

# CFGS DESARROLLO DE APLICACIONES WEB



GENERALITAT  
VALENCIANA

Conselleria d'Educació,  
Cultura i Esport



**Florida**

Universitària

Proyecto Final de Ciclo

## GESTIONA TU GARAJE 2024-2025 GESTIONA TU GARAJE

*Apellidos y nombre del autor/a: David Reinón García*

*Fecha de entrega: 19/05/2025*

## Contenidos

1. Resumen del proyecto: Que se propone y de que está formado .....	3
- Castellano.....	3
-Ingles (Abstract) .....	5
2. Justificación y objetivos del proyecto: Para qué se propone, a quién se dirige y que se pretende conseguir. ....	7
Objetivos del proyecto .....	7
Requisitos del sistema.....	8
3. Desarrollo del proyecto .....	9
a. Análisis del mercado y posible modelo de negocio: qué propuestas similares hay, cuál es el valor añadido de esta propuesta, cómo se podría explotar, etc. ....	9
b. Metodologías Utilizadas.....	13
c. Descripción de los componentes de la aplicación: mockups, estructura de páginas, backend, frontend, tecnologías utilizadas, etc. ....	16
d. Problemas/dificultades encontradas en el desarrollo del PFC, y que soluciones se han buscado para su solución. ....	25
e. Resultados obtenidos .....	26
4. Conclusiones: conclusiones obtenidas tras la realización del PFC, tanto a nivel profesional como personal. ....	27
5. Líneas futuras de trabajo .....	28
6. Bibliografía.....	29

# 1. Resumen del proyecto: Que se propone y de que está formado

- CASTELLANO.

Este Proyecto de Final de Ciclo (TFC) de desarrollo de aplicaciones web que estoy abordando consiste en el desarrollo de una aplicación para la **gestión de un garaje orientado al alquiler de plazas de aparcamiento**, pensada específicamente para ser utilizada en un entorno real siguiendo los requisitos del cliente, en este caso el propietario del garaje, teniendo en cuenta sus preferencias personales a la hora de elegir los componentes visuales y funcionalidades.

El objetivo principal del proyecto es sustituir una gestión mucho más arcaica, que cuenta con todos los datos descentralizados y el flujo de trabajo de 'todo a papel y a mano'. Por eso se ofrece una herramienta que sustituye esto por completo: **intuitiva, funcional y responsiva de fácil acceso desde dispositivos móviles**, pensando en su uso diario desde cualquier lugar.

El sistema ha sido desarrollado utilizando las tecnologías más adecuadas una vez estudiado el caso. **Next.js con TypeScript** para el front-end, junto con **Supabase** como servicio de back-end y gestión de base de datos, usando la capa gratuita. Además, se ha empleado **shadcn/ui**, una librería de componentes de interfaz modernos, pero al mismo tiempo modificables, para agilizar el desarrollo de una interfaz de usuario coherente, atractiva y accesible.

Las funcionalidades claves de la aplicación son:

- **Gestión CRUD de clientes y vehículos**, para poder ver y manipular toda la información necesaria. Teniendo en cuenta los requisitos funcionales proporcionados por el propietario del garaje.
- **Tablas tipo DataGrid** para visualizar la información de la mejor forma, y con las funcionalidades de crear, editar y eliminar entradas, tanto de clientes como de coches, de forma sencilla e intuitiva.
- Una **pantalla de inicio a modo de Dashboard**, que presenta información relevante como plazas ocupadas, número de clientes activos y actividad reciente.

Un añadido con mucho valor, ya que un panel de información te permite poder analizar las distintas épocas y facilitar la toma de decisiones de cara a futuro.

- **Navegación lateral responsive**, con posibilidad de colapsar el menú a solo iconos, facilitando la experiencia desde móviles y dejando espacio para la información importante.
- Una sección de **ajustes** para funciones complementarias como cerrar sesión, acceder a ayuda y futuras extensiones.

El proyecto ha seguido un enfoque **práctico y orientado al usuario final**, con el objetivo de que sea implementado en un plazo corto de tiempo. La **usabilidad y simplicidad** de la interfaz ha sido uno de los objetivos en todo momento, siguiendo principios de diseño centrado en el usuario. Tal y como indica (Krug, 2014), una interfaz efectiva debe ser intuitiva y no requerir esfuerzo cognitivo innecesario por parte del usuario, principio que se ha aplicado en el desarrollo de esta herramienta.

- INGLÉS (ABSTRACT)

This Final Web Application Development Project (TFC) focuses on the creation of a web-based system for managing a parking garage oriented toward the rental of parking spaces. The application is specifically designed for real-world use, based on the client's requirements—in this case, the garage owner—including personal preferences for visual components and functionality.

The main objective of the project is to replace a much more outdated system, where all data is decentralized, and the entire workflow relies on manual, paper-based processes. This new tool is intended to completely replace that method with a solution that is intuitive, functional, and responsive—easily accessible from mobile devices and suitable for daily use from anywhere.

The system has been developed using technologies selected after analyzing the specific needs of the project. **Next.js with TypeScript** was used for the frontend, and **Supabase** for backend services and database management, leveraging its free tier. In addition, the **shadcn/ui** component library was employed to build a modern yet customizable user interface, allowing for a consistent, attractive, and accessible design.

The key features of the application include:

- **CRUD management of clients and vehicles**, enabling access to and manipulation of all necessary information, tailored to the functional requirements provided by the garage owner.
- **DataGrid-style tables** for optimal data visualization, with simple and intuitive options to create, edit, and delete both client and vehicle entries.
- A **dashboard-style home screen** that displays relevant information such as occupied spaces, number of active clients, and recent activity. This acts as a small information panel to help analyze usage patterns and support ongoing improvements.

- A **responsive sidebar navigation**, which can be collapsed to icon-only mode, improving the user experience on mobile devices and maximizing space for key information.
- A **settings section** for complementary functions such as logging out, accessing help, and enabling future extensions.

The project has followed a practical and user-focused approach, aiming for quick deployment and real-life implementation. Usability and simplicity have been central goals throughout development, in line with user-centered design principles. As (Krug, 2014) points out, an effective interface should be intuitive and require minimal cognitive effort from the user—a principle that has guided the creation of this tool.

## 2. Justificación y objetivos del proyecto: Para qué se propone, a quién se dirige y que se pretende conseguir.

El proyecto se ha propuesto para solucionar **un problema real** en base a unas necesidades detectadas, que posteriormente han sido **estudiadas y proyectadas** para ser implementadas en una aplicación web.

La idea es que todo el flujo de eventos del garaje que hay que realizar, como, por ejemplo: que se dé de alta o de baja un coche, que haya que consultar la información de algún cliente para contactar con él o para realizar alguna gestión, que haya que consultar el registro para ver quién había ocupado una plaza durante varios meses, etc. Todos estos procesos que ahora cuestan un tiempo razonable, ya que hay que consultarlo de un papel o de varias fuentes de información (contactos del móvil, documentos, etc.), **reemplazarlo digitalmente** para que cueste pocos minutos, ya que lo tienes accesible en uno o dos clicks, de forma clara e intuitiva.

