



I.E.S. “ÁGORA”

Departamento

Informática y Comunicaciones

Ciclo Formativo de Grado Superior

Desarrollo de Aplicaciones Web



PropiGest

(Herramienta gestora de propiedades inmobiliarias)

Autor: David García Rodríguez

Tutor: Raquel Portugal Iglesias

Cáceres, a 26 de junio de 2025

Visto Bueno en la entrega del Proyecto de Ciclos Formativos de Grado Superior

Nombre y apellidos del Profesor : RAQUEL PORTUGAL IGLESIAS

Nombre y apellidos del alumno/os: DAVID GARCÍA RODRÍGUEZ

Título del Proyecto: PropiGest

Grupo: 2W Bilingüe

A través del presente documento, el Profesor tutor acredita que el alumno citado anteriormente ha entregado en modo, forma y fecha el Proyecto de Ciclos Formativos de Grado Superior arriba mencionado.

En Cáceres a 04 de Junio de 2025

Vº Bº EL TUTOR

Fdo.: Raquel Portugal Iglesias

Índice

INTRODUCCIÓN2

DESARROLLO DEL PROYECTO4

ALTERNATIVAS.....10

POSIBLES MEJORAS.....12

MODELO DE NEGOCIO14

RESULTADOS.....15

CONCLUSIONES.....17



Introducción

Este proyecto se desarrolla como solución a la necesidad de digitalizar y facilitar la gestión de propiedades inmobiliarias en alquiler, especialmente orientado a propietarios privados o agencias inmobiliarias que ofrecen alquileres a inquilinos. En la actualidad, muchas de estas personas o entidades recurren a métodos tradicionales o herramientas poco especializadas que dificultan el seguimiento efectivo de reservas, mantenimientos, pérdidas económicas o mala experiencia del usuario final, debido a la poca claridad de estas herramientas y claramente no están diseñadas para este tipo de tareas en específico.

La aplicación desarrollada brinda mediante el uso centralizado unas herramientas necesarias al propietario o agencia encargada de un determinado número de inmuebles para poder gestionar las reservas y administrar el inmueble de la forma más eficaz y sencilla posible, una interfaz intuitiva, sencilla y colores ligeros proporcionará a los usuarios (propietarios o agencias) una experiencia sencilla y agradable para poder llevar sus inmuebles de la forma más fácil y rentable posible, gracias a las funciones de mi aplicación.

Entre ellas están algunas como el registro de las propiedades, ubicación en un mapa, gestión automática de la disponibilidad mediante reservas, control de los mantenimientos que se las realiza, control de los movimientos y resúmenes de balances económicos visuales y claros, por lo que facilita la toma de decisiones y el análisis de la rentabilidad. Este conjunto de herramientas no solo busca facilitar la gestión diaria, sino también mejorar la eficiencia operativa y la experiencia del usuario, tanto a nivel técnico como funcional.

La motivación principal para desarrollar este proyecto nace del interés personal por el desarrollo web, la programación orientada a la solución de problemas reales y la experiencia acumulada durante el ciclo formativo y prácticas en la empresa. Asimismo, he tenido en cuenta la creciente demanda de aplicaciones que brinden la capacidad de acceso centralizado de sus datos, en este caso de sus inmuebles, incorporando funciones que agreguen ese valor funcional y de reconocimiento a la aplicación sin olvidar la interacción y experiencia visual del usuario, que es el principal consumidor de este producto mediante el cual podrá satisfacer sus necesidades y un incremento de su productividad, pudiendo organizar las reservas, mantenimientos y movimientos de sus inmuebles con tan solo un dispositivo (móvil, tablet, ordenador...) y una conexión a internet.

Espero que sea más que un solo proyecto de final de grado y permita ayudar a todos estos usuarios potenciales hacia los que dirijo la aplicación facilitando la administración de sus inmuebles.

