, se ha de instalar composer con la siguiente instrucción:

composer install

Después de la instalación de composer, se procede a instalar las dependencias con el siguiente comando:

npm install

5.1.4 Configuración

5.1.4.1 Variables de entorno

A través del archivo. env se realizará la configuración de las variables del entorno. Se configurará la conexión a la base de datos y el servidor mail que se encargará del envío de los emails a los clientes.

Figura 31. Configuración de la Base de Datos

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=hairbooker
DB_USERNAME=adminrosado
DB_PASSWORD=rosadoruiz
```

Figura 32. Configuración del servidor Mail

```
MAIL_MAILER=smtp
MAIL_HOST=sandbox.smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=0e5d666db2a635
MAIL_PASSWORD=7a52f63bbc9219
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS="hbookercompany@outlook.com"
MAIL_FROM_NAME="HairBooker"
```

5.1.4.2 *Configuración Tailwind*

Tailwind también se debe configurar antes de su uso en el proyecto. En la configuración podemos establecer donde debe buscar los estilos, personalización de sus clases (nuevas animaciones, nuevos tipos de letras, nuevos colores...).

Figura 33. Configuración de Tailwind CSS

```
/** @type {import('tailwindcss').Config} */
export default {
  content: [
    "./resources/**/*.{blade.php}",
    "./resources/**/*.{js}",
    "./resources/**/*.{vue}",
  ],
  theme: {
    extend: {
      fontFamily: {
        noto: ["Noto Sans JP", "sans-serif"],
        work: ["Work Sans", "sans-serif"],
      },
      transitionProperty: {
        'height': 'height',
      },
    },
  },
  plugins: [
    require(`String id: "tailwindcss-animated"`),
    function ({ addUtilities }) {
      addUtilities({
        ".text-shadow-purple": {
          "text-shadow": "2px 2px 2px rgba(238, 153, 249, 0.8)",
        },
        ".text-shadow-lg": {
          "text-shadow": "4px 4px 4px rgba(0, 0, 0, 0.7)",
        },
      });
    },
  ],
};
```

Para utilizar Tailwind CSS en el proyecto, debemos integrarlo con Vite, para ello importaremos tailwind en el archivo css app.css del proyecto Laravel, para que Vite detecte que debe compilar los estilos de Tailwind CSS.

Figura 34. Conexión Tailwind con Vite

```
@tailwind base;  
@tailwind components;  
@tailwind utilities;
```

5.1.4.3 Configuración de Vite

El compilador de assets que utiliza Laravel, Vite, también necesita una configuración para un correcto funcionamiento.

Figura 35. Configuración de Vite

```
import { defineConfig } from 'vite';
import laravel from 'laravel-vite-plugin';