Va dirigido al **dueño del garaje y/o las personas encargadas de gestionarlo**, facilitándoles el trabajo de manera drástica, con la posibilidad de que, en un futuro, pueda adaptarse a más negocios similares. Ahorrándoles tiempo, aumentando su productividad en las tareas y evitando errores humanos en la medida de lo posible, ya que es casi imposible controlar esto al completo. 'En cualquier entorno laboral, el error humano es una realidad inevitable' (Redsinergia, 2024)

### OBJETIVOS DEL PROYECTO

- Sustituir el sistema de gestión manual por una herramienta digital eficiente, clara y accesible.
- Facilitar el seguimiento de la actividad más reciente (ocupación, clientes activos, entradas/salidas recientes).
- Garantizar la integridad y organización de los datos relacionados con clientes y vehículos.
- Ofrecer una interfaz intuitiva que reduzca la curva de aprendizaje para el usuario final.

- Desarrollar una base tecnológica sólida y escalable para futuras mejoras o integraciones.

## REQUISITOS DEL SISTEMA

Para lograr estos objetivos, el proyecto se estructura en tres niveles de requisitos:

### - **Requisitos mínimos (imprescindibles para el uso básico):**

- Autenticación de usuarios y control de acceso.
- Gestión CRUD completa de clientes.
- Gestión CRUD completa de vehículos asociados a clientes.
- Relación lógica entre vehículos y clientes.
- Visualización de información en tablas claras y navegables.
- Interfaz responsive para uso desde móviles.
- Navegación lateral colapsable.

### - **Requisitos deseables (mejoran la experiencia y funcionalidad):**

- Mini Dashboard con estadísticas clave (plazas ocupadas, actividad reciente...).
- Registro de actividad reciente (creaciones, actualizaciones, bajas).
- Validaciones visuales y feedback al usuario (por ejemplo, confirmación al eliminar).
- Interfaz personalizable con elementos visuales agradables.

### • **Requisitos futuros (ampliaciones o funcionalidades extra):**

- Historial completo de actividad de cada cliente o vehículo.
- Más información de cada plaza con un registro completo en la base de datos.
- Filtrado y búsqueda personalizable dentro de las tablas.
- Generación de informes o exportación de datos.
- Notificaciones (por ejemplo, alertas por plaza próxima a vencer).
- Adaptación a más tipos de negocios de alquiler o gestión (trasteros, naves...).
- Gestión de usuarios con distintos roles (administrador, empleado...).



### 3. Desarrollo del proyecto

a. Análisis del mercado y posible modelo de negocio: qué propuestas similares hay, cuál es el valor añadido de esta propuesta, cómo se podría explotar, etc.

Se ha observado en los últimos años como cada vez está más presente la necesidad de digitalizar las tareas de gestión que hasta ahora se hacían de forma manual basados en papel, como por ejemplo llevar informes, facturas, seguimiento de clientes, entre muchas otras..., por parte tanto de las empresas grandes como también los pequeños negocios, y esto se debe por varias razones principalmente:

- **Coste extra de tiempo y dinero.**

Cada tarea por realizar es mucho más costosa y a la larga eres menos productivo y puedes abordar menos demanda. Que es igual a menos clientes, menos dinero y en conclusión a menos rentabilidad de tu negocio.

- **Minimizar errores humanos.**

Se ha demostrado muchas veces, cuando vemos, por ejemplo, brechas de seguridad graves en las compañías más grandes del mundo, como Apple, Microsoft, etc. que casi siempre son por errores humanos.

Es normal equivocarse ya que somos personas y no robots (hay que decir que hasta la Inteligencia artificial se equivoca). Pero es muy importante plantear como se puede reducir esto al máximo, ya que puede desencadenar desde consecuencias más leves, como que tengas que ajustar o rectificar posteriormente algún dato o informe, hasta muy graves..., como por ejemplo tener una fuga de los datos de tus clientes o transferir información delicada por algún despiste, y que te puedan envolver en un proceso judicial ya que esto puede ser delito.

Por lo cual en una aplicación donde todo esta centralizado, de forma intuitiva te muestra todos los pasos que tienes que hacer por cada tarea, y existen las validaciones que avisarán al usuario cada vez que haya algún error, es algo muy valioso e imprescindible.

- **Escalabilidad.**

Esto puede aplicarse tanto a empresas muy grandes (para serlo aún más) hasta particulares para darle más valor al negocio. Cuando cada vez cuentas con más clientes y más tareas a realizar (dar de alta, de baja, facturas, etc.) llega un momento que de forma manual no puedes crecer más, y este tipo de aplicaciones te solucionan este problema.

- **Comodidad a nivel personal**

En el día a día de cualquier trabajo, si cada pequeña tarea que tenemos que hacer sabemos que nos va a costar un tiempo razonable, y que algo que en un sistema automatizado nos puede costar media hora, manualmente nos llega, por ejemplo, a costar el triple de tiempo, acaba siendo frustrante y sintiéndote poco productivo. Las personas queremos tener éxito profesional y ganar mucho dinero, pero, sobre todo, a nivel personal, estar los más cómodos en nuestra rutina, en nuestro día a día, y contar con los recursos y herramientas que mejor nos puedan hacer desempeñar nuestro trabajo.

No es un punto que vaya directamente con una razón enfocada al nivel empresarial ni de negocio, pero es una conclusión personal que creo que tiene su gran parte de sentido.



FIGURA 1: RESUMEN DE ALGUNOS PASOS PARA EMPEZAR A DIGITALIZAR DOCUMENTACIÓN

Podemos encontrar muchas propuestas en el mercado relacionadas con garajes para estacionar y plazas de aparcamiento, ya que es un servicio que está muy presente en el mundo actual y cuenta con mucha demanda.

Investigando un poco vemos plataformas muy conocidas como **Parclick**, **Telpark** o **ElParking**, que ya llevan varios años estableciéndose en el sector, pero estas junto la mayoría que he encontrado más conocidas, son softwares bastantes complejos y enfocados sobre todo al usuario final, te ofrecen el servicio de identificar las plazas más cercanas dependiendo de tu ciudad, reservar, entradas express, la posibilidad de pagar a través del móvil, los puntos con carga eléctrica, etc. No están dirigidas para un dueño de un garaje con plazas de aparcamiento en el cual el o varias personas de confianza llevan su propia gestión.

