

PROYECTO PADEL APP SH | DAWSON



Contenido

PARTE 1 DEL PROYECTO	3
1. Descripción del producto y proyecto a realizar a continuación	3
2. Descripción del cliente a realizar el proyecto	3
3. Justificación del por qué del proyecto y tecnologías a usar	4
4. Estudio de mercado y desarrollos similares o usables a este proyecto	5
5. Justificación del desarrollo del proyecto	6
6. Análisis DAFO (Debilidades, Amenazas, Fortalezas y Oportunidades)	6
7. Tecnologías y recursos utilizados	7
8. Requisitos funcionales y no funcionales	9
9. Planificación del desarrollo del Proyecto	11
PARTE 2 DEL PROYECTO	12
10. Diagrama de casos de uso	12
11. DIAGRAMA DE CLASES	12
12. Diagrama de secuencia de la Aplicación (UML)	13
13. Desarrollo de interfaces (MockUp)	14
PARTE 3 DEL PROYECTO	15
14. Pruebas, Interfaz y rendimiento	15
Login	15
Register	16
Verify Email	16
Verify Email Link	17
Forgot Password	18
Home	20
Reserve	21
Reserve Store	21
Desarrollos a implementar	23
15. Memoria Económica	24
16. Licencia	24

PARTE 1 DEL PROYECTO

Documentación y Análisis previo del proyecto a realizar:

1. Descripción del producto y proyecto a realizar a continuación

El proyecto que desarrollaré en los días siguientes es una aplicación web de reservas para un club de pádel, en este caso la empresa se llama PÁDEL SH (PÁDEL SANTIAGO HERNÁNDEZ).

Aún existiendo empresas de reservas de pistas de pádel o de reservas de mesa de restaurantes adaptables a las pistas, mi desarrollo se distingue por una parte más local incluyendo métodos y funciones que gestionan de manera eficiente la automatización de las pistas de pádel.

La aplicación contará con varias funcionalidades claves, como el “Registro” y “Login” de cada usuario del club, el control de duraciones y horario de cada reserva de cada pista, contando con la capacidad de elegir la duración de este, así como, manteniendo informado a los usuarios a través del correo electrónico personalizado del club.

Aspectos a mejorar en la estabilidad a un futuro del desarrollo web sería creando una versión de aplicación a multiplataforma que permita un aumento de alcance a personas que no se lleven muy bien con la funcionalidad de una página web. Además de desarrollar ‘queries’ y seguridad para esta aplicación lo que ayudaría a la estabilidad del desarrollo. Como último sería la capacidad de paneles de control a los administrados en el que pudieran hacer los cambios que quisieran e incorporando informes estadísticos de funcionamiento.

Se intenta que este proyecto ayude a la eficiencia de gestión de las pistas de pádel y que proporcione una solución flexible que evolucione y se adapte a las necesidades de que cada cliente.

2. Descripción del cliente a realizar el proyecto

El cliente es una empresa que ofrece distintos servicios a sus clientes, en este caso el desarrollo se enfoca en la reserva de las 2 pistas ‘indoor’ de pádel simples, con la posibilidad de a futuro implementar mejores herramientas para gestionar este desarrollo y que se adapte a una solución flexible y automática.

La empresa opera a nivel local en la ciudad de Zaragoza, pero con posibilidad de abrir sus servicios nacionalmente.

3. Justificación del por qué del proyecto y tecnologías a usar

Este proyecto se eligió ya que el objetivo es mejorar en el desarrollo de automatizaciones y “api’s” así como generar visuales a través de métodos pocos usados y complicados.

La aplicación se enfocará sobre todo en el desarrollo de automatización sobre el horario de las pistas, es decir, un visual que permita al usuario elegir su reserva, aunque sea para el año que viene, ya que se ajusta a la base de datos a través de “fetch’s” en la “URL”. Mejorando así tanto la página web como el aumento de mi conocimiento a futuros proyectos.

Las tecnologías que iba a proceder a usar iban a ser un propio servidor local como “XAMPP”, en el que me proporciona un servidor ‘Apache’ y ‘MySQL’ para hacer pruebas de manera local y así no molestar a la parte de producción de la página oficial.

El cliente (en este caso el jurado del proyecto) debe verlo en producción por lo tanto debía proceder a realizar el proyecto en un sistema de producción local y un ‘framework’ ya visto que permita enseñar lo aprendido durante la formación, en este caso he usado para una producción local que luego tuvo que ser descartada y luego en producción:

- XAMPP: Permite crear una distribución fácil de instalar para desarrolladores que se están iniciando en el mundo de Apache y permite el sistema de gestión de bases de datos MySQL. Link: <https://www.apachefriends.org/es/index.html>
- Docker: Que es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. Link: <https://www.docker.com/>
- Laravel: Es un ‘framework’ de código abierto para desarrollar aplicaciones y servicios web con PHP. Su filosofía es desarrollar código PHP de forma elegante y simple. Link: <https://laravel.com/>
- Compose: es una herramienta para definir y ejecutar aplicaciones de Docker de varios contenedores. Se usa un archivo YAML para configurar los servicios de la aplicación. Link: <https://docs.docker.com/compose/>
- Artisan: es la interfaz de línea de comandos incluida con Laravel. Artisan existe en la raíz de su aplicación como ‘artisan script’ y proporciona una serie de comandos útiles que pueden ayudar mientras crea su aplicación. Link: <https://laravel.com/docs/11.x/artisan>

4. Estudio de mercado y desarrollos similares o usables a este proyecto

En este caso hacemos un pequeño estudio de los desarrollos que tienen otras empresas que pueden usar, pero con un coste mucho mayor y un acercamiento más frío. En este caso todas sirven para lo mismo, hacer reservas de pista/instalaciones de todo tipo.

- Playtomic: Aunque ‘Playtomic’ no es un desarrollo propio se podría implementar en este caso el cliente por un precio mensual/anual perdiendo beneficios ya que no es un desarrollo propio, como tal es el competidor más grande.

Link: <https://playtomic.io/>

- Padel Around: Está dirigida a la comunidad de jugadores de pádel y permite a quienes la descarguen que ubiquen las instalaciones más cercanas a su ubicación, localizar a otros jugadores que quieran disputar un partido. Es una aplicación multiplataforma, en este caso el desarrollo es muy antiguo pero cuenta con una comunidad bastante grande.

Link: <https://www.padeladdict.com/padel-around-app/>

- Sporttia: Sporttia es otra empresa como Playtomic que no es específica para el pádel pero si cuenta con un sistema de gestión para las reservas de cualquier actividad deportiva, en este caso es la más moderna de todas, y por tanto, la que más fijarse a como dirigen sus gestiones.

Link: <https://www.sporttia.com/>

Aunque estos desarrollos se enfocan en las reservas de actividades deportivas ninguna es como la reserva propia para un horario de un club, por lo tanto en mi caso aprenderé a como integrar zonas horarias para la reserva automática de pistas.

