



---

**TRABAJO DE FIN DE GRADO 2º DAM DE ADRIÁN GARCÍA JIMÉNEZ**

---

***MyCarCheck***



**3 DE DICIEMBRE DE 2025**

IES VALLE INCLÁN  
Tutor: Andrés Maroto

## Contenido

Introducción:.....	4
Requisitos Funcionales: .....	6
Requisitos no Funcionales: .....	7
Diagrama de clases – Explicación técnica: .....	8
Modelos de datos: .....	8
Persistencia: .....	8
Actividades visuales: .....	8
Adaptadores: .....	9
Relaciones clave:.....	9
Diagrama UML: .....	10
Diagrama de Gantt: .....	11
Tabla de presupuesto estimado:.....	12
Diagrama de casos de uso:.....	13
Diagrama relacional de la base de datos: .....	15
Estructuras de las tablas:.....	16
Tabla usuario:.....	16
CAMPO .....	16
TIPO.....	16
CLAVE.....	16
DESCRIPCIÓN .....	16
<b>idUsuario</b> .....	16

INT .....	16
PK .....	16
Identificador único .....	16
<b>Nombre</b> .....	16
TEXT .....	16
Nombre de usuario.....	16
<b>Contraseña</b> .....	16
TEXT .....	16
Contraseña cifrada.....	16
Tabla Repostaje: .....	16
Tabla Vehículo:.....	17
Manual de usuario:.....	18
Bienvenida .....	18
Primeros pasos.....	18
Menú principal .....	18
Añadir y gestionar vehículos.....	18
Registrar repostajes.....	19
Consultar estadísticas .....	20
Cerrar sesión o eliminar cuenta .....	20
Consejos prácticos .....	20
En resumen .....	20
Manual de uso:.....	21
Navegación entre pantallas: .....	21

Pantalla del login: .....	21
Pantalla de registro: .....	22
Menú principal: .....	23
Pantalla de agregar vehículo:.....	24
Pantalla para ver vehículos:.....	26
Pantalla de estadísticas: .....	30
Botón de cerrar sesión: .....	30
Botón de eliminar cuenta:.....	30
Conclusiones:.....	31

## **Introducción:**

El presente proyecto, titulado ***MyCarCheck***, surge como solución a una necesidad cotidiana: disponer de una herramienta digital que permita llevar un control organizado y accesible del mantenimiento y repostajes de un vehículo personal. Como usuario habitual de coche, he detectado que muchas personas no registran sus gastos de combustible ni el estado de su vehículo, lo que dificulta la planificación de revisiones, el control del consumo o la detección temprana de anomalías. Esta aplicación permite llevar un registro diario de todos los gastos derivados del uso habitual del vehículo, facilitando una gestión más eficiente y responsable.

En un contexto donde la digitalización de procesos cotidianos es cada vez más común, disponer de una herramienta que centralice el historial de mantenimiento y consumo resulta especialmente útil. Este tipo de soluciones no solo mejoran la organización personal, sino que también contribuyen a una conducción más consciente y sostenible.

Este proyecto se ha desarrollado como parte del módulo de Proyecto Final del Grado Superior en Desarrollo de Aplicaciones Multiplataforma. He elegido esta temática porque me permite aplicar conocimientos clave adquiridos durante el ciclo y otros que he ido aprendiendo según avanzaba la aplicación, adentrándome en nuevos lenguajes de programación como Kotlin, los cuales nunca había utilizado previamente. Migrar todo el código anterior a este lenguaje me ha permitido mejorar la escalabilidad del proyecto y trabajar con una tecnología más versátil y potente orientada al desarrollo móvil.

Además, he profundizado en el uso de Android Studio como entorno principal, entorno que me gustaría que fuese el punto de partida para mi especialización como desarrollador móvil. A lo largo del desarrollo, he aprendido a crear interfaces funcionales, gestionar bases de datos SQLite, implementar control de versiones con Git y GitHub, y documentar técnicamente el software de forma estructurada.

El objetivo principal del proyecto es desarrollar una aplicación funcional, intuitiva y técnicamente sólida que permita al usuario registrar repostajes, gastos y revisiones de su vehículo, ofreciendo además métricas como el consumo medio y el coste por kilómetro. Todo ello con una interfaz clara y una estructura de datos bien definida.

Para llevar a cabo el proyecto, he combinado el uso de herramientas como GitHub para el control de versiones y Android Studio como entorno principal de desarrollo. Esta forma de trabajo me ha permitido detectar errores, documentarlos y corregirlos de forma progresiva.

Este proyecto no solo me ha permitido consolidar conocimientos técnicos, sino también desarrollar una actitud crítica frente a los errores, mejorar mi capacidad de documentación y reforzar mi autonomía como desarrollador. Ha sido una experiencia que me ha motivado a seguir explorando el desarrollo móvil como posible especialización profesional.