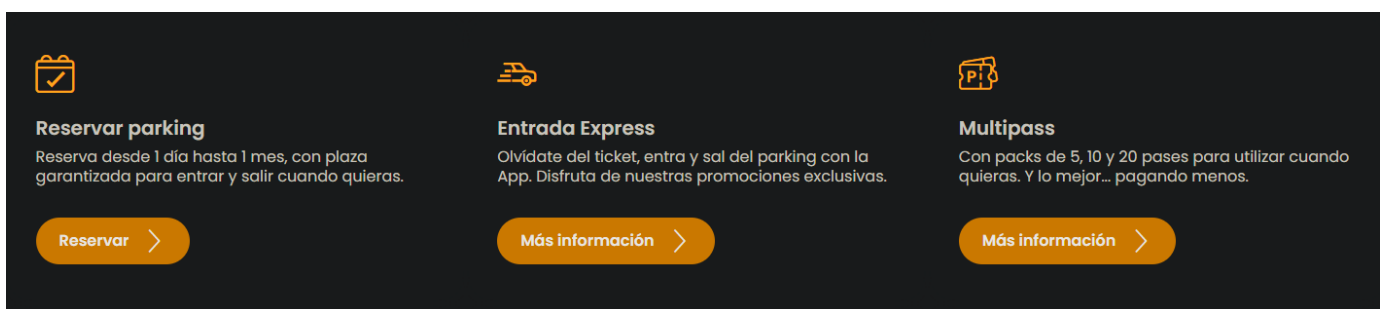


FIGURA 2: ALGUNOS PRODUCTOS QUE OFRECE 'TELPARK' Y PROMOCIONA EN SU WEB

En el caso de mi proyecto '*Gestiona Tu Garaje*' la propuesta es distinta, y su principal valor es que se trata de una herramienta pensada **específicamente para el propietario de pequeños garajes** o servicios de este tipo, que necesita llevar un control exhaustivo de sus clientes, vehículos y ocupación, con una solución intuitiva y económica. Al usar tecnologías modernas y gratuitas, como con el gestor de Base de Datos Supabase en su capa gratuita, permite mantener bajos los costes de desarrollo y despliegue.

Aunque actualmente se plantea como un proyecto académico y una solución puntual para un cliente real, el sistema podría evolucionar hacia un modelo de negocio basado en:

- **SaaS (Software as a Service)** para pequeños garajes: permitiendo que cualquier gestor acceda a una versión personalizada por una suscripción mensual básica.

- **Modelo freemium**, donde funciones básicas sean gratuitas y otras avanzadas (como informes automáticos, multiusuario, estadísticas históricas o implementación de IA) formen parte de un plan de pago.
- **Distribución bajo licencia** para grandes empresas que gestionen múltiples garajes.
- **Servicios de personalización** para negocios con funcionamiento similares (trasteros, almacenes, etc.).

Este análisis demuestra que el proyecto no solo es viable a nivel técnico, sino que cuenta con un **nicho potencial** poco cubierto en el mercado actual, ofreciendo una alternativa adaptada, escalable y de bajo coste.

## b. Metodologías Utilizadas.

La principal metodología que he utilizado para el desarrollo de este proyecto de desarrollo web es un **marco de trabajo ágil** llamado **SCRUM**, junto al **tablero Kanban** para la organización de cada tarea y en qué estado se encontraba cada una.

Hoy en día es inimaginable trabajar en un proyecto, sobre todo de desarrollo de software, sin usar una metodología agile como SCRUM entre otros, que te ofrece una **flexibilidad y adaptabilidad** concreta que no te ofrecen otros marcos de trabajo.

La metodología ágil beneficia a los equipos al permitir la planificación adaptativa, la ejecución rápida y la evaluación continua, lo que lleva a mejores resultados.  
(Atlassian, 2025)

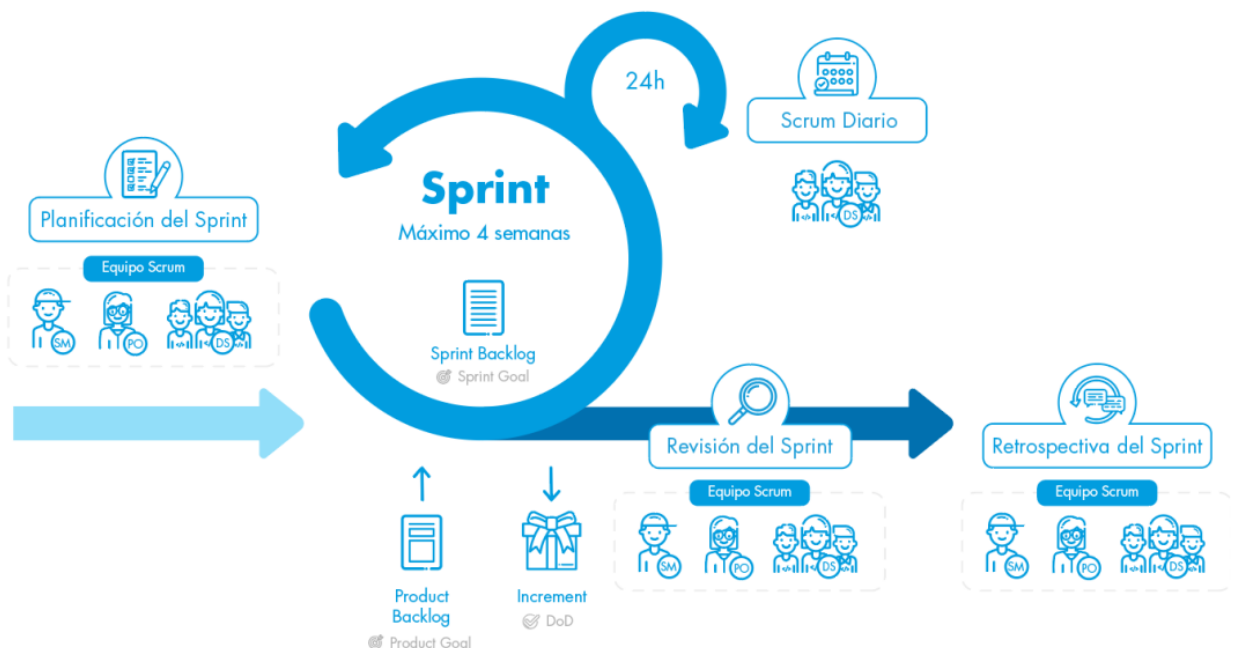
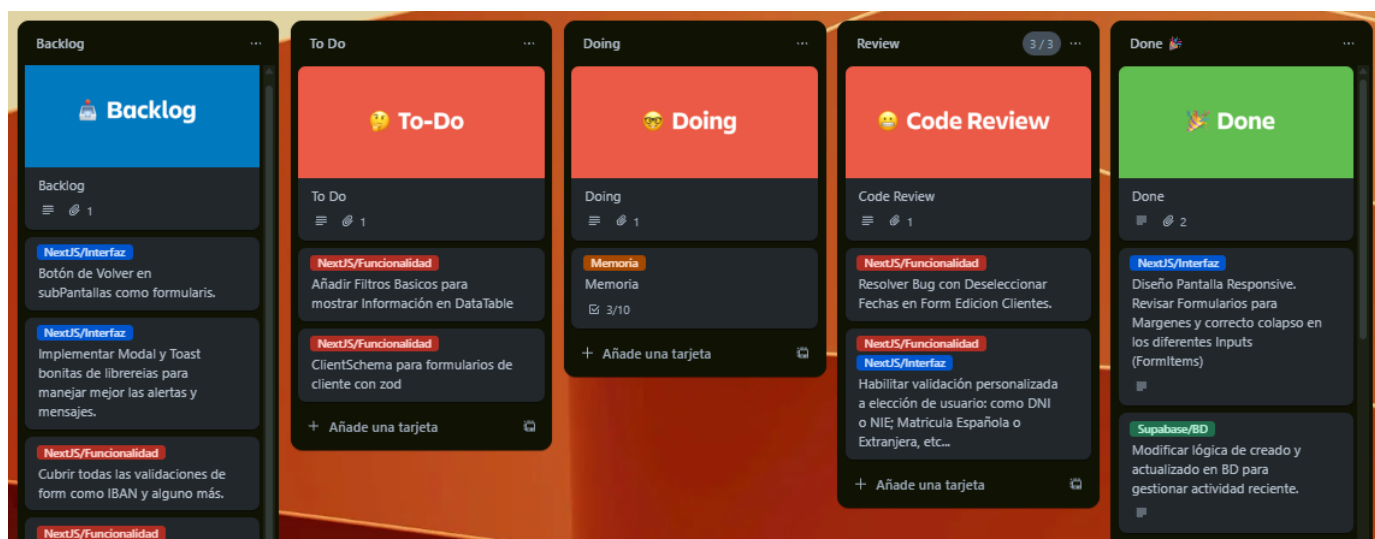


