



Universidad de Valladolid

Escuela Ingeniería Informática

TRABAJO FIN DE GRADO

Ingeniería Informática
Mención Ingeniería de Software

Routes4Riders

Autor:

Ángel Posada García

Tutor:

Joaquín Adiego Rodríguez

Resumen

Actualmente es innegable la continua implantación y mejora de las tecnologías de ubicación en los dispositivos móviles, así como el uso de las mismas para recibir instrucciones en nuestros desplazamientos cotidianos. El objetivo de este proyecto es aprovechar esta nueva y creciente tecnología no solo en el ámbito mas funcional de la misma, si no crear una herramienta que aúne la utilidad de una guía en nuestros desplazamientos con una herramienta recreativa para los usuarios de motocicletas. La aplicación desarrollada consiste en un sistema de geolocalización en tiempo real, que además permitirá bajo petición del usuario realizar una grabación de los puntos por los que transcurre el recorrido, la aplicación los almacenará posteriormente, además también será capaz de representarlos sobre un mapa mediante líneas. Finalmente permitirá realizar de nuevo dichos recorridos recibiendo indicaciones para el desplazamiento por los mismos.

Índice general

Índice de tablas	7
Índice de figuras	9
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Metodología	2
1.4. Estructura de la memoria	3
2. Estado del arte	5
2.1. Introducción	5
2.2. Método de geo-localización combinando redes de telefonía y satélites	6
2.3. Arquitectura Android	7
3. Planificación	9
3.1. Introducción	9
3.2. Vista general del proyecto	9
3.3. Recursos	9
3.4. Estimación temporal	10
3.5. Desviaciones	12
3.6. Gestión de riesgos	12
3.7. Monitorización de los riesgos	16
4. Análisis	19
4.1. Introducción	19
4.2. Requisitos	19
4.2.1. Requisitos funcionales	19
4.2.2. Requisitos no funcionales	21
4.2.3. Requisitos de información	25
4.3. Restricciones	26
4.4. Modelo de casos de uso	27
4.4.1. Actores primarios	27
4.4.2. Actores secundarios	27
4.4.3. Diagrama de casos de uso	27

4.4.4. Matriz de requisitos frente a casos de uso	29
4.4.5. Especificación de casos de uso de análisis	30
4.5. Realización de casos de uso de análisis	41
4.5.1. Modelo de dominio	41
4.5.2. Diagrama Entidad Relación	43
4.5.3. Diagramas de actividad	44
5. Arquitectura y diseño	55
5.1. Introducción	55
5.2. Aproximación inicial al diseño	55
5.3. Arquitectura propuesta del sistema cliente	56
5.4. Topología del sistema	58
5.5. Mecánica del sistema	59
6. Pruebas	61
6.1. Introducción	61
6.2. Resultado prototipo 1	61
6.2.1. Introducción	61
6.2.2. Pruebas	63
6.2.3. Conclusiones	63
6.3. Resultado prototipo 2	64
6.3.1. Introducción	64
6.3.2. Pruebas	64
6.3.3. Conclusiones	65
7. Conclusiones y líneas futuras	67
7.1. Conclusiones	67
7.1.1. Líneas futuras	68
A. Glosario	71
B. Acrónimos	73
C. Manual de usuario	75
D. Manual de instalación	81
E. Contenido de CD	83
Bibliografía	85

Índice de tablas

3.1. Recursos humanos del proyecto	10
3.2. Recursos técnicos del proyecto	10
3.3. Tabla de restricciones del sistema	11
3.4. Gestión del Riesgo01	13
3.5. Gestión del Riesgo02	13
3.6. Gestión del Riesgo03	14
3.7. Gestión del Riesgo04	14
3.8. Gestión del Riesgo05	15
3.9. Gestión del Riesgo06	15
3.10. Gestión del Riesgo07	15
3.11. Gestión del Riesgo08	16
4.1. Tabla de requisitos funcionales	21
4.2. Tabla de requisitos no funcionales	25
4.3. Tabla de requisitos de información	25
4.4. Tabla de restricciones del sistema	26
4.5. Matriz de requisitos y casos de uso	29
4.6. Descripción del CU01	30
4.7. Descripción del CU02	31
4.8. Descripción del CU03	32
4.9. Descripción del CU04	33
4.10. Descripción del CU05	34
4.11. Descripción del CU06	35
4.12. Descripción del CU07	36
4.13. Descripción del CU08	37
4.14. Descripción del CU09	38
4.15. Descripción del CU10	39
4.16. Descripción del CU11	40

Índice de figuras

1.1. Esquema general de la metodología iterativa e incremental	3
2.1. Esquema general de capas del sistema operativo Android.	7
3.1. Diagrama de Gantt esperado en el proyecto	11
4.1. Diagrama de casos de uso	28
4.2. Diagrama de clases	41
4.3. Diagrama Entidad Relación	43
4.4. Diagrama de actividad del CU01	44
4.5. Diagrama de actividad del CU02	45
4.6. Diagrama de actividad del CU03	46
4.7. Diagrama de actividad del CU04	47
4.8. Diagrama de actividad del CU05	48
4.9. Diagrama de actividad del CU06	49
4.10. Diagrama de actividad del CU07	50
4.11. Diagrama de actividad del CU08	51
4.12. Diagrama de actividad del CU09	52
4.13. Diagrama de actividad del CU10	53
4.14. Diagrama de actividad del CU11	54
5.1. Diagrama general de la arquitectura de la aplicación	55
5.2. Diagrama general de la arquitectura de la aplicación	56
5.3. Diagrama de despliegue del sistema	58
6.1. Pantalla menú del prototipo 1	62
6.2. Pantalla petición permisos prototipo 1	62
C.1. Pantalla registro de la aplicación	75
C.2. Pantalla datos del usuario de la aplicación	76
C.3. Pantalla datos moto de la aplicación	77
C.4. Pantalla carga de la aplicación	78
C.5. Pantalla principal de la aplicación	79
C.6. Pantalla de listado de rutas	79
C.7. Pantalla de previsualización de rutas	80

D.1. Pantalla de seguridad permitir instalación	82
D.2. Pantalla de instalación	82

Capítulo 1

Introducción

1.1. Motivación

Actualmente el uso de las motocicletas además de un fin práctico en los desplazamientos diarios, también tienen un público cuyo objetivo es disfrutar de la misma, aprovechando su tiempo libre para realizar desplazamientos sin un destino, simplemente por el hecho de disfrutar su vehículo por carreteras secundarias con muchas curvas y apenas tráfico.

El presente documento es la propuesta elaborada como respuesta a la asignatura Trabajo Fin de Grado (TFG en adelante) del Grado en Ingeniería Informática mención de Ingeniería de Software de la Escuela de Ingeniería Informática de la Universidad de Valladolid.

A día de hoy la tecnología permite que estos desplazamientos puedan ser almacenados, para en un futuro repetirlos o compartirlos con otros usuario que utilicen la motocicleta como vehículo recreativo. Actualmente gran parte de los sistemas que existen para estos vehículos no llegan más allá de lo que sería un simple GPS, los cuales no permiten un grabado de rutas y tampoco la posibilidad de compartir las mismas. Hay algunos ejemplos previos pero aún con buenas funcionalidades, los usuarios están muy disgregados geográficamente y no existe una comunidad en las mismas.

En el presente Trabajo de Fin de Grado se presenta el desarrollo de una aplicación que permita grabar y compartir las rutas realizadas por los usuarios.

En relación con lo anterior también se busca la integración con servicios de terceros que nos otorguen datos sobre puntos de interés en las rutas grabadas, lo cual no solo haría mas interesante el uso para los usuarios objetivos sino que también fomentaría el comercio local, así como a nivel cultural de las zonas.

Actualmente Google presta su servicio para obtener todo este tipo de información así como las instrucciones para realizar los recorridos previamente grabados, por tanto utilizaremos este servicio ex-

terno en lo referente al almacenamiento, presentación y uso de los datos para representar y recorrer las diferentes rutas almacenadas.

1.2. Objetivos

El objetivo principal de este proyecto es el desarrollo de una aplicación móvil para la plataforma Android que consiste en una aplicación que permita a los usuarios de motocicletas con fines recreativos almacenar rutas en el dispositivo y compartirlas, así como obtener rutas compartidas por otros usuarios. Un desglose de este objetivo final podría ser:

1. Implementar una aplicación integrando GoogleMaps y la geolocalización a través del GPS del dispositivo.
2. Diseño de una interfaz atractiva y usable.
3. Permitir la integración con un servicio externo para compartir la información de las rutas.
4. Aplicar técnicas de redes sociales para crear una comunidad.
5. Integración con servicios de localización, para la ubicación de puntos de interés.
6. Crear un sistema de evaluación de dificultad de rutas e hitos en las mismas para crear un sistema de gamificación.
7. Integración de los servicios de Google maps para representación y realizar recorrido guiado de las rutas previamente almacenadas.
8. Extraer información de opiniones para mejorar la aplicación.
9. Extraer información acerca del crecimiento de la comunidad.

1.3. Metodología

Para este proyecto no es posible utilizar el modelo en cascada de la metodología de ingeniería, esto se debe a que al ser una aplicación relativamente nueva y sin referencias existentes ha de estar sujeta a cambios del equipo de trabajo o requisitos dinámicos.

Junto a esto se une la necesidad de tener pequeñas versiones iniciales para ir probando las diferentes funcionalidades aisladas del producto. En función de estas necesidades, se opta por el modelo iterativo e incremental de la metodología de ingeniería.

La metodología se basa en producir versiones iniciales o prototipos del producto, de forma periódica. Con ello se obtienen mejoras sucesivas del producto, incrementando la calidad y funcionalidad final. En la figura 1.1 se muestra un esquema de los pasos que se siguen en la metodología elegida.

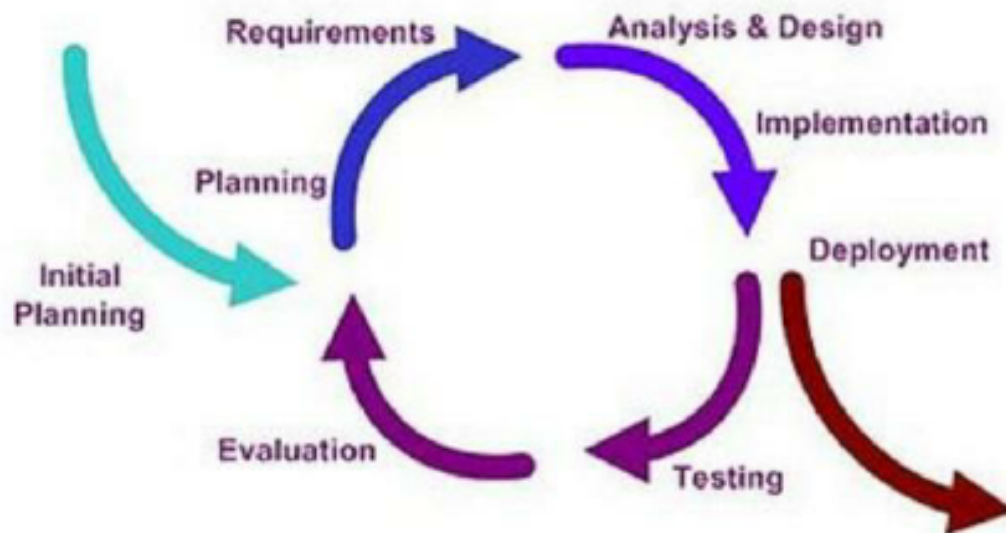


Figura 1.1: Esquema general de la metodología iterativa e incremental

1.4. Estructura de la memoria

La estructura de la memoria presentada en el documento consta de dos partes. La primera parte se compone por el conjunto de capítulos que contienen la información más relevante sobre el proyecto, la segunda esta compuesta por el conjunto de anexos en los cuales se incluye información complementaria. El citado esquema se detalla a continuación:

Capítulos:

1. **Introducción:** Presenta la motivación del proyecto, así como los objetivos a lograr, el contexto en el cual esta enmarcado y la metodología seleccionada y utilizada para llevar a cabo el proyecto.
2. **Marco teórico:** Se describe el estado del arte en el que se encuentra encuadrado el proyecto. En concreto, sobre la tecnología de la localización móvil, mediante redes móviles y satélites GPS, recogida de datos y aspectos de realimentación y adaptación al usuario (gamificación).
3. **Planificación:** Se definen los recursos y la estimación temporal del proyecto así como las desviaciones en el itinerario marcado. Incluimos también la descripción de la gestión de riesgos.
4. **Análisis:** Se definen los requisitos, tanto funcionales como no funcionales, del proyecto. Incluimos también la descripción y modelado de los casos de uso de la aplicación. Finalmente se detalla el modelo de dominio del sistema.
5. **Arquitectura y diseño:** Se define la estructura software del sistema junto con el diseño tanto la interfaz como los procesos realizables por usuario del mismo.
6. **Pruebas:** Se detallan los puntos por los que ha pasado el desarrollo a la hora de comprobar la implementación de las nuevas funcionalidades que se implementaban de forma incremental.

7. **Conclusiones:** Se detallan los puntos más importantes extraídos del proyecto y las líneas de continuación como trabajo futuro.

Anexos:

- A **Glosario:** Anexo donde se definen los conceptos específicos de este proyecto.
- B **Acrónimos:** Se listan las palabras de las siglas que están presentes a lo largo de esta memoria.
- C **Manual de usuario:** Se presenta una guía para el usuario de la aplicación.
- D **Manual de instalación:** Se describen los requerimientos para poder instalar y lanzar la aplicación por los usuarios.
- E **Contenido del CD:** Se define el contenido almacenado en el CD entregado.

Capítulo 2

Estado del arte

2.1. Introducción

En el siguiente capítulo se procede a describir como se encuentra actualmente el estado del arte de la tecnología utilizada para realizar este proyecto.

Los sistemas basados en localización son servicios actualmente integrados en un dispositivo móvil (junto con la o las aplicaciones que facilitan dicha información según la localización actual), operan bajo una tecnología de información geográfica, combinando el posicionamiento mediante GPS y las redes móviles. Además, esta tecnología permite ofrecer a los usuarios una personalización en lo referente a las ubicaciones, recorridos e información de los mismos.

La localización en los dispositivos móviles puede ser efectuada de forma directa, accionada por el usuario, o de forma automática, mediante los procesos en segundo plano que los móviles ya son capaces de gestionar de forma transparente. La localización es el núcleo fundamental para cualquier aplicación cuyo objetivo sea ofrecer servicios de ubicación geográfica en tiempo real.

A continuación se describen ejemplos de las aplicaciones de esta tecnología:

- Búsqueda por proximidad: ¿Dónde está el punto más cercano a la ubicación actual detectada por el dispositivo?
- Instrucciones de conducción: ¿Cómo hago para llegar a cierto lugar?
- Condiciones de tráfico: Tráfico de la ruta elegida en tiempo real o históricos de la misma
- Información sobre la latitud, longitud y dirección
- Integración con servicios de terceros, para la ubicación de puntos de interés.

- Búsqueda rápida entre varias rutas para permitir la selección de la ruta con el tiempo más corto de desplazamiento, o de menor distancia, o con otros parámetros adicionales
- Instrucciones de viaje para un origen y destino sobre un mapa
- Servicios de seguimiento suministrados por compañías de transporte publico, reparto o taxis
- Alerta de zonas: Avisos programados para zonas, establecimientos o lugares concretos

2.2. Método de geo-localización combinando redes de telefonía y satélites

GPS: Desde hace varios años los dispositivos telefónicos ya incluyen de forma habitual receptores de GPS. El GPS o Sistema de Posicionamiento Global es una red basada en una red de satélites que orbitan alrededor de la Tierra.

Nuestro dispositivo y gracias al receptor incorporado usa al menos 4 de estos satélites que estén visibles para nuestro dispositivo, cada uno de los satélites emite una señal sobre su ubicación de forma periódica. Gracias a la transformación de esa comunicación en unas latitud, longitud, altura y tiempo se calcula la ubicación, este proceso se denomina triangulación. A mayor numero de satélites visibles para el dispositivo, más exacta será esta triangulación.

GSM(redes de telefonía): Es el sistema de ámbito global utilizado para las comunicaciones móviles, o dicho de forma mas simple, es un sistema implantado a nivel global por el cual esta disponible la comunicación entre los dispositivos telefónicos móviles en general.

En muchos puntos de la superficie terrestre se encuentran ubicadas torres y/o antenas que nos brindan el servicio de telefonía, estos dispositivos de telecomunicaciones son los que permiten que nuestros teléfonos tengan cobertura y puedan realizar y recibir llamadas, actualmente también suministran la capacidad de navegación a los dispositivos móviles mediante redes como pueden ser HSPA.

La calidad de estas redes se ve influida por varios factores, la proximidad a una de las mencionadas torres o antenas, el tiempo que tarda la señal en transferirse entre las torres en funcion del destinatario de la comunicación esto establecerá el numero de saltos entre torres y por último la fuerza de la señal, pero al móvil disponer de esa información es capaz de calcular la localización de nuestro dispositivo. Este método es menos preciso en comparación a la ubicación mediante GPS, pudiendo tener un margen de error de hasta 200m.

Actualmente y con la necesidad de los dispositivos móviles de estar continuamente geo-localizados, para que un gran número de sus aplicaciones funcionen, se realiza una combinación de ambos sistemas para ofrecer una ubicación lo mas precisa posible y compensar las carencias que cada uno de los sistemas tiene por separado.

2.3. Arquitectura Android

Como para este proyecto se implementará una aplicación en el sistema operativo Android, como se indica en los requisitos no funcionales del sistema (ver tabla 4.2), en este apartado se procede a describir su estructura básica. En la figura 2.1 se muestra un esquema dividido por capas de la arquitectura Android.

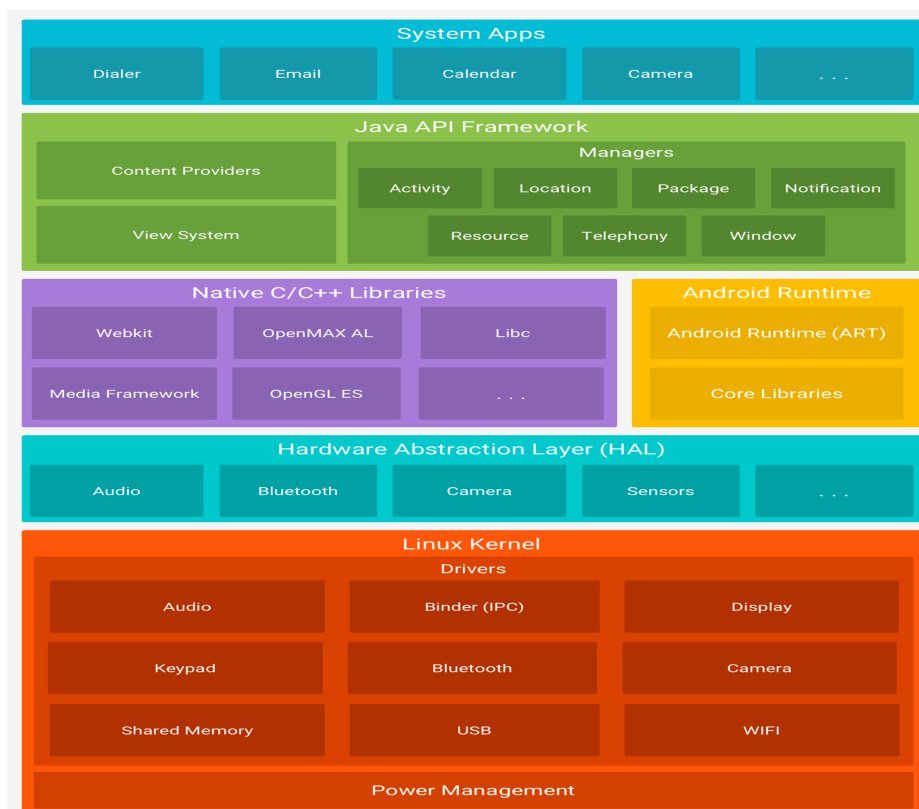


Figura 2.1: Esquema general de capas del sistema operativo Android.