## **Requisitos Funcionales:**

Los requisitos funcionales definen las acciones que el sistema debe ser capaz de realizar. Para **MyCarCheck**, he identificado los siguientes:

- Registrar nuevos repostajes, incluyendo litros, precio por litro, fecha y kilómetros actuales.
- Calcular automáticamente el precio total del repostaje y el consumo medio.
- Consultar el historial de repostajes realizados.
- Registrar vehículos y asociar cada repostaje a un vehículo concreto.
- Mostrar una lista de repostajes ordenada por fecha.
- Editar o eliminar registros de repostaje.
- Editar o eliminar vehículos
- Almacenar los datos localmente mediante una base de datos SQLite.
- Validar los campos obligatorios antes de guardar un registro.
- Mostrar mensajes de confirmación o error tras cada acción del usuario.
- Permitir la navegación entre pantallas mediante botones o menú.

## **Requisitos no Funcionales:**

Los requisitos no funcionales definen las características de calidad del sistema.

Para este proyecto se han considerado:

- Interfaz clara, intuitiva y adaptada a dispositivos móviles Android.
- Respuesta rápida del sistema y sin bloqueos perceptibles para el usuario.
- Desarrollada en Kotlin, siguiendo buenas prácticas de programación.
- Control de versiones mediante Git y alojado en un repositorio en GitHub.
- Estructura del proyecto orientada a la escalabilidad y el mantenimiento futuro.
- Documentación técnica actualizada y reflejando la estructura del código y la base de datos.
- Compatibilidad con versiones recientes de Android (API 33 o superior).
- Instalación mediante APK o desde el entorno de desarrollo.
- Preparación para futuras mejoras como sincronización en la nube o exportación de datos.

## **Diagrama de clases – Explicación técnica:**

El diagrama de clases representa la estructura lógica del sistema **MyCarCheck**, desarrollado en Android con Kotlin. Se han modelado las entidades principales, las actividades visuales, los adaptadores y la clase de persistencia, reflejando cómo se relacionan los componentes y qué función cumplen dentro del sistema.

### **Modelos de datos:**

- **Usuario:** representa al usuario registrado en el sistema. Cada usuario, puede vincular vehículos a un perfil.
- **Vehículo:** contiene atributos como matricula, marca, modelo, tipoCombustible. Cada usuario puede tener varios vehículos.
- **Repostaje:** registra datos como litros, precio, fecha, kilometraje. Cada vehículo puede tener múltiples repostajes asociados.

Relación: Usuario → Vehículo → Repostaje (1:N en ambos casos)

### **Persistencia:**

- **BaseDeDatos:** clase central que gestiona la persistencia local mediante SQLite. Incluye métodos CRUD para **Usuario**, **Vehículo** y **Repostaje**, así como funciones específicas como `editarVehiculo`, `editarRepostaje`, `getVehiculoPorId` y `getRepostajePorId` para facilitar la edición y recuperación individual de registros.
- Todas las actividades acceden a esta clase para guardar o recuperar datos.

### **Actividades visuales:**

Cada Activity representa una pantalla funcional del sistema:

- **MainActivity:** punto de entrada de la app.
- **LoginActivity / RegistroActivity:** gestión de acceso básico.

- **MenuPrincipalActivity**: panel de navegación principal.
- **AgregarVehiculoActivity**: formulario para registrar vehículos.
- **ListaVehiculosActivity**: muestra los vehículos registrados.
- **AgregarRepostajeActivity**: permite registrar un repostaje.
- **ListaRepostajesActivity**: muestra los repostajes de un vehículo.
- **EstadisticasActivity**: calcula y muestra estadísticas de consumo.
- **HistorialRepostajesActivity**: vista alternativa para consultar repostajes.
- **EditarVehiculoActivity**: permite modificar los datos de un vehículo ya registrado.
- **EditarRepostajeActivity**: permite modificar los datos de un repostaje existente.

Todas estas actividades interactúan con **BaseDeDatos** y con los modelos mediante Intent y validación previa.