FIGURA 3: ESQUEMA QUE REPRESENTA LAS FASES Y ELEMENTOS DE UN SPRINT EN EL MARCO SCRUM

- **SCRUM** se caracteriza por que mediante los '**sprints**', que son iteraciones cortas y regulares para desarrollar unas tareas concretas que se seleccionan antes de empezar del **backlog** (pila de tareas a realizar durante el desarrollo al completo), siempre se cuenta con una entrega continua y una versión de valor de la aplicación que se está desarrollando, pudiéndola **mejorar constantemente** con el feedback del cliente. **Dejando atrás las metodologías en cascada**, en las cuales el cliente probaba directamente la aplicación en una de sus versiones finales, donde se daba la situación en la cual pedía varios cambios, que para ser realizados había que deconstruir mucha lógica de la aplicación que se pensaron con un enfoque distinto al que se quería llegar, perdiendo tiempo y dinero, que se podría haber ahorrado con una comunicación constante.

En este caso para el desarrollo del PFC, al ser un proyecto individual de cada alumno, se tenían que **adaptar los elementos** que están pensados para un grupo de trabajo donde cada uno tiene su propósito, a una sola persona. Por lo cual los alumnos tienen que asumir también el rol de **Producto Owner** (quien se responsabiliza de su Proyecto, quien gestiona y organiza su backlog o pila de tareas pendientes) y el tutor que revisa nuestros trabajos en cada reunión de sprint será el **SCRUM Master** (quien supervisa y modera dicha reunión, aconsejando en aquellos puntos en los que crea conveniente)

- **Kanban** tiene mucho sentido en el contexto actual, ya que se adapta fácilmente a la forma de trabajar que usamos, y como he dicho mencionado anteriormente te ayuda drásticamente en la organización y realización de las tareas.



**FIGURA 4: TABLERO KANBAN DEL DESARROLLO DE LA APLICACIÓN 'GESTIONA TU GARAJE'**

Como vemos en la Figura proporcionada, que es una imagen real del tablero de mi proyecto, se trata de un tablero organizado por varias fases (de base siempre tiene 'To Do', 'Doing' y 'Done' y le he añadido algunos más) donde dentro tiene varias tarjetas que son cada tarea para realizar, dependiendo de en qué estén se moverán de una fase a otra, pudiendo saber tus avances, cuales están en proceso o cuales son necesarias revisar de cara al futuro.

### C. Descripción de los componentes de la aplicación: mockups, estructura de páginas, backend, frontend, tecnologías utilizadas, etc.

- **Mockup:**

El mockup ha sido una parte muy importante en la **parte de diseño y experiencia de usuario** para poder plasmar la idea principal del proyecto y **consensuar de forma rápida** que elementos veía más accesible el propio cliente. Aunque lógicamente la versión final de la app ha cambiado en muchos aspectos, el mockup ha sido necesario, cumpliendo totalmente su finalidad como guía de desarrollo.

Para crear esto he utilizado **Figma**, una de las plataforma **más usadas y conocidas hoy en día** para casos de diseño de software. Ya que cuenta con una interfaz muy intuitiva, al mismo tiempo de muchas opciones de ordenación de elementos, y con un plan totalmente gratis.

Personalmente, al haberlo gastado anteriormente para otros proyectos y trabajos académicos, la curva de aprendizaje era mucho menor que con otro software nuevo, y al contar con tan poco tiempo para realizar el proyecto completo, era sin duda la mejor opción.

Enlace para visualización en alta calidad del wireframe para 'Gestiona Tu Garaje':  
[WIREFRAMER - GARAGE MANAGEMENT – FIGMA.](#)

- **Estructura de páginas:**

La aplicación '**Gestiona Tu Garaje**' ha sido diseñada como una **herramienta de gestión** y no como una landing page o sitio web informativo orientado al público general. Su objetivo es facilitar al administrador del garaje la administración diaria de clientes, vehículos y plazas ocupadas, mediante una interfaz tipo **dashboard**, con enfoque educativo. Esto implica una estructura centrada en la claridad y facilidad de acceso a las funcionalidades clave, más que en un diseño visual orientado a captar la atención de los usuarios.



## - Navegación/Sidebar:

El elemento principal de cualquier aplicación, con el cual nos movemos por todas las pantallas y podemos acceder a todas las funcionalidades disponibles.