Empezando desde abajo lo primero que encontramos es el Kernel de Linux: Esta capa es la base de la plataforma Android. La maquina virtual de Android (ART) se basa en el kernel de Linux para funcionalidades subyacentes, como la generación de subprocessos y la administración de memoria de bajo nivel.

Esto permite que Android aproveche funciones de seguridad claves y, al mismo tiempo, permite a los fabricantes de dispositivos desarrollar controladores de hardware para un kernel conocido.

La siguiente es la capa de abstracción de hardware (HAL), esta capa nos brinda interfaces estándares que hacen accesible las capacidades de hardware del dispositivo al framework de la Java API. La HAL esta compuesta por varios módulos de biblioteca, cada uno de estos implementa la interfaz para un tipo específico de componente hardware, Ej: cámara, bluetooth...

La tercera capa esta compartida por la maquina virtual de Android y las bibliotecas nativas de C y C++: En la maquina virtual de Android dependemos de la versión del sistema operativo instalada en

el terminal, para los dispositivos con Android 5.0 (nivel de API 21) o versiones posteriores, cada app ejecuta sus propios procesos con sus propias instancias del tiempo de ejecución de Android (ART). Entre las funciones mas significativas del ART se encuentra la compilación ahead-of-time (AOT) y just-in-time (JIT) así como la capacidad de establecer puntos de control para controlar campos específicos. Antes de Android 5.0 (nivel de API 21), Dalvik era el tiempo de ejecución del sistema operativo. Si tu app se ejecuta bien en el ART, también debe funcionar en Dalvik, pero es posible que no suceda al revés.

Por otro lado encontramos las bibliotecas C/C++ nativas las cuales son requeridas por componentes y servicios centrales del sistema Android, como el ART y la HAL, los cuales se basan en código nativo que requiere bibliotecas nativas escritas en C y C++.

En la cuarta capa esta el framework de la Java API, todo el conjunto de funciones del SO Android está disponible mediante API escritas en el lenguaje Java. Estas API son los cimientos que se necesitan para crear apps de Android simplificando la reutilización de componentes del sistema. Esto permite a los desarrolladores un acceso total a las mismas API del framework que usan las apps del sistema Android.

Por último está la capa mas cercana al usuario, esta es la capa de apps del sistema, en Android se incluye un conjunto de apps para correo electrónico, mensajería SMS, calendarios, navegación en Internet y contactos, entre otros elementos, la cuales son las básicas para el uso de los dispositivos. Estas apps iniciales pueden ser sustituidas por otras a elección del usuario, aunque existen algunas excepciones, como puede ser la app de ajustes.

Estas apps proporcionan capacidades claves a los desarrolladores para que pueden acceder desde sus propias apps. Por ejemplo, si en tu app necesita entregar un mensaje SMS.

Por lo tanto la aplicación móvil desarrollada será ubicada en la última capa, aunque también hará uso de los servicios ofrecidos por las capas inferiores.

Capítulo 3

Planificación

3.1. Introducción

En el presente capítulo se pasa a exponer los recursos disponibles, para el calendario planificado, así como los hitos a alcanzar, también se van a indicar los riesgos a tener en cuenta y los costes para el desarrollo del proyecto. Dado que se ha decidido llevar a cabo una metodología iterativa e incremental para este desarrollo (ver 1.3), la planificación se divide en varias iteraciones en las cuales se realizan todas las fases del desarrollo del proyecto.

3.2. Vista general del proyecto

El objetivo de este proyecto es desarrollar una herramienta tecnológica que permita el grabado sobre mapa de las rutas realizadas con vehículos, aunque la idea inicial es para motocicletas. Para ello se busca utilizar la API suministrada por Google Maps. Además se tratara de aplicar técnicas de gamificación para motivar su uso a los usuarios. Todo lo anteriormente comentado compondrá una aplicación móvil que se llamará Routes4Riders.

3.3. Recursos

Los recursos disponibles en el proyecto se clasifican en dos tipos:

- **Recursos humanos:** aquellas personas encargadas de realizar la investigación y el desarrollo del proyecto.
- **Recursos técnicos:** consisten en el material necesario para desarrollar las tareas asociadas al proyecto.

En cuanto a los recursos humanos presentes en el proyecto, se listan en la tabla 3.1. Se resumen en el profesor que se encarga de tutorizar este trabajo así como el alumno que lo desempeña.

Nombre	Rol en el proyecto
Joaquín Adiego Rodríguez	Tutor
Ángel Posada García	Alumno

Tabla 3.1: Recursos humanos del proyecto

Los recursos técnicos utilizados en el proyecto se listan en la tabla 3.2.

Recurso
Ordenador MSI GT72-2QD
Monitor AOC
Xiaomi MIA1
OverLeaf (web, versión gratuita)
Dropbox (version gratuita)
GitHub (version estudiante)
GitKraken (version estudiante)
Android Studio V 2.0

Tabla 3.2: Recursos técnicos del proyecto

3.4. Estimación temporal

En este apartado se procede a detallar las distintas iteraciones del proyecto así como la estimación temporal previa de las mismas. Esta estimación se realiza a nivel global, debido a que al seguir una metodología iterativa e incremental, se refinan y ajustan los requisitos en cada iteración. La citada estimación inicial está representada en la Tabla 3.3. En la figura 3.1 se muestra el diagrama de Gantt del calendario estimado como primera aproximación.

Iteración	Duración	Periodo
Estudio del estado del arte	11 días	29/01/18 - 12/02/18
Análisis de requisitos	7 días	14/02/18 - 22/02/18
Diseño de bocetos	2 días	26/02/18 - 27/02/18
Implementación del prototipo (1)	18 días	28/02/18 - 23/03/18
Pruebas del prototipo (1)	6 días	26/03/18 - 02/04/18

Análisis resultado de las pruebas (1)	3 días	03/04/18 - 05/04/18
Cambios tras resultados	6 días	06/04/18 - 13/04/18
Revisión de requisitos	3 días	16/04/18 - 18/04/18
Implementación del prototipo (2)	6 días	19/04/18 - 26/04/18
Pruebas del prototipo (2)	8 días	27/04/18 - 08/05/18
Análisis resultado de las pruebas (2)	2 días	08/05/18 - 10/05/18
Cambios finales	6 días	11/05/18 - 18/05/18

Tabla 3.3: Tabla de restricciones del sistema

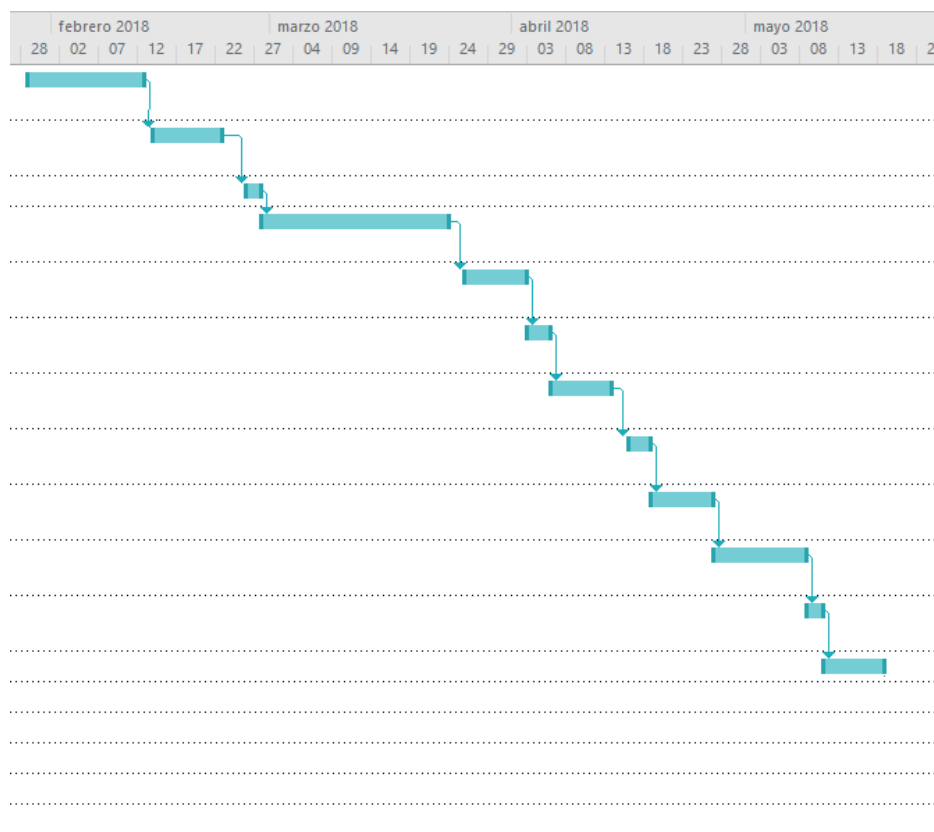


Figura 3.1: Diagrama de Gantt esperado en el proyecto

3.5. Desviaciones

En cuanto a las desviaciones temporales del proyecto no hay ningún acontecimiento destacable. No ha habido ninguna desviación significativa. Esto es en parte, debido a que las iteraciones de las que se compone la metodología seguida permitían establecer los requisitos al comienzo de cada una de ellas, unido a una gestión de riesgos adecuada. Esto permitió conseguir una versión del producto, o prototipo, que se ajustaba a los requisitos de dicha planificación. Como consecuencia, se ha conseguido un producto, que cumple con los requisitos establecidos en cada una de las iteraciones, recogidos todos ellos en el capítulo 4.

Finalmente y con el fin de evitar que se produjeran desviaciones temporales, se han dejado como trabajo para una nueva versión estable alguna de las funcionalidades propuestas en primera instancia, debido a diversos motivos, que se expone a continuación:

Compartir rutas mediante un método propio implementado en la propia aplicación, se ha sustituido para que se puedan compartir mediante la aplicación del Google Maps.

El recorrido de las rutas se realiza también mediante Google Maps, se buscaba realizar un navegador interno.

También se ha retrasado la externalización del servicio encargando del mantenimiento y devolución los datos a un servidor, se ha implementado en la propia aplicación.

Por último alguna idea que se había propuesto en las pruebas de gamificación.

Todas estas implementaciones se han pospuesto mayoritariamente por la falta de tiempo y de compaginación de la beca a jornada completa y el TFG, se indica también el cambio de domicilio debido a la beca como otro factor que ha repercutido en la toma de esta decisión.

3.6. Gestión de riesgos

Se considera un riesgo a cualquier acontecimiento ya sea presente o futuro que afecte directamente y de forma negativa a un proyecto.

Los responsables del proyecto han de ser capaces de anticiparse a los riesgos que puedan surgir así como evaluar y tener en cuenta su posible impacto, también se ha de desarrollar un plan de acción en caso de que ocurriese cualquiera de ellos

Debido a esto el plan de gestión de riesgos sera la herramienta que nos permita definir con antelación las acciones a tomar para gestionar aquellos que puedan surgir a lo largo del desarrollo del proyecto, pues tener una buena gestión de riesgos es un factor a tener en cuenta en el ahorro de costes y la actuación en situaciones de índole negativa.

El proceso de gestión de riesgos se compone de estas etapas:

- Identificación y análisis de riesgos.
- Plan de acciones y monitorización de riesgos.
- Mitigación de riesgos.

Ahora se procede a describir los posibles riesgos contemplados a lo largo del proyecto junto al plan de contingencia asociado a cada uno de ellos.

Riesgo 01	Compañía cese el soporte de una herramienta utilizada
Probabilidad	Baja, dado que es una tecnología en auge, de código libre, y que actualmente cuenta con una amplia comunidad de soporte.
Origen	Terceras partes
Impacto	Alto. Retraso de tiempo debido a implementar y adaptar el sistema a la nueva herramienta.
Reducción/Supervisión	Inspección del estado actual del soporte al comienzo y fin de cada iteración.
Plan de contingencia	Búsqueda previa de, al menos, otra herramienta similar con la misma finalidad.

Tabla 3.4: Gestión del Riesgo01

Riesgo 02	Insuficiencia de conocimientos del alumno
Probabilidad	Media, al ser un proyecto con una tecnología solo vista en una asignatura y englobar muchas posibilidades, pueden surgir problemas en ámbitos de la tecnología que el alumno desconocía que fueran necesarios para llevar a cabo el desarrollo del proyecto.
Origen	Interno.
Impacto	Medio. Retraso de tiempo debido a la búsqueda de documentación y comprensión de la misma.
Reducción/Supervisión	Búsqueda de manuales al principio de cada iteración, para evitar estancarse en medio de la misma.
Plan de contingencia	Búsqueda de manuales, webs de soporte o ayuda del propio tutor del TFG el cual también tutorizó la asignatura anteriormente comentada.

Tabla 3.5: Gestión del Riesgo02

Riesgo 03	Escasez de voluntarios para las pruebas
Probabilidad	Baja, debido a que se dispone de contactos que son el público objetivo así como el propio alumno que puede realizar las pruebas intermedias.
Origen	Externo
Impacto	Alto. Sin las pruebas no se podrán obtener resultados de usabilidad ni satisfacción con el producto, así como el correcto funcionamiento del mismo en varios ámbitos.
Reducción/Supervisión	Al ser un colectivo que disfruta realizando la actividad bastaría con pequeños incentivos
Plan de contingencia	Localizar un numero razonable de personas dispuestos a realizar las pruebas, por si al final hay alguna baja.

Tabla 3.6: Gestión del Riesgo03

Riesgo 04	Problemas en la API de Google Maps
Probabilidad	Baja, debido a que la API tiene poca carga esta alojada y mantenida por un tercero de alta fiabilidad.
Origen	Externo
Impacto	Alto. La aplicación perderá una de sus funcionalidades principales.
Reducción/Supervisión	Servicio de monitor suministrado por el propio proveedor de la API. Alarma y notificaciones en caso de fallo.
Plan de contingencia	Contactar con el proveedor para tratar de solucionar la incidencia a la mayor brevedad posible.

Tabla 3.7: Gestión del Riesgo04

Riesgo 05	Baja del alumno
Probabilidad	Media, al ser temporada de cambio de estación en pleno desarrollo del proyecto hay posibilidad de coger catarros o virus similares. También se puede dar una baja probabilidad de enfermedad grave o accidente
Origen	Externo
Impacto	Puede ser marginal o muy alto dependiendo del motivo de la baja
Reducción/Supervisión	Llevar una gestión anticipada del tiempo disponible restante y actualizarla en función de las posibles incidencias
Plan de contingencia	En caso de baja prolongada realizar las comunicaciones pertinentes al tutor así como otros posibles interesados, comunicando así la situación y tratando de ver otras fechas u opciones disponibles en función de la duración de dicha baja

Tabla 3.8: Gestión del Riesgo05

Riesgo 06	Proceso de diseño pobre.
Probabilidad	Media, el alumno no esta familiarizado ni tiene experiencia lo cual puede provocar un diseño erróneo en la primera iteración que se realice.
Origen	Interno
Impacto	Alto, supondría realizar mas iteraciones a mayores de las que se consideran inicialmente para realizar un diseño adecuado
Reducción/Supervisión	Realizar revisiones periódicas junto al tutor, así como seguir las guías y patrones ampliamente extendidos y estandarizados
Plan de contingencia	Utilizar los patrones y guías de diseño disponibles para realizar un diseño lo mas adecuado posible.

Tabla 3.9: Gestión del Riesgo06

Riesgo 07	Escasez puntual de tiempo.
Probabilidad	Alta, al estar realizando una beca a jornada completa durante el desarrollo del proyecto
Origen	Externo
Impacto	Medio inicialmente y alto en fases mas avanzadas
Reducción/Supervisión	Control de las horas disponibles y asignar mas horas los días en los que se disponga de ellas
Plan de contingencia	Estimar previamente y de forma semanal el tiempo disponible dejando una cierta holgura para contratiempos inesperados

Tabla 3.10: Gestión del Riesgo07

Riesgo 08	Fallo de alguna de las herramientas usadas.
Probabilidad	Media, al utilizarse un IDE en una versión estable y con extensiva documentación y dispositivos sin problemas previos de funcionamiento
Origen	Externo
Impacto	Alto, en caso de que sea un problema para el cual no se encuentre solución en la documentación existente, bajo en caso contrario. Alto en caso de avería de algún dispositivo
Reducción/Supervisión	Elegir bien la herramienta inicialmente, así como configurar la misma de una forma adecuada para evitar fallos por re-configuraciones durante el desarrollo del proyecto
Plan de contingencia	Revisar la documentación para la configuración inicial y tener opciones de recambio para los dispositivos

Tabla 3.11: Gestión del Riesgo08

3.7. Monitorización de los riesgos

En esta sección se procede a detallar los efectos de los riesgos enumerados anteriormente y que se han producido durante el desarrollo del proyecto. Tras cada iteración se llevaba a cabo una revisión de los riesgos anteriormente citados en el proyecto y en qué medida se han producido en la iteración finalizada o podrían producirse para la siguiente, además de tener en cuenta las consecuencias conllevaban o conllevarían.

El Riesgo 1 no se produjo durante todo el desarrollo del proyecto pues la versión de Android que se utiliza no dejó de recibir soporte y tampoco se planea que ocurra a corto plazo.

El Riesgo 02 se produjo de forma mas acusada en los primeros meses, debido a que se tuvo que realizar un gran trabajo inicial para la parte de la memoria asociada al proyecto en sus diferentes apartados, estado del arte, la tecnología existente relacionada.

Del Riesgo 03 se indica que no se produce en el proyecto debido a que se obtuvieron recomendaciones por parte del tutor del proyecto, así como de algún usuario de los que probaron la aplicación en alguna de sus versiones preliminares, estos comentarios sobre aspectos de la interfaz y las sugerencias de cambios para mejorarla ayudaron a confeccionar la interfaz final.

El Riesgo 04 no se ha contemplado en el proyecto, pues la API cuenta actualmente con soporte y no se prevé que esto deje de ser así a corto plazo.

En referencia al Riesgo 05, tampoco se ha producido durante el desarrollo del proyecto.

Para el Riesgo 06 se contó con un diseño previo realizado en la asignatura de sistemas móviles lo cual aportó una mejor calidad en el proceso de diseño.

En lo referente al Riesgo 7 se produjo de forma intermitente durante todas las iteraciones del proyec-

Planificación

to, esto fue debido al desplazamiento del lugar de residencia habitual del alumno, así como a la beca que se estaba realizando de forma paralela al desarrollo del proyecto. Esto supuso realizar reajustes de objetivos del plan semanal anteriormente comentado, tratando de ajustarse lo mas posible a los objetivos previamente establecidos, aunque finalmente si afectó a la planificación y a los objetivos de alcance marcados inicialmente.

Por último el Riesgo 8 no se produjo ni en los dispositivos ni con el entorno de desarrollo seleccionado.

Capítulo 4

Análisis

4.1. Introducción

En este capítulo se procede a la definición de todos los requisitos del sistema que se ha propuesto. Además se van a presentar las restricciones del proyecto así como la descripción y modelización de los casos de uso, junto con el análisis detallado de los mismos. También se incluye el diagrama del modelo de dominio del sistema.