5. Justificación del desarrollo del proyecto

El auge del pádel como deporte de ocio y competitivo ha aumentado la demanda de instalaciones deportivas, y con ella la necesidad de herramientas que permitan una gestión eficaz de las reservas de pistas. Actualmente, muchos clubes deportivos dependen de sistemas manuales, reservas telefónicas o aplicaciones genéricas que no están optimizadas para las necesidades específicas de este deporte.

Objetivos del proyecto:

Desarrollar un sistema de reservas de pistas de pádel que sea fácil de gestionar para usuarios. El sistema es intuitivo, accesible desde dispositivos móviles u ordenadores y permite gestionar la disponibilidad en tiempo real para optimizar la ocupación de las pistas.

6. Análisis DAFO (Debilidades, Amenazas, Fortalezas y Oportunidades)

Debilidades:

- Requiere inversión inicial en desarrollo e infraestructura tecnológica.
- Posible resistencia al cambio o reacción reacia por parte de usuarios acostumbrados a métodos tradicionales.
- Necesidad de mantenimiento técnico y actualizaciones periódicas.

Amenazas:

- Competencia de aplicaciones ya establecidas en el mercado.
- Usuarios con poca experiencia en el uso de herramientas digitales pueden sentirse excluidos.
- Posibles problemas técnicos que afecten la confiabilidad del sistema (caídas del servidor, bugs, deprecateds...).

Fortalezas:

- Personalización para las necesidades específicas de la gestión de pistas de pádel.
- Facilidad de uso y diseño intuitivo que mejora la experiencia del usuario.
- Integración de notificaciones en tiempo real.

Oportunidades:

- Incremento del número de jugadores de pádel que buscan opciones más cómodas para reservar.
- Posibilidad de monetización adicional, como anuncios de eventos o servicios complementarios (alquiler de material, clases).
- Escalabilidad del sistema para expandirse a otros deportes o clubes deportivos.

7. Tecnologías y recursos utilizados

El desarrollo del sistema de reservas de pistas de pádel ha requerido la integración de diversas tecnologías y recursos, tanto a nivel de software como de hardware. A continuación, se detallan los principales componentes utilizados:

Lenguajes de Programación y Tecnologías Web

1. **PHP:** Lenguaje principal para el desarrollo del backend, responsable de la lógica del negocio, la interacción con la base de datos y la generación dinámica de las páginas web.
2. **SQL (MySQL o MariaDB):** Lenguaje utilizado para la creación, gestión y consulta de la base de datos, garantizando el almacenamiento eficiente y seguro de los datos relacionados con usuarios, reservas y disponibilidad de pistas.
3. **JavaScript:** Empleado en el frontend para añadir interactividad al sistema, como validación de formularios, actualizaciones dinámicas de la interfaz y notificaciones en tiempo real.
4. **HTML:** Lenguaje de marcado usado para estructurar las páginas del sistema, asegurando accesibilidad y compatibilidad en múltiples dispositivos.
5. **CSS:** Herramienta clave para el diseño y personalización de la interfaz de usuario, garantizando una experiencia atractiva e intuitiva.

Hardware Requerido

1. **Ordenador de Desarrollo:** Utilizado para el diseño, programación y prueba del sistema. Las especificaciones mínimas incluyen:
 - Procesador de 4 núcleos.
 - Memoria RAM de al menos 8 GB.
 - Espacio de almacenamiento SSD para acelerar la ejecución de tareas.
2. **Servidor de Producción:** Un servidor, con sistema operativo Linux, para alojar el sistema y garantizar accesibilidad en línea.

Sistema Operativo y Herramientas de Desarrollo

1. **Sistema Operativo Linux:** Elegido por su estabilidad, flexibilidad y robustez en entornos de desarrollo y producción.
2. **Visual Studio Code (VSCode):** Editor de código principal, destacado por su interfaz amigable, soporte de extensiones y depuración eficiente.

3. **Vim:** Editor de texto alternativo utilizado para modificaciones rápidas en el servidor directamente desde la terminal.
4. **Terminal de Linux:** Herramienta esencial para ejecutar comandos, realizar despliegues y gestionar configuraciones en el sistema operativo.

Otros Recursos y Software

1. **Navegadores Web (Firefox/Chrome):** Para realizar pruebas y garantizar compatibilidad del sistema en diferentes plataformas.
2. **Gestor de Versiones (Git):** Usado para el control de versiones y la colaboración en el desarrollo del proyecto en GitHub.
3. **Frameworks y Librerías Adicionales:**
 - **Bootstrap:** Para el diseño responsivo de la interfaz de usuario.
 - **jQuery:** Para facilitar tareas de JavaScript como manipulación del DOM y manejo de eventos.

Infraestructura y Servicios

1. **Servidor Web (Apache):** Configurado para servir las páginas del sistema de reservas.
2. **Base de Datos MySQL/MariaDB:** Instaladas en el servidor para gestionar la información del sistema.
3. **Servicios de Notificaciones:** Opcionalmente, integración con servicios para enviar alertas a los usuarios, como correos electrónicos.

8. Requisitos funcionales y no funcionales

Requisitos Funcionales

Estos requisitos describen las funcionalidades esenciales que el sistema debe cumplir para satisfacer las necesidades del usuario:

1. Gestión de Usuarios:

- Registro y login de usuarios.
- Permitir a los usuarios recuperar o cambiar sus contraseñas en caso de olvido.

2. Reserva de Pistas:

- Consultar disponibilidad de pistas en tiempo real, permitiendo búsquedas por fecha y hora.
- Realizar reservas de forma sencilla e intuitiva.
- Enviar notificaciones al usuario sobre el estado de sus reservas (confirmación y recordatorio).

3. Gestión de Horarios:

- Configuración de horarios disponibles.

4. Panel de Administración:

- Visualización de un calendario general con todas las reservas realizadas.

5. Control de Seguridad:

- Gestión de accesos con roles y permisos claros.
- Validación de datos de entrada para evitar errores o mal uso del sistema.
-

Requisitos No Funcionales

Los requisitos no funcionales definen las características de calidad y rendimiento del sistema:

1. Arquitectura Basada en MVC:

El sistema debe implementarse bajo el patrón **Modelo-Vista-Controlador (MVC)** para garantizar una separación clara de responsabilidades:

- **Modelo:** Gestiona la lógica de negocio y las interacciones con la base de datos, asegurando integridad y consistencia de los datos.
- **Vista:** Se encarga de presentar la interfaz de usuario, diseñada para ser intuitiva, responsiva y atractiva mediante HTML, CSS y JavaScript.
- **Controlador:** Actúa como intermediario, gestionando las solicitudes del usuario y llamando al modelo o la vista según sea necesario. Esta arquitectura facilita el mantenimiento, la escalabilidad y la reutilización del código, permitiendo futuras mejoras sin afectar el sistema en su conjunto.

2. Rendimiento y Escalabilidad:

- El sistema debe ser capaz de manejar múltiples usuarios simultáneamente sin afectar su rendimiento.
- La estructura de la base de datos y las consultas SQL deben estar optimizadas para minimizar los tiempos de respuesta.

3. Compatibilidad y Accesibilidad:

- El sistema debe ser compatible con los principales navegadores web (Chrome, Firefox, Edge).
- Debe adaptarse a dispositivos móviles, tabletas y ordenadores mediante un diseño 'responsive'.