export default defineConfig( UserConfig config: {
    plugins: [
        laravel( String / PluginConfig config: {
            input: [
                'resources/css/app.css',
                'resources/js/app.js',
            ],
            refresh: true,
        }),
    ],
    resolve: {
        alias: {
            '@': '/resources/js',
        },
    },
    build: {
        onError: (error) => {
            console.error( message: 'Error de construcción:', optionalParams[0]: error);
        },
        onWarn: (warning) => {
            console.warn( message: 'Advertencia de construcción:', optionalParams[0]: warning
        },
    },
});
```

5.1.5 Base de Datos

La base de datos de *Hairbooker* está compuesta por cinco tablas diseñadas específicamente para el proyecto, además de otras tablas proporcionadas y utilizadas por el framework Laravel. Las tablas propias incluyen aquellas destinadas a la gestión de usuarios, citas, peluquerías, servicios y relaciones entre ellas, mientras que las tablas del framework se encargan de funcionalidades como la autenticación, las notificaciones o el manejo de las migraciones.

Las tablas propias creadas para el proyecto son las siguientes:

- **Users:** tabla para almacenar la información de los usuarios.

hairbooker users	
⌚	id : bigint(20) unsigned
🔑	dni : varchar(255)
👤	name : varchar(255)
🚹	sex : enum('masculino','femenino')
✉	email : varchar(255)
📞	phone : varchar(255)
🔒	password : varchar(255)
👤	role : enum('cliente','propietario')
🖼	avatar : varchar(255)
🕒	email_verified_at : timestamp
🔗	remember_token : varchar(100)
🕒	created_at : timestamp
🕒	updated_at : timestamp

- **Hairdressers:** tabla para almacenar información sobre las peluquerías. Las peluquerías estarán conectadas a la tabla users mediante su clave primaria.

		hairbooker hairdressers
		<code>id : bigint(20) unsigned</code>
		<code>cif : varchar(255)</code>
		<code>name : varchar(255)</code>
		<code>address : varchar(255)</code>
		<code>phone : varchar(255)</code>
		<code>owner_id : bigint(20) unsigned</code>
		<code>created_at : timestamp</code>
		<code>updated_at : timestamp</code>

- **Services:** tabla para almacenar información sobre los servicios ofertados en cada peluquería. Los servicios se vincularán con la tabla hairdresser a través del id de esta.

		hairbooker services
		<code>id : bigint(20) unsigned</code>
		<code>name : varchar(255)</code>
		<code>description : text</code>
		<code>price : decimal(8,2)</code>
		<code>hairdresser_id : bigint(20) unsigned</code>
		<code>created_at : timestamp</code>
		<code>updated_at : timestamp</code>

- **Appointments:** tabla que almacena la información relacionada con las citas de los clientes en una peluquería. Estas citas estarán enlazadas con las tablas users, services y hairdressers a través de sus respectivos identificadores (*id*). En el caso de la tabla users, la relación estará condicionada al rol del usuario, el cual deberá ser específicamente cliente.

	hairbooker appointments
⌚	id : bigint(20) unsigned
#	client_id : bigint(20) unsigned
⌚	unregistered_client : varchar(80)
#	service_id : bigint(20) unsigned
#	hairdresser_id : bigint(20) unsigned
⌚	start : datetime
⌚	end : datetime
⌚	status : enum('pendiente','confirmado','cancelado')
⌚	created_at : timestamp
⌚	updated_at : timestamp

- **Signups:** tabla destinada a almacenar información sobre los registros de los clientes en las peluquerías. Está vinculada con la tabla users mediante el identificador (*id*) de los usuarios cuyo rol es cliente, y con la tabla hairdressers a través de su correspondiente identificador.

	hairbooker signups
⌚	id : bigint(20) unsigned
⌚	client_id : bigint(20) unsigned
⌚	hairdresser_id : bigint(20) unsigned
⌚	created_at : timestamp
⌚	updated_at : timestamp

Respecto a las tablas de Laravel, son las siguientes:

- **Migrations:** tabla que almacena información sobre las migraciones ejecutadas por Laravel.

v hairbooker migrations	
id	: int(10) unsigned
migration	: varchar(255)
batch	: int(11)

- **Notifications:** tabla destinada a almacenar información sobre las notificaciones a los usuarios de las peluquerías.

v hairbooker notifications	
id	: char(36)
type	: varchar(255)
notifiable_type	: varchar(255)
notifiable_id	: bigint(20) unsigned
data	: text
read_at	: timestamp
created_at	: timestamp
updated_at	: timestamp

- **Password_reset_tokens:** tabla que sirve para almacenar información sobre el restablecimiento de los tokens de los usuarios.

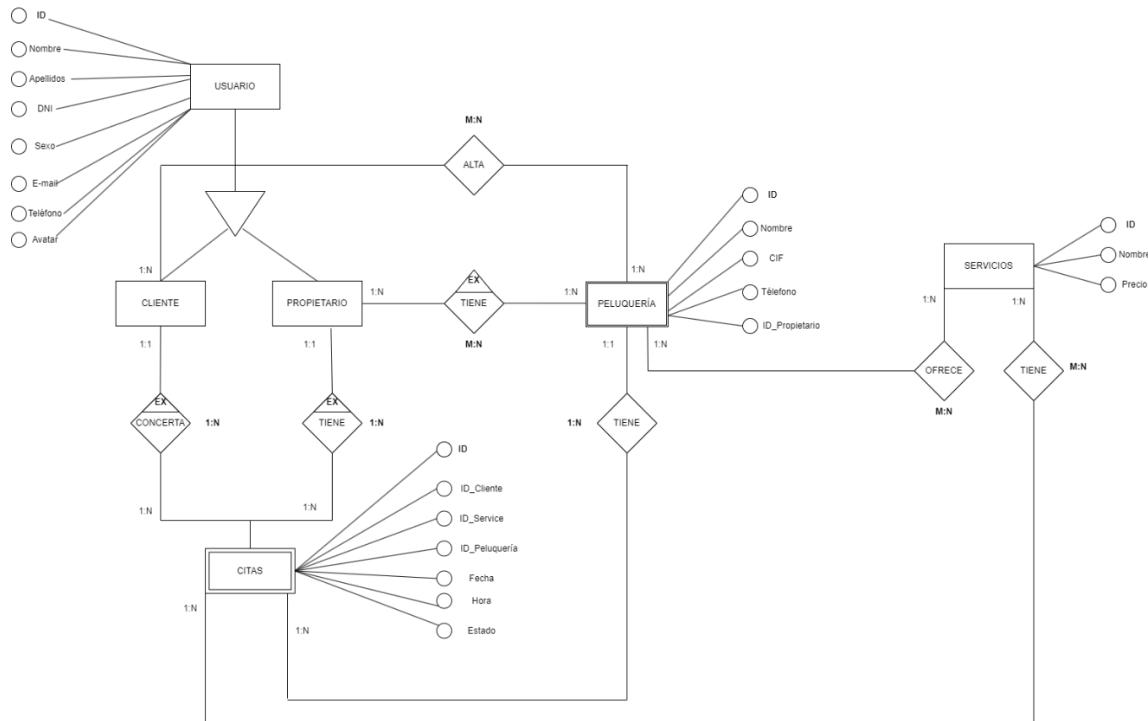
v hairbooker password_reset_tokens	
email	: varchar(255)
token	: varchar(255)
created_at	: timestamp

- **Password_access_tokens:** tabla que sirve para almacenar información sobre los tokens de acceso de los usuarios.

hairbooker personal_access_tokens	
id :	bigint(20) unsigned
tokenable_type :	varchar(255)
# tokenable_id :	bigint(20) unsigned
name :	varchar(255)
token :	varchar(64)
abilities :	text
last_used_at :	timestamp
expires_at :	timestamp
created_at :	timestamp
updated_at :	timestamp

El diagrama entidad-relación (ER) de la base de datos de Hairbooker representa de manera visual las tablas que conforman el sistema, junto con las relaciones existentes entre ellas. Este diagrama refleja la estructura lógica del modelo de datos, destacando las entidades principales como usuarios, peluquerías, servicios y citas.