4.2. Requisitos

En esta sección se procede a enumerar y describir los requisitos funcionales y los no funcionales del proyecto, debido a su importancia en el proceso de desarrollo del mismo, mas concretamente del producto software, la aplicación Routes4Riders, también se realizara la enumeración y descripción de los requisitos de información, en estos último se describirá las condiciones necesarias para la aplicación así como los requisitos de los dispositivos objetivo de la misma. También incluye las condiciones de almacenamiento de los datos, formato, tecnología de base de datos.

4.2.1. Requisitos funcionales

En la tabla 4.1 se muestra un listado de los requisitos funcionales del sistema.

ID	Ver.	Nombre	Descripción	Prioridad
RF01	1.0	Almacenar datos usuario	El sistema obtendrá los datos del usuario introducidos por el mismo en el sistema	Alta

RF02	1.0	Visualizar datos usuario	El sistema debe mostrar los datos almacenados del usuario	Alta
RF03	1.0	Selección de vehículo	El sistema debe mostrar los vehículos almacenados y guardar la selección del usuario	Media
RF04	1.0	Modificar datos usuario	El sistema debe permitir modificar los datos del usuario y almacenar los cambios	Media
RF05	1.0	Mostrar ubicación en tiempo real	El sistema debe actualizar la posición sobre el mapa de forma automática y en tiempo real	Alta
RF06	1.0	Grabación de rutas en tiempo real	El sistema debe grabar el recorrido realizado por el usuario cuando sea requerido	Alta
RF07	1.0	Acceso a las rutas grabadas	El sistema debe mostrar el listado de las rutas grabadas en el dispositivo	Alta
RF08	1.0	Visualizar las rutas grabadas	El sistema debe mostrar sobre el mapa los recorridos grabados	Alta
RF09	1.0	Eliminar las rutas grabadas	El sistema debe permitir eliminar las rutas almacenadas	Media

RF10	1.0	Indicaciones hasta inicio de ruta	El sistema debe transferir el punto de inicio para que se den las indicaciones hasta el mismo	Media
RF11	1.0	Indicaciones en ruta	El sistema debe transferir los puntos de la ruta para que se den las indicaciones del trayecto	Media
RF12	1.0	Visualizar datos vehículo	El sistema debe mostrar los datos del vehículo	Alta
RF13	1.0	Eliminar datos vehículo	El sistema debe permitir eliminar los datos del vehículo guardado	Media
RF14	1.0	Modificar datos vehículo	El sistema debe permitir modificar los datos del vehículo guardado	Media

Tabla 4.1: Tabla de requisitos funcionales

4.2.2. Requisitos no funcionales

En la tabla 4.2 se muestra un listado de los requisitos funcionales del sistema.

ID	Ver.	Nombre	Descripción	Importancia
1.Propiedades del sistema				
1.1.Facilidad de uso				
a)Instalación y administración				
RNF01	1.0	Instalación del sistema	El sistema dispondrá su instalación y realización los pasos iniciales de la forma mas sencilla posible para que el propio usuario pueda realizarlo.	Critica

RNF02	1.0	Actualización del sistema	El sistema facilitará la actualización del mismo, de forma que los usuarios del sistema cliente sean capaces de realizarla.	Deseable
b) Interacción con el Usuario				
RNF03	1.0	Confirmación anterior acción importante	El sistema solicitará confirmación al Usuario al realizar todas las tareas de mayor relevancia.	Deseable
RNF04	1.0	Avisos de utilización errónea	El sistema informará de errores en el uso o acciones no permitidas realizadas en la aplicación.	Deseable
1.2. Confiabilidad				
RNF05	1.0	Realización de copias de seguridad	El sistema realizará copias de respaldo semanales de la información de su base de datos.	Deseable
RNF06	1.0	Notificación de errores	El sistema proporcionará notificaciones de forma interna errores en el sistema mediante un registro de errores.	Crítica
1.3. Rendimiento				
a) Capacidad				
RNF07	1.0	Capacidad de la base de datos	La base de datos del sistema debería ser capaz de almacenar y tratar correctamente la información relativa al usuario, vehículo .	Deseable
RNF08	1.0	Escalabilidad de la base de datos	La base de datos del sistema debería ser capaz de aumentar el número de vehículos disponibles de manera sistemática.	Deseable
b) Tiempo de respuesta				

RNF09	1.0	Tiempo de transiciones	El sistema debería ser capaz de realizar las transiciones de las actividades en condiciones normales, en menos de dos segundos.	Crítica
RNF10	1.0	Tiempo de carga mapa	El sistema debería ser capaz de realizar la carga del mapa de referencia en condiciones normales, en menos de cuatro segundos.	Crítica
c) Reinicio				
RNF11	1.0	Autoreinicio del sistema servidor	El sistema servidor será capaz de reiniciarse, en caso de que un error del propio sistema lo requiera, en menos de tres minutos.	Crítica
1.4. Soporte				
a) Plataforma				
RNF12	1.0	Acceso al sistema cliente	El sistema cliente será accesible mediante un dispositivo con Android 4.1. o superior.	Crítica
RNF13	1.0	Plataforma Google	El sistema utilizará los servicios proporcionados por Google para mostrar los mapas, así como para el sistema de navegación e indicaciones.	Crítica
b) Organización				

RNF14	1.0	Organización del sistema	La sistema tendrá una configuración en tres niveles: I. La lógica de presentación y aplicación recaerá sobre un cliente grueso (aplicación móvil). II. La lógica de acceso al sistema y recopilación de estadísticas de uso recaerá sobre un servidor web de terceros (Google APIs). III. La lógica de almacenamiento de datos en logs recaerá sobre un servidor web de terceros.	Crítica
c) Localización				
RNF15	1.0	Idiomas de la interfaz	El sistema soportará como mínimo la localización en los idiomas inglés y castellano.	Crítica
RNF16	1.0	Codificación del texto	El sistema soportará como mínimo la codificación UTF-8.	Crítica
1.5. Seguridad				
RNF17	1.0	Conexiones cliente servidor	El sistema debería utilizar el estándar de Internet TLS/SSL definido por la IETF para todas sus comunicaciones cliente servidor.	Crítica
RNF18	1.0	Acceso a los logs	El acceso a los logs del sistema servidor sólo podrá llevarse a cabo por los administradores del mismo.	Crítica
2. Interfaces del sistema				
2.1. Accesibilidad, diseño y navegación				
RNF19	1.0	Diseño de interfaz adaptativo	La interfaz gráfica cumplirá con un diseño adaptativo para su visualización en cualquier dispositivo móvil.	Crítica

RNF20	1.0	Accesibilidad	El sistema debería cumplir los requisitos mínimos de accesibilidad para Android.	Crítica
2.2. Interfaces con sistemas externos o dispositivos				
RNF21	1.0	Acceso a Google Maps	El sistema se comunicará con la plataforma de Google Maps a través de su API.	Crítica
3. Documentación del sistema				
RNF21	1.0	Acceso al manual de instalación	El sistema debería contar con un manual de instalación.	Crítica
RNF21	1.0	Acceso al manual de usuario	El sistema debería contar con un manual de usuario.	Crítica

Tabla 4.2: Tabla de requisitos no funcionales

4.2.3. Requisitos de información

En la tabla 4.3 se muestran los requisitos de información del sistema.

ID	Ver.	Nombre	Descripción
RIN01	1.0	Acceso datos usuario	El sistema accederá a los datos del usuario: nombre, correo, marca y modelo de vehículo
RIN02	1.0	Composición de una ruta	Una ruta se compondrá de puntos almacenados y representados mediante el objeto polyline
RIN03	1.0	Composición de los puntos	Cada punto contara de los datos de latitud, longitud y ruta a la que pertenecen
RIN04	1.0	Almacenar datos	Para el almacenamiento de los datos se usara una base de datos SQLite
RIN05	1.0	Composición de la moto	Cada moto contara con un id, marca y modelo, se cargaran inicialmente desde un fichero SQL
RIN06	1.0	Composición datos usuario	El usuario tendrá un ID, nombre de usuario, correo electrónico, así como la marca y modelo de su vehículo

Tabla 4.3: Tabla de requisitos de información

4.3. Restricciones

En la tabla 4.4 se listan las restricciones del sistema, impuestas por la tecnología y los recursos disponibles.

ID	Ver.	Nombre	Descripción	Prioridad
RES01	1.0	Necesidad de permisos de localización	El dispositivo móvil en el que se ejecute la aplicación necesitara dar a la misma permisos de ubicación	Alta
RES02	1.0	Necesidad de permisos de acceso a internet móvil	El dispositivo móvil en el que se ejecute la aplicación necesitara dar a la misma permisos de acceso a internet móvil	Alta
RES03	1.0	Necesidad de permisos de localización	El dispositivo móvil en el que se ejecute la aplicación necesitara dar a la misma permisos de ubicación	Alta
RES04	1.0	Necesidad de Google Play Services en el dispositivo	El dispositivo móvil en el que se ejecute la aplicación necesitara la última versión disponible de la aplicación de Google Play Services instalada	Alta
RES05	1.0	Necesidad de Google Maps en el dispositivo	El dispositivo móvil en el que se ejecute la aplicación necesitara la última versión disponible de la aplicación de Google Maps	Alta
RES06	1.0	Necesidad de receptor GPS en el dispositivo	El dispositivo móvil en el que se ejecute la aplicación necesitara disponer de receptores señal GPS	Alta
RES06	1.0	Necesidad de permisos acceso ubicación redes móviles	El dispositivo móvil en el que se ejecute la aplicación necesitara disponer de permisos para obtener la ubicación mediante las redes móviles	Alta

Tabla 4.4: Tabla de restricciones del sistema

4.4. Modelo de casos de uso

En las secciones que se muestran a continuación se pasa a detallar los actores, casos de uso y diagramas de actividad del sistema.

4.4.1. Actores primarios

- Usuario: Rol genérico que representara a cualquier persona que utilice la aplicación en su dispositivo móvil.

4.4.2. Actores secundarios

- Google Maps: Sistema externo que se encarga de proveer la navegación e indicaciones.
- Google Maps API: Sistema externo que se encargara de proporcionar el mapa así como la monitorización continua del desplazamiento, para poder grabar el mismo. También será el encargado de reflejarlo en el mapa a modo de vista previa.

4.4.3. Diagrama de casos de uso

En la figura 4.1 se puede ver el diagrama de casos de uso del sistema. Los casos de uso indican las comunicaciones que realizan los actores con el sistema y permite ver de forma mas completa la funcionalidad de la aplicación anteriormente indicada en los requisitos funcionales.



Figura 4.1: Diagrama de casos de uso

4.4.4. Matriz de requisitos frente a casos de uso

En la tabla 4.5 se representa la dependencia existente entre los casos de uso y los requisitos funcionales.

Requisitos	Casos de uso										
	CU01: Registrarse	CU02: Ver datos moto	CU03: Ver datos usuario	CU04: Modificar datos usuario	CU05: Modificar datos moto	CU06: Iniciar Ruta	CU07: Finalizar ruta	CU08: Ver rutas realizadas	CU09: Mostrar en mapa	CU10: Realizar ruta	CU11: Ir a inicio
RF01	X										
RF02			X								
RF03		X									
RF04				X							
RF05						X	X				
RF06						X	X				
RF07								X			
RF08									X		
RF09								X			
RF10											X
RF11										X	
RF12		X									
RF13					X						
RF14					X						

Tabla 4.5: Matriz de requisitos y casos de uso

4.4.5. Especificación de casos de uso de análisis

En este apartado se describen los once casos de uso del sistema que se han enunciado en el apartado 4.4.3.

Caso de uso	CU01: Registrarse
Versión	1.0
Dependencias	RF01
Actor principal	Usuario
Actor secundario	Ninguno
Descripción	Un usuario desea registrarse en el sistema.
Precondiciones	1. El usuario no esta previamente registrado.
Flujo básico	1. El caso de uso se inicia cuando el usuario inicia la aplicación por primera vez. 2. El sistema muestra el formulario para la recogida del correo y contraseña. 3. El usuario introduce los datos y pulsa el botón entrar. 4. El sistema muestra los datos en la siguiente pantalla junto a otros campos a rellenar. 5. El usuario selecciona nueva moto. 6. El caso de uso continua en el CU02. 7. El sistema muestra los campos con los datos del usuario y vehículo. 8. El usuario confirma la información. 9. El sistema almacena la información proporcionada. 10. El caso de uso finaliza.
Postcondiciones	1. Los datos del usuario son almacenados en la base de datos
Flujo alternativo	FA01. El usuario pulsa cancelar en la selección de moto. 1. El caso de uso continua en el paso 4 del flujo básico.
Excepciones	EX01. Datos erróneos. 1. Se muestra un mensaje de aviso al usuario. 2. Se vuelve al paso 2 o 4 del flujo básico.
Frecuencia	Baja

Tabla 4.6: Descripción del CU01

Caso de uso	CU02: Ver datos moto
Versión	1.0
Dependencias	RF03
Actor principal	Usuario
Actor secundario	Ninguno
Descripción	Un usuario desea ver datos de las motos disponibles.
Precondiciones	1. El usuario esta previamente registrado.
Flujo básico	1. El caso de uso se inicia cuando el usuario pulsa el botón de nueva moto. 2. El sistema muestra dos campos con marca y modelo por defecto 3. El usuario selecciona una marca. 4. El sistema muestra los modelos de esa marca. 5. El caso de uso finaliza.
Postcondiciones	Ninguna
Flujo alternativo	FA01. El usuario pulsa cancelar en la selección de moto. 1. El caso de uso finaliza.
Excepciones	Ninguna
Frecuencia	Media

Tabla 4.7: Descripción del CU02

Caso de uso	CU03: Ver datos usuario
Versión	1.0
Dependencias	RF02
Actor principal	Usuario
Actor secundario	Ninguno
Descripción	Un usuario desea visualizar sus datos en el sistema.
Precondiciones	1. El usuario esta previamente registrado.
Flujo básico	1. El caso de uso se inicia cuando el usuario pulsa el botón de menú. 2. El sistema muestra los datos asociados al usuario. 3. El caso de uso finaliza.
Postcondiciones	Ninguna
Flujo alternativo	Ninguno
Excepciones	Ninguna
Frecuencia	Alta

Tabla 4.8: Descripción del CU03

Caso de uso	CU04: Modificar datos usuario
Versión	1.0
Dependencias	RF04
Actor principal	Usuario
Actor secundario	Ninguno
Descripción	Un usuario desea modificar sus datos en el sistema.
Precondiciones	1. El usuario esta previamente registrado.
Flujo básico	1. El caso de uso se inicia cuando el usuario pulsa el botón de confirmar datos. 2. El sistema recoge la información de los campos de información de usuario. 3. El sistema almacena los datos actualizados. 4. El caso de uso finaliza.
Postcondiciones	1. Los datos del usuario son almacenados en la base de datos
Flujo alternativo	FA01. El usuario pulsa cancelar en los datos personales. 1. El caso de uso finaliza.
Excepciones	EX01. Datos erróneos. 1. Se muestra un mensaje de aviso al usuario. 2. Se vuelve al paso 2 del flujo básico.
Frecuencia	Baja

Tabla 4.9: Descripción del CU04

Caso de uso	CU05: Modificar datos moto
Versión	1.0
Dependencias	RF013 y RF14
Actor principal	Usuario
Actor secundario	Ninguno
Descripción	Un usuario desea modificar la información de su vehículo en el sistema.
Precondiciones	1. El usuario esta previamente registrado.
Flujo básico	1. El caso de uso se inicia cuando el usuario pulsa el botón nueva moto. 2. El sistema muestra dos campos con marca y modelo por defecto 3. El usuario selecciona una marca. 4. El sistema muestra los modelos de esa marca. 5. El usuario selecciona el modelo. 6. El usuario confirma los datos. 7. El sistema almacena la información de la nueva moto. 8. El caso de uso finaliza.
Postcondiciones	1. Los datos del vehículo son almacenados en la base de datos
Flujo alternativo	FA01. El usuario pulsa cancelar en la selección de moto. 1. El caso de uso finaliza.
Excepciones	Ninguna
Frecuencia	Baja

Tabla 4.10: Descripción del CU05

Caso de uso	CU06: Iniciar ruta
Versión	1.0
Dependencias	RF05 y RF06
Actor principal	Usuario
Actor secundario	Google Maps API
Descripción	Un usuario desea comenzar grabación de una ruta.
Precondiciones	1. El usuario esta previamente registrado.
Flujo básico	1. El caso de uso se inicia cuando el usuario pulsa el botón iniciar ruta. 2. El sistema crea el id de una nueva ruta y lo almacena. 3. El sistema inicia la obtención de las coordenadas en tiempo real. 4. El sistema crea un id para cada punto y los almacena asociados al id de ruta. 5. El sistema realiza esto de forma automática hasta que usuario inicie el (CU07).
Postcondiciones	Ninguna
Flujo alternativo	Ninguno
Excepciones	Ninguna
Frecuencia	Alta

Tabla 4.11: Descripción del CU06

Caso de uso	CU07: Finalizar ruta
Versión	1.0
Dependencias	RF05 y RF06
Actor principal	Usuario
Actor secundario	Google Maps API
Descripción	Un usuario desea finalizar grabación de una ruta.
Precondiciones	1. El usuario esta previamente registrado y ha iniciado una ruta.
Flujo básico	1. El caso de uso se inicia cuando el usuario pulsa el botón terminar ruta. 2. El sistema almacena el último punto grabado. 3. El sistema finaliza el servicio automático. 4. El caso de uso finaliza.
Postcondiciones	1. La grabación de la ruta se detiene 2. La ruta se guarda correctamente en la base de datos
Flujo alternativo	Ninguno
Excepciones	Ninguna
Frecuencia	Alta

Tabla 4.12: Descripción del CU07

Caso de uso	CU08: Ver rutas realizadas
Versión	1.0
Dependencias	RF07 y RF09
Actor principal	Usuario
Actor secundario	Google Maps API
Descripción	Un usuario desea ver las rutas almacenadas.
Precondiciones	1. El usuario esta previamente registrado y ha grabado una ruta previamente.
Flujo básico	1. El caso de uso se inicia cuando el usuario pulsa el botón mostrar rutas. 2. El sistema muestra el listado de rutas grabadas. 3. El caso de uso finaliza.
Postcondiciones	Ninguna
Flujo alternativo	Ninguno
Excepciones	Ninguna
Frecuencia	Alta

Tabla 4.13: Descripción del CU08

Caso de uso	CU09: Mostrar en mapa
Versión	1.0
Dependencias	RF08
Actor principal	Usuario
Actor secundario	Google Maps API
Descripción	Un usuario desea ver una ruta en el mapa.
Precondiciones	1. El usuario esta previamente registrado y ha grabado una ruta previamente.
Flujo básico	1. El caso de uso se inicia cuando el usuario pulsa el botón mostrar rutas. 2. El sistema muestra el listado de rutas grabadas. 3. El usuario selecciona una de las rutas almacenadas. 4. El sistema carga en el mapa la ruta trazada con líneas sobre el mapa. 5. El caso de uso finaliza.
Postcondiciones	Ninguna
Flujo alternativo	Ninguno
Excepciones	Ninguna
Frecuencia	Alta

Tabla 4.14: Descripción del CU09

Caso de uso	CU10: Realizar ruta
Versión	1.0
Dependencias	RF11
Actor principal	Usuario
Actor secundario	Google Maps API y Google Maps
Descripción	Un usuario desea realizar una ruta grabada.
Precondiciones	1. El usuario esta previamente registrado y ha ha grabado una ruta previamente.
Flujo básico	1. El caso de uso se inicia cuando el usuario pulsa el botón realizar ruta. 2. El sistema envía los datos a la aplicación de Google Maps. 3. El caso de uso finaliza.
Postcondiciones	Ninguna
Flujo alternativo	Ninguno
Excepciones	Ninguna
Frecuencia	Alta

Tabla 4.15: Descripción del CU10

Caso de uso	CU11: Ir a inicio
Versión	1.0
Dependencias	RF10
Actor principal	Usuario
Actor secundario	Google Maps API y Google Maps
Descripción	Un usuario desea ir al inicio de una ruta.
Precondiciones	1. El usuario esta previamente registrado y ha grabado una ruta previamente.
Flujo básico	1. El caso de uso se inicia cuando el usuario pulsa el botón ir a inicio. 2. El sistema envía los datos a la aplicación de Google Maps. 3. El caso de uso finaliza.
Postcondiciones	Ninguna
Flujo alternativo	Ninguno
Excepciones	Ninguna
Frecuencia	Alta

Tabla 4.16: Descripción del CU11

4.5. Realización de casos de uso de análisis

4.5.1. Modelo de dominio

En la figura 4.2 se muestra el modelo de dominio en el que se ha basado el sistema. En el se muestran las clases mas significativas que se han obtenido de los requisitos y de los casos de uso.