4. Seguridad:

- Cifrado de contraseñas en la base de datos mediante algoritmos robustos como bcrypt.
- Prevención de vulnerabilidades como SQL Injection y Cross-Site Scripting (XSS) mediante validación de datos y uso de parámetros en consultas SQL (auth).

5. Usabilidad:

- El diseño de la interfaz debe ser claro y fácil de usar, con instrucciones y mensajes de error amigables para el usuario.

6. Mantenibilidad:

- Uso de un sistema de control de versiones (Git) para gestionar el desarrollo y los cambios en el proyecto.
- Código bien documentado e indexado, respetando los principios de “**Clean Code**” para facilitar futuras mejoras.

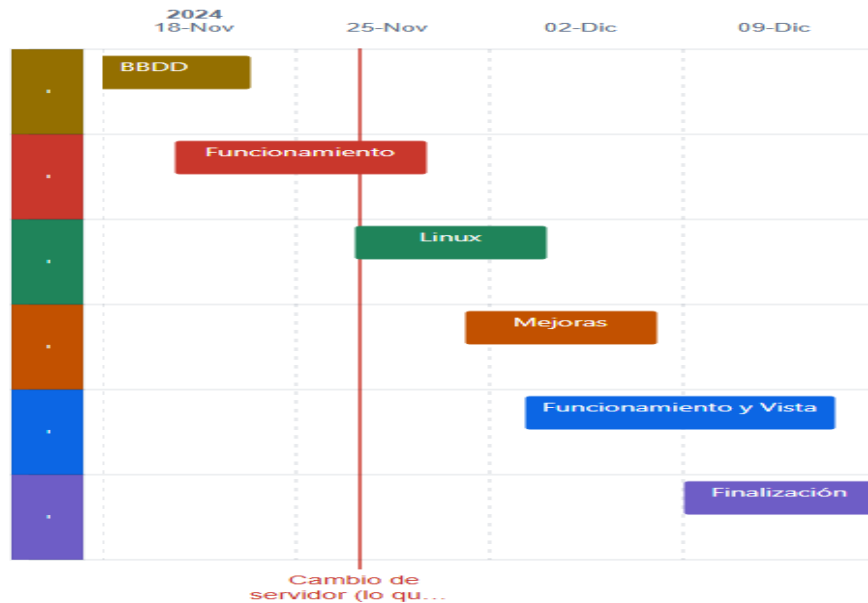
7. Disponibilidad y Fiabilidad:

- El sistema debe estar disponible al menos el 99% del tiempo, evitando caídas prolongadas que afecten la experiencia del usuario.
- Implementación de pruebas unitarias y funcionales para garantizar la estabilidad antes de cada despliegue.

8. Software Utilizado:

- Desarrollo en un entorno Linux para garantizar estabilidad, con herramientas como Visual Studio Code y Git para la colaboración y el control del proyecto.

9. Planificación del desarrollo del Proyecto



1. SEMANA 1:

- BDD: Básicamente es el pensamiento de como implementar los datos a lo que se requiere en el proyecto.
- Diseño de tablas, relaciones e implementación de campos y tipos.
- Empezar a configurar el Funcionamiento en local a través de PHP y definir la arquitectura del proyecto.

2. SEMANA 2:

- Empezar el funcionamiento de la aplicación, así como haciendo una interfaz intuitiva y fácil.
- Aspectos de la accesibilidad, garantizando el acceso a personas con diversas dificultades para entender sistemas complejos.

3. SEMANA 3:

- Aquí desgraciadamente se busca hacer el desarrollo en producción, es decir que debía configurar un entorno de desarrollo desde 0 para implementar lo que quería el "cliente" (el jurado) con lo aprendido en el curso.
- La configuración es complicada ya que el servicio de contenedores es frágil y necesita mucha paciencia ya que un fallo requiere volver a empezar.
- Mejoras del servidor terminadas, es decir, el servidor ya es totalmente útil y es posible agregar mejores de comprensión gracias a lo instalado, como el "PHP ARTISAN".

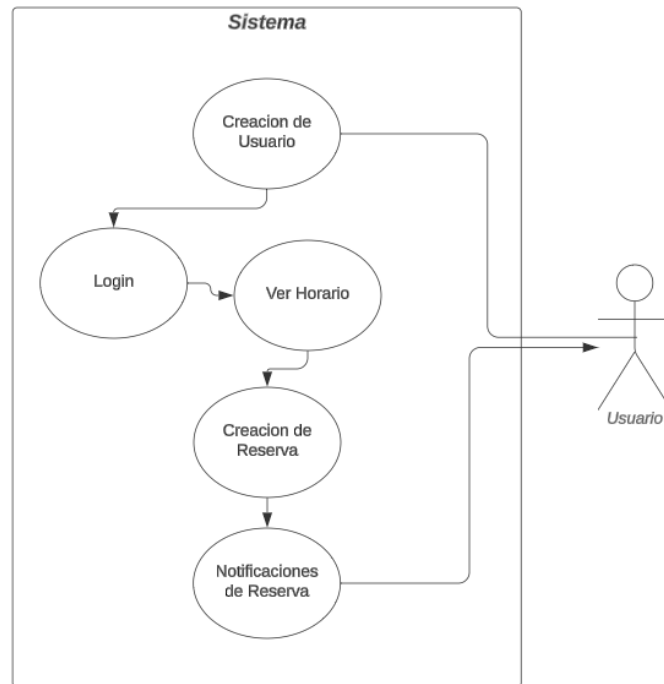
4. SEMANA 4:

- Implementar de nuevo el funcionamiento que tenia en el primer desarrollo en local adaptándola al modelo creado por Docker, Compose y Laravel.
- Finalización del proyecto con poco tiempo por el cambio de servidor, aun así haciéndolo con paciencia y aprendiendo de lo solicitado.