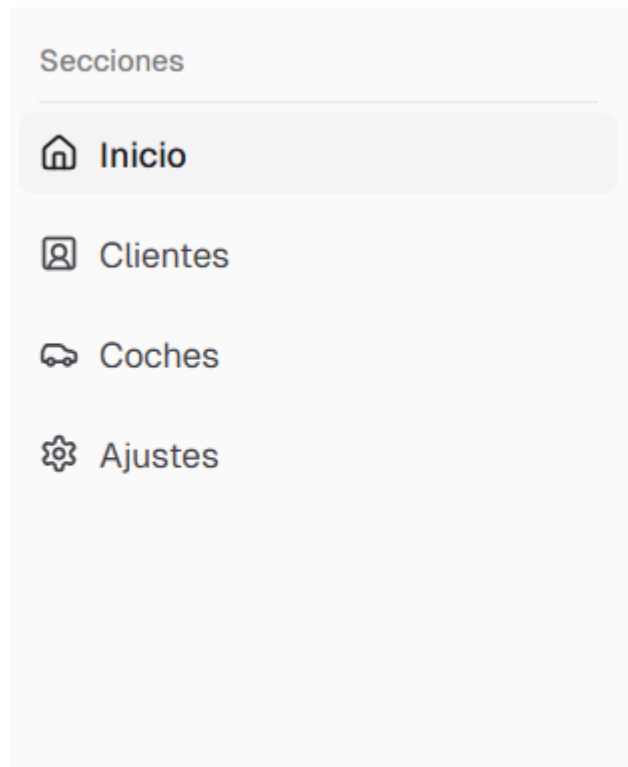
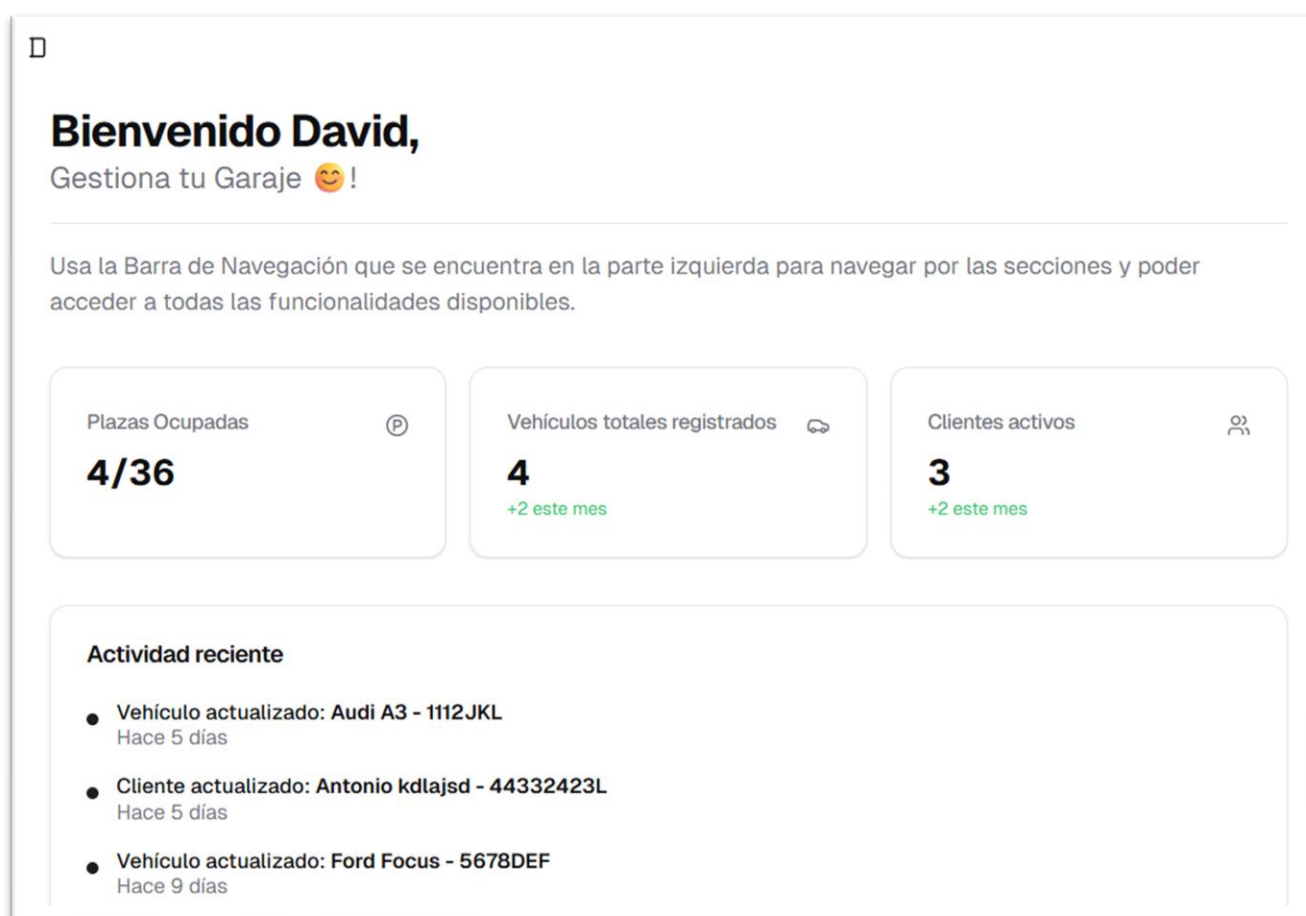


FIGURA 5: BARRA LATERAL DE LA APLICACIÓN GESTIONA TU GARAJE

En mi caso concreto, para '**Gestiona Tu Garaje**' he decidido usar una barra lateral (sidebar) ya que lo realmente importante es que la interfaz sea intuitiva y educativa. Esta barra está pensada para no quitar foco a la información de la página principal, incluso en vista de móvil se esconde automáticamente y para verla tendremos que activarla con un botón, liberando así espacio en pantallas más pequeñas.

- **Contenido Principal:**

Encontramos las distintas funcionalidades de la aplicación, sin distracciones ni diseños extravagantes. Desde la página de inicio con a modo Dashboard con un panel de la información principal, las páginas de gestión de clientes y coches relacionadas entre sí, hasta la página de ajustes con información de valor y la opción de cerrar sesión en tu cuenta.



**FIGURA 6: PANTALLA DE INICIO DE LA APLICACIÓN GESTIONA TU GARAJE**

## Cientes

Aquí puedes ver la lista de clientes registrados.

+ Añadir

Nombre	Apellidos	Teléfono	Email	Fecha Entrada	Coche	Matrícula	Acciones
Carlos	Garcia	987654213	carlos.garcia@example.com	31/1/2025	BMW X5	9101GHI	 
Antonio	kdlajsd	432094923		4/5/2025	-	-	 
Juan	Pérez	123456789	juan.perez@example.com	15/7/2025	Toyota Corolla	1234ABC	 
Maria	Lopez	987654321		28/2/2025	Ford Focus	5678DEF	 

FIGURA 7: DATAGRID DE LA PÁGINA DE CLIENTE DE LA APLICACIÓN GESTIONA TU GARAJE

## Ajustes

Aquí puedes gestionar tu cuenta o encontrar información de soporte.

Versión App  
1.0.0

Soporte  
Contacta con el administrador

Cuenta  
Usuario: David  
Email: davidreinondev@gmail.com  
ID: f22e7697-5ebc-453d-b6d4-054cf7e0e5ab  
Fecha de Creación: 16/5/2025

 Cerrar sesión

FIGURA 8: PANTALLA AJUSTES DE LA APLICACIÓN GESTIONA TU GARAJE

- **Backend:**

Lo que he utilizado para la gestión de todo el backend de mi proyecto ha sido las distintas herramientas que ofrece **Supabase**.