Figura 36. Diagrama Entidad - Relación de Hairbooker



5.2 Documentación de usuario

Hairbooker es una aplicación web orientada a la gestión eficiente de citas en peluquerías y barberías. Su objetivo principal es ofrecer una plataforma sencilla e intuitiva donde los usuarios, tanto clientes como propietarios de peluquerías, puedan gestionar sus citas de manera ágil. Los clientes podrán realizar reservas, modificar o cancelar citas, mientras que los propietarios podrán administrar sus servicios, gestionar las citas de los clientes y mantener contacto con ellos a través de la plataforma.

5.2.1 Requisitos previos

Para poder utilizar *Hairbooker* se necesita disponer de un navegador. Los recomendados son Google Chrome, Opera y Microsoft Edge. Además, se necesitará disponer de una conexión estable y permanente a Internet para acceder a la aplicación.

Respecto a los dispositivos compatibles, *Hairbooker* es compatible con ordenadores, tabletas y dispositivos móviles.

5.2.2 Guía de usuario

Para cualquier tipo de usuario a la hora de entrar a *Hairbooker* es realizar el registro a la web. Para ello, debemos ir a la pantalla de registro e introducir los datos solicitados. El registro se realizará a partir del e-mail del usuario.

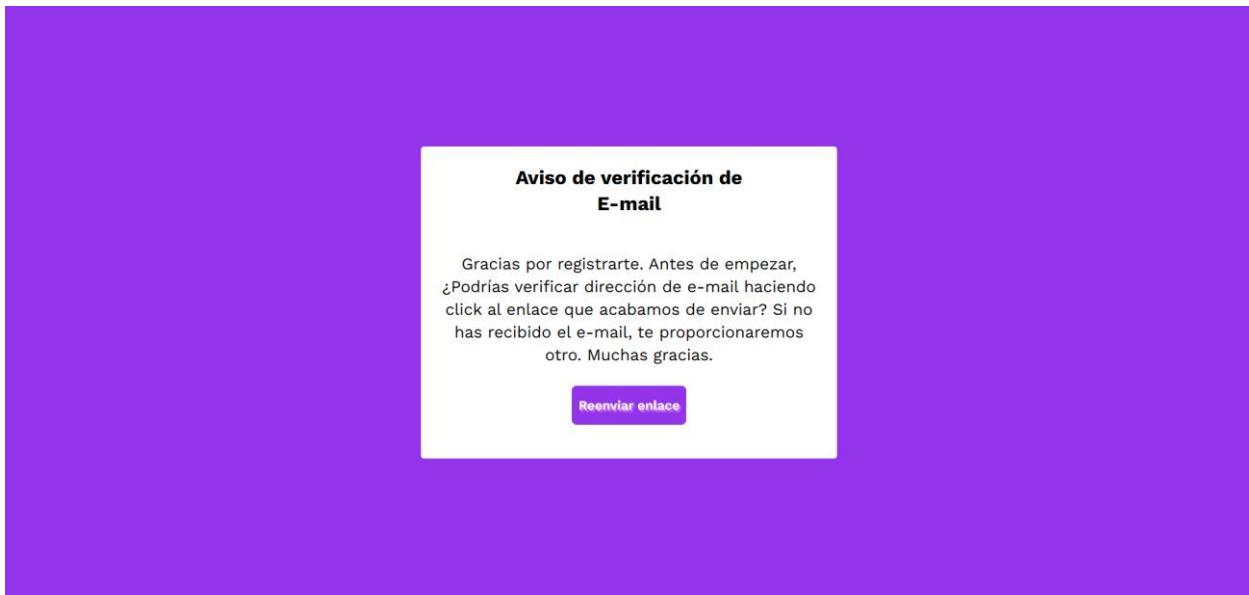
Figura 37. Registro de usuario

Registro

Nombre	<input type="text" value="Francisco Garcia"/>	
Correcto		
DNI	<input type="text" value="06292184T"/>	
Correcto		
Sexo	<input type="text" value="Hombre"/>	
Rol	<input type="text" value="Propietario"/>	
E-mail	<input type="text" value="francisco.garcia@outlook.com"/>	
Correcto		
Teléfono	<input type="text" value="665613317"/>	
Correcto		
Contraseña	<input type="password" value="....."/>	 
Correcto		
Repite la contraseña	<input type="password" value="....."/>	 
Correcto		
Registrarse		
¿Ya tienes una cuenta? Iniciar sesión		

Una vez introducidos los datos, el usuario deberá verificar en su email el registro (para simular la verificación se ha utilizar la herramienta mailtrap). Si el usuario no ha recibido en su email ningún enlace para la verificación, podrá realizar click en el botón para reenviar el enlace.

Figura 38. Mensaje para verificación de email y email recibido.



The screenshot shows the Mailtrap inbox interface. On the left is a sidebar with navigation links: Home, Email API/SMTP, Email Testing (selected), Email Marketing, Contacts, Sending Domains, Templates, Billing, Settings, and Help Center. The main area shows a list of received emails under 'Inboxes > My Inbox > Confirme su correo electrónico'. The first email is highlighted in blue and has a timestamp of '23 minutes ago'. Below the list are several other emails from 'HairBooker' and 'Santiago Rosado' with various subject lines like 'Confirme su correo electrónico' and 'Notificación de restablecimiento de contraseña'. To the right of the list is a preview pane titled 'Confirme su correo electrónico' showing the contents of the selected email. The preview includes the recipient's name ('Francisco García'), the message body ('¡Hola! Por favor, haga clic en el botón de abajo para verificar su dirección de correo electrónico.'), and a 'Confirme su correo electrónico' button. At the bottom of the preview pane, there is a note about accessibility and a 'Copiar' (Copy) button.

En el correo, una vez verificado el email, el usuario tendrá acceso al dashboard. En el dashboard las acciones se diferenciarán según el tipo de usuario.

Si el usuario olvida su contraseña al intentar iniciar sesión, podrá restablecerla haciendo clic en el enlace "¿Olvidaste tu contraseña?". Al seleccionar este enlace, se mostrará un formulario donde deberá ingresar su dirección de correo electrónico. Luego, se le enviará un enlace a su correo para proceder con el restablecimiento de la contraseña.

Figura 39. Proceso de restablecimiento de la contraseña.



The screenshot shows an email inbox from Mailtrap. A specific email titled 'Notificación de restablecimiento de contraseña' is selected. The email content is as follows:

Notificación de restablecimiento de contraseña

From: HairBooker <hbbookercompany@outlook.com>
To: <francisco.garcia@outlook.com> a few seconds ago

Confirmé su correo electrónico
to: <francisco.garcia@outlook.com> 22 minutes ago

Confirmé su correo electrónico
to: <francisco.garcia@outlook.com> an hour ago

Confirmé su correo electrónico
to: <marijose@outlook.es> 16 days ago

Notificación de restablecimiento de contraseña
to: <santiago.rosado@outlook.es> 16 days ago

Notificación de restablecimiento de contraseña
to: <santiago.rosado@outlook.es> 16 days ago

Confirmé su correo electrónico
to: <arturo@hotmail.com> 19 days ago

Confirmé su correo electrónico
to: <gury.ruiz@hotmail.com> a month ago

The email body includes a greeting '¡Hola!', a message about receiving the password reset request, and a 'Restablecer contraseña' button. It also states that the link will expire in 60 minutes and includes 'Saludos, Laravel'.



5.2.2.1 Cliente

En primer lugar, el cliente debe darse de alta en la peluquería que el quiera para poder ver las citas de esta y gestionar las suyas propias. Para ello, en la pantalla de “Darse de alta” en una peluquería, debe buscar en el seleccionable la que prefiera.