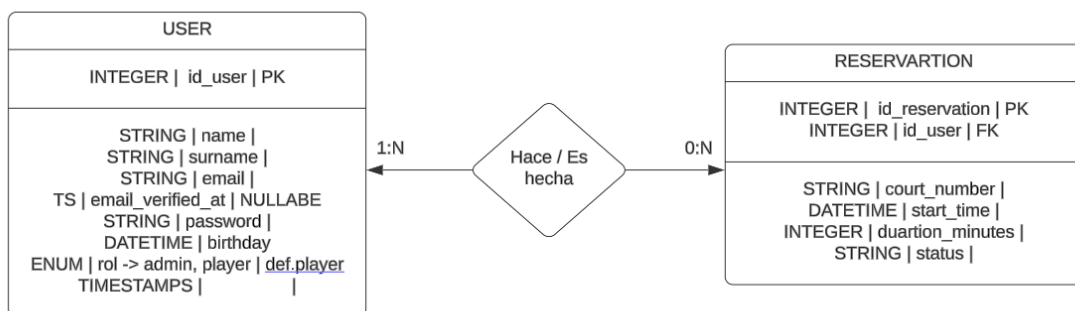
PARTE 2 DEL PROYECTO

Documentación: Análisis y Diseño

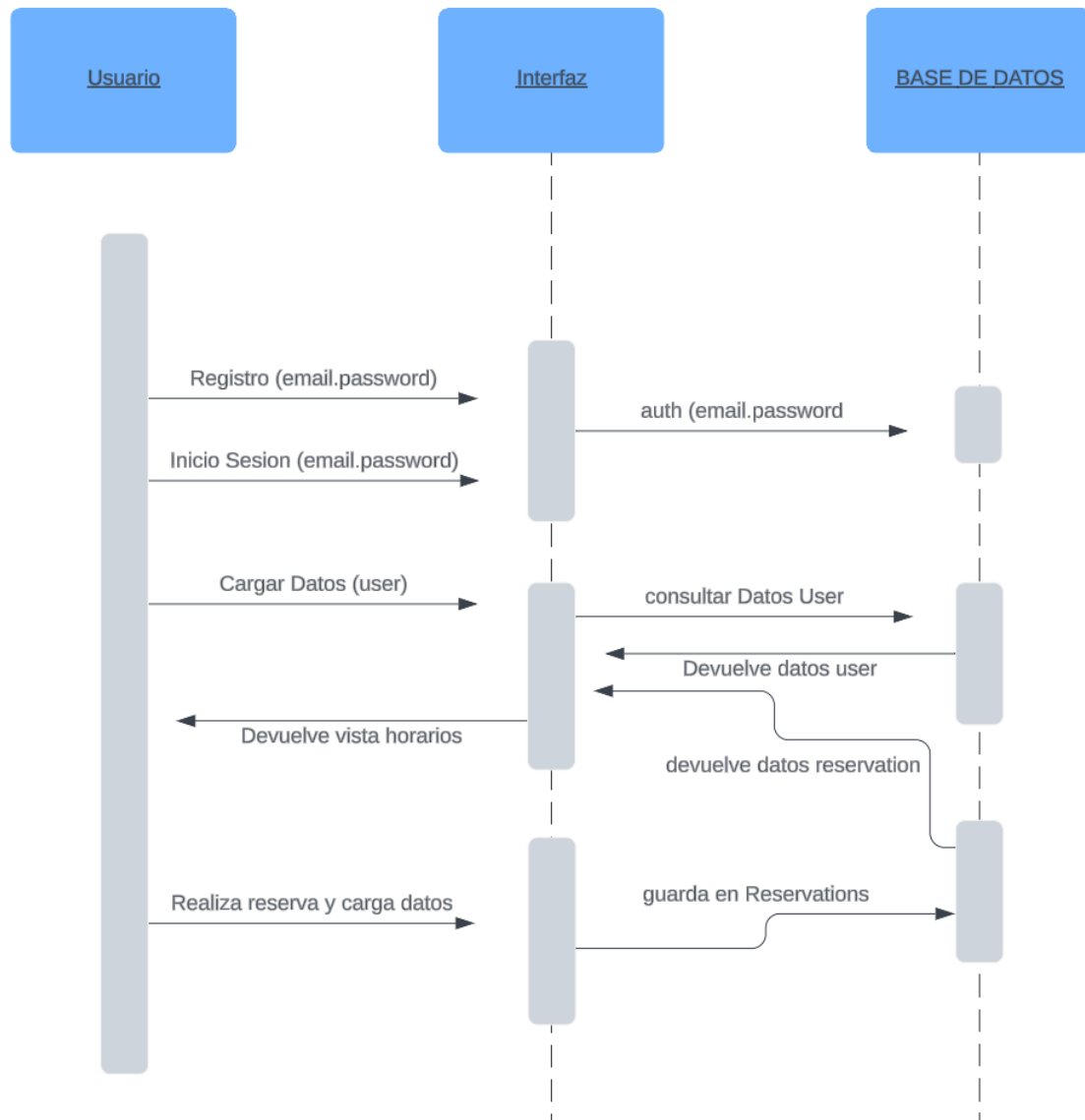
10. Diagrama de casos de uso



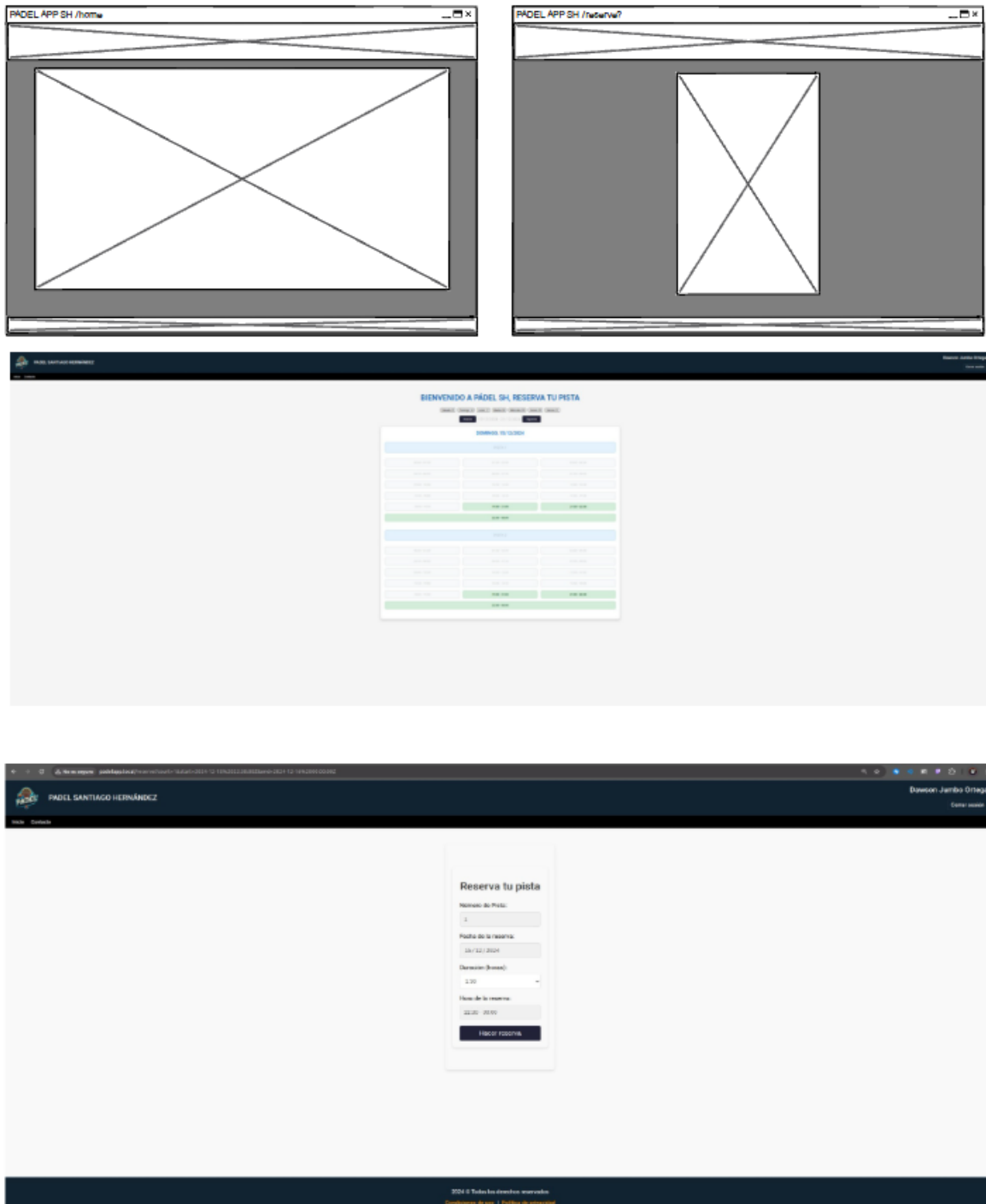
11. DIAGRAMA DE CLASES



12. Diagrama de secuencia de la Aplicación (UML)



13. Desarrollo de interfaces (MockUp)



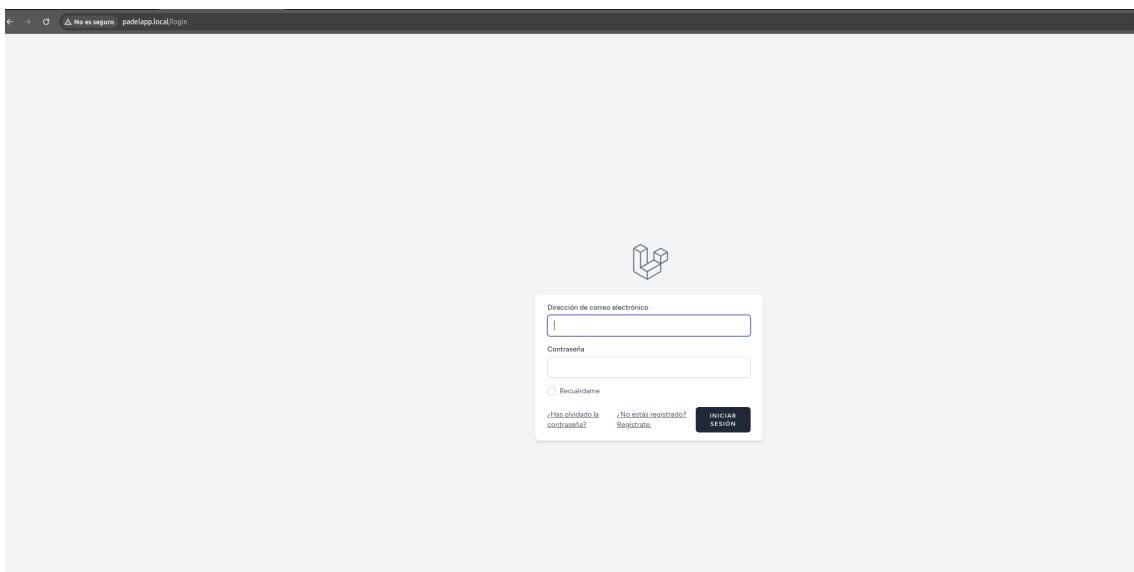
PARTE 3 DEL PROYECTO

Implementación, Interfaz y Pruebas:

14. Pruebas, Interfaz y rendimiento

Ahora vamos a proceder a hacer unas pruebas de la aplicación explicando cada funcionamiento.

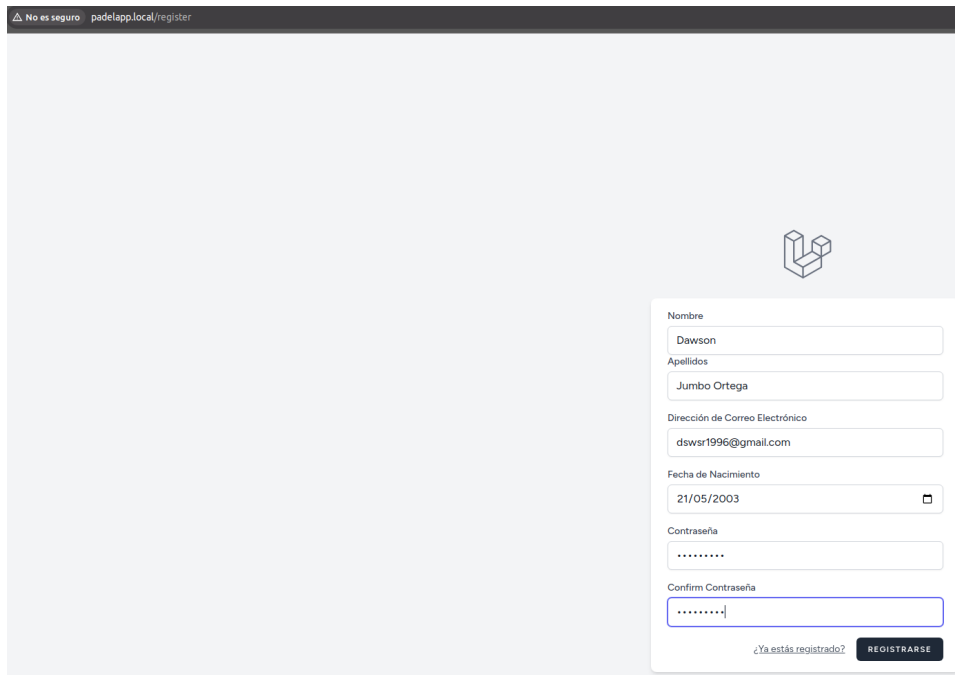
Login



Se redirige al /login, gracias web.php que me permite redirigir teniendo en cuenta el check de si el usuario está Autorizado o no:

```