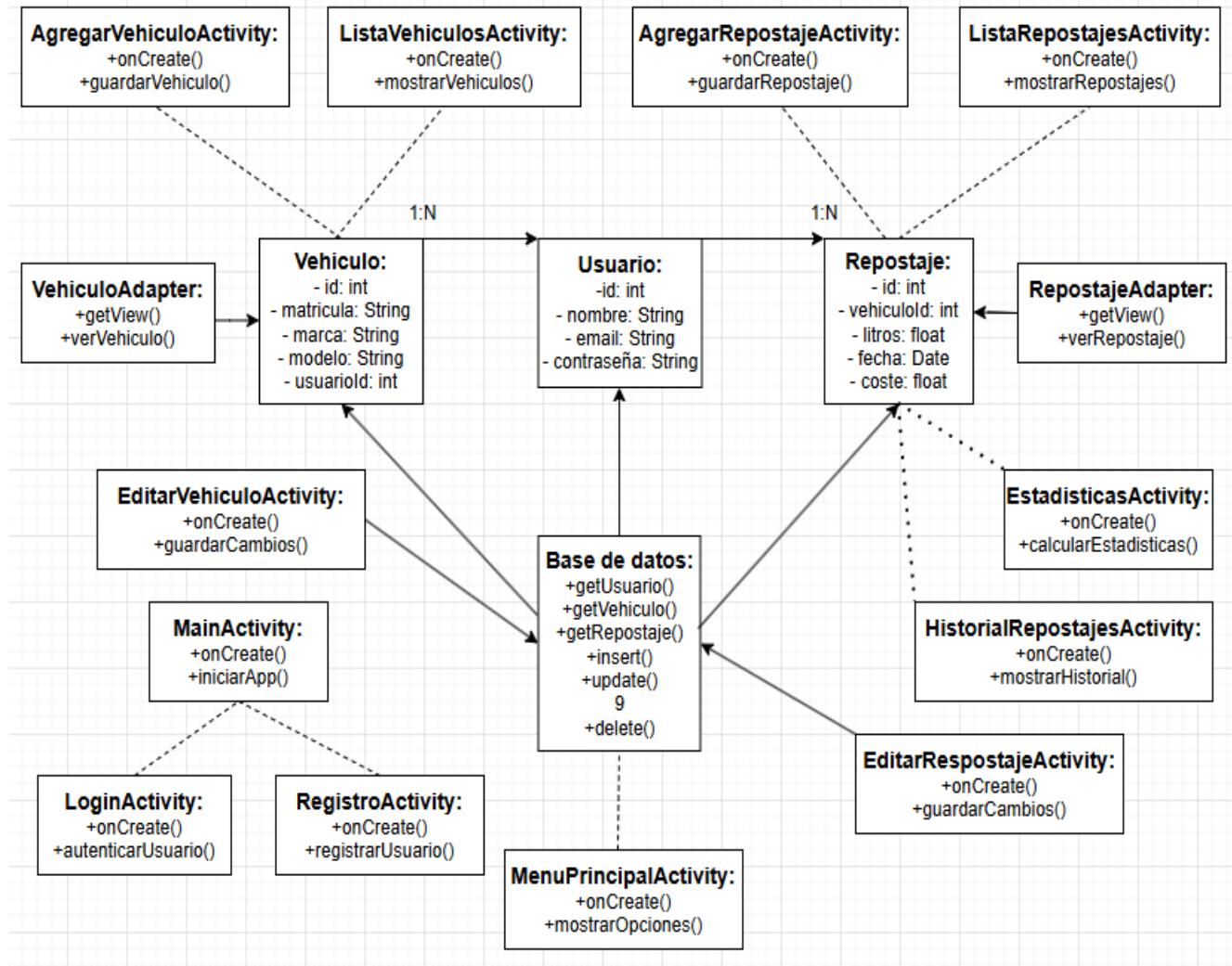
#### **Adaptadores:**

- **VehiculoAdapter** y **RepostajeAdapter**: gestionan la visualización de listas en RecyclerView, conectando los modelos con sus respectivos layouts (*item\_vehiculo.xml*, *item\_repostaje.xml*).

#### **Relaciones clave:**

- Las actividades acceden a **BaseDeDatos** para realizar operaciones sobre los modelos.
- Los adaptadores reciben listas de objetos (**Vehículo**, **Repostaje**) y los renderizan en pantalla.
- Las relaciones entre entidades están claramente definidas para garantizar trazabilidad y validación.

## Diagrama UML:



## Diagrama de Gantt:

El diagrama de Gantt del proyecto **MyCarCheck** refleja la planificación temporal del desarrollo, distribuyendo las tareas clave desde el análisis inicial hasta la entrega final. Cada fase se representa con su duración estimada y dependencias, permitiendo visualizar el progreso y controlar los tiempos de implementación, pruebas y documentación. Esta herramienta facilita la organización del trabajo y garantiza una ejecución ordenada del TFG.

