Ecudau: Aplicación Android para analizar rutas en coche

Nombre Estudiante: David Pérez Conde
Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

Nombre Consultor/a: David Escuer Latorre
Profesor/a responsable de la asignatura: Carles Garrigues Olivella

Fecha de Entrega: 06/2018

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Ecudau: Aplicación Android para analizar rutas en coche*
Nombre del autor:	*David Pérez Conde*
Nombre del consultor/a:	*David Escuer Latorre*
Nombre del PRA:	*Carles Garrigues Olivella*
Fecha de entrega (mm/aaaa):	06/2018
Titulación:	Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles
Idioma del trabajo:	*Castellano*
Palabras clave	*Android, OBDII, GPS*

Resumen del Trabajo:

Hoy en día tenemos aplicaciones móviles que analizan el rendimiento de cualquier cosa, desde nuestra actividad física hasta la eficiencia energética de nuestro hogar.

Existe una carencia en cuanto al analizar el rendimiento de algo que hacemos en nuestro día a día: el uso de nuestro coche.

Analizando nuestra manera de conducir y la manera en la que se comporta nuestro vehículo podremos por un lado modificar la manera en la que se conduce para ahorrar combustible y por otro lado detectar alguna anomalía en el motor.

La finalidad de este proyecto es crear una aplicación Android nativa que supla las carencias descritas anteriormente que permita obtener y analizar rutas hechas con nuestro vehículo.

Para la obtención de datos objetivos utilizará por un lado el GPS del dispositivo y por otro un pequeño adaptador bluetooth conectado al vehículo.

Para la obtención de datos subjetivos se utilizará una interfaz sencilla donde incluso se podrán grabar notas de voz.

Una vez obtenidos los datos podremos ver los resultados obtenidos mediante gráficas y otros medios de explotación de datos.

Abstract:

Nowadays we have mobile applications that analyze the performance of everything, from our physical activity to the energy efficiency of our home.

There is a lack in terms of analyzing the performance of something we do in our day to day: our car.

Analyzing our way of driving and the way in which our vehicle behaves we can, on the one hand, modify the way we drive to save fuel and, on the other hand, detect any anomaly in the engine.

The purpose of this project is to create a native Android application that meets the needs described above to obtain and analyze routes made with our vehicle.

In order to obtain objective data, the device's GPS will be used on the one hand and a small bluetooth adapter connected to the vehicle on the other.
To obtain subjective data, a simple interface will be used where you can even record voice notes.

Once the data is obtained we can see the results obtained through graphs and other means of data exploitation.

Índice

1. INTRODUCCIÓN 2
1.1 CONTEXTO Y JUSTIFICACIÓN DEL TRABAJO 2
1.2 OBJETIVOS DEL TRABAJO 4
1.3 ENFOQUE Y MÉTODO SEGUIDO 5
1.4 PLANIFICACIÓN DEL TRABAJO 7
1.5 BREVE SUMARIO DE PRODUCTOS OBTENIDOS 13
1.6 BREVE DESCRIPCIÓN DE LOS OTROS CAPÍTULOS DE LA MEMORIA 13
2. DISEÑO CENTRADO EN EL USUARIO 15
2.1 USUARIOS Y ESCENARIOS DE USO 15
3. DISEÑO CONCEPTUAL 21
4. PROTOTIPADO 22
4.1 ÁRBOL DE NAVEGACIÓN 26
5. EVALUACIÓN 27
5.1 DEFINICIÓN DE LOS CASOS DE USO 27
5.2. DISEÑO DE LA ARQUITECTURA 34
5.3 BASE DE DATOS 35
6. IMPLEMENTACIÓN 37
6.1 INTRODUCCIÓN 37
6.2 HERRAMIENTAS UTILIZADAS 37
6.3 LIBRERÍAS Y APIS 39
6.4 IMPLEMENTACIÓN DE ECUDAU 40
6.5 PRUEBAS 48
7. CONCLUSIONES 51
8. BIBLIOGRAFÍA 52
9. ANEXOS 55

Índice de Ilustraciones

Ilustración 1 - Diagrama de Gantt....11
Ilustración 2 - Análisis de riesgos....12
Ilustración 3 - Splash....22
Ilustración 4 – Menú principal....22
Ilustración 5 – Lista de coches....23
Ilustración 6 – Crear coche....23
Ilustración 7 – Editar coche....23
Ilustración 8 – Pantalla principal (No conectado)....23
Ilustración 9 – Conectar dispositivo....24
Ilustración 10 – Pantalla principal (Conectado)....24
Ilustración 11 – Empezar ruta....24
Ilustración 12 – En ruta....24
Ilustración 13 – Crear incidencia....25
Ilustración 14 – Dictar incidencia....25
Ilustración 15 - Ruta pausada....25
Ilustración 16 - Lista de rutas....25
Ilustración 17 - Estadísticas de ruta....26
Ilustración 18 - Árbol de navegación....26
Ilustración 19 - Diagrama de casos de uso....27
Ilustración 20 - Arquitectura MVC....34
Ilustración 21 - Diagrama de clases....36
Ilustración 22 - Distribución de versiones Android....37
Ilustración 23 - OBDSim....38
Ilustración 24 - Estructura de la app....40
Ilustración 25 - Diagrama de clases definitivo....43
Ilustración 26 - Cambios de vista: menú superior....44
Ilustración 27 – Cambios de vista: listado de rutas....44
Ilustración 28 - Cambios de vista: combustible....45
Ilustración 29 - Alertas de audio desactivadas....47

Índice de tablas

Tabla 1 - Comparación con aplicaciones existentes 3
Tabla 2 - Planificación. Plan de trabajo 7
Tabla 3 - Planificación. Diseño y arquitectura 7
Tabla 4 - Planificación. Fase de desarrollo 8
Tabla 5 - Planificación. Entrega final 9
Tabla 6- Desglose de dedicación 10
Tabla 7 - Ficha del usuario 1 16
Tabla 8 - Ficha del usuario 2 17
Tabla 9 - Ficha del usuario 3 18
Tabla 10 - Caso de uso "Listar coches" 28
Tabla 11 - Caso de uso "Guardar coche" 28
Tabla 12 - Caso de uso "Borrar coche" 29
Tabla 13 - Caso de uso "Conectar GPS" 30
Tabla 14 - Caso de uso "Conectar OBD" 30
Tabla 15 - Caso de uso "Crear ruta" 31
Tabla 16 - Caso de uso "Crear incidencia" 32
Tabla 17 - Caso de uso "Abrir ruta" 32
Tabla 18 - Caso de uso "Borrar ruta" 33
Tabla 19 - Caso de uso "Listar rutas" 33
Tabla 20 - Pruebas: Conectar OBD 48
Tabla 21 - Pruebas: Grabar ruta 49
Tabla 22 - Pruebas: Listar rutas 49
Tabla 23 - Pruebas: Estadísticas de ruta 50
Tabla 24 - Pruebas: Listar vehículos 50

1. Introducción

1.1 Contexto y justificación del Trabajo

Hoy en día cualquier objeto cotidiano está conectado a internet, es lo que se llama IOT[1].
El tener objetos conectados nos proporciona estadísticas de uso, estadísticas de rendimiento, datos en tiempo real, accesibilidad de manera remota… etc.

Uno de los objetos con más desarrollo en materia de conectividad siguen siendo los vehículos. Hoy en día son capaces de detectar colisiones, de reportar accidentes de manera automática, identificar señales en la carretera e incluso la ansiada conducción autónoma.

El hecho de poder obtener datos objeticos de vehículos es algo que para los aficionados y no tan aficionados al automovilismo les sería de gran utilidad. Por una lado, para usuarios especializados les permitiría analizar el correcto funcionamiento de sus motores e incluso evitar averías futuras y por otro, para usuarios cotidianos, les podría proporcionar información relevante en cuanto a su manera de conducción y cómo hacerla más eficiente.

Aquellos que desean obtener datos de sus vehículos, lo realizan en talleres especializados pagando una costosa cuantía.

Una vez obtenida esta información, tampoco tienen manera de sacarle rendimiento, no pueden explotar esos datos, si no que únicamente disponen de los informes que son proporcionados por el taller especializado.

Aparte de aficionados al automovilismo y usuarios de a pie, también existen dueños de pequeños talleres que podrían dar a sus clientes un informe de un antes y después de una modificación o arreglo, dotando a sus servicios de un valor añadido.

1.1.2 Aplicaciones existentes con características similares

Aplicación	Ventajas	Inconvenientes
Torque PRO[2]	• Cumple con todas las funcionalidades • Visualización de datos en tiempo real • Borrar errores en la ECU[3]	• Muy poco intuitiva • Muy compleja • Interfaz muy desactualizada • Explotación de datos básica
OBD Car Doctor[4]	• Borrar errores en la ECU • Visualización de datos en tiempo real	• No obtiene datos en ruta
Obd Arny - OBD2 | ELM327[5]	• Borrar errores en la ECU • Visualización de datos en tiempo real	• No obtiene datos en ruta

Tabla 1 - Comparación con aplicaciones existentes

En general, existen una multitud de aplicaciones que realizan análisis de la ECU de un vehículo a través de un dispositivo OBD[6].
No obstante, todas ellas son bastante complejas de utilizar e incluso configurar.
La idea de Ecudau es obtener datos de manera muy intuitiva y con las mínimas acciones del usuario posibles.

Por otro lado, la mayoría de estas aplicaciones se basan en la monitorización en tiempo real y no en hacer una ruta y luego estudiar resultados.
Sólo la aplicación Torque PRO dispone de esta característica y la explotación de datos que tiene es bastante anticuada.

1.2 Objetivos del Trabajo

Objetivo general:

- Desarrollar una aplicación que obtenga datos objetivos y subjetivos de vehículos comerciales en ruta.

Objetivos específicos:

- Para este trabajo fin de master los objetivos que definirán el MVP[7] son los siguientes:
    - Conectar un dispositivo bluetooth ODB para obtener datos objetivos de la ECU.
    - Obtener datos de geolocalización mientras se está realizando una ruta
    - Poder registrar datos subjetivos tales como ruidos o vibraciones y añadir comentarios por voz.
    - Gestionar vehículos (crear, modificar, eliminar)
    - Gestionar rutas (crear, modificar, eliminar)
    - Ver un informe básico de los datos registrados de cada ruta
    - La aplicación se deberá poder funcionar 100% offline
    - La aplicación permitirá exportar los datos generados en una ruta en formato JSON.

1.3 Enfoque y método seguido

En la actualidad existen diversas aplicaciones de navegación que permiten registrar rutas.
También existen aplicaciones que se conectan a nuestro vehículo dándonos información en tiempo real de nuestro motor e incluso ver los errores que se almacenan en la ECU.

La aplicación que se plantea desarrollar reúne algunas de las características citadas aportando otras que marcarán la diferencia.
Por un lado, hacer una interfaz gráfica muy simple e intuitiva para que la obtención de datos de nuestros vehículos sea muy sencilla y por otro lado, la flexibilidad de poder explotar los datos como queramos.

Se opta por realizar una aplicación nativa debido a que el rendimiento a la hora de obtener y persistir datos debe ser máxima.

1.3.1 Diseño centrado en el usuario

Debido a los objetivos del proyecto queda claro que la usabilidad de la aplicación a crear es imprescindible a la hora de diseñar y desarrollar el producto. La idea es que el alcance de esta aplicación sea máximo aun siendo a priori una aplicación de carácter muy técnico. Por lo tanto será necesario definir cómo serán los diferentes tipo de clientes potenciales y sus necesidades y seguir las normas básicas en el diseño de interfaces para dispositivos móviles:

- Navegación simple para el usuario, haciendo que la app sea intuitiva. Las posibles acciones deben estar claras en todo momento.
- Una app eficiente permitirá al usuario realizar una acción en el menor número de pasos posible.
- Debe haber coherencia en la estructura de las distintas páginas de la app.
- Al realizar cualquier acción el usuario debe recibir feedback inmediato para saber que su acción ha sido reconocida y que va a recibir respuesta, o que ha habido algún problema.

1.3.2 Arquitectura y tecnologías usadas en la aplicación

Para la estructura de la aplicación se optara por el patrón MVC[8] (Model View Controller)

Para la conectividad bluetooth con un dispositivo ODB se aprovechará el negocio de la aplicación AndrOBD[9] publicada en GitHub:

Para el registro de incidencias por voz se utilizará la librería nativa Android SpeechRecognizer[10], dado que puede funcionar en modo offline.