// Ruta principal de bienvenida
Route::get('/', function () {
    return Auth::check() ? redirect('/home') : redirect('/login');
});
```

Register

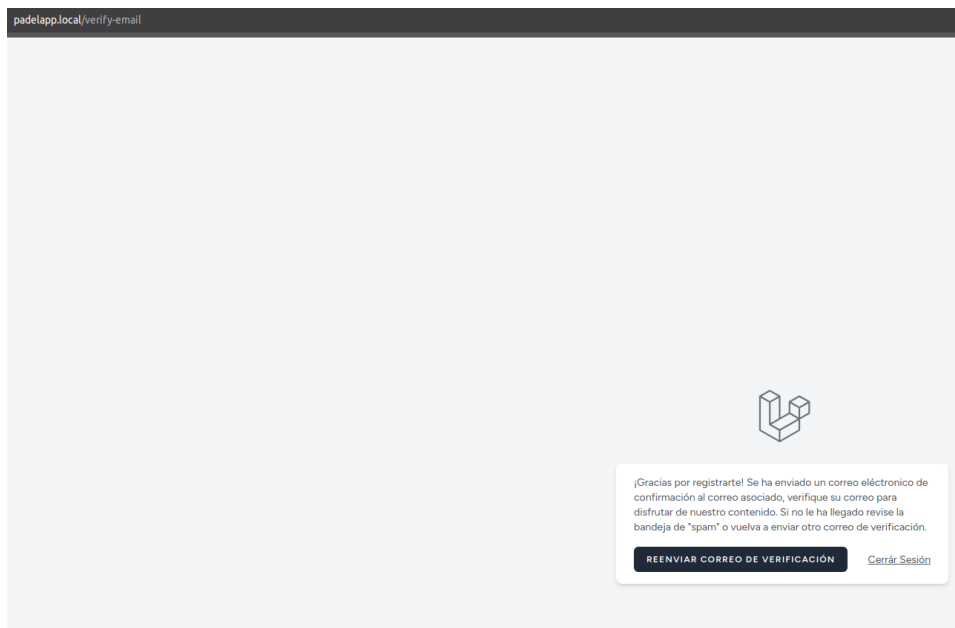


The screenshot shows a web browser window with the address bar displaying "No es seguro" and "padelapp.local/register". The page has a light gray background with a dark header. In the center, there is a registration form with a logo above it. The form fields are: "Nombre" (Dawson), "Apellidos" (Jumbo Ortega), "Dirección de Correo Electrónico" (dswsr1996@gmail.com), "Fecha de Nacimiento" (21/05/2003), "Contraseña" (masked with dots), and "Confirm Contraseña" (masked with dots). At the bottom of the form, there is a link "¿Ya estás registrado?" and a "REGISTRARSE" button.