Tarea								
Análisis de requisitos	Analisis de requisitos							
Diseño de base de datos		Diseño de base de datos						
Diseño de interfaz (layouts)			Diseño de interfaz (layouts)					
Implementación de modelos				Implementación de modelos				
Implementación de actividades					Implementación de actividades			
Persistencia con SQLite		Persistencia con SQLite						
Adaptadores y listas						Adaptadores y listas		
Estadísticas y lógica extra							Estadísticas y lógica extra	
Pruebas y validaciones								Pruebas y validaciones
Redacción de la memoria				Redacción de la memoria				
Revisión y entrega final							Revisión y entrega final	
	07-Octubre	14-Octubre	21-Octubre	28-Octubre	04-noviembre	11-noviembre	18-noviembre	25-noviembre

Tarea	Fecha	Total de horas
-------	-------	----------------

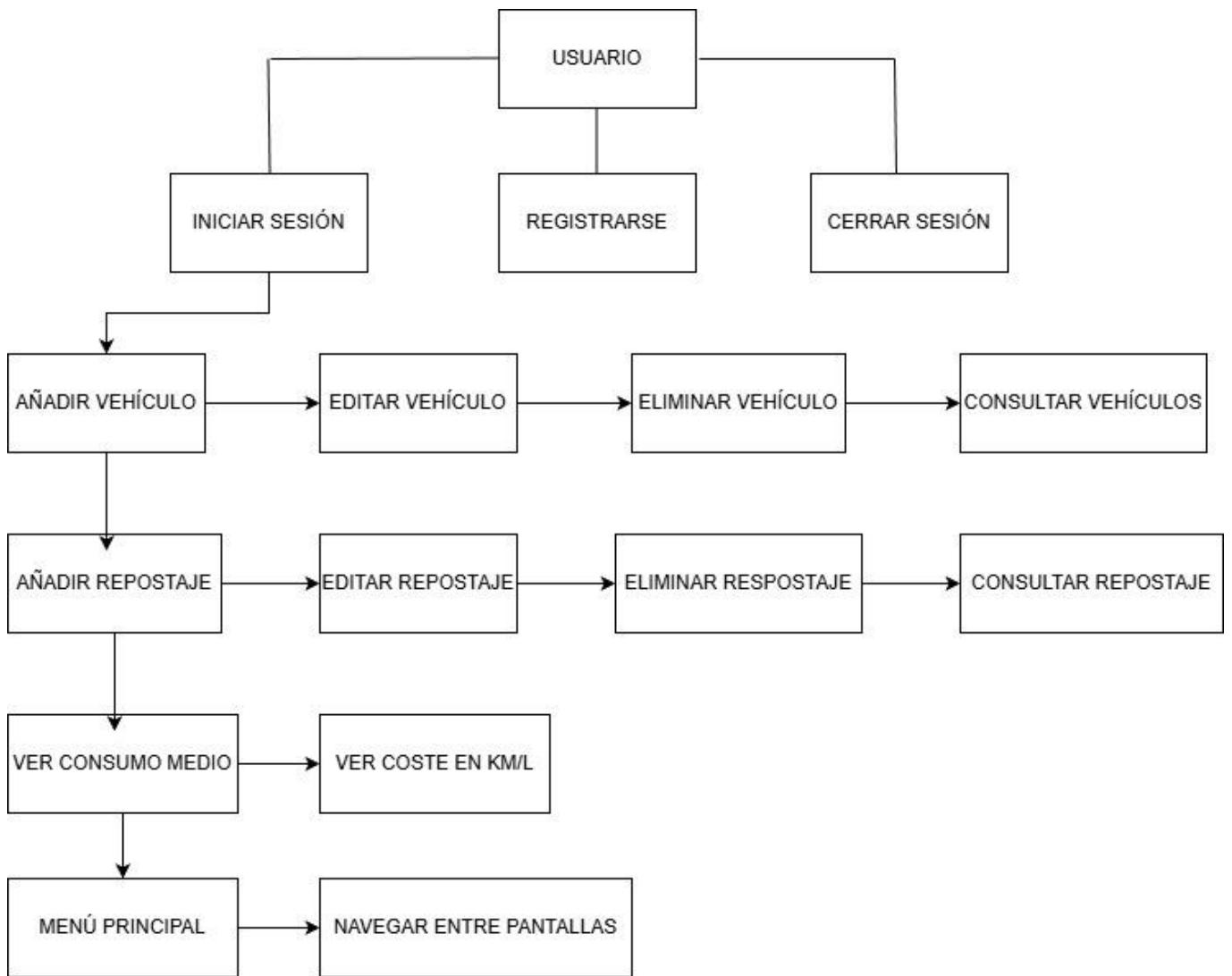
Diseño de interfaz (layouts)	14-21 octubre	20 h
Implementación de modelos	21-28 octubre	35 h
Implementación de actividades	21 octubre - 4 noviembre	55 h
Persistencia con SQLite	14-21 octubre	15 h
Adaptaciones y listas	4-11 noviembre	20 h
Estadísticas y lógica extra	11-18 noviembre	30 h
Pruebas y valoraciones	18-25 noviembre	25 h
Redacción de memoria	7 octubre - 18 noviembre	60 h
Revisión y entrega final	11-25 noviembre	20 h

### Tabla de presupuesto estimado:

El presupuesto del proyecto **MyCarCheck** se ha elaborado como una estimación teórica, considerando el coste de desarrollo, documentación, y otros recursos implicados. Se han utilizado herramientas gratuitas, por lo que no se han requerido licencias de pago. El coste total estimado asciende a 4.050€, reflejando el valor profesional del trabajo realizado.