Figura 40. Selección de peluquería para darse de alta.

A screenshot of the HairBooker administration panel. On the left, a dark sidebar titled "Panel de administración" contains links for "Menu", "Perfil", "Mis citas", "Darse de alta en peluquería", and "Cerrar Sesión". The main content area has a yellow header with the "HAIRBOOKER" logo and a user icon. Below the header, a section titled "Date de alta en tu peluquería preferida" asks "Selecciona tu peluquería del siguiente listado" and shows a dropdown menu with options: "Selecciona peluquería", "BarberJuanjo" (which is selected), and "Juanjo Barberos". A separate section titled "Peluquerías a las que te has dado de alta" shows a box containing "Juanjo Barberos".

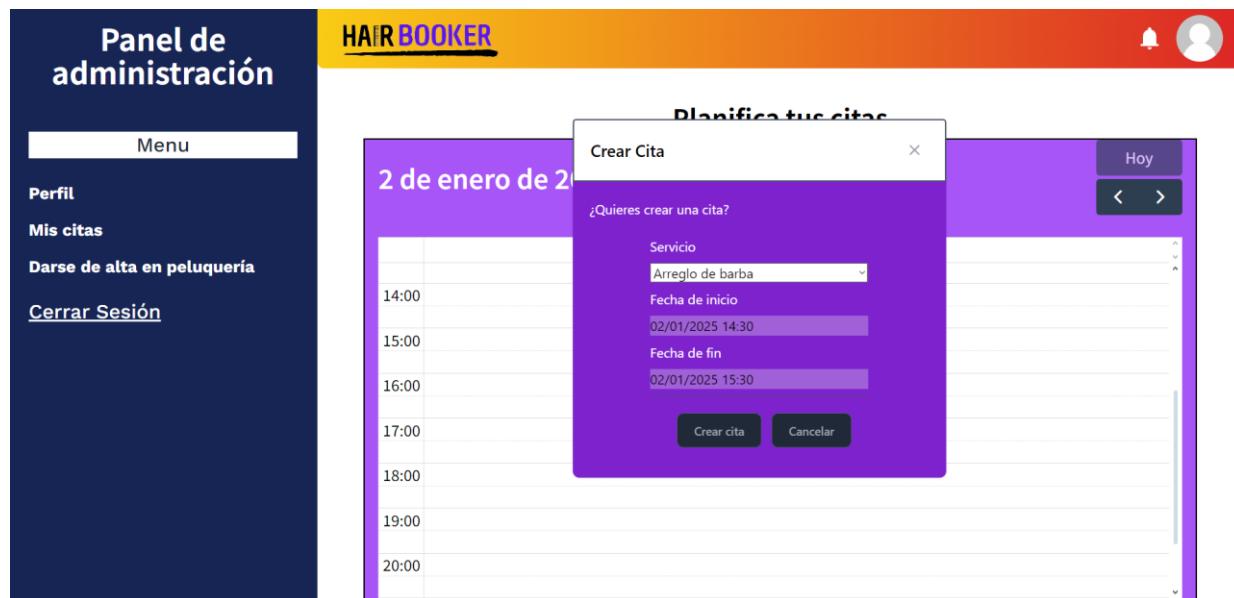
Una vez registrado, el usuario podrá gestionar sus citas en la sección "Mis citas". En este apartado, podrá visualizar las citas de la peluquería que ha

seleccionado a través de un calendario interactivo. El usuario tendrá la opción de hacer clic en los horarios disponibles donde aún no se hayan registrado citas, lo que le permitirá agendar nuevas citas de forma rápida y sencilla.

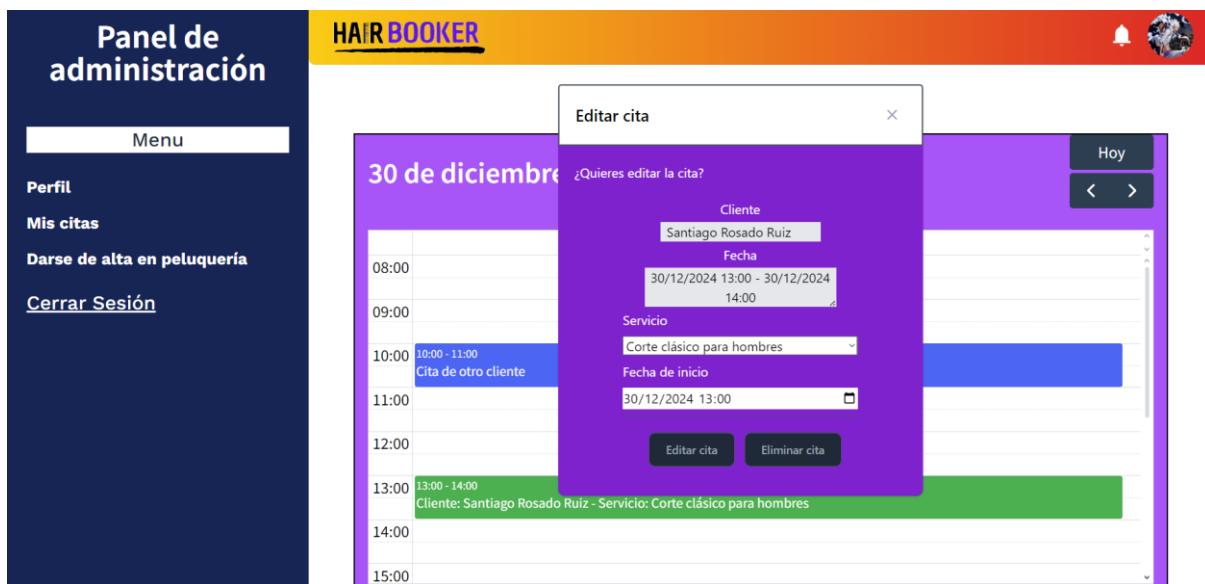
Las acciones que puede realizar el usuario cliente en el calendario son:

- **Creación de una cita:** el usuario cliente podrá reservar una cita haciendo click en las horas disponibles en el calendario. Se mostrará un formulario con los datos al introducir.

Figura 41. Reservar una cita.



- **Modificación de una cita:** el usuario cliente podrá modificar sus citas, pero no las de otros clientes. Para la modificación, el usuario debe hacer click en la cita deseada para mostrar el formulario de edición.

Figura 42. Modificación de una cita.

- **Eliminación de una cita:** el usuario cliente podrá eliminar cualquier cita haciendo clic sobre ella. Al hacerlo, se mostrará el mismo formulario utilizado para la modificación, incluyendo un botón de “Eliminar”. Al pulsar este botón, aparecerá un aviso de confirmación para proceder con la eliminación. Si el propietario confirma, la cita será eliminada de forma definitiva.

5.2.2.2 Propietario

En primer lugar, el propietario debe registrar su peluquería en la aplicación a través de la sección “Crear peluquería”. En esta sección, se completará un formulario con los datos necesarios para crear la peluquería.

Figura 43. Registro de la peluquería.

Panel de administración

HAIRBOOKER

Añade tu peluqueria

Nombre	Urban Barber	Correcto
CIF	B12345353	Correcto
Télefono	976221522	Correcto
Dirección	Calle Mayor, 15, 13720, Madrid	Correcto

En segundo lugar, una vez registrada la peluquería, el propietario podrá añadir los servicios que ofrece en la sección “Gestión de Servicios”. Al no tener servicios disponibles, se mostrará en la pantalla un mensaje de que no existen servicios. El usuario podrá crear servicios haciendo click en el botón de “Crear”. A continuación, se le mostrara al usuario un formulario para el registro del servicio.

Figura 44. Gestión de servicios de la peluquería.

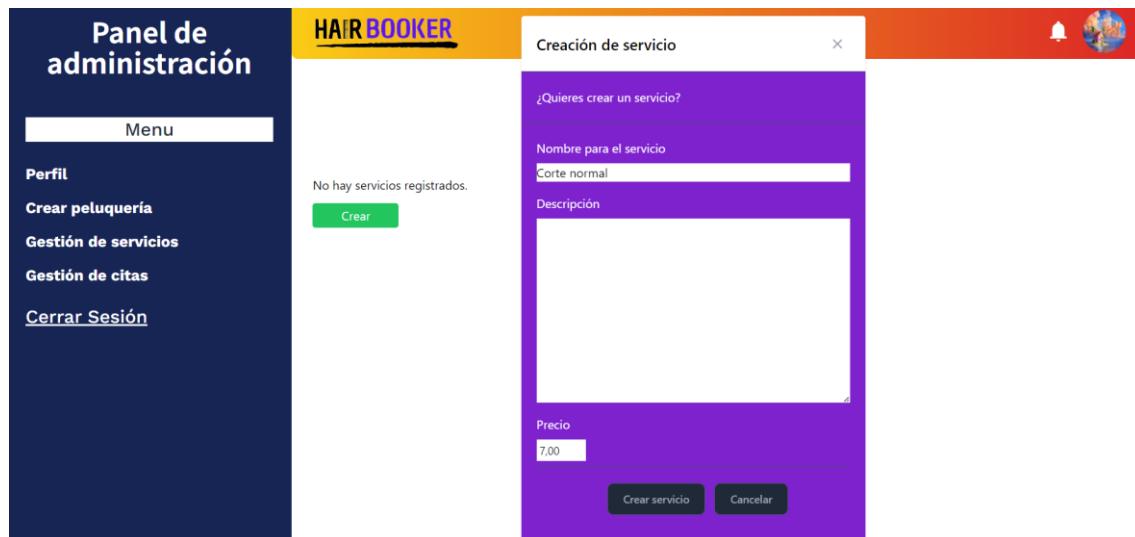
Panel de administración

HAIRBOOKER

Seleciona una peluquería a la que quieras añadir servicios

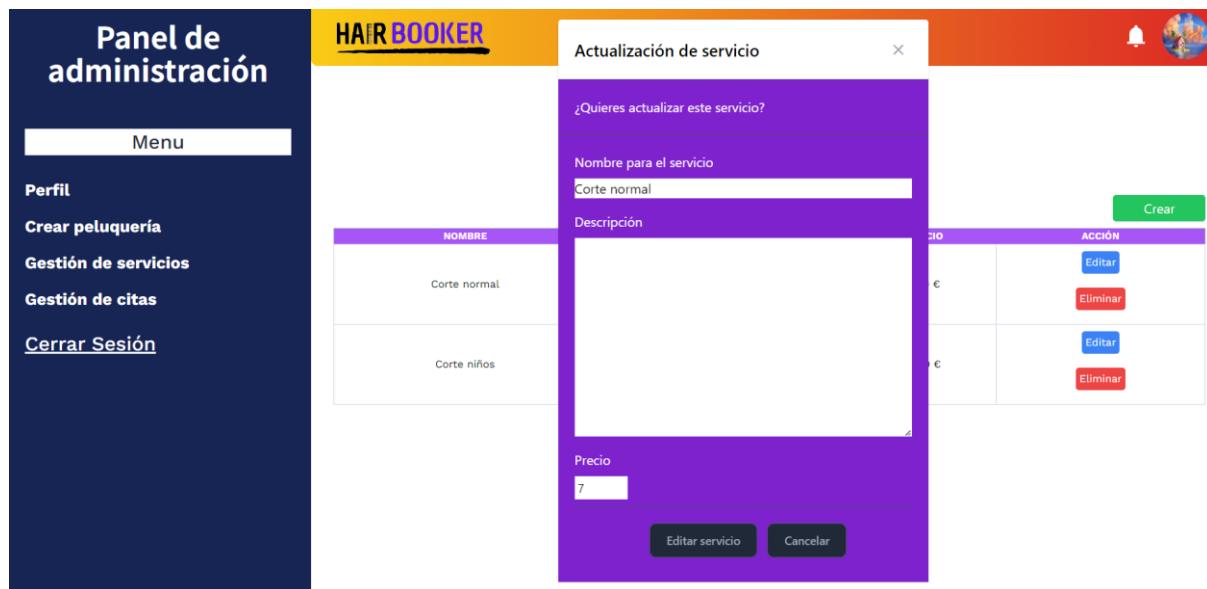
Selecciona peluquería
Selecciona peluquería
BarberJuanjo
Juanjo Barberos
Urban Barber

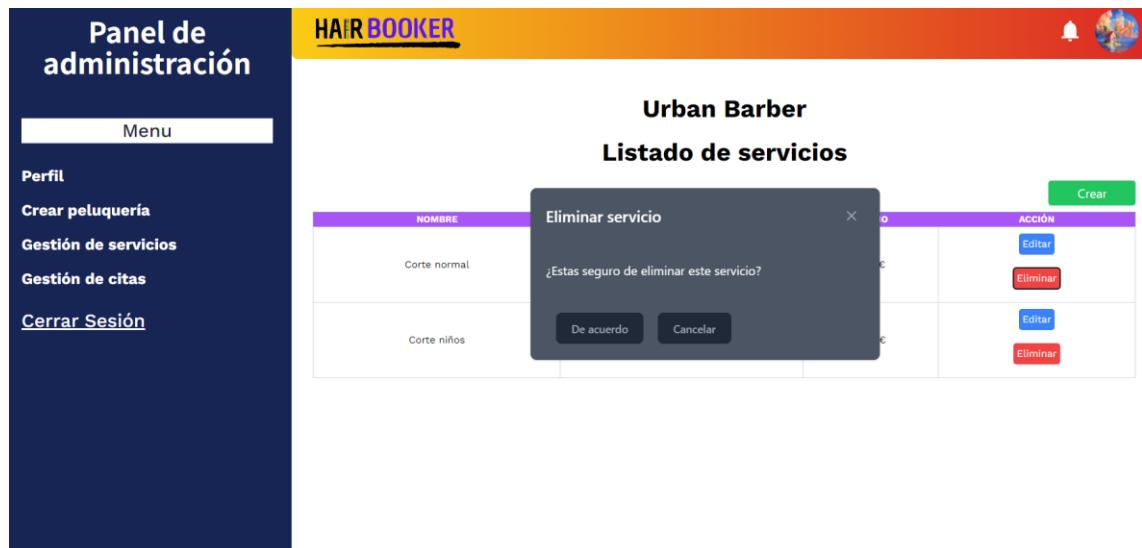
Documentación - HairBooker



Una vez añadido, se mostrará una tabla al usuario donde haciendo click en el servicio, podrá editar o eliminar el servicio.

Figura 45. Edición y eliminación del registro.





Una vez realizada la creación de los servicios, a través de la sección “Gestión de citas”, el usuario-propietario, podrá gestionar sus citas haciendo click en la sección “Gestión de Citas”. Aquí podrá gestionar todas las citas de sus clientes a través del calendario interactivo. Las acciones que puede realizar el usuario en el calendario son las siguientes:

- **Creación de una cita:** el usuario propietario tendrá la posibilidad de interactuar con las horas visibles en el calendario. Al hacer clic en una hora específica, se desplegará un formulario diseñado para añadir una nueva cita. A diferencia del cliente, el propietario podrá seleccionar entre un cliente registrado en su peluquería o agregar manualmente la información de un cliente que no esté dado de alta. Asimismo, el propietario tendrá la opción de definir el estado de la cita durante su creación.

Figura 46. Creación de una cita.

The figure consists of two screenshots of the HairBooker application. The top screenshot shows the main dashboard with a sidebar titled 'Panel de administración' containing 'Menu', 'Perfil', 'Crear peluquería', 'Gestión de servicios', 'Gestión de citas', and 'Cerrar Sesión'. The main area is titled 'HAIRBOOKER' and has a sub-section 'Elige una peluquería para gestionar tus citas' with a dropdown menu showing 'Selecciona peluquería', 'BarberJuanjo', 'Juanjo Barberos', and 'Urban Barber' (which is highlighted). A note at the bottom says 'Recuerda crear servicios para que los clientes puedan escoger los servicios de tu peluquería.' The bottom screenshot shows a modal window titled 'Crear Cita' with the question '¿Quieres crear una cita?'. It includes fields for 'Cliente' (set to 'No hay clientes disponibles' with an 'Añadir manualmente' button), 'Servicio' (set to 'Corte normal'), 'Fecha de inicio' (set to '02/01/2025 08:00'), 'Fecha de fin' (set to '02/01/2025 09:00'), and 'Estado' (set to 'Confirmado'). The background shows a calendar for January 2nd, 2025, with time slots from 08:00 to 15:00. Buttons for 'Crear cita' and 'Cancelar' are at the bottom of the modal.