En cuanto a la GUI general de la aplicación se tomarán aplicaciones de referencia como:

- Strava[11]: Aplicación para obtener y publicar rutas en bicicleta
- Waze: [12] Aplicación de navegación donde se pueden reportar incidencias en carretera.

1.4 Planificación del Trabajo

En este apartado vamos a comentar las distintas tareas que se han llevado a cabo para la realización del proyecto, así como un diagrama de Gantt (Ilustración 1) donde se puede apreciar tanto la duración temporal de cada una de ellas.

A continuación se desglosan las tareas a realizar:

Tarea 1 - Plan de trabajo	
Periodo	21 Febrero – 14 Marzo
Tareas	**Redacción de la memoria** ▪ Contexto y justificación del trabajo ▪ Objetivos del trabajo ▪ Enfoque y método elegido ▪ Planificación del trabajo ▪ Productos obtenidos

Tabla 2 - Planificación. Plan de trabajo

Tarea 2 – Diseño y arquitectura	
Tarea 2.1 – Definición del diseño	
Periodo	15 Marzo – 27 Marzo
Tareas	▪ Usuarios y contexto de uso ▪ Diseño conceptual ▪ Prototipos ▪ Evaluación
Tarea 2.2 – Definición de la arquitectura	
Periodo	22 Marzo – 04 Abril
Tareas	▪ Casos de uso ▪ Diseño de la arquitectura

Tabla 3 - Planificación. Diseño y arquitectura

Tarea 3 – Desarrollo + Test	
Tarea 3.1	
Periodo	05 Abril – 09 Abril
Tareas	▪ Modelo de negocio ▪ Base de datos local + ORM[13] ▪ Entidades ▪ Operaciones CRUD[14] para entidades
Tarea 3.2	
Periodo	10 Abril – 16 Abril
Tareas	▪ Menú ▪ Splash ▪ Gestión de entidades (CRUD)
Tarea 3.3	
Periodo	17 Abril – 10 Mayo
Tareas	▪ Gestión de la conectividad con OBD ▪ Gestión de la conectividad GPS ▪ Obtención de datos ▪ Creación de rutas ▪ Crear incidencias en rutas
Tarea 3.4	
Periodo	11 Mayo – 16 Mayo
Tareas	▪ Formulario de estadísticas por ruta
Tarea 3.5	
Periodo	14 Abril – 16 Mayo
Tareas	▪ Testeo de la aplicación

Tabla 4 - Planificación. Fase de desarrollo

Tarea 4 – Entrega final	
Tarea 4.1 – Memoria	
Periodo	17 Mayo – 25 Mayo
Tareas	▪ Documentación y memoria
Tarea 4.2 – Vídeo	
Periodo	26 Mayo – 06 Junio
Tareas	▪ Realización del vídeo de final de entrega

Tabla 5 - Planificación. Entrega final

A continuación se desglosa el reparto de la dedicación a cada fase del proyecto:

Desglose de dedicación	
Tarea 1 – Plan de trabajo	
Periodo	21 Febrero – 14 Marzo
Dedicación total	30 horas
Dedicación en día laborable	1 hora
Dedicación en día festivo	2,5 horas
Tarea 2 – Diseño	
Periodo	14 Marzo – 04 Abril
Dedicación total	42 horas
Dedicación en día laborable	1 hora
Dedicación en día festivo	4,5 horas
Tarea 3 – Plan de trabajo	
Periodo	21 Febrero – 14 Marzo
Dedicación total	30 horas
Dedicación en día laborable	1 hora
Dedicación en día festivo	2,5 horas

Tarea 4 – Desarrollo + Test	
Periodo	05 Abril – 16 Mayo
Dedicación total	186 horas
Dedicación en día laborable	3 horas
Dedicación en día festivo	8 horas
Tarea 5 – Entrega final	
Periodo	17 Mayo – 06 Junio
Dedicación total	42 horas
Dedicación en día laborable	1 hora
Dedicación en día festivo	4,5 horas
	Total: 300 horas

Tabla 6- Desglose de dedicación

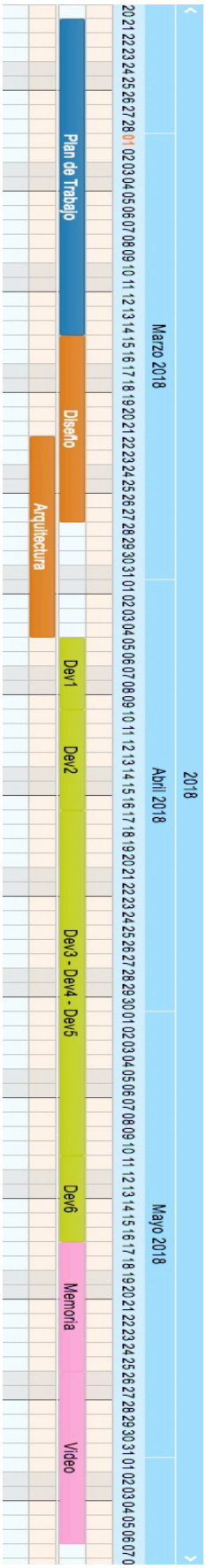

Ilustración 1 - Diagrama de Gantt

1.4.1 Análisis de Riesgos

Como ya se ha descrito anteriormente, uno de los objetivos más ambiciosos es conseguir conectar el dispositivo Android a la Unidad de Control del Motor (ECU) de un vehículo.

No obstante, dado que se trata de un concepto muy técnico es relevante realizar un análisis de los posibles riesgos que impedirían obtener el resultado deseado:

		IMPACTO		
		Bajo	Medio	Alto
PROBABILIDAD	Baja	Muy bajo	Bajo	Medio
	Media	Bajo	Medio	Alto
	Alta	Medio	Alto	Muy alto

Ilustración 2 - Análisis de riesgos

- El dispositivo bluetooth se estropea:
  - Probabilidad: **Media**
  - Impacto: **Medio**
  - Solución: Adquirir otro nuevo

- La ECU del vehículo de pruebas se estropea:
  - Probabilidad: **Baja**
  - Impacto: **Alto**
  - Solución: Utilizar otro vehículo de pruebas

- No se consigue conectar la aplicación con la ECU por problemas técnicos:
  - Probabilidad: **Media**
  - Impacto: **Alto**
  - Solución: Optar por un desarrollo alternativo

- No se consigue conectar la aplicación con la ECU por no disponer de suficiente presupuesto de horas:
  - Probabilidad: **Alta**
  - Impacto: **Alto**
  - Solución: Optar por un desarrollo alternativo

Para los casos donde el resultado del análisis sea Alto o Muy Alto se optará por no implementar finalmente esta funcionalidad.
No obstante, para complementar la carencia de esta funcionalidad se desarrollará la siguiente funcionalidad:

- Realizar copias de seguridad en la base de datos en tiempo real proporcionada por Google (Firebase[15])

1.5 Breve sumario de productos obtenidos

Al finalizar el proyecto se crearán los siguientes productos:

- Aplicación para dispositivo móvil de plataforma Android con las siguientes funcionalidades:
    - Conectar un dispositivo bluetooth ODB para obtener datos objetivos de la ECU.
    - Obtener datos de geolocalización mientras se está realizando una ruta
    - Poder registrar datos subjetivos tales como ruidos o vibraciones y añadir comentarios por voz.
    - Gestionar vehículos (crear, modificar, eliminar)
    - Gestionar rutas (crear, modificar, eliminar)
    - Ver un informe básico de los datos registrados de cada ruta
    - La aplicación se deberá poder funcionar 100% offline
    - La aplicación permitirá exportar los datos generados en una ruta en formato JSON.

- Código fuente de la aplicación.
- Manual de despliegue del proyecto que explicará cómo desplegar el proyecto y ponerlo en marcha partiendo del código fuente.
- Manual de usuario de la aplicación
- Memoria del proyecto.
- Vídeo de presentación del producto en el que se explican los objetivos del trabajo fin de máster, los aspectos más importantes del desarrollo, las decisiones tomadas a lo largo del proceso, el resultado y las conclusiones.

1.6 Breve descripción de los otros capítulos de la memoria

En los siguientes capítulos se describen los pasos seguidos hasta la obtención de la aplicación Ecudau.

En primer lugar se describen los pasos seguidos en el diseño de la aplicación. Partiendo del análisis de los usuarios potenciales se han definido los casos de uso de la aplicación y a continuación se ha realizado un prototipo de las pantallas más relevantes. Por otro lado se describen las características de la base de datos necesaria para el funcionamiento de nuestra app.

En el siguiente capítulo se describen los pasos seguidos en el desarrollo de la aplicación, explicando las dificultades encontradas y los cambios a los que han conllevado.
Por último se explican las conclusiones obtenidas a lo largo del trabajo, así como las líneas futuras y las propuestas de mejora.

2. Diseño Centrado en el Usuario

En este capítulo trataremos los puntos iniciales para cualquier desarrollo de una aplicación. Éstos se dividen en una investigación y toma de requisitos de usuarios potenciales, una elaboración de casos de uso y flujos de interacción, y por último un diseño de prototipo de cómo será la aplicación.

Es importante analizar muy bien a los posibles usuarios, ya que a partir de este análisis podremos detectar las funcionalidades que deberá tener la aplicación y evitar otras que no sean de gran importancia.

2.1 Usuarios y escenarios de uso

2.1.1 Usuarios

Como principales usuarios de la aplicación Ecudau podemos encontrar desde aficionados al motor que simplemente desean ver los valores que devuelve el motor de su coche como hobby, como mecánicos o profesionales del motor que quiere obtener datos del coche para realizar evaluaciones o mejoras.

A través de las fichas que se presentan a continuación, conoceremos más detalladamente las características, perfiles, y objetivos de los usuarios.

Ficha 1	
Nombre	Adrián Fernández
Edad	35 años
Profesión	Dueño de un taller mecánico
Nivel de conocimientos tecnológicos	• Redes sociales: **Alto** • Smartphone y Tablet: **Alto** • IT e Internet: **Medio**
Descripción	Adrián es un enamorado de la mecánica del automóvil. Empezó muy joven a interesarse por el automovilismo y ahora tiene su propio taller. Trabaja alrededor de 11 horas cada día, entre semana dedica el taller a sus clientes y los fines de semana lo usa para su disfrute, solo o con amigos.
Necesidades y objetivos	A menudo le gustaría proporcionar a sus clientes un valor añadido en sus servicios de mecánica. Un informe antes y después de una mejora o al arreglar una avería. El problema es que las herramientas que hay actualmente en el mercado se van de su presupuesto y además la mayoría de las máquinas de diagnóstico son demasiado grandes como para introducirlas dentro del vehículo y hacer rutas registrando datos.
Dispositivos y plataformas	• Smartphone: Android • Tablet: Android • Portátil: - • PC: Windows
Nivel de conocimientos de mecánica del automóvil	• **Alto**

Tabla 7 - Ficha del usuario 1

Ficha 2	
Nombre	Cristian Gómez
Edad	26 años
Profesión	Estudiante
Nivel de conocimientos tecnológicos	• Redes sociales: **Alto** • Smartphone y Tablet*:* **Alto** • IT e Internet: **Alto**
Descripción	Cristian estudia ingeniería química y se encuentra en su último año de carrera. Se compró un coche con ayuda de sus padres y se ha convertido en su mayor afición. Le gusta quedar con sus amigos para hacer rutas en coche o simplemente hablar sobre mejoras en sus coches, nuevos modelos y automoción en general.
Necesidades y objetivos	Cuando Cristian queda con sus amigos a hacer rutas les gustaría poder comparar el rendimiento que tienen sus vehículos, ver quién ha dado lo mejor de su motor en un sector o ver la diferencia de rendimiento de sus motores después de realizar alguna modificación, como por ejemplo una reprogramación o instalar un nuevo filtro de aire.
Dispositivos y plataformas	• Smartphone: Android • Tablet: iOS • Portátil: Windows • PC: -
Nivel de conocimientos de mecánica del automóvil	• **Medio**

Tabla 8 - Ficha del usuario 2

Ficha 3	
Nombre	Martín Dueñas
Edad	38 años
Profesión	Profesor particular
Nivel de conocimientos tecnológicos	• Redes sociales: **Medio** • Smartphone y Tablet: **Medio** • IT e Internet: **Bajo**
Descripción	Martín está casado y tiene dos hijos pequeños de 3 y 5 años. Actualmente se dedica a dar clases particulares a domicilio de física, química y matemáticas a estudiantes desde primaria hasta bachillerato. Martín vive en una ciudad donde el transporte público no es muy eficiente y se traslada en su vehículo particular para asistir a las clases de sus alumnos.
Necesidades y objetivos	Martín cree que a pesar de que tiene un vehículo relativamente nuevo y eficiente, cree que podría cambiar su manera de conducir para hacer que los trayectos para asistir a sus clases particulares sean más económicos y así ahorrar dinero a final de mes. El problema que tiene es que después de muchos años conduciendo ha cogido muchas manías que tiene que corregir y necesita alguna manera de analizar su manera de conducir para ver qué puntos son críticos para ser corregidos.
Dispositivos y plataformas	• Smartphone: iOS • Tablet: - • Portátil: OSX • PC: -
Nivel de conocimientos de mecánica del automóvil	• **Bajo**

Tabla 9 - Ficha del usuario 3

2.1.2 Escenarios de uso

Se han definido los siguientes escenarios de uso para los usuarios potenciales que acabamos de definir.