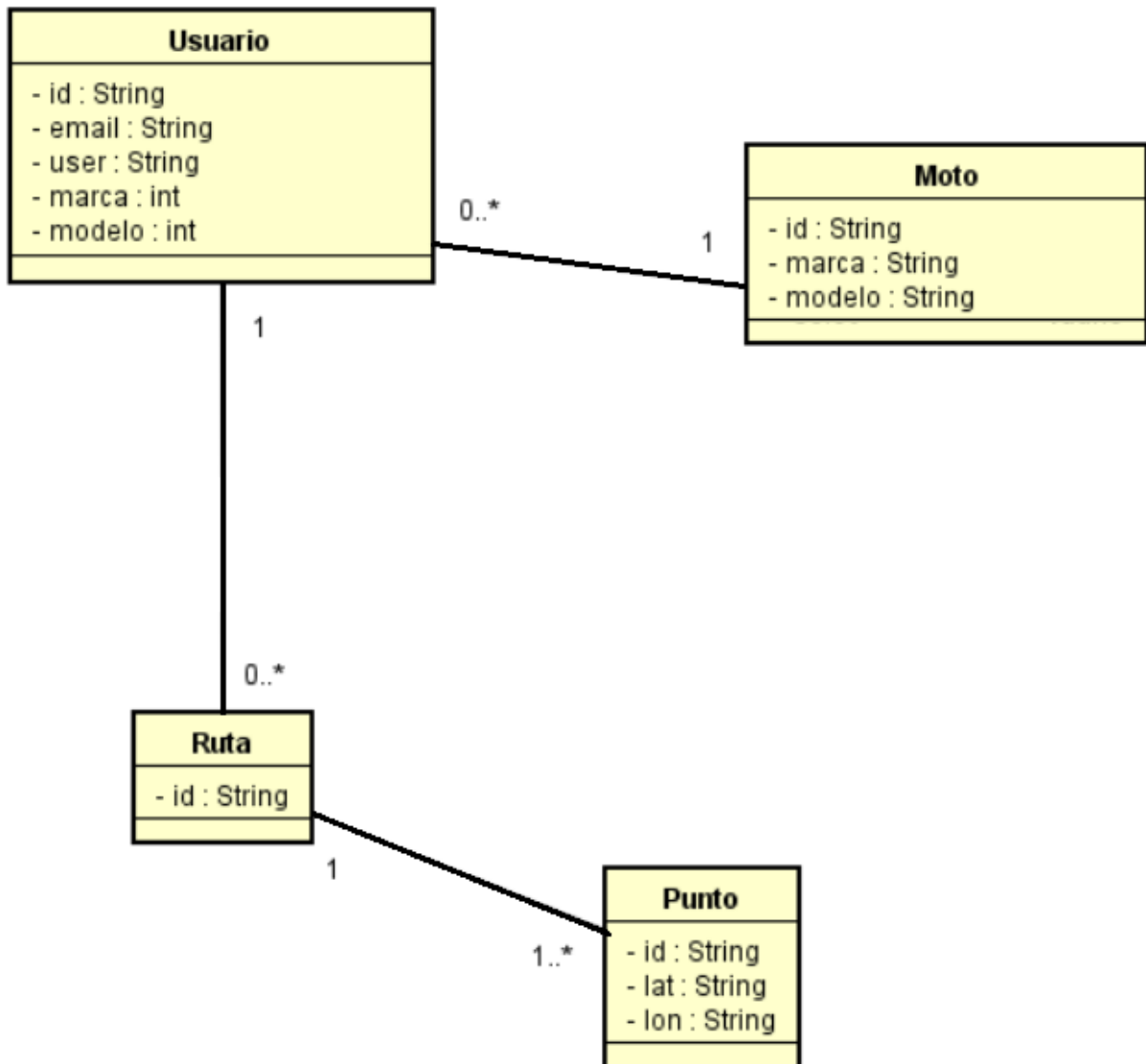


Figura 4.2: Diagrama de clases

A continuación se procede a describir los elementos que se muestran en la figura 4.2.

Clase: Usuario

- **Descripción:** Clase que modela a un usuario de la aplicación.
- **Responsabilidades:** Identifica al usuario dentro de la aplicación, mostrar y modificar sus datos.
- **Atributos:**
 - id: Identificador del usuario.
 - email: Correo de registro del usuario.
 - user: Nombre o alias del usuario.
 - marca: marca de la moto asociada al usuario.
 - modelo: modelo de la moto asociada al usuario.

Clase: Moto

- **Descripción:** Clase que modela a una moto de la aplicación.
- **Responsabilidades:** Identifica a cada moto dentro de la aplicación, mostrar la información de las mismas.
- **Atributos:**
 - id: Identificador de la moto.
 - marca: Atributo marca de la moto.
 - modelo: Atributo modelo de la moto, dependiente de la marca.

Clase: Ruta

- **Descripción:** Clase que modela a una ruta de la aplicación.
- **Responsabilidades:** Identificar unívocamente, almacenar y mostrar las rutas en la aplicación.
- **Atributos:**
 - id: Identificador de la ruta.

Clase: Punto

- **Descripción:** Clase que modela los puntos que componen una ruta.

- **Responsabilidades:** Identificar mediante coordenadas puntos en el mapa por los que pasa el usuario de forma automática, asociarlos a una única ruta.
- **Atributos:**
 - id: Identificador del punto.
 - lat: Latitud de la ubicación en la que se genera el punto.
 - lon: Longitud de la ubicación en la que se genera el punto.

4.5.2. Diagrama Entidad Relación

En este apartado se puede apreciar en la figura 4.3 el diagrama entidad relación correspondiente a la base de datos de la aplicación.

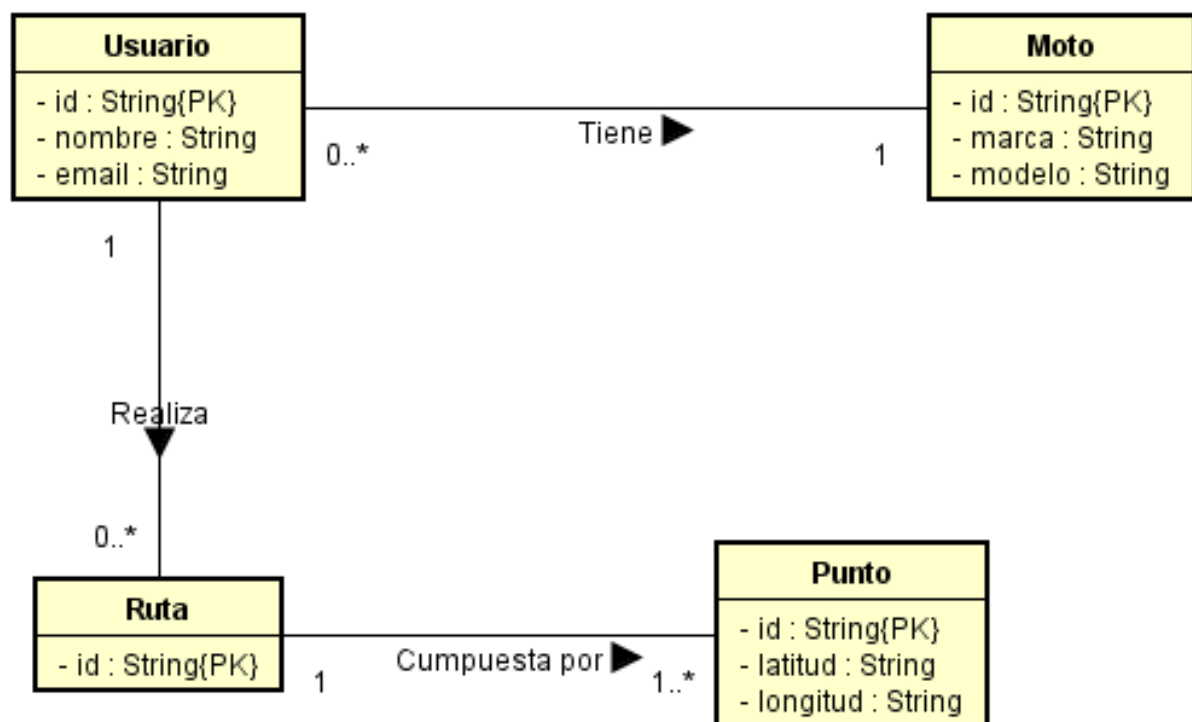


Figura 4.3: Diagrama Entidad Relación

4.5.3. Diagramas de actividad

En el siguiente apartado se muestra el flujo de eventos entre las clases del dominio que se realiza en los once casos de uso mediante los diagramas de actividad de los mismos.