El proyecto abarca diversas áreas de trabajo dentro del desarrollo de aplicaciones web:

- Diseño de la interfaz de usuario
- Desarrollo del backend en PHP
- Conexión a bases de datos MySQL
- Uso de sesiones de usuarios
- Uso de JavaScript para la interactividad del usuario y bloques funcionales
- Implementación de mapas mediante Leaflet
- Desarrollos de funcionalidades como calendarios (FullCalendar), filtros (DataTable) y gráficos estadísticos (Chart.js)

La plataforma ha sido diseñada teniendo en cuenta criterios de usabilidad y mantenimiento, y se ha trabajado para que sea responsive y funcional desde diferentes dispositivos, por lo que no estar en casa no será un impedimento para poder ocuparte de tus propiedades.

También he usado la plataforma Github para realizar el control de versiones del código, subiendo “commits” a la plataforma al realizar las tareas que me he ido marcado también desde una de sus herramientas “projects”, en la que puedes crear una TODO list.

Por último, cabe aclarar que este proyecto se encuentra protegido por la Ley de **Propiedad Intelectual (Real Decreto Legislativo 1/1996, de 12 de abril)** y por la **Directiva 2001/29/CE del Parlamento Europeo** sobre derechos de autor en la sociedad de la información. [Ver documento adjunto “AvisoLegal.pdf”](#)

Desarrollo del proyecto

El desarrollo del proyecto se ha realizado de forma progresiva, comenzando con una planificación inicial de estructura de datos, funcionalidades y diseño, seguida de la implementación de los distintos módulos, tablas de la base de datos y su integración final conjunta. La base del proyecto es una aplicación web desarrollada principalmente con tecnologías HTML, CSS, JavaScript, PHP y MySQL, respetando los principios de usabilidad, escalabilidad y mantenibilidad.

- **Estructura general de la aplicación**

La aplicación está dividida en distintas vistas, orientadas al usuario propietario o gestor.

Las principales secciones desarrolladas han sido:

- **Inicio:** Panel de visualización con listado de propiedades registradas, con datos relevantes, gráficas y accesos rápidos a cada sección de gestión.
- **Vista Mapa:** Muestra todas las propiedades geolocalizadas mediante integración con Leaflet.js, permitiendo visualizar su ubicación y estado de disponibilidad.
- **Detalles propiedad:** Muestra los detalles de cada propiedad seleccionada, encontraremos las imágenes de la propiedad, si son más de una mediante un carrusel, junto a toda la información de la misma, previsualización de un mini mapa y acciones de editar o eliminar la propiedad. Un menú circular con tres botones nos mostrará las siguientes funciones de cada propiedad individualmente:
 - **Balance:** Nos genera automáticamente un resumen mediante gráficos y datos junto a una tabla con el histórico de movimientos de la propiedad a la que hayamos accedido, pudiendo añadir ingresos y gastos personalizados, todo esto solo para la propiedad a la que accedimos individualmente.
 - **Reserva:** Nos genera automáticamente un calendario con todas las reservas programadas y pasadas junto a una tabla con el histórico de reservas, pudiendo añadir una reserva para esta propiedad individualmente insertando los datos del inquilino y de la reserva.
 - **Mantenimiento:** Nos genera automáticamente un calendario con todos los mantenimientos, programados y realizados junto a una lista con todas las tarjetas de los datos de los mantenimientos, pudiendo programar o registrar un nuevo mantenimiento.

- **Resúmenes generales:** En el menú encontramos varios iconos que nos dan acceso a los siguientes resúmenes generales de todas las propiedades del usuario, es decir, lugar donde podrá encontrar toda la información global sobre todas sus propiedades registradas en la aplicación.
 - **Balances:** En esta sección encontramos el resumen completo de balances de todas las propiedades del usuario, encontramos un apartado superior con gráficos que representan los totales de balances y la propiedad con mayor balance de todas las propiedades del usuario, datos sobre balance total, ingresos totales, gastos totales y finalmente una tabla a modo histórico con todos los movimientos (ingresos y gastos) con filtros y búsqueda.
 - **Reservas:** En esta sección encontramos el resumen completo de reservas de todas las propiedades del usuario, encontramos un apartado superior con la propiedad más reservadas del mes, grafico con reservas por meses y un total de propiedades disponibles y no disponibles, un calendario en el que se muestran todas las reservas de todas las propiedades y finalmente una tabla también a modo histórico con todas las reservas con filtros y búsqueda.
 - **Mantenimientos:** En esta sección encontramos el resumen completo de mantenimientos de todas las propiedades del usuario, encontramos un apartado superior con la propiedad con más mantenimientos del mes, grafico con el número de mantenimientos por mes y un gráfico con comparación entre estados de los mantenimientos registrados para todas las propiedades, un calendario en el que se muestran todos los mantenimientos de todas las propiedades (programados y completados) y finalmente un listado con todas las tarjetas de todos los mantenimientos registrados de todas las propiedades del usuario.