Adrián

Adrián acaba de recibir un coche que tiene la sonda lambda estropeada. El coche funciona correctamente lo único que la mezcla de gasolina y aire es incorrecta.
Adrián conecta en adaptador OBD al vehículo estropeado y realiza una ruta corta para registrar los valores de la mezcla incorrecta del combustible y el consumo abusivo a consecuencia de este error.
Una vez solventado el problema vuelve a realizar la misma ruta registrando los valores correctos tanto de la mezcla de combustible como del consumo medio del vehículo.
Exporta las gráficas de los datos a PDF desde la aplicación y se lo enseña a su cliente a la entrega del vehículo explicando los efectos beneficiosos de la solución al problema.

Cristian

Cristian ha quedado con sus amigos en el parking del centro comercial Las Gavarras de Tortosa. Todos ellos vienen con sus respectivos coches.
Uno de sus amigos ha oído hablar de una aplicación que sirve para analizar rutas en coche y deciden descargarla.
Como buenos aficionados al automóvil, todos ellos ya cuentan con un receptor bluetooth para obtener datos de la ECU, lo conectan y salen en ruta.
Una vez han vuelto al centro comercial, cenando analizan los resultados y comparan los valores obtenidos de potencia, torque, etc…

Martín

Martín ha oído hablar de una aplicación que permite registrar datos del vehículo para luego analizarlos después.
Decide bajarse la aplicación y se da cuenta de que le hace falta un pequeño dispositivo para emitir la información a la aplicación.
Compra directamente el que le sugiere la aplicación y tras 48h lo tiene en su casa listo para ser utilizado.
Busca en YouTube dónde se encuentra el su coche el puerto para conectar el dispositivo OBD y lo conecta sin mayor problema.
Al asistir a una de sus clases decide registrar la ruta que realiza hasta llegar al domicilio de uno de sus alumnos.

Una vez en su casa, Martín vuelve a abrir la aplicación para ver las estadísticas relativas al consumo de la ruta que ha realizado ese mismo día y

se percata de que hay puntos donde está pisando a fondo el acelerador, la velocidad no aumenta como debería y el consumo se dispara.
Se acaba de dar cuenta de que el porcentaje de pisado del pedal del acelerador influye directamente en la inyección del vehículo y por tanto en su consumo.

A la semana siguiente Martín vuelve a analizar la ruta después de cambiar su manera de conducción y se percata de que ha conseguido bajar 2L el consumo medio en su ruta habitual.

3. Diseño conceptual

En este capítulo se detallan los diferentes escenarios de uso en los que el usuario podrá utilizar las funcionalidades de la aplicación y como lo hará.

Escenario de uso – El usuario quiere crear una nueva ruta para mejorar su eficiencia al conducir

Descripción*:*

> En este escenario nos encontramos con un usuario que cree que el consumo de su vehículo es excesivo respecto a lo que las especificaciones del vehículo detallan.
> El usuario decide hacer uso de la aplicación y esa misma semana realiza el trayecto más habitual.
> Después de registrar datos de conducción en una ruta de unos 20 minutos aproximadamente llega a su casa y se dispone a analizar con detenimiento los datos registrados.
> Entra en los detalles de la ruta y se percata que cuando el porcentaje de pisado del acelerador es superior al 50% el consumo se dispara, independientemente de la marcha en la que se encuentre.
> Al día siguiente vuelve a realizar la ruta aplicando las conclusiones que sacó al analizar la ruta y tras volver a analizar esta nueva ruta descubre que el consumo ha bajado unos 2 litros/100km respecto a la media que solía tener para la misma ruta.

Escenario de uso – El usuario quiere crear una ruta porque ha detectado ciertas anomalías al conducir su vehículo

Descripción*:*

> En este escenario de uso, el usuario lleva notando hace unas semanas que cuando el coche llega a las 2100 rpm el coche da tirones y teme que sea el inicio de un fallo más grande.
> Decide abrir la aplicación y hacer una ruta donde cada vez que el coche llega a las 2100 rpm y da los tirones el usuario registra una anomalía en la ruta.
> Cuando da por concluida la ruta, analiza los datos y observa que los datos correspondientes al voltaje de la ECU distan mucho de la media cuando llegan a 2050 rpm.
>
> Al observar esta anomalía, el usuario acude a su mecánico de confianza donde le explica el problema y le enseña los datos registrados en la aplicación.

4. Prototipado

El prototipo de una aplicación nos permite mostrar con el máximo de detalle qué apariencia tendrá la aplicación.
Esto permite al cliente/usuario evaluar el prototipo y proponer cambios o aceptar el diseño para comenzar a desarrollar la aplicación.
A continuación se muestra el prototipo de la aplicación Ecudau realizado con la aplicación Balsamiq[16]:

Ilustración 3 - Splash

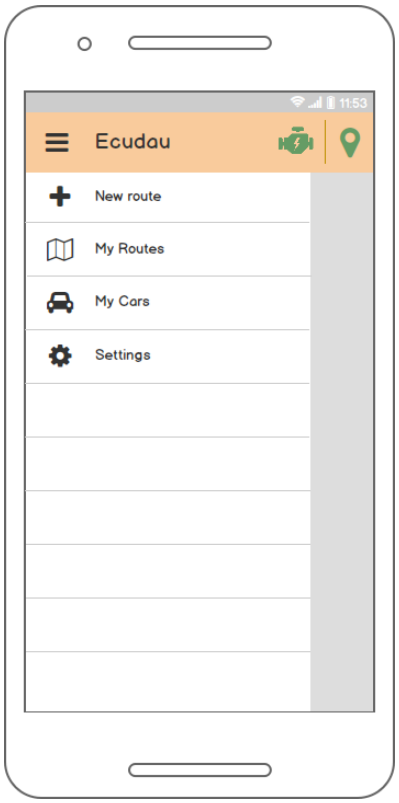

Ilustración 4 – Menú principal

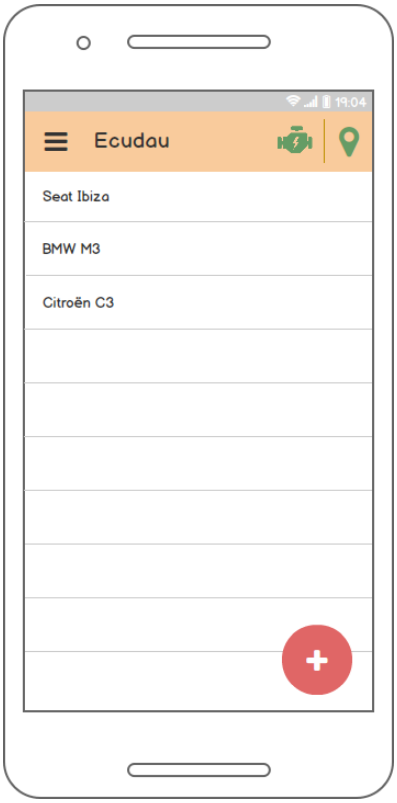

Ilustración 5 – Lista de coches

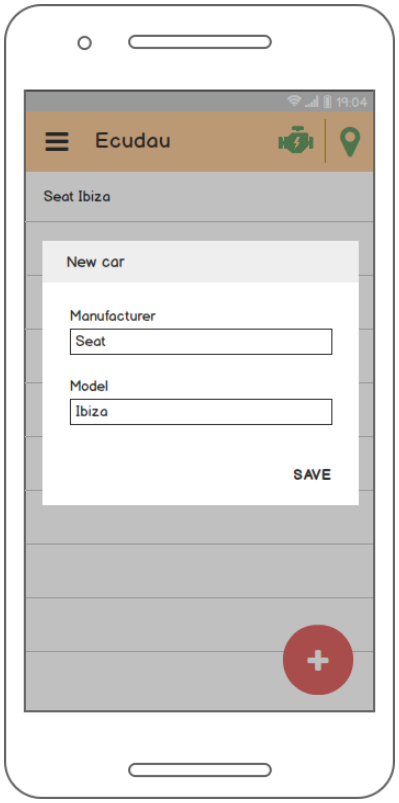

Ilustración 6 – Crear coche

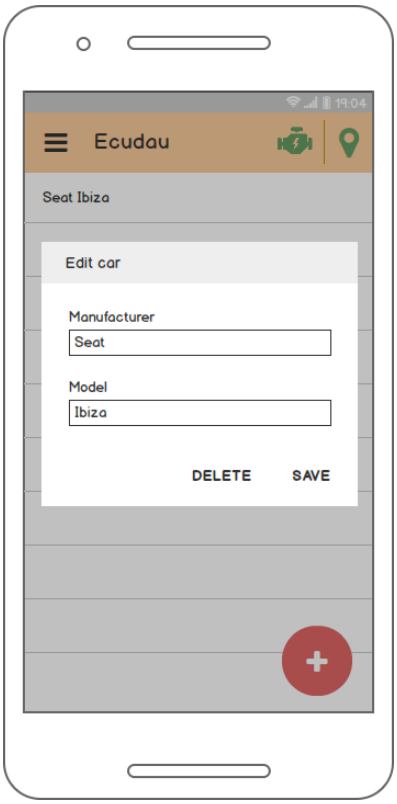

Ilustración 7 – Editar coche

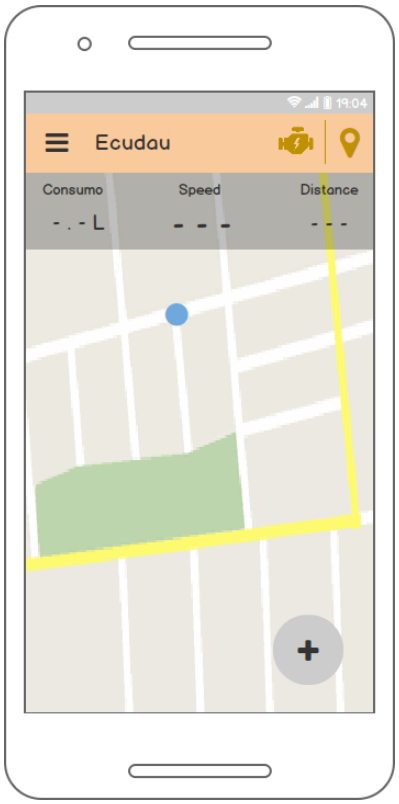

Ilustración 8 – Pantalla principal (No conectado)