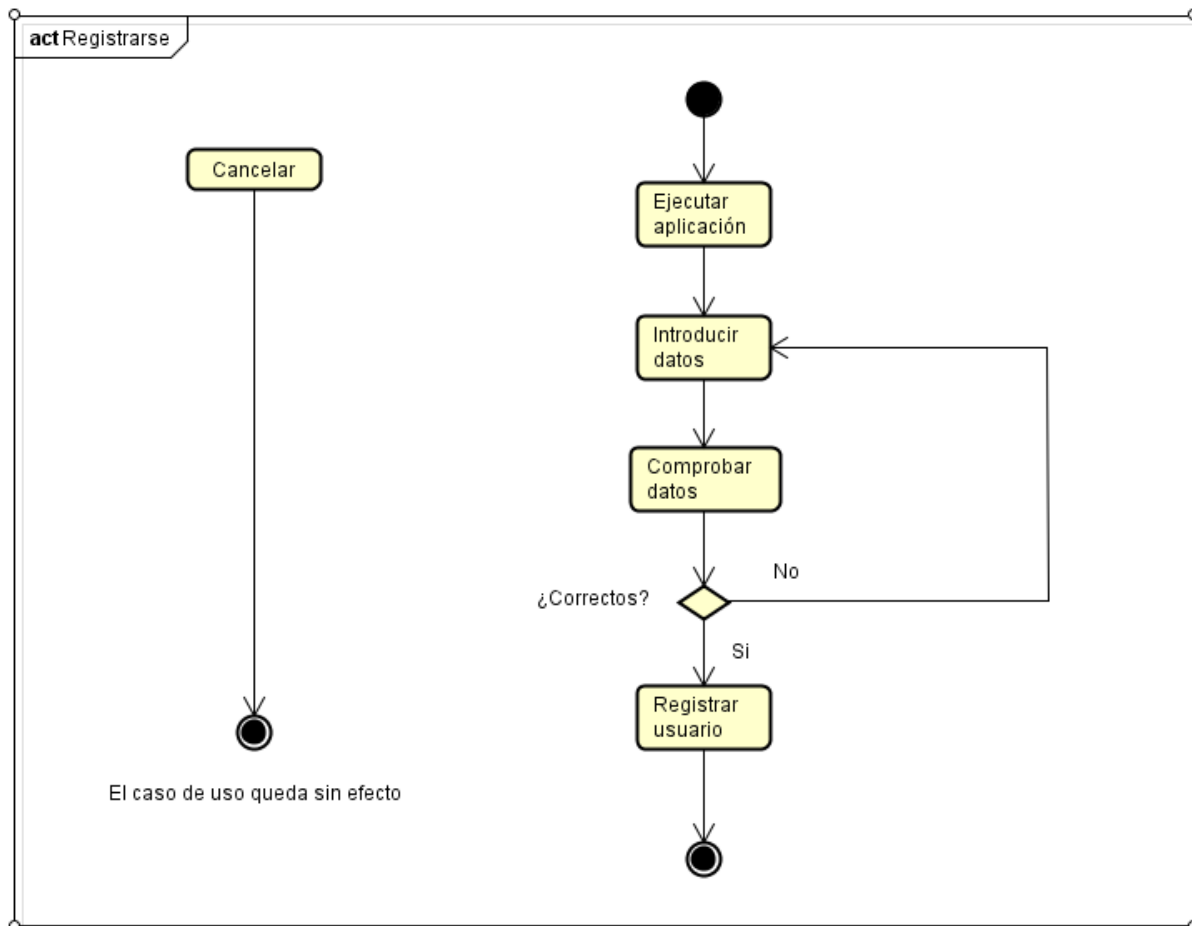


Figura 4.4: Diagrama de actividad del CU01

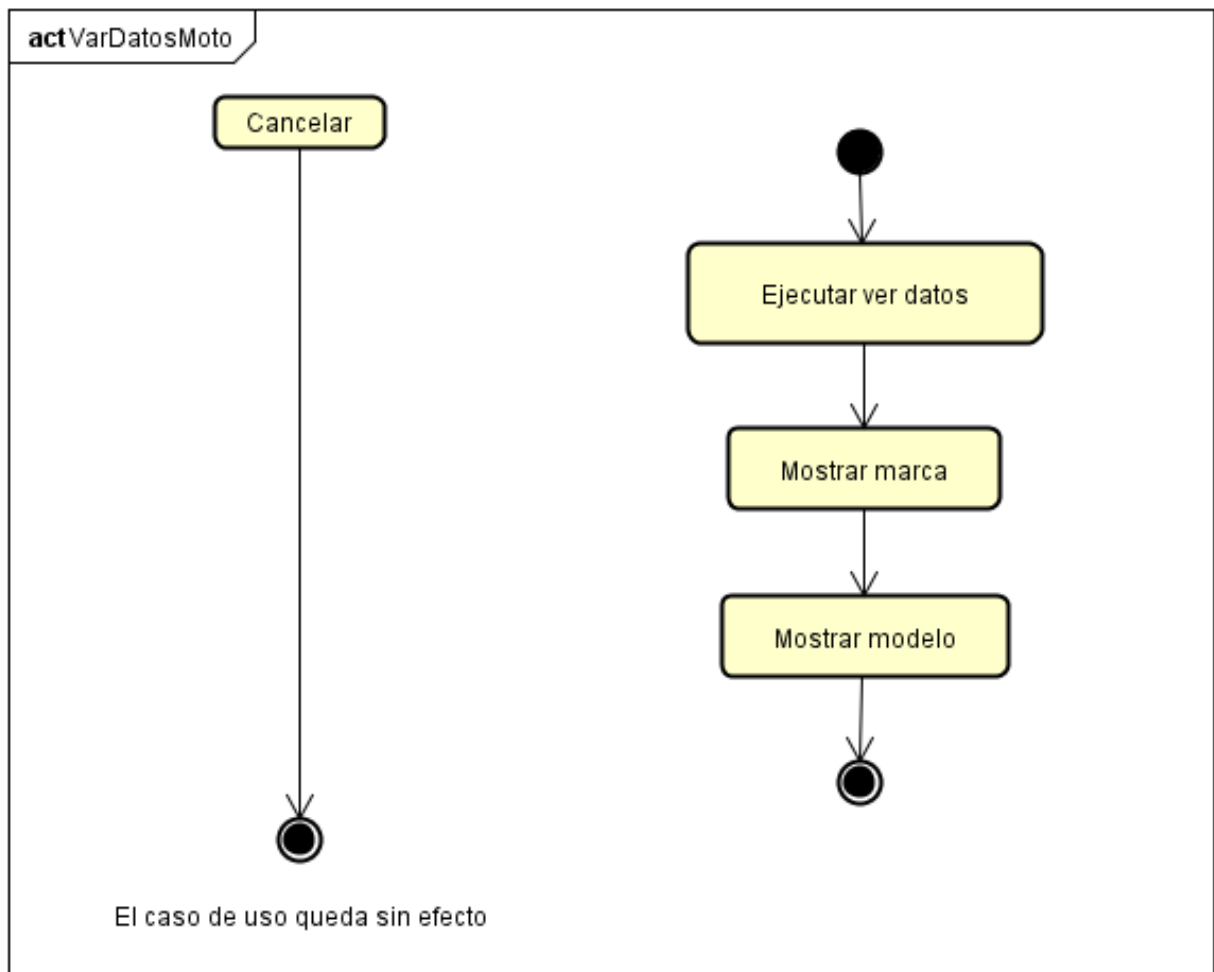


Figura 4.5: Diagrama de actividad del CU02

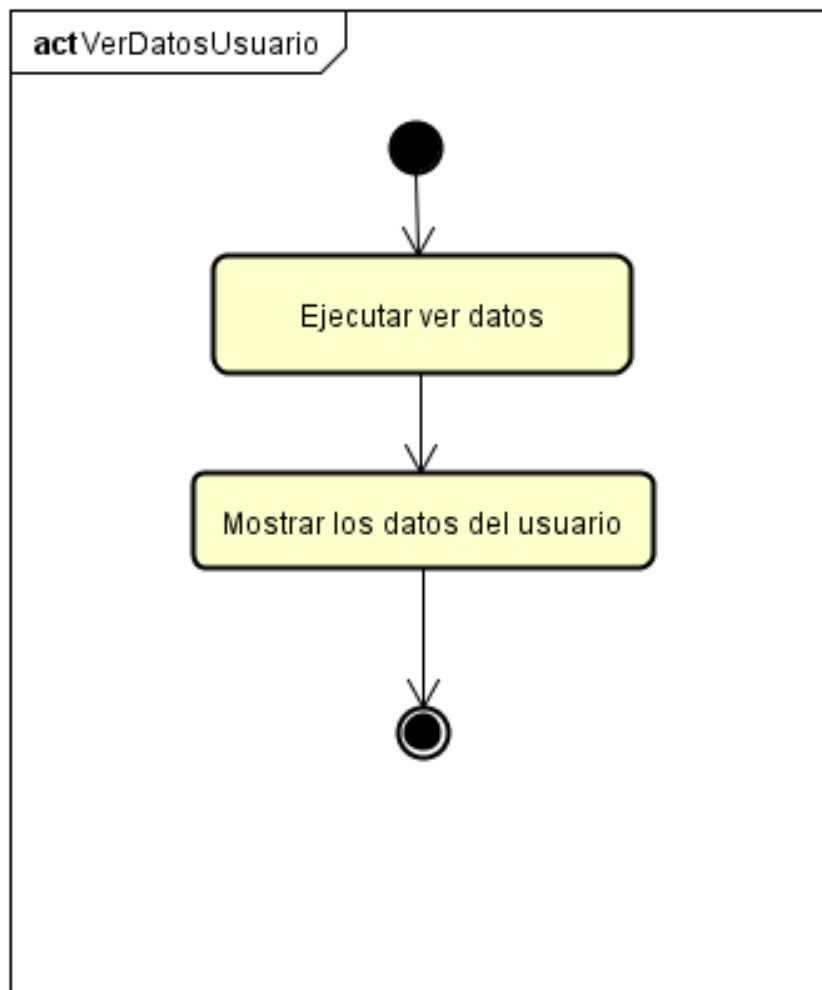


Figura 4.6: Diagrama de actividad del CU03

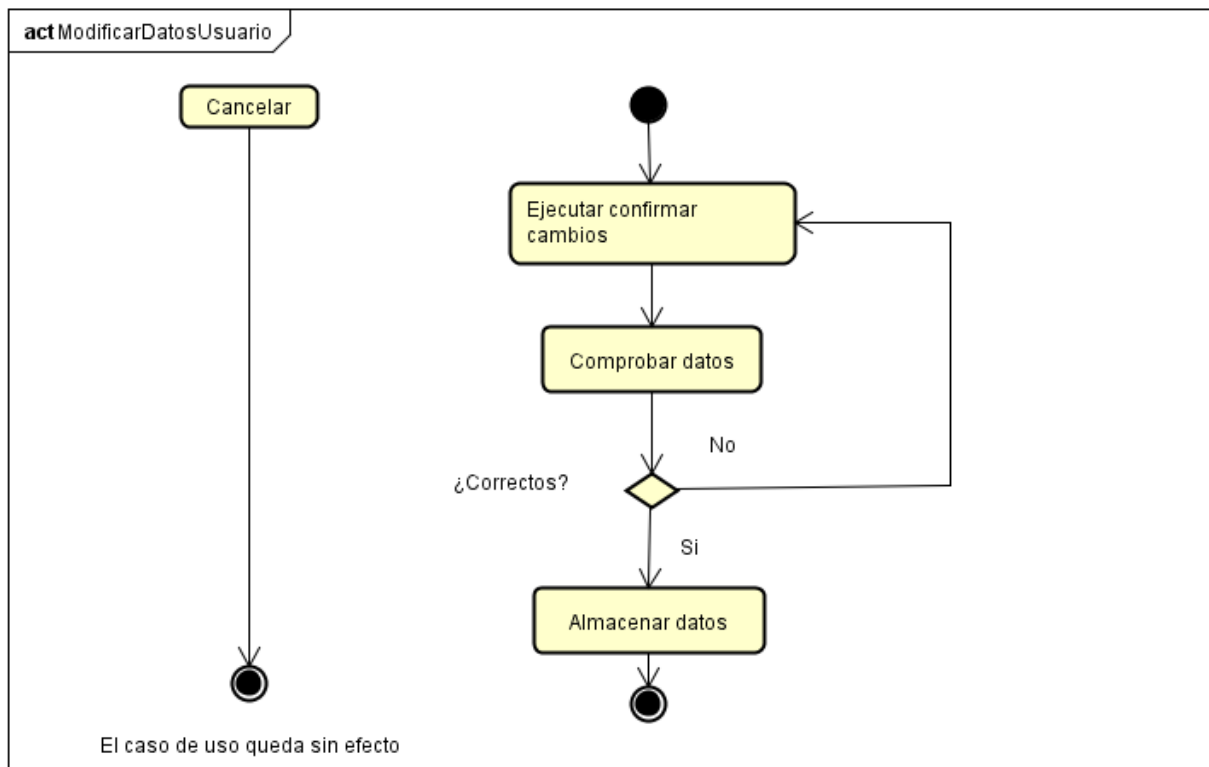


Figura 4.7: Diagrama de actividad del CU04

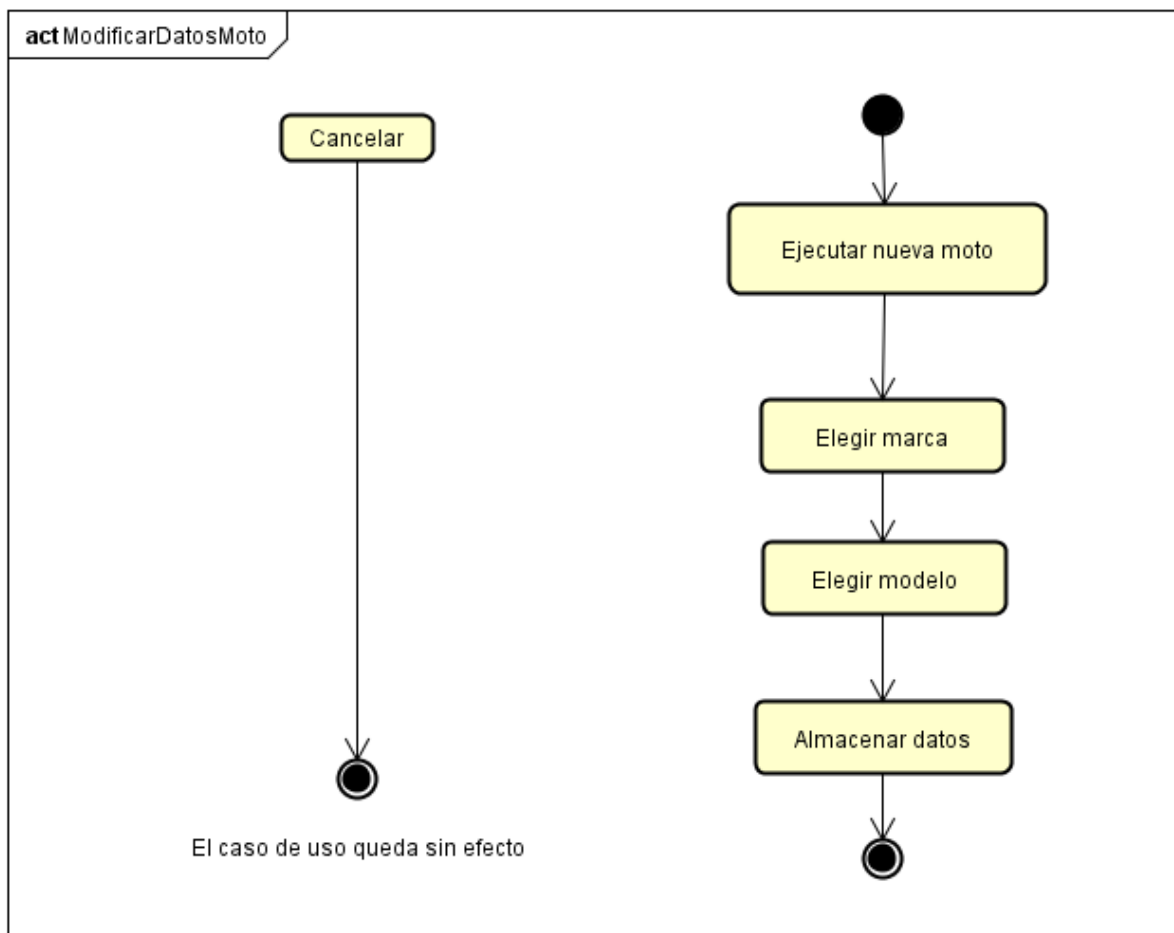


Figura 4.8: Diagrama de actividad del CU05

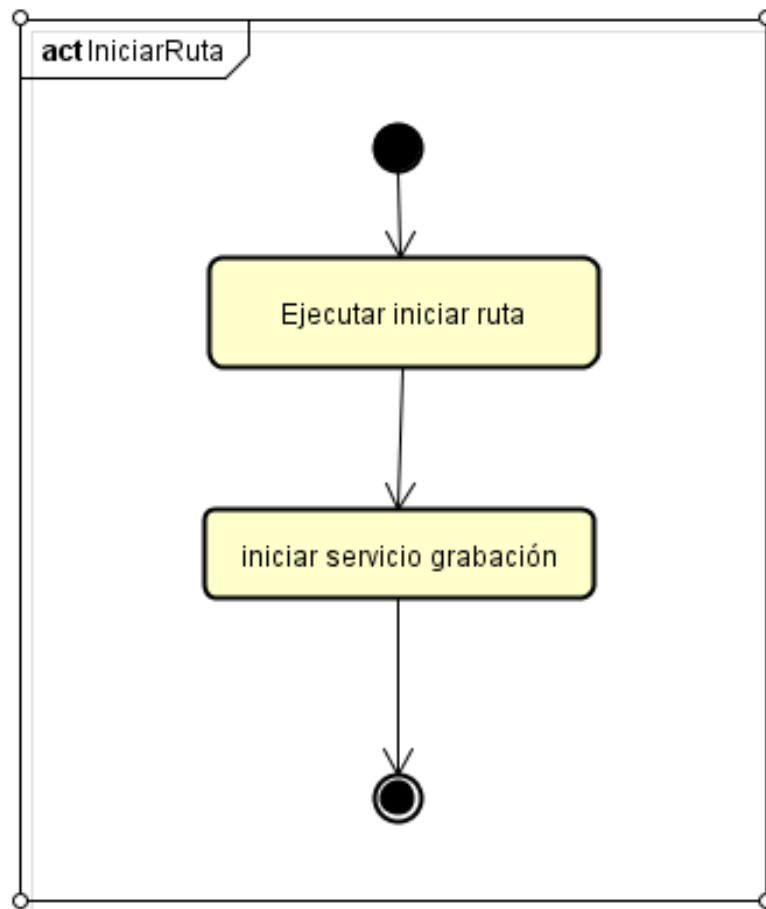


Figura 4.9: Diagrama de actividad del CU06

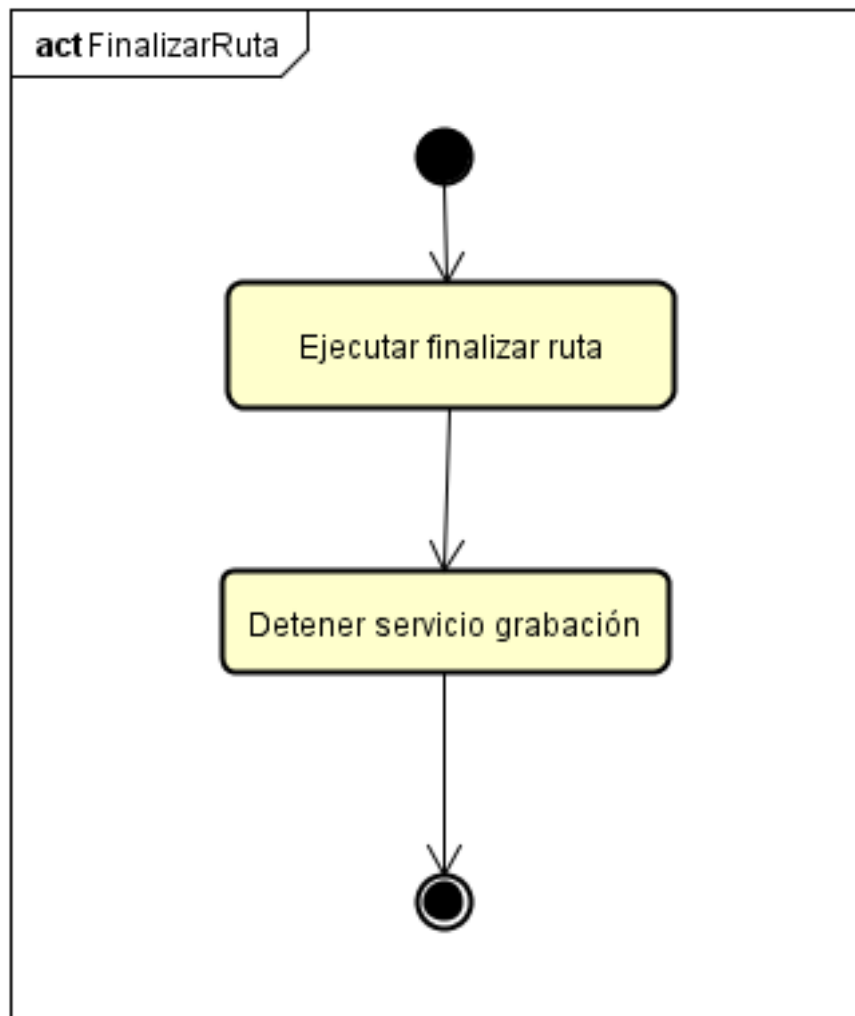


Figura 4.10: Diagrama de actividad del CU07

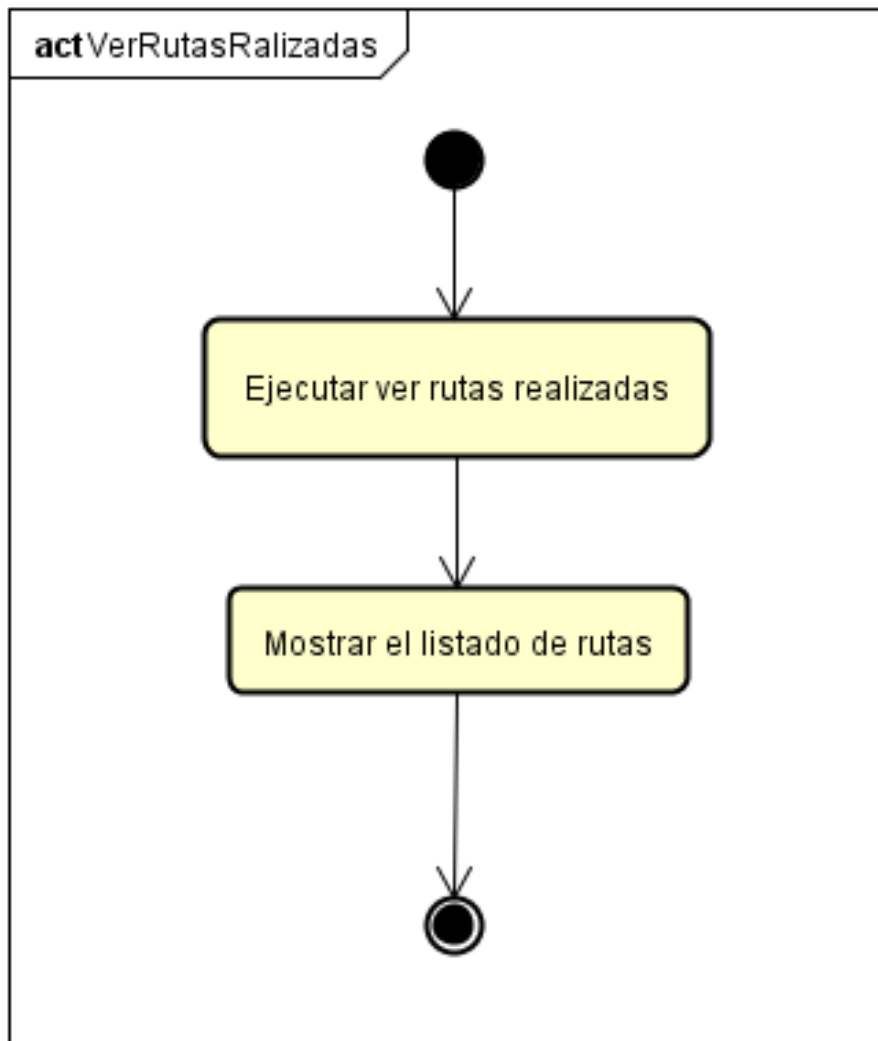


Figura 4.11: Diagrama de actividad del CU08

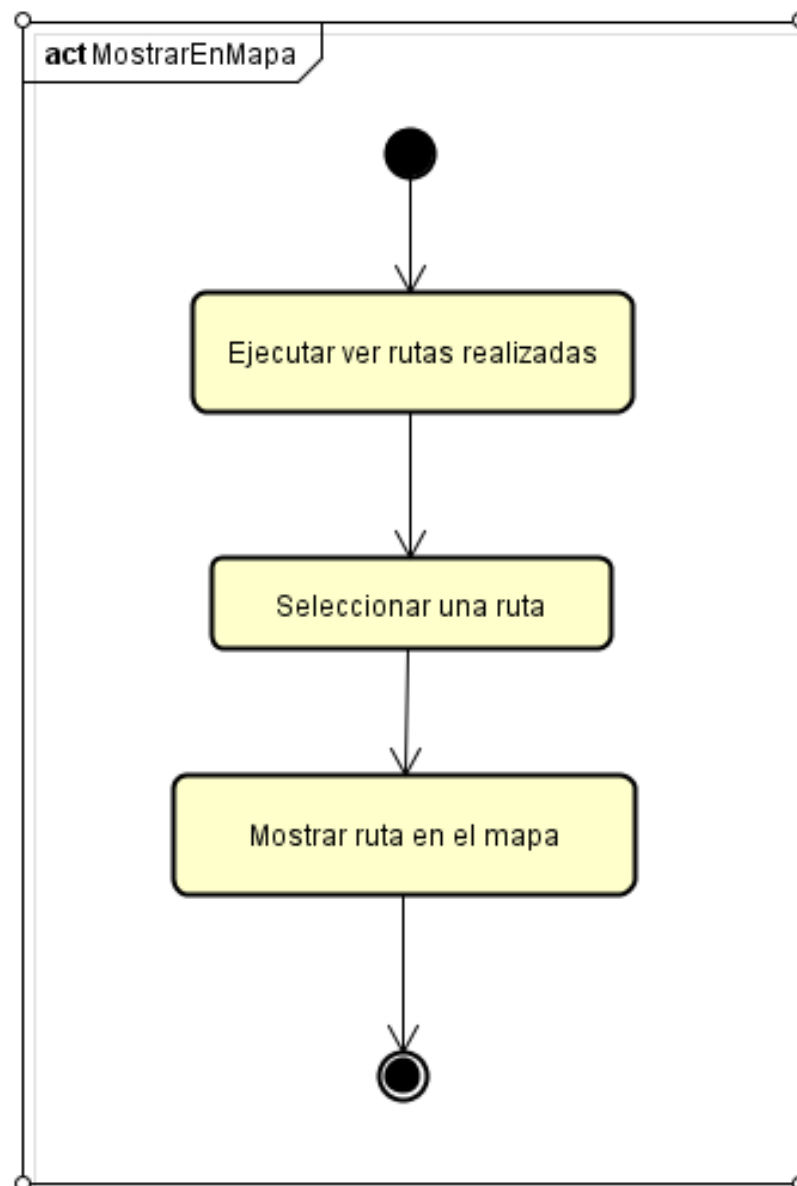


Figura 4.12: Diagrama de actividad del CU09

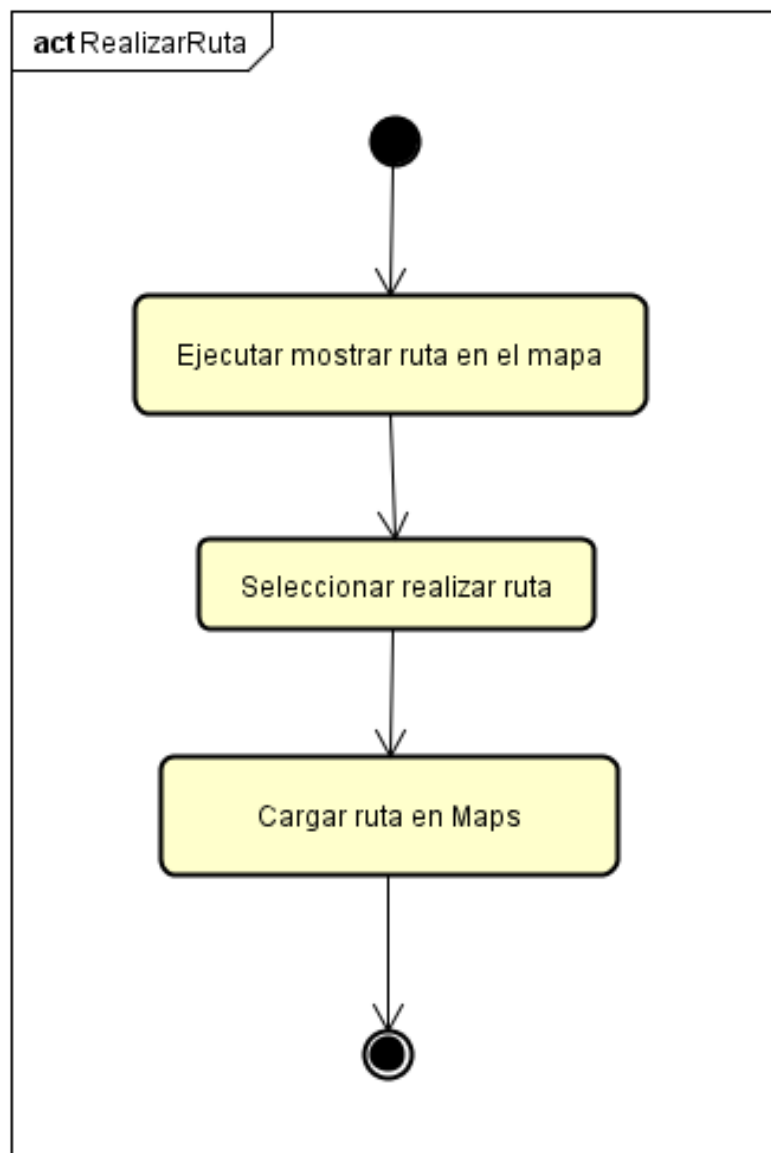


Figura 4.13: Diagrama de actividad del CU10

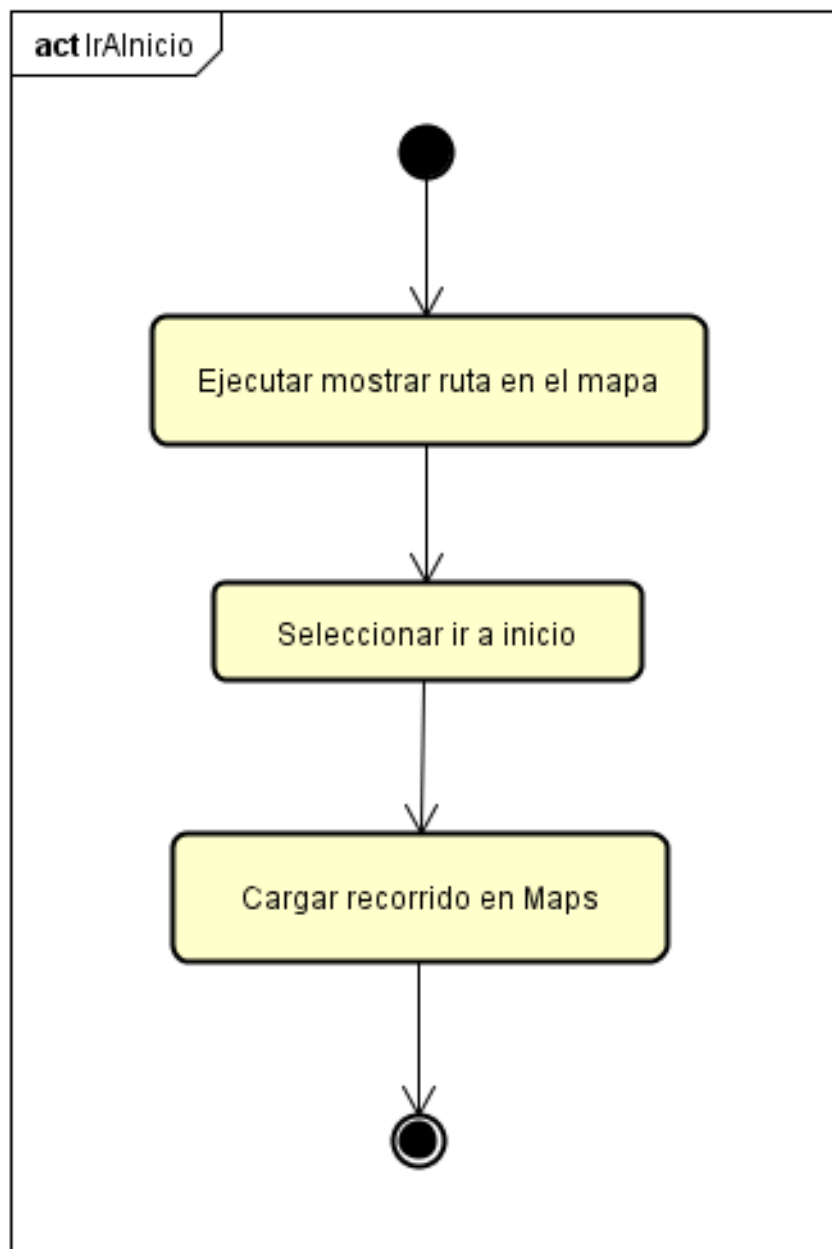


Figura 4.14: Diagrama de actividad del CU11

Capítulo 5

Arquitectura y diseño

5.1. Introducción

En este capítulo se procede a detallar los aspectos referentes a la arquitectura del sistema, junto con el diseño gráfico de la interfaz visual del mismo. Para esto se procede a describir los patrones de diseño seguidos así como las dinámicas implementadas, junto con los lenguajes de programación y la tecnología aplicada.

5.2. Aproximación inicial al diseño

Como visión general a la dinámica de la aplicación y las interacciones de la misma, en la figura 5.1 se muestra el diagrama de la arquitectura propuesta, en él se encuentran los elementos que participan en la misma.

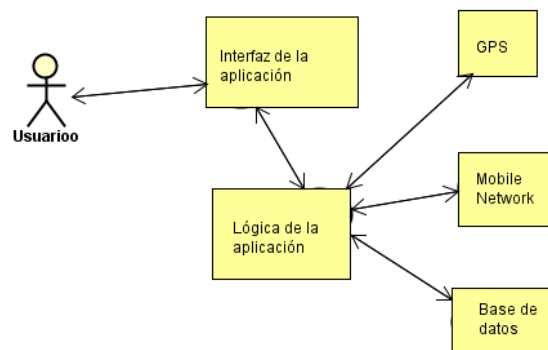


Figura 5.1: Diagrama general de la arquitectura de la aplicación

El módulo de Lógica de la aplicación contiene todo lo referente a la lógica de negocio, el almacenamiento y visionado de la información del usuario, la visualización de los mapas y la grabación y posterior representación de las rutas.

La base de datos es la encargada de almacenar todos los datos anteriormente mencionados, los cuales se van a actualizar por el usuario o de forma automática como es el caso de las rutas.

Por otro lado todo esto se representa al usuario mediante una interfaz que sera la que interactuara con el módulo de la lógica para representar los datos y transmitir las acciones realizadas por el usuario a la lógica para que esta realice los procesos necesarios.

Además la lógica utiliza de forma conjunta tanto el GPS como la red móvil des dispositivo en el que se está ejecutando, es necesaria al menos una de ellas pero, se han utilizado las dos para mejorar así la precisión de la ubicación y realizar una mejores grabaciones de las rutas.

5.3. Arquitectura propuesta del sistema cliente

En este apartado se busca enseñar una visión general de la arquitectura propuesta para el sistema. Como se muestra a continuación, en la figura 5.2 se expone la arquitectura lógica de la aplicación.

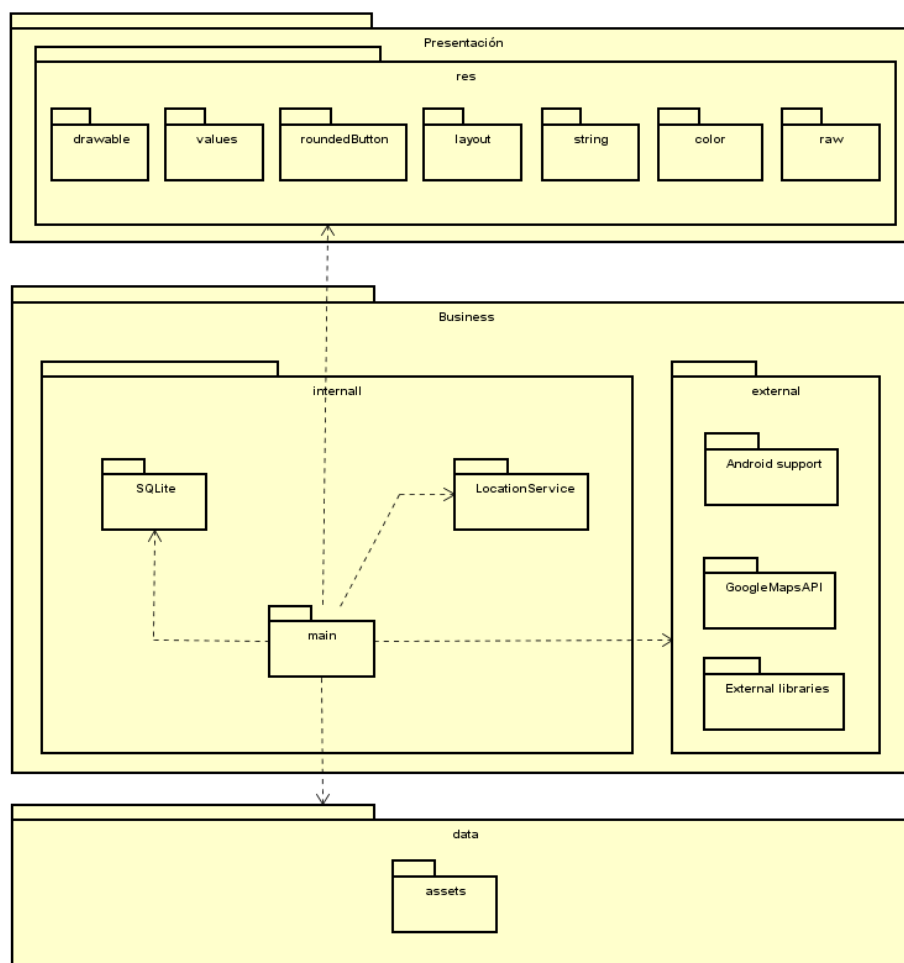


Figura 5.2: Diagrama general de la arquitectura de la aplicación

En esta imagen se pueden ver 3 capas diferenciadas que componen la aplicación. La primera de ellas comenzando desde arriba es la capa de presentación, Android pone a disposición del desarrollador una serie de elementos para la interfaz dentro del paquete denominado 'res', aquí se muestran algunas de ellas.

- drawable: Elementos visuales de la interfaz, entre ellos: botones, check box, desplegados...
- values: Unidades de medida para los componentes gráficos
- layout: Son los encargados de representar cada ventana de la aplicación y contener al resto de elementos, se escriben en XML.
- string: Cadenas de texto que aparecen para interactuar con el usuario, permite la internacionalización
- color: Define colores concretos para los elementos de la aplicación.
- raw: Otros elementos como archivos JSON.

La capa intermedia es la capa que contiene la lógica de negocio, en ellas nos encontramos dos grupos que engloban a los componentes de la lógica de negocio.

- internal: Aquellos elementos que se utilizan y además son propios de la aplicación.
 - main: Es el paquete principal de la lógica, es el encargado de gestionar el resto de paquetes, tanto de la lógica como lo que se muestra en la pantalla.
 - SQLite: Es el paquete encargado de manejar todos los datos con los que trabaja la aplicación.
 - LocationService: Es el paquete encargado del servicio de ubicación, realiza la actualización de la posición así como la grabación cuando sea necesario.
- external: Aquellos elementos que se utilizan pero que no son propios de la aplicación.
 - Android support: Permite el acceso a todos los elementos del sistema Android.
 - GoogleMapsAPI: Permite el uso del sistema de representación de mapas y ubicación de Google.
 - External libraries: En él se encuentran todas las bibliotecas externas utilizadas como apoyo al proyecto.

5.4. Topología del sistema

En la figura 5.3 se muestra el diagrama de despliegue del sistema.

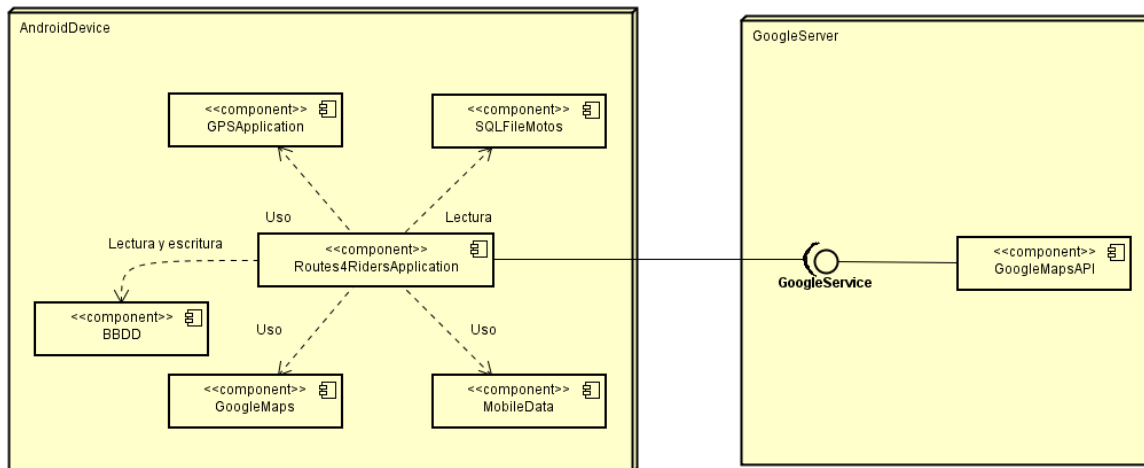


Figura 5.3: Diagrama de despliegue del sistema