Concepto	Detalle	Coste estimado
<b>Horas de desarrollo</b>	<b>220 h × 15 €/h</b>	<b>3.300€</b>
<b><i>Elaboración de manual de usuario</i></b>	<b>60 h × 12 €/h</b>	<b>720€</b>
<b>Licencias de software</b>	<b>Android Studio, SQLite (gratuito)</b>	<b>0€</b>
<b>Transporte (tutorías)</b>	<b>Desplazamientos puntuales</b>	<b>30€</b>
<b>Total estimado</b>		<b>4.050 €</b>

## Diagrama de casos de uso:



El diagrama de casos de uso representa las interacciones entre el usuario y las funcionalidades principales de la aplicación **MyCarCheck**. Este modelo permite visualizar de forma clara y estructurada qué acciones puede realizar el usuario dentro del sistema.

El único actor del sistema es el **Usuario registrado**, que accede a la aplicación mediante un login o registro en caso de no estar registrado, para poder realizar operaciones relacionadas con la gestión de vehículos del usuario, añadiendo o

editando estos vehículos, así como registrando repostajes o editarlos, y consultar estadísticas sobre estos repostajes.

Entre los casos de uso más relevantes se encuentran:

- **Autenticación:** permite al usuario iniciar sesión, registrarse y cerrar sesión, garantizando un acceso seguro con su usuario y contraseña registradas previamente.
- **Gestión de vehículos:** incluye funcionalidades para añadir, editar, eliminar y consultar vehículos registrados, vinculados al perfil del usuario que esté registrado.
- **Gestión de repostajes:** permite registrar nuevos repostajes, modificarlos o eliminarlos, así como consultar el historial completo asociado a cada vehículo del usuario.
- **Estadísticas:** ofrece métricas como el consumo medio y el coste por kilómetro, calculadas automáticamente a partir de los datos introducidos en los repostajes.
- **Navegación:** facilita el acceso al menú principal y la transición entre pantallas mediante botones y estructuras visuales intuitivas.

Cada caso de uso está conectado directamente con el actor principal mediante relaciones simples. Este enfoque garantiza que todas las funcionalidades estén disponibles de forma clara, accesible y sencilla.

### **Diagrama relacional de la base de datos:**

La base de datos de **MyCarCheck** está pensada para guardar toda la información que el usuario introduce en la app: sus vehículos, los repostajes que hace y sus datos de acceso. Está hecha con SQLite, que permite guardar los datos directamente en el móvil sin necesidad de conexión. Todas las relaciones son 1:N.

Tiene tres tablas principales:

- **Usuario**: guarda el nombre, email y contraseña del usuario.
- **Vehículo**: guarda los datos del coche (matrícula, marca, modelo, tipo de combustible) y está vinculado al usuario.
- **Repostaje**: guarda cada vez que se echa gasolina (litros, precio, fecha, kilómetros) y está vinculado al vehículo.
- Un **usuario** puede tener varios **vehículos**.
- Cada **vehículo** puede tener varios **repostajes**.

La relación es de uno a muchos en ambos casos:

**Usuario → Vehículo → Repostaje**

*Estructuras de las tablas:*

**Tabla usuario:**

CAMPO	TIPO	CLAVE	DESCRIPCIÓN
<b>idUsuario</b>	<b>INT</b>	<b>PK</b>	<b>Identificador único</b>
<b>Nombre</b>	<b>TEXT</b>		<b>Nombre de usuario</b>
<b>Contraseña</b>	<b>TEXT</b>		<b>Contraseña cifrada</b>

**Tabla Repostaje:**

CAMPO	TIPO	CLAVE	DESCRIPCIÓN
<b>idRepostaje</b>	<b>INT</b>	<b>PK</b>	<b>Identificador único</b>
<b>idVehículo</b>	<b>INT</b>	<b>FK</b>	<b>Referencia al vehículo</b>
<b>Fecha</b>	<b>TEXT</b>		<b>Fecha del repostaje</b>
<b>kmAnterior</b>	<b>INT</b>		<b>Km antes del repostaje</b>
<b>kmActuales</b>	<b>INT</b>		<b>Km en el momento del repostaje</b>
<b>Litros</b>	<b>INT</b>		<b>Litros repostados</b>
<b>precioLitro</b>	<b>INT</b>		<b>Precio del litro de combustible</b>