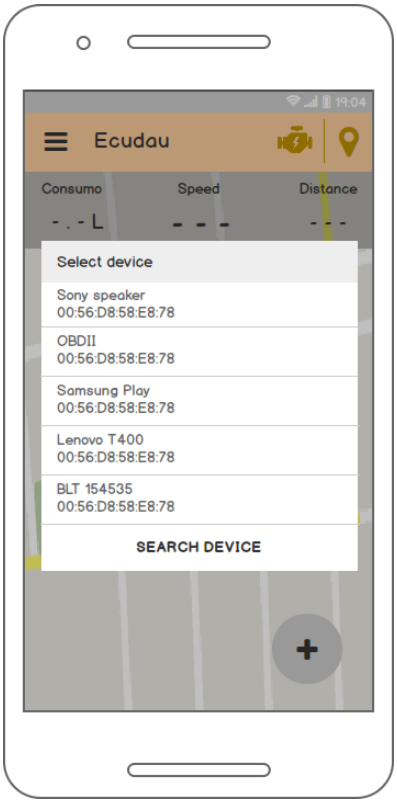

Ilustración 9 – Conectar dispositivo

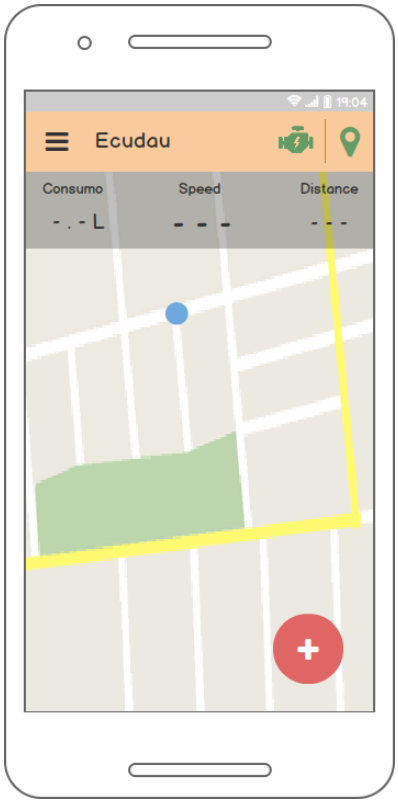

Ilustración 10 – Pantalla principal (Conectado)

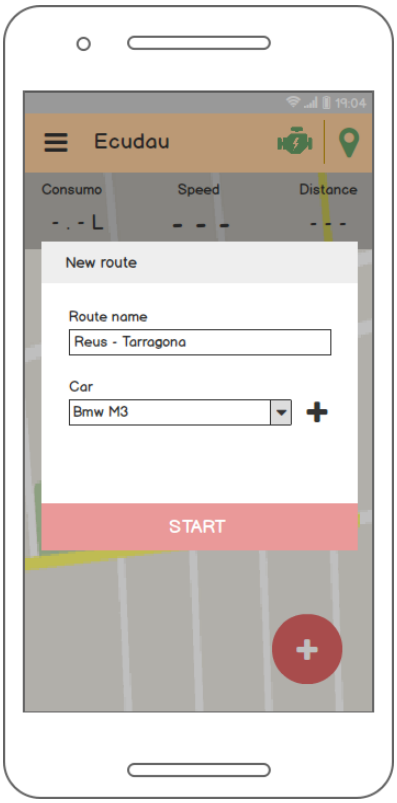

Ilustración 11 – Empezar ruta

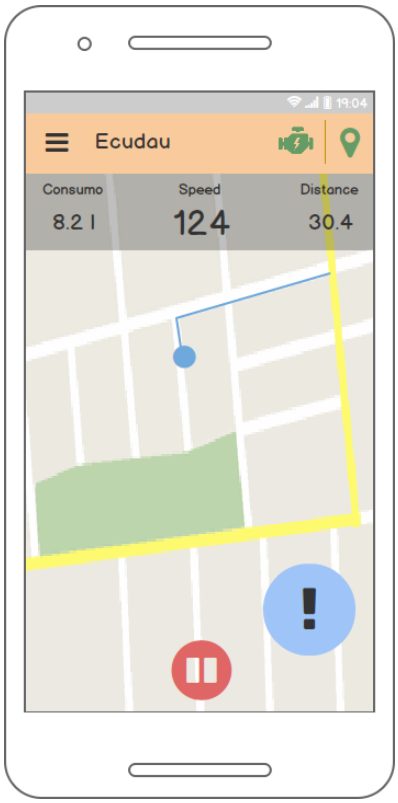

Ilustración 12 – En ruta

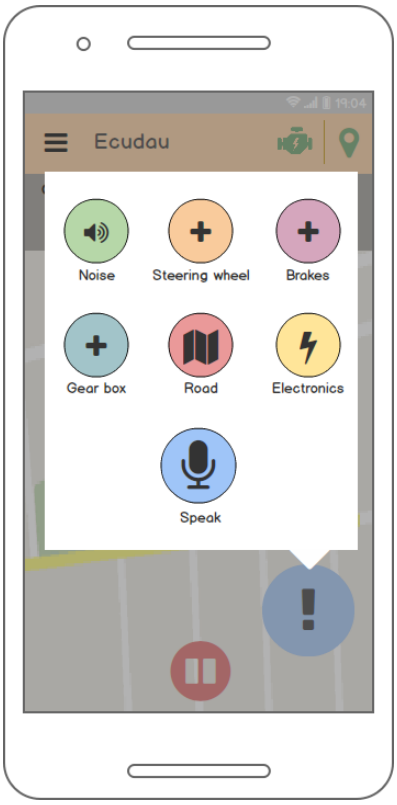

Ilustración 13 – Crear incidencia

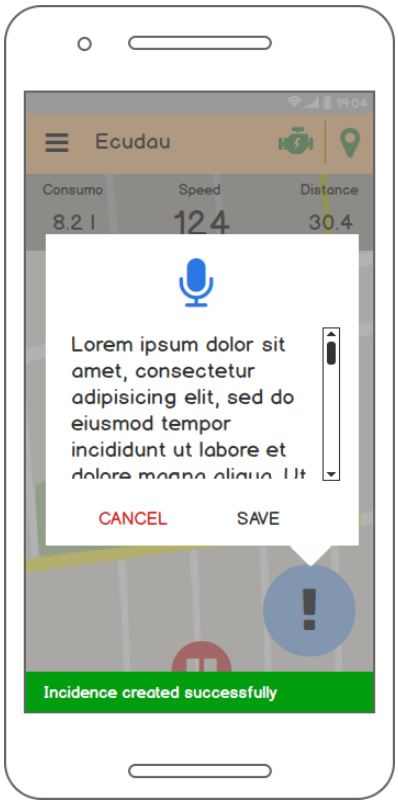

Ilustración 14 – Dictar incidencia

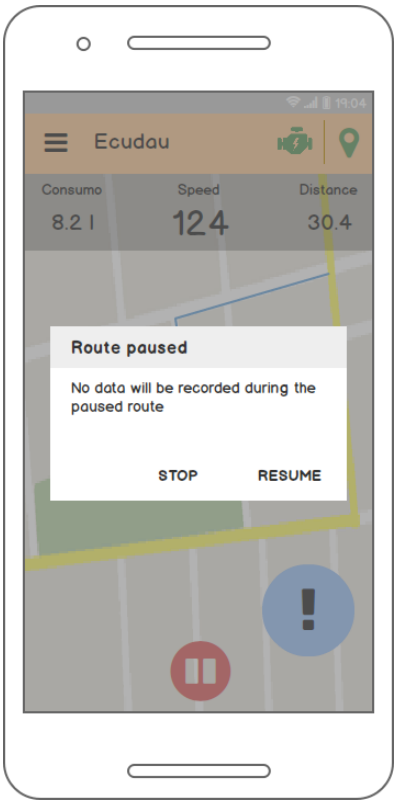

Ilustración 15 - Ruta pausada

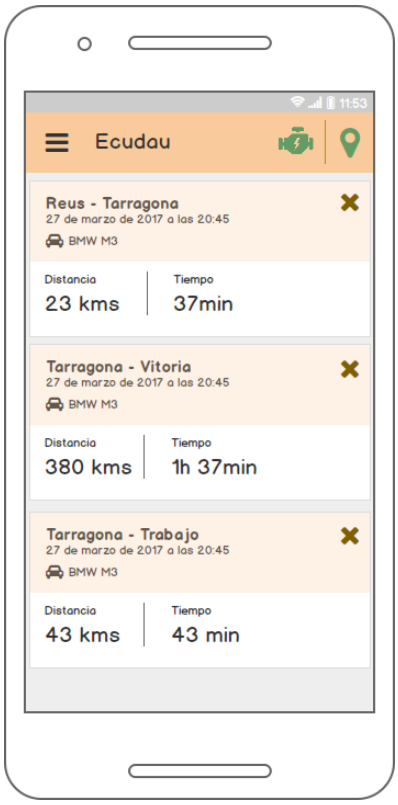

Ilustración 16 - Lista de rutas

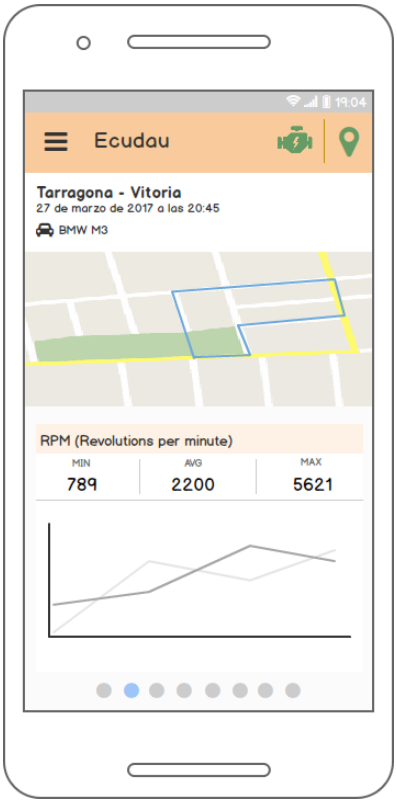

Ilustración 17 - Estadísticas de ruta

4.1 Árbol de navegación

A continuación se detalla el árbol de navegación de las diferentes pantallas de la aplicación.
Siguiendo el objetivo principal de la aplicación vemos que es una navegación muy sencilla donde no hay apenas profundidad en sus nodos, de esta manera el usuario sabrá en todo momento donde se encuentra dentro de la aplicación.

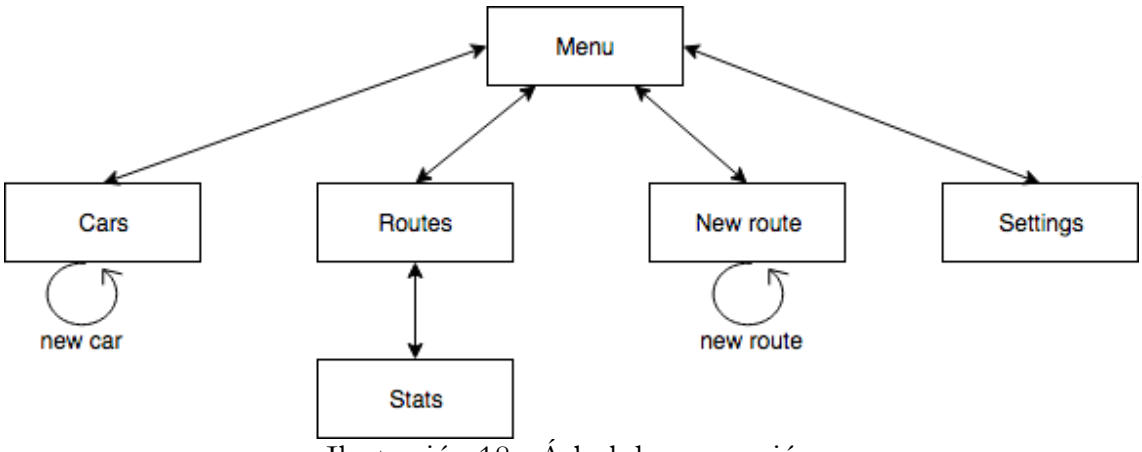

Ilustración 18 - Árbol de navegación

5. Evaluación

En este capítulo se detallarán los casos de uso así como el diseño de la arquitectura seleccionada para desarrollar la aplicación

5.1 Definición de los casos de uso

Ilustración 19 - Diagrama de casos de uso

UC01 - Listar coches	
Actores	Usuario
Precondiciones	-
Flujo	*Flujo principal:* 1. El usuario desde cualquier parte de la aplicación abre el menú lateral. 2. Entre las opciones del menú principal selecciona la opción *"Mis coches"*
Postcondiciones	El usuario verá un listado con los coches creados anteriormente.