**Supabase** es una plataforma **BaaS (Backend as a Service)** alojada en la nube, donde puedes gestionar principalmente BBDD y más servicios de backend muy útiles y con un gran potencial, como autenticación con todo tipo de proveedores, almacenamiento en buckets, eventos real time, etc. Es una **alternativa Open Source** muy conocida a **Firebase de Google**, pero gratuita y muy apoyada por la comunidad de desarrollo.

Cuenta con una **capa gratuita** donde puedes tener hasta dos proyectos en tu cuenta completamente gratis, y es perfecto para proyectos personales o que tienen una demanda pequeña/mediana. Lo mejor es que ofrece prácticamente todas las funcionalidades disponibles que en el plan de pago, lo único que cambia, lógicamente, es la potencia y el límite de almacenamiento, pero recalco que para proyectos que no están pensados a escalar mucho ni muy rápido es ideal.

Respecto a la organización de mi proyecto, después de estudiar las necesidades y lo que quería el cliente, se empezó a realizar los esquemas y a plantear como finalmente se querían relacionar y guardar los datos.

Finalmente se llegó a la conclusión de que necesitábamos una tabla de **clientes** (personas que hacen uso del aparcamiento) y otra de sus **coches**, relacionadas entre sí, con la mayor información de valor en cada tabla. Podemos ver en la siguiente figura el diagrama en su versión final de las tablas.

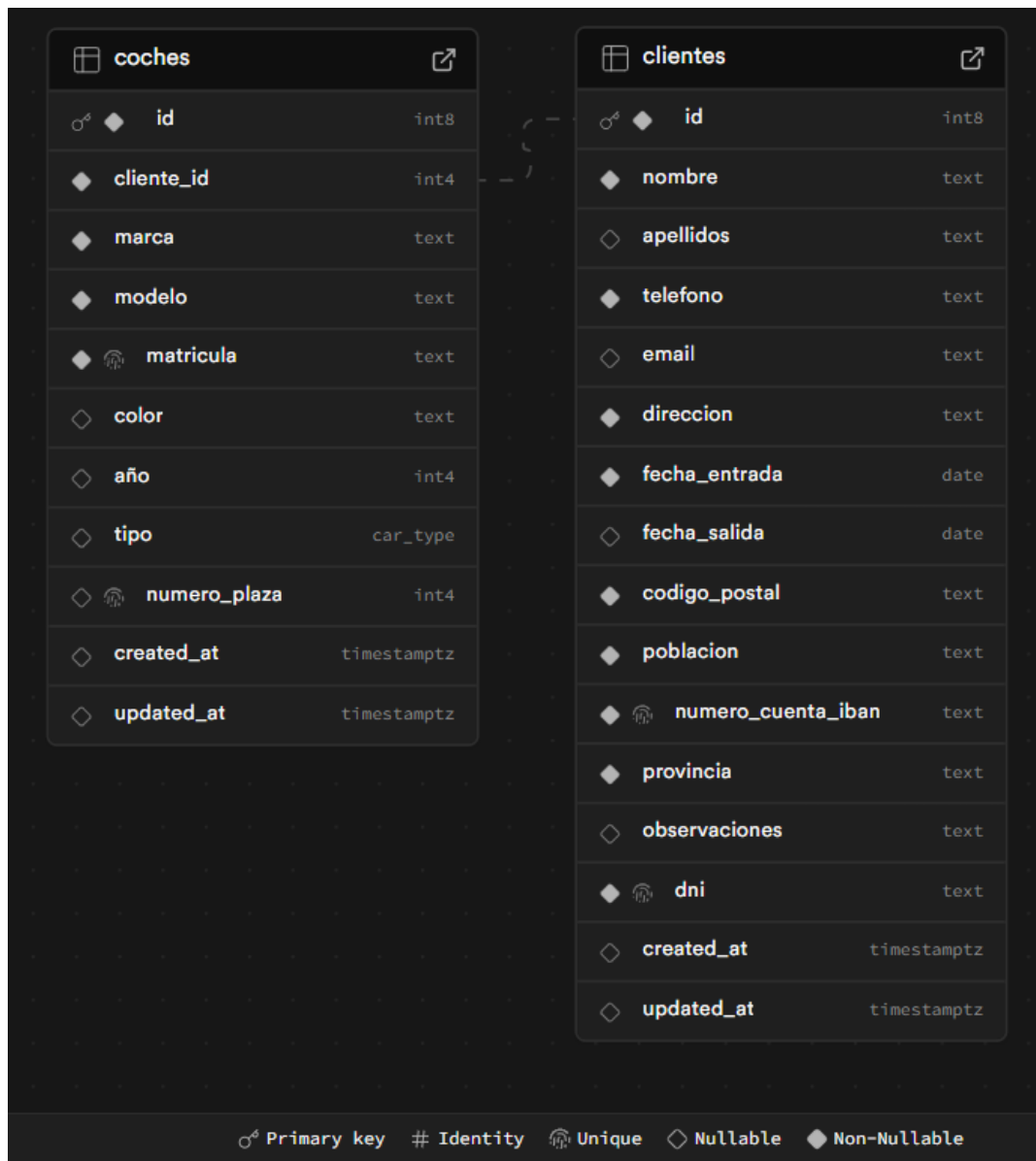


FIGURA 9: IMAGEN DEL DIAGRAMA FINAL DE MI BASE DE DATOS

Por otro lado, como se ha podido ver en los puntos anteriores, también he implementado en la página de inicio de la aplicación web, un **panel con información** y datos más recientes a modo de Dashboard, que creo que, aunque no es la principal funcionalidad, lleva la app un paso más allá, **dándole valor**, y ayudando también al dueño del garaje a **saber cómo transcurre el negocio** y los últimos movimientos que se han realizado. Menciono esto, ya que el llevar esta tarea acabo ha resultado ser **más costoso desde el backend** que desde el frontend, ya que aparte de hacer llamadas más complejas con varios filtros, sobre todo era el hecho de que se han tenido que modificar las tablas para **añadir campos como fechas de creación y actualización** de cada registro, que se rellenan automáticamente con triggers y funciones personalizadas en la BD.

En la siguiente figura se puede apreciar los triggers que hay actualmente cubriendo esto, Supabase nos ofrece una interfaz muy completa cuando los creamos y nos muestra con claridad las características: como a que tabla afectan, que función ejecuta, cuando se disparan o si están habilitados.

Name	Table	Function	Events	Orientation	Enabled
update_client_updated_at	clientes	update_client_updated_at_column	BEFORE UPDATE	ROW	✓
update_coche_updated_at	coches	update_coche_updated_at_column	BEFORE UPDATE	ROW	✓

FIGURA 10: TRIGGERS EN MI BASE DE DATOS ACTUAL

- **Front-end:**

Para el desarrollo de la interfaz de usuario y la lógica del front-end '**Gestiona Tu Garaje**', se ha optado por utilizar **Next.js** junto con **TypeScript**, tecnologías muy usadas hoy en día por su eficiencia y escalabilidad, que son perfectamente compatibles con mi back-end en Supabase, de hecho, por regla general se suelen implementar en conjunto.