Como vemos al dirigirnos a registrarnos porque no tenemos cuenta nos lleva a un form sencillo donde rellenamos los campos y este es creado.

Verify Email

Como vemos, al registrarnos nos redirige automáticamente a verify-email, donde nos explica que debemos hacer y nos permite reenviar por si no lo veo o cerrar sesión si deseas ir atrás.



The screenshot shows a web browser window with the address bar displaying "padelapp.local/verify-email". The page has a light gray background with a dark header. In the center, there is a verification message box with a logo above it. The message text is: "¡Gracias por registrarte! Se ha enviado un correo electrónico de confirmación al correo asociado, verifique su correo para disfrutar de nuestro contenido. Si no le ha llegado revise la bandeja de 'spam' o vuelva a enviar otro correo de verificación." Below the message, there are two buttons: "REENVIAR CORREO DE VERIFICACIÓN" and "Cerrar Sesión".

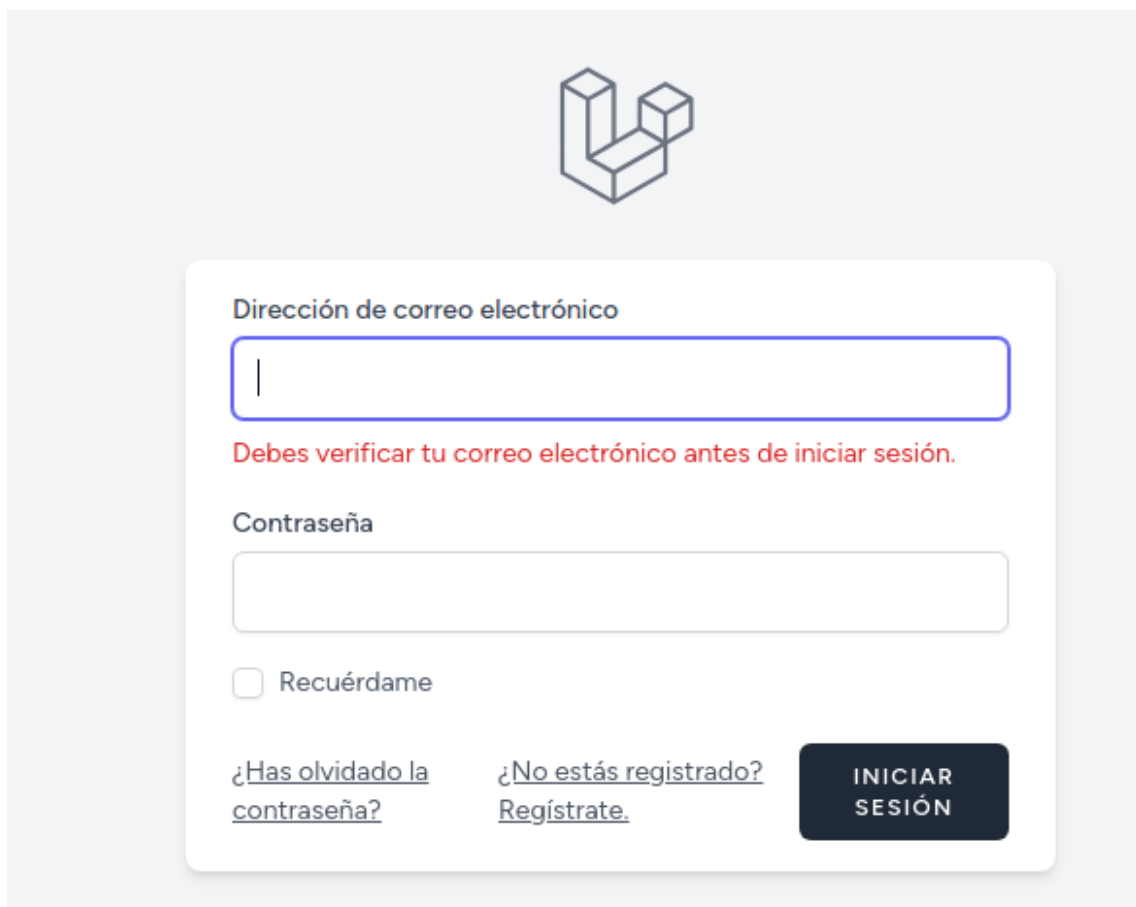
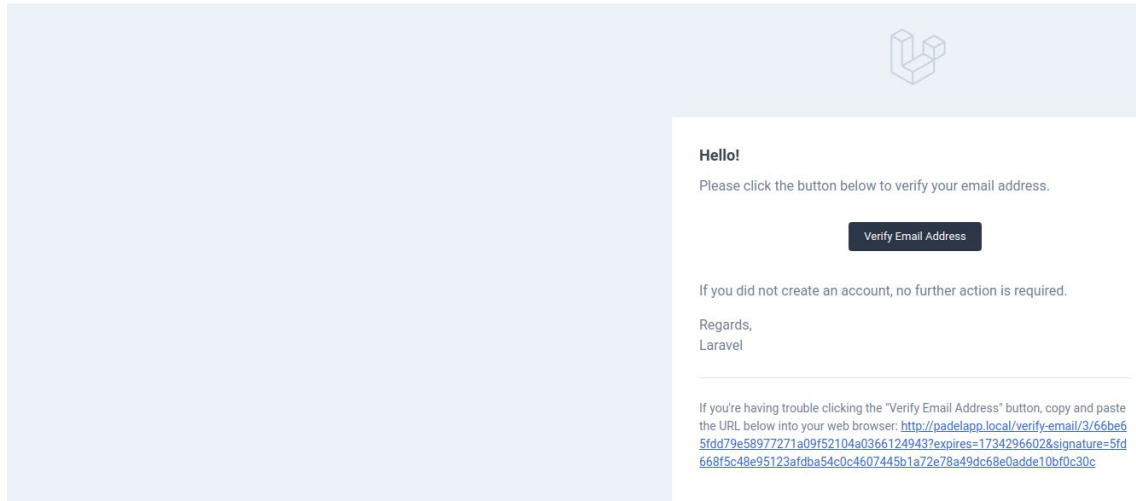
Verify Email Link

Como vemos nos llega al correo una vista de verificación de correo, ya que si no hacemos esto no nos permitirá iniciar sesión, la prueba en la siguiente captura.