**Tabla Vehículo:**

CAMPO	TIPO	CLAVE	DESCRIPCIÓN
<b>idVehiculo</b>	INT	PK	Identificador único
<b>idUsuario</b>	INT	FK	Referencia al usuario
<b>Matrícula</b>	TEXT		<b>Matrícula del vehículo</b>
<b>Marca</b>	TEXT		<b>Marca del vehículo</b>
<b>Modelo</b>	TEXT		<b>Modelo del vehículo</b>
<b>Motor</b>	INT		<b>Potencia del vehículo</b>
<b>TipoCombustible</b>	ENUM		<b>Tipo de combustible del vehículo</b>
<b>anoMatriculacion</b>	INT		<b>Año en el que se matriculó el vehículo</b>
<b>kmActuales</b>	INT		<b>Kilómetros que tiene en este momento</b>

## Manual de usuario:

### *Bienvenida*

Esta aplicación está pensada para ayudarte a llevar un control sencillo y ordenado de tus vehículos y repostajes. La idea es que no tengas que apuntar nada en papeles ni hojas sueltas: todo queda guardado en tu móvil, accesible en cualquier momento.

### *Primeros pasos*

Cuando abres la aplicación por primera vez, verás la pantalla de **login**.

- Si ya tienes cuenta, introduce tu nombre de usuario y contraseña.
- Si es tu primera vez, pulsa en “**Registrar**” y crea tu usuario en unos segundos.

Una vez dentro, llegarás al **menú principal**, el punto de partida para todas las funciones.

### *Menú principal*

Aquí encontrarás varias opciones claras y directas:

- **Agregar vehículo** → para registrar un coche nuevo.
- **Ver vehículos** → para consultar la lista de coches que ya tienes guardados.
- **Estadísticas** → para ver cómo evoluciona tu consumo y tus gastos.
- **Cerrar sesión** → para salir de tu cuenta.
- **Eliminar cuenta** → si quieres borrar todo tu perfil y datos.

### *Añadir y gestionar vehículos*

- Pulsa **Agregar vehículo**.
- Introduce matrícula, marca, modelo y año de matriculación.

- Guarda los datos y listo: tu coche aparecerá en la lista.

En la lista de vehículos verás:

- La información básica del coche.
- Los kilómetros actuales.
- Botones para **ver repostajes** o **eliminar vehículo**.

### *Registrar repostajes*

Cada vez que repostes, puedes añadirlo a la app:

1. Entra en tu vehículo y pulsa **Ver repostajes**.
2. Pulsa **Agregar repostaje**.
3. Rellena los campos:
  - Litros repostados.
  - Precio por litro.
  - Kilómetros actuales.
  - Fecha (formato dd/mm/yyyy).
4. Pulsa **Guardar**.

La aplicación comprobará que los datos son correctos (por ejemplo, que los kilómetros actuales sean mayores que los anteriores) y guardará el repostaje en tu historial.

También puedes **editar** o **eliminar** repostajes desde la lista, usando los botones correspondientes.

### *Consultar estadísticas*

En el menú principal, pulsa **Estadísticas**. Aquí verás un resumen de tu consumo medio, el coste total y cómo han ido evolucionando tus repostajes. Es una forma rápida de saber cuánto gastas y cómo se comporta tu coche.

### *Cerrar sesión o eliminar cuenta*

- **Cerrar sesión:** sales de tu perfil, pero tus datos siguen guardados.
- **Eliminar cuenta:** borra tu usuario y todos los vehículos y repostajes asociados. Ten en cuenta que esta acción no se puede deshacer.

### *Consejos prácticos*

- Introduce siempre los kilómetros actuales de forma correcta para que las estadísticas sean fiables.
- Usa el formato de fecha **dd/mm/yyyy**.
- Si te equivocas al introducir un dato, puedes editar lo fácilmente.
- Mantén tu cuenta activa para que no pierdas tu historial.

### *En resumen*

**MyCarCheck** está pensada para ser sencilla:

- Registros tu usuario.
- Añades tus coches.
- Guardas cada repostaje.
- Consultas estadísticas cuando lo necesites.

## Manual de uso:

### *Navegación entre pantallas:*

#### Pantalla del login:

Nuestra primera pantalla, será la pantalla del login. En caso de no tener una cuenta registrada, nos pedirá que nos registremos. Si ingresamos unas credenciales erróneas, nos mostrará un mensaje en pantalla en el que nos dirá que nuestras credenciales son incorrectas.



Pantalla de registro:

La pantalla de registro no pide muchos campos, simplemente te pide un **nombre de usuario** y una **contraseña** para poder acceder.