- **Modificación de una cita:** el usuario propietario tendrá la capacidad de modificar cualquier cita perteneciente a los clientes de su peluquería, a diferencia del usuario cliente, quien únicamente podrá realizar cambios en sus propias citas y, además, podrá modificar el estado de las citas. Para su modificación, se debe hacer click sobre la cita y se mostrará un formulario con los datos de la cita para su actualización.

Figura 47. Modificación de una cita.

- **Eliminación de una cita:** El usuario propietario podrá eliminar cualquier cita haciendo clic sobre ella. Al hacerlo, se mostrará el mismo formulario utilizado para la modificación, incluyendo un botón de “Eliminar”. Al pulsar este botón, aparecerá un aviso de confirmación para proceder con la eliminación. Si el propietario confirma, la cita será eliminada de forma definitiva.

5.3 Manual de instalación y configuración

Hairbooker no requiere instalación ni configuración adicional, salvo el uso de *Mailtrap*, un servidor SMTP falso que permite probar y visualizar los correos electrónicos enviados.

El proyecto se encuentra desplegado en la plataforma de servicios en la nube *Heroku* y es accesible a través del siguiente enlace: <https://hairbooker-b4b3513484f2.herokuapp.com>.

Para probar la funcionalidad de verificación de correo electrónico, necesaria para utilizar la web, se emplea *Mailtrap*. Se ha creado una cuenta específica para facilitar el acceso y realizar las pruebas de verificación de correo o restablecimiento de contraseña.

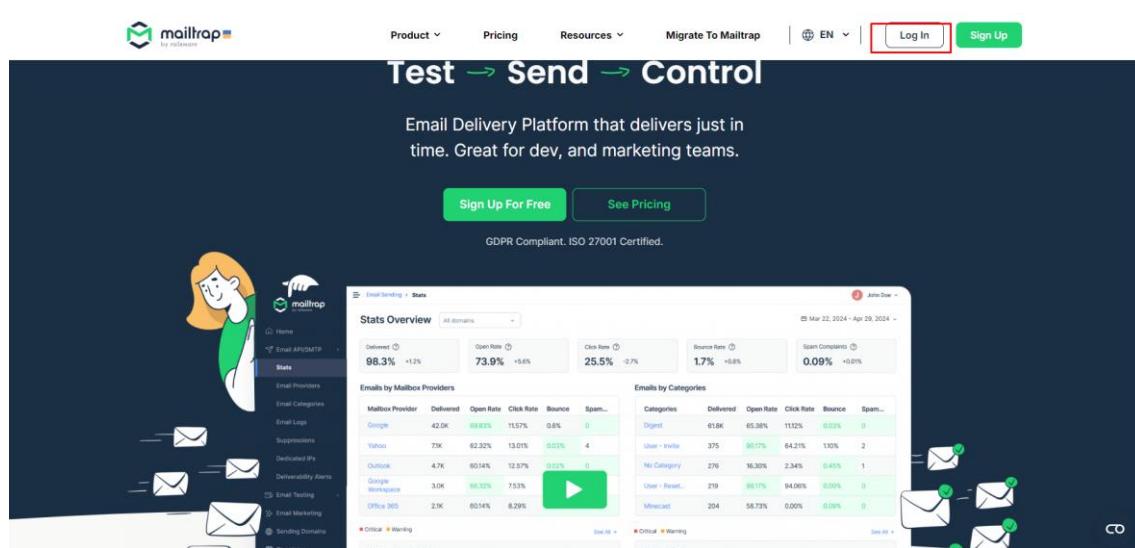
El procedimiento es el siguiente:

- Acceder a la web de Mailtrap: <https://mailtrap.io>.
- Iniciar sesión con las credenciales proporcionadas:
 - **Usuario:** hbookercompany@gmail.com
 - **Contraseña:** hairbooker

Una vez dentro, dirigirse al panel de Inbox, donde se recibirán todos los correos electrónicos enviados desde *Hairbooker*.

En las imágenes a continuación, se detalla todo el proceso para mayor claridad.

Figura 48. Inicio de sesión en *Mailtrap*.



Documentación - HairBooker

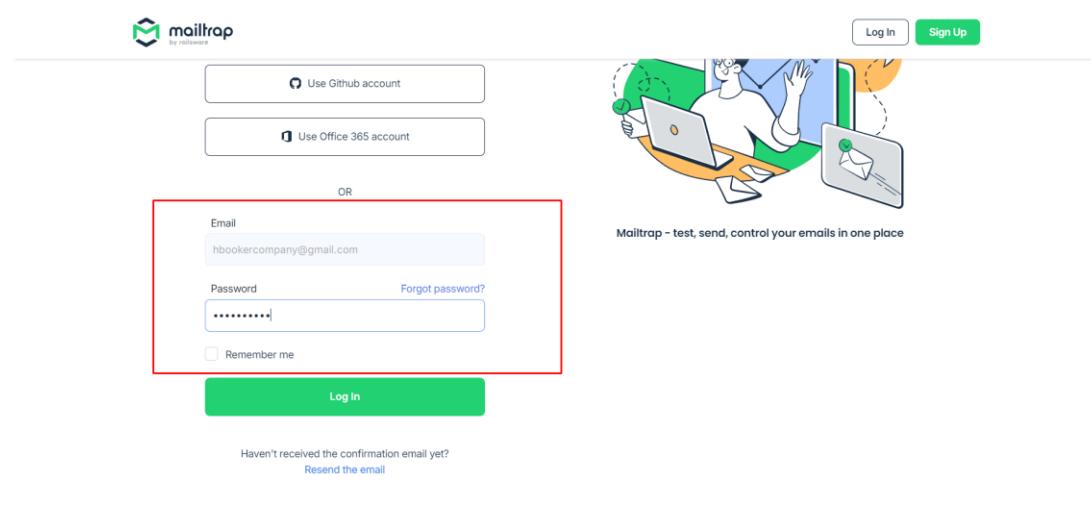


Figura 49. Acceso a los e-mails en *Mailtrap*.

Email Testing
Test Emails on Staging
Recent inbox Sent
My inbox 8 minutes ago

Email API/SMTP
Send emails on production via SMTP or API
Start sending >
Real SMTP Server
SPF, DKIM, DMARC, Dedicated IP
Deliverability Alerts

Email Campaigns
Deliver newsletters, promotional offers, and other types of marketing campaigns
Start campaign >
No-code email builder
Scheduled Sending
Contacts management
Performance tracking

Inboxes

Notificación de restablecimiento de contraseña

From: HairBooker <hbookercompany@outlook.com>
To: <arturo@hotmail.com>

Este enlace de restablecimiento de contraseña expirará en 60 minutos.
Si no ha solicitado el restablecimiento de contraseña, omita este mensaje de correo electrónico.
Saludos,
hairbooker

Si está teniendo problemas al hacer clic en el botón "Restablecer contraseña", copie y pegue la URL de abajo en su navegador web: <https://hairbooker-b6b3513.a0af2.beefcannon.net/confirm/62e0407f1-0771-474c-804f-4ac0nn7c4>

5.3.1 Instalación y configuración en local

5.3.1.1 Requerimientos previos

Para la instalación en local de Hairbooker es necesario los siguientes requisitos:

- **XAMPP:** <https://www.apachefriends.org/es/index.html>
- **Composer:** <https://getcomposer.org/download/>.
- **Node.js:** <https://nodejs.org/en/download/>.
- **Git:** <https://git-scm.com/downloads>.

5.3.1.2 Clonar el repositorio

Los pasos para clonar el repositorio en XAMPP son los siguientes:

1. Abre el terminal del sistema.
2. Navega a la carpeta htdocs de XAMPP -> cd /ruta/a/xampp/htdocs
3. Clona el repositorio desde GitHub con el siguiente comando -> git clone https://github.com/Santyr96/repositorioRosado.git

5.3.1.3 Configuración del entorno

Para la configuración del entorno, sigue los siguientes pasos:

- En tu editor de texto o de forma manual, duplica el archivo .env.example y renómbralo a .env.
- Abre el archivo .env y configura las siguientes variables:
 - DB_DATABASE: Nombre de tu base de datos (por ejemplo, hairbooker).

- DB_USERNAME: Usuario de la base de datos (por ejemplo, root).
 - DB_PASSWORD: Contraseña del usuario de la base de datos (por ejemplo, sin contraseña en XAMPP).
 - APP_URL: Cambia a http://localhost/HairBooker.
- General la clave de la aplicación con el siguiente comando -> php artisan key:generate

5.3.1.4 Instalación de dependencias

Dentro del proyecto, en la terminal, instala las dependencias de PHP con el comando -> composer install.

Posteriormente, instala las dependencias de JavaScript con el comando -> npm install.

5.3.1.5 Configuración de Apache

Sigue los siguientes pasos para configurar el servidor de Apache:

- Abre el archivo httpd.conf y descomenta la siguiente línea: #Include conf/extra/httpd-vhosts.conf. Posteriormente, busca la línea de DocumentRoot y modificala con este contenido:

```
DocumentRoot "C:/xampp/htdocs/repositorioSantiago/HairBooker/public"
```

```
<Directory " C:/xampp/htdocs/repositorioSantiago/HairBooker/public ">
```

```
    Options Indexes FollowSymLinks
```

```
    AllowOverride All
```

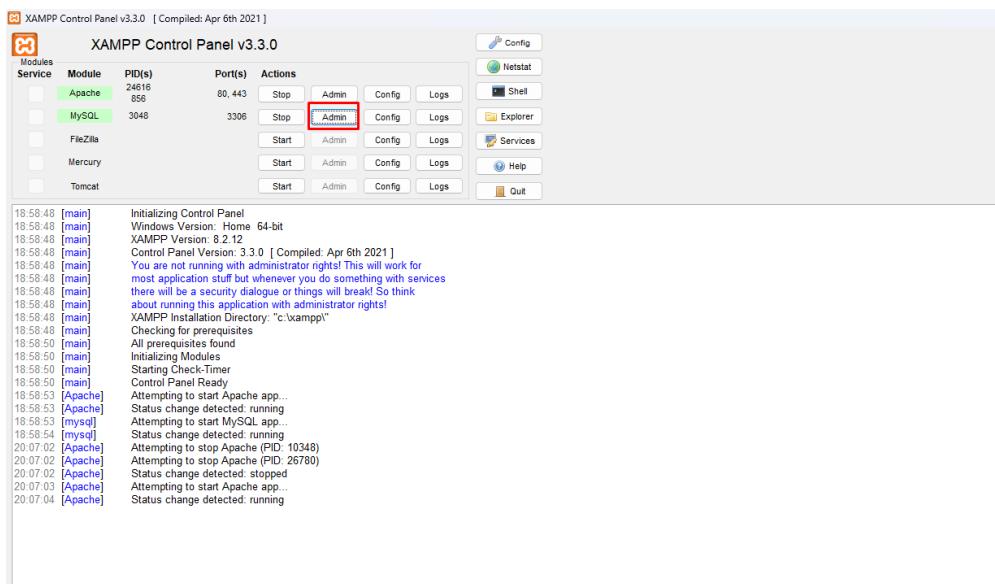
```
    Require all granted
```