/erify Email Address ▾ Recibidos x

xample <ortelcontacto@gmail.com>
ara mí ▾

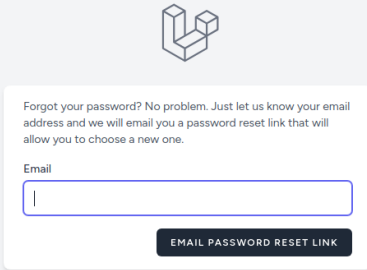
Traducir al español x



Forgot Password

Como vemos redirige perfecto y envía al correo notificación del cambio de contraseña.

△ No es seguro padelapp.local/forgot-password



Forgot your password? No problem. Just let us know your email address and we will email you a password reset link that will allow you to choose a new one.

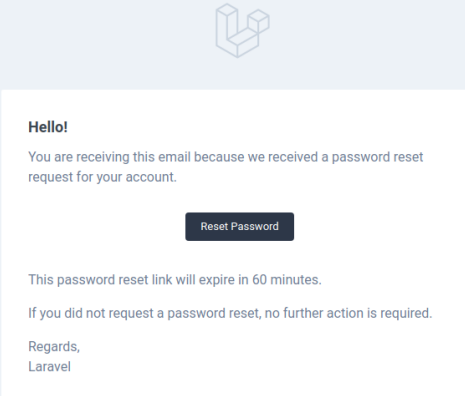
Email

EMAIL PASSWORD RESET LINK

Reset Password Notification

Example <ortelcontacto@gmail.com>
para mi

Traducir al español



Hello!

You are receiving this email because we received a password reset request for your account.

Reset Password


This password reset link will expire in 60 minutes.

If you did not request a password reset, no further action is required.

Regards,
Laravel

Al darle al botón redirige a una pagina en la que cambiamos la contraseña:

padelapp.local/reset-password/21ea99e09dbc7e9dc869b63963b0f2c7d8ca10f4fbcc63b70d8cf931f76d4e7?email=dswsr1996%40gmail.com



Email
dswsr1996@gmail.com


Password
.....

Confirm Password
.....|

RESET PASSWORD

Luego de cambiarla nos redirige al login con el mensaje de que nuestra contraseña ha sido cambiada:

padelapp.local/login



Your password has been reset.

Dirección de correo electrónico
|

Contraseña
|

☐ Recuérdame

[¿Has olvidado la contraseña?](#) [¿No estás registrado? Regístrate.](#)

INICIAR SESIÓN

Home

Una vez completado la verificación del email, redirige a home.



Como vemos muestra el horario de pista 1 y pista 2 a través de home.js, también muestra la hora y el día de hoy, al igual que puede pasar de anterior a siguiente cambiando la semana la y los botones con el día correspondiente:



Reserve

Si le doy a cualquier horario de reserve aparece el horario a través de un fetch y api en la url:

`http://padelapp.local/reserve?court=1&start=2024-12-15%2022:30:00Z&end=2024-12-16%2000:00:00Z`

The screenshot shows a web browser window with the URL `http://padelapp.local/reserve?court=1&start=2024-12-15%2022:30:00Z&end=2024-12-16%2000:00:00Z`. The page header includes the Padel logo and the name 'PADEL SANTIAGO HERNÁNDEZ', with links for 'Inicio' and 'Contacto'. The main content area features a 'Reserva tu pista' form with the following fields: 'Número de Pista' (set to 1), 'Fecha de la reserva' (set to 15/12/2024), 'Duración (horas)' (set to 1:00), and 'Hora de la reserva' (set to 22:30 - 23:30). A 'Hacer reserva' button is at the bottom of the form. The footer contains the text '2024 © Todos los derechos reservados' and links for 'Condiciones de uso' and 'Política de privacidad'.

Reserve Store

Como vemos al hacer la reserva con los datos puestos aunque me hayan pasado de 22:30 a 00:00 lo pone perfecto con la duración de 1 hora (de 22:30 a 23:30) y lo pone en rojo y no es clickable:

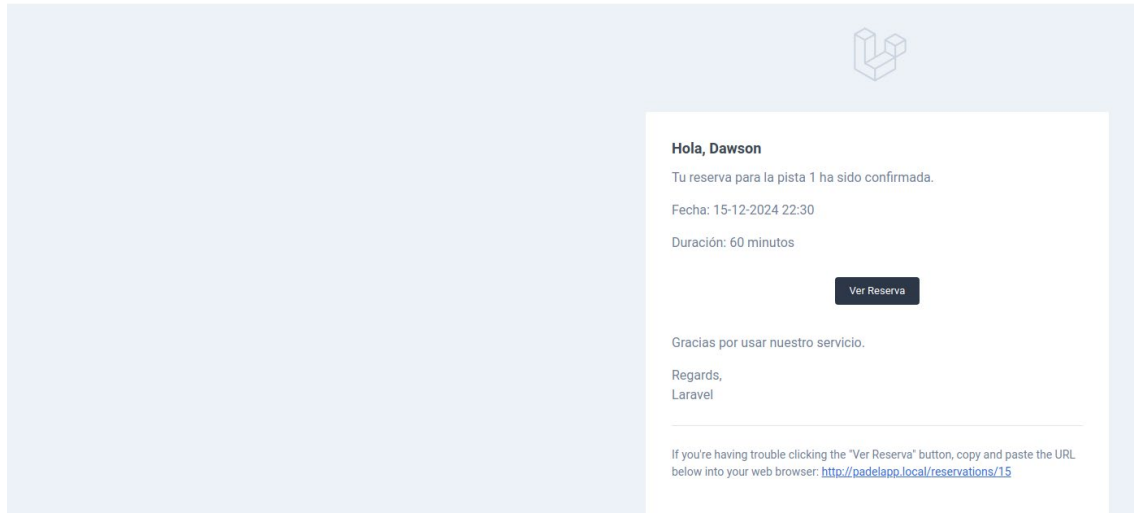
The screenshot displays a 'Reserve Store' interface with three time slots in the top row: '18:00 - 19:30' (light blue), '19:30 - 21:00' (light blue), and '21:00 - 22:30' (green). Below these, a red bar represents a reserved slot for '22:30 - 23:30', with the word 'Reservado' centered underneath.