Tabla 10 - Caso de uso "Listar coches"

UC02 - Guardar coche	
Actores	Usuario
Precondiciones	*UC02.* Es necesario listar los coches primero *UC01.* Es necesario tener creado un coche (solo al editar)
Flujo	Flujo principal (crear): 1. El usuario desde cualquier parte de la aplicación abre el menú lateral. 2. Entre las opciones del menú principal selecciona la opción *"Mis coches"* 3. El usuario hace clic en el botón de crear un nuevo vehículo. 4. El sistema abrirá un *dialog* donde completar los campos de "Fabricante" y "Modelo" 5. El usuario completará los campos y dará al botón de guardar. Flujo alternativo (editar): 1. El usuario desde cualquier parte de la aplicación abre el menú lateral. 2. Entre las opciones del menú principal selecciona la opción *"Mis coches"* 3. El usuario hace clic en uno de los vehículos creados. 4. El sistema abrirá un *dialog* donde modificar los campos de "Fabricante" y "Modelo" 5. El usuario modificará los campos y dará al botón de guardar.
Postcondiciones	Un nuevo coche se creará se creará en el sistema y será visible en el listado de coches.

Tabla 11 - Caso de uso "Guardar coche"

UC03 - Borrar coche	
Actores	Usuario
Precondiciones	*UC01.* Es necesario tener creado un coche *UC02.* Es necesario listar los coches primero El coche no debe tener rutas asociadas.
Flujo	Flujo principal: 1. El usuario desde cualquier parte de la aplicación abre el menú lateral. 2. Entre las opciones del menú principal selecciona la opción *"Mis coches"* 3. El usuario hace clic en el botón de crear un nuevo vehículo. 4. El sistema abrirá un *dialog* donde completar los campos de "Fabricante" y "Modelo" 5. El usuario dará al botón de "Eliminar". 6. El sistema detecta que no hay ninguna ruta realizada con ese vehículo y elimina el coche. Flujo alternativo: 1. El usuario desde cualquier parte de la aplicación abre el menú lateral. 2. Entre las opciones del menú principal selecciona la opción *"Mis coches"* 3. El usuario hace clic en el botón de crear un nuevo vehículo. 4. El sistema abrirá un *dialog* donde completar los campos de "Fabricante" y "Modelo" 5. El usuario dará al botón de "Eliminar". 6. El sistema detecta que existen rutas realizadas con ese coche y alerta al usuario de borrar primero las rutas hechas con ese coche no permitiendo al usuario borrar el coche.
Postcondiciones	El coche seleccionado será eliminado de la aplicación y no será visible en el listado de coches.

Tabla 12 - Caso de uso "Borrar coche"

UC04 - Conectar GPS	
Actores	Usuario
Precondiciones	El usuario tiene que tener el GPS del dispositivo encendido.
Flujo	Flujo principal: 1. El usuario desde cualquier parte de la aplicación hace clic en el icono de GPS situado en la barra de menú.
Postcondiciones	El usuario verá que el sistema detecta que el GPS está obteniendo datos, poniéndose el icono de color verde.

Tabla 13 - Caso de uso "Conectar GPS"

UC05 - Conectar OBD	
Actores	Usuario
Precondiciones	El usuario tiene que tener el Bluetooth del dispositivo encendido.
Flujo	*Flujo principal:* 1. El usuario desde cualquier parte de la aplicación hace clic en el icono de OBD situado en la barra de menú. 2. El sistema abrirá un *dialog* con dispositivos *bluetooth* enlazados previamente. 3. El usuario seleccionará el correspondiente al OBD. Flujo principal: 1. El usuario desde cualquier parte de la aplicación hace clic en el icono de OBD situado en la barra de menú. 2. El sistema abrirá un *dialog* con dispositivos *bluetooth* enlazados previamente. 3. El usuario no encuentra en el listado el correspondiente al adaptador OBD. 4. El usuario clica el botón de "Buscar dispositivo" 5. El conector Bluetooth OBD aparece en pantalla 6. El usuario conecta el dispositivo OBD.
Postcondiciones	El usuario verá que el sistema detecta que el OBD está obteniendo datos.

Tabla 14 - Caso de uso "Conectar OBD"

UC06 - Crear ruta	
Actores	Usuario
Precondiciones	*UC04*. El GPS tiene que estar conectado. *UC05*. El OBD tiene que estar conectado.
Flujo	Flujo principal: 1. El usuario desde la pantalla principal le dará al botón de crear ruta. 2. El sistema abrirá un *dialog* donde poner un nombre a la ruta y donde elegir el vehículo para la ruta. 3. El usuario escribe un nombre para la ruta 4. El usuario selecciona un coche 5. El usuario pulsa el botón de "Empezar"
Postcondiciones	El sistema creará una nueva ruta donde registrará datos de la ECU y de la localización del usuario.

Tabla 15 - Caso de uso "Crear ruta"

UC07 - Crear incidencia	
Actores	Usuario
Precondiciones	*UC06*. El usuario tiene que estar registrando una ruta. *UC04*. El GPS tiene que estar conectado. *UC05*. El OBD tiene que estar conectado.
Flujo	Flujo principal: 1. El usuario desde la pantalla de ruta en curso hará clic en el botón de registrar una nueva incidencia. 2. El sistema abrirá un *dialog* donde el usuario puede crear con un simple clic una incidencia. 3. El usuario selecciona un tipo de incidencia. 4. El sistema muestra un mensaje de confirmación de que la incidencia se ha creado correctamente. Flujo alternativo: 5. El usuario desde la pantalla de ruta en curso hará clic en el botón de registrar una nueva incidencia. 6. El sistema abrirá un *dialog* donde el usuario puede crear con un simple clic una incidencia.

	7. El usuario selecciona la incidencia de tipo hablado. 8. El usuario dicta la descripción de la incidencia. 9. El sistema registra lo que el usuario va dictando. 10. El usuario hace clic en el botón de "Guardar". 11. El sistema muestra un mensaje de confirmación de que la incidencia se ha creado correctamente.
Postcondiciones	Una nueva incidencia se creará para la ruta en curso en un tiempo y en unas coordenadas concretas.

Tabla 16 - Caso de uso "Crear incidencia"

UC08 - Abrir ruta	
Actores	Usuario
Precondiciones	*UC06*. Es necesario tener creada una ruta *UC10*. Es necesario listar las rutas primero
Flujo	Flujo principal: 1. El usuario desde cualquier parte de la aplicación abre el menú lateral. 2. Entre las opciones del menú principal selecciona la opción *"Mis rutas"* 3. El usuario hará clic sobre aquella ruta que quiera abrir. 4. El sistema mostrará una nueva pantalla con los detalles de la ruta.
Postcondiciones	Una nueva pantalla se abrirá con los detalles y estadísticas de la ruta seleccionada.

Tabla 17 - Caso de uso "Abrir ruta"

UC09 - Borrar ruta	
Actores	Usuario
Precondiciones	*UC06*. Es necesario tener creada una ruta *UC10*. Es necesario listar las rutas primero
Flujo	Flujo principal: 5. El usuario desde cualquier parte de la aplicación abre el menú lateral. 6. Entre las opciones del menú principal selecciona la opción *"Mis rutas"*

	7. El usuario hará clic en el icono de borrar de aquella ruta que desee borrar. 8. El sistema borrará la ruta y no será visible en el listado de rutas.
Postcondiciones	La ruta seleccionada será eliminada de la aplicación y no será visible en el listado de rutas.

Tabla 18 - Caso de uso "Borrar ruta"

UC10 - Listar rutas	
Actores	Usuario
Precondiciones	-
Flujo	Flujo principal: 1. El usuario desde cualquier parte de la aplicación abre el menú lateral. 2. Entre las opciones del menú principal selecciona la opción *"Mis rutas"*
Postcondiciones	El usuario verá un listado con las rutas creadas anteriormente.

Tabla 19 - Caso de uso "Listar rutas"

5.2. Diseño de la arquitectura

La característica principal del MVC es la división entre los datos, la lógica de negocio y la interfaz de usuario. De esta forma si realizamos un cambio en el controlador no afectará a los otros aspectos de la aplicación y lo convierte más modular.

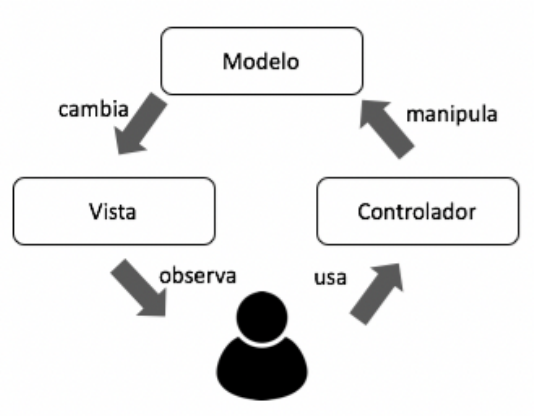

Ilustración 20 - Arquitectura MVC

El **modelo:** Es la representación de la información con la cual el sistema opera compuesta por las clases del modelo de datos.

La **vista:** Es la encargada de mostrar los datos (Fragments[17], Activities[18], views[19]...)

El **controlador:** Es el encargado de manipular los datos de la aplicación. Estará compuesta por todas las clases de negocio (Data Transfer Objects[20], Business Objects[21], etc)

5.3 Base de datos

Para gestionar la alta cantidad de información que tendrá que almacenar la aplicación se optará por usar una base de datos local en el propio dispositivo.

Después de una investigación previa de las opciones que aporta hoy en día diferentes ORM para dispositivos Android se ha optado por Realm[22].
Realm es uno de los últimos ORMs salidos al mercado y tras mirar varios informes de benchmarking, parece que las velocidades de escritura y lectura superan a la mayoría de sus competidores.
Dado que se necesitará almacenar mucha información por minuto, se posiciona como la mejor opción.

5.3.1 Diagrama de clases

El diagrama de clases, nos va a permitir mostrar la relación entre las entidades de nuestra app, y que conforman parte del controlador de nuestra arquitectura:

Route

El objeto ruta será la principal clase de la aplicación. Cada objeto ruta tendrá un vehículo asociado y un listado de incidencias.
Además, también compone esta clase un listado de datos obtenidos tanto por el GPS como por el OBD.

OBD Data:

Esta clase representa un conjunto de datos obtenidos en un momento y lugar determinado. Por este motivo, lo componen por un lado datos de coordenadas y por otro, un objeto fecha.
Los datos se almacenarán en un listado de OBD Items

OBD Ítem:

Representa la unidad mínima de información. Cada objeto estará formado por una descripción, el valor obtenido, las unidades de medida, el mínimo y el máximo valor posible.

Car:

El coche es una entidad básica para poder asociar las rutas que crea el usuario con un vehículo concreto.

Incidence:

La clase incidencia representa una observación subjetiva que el usuario ha captado y quiere reflejarla en la ruta. Las incidencias, como los datos objetivos, se crean en un momento y en un lugar específico, por este motivo, se almacena un objeto fecha y las coordenadas geográficas.
El campo descripción servirá para almacenar la información dictada por el usuario. Cada objeto Incidence tendrá un tipo de incidencia asociado.

Incidence Type:

Representa un tipo concreto de incidencia. Simplemente sirve para poder diferenciar unas incidencias de otras.