Next.js es un **framework reactivo** basado en **ReactJS** que nos ofrece algunas características extras como **enrutamiento directo** mediante la estructura del propio proyecto, generación estática y dinámica de páginas, y **optimización de la carga** desde el lado del servidor (SSR), algo que resulta ideal para aplicaciones tipo dashboard donde se gestionan múltiples vistas internas.

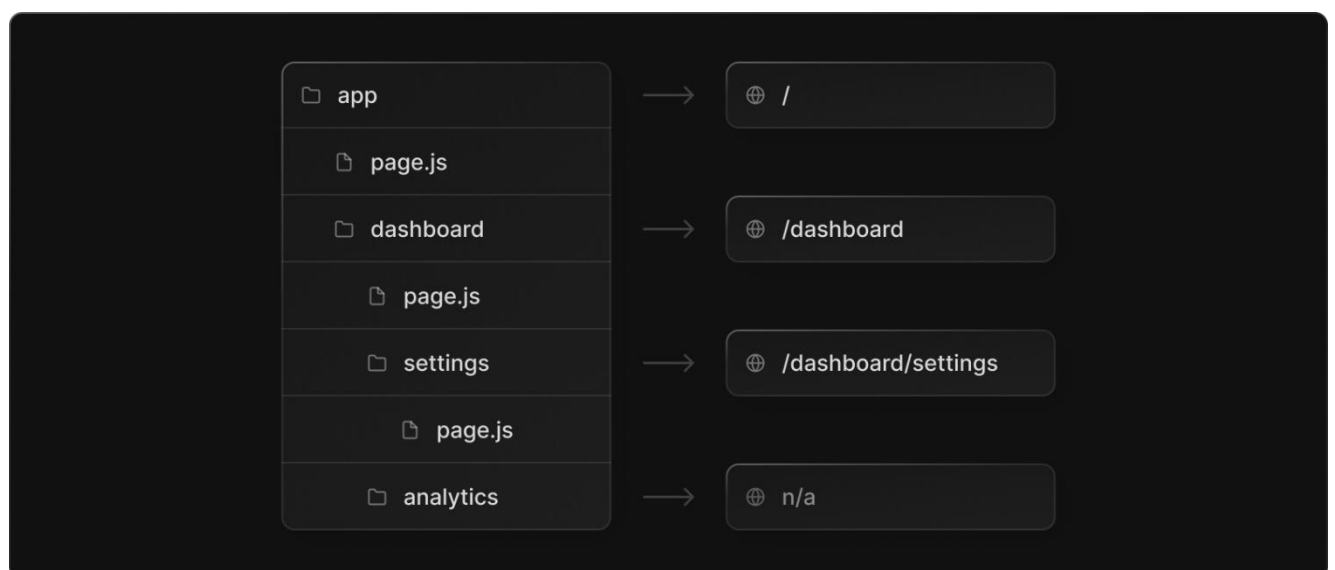


FIGURA 11: EJEMPLO VISUAL DEL ENRUTAMIENTO POR DIRECTORIOS

Y además usándolo con TypeScript, que es como **JavaScript fuertemente tipado**, tenemos un **control infinitamente mejor** del flujo de los datos dentro de la aplicación, y es todo mucho más entendible. Y aunque la curva de aprendizaje puede ser alta al principio, si en un futuro se quiere escalar el desarrollo por ejemplo a un equipo de varias personas, el código será mucho más accesible y propenso a la detección rápida de errores, al contar con el tipado tan estricto que ofrece.

Para el diseño visual se ha utilizado **Tailwind CSS**, un framework basado en el tradicional CSS que permite componer estilos de forma rápida y estructurada directamente desde las clases de los componentes de React. Esto facilita una construcción más ágil del frontend, especialmente en este tipo de proyectos donde el diseño no busca ser extremadamente personalizado, pero sí limpio y coherente.

Adicionalmente, se ha incorporado la librería **shadcn/ui**, que ofrece componentes con la ventaja de ser personalizables y consistentes con el estilo de Tailwind CSS. Esta combinación permite tener todos los elementos bien integrados entre sí.

- **Control de Versiones**

He usado la tecnología **Git**, junto a **GitHub** para poder gestionar el reportorio en la nube, y tener acceso a el historial de versiones en todos los dispositivos. Desarrollar cualquier programa usando Git es esencial para tener un control de todos los cambios y poder volver atrás y revisar cualquier versión de nuestra aplicación si es necesario.

Git es una de las herramientas de control de versiones **más populares**, y GitHub es una plataforma basada en la web que utiliza Git para gestionar repositorios de código (Quintana, 2024)

Es esencial sobre todo si el proyecto se trabaja juntamente con un equipo de desarrollo (que es como normalmente se trabaja), para poder hacer uso de las diferentes ramas en local y remoto, que no dejan ser bifurcaciones del código para no entorpecerse entre sí en el caso de que estemos tocando los mismos archivos. Aunque no es el caso de este proyecto que se hace de forma individual, sigue tiene grandes ventajas y proporciona más información del proyecto de **cara a futuros Sprints y flujos de trabajo**. Por lo cual es algo **imprescindible** a la hora de programar cualquier idea.

Graph	Description
<b>main</b> <i>origin</i> origin/HEAD	<b>Añadidas imágenes de información sobre la estructura y elementos de la base de datos</b> feat: add background image in sign in screen feat: improve some styles and fix imports routes feat: add Ajustes page for user account management and logout functionality chore: unused auth pages moved feat: login implemented with global context and global session validation feat: Change structure to manage user session with different layouts feat: authentication and global user context in process refactor: created home page component chore: Add reset password screen to protected route

**FIGURA 3: CAPTURA DE LOS ÚLTIMOS CAMBIOS DE MI PROYECTO**



#### d. Problemas/dificultades encontradas en el desarrollo del PFC, y que soluciones se han buscado para su solución.

En general he podido desarrollar el PFC sin problemas muy graves que me hicieran tener que reestructurar mucho código, pero sí que he tenido varios inconvenientes, como es normal, en cualquier proyecto de desarrollo de software y destacaría los siguientes:

- **Curva de Aprendizaje alta en las tecnologías implementadas.**

No tengo ninguna duda de que las tecnologías y herramientas que he utilizado para desarrollar y gestionar todo el proyecto han sido una gran elección, porque han cumplido con creces lo que se esperaba y finalmente han hecho que desarrolle un proyecto mucho más profesional tanto para el cliente final como técnicamente hablando.

Pero eso no quita que tengan algunas partes complejas de usar y entender, y al no tener mucha experiencia desarrollando con estas (sabía lo básico y sus principales funcionalidades) llegue un punto en el cual te salen muchos errores. Porque intentas implementar algo que no se puede hacer de esa forma, ya que estas acostumbrado a otro lenguaje, o tienes que pasar mucho tiempo leyendo documentación y aprendiendo con algunos tutoriales para hacer cosas más complejas.