- **Partes más importantes**

- **Gestión automática de disponibilidad:** Uno de los retos técnicos más importantes fue implementar la lógica para que las propiedades cambien automáticamente su estado de “disponible” a “no disponible” en función de las fechas registradas en el sistema de reservas. Para ello desarrollé un script en PHP que se ejecuta en segundo plano una vez al día en el primer acceso del día del primer usuario, es decir, cuando sean las 00:00 el próximo usuario que acceda lanzará la petición de revisión de estados, esto generará un log que se consultará constantemente para no volver a lanzar la petición, ya que almacena la fecha del día y no volverá a lanzarlo hasta que no se modifique ese log.
- **Mapa interactivo con leaflet.js:** Incorporé esta librería para mostrar la localización de los inmuebles. Integré correctamente con los datos provenientes de la base de datos y adapté a diseños responsive en dispositivos móviles, logrando una visualización fluida y clara en cualquier tamaño de pantalla.
- **Calendario de mantenimientos y reservas:** Mediante el uso de una api llamada FullCalendar implementé un calendario en los apartados de mantenimientos y reservas para poder mostrar los mantenimientos y reservas en estos, lo que permite ver y planificar con un mejor claridad los eventos cercanos acerca de la propiedad en el caso de los detalles de una propiedad o todos en caso de los resúmenes de todas las propiedades, además permite ver los eventos pasados y futuros.
- **Seguridad y gestión de sesiones:** Para mí uno de los puntos más importantes a la hora de tratar con datos de usuarios y por ello, debido a que el aumento de ciberataques no deja de aumentar día a día me veo comprometido a asegurar de la mejor forma los datos de los usuarios que confían en mi herramienta para datos tan importantes como son los datos de sus propiedades, datos financieros, inquilinos... Por ello ha sido uno de los puntos fuertes a tratar y a tener en cuenta durante el desarrollo de la aplicación, por lo que he implementado un sistema robusto de seguridad y gestión de sesiones.

1. Control de sesiones:

Se ha implementado un sistema de autenticación y gestión de sesiones que asegura que solo los usuarios puedan acceder a las funcionalidades privadas. Esto incluye:

- a. Validación del inicio de sesión y almacenamiento seguro del estado de inicio de sesión en variables `$_SESSION`.
- b. Verificación en cada vista protegida mediante condicionales que bloquean el acceso si el usuario no está autenticado en el sistema
- c. Redirección automática a la página de login en caso de acceso no autorizado.

2. Prevención de inyecciones SQL:

Para evitar inyecciones SQL, se ha utilizado el mecanismo de consultas preparadas con sentencias **PDO**, que permiten separar completamente la lógica de la consulta de los valores insertados por el usuario. Gracias a esto, incluso si un atacante intenta introducir código malicioso, no se ejecutará dentro de las sentencias SQL.

3. Protección contra inyecciones por URL:

Durante el desarrollo detecté que la aplicación podría ser vulnerable a modificaciones de parámetros en la URL. Para solucionarlo implemente la redirección o salida inmediata si los parámetros esperados no están presentes o son inválidos.