Si miramos de izquierda a derecha lo primero que se observa es el paquete denominado 'Android-Device' este paquete estaría representando a un dispositivo que conste de sistema operativo Android, en el cual se instalará la aplicación móvil desarrollada en este proyecto 'Routes4RidersApplication'.

Junto con la aplicación este dispositivo ha de disponer de servicio de ubicación por GPS y tener habilitados y disponibles los datos móviles. Esto es necesario para la aplicación para poder funcionar de forma correcta puesto que conocer la ubicación es el requisito principal para la misma.

En adición a lo anterior el sistema ha de contar con la aplicación de Google Maps instalada previamente para poder realizar las funciones de guiado, mediante indicaciones de voz y por representación del recorrido en un mapa.

El componente de base de datos es proporcionado por el propio sistema operativo Android para todas las aplicaciones que se desarrollan para el mismo, en este caso sera utilizado de forma frecuente por la aplicación y es necesario la configuración y gestión del mismo por parte de la aplicación.

El componente SQLFileMotos esta unido al componente de base de datos, pues para que la aplicación disponga de un listado de motos en la misma, este fichero es ejecutado al inicio para poblar la base de datos con un grupo de marcas de motocicletas así como varios modelos para cada una de ellas.

Por otro lado 'AndroidDevice' utiliza la API de Google 'MapsAPI' que nos suministra los datos nece-

sarios para representar y mostrar la ubicación del dispositivo en todo momento, esto es necesario para cargar los mapas, representar la ubicación en los mismos y recoger los datos de las coordenadas.

5.5. Mecánica del sistema

En esta sección se busca realizar un resumen, de todas las dinámicas implementadas en el sistema. Esta información se encuentra completa y disponible en el anexo C en el cual se especifica un manual de usuario con una guía básica de uso del sistema.

Inicialmente se dan dos posibilidades al intentar interactuar con la aplicación, que el usuario no se haya registrado previamente, o que el usuario ya este registrado.

Si el usuario no se ha registrado previamente la interacción con la aplicación comienza con la petición de registro, una vez introducidos los datos requeridos se muestra el mapa, en el caso contrario, al estar registrado el usuario se muestra una pantalla de carga y se muestra el mapa.

Una vez en la vista del mapa y mostrando nuestra posición actual, previamente puede ser necesario proporcionar a la aplicación acceso a la ubicación si es el primer uso o si no se ha seleccionado permitir siempre acceso a la ubicación.

En este punto la aplicación estaría mostrando la interfaz principal y ya podría usarse a modo de GPS sin asistente navegación solo para conocer el camino y la ubicación por la que se circula, a este estado y pulsando el botón de iniciar ruta se comenzaría a grabar el trayecto que se esta realizando, hasta pulsar el botón de terminar ruta que de tendría la grabación de la misma, aunque se mantendría la funcionalidad de GPS.

En los botones superiores de esa interfaz principal podremos acceder, con el superior izquierdo a ver y modificar los datos de usuario y con el superior, podremos ver las rutas almacenadas.

Como último detalle si el usuario entra en la rutas y pulsa en una de ellas esta se mostrará en el mapa a modo de vista previa y ofrecerá las opciones de ir al inicio de la misma, o realizar la ruta con indicaciones por voz en ambos casos gracias a Google Maps, aplicación que ha de estar instalada en el dispositivo.

Capítulo 6

Pruebas

6.1. Introducción

En este capítulo se busca mostrar los resultados de los prototipos resultantes de cada iteración realizada, las pruebas se han realizado de forma interna por el propio alumno y por el tutor de la asignatura, de forma externa al desarrollo del proyecto personas cercanas al alumno han colaborado en las pruebas, tratando así de realizar unas pruebas lo mas extensas posible en cuanto a variedad de dispositivos y versiones de Android.

Esto permitió ajustar la aplicación lo mas posible para diferentes terminales y leves problemas concretos que se producían en algunos de ellos, al no disponer de una gran muestra en cuanto a los dispositivos para probar, se han generalizado los resultados de esas pruebas para el resto de dispositivos y versiones de Android que no han podido ser probadas

Al realizarse dos prototipos diferentes durante el desarrollo se mostraran los resultados de forma independiente.

6.2. Resultado prototipo 1

6.2.1. Introducción

El primer prototipo se centro en la integración de la API de Google Maps para poder obtener los datos de navegación necesarios, este prototipo inicial contaba con una interfaz muy simple que buscaba únicamente la activación y desactivación del servicio que graba las rutas de forma automática y permitir ver las rutas almacenadas como muestra en la figura6.1.

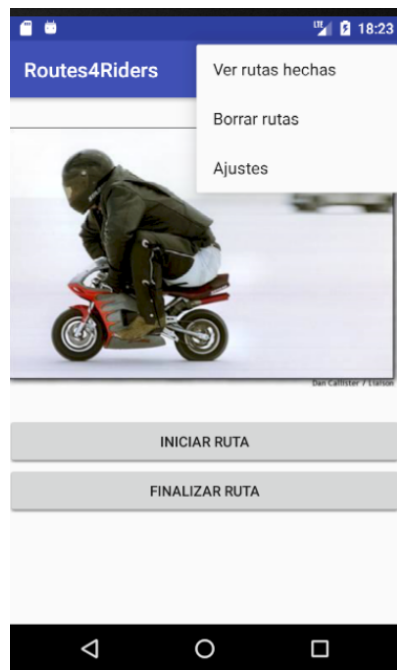


Figura 6.1: Pantalla menú del prototipo 1

También se implementó la petición explícita de permisos para la ubicación en este primer prototipo, pues la versión objetivo de Android indica la necesidad de realizarlo de esta manera y no asumir una aceptación implícita, esto se puede ver en la figura 6.2.