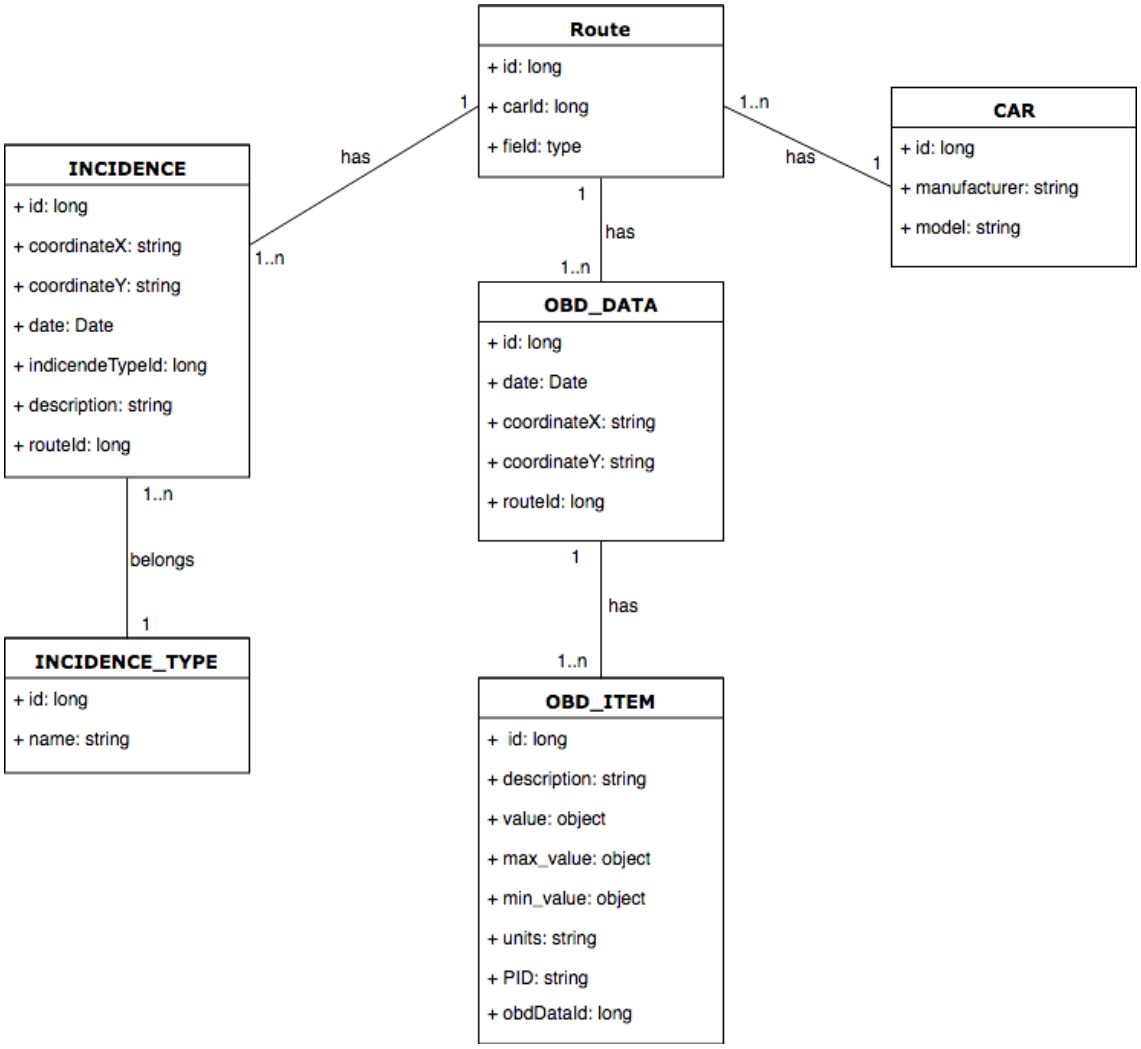

Ilustración 21 - Diagrama de clases

6. Implementación

6.1 Introducción

Para desarrollar la aplicación se ha optado por un desarrollo nativo únicamente para smartphones y con soporte para una versión mínima de Android 4.1 (SDK 16). Con esta se cubre el 99% de los dispositivos que hay actualmente en uso, con lo que se traduce en prácticamente toda la comunidad de smartphones Android.

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.4%
4.1.x	Jelly Bean	16	1.7%
4.2.x		17	2.2%
4.3		18	0.6%
4.4	KitKat	19	10.5%
5.0	Lollipop	21	4.9%
5.1		22	18.0%
6.0	Marshmallow	23	26.0%
7.0	Nougat	24	23.0%
7.1		25	7.8%
8.0	Oreo	26	4.1%
8.1		27	0.5%

Ilustración 22 - Distribución de versiones Android

6.2 Herramientas utilizadas

Para el desarrollo de la aplicación se han utilizado las siguientes herramientas.

6.2.1 Android Studio[23]

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA[24]. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan tu productividad durante la compilación de apps para Android.

6.2.2 Adobe Photoshop CS6[25]

Adobe Photoshop es un editor de gráficos rasterizados desarrollado por Adobe Systems Incorporated. Usado principalmente para el retoque de fotografías y gráficos, en el caso de este proyecto ha sido utilizado para la creación de iconos e imágenes customizadas.

6.2.3 Realm Studio[26]

Para gestionar toda la información almacenada en la aplicación móvil de este proyecto se ha optado por Realm Database. Es una base de datos rápida, fácil de usar y una alternativa open source a SQLite.
Se eligió esta tecnología y no otras de características similares principalmente por su velocidad. Viendo algún benchmarking destacaba frente a sus oponentes. Por otro lado, permite hacer inserciones en segundo plano, algo que también ha sido muy útil para este proyecto.
Realm Studio es un cliente de escritorio que permite visualizar de una manera sencilla los datos almacenados en una base de datos Realm.

6.2.4 GitLab[27]

Gitlab es un servicio web de control de versiones y desarrollo de software colaborativo basado en Git. Se ha optado por esta herramienta y no otras como Github dado que era una herramienta que ya conocía y que permite gestionar repositorios privados de manera gratuita.

6.2.5 OBDSim[28]

OBDSim es un simulador que proporciona información de un vehículo al que se le hubiese conectado un dispositivo bluetooth al puerto OBD.
Este simulador, configurado a un puerto bluetooth habilitado para envío de información, consigue enviar información como revoluciones por minuto, velocidad, posición del acelerador, etc…, como si fuese un vehículo de verdad.
El uso de este simulador ha sido crítico, dado que sin él, la dependencia de un vehículo físico hubiese hecho imposible el desarrollo de esta aplicación.

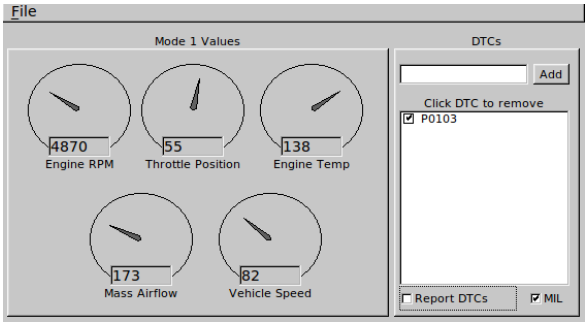

Ilustración 23 - OBDSim

6.3 Librerías y APIs

Para el desarrollo de este proyecto, se han utilizado varias librerías y a continuación se detallan algunas de las más importantes.

6.3.1 AndrOBD

Como se comentó en capítulos anteriores de este documento, este proyecto partía de una aplicación ya hecha y Open Source llamada AndOBD.
Esta aplicación ya tenía implementado la parte de la conectividad bluetooth y el envío de códigos con la información del vehículo.
Todo este negocio se ha reutilizado, no obstante se han añadido nuevas funcionalidades y se ha implementado un rediseño completo de la aplicación.

6.3.2 Google Maps[29]

Google Maps es un servidor de aplicaciones de mapas en la web que pertenece a Google.
Esta librería se ha utilizado tanto en la creación de una nueva ruta como posteriormente en el apartado de estadísticas de la ruta.
Se ha optado por este servicio y no otros por sus infinitas posibilidades, por su gratuidad, al menos para el uso en esta aplicación y por disponer de un modo Lite, que optimiza su uso en listas.

6.3.3 MaterialDrawer[30]

MaterialDrawer es una librería flexible y fácil de usar que proporciona un menú lateral Drawer con una configuración mínima.
Además sigue la guía de estilos de Material Design[31], una guía de estilos diseñada por Google para, entre otros, proyectos Android.
Se ha optado por este componente en vez de implementar el menú desde cero por su bajo coste en cuanto a tiempo, su simpleza y rendimiento.

6.3.4 MPAndroidChart[32]

MPAndroidChart es una librería para crear gráficos de datos en proyectos Android.
Esta librería se ha utilizado para mostrar las estadísticas de una ruta.

6.3.5 PageIndicatorView[33]

Esta librería añade de una manera muy sencilla un indicador de la página actual del componente ViewPager de Android.
Esta librería se ha utilizado para mostrar las estadísticas de una ruta.

6.4 Implementación de Ecudau

6.4.1 Estructura de la app

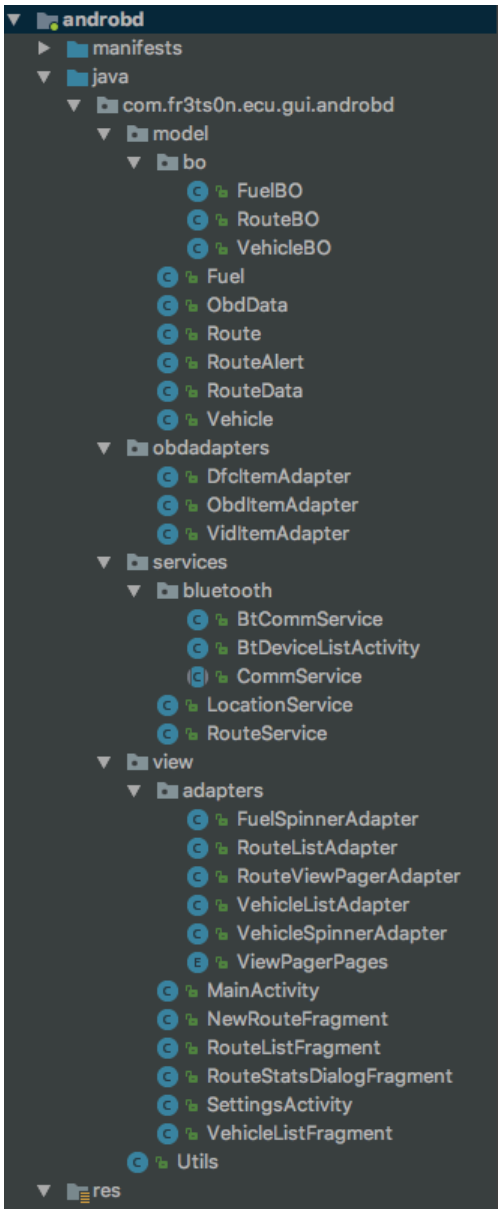

Ilustración 24 - Estructura de la app

Una vez hecho el refactor de aquellas clases que no son necesarias para el objetivo de este proyecto, el árbol del proyecto ha quedado de la siguiente manera:

Data adapters:
En este directorio se encuentran las clases que se encargan de la transferencia de información entre la librería (library) de conexión con el dispositivo bluetooth OBD2 y la aplicación.

Model:
En este directorio se encuentran las clases del modelo. Estas clases extienden de RealmObject para que puedan ser utilizadas por Realm Database.

Model.BO:
En este directorio se encuentran las clases que realizan operaciones (crear, obtener, eliminar y modificar) sobre las diferentes clases del modelo.

Services:
En este paquete encontramos clases que ofrecen un servicio a la aplicación. Hay principalmente 3 servicios: un servicio que gestiona la localización, uno que gestiona la conectividad bluetooth y otro que gestiona operaciones que se realizan al crear una ruta.

View:
En este directorio se encuentran todas las clases que se encargan de gestionar la vista de la aplicación. Activities y fragments.

View.Adapters:
En este paquete se encuentran los adaptadores necesarios para algunos de los componentes que forman la vista de la aplicación.
Adaptadores para el componente Spinner[34] y adaptadores para el componente RecyclerView[35].

6.4.2 Revisión del modelado

Durante el proceso de implementación, surgen algunas variaciones con respecto al modelo de clases planificado. Surgen necesidades relacionadas con la visualización en las vistas y con cálculos de valores que hace que tengamos que incluir algunos atributos adicionales a las clases ideadas en el diseño. Además, también se ha modificado algunos nombres para hacer más intuitivo el modelo:

Incidence → RouteAlert
OBD_Data → RouteData
OBD_Item → ObdData
Car → Vehicle

A continuación se detallan los cambios respecto al modelo de clases planificado:

Route:
Se han añadido los campos:
- Description: Nombre para la ruta que cree el usuario.
- Start date: Fecha/hora de comienzo de la ruta
- End date: Fecha/hora de comienzo de la ruta

ObdData:
Se han añadido los siguientes campos para poder registrar todos los datos que devuelve la librería para la conexión con el vehículo. A continuación se muestra un listado con todos los valores, una descripción y un ejemplo:

- Max: Máximo valor posible del valor solicitado
  - "655.35"
- Description: Literal con la descripción del código solicitado
  - "Air Flow Rate (MAF sensor)"
- Min: Mínimo valor posible del valor solicitado
  - "0.0"
- Value: Valor del código solicitado
  - "4.82"
- Fmt: Patrón para dar formato al valor obtenido
  - "%.2f"
- Pid: Código del valor solicitado
  - "16"
- Mnemonic: Código mnemotécnico del valor solicitado
  - "mass_airflow"
- Units: Unidad del valor solicitado
  - "g/s"

RouteAlert:
Se ha eliminado la clase RouteAlertType para simplificar el modelo. Como consecuencia, el tipo de incidencia se almacena en forma de String en la clase RouteAlert.