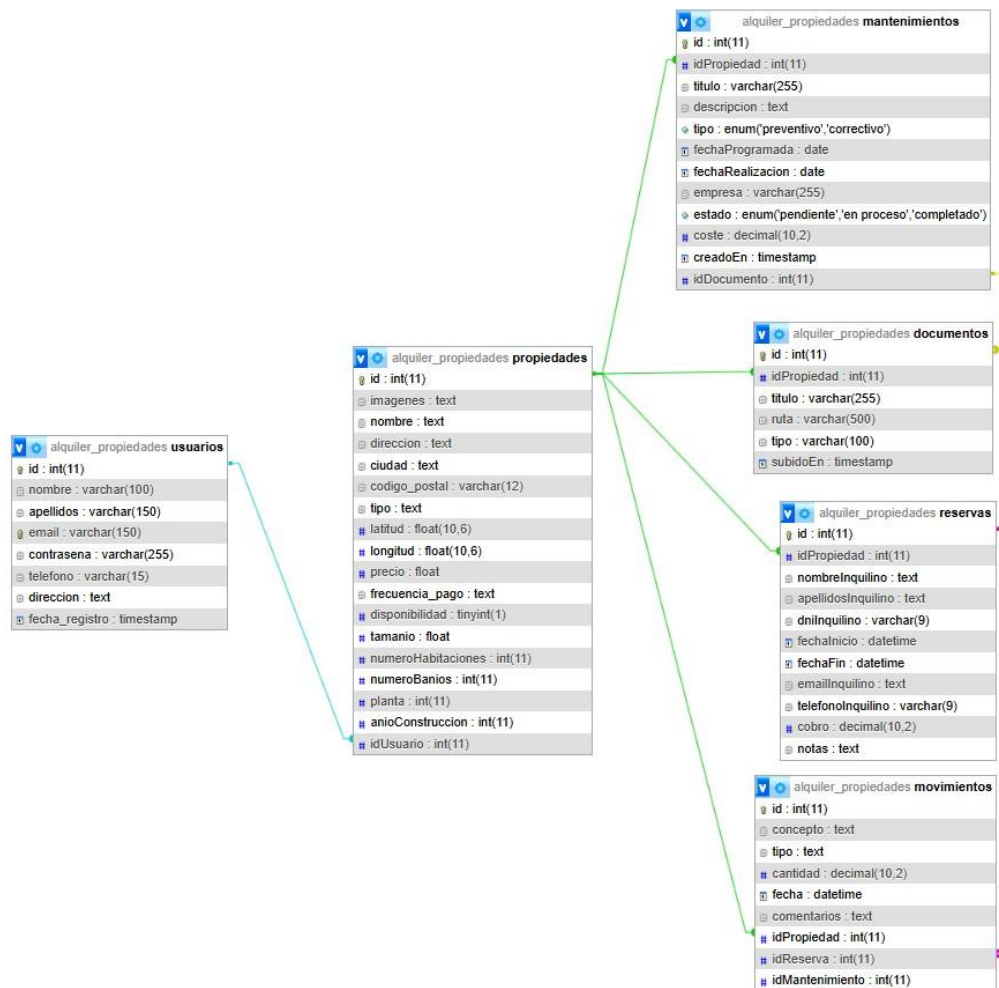
4. Restricción de navegación entre directorios:

He tenido especial cuidado en evitar la navegación entre directorios mediante rutas manipuladas por URL. Para ello:

- a. He definido una estructura clara de rutas protegidas por controladores.
- b. He restringido accesos mediante **htaccess** cuando ha sido necesario, y con verificaciones en PHP.
- c. He eliminado posibles puntos de entrada directa a scripts sin autenticación, como pueden ser archivos que procesan datos.

Esquema de la base de datos

A continuación, se muestra el esquema general de la base de datos del proyecto, diseñada en MySQL. La estructura ha sido pensada para mantener la coherencia entre entidades y permitir una escalabilidad futura sin complicaciones, además de facilitar las consultas eficientes y permitir una escalabilidad progresiva.



Tablas principales y relaciones:

- **Usuarios:** Contiene los datos de los propietarios que pueden gestionar una o más propiedades. Incluye campos como nombre, apellidos, email, contraseña (encriptada), teléfono, dirección y fecha de registro.
- **Propiedades:** Almacena toda la información referente a cada inmueble, incluyendo ubicación (latitud y longitud), precio, frecuencia de pago, estado de disponibilidad, número de habitaciones, baños, planta, tamaño y año de construcción. Se vincula a *usuarios* mediante *idUsuario*.