## Registro de Usuario

Nombre de usuario  
admin

Contraseña  
\*\*\*\*

**Registrar**

[Volver al login](#)

Tenemos implantada la funcionalidad de comprobar si tenemos dos usuarios con el mismo nombre, para que no haya duplicados. En caso de tener duplicados, nos mostrará un aviso indicando que ya existe ese nombre de usuario.



El nombre de usuario ya existe

Una vez registrados y pulsemos el botón de registrar, nos redireccionará al menú de login nuevamente para iniciar la sesión.

## Menú principal:

Una vez accedamos con nuestra cuenta, nos encontraremos con el menú principal.



En este menú tendremos las opciones de:

- **Agregar un vehículo:** Opción que nos permitirá agregar vehículos para su posterior uso
- **Ver vehículos:** Opción en la que veremos cada vehículo registrado en el punto anterior. Dentro de esta opción, también podremos agregar los repostajes a cada vehículo registrado así como ver el historial de repostajes que hayamos añadido.
- **Estadísticas:** Opción en la que veremos las estadísticas asociadas a nuestros vehículos.
- **Cerrar sesión:** Opción que cerrará nuestra sesión actual y nos devolverá a la pantalla del login.
- **Eliminar cuenta:** Opción que nos permitirá eliminar nuestra cuenta, así como todos los vehículos y repostajes asociados a nuestra cuenta. Esta opción no tiene manera volver atrás.

## Pantalla de agregar vehículo:

En esta pantalla encontraremos los campos necesarios para agregar un vehículo a nuestra aplicación. Nos pedirá una **marca, modelo, matrícula, km actuales, año de matriculación, potencia y el tipo de combustible** que seleccionaremos en un desplegable. Una vez llenados todos estos campos, guardaremos nuestro vehículo y podremos empezar a realizar acciones sobre este.

### Agregar Vehículo

Marca  
Mercedes

Modelo  
Benz

Matrícula  
2024BCC

Kilómetros actuales  
120000

Año de matriculación  
2004

Potencia (CV)  
180

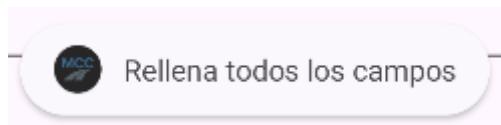
Tipo de combustible  
Gasolina ▾

**Guardar vehículo**

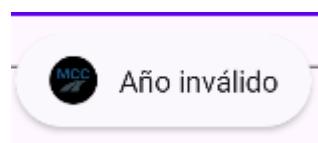
[Volver al inicio](#)

En esta pantalla tenemos varias comprobaciones a la hora de registrar el vehículo.

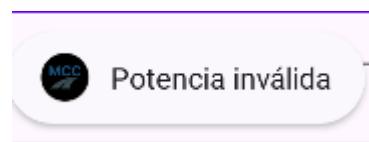
Si hay algún **campo vacío**, nos mostrará un aviso.



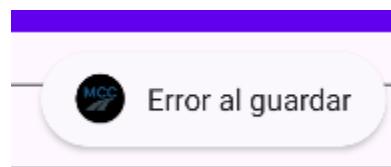
Si el año de matriculación es **inferior a 1950 y posterior al año actual**, mostrará un aviso.



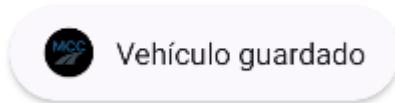
Si la **potencia es igual o menor a 0**, nos mostrará un aviso.



Si tenemos otro vehículo con **esa misma matrícula**, nos mostrará un aviso.



Una vez que todos nuestros campos sean correctos, pulsaremos el botón de crear, nuestro vehículo se habrá creado correctamente y ya podremos acceder a las acciones asociadas a él.



Pantalla para ver vehículos:

Una vez creado nuestro vehículo, podremos verlo desde esta pantalla.

Desde los botones podremos ver el **historial de repostajes**, editar los datos del vehículo creado o eliminarlo.



En la parte de abajo podremos registrar un repostaje o volver al inicio.



Si pulsamos en registrar un repostaje, nos pedirá que selecciones el vehículo sobre el que queremos registrarlo para pasar a la pantalla de registrar repostaje.

## Selecciona un vehículo

2024BCC - Mercedes

Si pulsamos sobre **ver repostajes**, nos aparecerá este menú:



En él, podremos **registrar un repostaje** o **volver al menú principal**.