Fuel:
Debido a que para calcular el consumo del vehículo es necesario especificar el combustible, ha sido necesario añadir la clase Fuel. En ella se almacena el tipo de combustible y dos valores numéricos para realizar el cálculo del consumo.

El diagrama de clases después de las modificaciones necesarias ha quedado de la siguiente manera:

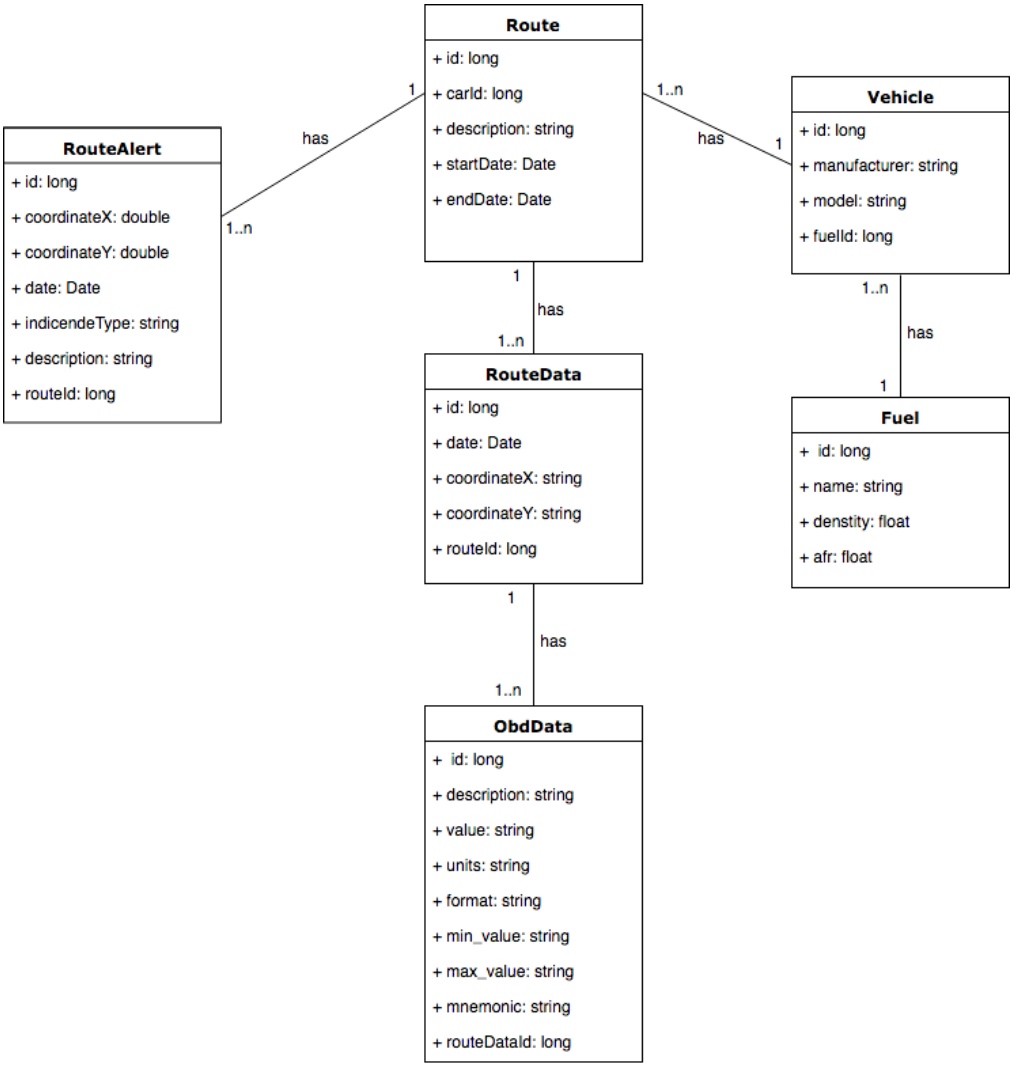

Ilustración 25 - Diagrama de clases definitivo

6.4.3 Revisión de la vista

A la hora de implementar Ecudau, partimos del diseño de prototipos realizados. Durante el proceso de desarrollo de vistas han surgidos diferentes variaciones que se van a listar y a justificar a continuación.

Iconos de estado de conectividad

Se ha decidido eliminar estos iconos de la barra superior por dos motivos. El primero, porque el usuario ya obtiene feedback del estado de la conectividad mediante el botón de conectar o empezar una nueva ruta.
Si el botón está azul, es que no hay ningún OBD conectado, si por el contrario está rojo, hay conexión y se puede empezar una nueva ruta.
El segundo motivo es por estética, quedando una vista más limpia.

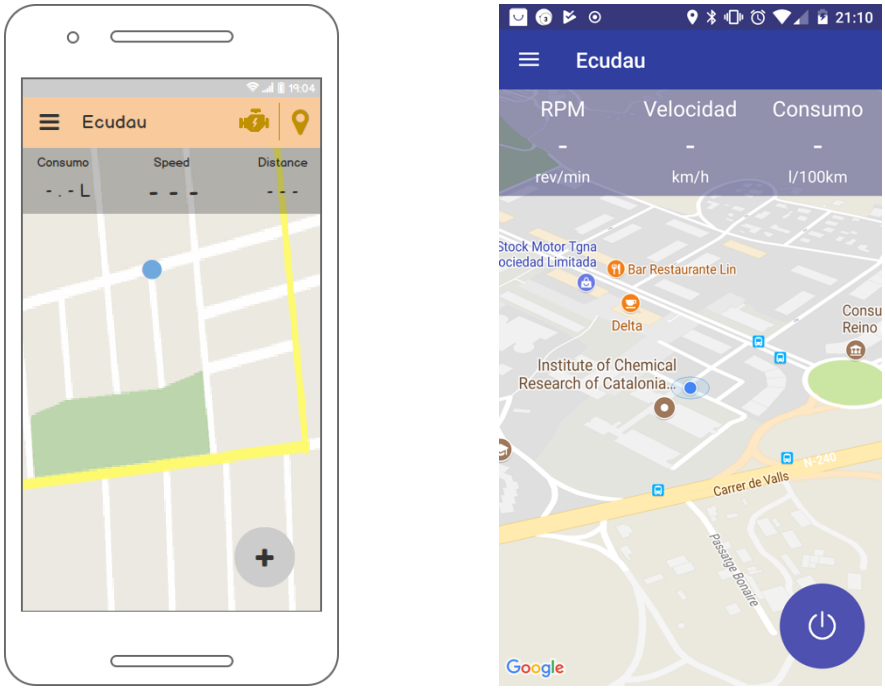

Ilustración 26 - Cambios de vista: menú superior

Listado de rutas

Se ha incluido una pre-visualización de la ruta en el ítem del listado para poder identificar mejor la ruta.

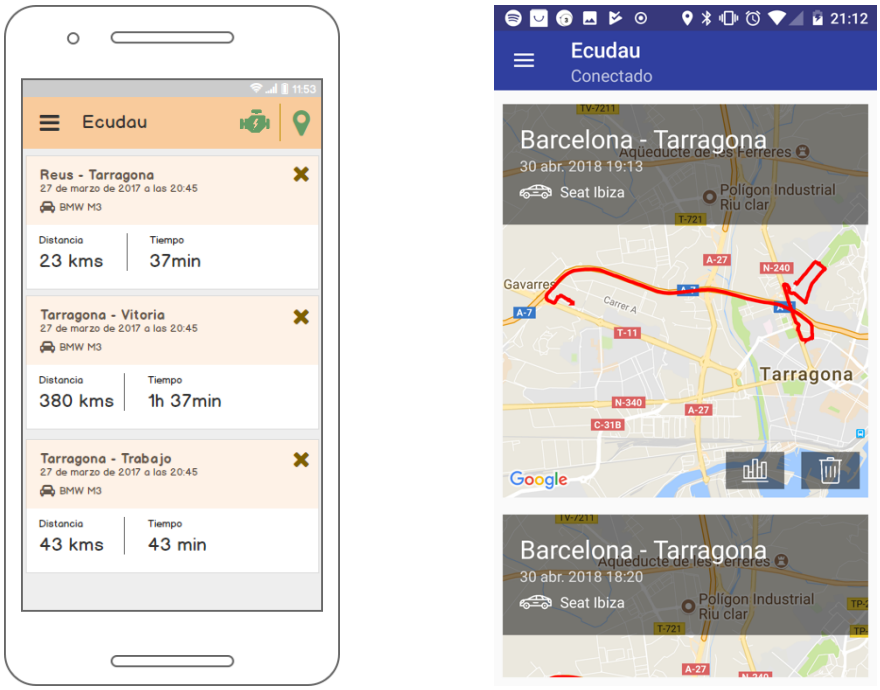

Ilustración 27 – Cambios de vista: listado de rutas

Incluir el combustible en el vehículo

Durante la implementación ha sido obligatorio incluir el tipo de combustible del vehículo dado que es necesario para poder realizar los cálculos del consumo. Los datos que han sido necesarios son el AFR[36] (air-fuel-ratio) y la densidad del combustible, ambos valores son numéricos.
Para este proyecto únicamente se han añadido los dos combustibles más comunes: diésel y gasolina.

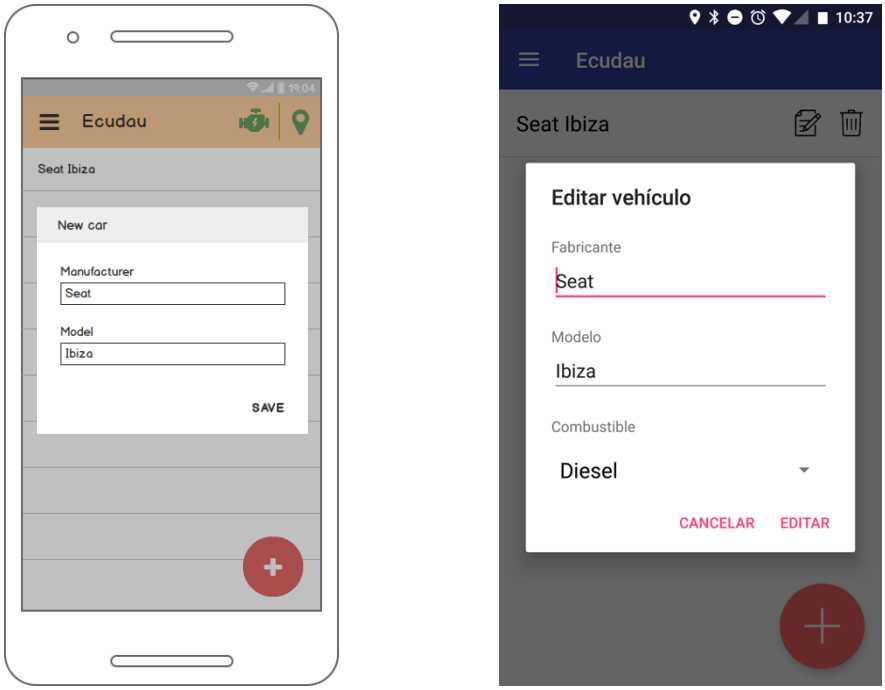

Ilustración 28 - Cambios de vista: combustible

6.4.4 Revisión de la planificación

Para la fase de implementación de esta aplicación se contaba con una dedicación aproximada de 186 horas, repartidas en una dedicación de 3 horas entre semana y 8 horas en días festivos.
Esta planificación ha sido posible cumplirla para la mayoría de los días, no obstante, algunos días laborables solo se ha podido dedicar un máximo de dos horas teniendo que recuperar estas horas en fin de semana, llegando a dedicar hasta un máximo de 13 horas por día.