- **Requerimientos del cliente**

En general no ha pasado mucho en este periodo de tiempo, pero algunas veces se ha intentado seguir la metodología SCRUM con sprints cada 2-3 semanas para tener una entrega continua, y mucha comunicación con el cliente para poder ir cambiando las cosas a tiempo y que nos quedemos satisfechos las dos partes con la versión final de la aplicación, pero aun así no siempre se ha conseguido.

Ya que los clientes o las personas a las cuales va dirigida la app, muchas veces no sabes lo que necesitan exactamente, ni se hacen una idea de que tenemos que hacer para implementar todas las soluciones. Por lo cual hay algunas veces que te piden una cosa, luego no era lo que se esperaban o en

realidad lo querían de otra forma, etc y tienes que cambiar y reestructurar muchas cosas que ya tenías avanzadas.

#### - **Gestión del Tiempo**

Al trabajar de forma individual y contar con un periodo largo puedes confiarte, y si no defines bien exactamente tus necesidades y que flujo de trabajo tienes que seguir muchas veces te puedes perder, y gastar demasiado tiempo en funcionalidades o detalles totalmente innecesarios, para luego ir con el tiempo de entrega muy justo.

Tampoco es algo que me haya pasado en exceso, pero sí que es verdad que ha habido alguna reunión del sprint que al enseñar mis avances me he podido dar cuenta de que había gastado demasiado tiempo en 'tonterías' y cosas que al final el usuario final no tiene muy en cuenta.

Aparte de desarrollar y hacer tareas, de vez en cuando es necesario hacer retrospectiva de en qué punto te encuentras, y que cosas son más necesarias en ese preciso momento.

### e. Resultados obtenidos

El resultado final del proyecto creo que ha sido muy exitoso, ya que se han cumplido por completo con **todos los requisitos que se plantearon** desde el principio, cubriendo los especificados al principio de la memoria, tanto los **mínimos** como los **deseables**, teniendo una **versión final muy sólida y funcional**, que, para ser la primera versión y prácticamente una beta, ahora mismo ya puede empezar a ser usada por el cliente (dueño del garaje).

A nivel más técnico que es la parte que menos se ve, creo también que se han cumplido totalmente con las expectativas, se ha conseguido sacar adelante la solución con una **muy buena base**, siguiendo en todo momento **buenas prácticas y código limpio**, implementando de forma correcta todas las tecnologías y recursos al completo. Seguro que sigue habiendo muchas cosas por mejorar, porque siempre se puede mejorar, pero la deuda técnica ahora mismo será muy baja y la base del código está totalmente preparada para poder seguir desarrollando más funcionalidades y **escalar el proyecto sin ningún tipo de problema**, solo o con un equipo de desarrollo que se sume.

## 4. Conclusiones: conclusiones obtenidas tras la realización del PFC, tanto a nivel profesional como personal.

Las conclusiones obtenidas después de la realización del PFC son las siguientes:

**A nivel profesional**, ha sido una gran experiencia para saber gestionar la creación desde cero de un proyecto real y medianamente complejo para el nivel actual que tenemos.

Se siente un **reto superado** el hecho de haber seguido las distintas metodologías de trabajo y haber podido sacar el máximo partido a tu tiempo, el aprendizaje de nuevas tecnologías y la resolución de errores e inconvenientes. Cada día ves como dominas más ciertos aspectos y vas dando pasitos hacia delante para tu futura carrera laboral.

**A nivel personal**, me llevo un sabor agridulce, ya que estoy muy orgulloso del trabajo que he hecho y de todo lo conseguido, pero ha sido **difícil gestionarlo**.

Después de estar trabajando alrededor de 8 horas, ponerte a programar de nuevo en un proyecto importante tu solo, se hace pesado. Tienes muchas cosas que hacer y que aprender, y darle mucho cariño si realmente quieres que salga algo de lo que este orgulloso.

Esto te quita tiempo también para salir a la calle y hacer cosas que forman parte de ti, que quieras que no es un punto muy importante. Es un esfuerzo que hay que asumir en un periodo corto de tiempo, pero hay que estar mentalizado que conlleva mucho trabajo, el adaptarte a un entorno real de trabajar en una empresa y además hacer paralelamente el PFC.

## 5. Líneas futuras de trabajo

El proyecto tiene muchas posibilidades de crecimiento de aquí al futuro en términos de mejora técnica, como de idea de comercialización y planes de negocio.

Al ser un proyecto para un cliente real, queda seguir puliendo esta primera versión final, que, aunque sea sólida se puede mejorar bastante. Sería muy interesante hacer una reunión con el dueño del garaje para ver que futuras mejoras le gustaría, dentro de las que ya están especificadas en los requisitos futuros u otras nuevas.

Por ejemplo, **desde la vista de desarrollador** sería muy interesante empezar a crear **elementos de filtrado y búsqueda** dentro de los DataGrid, aunque no hay muchos registros ya que es una solución para un particular (un negocio a pequeña escala), y no es muy difícil encontrar un registro de un cliente o un coche en concreto, esto mejoraría notablemente la **experiencia de usuario** y le daría un valor añadido al programa.

Por otro lado, más estructural se podría hacer un replanteamiento de la base de datos para concretar más detalladamente que datos contienen las **columnas de actividad reciente, que nutren el Dashboard**, o crear una tercera tabla enfocada solo a las plazas de garajes como tal, y que estas cuenten con información extra de por ejemplo todos los coches que han pasado por esa plaza, las dimensiones, etc.... Ya que ahora mismo es simplemente un campo de la tabla coches, que para nada es un error ya que era lo más lógico y correcto en el planteamiento inicial.

Respecto al **plano de comercialización** como he comentado anteriormente en el punto de plan de negocio, es una aplicación que puede nutrir a un **niche con gran potencial**, y que ahora mismo está hecha a medida para un solo cliente, pero que en un futuro se podría centralizar para que todo tipo de particulares y empresas pequeñas lo puedan utilizar para sus garajes de aparcamiento. Y quién sabe en un futuro más lejano, si la aplicación triunfa, poder contar con un buen equipo de desarrollo y **centralizar la aplicación para todo tipo de servicios de alquiler** como por ejemplo trasteros, o plantas bajas, aparte de plazas de parking para coches.

## 6. Bibliografía

Atlassian. (Mayo de 2025). *¿Qué es la metodología ágil?* Obtenido de <https://www.atlassian.com/es/agile>

Krug, S. (2014). *Don't Make Me Think, Revisted: A Common Sense Approach to Web Usability*. New Riders.

Quintana, G. (31 de Julio de 2024). *Cómo Usar Git y GitHub para el Control de Versiones*. Obtenido de <https://latincloud.com/blog/como-usar-git-github-para-control-versiones/>

Redsinergia. (Marzo de 2024). *Error humano: Cómo minimizar y gestionar desafíos*. Obtenido de <https://redsinergia.com/error-humano/>