Como vemos en el correo llega un mensaje y si fuera un intervalo disponible sale el recordatorio a reserva:


Reserva Confirmada   Recibidos x

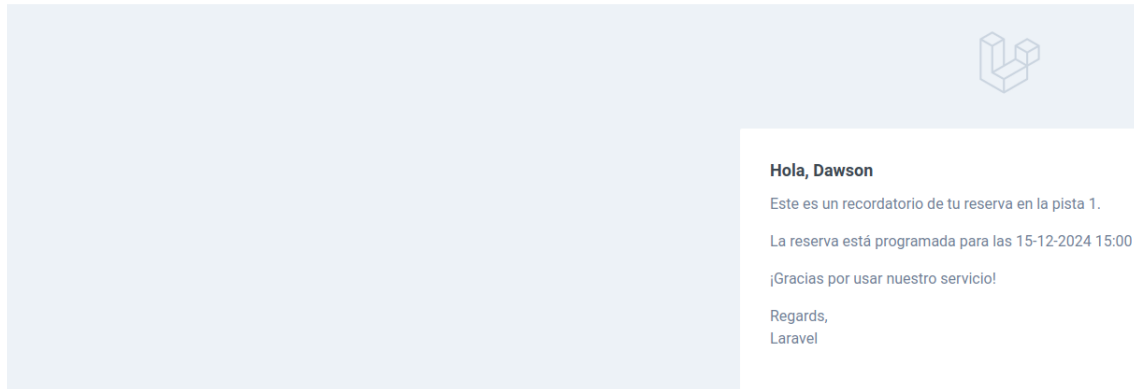
Example <ortelcontacto@gmail.com>
para mí 

 Traducir al español 



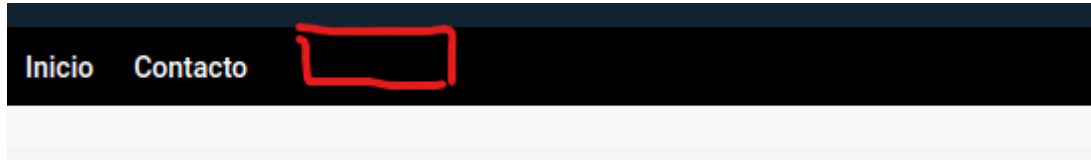
Recordatorio de tu Reserva   Recibidos x

Example <ortelcontacto@gmail.com>
para mí 

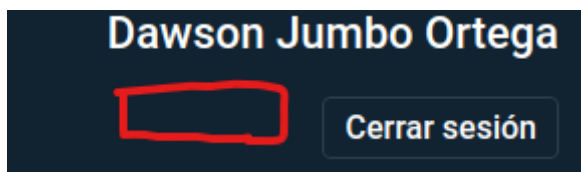


Desarrollos a implementar

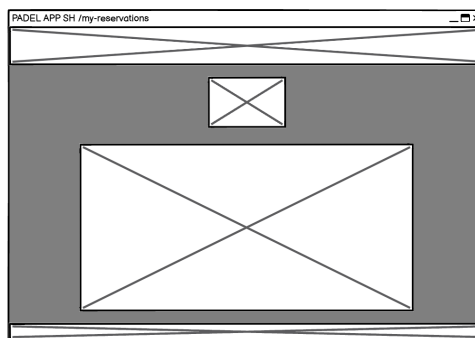
Vista Reservation en nav bottom, para ver las reservas hechas, esta hecho pero no implementado.



Edit de Profile, desarrollado pero no implementado aún:



También faltaría el route de admin, en el que la vista my-reservation podría ser una vista de una tabla con calendario para revisar las reservas hechas (admin podría todo) y también tendría vistas escondidas que permitirían a admin borrar las reservas que se cancelen a través de un contacto:



```

<div class="filter-calendar">
  <form action="{{ route('reservations.my') }}" method="GET" class="filter-form">
    <label for="filter_date">Filtrar por fecha:</label>
    <input type="date" id="filter_date" name="filter_date" value="{{ (request('filter_date', now())->toString()) }}">
    <button type="submit" class="btn btn-primary">Buscar</button>
  </form>
</div>

<div class="table-responsive">
  <table class="table table-striped">
    <thead>
      <tr>
        <th>Fecha</th>
        <th>Pista</th>
        <th>Duración</th>
        <th>Estado</th>
        @if(auth()->user()->role === 'admin')
          <th>Acciones</th>
        @endif
      </tr>
    </thead>
    <tbody>
      @foreach ($reservations as $reservation)
        <tr>
          <td>{{ \Carbon\Carbon::parse($reservation->start_time)->format('d/m/Y') }}</td>
          <td>{{ $reservation->court_number }}</td>
          <td>{{ \Carbon\Carbon::parse($reservation->start_time)->format('H:i') }} - {{ \Carbon\Carbon::parse($reservation->start_time)->addMinutes($reservation->duration_minutes)->format('H:i') }}</td>
          <td class="{{ ( \Carbon\Carbon::parse($reservation->start_time)->isPast() ? 'status-finished' : ( \Carbon\Carbon::parse($reservation->start_time)->isFuture() ? 'status-active' : 'status-pending' )) }}">{{ \Carbon\Carbon::parse($reservation->start_time)->isPast() ? 'Finalizada' : ( \Carbon\Carbon::parse($reservation->start_time)->isFuture() ? 'Activa' : 'Activa') }}</td>
          @if(auth()->user()->role === 'admin')
            <td>
              @if ($reservation->active && !\Carbon\Carbon::parse($reservation->start_time)->isPast())
                <form action="{{ route('reservations.destroy', $reservation->id) }}" method="POST" class="action-form">
                  @csrf
                  @method('DELETE')
                  <button type="submit" class="btn btn-danger">Cancelar</button>
                </form>
              @else
                <span></span>
              @endif
            </td>
          @endif
        </tr>
      @endforeach
    </tbody>
  </table>
</div>

```

15. Memoria Económica

Especificaciones del dispositivo

Copiar

^

Nombre del dispositivo	DESKTOP-C9Q1HJ0
Procesador	12th Gen Intel(R) Core(TM) i7-12700H 2.70 GHz
RAM instalada	16.0 GB (15.7 GB usable)
Identificador de dispositivo	811B4F78-35F2-4DD2-B1A6-3338ESC9972E
Id. del producto	00330-80000-00000-AA814
Tipo de sistema	Sistema operativo de 64 bits, procesador basado en x64
Lápiz y entrada táctil	La entrada táctil o manuscrita no está disponible para esta pantalla

Vínculos relacionados

[Dominio o grupo de trabajo](#)
[Protección del sistema](#)
[Configuración avanzada del sistema](#)

Especificaciones de Windows

Copiar

^

Edición	Windows 11 Pro
Versión	24H2
Instalado el	06/10/2024
Versión del sistema operativo	26100.2605
Experiencia	Paquete de experiencia de características de Windows 1000.26100.36.0
Contrato de servicios de Microsoft	
Términos de licencia del software de Microsoft	

Como vemos las horas invertidas han sido aproximadamente 50 horas, teniendo en cuenta el gasto energético del ordenador, debería ser entre 11/12€ hora lo que equivaldría a entre 500 y 600 el proyecto.

16. Licencia

Este proyecto consiste en el desarrollo de una aplicación de escritorio para agilizar y centralizar la gestión académica y administrativa de un instituto. La aplicación ofrece características clave como el registro y acceso de estudiantes, el acceso de profesores, el seguimiento de tareas, la gestión de asistencia y el monitoreo de calificaciones. Además, incluye un sistema de correo electrónico integrado para la comunicación y notificaciones, lo que permite mantener a los usuarios informados sobre eventos importantes.

La aplicación está diseñada para mejorar la eficiencia en la gestión académica y es escalable para futuras mejoras, como una versión web, integración con APIs y herramientas administrativas avanzadas.