```
</Directory>
```

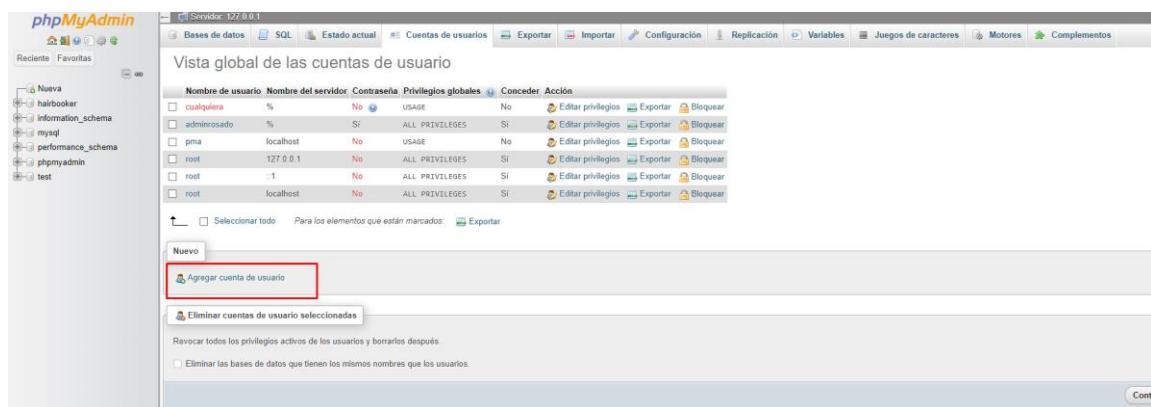
- Abre el archivo host ubicado en -> C:\Windows\System32\drivers\etc\hosts en Windows, en otros sistemas en /etc/hosts y agrega: **127.0.0.1 hairbooker**

5.3.1.6 Configuración de la base de datos

- Acceso a phpMyAdmin desde el panel de XAMPP:



- Accedemos a la pestaña usuarios:



- Ingresamos nombre de usuario y contraseña que queramos. En el host, seleccionamos localhost y agregamos todos los privilegios,

Este usuario y contraseña será el que indiquemos en el archivo .env descrito anteriormente.

Posteriormente, crea una nueva base de datos con el nombre hairbooker y dentro del proyecto, dentro de la carpeta Hairbooker, ejecuta las migraciones y los datos preparados con el siguiente comando -> php artisan migrate --seed.

5.3.1.7 Abrir la web

Ejecuta el servidor apache y la base de datos desde el panel de xampp:

The screenshot shows the XAMPP Control Panel interface. At the top, it displays the version "XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]". Below this is a table with columns: Service, Module, PID(s), Port(s), and Actions. The Actions column contains four buttons: Start, Admin, Config, and Logs. The "Start" button for both Apache and MySQL is highlighted with a red box. To the right of the table are several icons: Config, Netstat, Shell, Explorer, Services, Help, and Quit. The bottom half of the screen is a log window displaying system and application logs. Key entries include:

```

18:58:48 [main] Initializing Control Panel
18:58:48 [main] Windows Version: Home 64-bit
18:58:48 [main] XAMPP Version: 8.2.12
18:58:48 [main] Control Panel Version: 3.3.0 [ Compiled: Apr 6th 2021 ]
18:58:48 [main] You are not running with administrator rights! This will work for
most application stuff but whenever you do something with services
there will be a security dialogue or things will break! So think
about running this application with administrator rights!
18:58:48 [main] XAMPP Installation Directory: "c:\xampp\"
18:58:48 [main] Checking for prerequisites
18:58:50 [main] All prerequisites found
18:58:50 [main] Initializing Modules
18:58:50 [main] Starting Check-Timer
18:58:50 [main] Control Panel Ready
18:58:53 [Apache] Attempting to start Apache app...
18:58:53 [Apache] Status change detected: running
18:58:53 [mysql] Attempting to start MySQL app...
18:58:54 [mysql] Status change detected: running
20:07:02 [Apache] Attempting to stop Apache (PID: 10348)
20:07:02 [Apache] Attempting to stop Apache (PID: 26780)
20:07:02 [Apache] Status change detected: stopped
20:07:03 [Apache] Attempting to start Apache app...
20:07:04 [Apache] Status change detected: running
20:45:57 [Apache] Attempting to stop Apache (PID: 24616)
20:45:57 [Apache] Attempting to stop Apache (PID: 856)
20:45:57 [Apache] Status change detected: stopped
20:46:07 [mysql] Attempting to stop MySQL app...
20:46:07 [mysql] Status change detected: stopped

```

Abre el navegador y accede a <http://hairbooker>.

6 MANTENIMIENTO Y EVOLUCIÓN

6.1 Plan de mantenimiento y soporte

El plan de mantenimiento tiene como objetivo garantizar la funcionalidad y estabilidad del proyecto a lo largo del tiempo.

Para ello, se centrará la atención en diversas acciones para prevenir la aparición de problemas futuros a partir de estas acciones:

- **Actualización:** actualización periódica de las herramientas y dependencias utilizadas.
- **Revisiones:** revisiones de seguridad y rendimiento de manera trimestral para detectar posibles errores que supongan un riesgo para la plataforma.
- **Optimización:** estudio y análisis para el desarrollo de optimización del código y de la base de datos.
- **Usabilidad:** involucrar a usuarios para probar la plataforma y detectar problemas que hagan reducir la usabilidad de la web.

Una vez aparezcan los problemas es importante realizar acciones para acometer los errores que surjan a lo largo del tiempo de vida de la plataforma. Estas acciones son las siguientes:

- **Monitorización:** monitorizar los posibles errores que puedan surgir, tanto en el *backend* como en el *frontend* para establecer una prioridad a cada uno según su impacto en la funcionalidad.
- **Análisis de errores:** analizar los logs del servidor para lograr identificar el problema desde la raíz.
- **Solución de vulnerabilidades en la seguridad:** revisar configuraciones de autenticaciones en casos de se produzcan accesos indebidos.
- **Problemas reportados por el cliente:** ofrecer herramientas de comunicación a los clientes para que puedan reportar la aparición de problemas.

- **Documentación y mejora:** documentar todos los errores y sus soluciones para poder aplicar medidas preventivas para evitar que el problema vuelva a aparecer.

6.2 Actualizaciones y mejoras futuras

Como cualquier producto, *Hairbooker* necesita evolucionar y mantenerse actualizado para consolidarse como una herramienta imprescindible para los profesionales de la peluquería. Su enfoque principal siempre ha sido ofrecer un servicio accesible y fácil de usar, orientado especialmente a pequeñas peluquerías que buscan una solución intuitiva para gestionar sus citas. Con esta visión en mente, se ha elaborado una lista de mejoras a implementar tanto a corto como a largo plazo, con el objetivo de seguir optimizando la experiencia del usuario y potenciar el crecimiento de la plataforma.

- **Notificaciones a través de email:** actualmente, *Hairbooker* incorpora un sistema de notificaciones accesible a través de la propia web. Cuando el usuario accede a su dashboard, puede visualizar las notificaciones directamente desde allí. Sin embargo, nuestro objetivo a futuro es implementar un sistema de notificaciones por correo electrónico, de modo que el usuario pueda recibir información relevante sin necesidad de acceder a la plataforma. Esto garantizará una comunicación más eficiente y mejorará la experiencia del usuario al mantenerlo informado en todo momento.
- **Estadísticas:** aunque la plataforma *Hairbooker* ha sido diseñada principalmente como un sistema de gestión de citas, es fundamental proporcionar información valiosa al usuario propietario sobre el desempeño de su negocio. Por esta razón, uno de los objetivos clave para futuras mejoras es incorporar una sección dedicada a estadísticas. Esta funcionalidad permitirá al usuario acceder a datos relevantes sobre su negocio, como el número de citas gestionadas, servicios más solicitados, horarios de mayor demanda, entre otros.

Con esta información, el propietario podrá tomar decisiones más informadas y estratégicas para optimizar la gestión de su peluquería.

- **Comentarios:** con el objetivo de mejorar la comunicación entre el cliente y el propietario, se tiene previsto implementar en el futuro una sección de comentarios u observaciones al momento de modificar una cita. Esta funcionalidad permitirá que, durante cualquier cambio en una cita, tanto el cliente como el propietario puedan añadir comentarios u observaciones sobre el motivo de dicha modificación.
- **Establecimiento de horario:** permitir que los propietarios puedan establecer los horarios de apertura y cierre de la peluquería. De esta manera, los clientes sabrán el horario para asignar sus citas.

7 CONCLUSIONES

7.1 Evaluación del proyecto

El proyecto *Hairbooker* se ha desarrollado cumpliendo los plazos establecidos y logrando los objetivos iniciales planteados. Entre estos objetivos destaca la facilidad de uso, que se ha traducido en una navegación intuitiva y simplificada. Además, se ha implementado soporte para múltiples roles, asegurando que cada usuario acceda exclusivamente a las funcionalidades destinadas a su perfil.

Sin embargo, durante el desarrollo se presentaron diversos desafíos, principalmente relacionados con la adopción de nuevas tecnologías como Laravel y la integración de FullCalendar en el proyecto. A pesar de estas dificultades, se lograron resolver de manera efectiva, permitiendo que el proyecto se completara dentro de las estimaciones previstas y manteniendo un enfoque en la calidad del resultado final.

Para evaluar el proyecto, es importante destacar cuáles son las fortalezas y debilidades del proyecto:

7.1.1 Fortalezas

Las fortalezas de la plataforma *Hairbooker* son las siguientes:

- **Enfoque en el usuario final:** desarrollo de una interfaz intuitiva, diseñada para diferentes usuarios y segmentación en roles.
- **Tecnologías modernas:** uso de herramientas como Laravel y Tailwind CSS que aportan en escalabilidad, rendimiento y mantenimiento del proyecto.
- **Seguridad:** la autenticación basada en roles, junto con las validaciones realizadas tanto en el lado del cliente como en el servidor, garantiza un alto nivel de seguridad para la plataforma.

7.1.2 Debilidades

Las debilidades de la plataforma *Hairbooker* son las siguientes:

- **Falta de funcionalidades:** algunas características previamente planificadas, como las estadísticas, no se han podido implementar por limitaciones del tiempo.
- **Falta de soporte para otras lenguas:** la plataforma no incluye opciones para otros idiomas que no sean español.
- **Limitaciones en la personalización:** los usuarios, especialmente los propietarios, no podrán especificar el horario de su peluquería.

7.2 Cumplimiento de objetivos y requisitos

El proyecto ha cumplido con la mayoría de los objetivos y requisitos planteados. Se implementó un sistema de gestión de citas que permite a los usuarios reservar, modificar y cancelar citas de forma sencilla. Además, se incluyó un sistema de multiautenticación basado en roles, lo que asegura que solo los usuarios autorizados accedan a cada funcionalidad. La interfaz es atractiva y

responsive, optimizada para ser utilizada en diferentes dispositivos. La estructura del proyecto también permite incorporar fácilmente nuevas funcionalidades según las necesidades del mercado.

Sin embargo, algunas funcionalidades, como la integración con pasarelas de pago, las notificaciones automáticas y las estadísticas, quedaron como objetivos a implementar en futuras iteraciones.

7.3 Lecciones aprendidas y recomendaciones para futuros proyectos

Durante el desarrollo de este proyecto me he enfrentado a varios problemas que, con una planificación adecuada, podrían haberse evitado. Una de las lecciones aprendidas en este proceso es la importancia de mejorar la planificación previa, tanto de la interfaz como del servidor. Esto permite anticipar posibles inconvenientes y mejorar la eficiencia del desarrollo.

En segundo lugar, he aprendido que la implementación de pruebas unitarias de manera paralela al desarrollo del proyecto es crucial. Al finalizar una parte o sección del proyecto, es importante implementar pruebas para reducir la cantidad de errores que puedan surgir más adelante, lo que facilita la detección temprana de fallos.

Por último, he comprendido la necesidad de mantenerse flexible y abierto al uso de nuevas tecnologías. Sin embargo, es esencial evaluar cuidadosamente la curva de aprendizaje de cada tecnología antes de integrarla, ya que esto puede afectar tanto el tiempo de desarrollo como la calidad del producto final.

8 BIBLIOGRAFÍA Y REFERENCIAS

PMO Informática. (2018). ¿Qué es un requerimiento funcional? PMO Informática.

<https://www.pmoinformatica.com/2018/05/que-es-requerimiento-funcional.html>

Interempresas. (2020). El negocio de las peluquerías: Un sector en auge.

Interempresas.net. <https://www.interempresas.net/Peluqueria/Articulos/508772-El-negocio-de-las-peluquerias-un-sector-en-auge.html>

Business Research Insights. (2023). Web Development Market - Industry Size, Share, Trends, Growth, and Forecast 2023. *Business Research Insights*.

<https://www.businessresearchinsights.com/es/market-reports/web-development-market-109039>

DunaSoft. (n.d.). DunaSoft: Software de gestión para peluquerías.

https://www.dunasoftpc.com/?utm_source=google&utm_medium=paid&utm_campaign=marca&gad_source=1&gclid=CjwKCAiAm-67BhBIEiwAEVftNjvnoJaobV13VZdpcluQSU5LchCX_EXCfpif5yGNur11cUs9sTUt_BoCcEIQAvD_BwE

Treatwell. (n.d.). Treatwell: Reserva de tratamientos de belleza y bienestar.

https://www.treatwell.es/?utm_source=google&utm_medium=cpc&utm_campaign_id=388815262&utm_campaign=ES_ES_Brand_Country_AllTreatments_Pure&utm_click_id=CjwKCAiAm-67BhBIEiwAEVftNh1VT4cIltR4P56oH2T1Y_locDiDVvMvSqNuFp137MF2bY1AZVOrNBoCQcwQAvD_BwE&gad_source=1&gclid=CjwKCAiAm-67BhBIEiwAEVftNh1VT4cIltR4P56oH2T1Y_locDiDVvMvSqNuFp137MF2bY1AZVOrNBoCQcwQAvD_BwE

Booksy. (n.d.). Booksy: Plataforma para la gestión de citas en peluquerías y salones

de belleza. https://booksy.com/es-es/#ba_s=seo

Mailtrap. (n.d.). Mailtrap: Prueba de correos electrónicos en entornos de desarrollo. <https://mailtrap.io>

Heroku. (n.d.). *Heroku: Plataforma como servicio para aplicaciones modernas.*

https://www.heroku.com/?utm_source=google&utm_medium=paid_search&utm_campaign=emea_heraw&utm_content=general-branded-search-rsa&utm_term=heroku&utm_source_platform=GoogleAds&gad_source=1&gclid=CjwKCAiA7Y28BhAnEiwAAAdOJUIQOiyLpAdvbMf6wPvOhKsqsls2R5J5xwUmyHUfjBjOCWBe-W1TMgRoCwNsQAvD_BwE