- **Reservas:** Gestiona las reservas realizadas sobre las propiedades. Incluye la información del inquilino, fechas de inicio y fi, estado del cobro y observaciones. Está relacionada con propiedades mediante *idPropiedad*.
- **Mantenimientos:** Recoge los mantenimientos programados o realizados sobre una propiedad. Cada mantenimiento puede estar vinculado opcionalmente a un documento (factura) relacionado con *documentos* y a *movimientos*, ya que genera un gasto automáticamente. Se distingue entre mantenimientos preventivos y correctivos, y a cada uno tiene un estado de ejecución (pendiente, en proceso y completado).
- **Documentos:** Permite almacenar ficheros asociados a una propiedad, como facturas, contratos, informes... Se relaciona con *propiedades*.
- **Movimientos:** Tabla clave para la contabilidad. Recoge ingresos y gasto asociados a propiedades o mantenimientos, incluyendo tipo de movimientos, importe, fecha y notas. Puede estar asociado tanto a una propiedad como a una reserva o mantenimiento concreto.

Esta estructura ha sido pensada para facilitar una administración clara y ordenada. La normalización de las relaciones permite:

- Evitar la redundancia de datos.
- Realizar búsquedas y filtros eficientes.
- Permitir futuras ampliaciones, como agregar distintos roles de usuario, integración con pasarelas de pago o generación automática de informes.

Además, durante el desarrollo se han tenido en cuenta aspectos de seguridad en la gestión de esta base de datos, como:

- Validación de entradas del usuario
- Preparación de consultas SQL para evitar inyecciones (*mysqli_prepare* o *PDO*).
- Control de accesos por sesión.
- Restricción de acceso a rutas de documentos mediante comprobaciones del usuario autenticado

Alternativas

Durante la fase de planificación y diseño del proyecto, analicé distintas alternativas tecnológicas tanto para el desarrollo frontend como para el backend. El objetivo era elegir aquellas que mejor se adaptaran al alcance del proyecto, a los recursos disponibles y, sobre todo, a mi nivel de experiencia actual como desarrollador.

1. Frameworks de desarrollo full-stack

Valoré el uso de arquitecturas más modernas y robustas basadas en tecnologías como:

- a. **Angular:** Para el frontend, un framework muy potente para el desarrollo de interfaces SPA (interfaces en una sola página).
- b. **Java Spring Boot:** Para el backend, ampliamente utilizado en entornos profesionales y que ofrece herramientas muy completas para la seguridad, gestión de usuarios, servicios REST, etc.

Sin embargo, descarte el uso de ambas tecnologías ya que:

- La curva de aprendizaje en ambas es bastante pronunciada, y el tiempo disponible para la ejecución del proyecto era limitada.
- Requieren una infraestructura algo más compleja de despliegue y configuración (servidores java, control de versiones de las dependencias...).
- Aunque estoy formando me en ellas y tengo meses de experiencia en ellas, mi experiencia actual en las tecnologías de JS y PHP por ejemplo es mayor, por lo cual me decidí por unas tecnologías con las que me encontraba más familiarizado para poder centrarme mejor en la lógica de negocio del proyecto.

2. Diseño del frontend

En cuanto al diseño de la interfaz, se valoró inicialmente hacerla completamente personalizada con CSS puro. Pero finalmente opté por utilizar Bootstrap, una de las librerías CSS más usadas, ya que:

- a. Acelera el desarrollo de interfaces limpias y adaptables (responsive).
- b. Facilita el trabajo con componentes predefinidos como modales, botones, listas, tablas...
- c. Permite mantener una estructura coherente sin complicarse con detalles visuales minúsculos. Muy importante aquí, ya que el tiempo es un limitante.

3. Backend y lógica de servidor

Otra alternativa fue usar frameworks backend como Laravel (PHP) o Node.js con Express, pero finalmente decidí usar PHP nativo, ya que:

- a. Me permite entender mejor el flujo del backend, ya que no dependo de la estructura de un framework como puede ser Laravel.
- b. Está alineado con lo aprendido durante el curso, por lo que estoy más familiarizado con ello.
- c. Es más fácil de instalar y desplegar en cualquier servidor web tradicional (como xampp).