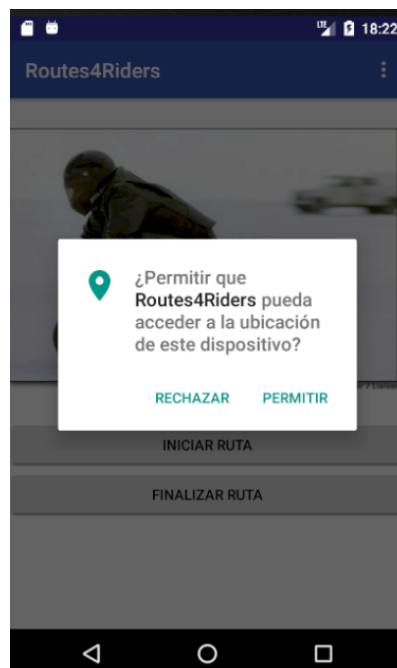


Figura 6.2: Pantalla petición permisos prototipo 1

Al ser un servicio en segundo plano no era necesario mantener la aplicación abierta ni la pantalla del dispositivo activa durante el funcionamiento del servicio.

6.2.2. Pruebas

Una vez instalada la aplicación en los dispositivos de los voluntarios, se realizó una explicación acerca del funcionamiento de la misma a una parte de ellos, mientras que al grupo restante no se les indicó nada acerca de su funcionamiento, esto permitiría mas adelante en la toma de resultados tener los diferentes puntos de vista. Usuarios con conocimiento previo de uso de la aplicación y usuario sin conocimiento de uso previo.

Una vez finalizada la explicación e instalación de la aplicación se fijó una fecha en la cual reunirse para comentar las impresiones que tenían sobre el uso y funcionamiento del primer prototipo de la aplicación.

En la reunión para ver los problemas fueron comunicadas diferentes incidencias, entre ellas muchas a nivel interfaz, pero como este primer prototipo no buscaba la interfaz ni la usabilidad como objetivo principal no era un problema, ya que la interfaz se pretendía cambiar totalmente.

El principal problema que se dio en algunos de los dispositivos fue debido al servicio en segundo plano que se encarga de la grabación de las rutas, en función de las configuraciones de energía del dispositivo este servicio era detenido por el dispositivo poco después de apagar la pantalla, solo continuaba la ejecución al entrar de nuevo a la aplicación para detener el servicio, esto suponía que solo se registrarán el inicio y el final del recorrido, pero no figuraban puntos intermedios en el mapa. Al tratar luego de ver la ruta esto generaba una línea recta entre origen y fin independiente de por donde se hubiera circulado.

Para solventar este error para el siguiente prototipo se implementó que mientras el servicio estuviera activo la pantalla se mantuviera también activa y así evitar que las políticas de ahorro de energía del dispositivo interfirieran en la aplicación, como se verá mas adelante esta implicó el uso de un mapa, además de la actualización de la posición para justificar mantener la pantalla activa. Esto se detalla mas adelante.

6.2.3. Conclusiones

Para este primer prototipo y tras evaluar las pruebas hemos obtenido mucha información sobre cambios en la interfaz que pueden ayudar a la idea inicial que se tenía sobre como implementar la misma, así mismo se ha obtenido información relevante acerca de los fallos que podía dar el servicio según dispositivo, lo cual es muy útil y se buscará corregir en el segundo prototipo.

6.3. Resultado prototipo 2

6.3.1. Introducción

Para este segundo prototipo y con las anotaciones sobre mejoras del primero, nos centraremos sobretodo en la implementación de una interfaz usable y atractiva para el usuario, así mismo se revisará el funcionamiento del servicio integrando la actualización de la posición en tiempo real sobre un mapa, de esta manera se puede justificar el por qué mantener la pantalla y la aplicación continuamente activa y en primer plano y así conseguimos evitar detenciones inesperadas del servicio durante la grabación de las rutas.

También se buscará pensando en futuras actualizaciones ampliar la aplicación con una recogida de datos sobre los usuarios y vehículos de los mismos, de esta manera se podrá crear mayor comunidad en próximas iteraciones, buscando un nivel mas allá del simple uso individual de esta aplicación, tratando de realizar una gamificación con logros en la misma.

6.3.2. Pruebas

Al igual que con el primer prototipo se realizó de nuevo la instalación de la aplicación en los dispositivos de los voluntarios, se volvió a dividir en dos grupos, pues al realizar un cambio completo de la interfaz, se podía realizar de nuevo la explicación a un grupo y a otro no para ver las diferencias entre los que si tenían una explicación previa de uso y los que no. En este segundo prototipo también se contó con la participación del tutor del TFG realizando una prueba desde el emulador de Android Studio, esto supone alguna limitaciones por el tipo de aplicación desarrollada pero si puede proporcionar comentarios valiosos en cuanto a interfaz y funcionamiento básico de la misma.

Una vez pasado el periodo acordado para que se realizaron las pruebas se recogen las opiniones y comentarios de los voluntarios.

Se reciben buenas impresiones acerca del funcionamiento de la aplicación, se ha solventado el problema del servicio en los dispositivos en los cuales suponía un fallo en el primer prototipo, así mismo los usuarios que trataron de volver atrás en la selección de moto sin aceptarla, pulsando el botón de retroceso en el dispositivo, indicaron que la aplicación se cerraba y abría o mostraba un mensaje de cierre accidental y se abría, esto dependía de la versión de Android instalada en el dispositivo. Este problema fue solventado incluyendo en la aplicación la opción de volver atrás mediante el botón del dispositivo, la cual no se había implementado.

En lo que a la interfaz se refiere los voluntarios nos indican una interfaz mas trabajada y la encuentran comprensible y sencilla se entender, los usuarios a los que se les explicó no encontraron ninguna dificultad. Por otro lado los que no recibieron dicha explicación con ella si que encontraron algún problema de comprensión en la pantalla principal con los botones superiores, pues los iconos elegidos no indican claramente que va a encontrarse en esas opciones, al no querer poner texto en ellos se ha tratado de buscar unos iconos lo mas intuitivos posible, aun así también se explica de forma detallada

en el manual de usuario la utilidad de dichos botones.

Otros comentarios acerca de la interfaz fueron los botones de inicio y fin de ruta, pues al ser excluyentes se indicó que se podían sustituir por un único botón que en función del estado cambiase el texto y la funcionalidad, evitando así confusión y ocupar parte del mapa con botones que no se pueden usar de forma independiente. También se comentó la ubicación del botón de aceptar en la pantalla de cambios en el perfil y de seleccionar moto, era una posición poco intuitiva, por esto esos botones se han colocado debajo de la última opción elegible en cada caso, de esta manera para aceptar basta con seguir el flujo de la página, lo cual es más intuitivo y rápido de visualizar para el usuario.

Si se trata de avanzar sin poner los datos correctamente en cualquier punto se muestra un aviso en la parte inferior de la pantalla, informando de ellos y no se pasa a la siguiente pantalla.

Si se trata de ir atrás, mediante el botón del dispositivo habilitado para ello en los datos de la moto para evitar ponerlos, la aplicación muestra los datos anteriores de la base de datos o lo deja en blanco, en este segundo caso entra en juego la prueba anteriormente comentada si se trata de avanzar a la siguiente ventana.

Si se trata de cargar una ruta y esta no tiene puntos, se muestra un aviso de la inexistencia de puntos en la ruta y se muestra la lista de selección de rutas de nuevo.

Si se trata de grabar una ruta o cualquier otra acción que necesite de los permisos de ubicación y estos no han sido aceptados se muestra un aviso recordando que no se han aceptado los permisos de ubicación y que por tanto la ruta no se grabará.

Si se desplaza la vista del mapa a una zona en la que no se vea nuestra posición esta regresa de forma automática al detectar cualquier variación en la ubicación.

Si se trata de utilizar las funciones de ir a inicio o realizar ruta sin la aplicación de Google Maps en el dispositivo se abrirá Google Maps en el navegador predefinido para mostrarla sobre el mapa, pero no se podrán recibir las indicaciones.

6.3.3. Conclusiones

De este segundo prototipo obtenemos ya una versión con la funcionalidad principal que se buscaba junto con algún complemento de cara a futuro. Los voluntarios han ayudado a asegurar en la medida de lo posible el funcionamiento es una serie de dispositivos diferentes, se reservan algunas opiniones y sugerencias como líneas de actuación futuras, pero actualmente la aplicación cumple su objetivo básico. Se extrae que los usuarios no han tenido excesivos problemas con el uso de la aplicación y el nivel de satisfacción es aceptable.

Las pruebas realizadas tanto en la grabación de los recorridos como en la recuperación y representación de los mismos ha sido satisfactoria, además del proceso por el cual se llamaba a Google Maps, que ha proporcionado la funcionalidad buscada sin problemas para los usuarios.

Capítulo 7

Conclusiones y líneas futuras

7.1. Conclusiones

En este último capítulo de esta memoria se intenta resumir las principales ideas del trabajo realizado. También se comentarán las posibles líneas futuras, las cuales dependen de los desarrollos aplazados que se indican en el capítulo 3 de esta memoria.

A lo largo de este proyecto se han llevado a cabo varias fases iterativas que han permitido desarrollar prototipos incrementales del sistema. Esto ha permitido mejorar la aplicación mediante opiniones de los voluntarios además de poder incrementar la funcionalidad de forma controlada, pudiendo tratar los imprevistos de una forma más controlada y con una menor repercusión en el desarrollo total.

Se ha desarrollado una herramienta que permitiera la grabación de rutas por las que se circula con un vehículo, pensando en motocicletas como objetivo principal.

Se implementó una aplicación móvil basada en geolocalización y servicios en segundo plano. Al contar con un pequeño grupo de usuarios de prueba, de los cuales se podía generalizar el público objetivo de la aplicación, se puso más atención a los detalles en cuanto a lo que la interfaz de usuario se refiere.

En cuanto a las opiniones con la aplicación, se recibieron opiniones buenas en general, tanto en el funcionamiento como en la interfaz de la misma.

Por último destacar que, aunque este tipo de aplicaciones aunque aún pueden mejorar, la API que proporciona Google en este caso, es muy completa permitiendo capturar todo tipo de información del GPS y antena del dispositivo, esto facilita el desarrollo y el tratamiento de esa información al trabajar con parámetros ya estandarizados.

7.1.1. Líneas futuras

Tras la finalización del proyecto, se indican las líneas por las que se podría continuar el mismo, las cuales ya se han comentado anteriormente en esta memoria.

La primera sería como ampliación de la misma en posibilidades que no se han podido abarcar durante el desarrollo de este trabajo.

Entre ellas se podrían destacar la creación de un web service o una API REST para externalizar todo el tratamiento de datos. Por otro lado se podría incluir la gamificación y los logros por hitos referentes a las rutas realizadas, completadas, distancias recorridas... Poder compartir las rutas mediante la propia aplicación gracias a la externalización antes comentada, así como gestionar un navegador interno en la propia aplicación.

Otro punto de mejora sería incluir mas información sobre las rutas, distancia, consumo estimado y todo en el listado de rutas antes de la visualización de las mismas.

Un segundo caso y mas ambicioso pasaría por tratar de aumentar el ámbito de la aplicación y convertirla en un producto comercial, para lo cual tendría que pasar a ser una aplicación en el mercado de aplicaciones, esto implicaría una revisión para tener en cuenta las posibles exigencias no contempladas para poder incluir la aplicación en el mercado correspondiente.

Anexos

Anexo A

Glosario

- Ruta: Estructura de datos compuesta por puntos y que simboliza una serie de líneas que plasmadas sobre el mapa llevan desde un punto de origen a uno de destino.
- Punto: Unidad mínima de la ruta, se traduce a un punto en cuanto a posición sobre el mapa se refiere y se traduce en sus valores de latitud y longitud.
- Grabación: Forma de denominar al proceso por el cual se recogen los puntos del mapa y se asocian a una ruta.
- Servicio: Trabajo en segundo plano de la aplicación que se encarga de realizar el trabajo de conversión de puntos a coordenadas en latitud y longitud.

Anexo B

Acrónimos

- AOT: Ahead Of Time.
- API: Application Programming Interface.
- ART: Android Runtime.
- CU: Caso De Uso.
- GPS: Global Positioning System.
- GSM: Global System for Mobile Communications.
- HAL: Hardware Abstraction Layer.
- IETF: Internet Engineering Task Force.
- JIT: Just In Time.
- JSON: JavaScript Object Notation.
- REST: Representational State Transfer.
- SQL: Structured Query Language.
- SMS: Short Message Service.
- SO: Operating System.
- SSL: Secure Sockets Layer.
- TLS: Transport Layer Security
- UTF-8: 8-bit Unicode Transformation Format
- XML: eXtensible Markup Language

Anexo C

Manual de usuario

En este anexo se procede a detallar de forma básica los elementos de la interfaz que va a encontrar el usuario en nuestra aplicación y la interacción que puede realizar con los mismos. En la figura C.1 se puede ver la primera pantalla que un usuario no registrado se va a encontrar al iniciar la aplicación.



Figura C.1: Pantalla registro de la aplicación

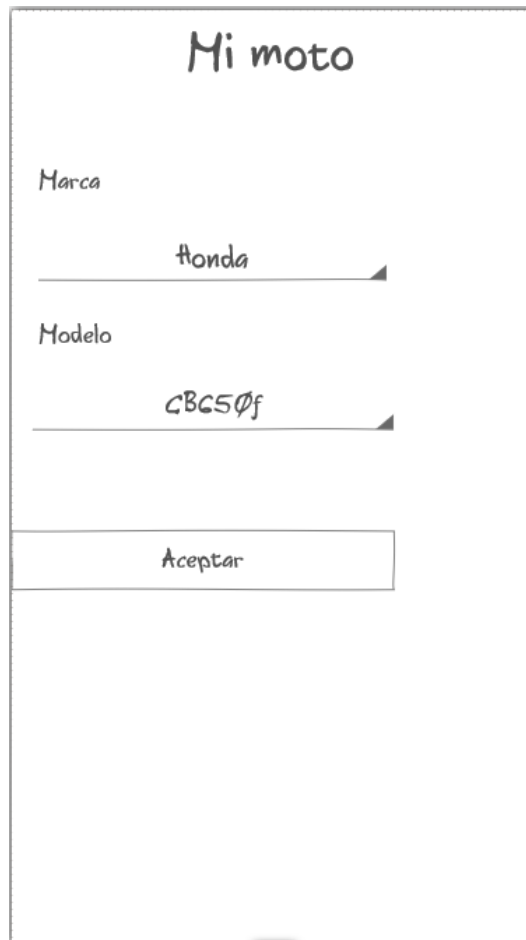
Esta pantalla contiene un formulario básico en el cual solo se muestra un campo de un usuario y uno de contraseña para poder acceder a la aplicación, una vez introducida la información al pulsar el botón entrar se accederá a la pagina para obtener mas información del usuario la cual se muestra en la figura C.2 que está a continuación. Esta pantalla permite recabar mas información sobre el usuario así como seleccionar la marca y el modelo de su motocicleta, aunque esto lo veremos en la siguiente figura.

The sketch shows a mobile app screen titled "Perfil". It contains the following elements from top to bottom:

- A user icon followed by a "Text Field" for the username.
- An envelope icon followed by a "Text Field" for the password.
- The text "Mi moto" followed by a green "+" button.
- A motorcycle icon followed by a "Text Field" for the bike model, and a red "X" button.
- A wide rectangular button labeled "Aceptar".

Figura C.2: Pantalla datos del usuario de la aplicación

Como muestra la imagen se piden datos a mayores y se incluirían aquí los introducidos en la primera pantalla, solo será necesario añadir el correo electrónico y el modelo de moto. Si se pulsa en la X se eliminaría la moto actual, en caso de haber seleccionado ya una y si se pulsa en el + para añadir una nueva pasaríamos a la figura C.3 la cual nos permite elegir de un listado la marca y modelos de nuestro vehículo. Si el modelo no esta actualmente indicar la que aparece por defecto y comentarla para que sea añadida en próximas actualizaciones.



Hand-drawn mockup of a mobile application screen titled "Mi moto". The screen contains two dropdown menus. The first dropdown is labeled "Marca" and has "Honda" selected. The second dropdown is labeled "Modelo" and has "CB650f" selected. Below these dropdowns is a button labeled "Aceptar".

Figura C.3: Pantalla datos moto de la aplicación

Una vez seleccionada la marca se mostrarán los modelos asociados disponibles. Una vez finalizada la selección se pulsa aceptar y se volverá a la pantalla de la figura.C.2

Para un usuario que ya ha realizado previamente todo este proceso y por tanto no es la primera vez que inicia la aplicación lo que inicialmente se mostrará es una pantalla de carga típica con la misma imagen de la pantalla inicial pero sin los campos de recogida de datos. Esta pantalla se mantendrá unos segundos y ya se pasará a la pantalla principal de la aplicación. La pantalla de carga se representa en la figura C.4 que se muestra a continuación.



Figura C.4: Pantalla carga de la aplicación

En ambos casos la pantalla a la que se llega y la cual es la principal de la aplicación es la que se muestra en la figura C.5, esta pantalla muestra nuestra posición en tiempo real sobre el mapa, junto con esto, en esta pantalla vemos varios botones que se comentarán a continuación:

- Botón superior izquierdo: Al pulsar este botón nos llevara a la pantalla de datos, la cual se representa en la figura C.2, desde la que se podrá revisar o actualizar los mismos.
- Botón superior derecho: Al pulsar este botón nos llevará al listado de las rutas que hemos guardado en el dispositivo, Al grabarlas, aparecerán identificadas por fecha y hora de realización, dicha pantalla se muestra en la figura C.6 y se comentará mas adelante la funcionalidad de la misma.
- Botón central inferior: Al pulsar este botón se inicia la grabación de la ruta y guardará de forma automática los puntos por los que nos desplazamos para mas adelante poder verlos o recorrerlos de nuevo. Este botón cambiará al ser pulsado y como se aprecia, cambia el texto de "Iniciar ruta" por el de "Detener ruta" que permitirá detener la grabación de la ruta que se este realizando actualmente.

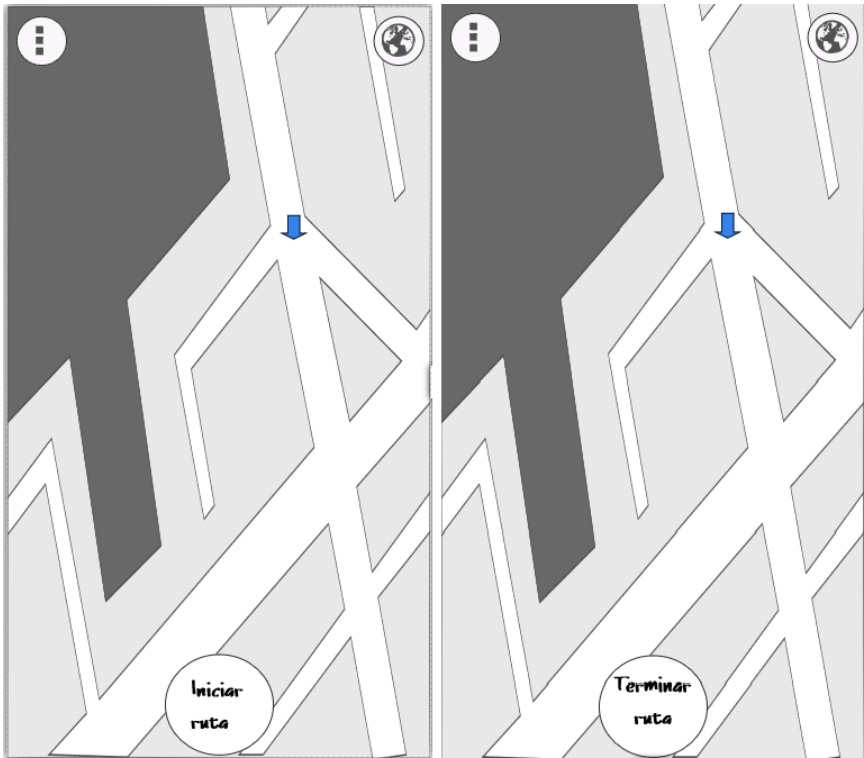


Figura C.5: Pantalla principal de la aplicación

Como se comentó anteriormente al pulsar en nuestras rutas pasamos a la siguiente pantalla.