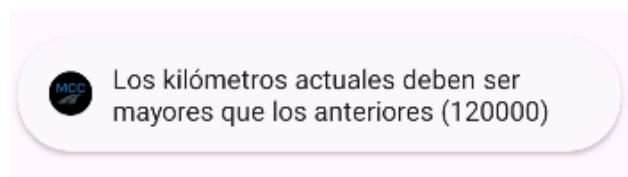
Si elegimos la opción de **registrar un repostaje**, nos encontraremos con este menú:

A screenshot of a form for registering a fueling. It contains four input fields with placeholder text and values: "Litros repostados" (50), "Precio por litro (€)" (1.50), "Kilómetros actuales" (120650), and "Fecha (dd/mm/yyyy)" (24/11/2025). Below the form, it says "Km anteriores: 120000". At the bottom are two buttons: a purple rounded rectangle labeled "Guardar Repostaje" and a white button with a black border labeled "Volver al inicio".

Aquí podremos **registrar un repostaje** para el vehículo seleccionado.

Nos pedirá los **litros** repostados, el **precio** del carburante por litro que hayamos repostado, los **kilómetros** que tenemos actualmente y la **fecha** en la que se ha realizado el repostaje.

A modo adicional, abajo nos mostrará información sobre los kilómetros que teníamos en el repostaje anterior, para que a la hora de repostar, si ingresamos un valor **inferior** a este, en este caso 120.000 km, nos muestre un aviso de que **no pueden ser inferiores a los anteriores**.



También tenemos otras comprobaciones a la hora de registrar el repostaje.

Si todos los campos **no están llenos**, nos mostrará un aviso.

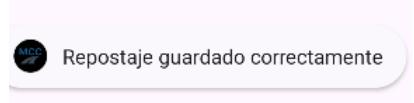


Si la fecha tiene un **formato distinto a dd/mm/yyyy**, nos mostrará un aviso.



Y como vimos antes, si los **kilómetros anteriores son distintos a los nuevos registrados**, nos mostrará un aviso.

Una vez rellenados estos datos, el repostaje se guardará correctamente y volveremos a la pantalla del historial de repostajes.



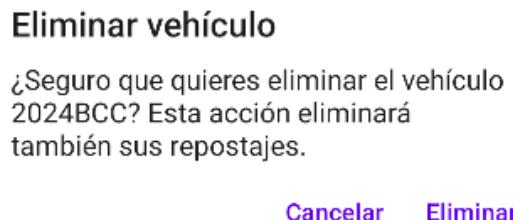
Una vez estemos en la pantalla de repostajes, podremos **ver los repostajes** que hemos realizado y registrado. Permitiéndonos la opción tanto de **editar** estos repostajes como de **eliminarlos**. A modo de resumen, también aparecerán estadísticas sobre estos repostajes, como será el **consumo medio de todos los repostajes** que hayamos realizado.



En la opción de editar vehículo, podremos editar los datos del vehículo actual al igual que en la opción de editar repostaje.

Si pulsamos sobre la opción de eliminar vehículo, nos mostrará un mensaje de confirmación, para no borrar en caso de haber pulsado en botón por error.

**Este cambio no tiene retorno.**



### Pantalla de estadísticas:

En esta pantalla podremos ver el **consumo global medio** de todos nuestros **vehículos**. Hará una **media** con **todos los repostajes de todos los vehículos** registrados y nos dará el **consumo global medio** de todos ellos juntos. También el **coste total** de estos repostajes, los **litros totales** repostados y los **kilómetros realizados** con nuestros vehículos.

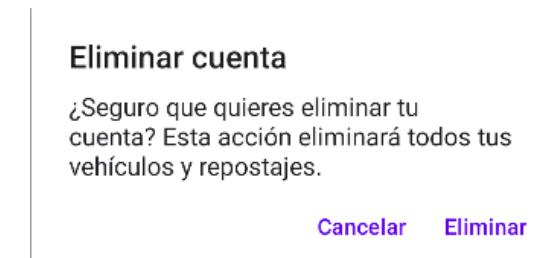


### Botón de cerrar sesión:

Este botón cerrará nuestra sesión actual y nos devolverá al menú de login para registrarnos nuevamente.

### Botón de eliminar cuenta:

Este botón nos permitirá eliminar nuestra cuenta así como todos los datos registrados en ella. Una vez pulsamos este botón no habrá opción de volver atrás.



## Conclusiones:

Este proyecto me ha servido para aprender de verdad cómo se construye una aplicación completa: desde pensar en los requisitos hasta verla funcionando en el móvil. He mejorado en Kotlin, en el manejo de bases de datos y en la organización del trabajo. También he aprendido a tener paciencia con los errores y a documentar mejor lo que hago.

Si lo volviera a empezar, intentaría simplificar algunas partes y usar una arquitectura más clara desde el principio, porque ahora sé lo importante que es para que el código sea fácil de mantener y ampliar. También emplearía más tiempo en pensar la estructura clara desde un principio para no tener que rehacer partes del código.

En resumen, ha sido un reto que me ha hecho crecer como desarrollador y me ha dado ganas de seguir aprendiendo en el mundo del desarrollo móvil.