4. Base de datos

Con respecto al motor de base de datos, analicé la posibilidad de una base de datos PostgreSQL y mongoDB, aunque finalmente decidí MySQL/MariaDB, ya que:

- a. Su total compatibilidad con PHP.
- b. Familiaridad previa con su sintaxis y estructura relacional.
- c. Integración sencilla con entornos de desarrollo locales como XAMPP.

5. Mapeo de propiedades

Para mostrar las propiedades geolocalizadas, estudie la posibilidad de utilizar:

- a. **Google Maps API**, con muchas funcionalidades y soporte.
- b. **Leaflet.js**, una librería JavaScript de código abierto más ligera.

Finalmente escogí Leaflet.js por ser gratuita, fácil de integrar, personalizable y no requerir claves de API.

6. Gestión de usuarios y sesiones

Barajé opciones más avanzadas como OAuth2 o autenticación mediante JWT (JSON Web Tokens), pero finalmente opté por un sistema de autenticación clásico con PHP y sesiones, por:

- a. Su simplicidad y rapidez de implementación.
- b. Adecuación al contexto académico y al alcance del proyecto, porque de nuevo el tiempo marca un límite importante a tener muy en cuenta con respecto a la planificación y organización del desarrollo.

Posibles Mejoras

Aunque el proyecto cumple con los objetivos propuestos y ofrece una solución funcional y estable para la gestión de propiedades inmobiliarias, existen diversas áreas en las que me gustaría seguir ampliando y escalando la aplicación para conseguir una aplicación final mejorada y algunas funciones más, algunas como:

1. Gestión de usuarios más avanzada

Podría mejorarse el sistema de autenticación y dejar un sistema más robusto en la autenticación, a su vez más sencillo para el usuario con las siguientes mejoras:

- a. Recuperación de contraseña mediante correo electrónico
- b. Verificación en dos pasos (2FA)
- c. Validación de email al registrarse

2. Notificaciones automáticas

Integrar un sistema de notificaciones, tanto por correo electrónico como en la propia plataforma (notificaciones internas), que informen sobre:

- a. Nuevas reservas (alertará automáticamente al inquilino para confirmarle su reserva, si se elimina se avisará también automáticamente).
- b. Mantenimientos programados o finalizados (se alertará a modo recordatorio al arrendatario sobre la proximidad de un mantenimiento programado, al igual con el finalizado sobre su cambio de estado a “finalizado”).
- c. Al acercarse a la fecha de una reserva, se mandará un mensaje tanto al arrendatario como al inquilino para recordárselo y que ambos estén pendientes de la reserva.

3. Plataforma abierta y reservas automáticas

Una de las grandes escalabilidades de esta aplicación sería implementar una sección de forma pública, mediante la cual podrían entrar los inquilinos, ver todas las propiedades, incluso buscar, y poder reservar ellos directamente mediante una pasarela de pago, generando así un sistema de reservas completo.

4. BackUp automático de la base de datos

Implementar un sistema de copias de seguridad automáticas, ya sea en local o en la nube para poder garantizar la integridad de los datos en caso de fallo o pérdida de datos.

5. Mejora de la automatización de las disponibilidades

Actualmente se está realizando de una forma semiautomática, pero lo ideal sería programar un job o cron para automatizarlo por completo. Este se ejecutaría todos los días a las 00:00, por lo que no sería necesario que entrara ningún usuario para activarlo.

6. Apartado de servicios

En este apartado el propietario subiría los contratos de una propiedad con la compañía eléctrica, de agua, gas, internet... agregando cantidad al mes, si es una cuantía variable por consumo, fija... y añadiendo mes a mes los gastos en la propiedad a la que corresponde, incluyendo también el documento del contrato.

7. Biblioteca de documentos

Acompañando la idea anterior, generaría un apartado a modo repositorio, donde se verían todos los documentos de la propiedad, contratos, escrituras...