Durante la implementación se han ido encontrando ciertos problemas que se han ido solucionando, pero como consecuencia han consumido más tiempo del planificado. A continuación se detallan los más destacados:

Posición del GPS inexacta
Descripción
Creando una ruta, se observaba que había ciertos momentos en los que el GPS devolvía un punto en el mapa muy alejado de la posición real del dispositivo. Como consecuencia, cuando se dibujaba la ruta en el mapa había picos que se alejaban totalmente del camino trazado por el vehículo.
Solución
Después de investigar, estaba utilizando una librería bastante antigua de localización. Leyendo la documentación oficial, se sustituye la librería y los callbacks de actualización de posición y el problema queda solucionado.

Tabla 20 - Error: GPS Inexacto

Tiempo de actualización en pantalla con mucho retraso
Descripción
Al hacer la ruta, la información en tiempo-real no se podía considerar en tiempo-real. Había mucho retraso entre que se revolucionaba el coche y se veía reflejado en la pantalla.
Solución
Leyendo la documentación oficial de la aplicación en la que se basa este proyecto, afirmaba que la cantidad de datos solicitados a la ECU del vehículo era directamente proporcional al tiempo de respuesta con los datos. Se estaban solicitando un total de 42 datos, cuando solamente eran necesario 5 (rpm, velocidad, flujo de aire y sonda lambda). Una vez limitados estos valores el problema quedó solucionado.

Tabla 21 - Error: Tiempo de refresco con retraso

Demasiada carga de trabajo en el hilo principal
Descripción
Constantemente veía por la consola de desarrollo que un número alto de frames no se estaban renderizando. Al parecer se estaba realizando mucha carga de trabajo en el hilo principal.
Solución
Después de inspeccionar el código se observa que el problema es que las inserciones en base de datos (10 cada 400ms después de optimizarlo) se estaban ejecutando en el hilo principal. La solución fue hacer estos inserts en segundo plano, Realm ya dispone una función que lo hace de manera automática.

Tabla 22 - Error: Demasiada carga en el hilo principal

La consecuencia de dar solución a estos problemas es que hay una funcionalidad que ha quedado fuera del alcance de este proyecto. Se pueden registrar incidencias en ruta, pero no será posible grabar una incidencia mediante reconocimiento de voz.

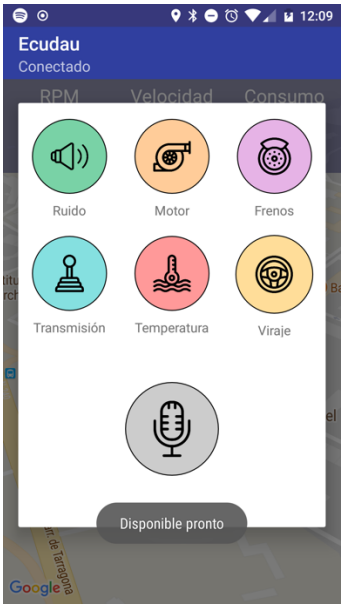

Ilustración 29 - Alertas de audio desactivadas

6.5 Pruebas

Para realizar las pruebas se ha utilizado un dispositivo físico como smartphone.
Para el vehículo se ha utilizado tanto un emulador como un vehículo físico.

Smartphone:
Samsung Galaxy S5 con sistema operativo Android 7.1.2

Vehículo emulado:
OBDSim instalado en un portátil Lenovo T400 con sistema operativo Windows 10.

Vehículo físico:
Seat Ibiza 1.6tdi (2010)

A continuación se detallan las pruebas realizadas:

Conectar OBD		
Prueba	Detalles	Resultado
Conectar un OBD sin tenerlo vinculado.	Bluetooth conectado.	OK. No aparece en la lista. Hace una búsqueda y una vez encontrado se conecta sin problemas.
Intentar conectar a un dispositivo que no sea un OBD.	Bluetooth conectado.	OK. Aparece un error por pantalla diciendo que no ha sido posible conectar el dispositivo.
Conectar un OBD teniéndolo ya vinculado	Bluetooth conectado.	OK. Se selecciona el dispositivo de la lista y se conecta sin problemas.
Abrir la aplicación.	Bluetooth desconectado	OK. La aplicación solicita permisos para conectar el bluetooth. Una vez los permiso son concedidos ya se puede conectar un dispositivo.

Tabla 23 - Pruebas: Conectar OBD

Grabar ruta		
Prueba	Detalles	Resultado
Crear una ruta sin vehículo disponible	OBD conectado.	OK. En vez de salir el diálogo para empezar ruta, sale un aviso diciendo que antes hay que crear un vehículo.
Crear una ruta sin descripción	OBD conectado.	OK. Aparece un error diciendo que se trata de un campo obligatorio.
Crear una ruta sin vehículo seleccionado	OBD conectado.	OK. Aparece un error diciendo que se trata de un campo obligatorio.
Ir al listado de rutas en medio de la creación de una ruta.	OBD conectado. Ruta empezada.	OK. El menú aparece bloqueado una vez empieza la ruta. Se desbloquea al finalizar.
Pausar/Reanudar ruta	OBD conectado. Ruta empezada.	OK. Aparece un mensaje de pausa. En el tiempo que este mensaje aparece no se registra ningún dato en el sistema.
Guardar ruta	OBD conectado. Ruta empezada.	OK. La ruta finaliza correctamente y aparece mensaje de confirmación.
Crear alerta/incidencia	OBD conectado. Ruta empezada.	OK. Aparece un diálogo donde seleccionar el tipo de incidencia a crear. Una vez seleccionada aparece un mensaje de confirmación.

Tabla 24 - Pruebas: Grabar ruta

Listar rutas		
Prueba	Detalles	Resultado
Abrir el listado sin ninguna ruta creada	-	OK. Aparece un listado vacío.
Abrir el listado donde una ruta no tiene datos de localización	-	OK. Para la ruta sin datos, aparece un mapa vacío sin nada dibujado.
Abrir listado donde todas las rutas tienen datos.	-	OK. Aparece un listado donde cada ruta tiene un mapa con la ruta dibujada.
Eliminar ruta	-	OK. Aparece un mensaje donde confirmaremos la acción. Una vez confirmada, la ruta será eliminada de la aplicación.

Tabla 25 - Pruebas: Listar rutas

Estadísticas de ruta		
Prueba	Detalles	Resultado
Abrir ruta sin fecha de finalización	-	OK. Se muestra la información de la ruta y en el campo de la duración aparece "undefined"
Ruta sin datos del vehículo	-	OK. Aparece un mapa sin ruta y las gráficas vacías.
Ruta con datos	-	OK. Aparece un mapa interactivo y gráficas con datos de la ruta.

Tabla 26 - Pruebas: Estadísticas de ruta

Listar vehículos		
Prueba	Detalles	Resultado
Abrir el listado sin vehículos	-	OK. Aparece un listado vacío.
Abrir el listado sin vehículos	-	OK. Aparece un listado con vehículos.
Crear/modificar vehículo sin datos necesarios	-	OK. Aparece un error advirtiendo de la obligatoriedad de los campos.
Crear/modificar vehículo con los datos necesarios	-	OK. Se crea/modifica el vehículo correctamente
Eliminar vehículo sin rutas asociadas	-	OK. Se elimina correctamente el vehículo.
Eliminar vehículo con rutas asociadas	-	OK. No se elimina el vehículo para que las rutas no queden inconsistentes apareciendo un mensaje describiendo la acción.

Tabla 27 - Pruebas: Listar vehículos

7. Conclusiones

El trabajo realizado en esta asignatura, ha permitido generar una app válida y funcional para cumplir con el objetivo planteado desde el inicio. Una app para crear y analizar rutas creadas con un vehículo.
El resultado, a nivel personal es bastante satisfactorio, a pesar de haber dejado de lado alguna funcionalidad.
Estos objetivos secundarios que no se ha podido realizar, en concreto la funcionalidad registrar incidencias mediante dictados por voz, ha sido como consecuencia de ir afrontando problemas durante el desarrollo y viendo como el tiempo disponible para finalizar la app se iba ajustando cada vez más.

En cuanto a la elección de la metodología de desarrollo fue un acierto elegir Metodología Ágil. Dedicar lo mínimo posible a documentar y dedicar lo máximo posible a la implementación del producto.
Se definió un MVP (Producto viable mínimo) que pudiese crear y analizar rutas y se ha podido llegar al objetivo.
Como resultado, tenemos una aplicación base que registra velocidad, revoluciones por minuto y, la más importante, consumo. Estos datos son los más relevantes para un ciudadano de a pie.

Ahora que tenemos la aplicación base, podemos ampliar sus funcionalidades:

- Subir los datos a la nube y hacer análisis más exhaustivos mediante interfaz web.
- Hacer benchmarking de vehículos haciendo la misma ruta.
- Obtener más datos del vehículo.
- Hacer una red social de conductores donde puedan ver, comentar y puntuar rutas.
- Exportar estadísticas a PDF.

Por último, expresar la satisfacción personal de elaborar un proyecto como este, donde he podido aplicar los conocimientos adquiridos en este Master.
Siempre he querido juntar mis dos aficiones, el desarrollo de aplicaciones y el automovilismo. Ahora que tengo una aplicación base, tengo donde poder seguir trabajando y disfrutando de este producto y quizás, algún día, poder comercializarlo.

8. Bibliografía

[1] IOT
https://es.wikipedia.org/wiki/Internet_de_las_cosas

[2] Torque PRO
https://torque-bhp.com/
https://play.google.com/store/apps/details?id=org.prowl.torque

[3] ECU
https://es.wikipedia.org/wiki/Unidad_de_control_de_motor

[4] OBD Car Doctor
https://obd-car-doctor.com/en-us/
https://play.google.com/store/apps/details?id=com.pnn.obdcardoctor

[5] Obd Arny – OBD2
https://play.google.com/store/apps/details?id=com.elm.scan.obd.arny

[6] OBD
https://es.wikipedia.org/wiki/OBD

[7] MVP
https://es.wikipedia.org/wiki/Producto_viable_m%C3%ADnimo

[8] MVC
https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador

[9] AndrOBD
https://github.com/fr3ts0n/AndrOBD

[10] Google Speech
https://developer.android.com/reference/android/speech/SpeechRecognizer.html

[11] Strava
https://www.strava.com/
https://play.google.com/store/apps/details?id=com.strava

[12] Waze
https://www.waze.com/es/
https://play.google.com/store/apps/details?id=com.waze

[13] ORM
https://es.wikipedia.org/wiki/Mapeo_objeto-relacional

[14] CRUD
https://es.wikipedia.org/wiki/CRUD

[15] Google Firebase
https://firebase.google.com

[16] Balsamiq
https://balsamiq.com/

[17] Android Fragment
https://developer.android.com/guide/components/fragments.html?hl=es-419

[18] Android Activity
https://developer.android.com/reference/android/app/Activity.html

[19] Android view
https://developer.android.com/reference/android/view/View.html

[20] Data Transfer Object
https://es.wikipedia.org/wiki/Objeto_de_transferencia_de_datos

[21] Business Object
https://es.wikipedia.org/wiki/Objeto_de_negocio

[22] Realm Database
https://realm.io/products/realm-database

[23] Android Studio
https://developer.android.com/studio/?hl=es-419

[24] IntelliJ IDEA
https://www.jetbrains.com/idea/

[25] Abode Photoshop
https://www.adobe.com/es/products/photoshop.html

[26] Realm Studio
https://www.realm.io/products/realm-studio/

[27] GitLab
https://about.gitlab.com/

[28] OBDSim
https://icculus.org/obdgpslogger/obdsim.html

[29] Google Maps
https://developers.google.com/maps/

[30] MaterialDrawer
https://github.com/mikepenz/MaterialDrawer

[31] Material Design
https://material.io/guidelines/

[32] MPAndroidChart
https://github.com/PhilJay/MPAndroidChart

[33] PageIndicatorView
https://github.com/romandanylyk/PageIndicatorView

[34] Spinner
https://developer.android.com/guide/topics/ui/controls/spinner

[35] RecyclerView
https://developer.android.com/guide/topics/ui/layout/recyclerview

[36] AFR
https://es.wikipedia.org/wiki/Factor_lambda_(AFR)

9. Anexos