8. Seguridad en la conexión

Se puede implementar cloudflare, para hacer que las conexiones estén libres de bots y tengan una conexión limpia y seguir, prevenir ciberataques como puede ser el DDOS o denegación de servicio, el sistema de cloudflare, que actúa de forma autónoma y sencilla, denegando las ips a los atacantes, vigilando las conexiones... entre otras muchas funcionalidades.

9. Mejoras generales

Me gustaría seguir refinando la aplicación en los siguientes puntos:

- **Nivel interfaz:** Me gustaría seguir refinando y detallando más detenidamente toda la interfaz de la aplicación para conseguir un nivel más profesional aún a la aplicación, continuar mejorando la intuitividad de la misma y conseguir una interfaz más robusta aún.
- **Nivel funcional/lógica de negocio:** Me gustaría seguir investigando necesidades e intentar encontrar funciones que le sean de una gran utilidad para el usuario pero sin generar un abuso de funciones que puedan "marear" al usuario, hasta el punto de que le pueda parecer pesada o nada intuitiva, para ello hay que encontrar un problema, analizarlo, planificar una solución y trabajar sobre ella para ofrecerle al usuario un herramienta super sencilla que solucione lo que es para él un gran problema.
- **Mejoras de los informes:** También me gustaría generar unos informes más completos, dando la opción también de hacerlos un poco más técnicos o detallados, pero sin perder la intuitividad para el usuario o subiendo el nivel de conocimientos necesarios para el uso de la aplicación.

Modelo de negocio

Este proyecto desarrollado no solo resuelve una necesidad concreta de digitalización y gestión para propietarios e inmobiliarios, sino que también presenta un alto potencial de escalabilidad tanto técnica como comercial, lo que lo convierte en una solución viable para ser ofrecida como servicio.

Desde el punto de vista de negocio, esta aplicación puede crecer mediante:

- **Ampliación del público objetivo:** Puede adaptarse fácilmente para ser utilizada por gestores turísticos, propietarios de alojamientos rurales, empresas de alquiler vacacional... entre otros muchos tipos de alquileres.
- **Modelo SaaS:** Puede comercializarse como una suscripción mensual o anual, permitiendo a propietarios o inmobiliarias gestionar sus inmuebles desde cualquier dispositivo, sin necesidad de instalaciones complejas.
- **Planes escalables:** Ofrecer distintos planes según el número de inmuebles gestionados, funcionalidades disponibles o número de usuarios que acceden.
- **Personalización para empresas:** La aplicación puede ofrecerse como solución personalizada para agencias que necesiten adaptar funcionalidades concretas o integrar con otros sistemas propios.
- **Impuestos de reservas:** Si se realizan la escalabilidad y los inquilinos pueden reservar las propiedades ellos mismo, se puede percibir una pequeña cantidad de la reserva para la aplicación a modo impuesto/tarifa por gestión.
- **Colaboración con empresas:** Se puede negociar con empresas del sector de mantenimientos que les interese aparecer en la página de mantenimientos, a modo mensualidad para aparecer en los mantenimientos como sugerencias de empresas de la zona a los usuarios.

Resultados

Tras el desarrollo de la aplicación, se han conseguido cumplir los principales objetivos planteados en la fase inicial del proyecto, ofreciendo una solución funcional, estable y útil para la gestión de propiedades en alquiler de forma centralizada e intuitiva.

- **Funcionalidad completa y operativa**

La aplicación permite gestionar un conjunto de propiedades inmobiliarias de forma eficaz mediante las siguientes características implementadas con éxito.

- **Gestión de inmuebles:** Alta, modificación y visualización de propiedades con todos sus datos relevantes.
- **Reservas inteligentes:** Sistema de reservas que controla automáticamente la disponibilidad de los inmuebles y genera automáticamente un ingreso para la propiedad según el cobro introducido en la reserva.
- **Mantenimientos:** Registro, seguimiento, planificación de mantenimientos preventivos o correctivos y generación de un gasto automático para la propiedad, de forma que al insertar el coste del mantenimiento se añade como gasto automáticamente en los movimientos de la misma propiedad.
- **Movimientos (Balances económicos):** Sistema de finanzas automático, recogiendo ingresos o gastos, automáticos o manuales por cada propiedad, generando así tanto un balance individual de la propiedad como un balance general de todas ellas.
- **Mapa interactivo:** Visualización geolocalizada de propiedades mediante la integración de Leaflet y OpenStreetMap, permitiendo visualizar todas las propiedades del usuario en un mapa, en el cual también podemos cambiar las capas (vista informativa (predeterminada), vista satélite o vista urbana).
- **Seguridad:** Gestión de sesiones y control de acceso, validación de entradas, protección frente a inyecciones SQL, navegación indebida por URL y la obligatoriedad de generar una contraseña segura para poder crear una cuenta de usuario en PropiGest, evitando así la probabilidad de suplantaciones de identidad y el acceso indebido a información de otro usuario, además de almacenar la contraseña de forma encriptada en la base de datos, lo que imposibilita una que un acceso o filtración no autoriza de la base de datos complique la tarea de los atacantes a la hora de suplantar la identidad y poder acceder a otros usuarios.

- **Interfaz adaptable:** Diseño responsive y ligero, utilizando HTML, CSS, JavaScript y BootStrap, con pruebas realizadas para asegurar una experiencia fluida desde dispositivos móviles y ordenadores.
- **Optimización del rendimiento**

Durante el desarrollo he abordado diferentes desafíos técnicos, como el ajuste dinámico del mapa en dispositivos móviles, el manejo eficiente de sesiones y redirecciones. También se han realizado pruebas para minimizar errores y optimizar el tiempo de carga.
- **Usabilidad y experiencia del usuario**

En el diseño de la interfaz he priorizado la simplicidad, claridad y facilidad de uso, de forma que cualquier propietario o usuario con conocimientos básicos pueda usar la plataforma sin necesidad de formación técnica. El sistema está pensado para reducir el nivel de clics innecesarios y presentar la información de una forma directa y sin esfuerzos innecesarios por no comprender la interfaz.
- **Escalabilidad y potencial comercial**

Además de su funcionalidad actual, la aplicación está diseñada pensando en su evolución como producto comercial, ya que:

 - Puede convertirse en un servicio SaaS, ofreciendo la plataforma a inmobiliarias, empresas de alquiler turístico o gestores de propiedades.
 - El sistema permite una gran escalabilidad a nivel técnico y comercial, incorporando nuevos módulos, usuarios y funcionalidades.
 - Puede replicarse fácilmente en otras zonas geográficas o adaptarse a otros idiomas, ampliando así su alcance comercial y de uso.

Conclusiones

El desarrollo de este proyecto ha representado una experiencia increíble de gran valor formativo y práctico, en la que he podido poner en uso y consolidar muchos de los conocimientos adquiridos no solo durante este ciclo, sino en las prácticas de fct y todo el tiempo que llevo en este mundo de la programación y desarrollo web.

Gracias a esta aplicación, he logrado construir una herramienta funcional, clara y orientada a solucionar una necesidad real: la gestión digital y centralizada de propiedades en alquiler por parte de los propietarios o agencias inmobiliarias. Su diseño modular y adaptable, junto con la interfaz amigable y las funcionalidades desarrolladas, hacen que el sistema sea fácilmente usable, escalable y mejorable.

Uno de los aspectos más enriquecedores ha sido trabajar en el apartado de seguridad, donde he podido implementar mecanismos contra inyecciones SQL, navegación indebida por rutas y URLs, gestión de sesiones seguras y validaciones que garantizan un entorno más protegido frente a errores o ataques comunes.

En el camino he encontrado dificultades, como la asincronía de funciones de las llamadas a backend, la coordinación entre eventos de JavaScript y elementos del DOM o el control de errores y validaciones en formularios, que me han ayudado a desarrollar una mayor autonomía y capacidad de resolución.

Finalmente, este proyecto no solo cumple su función como solución técnica, sino que también abre la puerta a futuras mejoras y aplicaciones reales en el entorno profesional. El hecho de haber planteado una arquitectura escalable permite su evolución hacia un servicio más amplio, que podría llegar a comercializarse como solución de software para gestión inmobiliaria.