Ruta realizada el: (fecha de ruta)
Ruta realizada el: (fecha de ruta)
Ruta realizada el: (fecha de ruta)
Ruta realizada el: (fecha de ruta)
Ruta realizada el: (fecha de ruta)

Figura C.6: Pantalla de listado de rutas

En la pantalla de la figura C.6 se puede ver el listado de rutas grabadas en la aplicación, si se hace click en cualquiera de ella pasamos a una previsualización de las mismas como se muestra en la figura C.7. En esta pantalla se aprecian dos botones en la parte inferior, ambos trabajan con la aplicación de

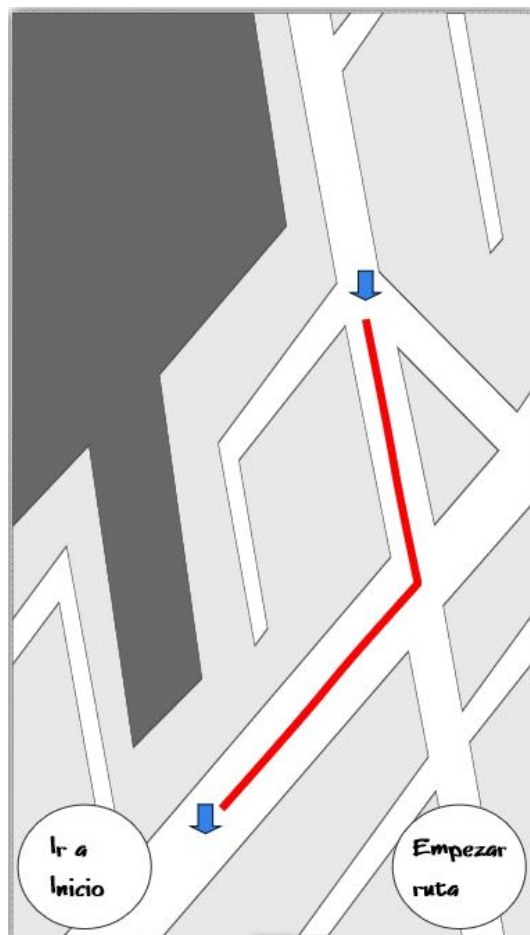


Figura C.7: Pantalla de previsualización de rutas

Google Maps del dispositivo:

- Botón inferior izquierdo: Al pulsar este botón se abrirá la aplicación de Google Maps a la que se le enviará el punto de inicio de la ruta como destino para que nos guíe desde nuestra ubicación actual al inicio de la ruta almacenada.
- Botón inferior derecho: Al pulsar este botón se abrirá la aplicación de Google Maps a la que se le enviarán los puntos de inicio o fin de la ruta almacenada para que nos ofrezca las indicaciones y recorrer de nuevo con las indicaciones proporcionadas por Google Maps. Gracias a la aplicación también se nos permite compartir las rutas grabadas en nuestro dispositivo con otros compañeros para la ruta.

Anexo D

Manual de instalación

En este anexo se proporciona una guía detallada de la instalación de la aplicación móvil, así como los requisitos mínimos necesarios para dicha instalación.

Antes de realizar la instalación de la aplicación se han de cumplir unos requisitos previos:

1. Dispositivo Android con versión 4.1 o superior.
2. El dispositivo móvil ha de disponer de al menos 20 MB de memoria interna disponible.
3. Dispositivo con receptor de satélites GPS.
4. Dispositivo con conexión a Internet.
5. Dispositivo con la última versión de la aplicación Google Maps instalada.
6. Dispositivo con la última versión disponible de Google Play Services instalada.

Si se cumplen estas condiciones previas estos son los pasos a seguir para realizar la instalación de la aplicación en el dispositivo (El ejemplo se realizara sobre un dispositivo con Android 6.0 por ser la versión objetivo del sistema que se ha elegido para la aplicación, el proceso puede variar ligeramente según la versión de Android del dispositivo). Lo primero que se necesita es permitir la instalación de aplicaciones de orígenes desconocidos, esto se muestra en la figura D.1. Para permitir se ha de ir a Ajustes, después al apartado de seguridad y allí hay que habilitar la opción de orígenes desconocidos, como esta opción mejora la seguridad del dispositivo al finalizar la instalación de la aplicación, es importante que se vuelva a deshabilitar esta opción.

Este paso previo es necesario al no ser una aplicación que se instala directamente del mercado de aplicaciones.

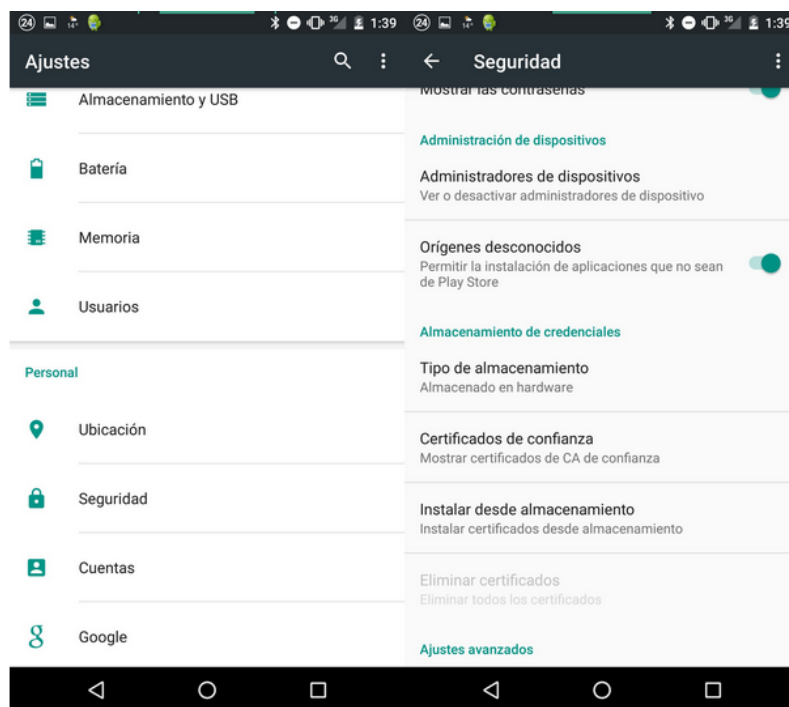


Figura D.1: Pantalla de seguridad permitir instalación

El siguiente paso será buscar en el administrador de archivos del dispositivo la aplicación en la carpeta donde se haya descargado, hay que encontrar el archivo (Routes4Riders.apk). Esto se muestra a continuación en la figura D.2.

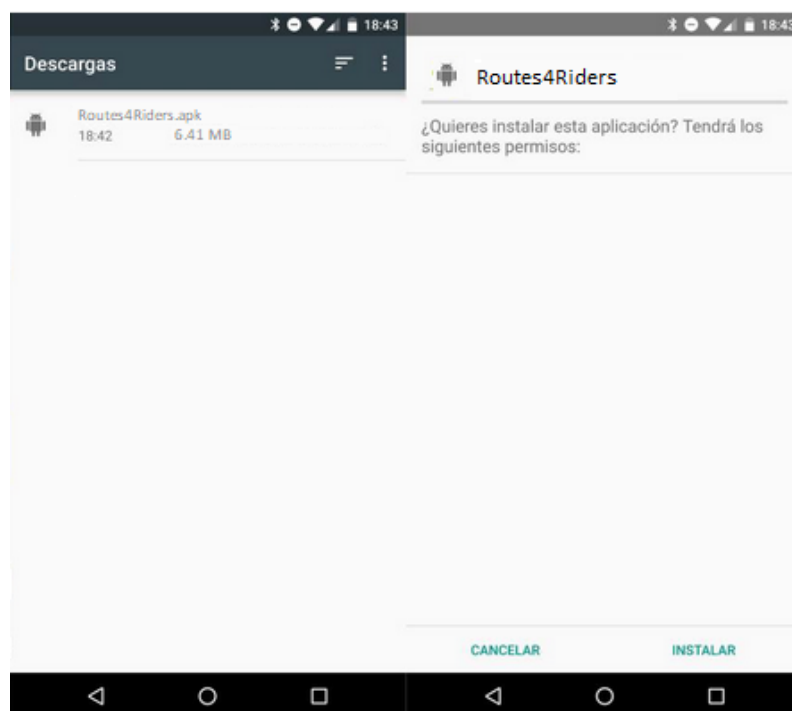


Figura D.2: Pantalla de instalación

Anexo E

Contenido de CD

- memoria.pdf
- Routes4Riders.zip (Carpeta comprimida con todo el código de la aplicación desarrollada).
- Routes4Riders.apk (Archivo de instalación de la aplicación desarrollada).

Bibliografía

- [1] Activity, A. (2018). Android Two Fragments in Same Activity. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/17580593/android-two-fragments-in-same-activity> [Accessed 29 Mar. 2018].
- [2] Android Developers. (2018). Arquitectura de la plataforma | Android Developers. [online] Available at: <https://developer.android.com/guide/platform/index.html?hl=es-419> [Accessed 28 Feb. 2018].
- [3] Android Developers. (2018). FragmentActivity | Android Developers. [online] Available at: <https://developer.android.com/reference/android/support/v4/app/FragmentActivity.html> [Accessed 11 Mar. 2018].
- [4] Android Developers. (2018). Platform Architecture | Android Developers. [online] Available at: <https://developer.android.com/guide/platform/index.html> [Accessed 10 Mar. 2018].
- [5] Android tutorials for hassle-free android development and programming. (2018). Android Splash Screen Tutorial : How to create animated splash screen in Android Studio. [online] Available at: <https://www.androidtutorialpoint.com/material-design/android-splash-screen-tutorial-create-animated-splash-screen-android-studio/> [Accessed 23 Apr. 2018].
- [6] android, g. (2018). getActivity() with in FragmentActivity: android. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/12693251/getactivity-with-in-fragmentactivity-android> [Accessed 13 Mar. 2018].
- [7] Android, H. (2018). How to update marker positions with data from Firebase Google Maps API Android. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/47681030/how-to-update-marker-positions-with-data-from-firebase-google-maps-api-android> [Accessed 30 Mar. 2018].
- [8] AndroidStudio (2018) setBackgroundColor with Hex Color Codes AndroidStudio. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/25837449/setbackgroundcolor-with-hex-color-codes-androidstudio> [Accessed 5 May 2018].
- [9] Anon, (2018). [online] Available at: https://www.researchgate.net/figure/CONCURRENT-ENGINEERING-PROCESS_fig1_271522071, [Accessed 18 Apr. 2018].
- [10] Coderzheaven.com. (2018). How to pass an object back from a finishing activity to the previous activity in android? – CoderzHeaven. [online] Available at:

- <http://www.coderzheaven.com/2013/03/24/pass-object-finishing-activity-previous-activity-android/> [Accessed 5 Mar. 2018].
- [11] ComputerHoy. (2018). Mejora el funcionamiento de tu GPS Android. [online] Available at: <https://computerhoy.com/paso-a-paso/moviles/mejora-funcionamiento-tu-gps-android-36591> [Accessed 26 Feb. 2018].
- [12] Console.developers.google.com. (2018). Google Cloud Platform. [online] Available at: <https://console.developers.google.com/google/maps-apis/apis/maps-android-backend.googleapis.com/metrics?project=routes4riders-190115&hl=es> [Accessed 4 Mar. 2018].
- [13] ContentProvider, C. (2018). Cannot add data to database using ContentProvider. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/49095049/cannot-add-data-to-database-using-contentprovider> [Accessed 22 Mar. 2018].
- [14] Developer.android.com. (2018). Platform Architecture | Android Developers. [online] Available at: <https://developer.android.com/guide/platform/index.html> [Accessed 3 Apr. 2018].
- [15] dot, G. (2018). Google Maps fragment follow blue dot. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/48312734/google-maps-fragment-follow-blue-dot?rq=1> [Accessed 24 Mar. 2018].
- [16] El Androide Libre. (2018). Enviar un mapa de Google Maps al teléfono, paso a paso. [online] Available at: <https://elandroidelibre.lespanol.com/2016/05/enviar-mapa-google-maps-al-movil.html> [Accessed 12 Mar. 2018].
- [17] Geekytheory.com. (2018). Tutorial Android - 10. Paso de parámetros entre Activities. [online] Available at: <https://geekytheory.com/tutorial-android-10-paso-de-parametros-entre-activities> [Accessed 27 Feb. 2018].
- [18] GitHub. (2018). Allow the user to set map to follow their real-time location · Issue 372 · OneBusAway/onebusaway-android. [online] Available at: <https://github.com/OneBusAway/onebusaway-android/issues/372> [Accessed 25 Apr. 2018].
- [19] GitHub. (2018). Request update to android instructions in README · Issue #20 · react-native-component/react-native-smart-splash-screen. [online] Available at: <https://github.com/react-native-component/react-native-smart-splash-screen/issues/20> [Accessed 13 Mar. 2018].
- [20] Google Developers. (2018). Cámara y vista | Google Maps Android API | Google Developers. [online] Available at: <https://developers.google.com/maps/documentation/android-api/views?hl=es-419> [Accessed 9 Apr. 2018].
- [21] Google Developers. (2018). Primeros pasos | Google Maps Android API | Google Developers. [online] Available at: <https://developers.google.com/maps/documentation/android-api/start?hl=es-419> [Accessed 4 Mar. 2018].
- [22] Gupt, M. (2018). Android Service and BroadcastReceiver Example - Truiton. [online] Truiton. Available at: <http://www.truiton.com/2014/09/android-service-broadcastreceiver-example/> [Accessed 21 Mar. 2018].

- [23] How, T. (2018). Android Studio SQLite Database Multiple Tables Example. [online] Instinct Coder. Available at: <http://instinctcoder.com/android-studio-sqlite-database-multiple-tables-example/> [Accessed 9 Mar. 2018].
- [24] IN, ANDROID, 2018, Android Studio DexIndexOverflowException: method ID not in. Stack Overflow [online]. 2018. [Accessed 6 March 2018]. Available from: <https://stackoverflow.com/questions/38714651/android-studio-dexindexoverflowexception-method-id-not-in>
- [25] latex, U. (2018). Using multicolumn in latex. [online] TeX - LaTeX Stack Exchange. Available at: <https://tex.stackexchange.com/questions/131867/using-multicolumn-in-latex> [Accessed 7 May 2018].
- [26] ListFragment, L. (2018). Long click on ListFragment. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/6732611/long-click-on-listfragment> [Accessed 28 Apr. 2018].
- [27] mapas, A. (2018). Ayuda con aplicación Android y mapas. [online] Stack Overflow en español. Available at: <https://es.stackoverflow.com/questions/10100/ayuda-con-aplicaci%C3%B3n-android-y-mapas> [Accessed 15 Apr. 2018].
- [28] Mapbox. (2018). Build a navigation app for Android | Mapbox. [online] Available at: <https://www.mapbox.com/help/android-navigation-sdk/> [Accessed 29 Mar. 2018].
- [29] Maps, E. (2018). Ensure that the "Google Maps Android API v2" is enabled. I am getting this error when I try to Implement Google Maps. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/40055893/ensure-that-the-google-maps-android-api-v2-is-enabled-i-am-getting-this-error/40056030> [Accessed 27 Feb. 2018].
- [30] Maps, S. (2018). Show blue dots icon on current location Google Maps. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/30440777/show-blue-dots-icon-on-current-location-google-maps/30466050> [Accessed 27 Apr. 2018].
- [31] Mobile.developer.com. (2018). Adding Map Markers and Handling Marker Events in Android Apps - Developer.com. [online] Available at: <https://mobile.developer.com/ws/android/programming/adding-map-markers-and-handling-marker-events-in-android-apps.html> [Accessed 29 Mar. 2018].
- [32] Mohsen Mirhoseini Argi's Personal Blog. (2018). How to launch Google map intent in android app with certain location, zoom and label | Mohsen Mirhoseini's Technical Blog. [online] Available at: <http://mirhoseini.com/how-to-launch-google-map-intent-in-android-app-with-certain-location-zoom-and-label/> [Accessed 27 Mar. 2018].
- [33] Mytrendin.com. (2018). Display current location by a marker using Google Maps API in Android. [online] Available at: <https://www.mytrendin.com/display-current-location-by-a-marker-on-the-map-in-android/> [Accessed 4 Apr. 2018].
- [34] OpenWebinars.net. (2018). Generar apk con Android Studio paso a paso. [online] Available at: <https://openwebinars.net/blog/generar-apk-android-studio/> [Accessed 15 May 2018].

- [35] Overleaf.com. (2018). Overleaf. [online] Available at: <https://www.overleaf.com/15277597xzvpvqkhrxwh> [Accessed 15 May 2018].
- [36] Revelo, J. (2018). Controles: Tutorial De Botones En Android. [online] Hermosa Programación: +50 Tutoriales Desarrollo Android. Available at: <http://www.hermosaprogramacion.com/2016/02/controles-tutorial-botones-android/> [Accessed 3 Apr. 2018].
- [37] Service, B. (2018). Broadcast Receiver within a Service. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/9092134/broadcast-receiver-within-a-service> [Accessed 6 Mar. 2018].
- [38] sgoliver.net. (2018). Mapas en Android – Google Maps Android API (2) | sgoliver.net. [online] Available at: <http://www.sgoliver.net/blog/mapas-en-android-google-maps-android-api-2/> [Accessed 28 Mar. 2018].
- [39] sgoliver.net. (2018). Mapas en Android – Google Maps Android API (3) | sgoliver.net. [online] Available at: <http://www.sgoliver.net/blog/mapas-en-android-google-maps-android-api-3/> [Accessed 2 Apr. 2018].
- [40] Slideshare.net. (2018). Fast-paced Introduction to Android Internals. [online] Available at: <http://www.slideshare.net/hamiltont/fastpaced-introduction-to-android-internals> [Accessed 27 Feb. 2018].
- [41] stop), A. (2018). Android Service needs to run always (Never pause or stop). [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/15758980/android-service-needs-to-run-always-never-pause-or-stop> [Accessed 10 Mar. 2018].
- [42] Tamada, R. and Tamada, R. (2018). Android getting application context in Fragments – Android Quick Tips | AndroidHive. [online] Tips.androidhive.info. Available at: <https://tips.androidhive.info/2013/10/android-getting-application-context-in-fragments/> [Accessed 9 Mar. 2018].
- [43] V., J. and V., J. (2018). Tutorial android 7: Eventos, OnClickListener para implementar un escuchador de eventos en un Boton | Codigoprogramacion. [online] Codigoprogramacion.com. Available at: <http://codigoprogramacion.com/cursos/android/143-tutorial-android-7-eventos-onclicklistener-para-implementar-un-escuchador-de-eventos-en-un-boton.html> [Accessed 8 Apr. 2018].
- [44] YouTube. (2018). Adding dependencies and libraries in Android Studio. [online] Available at: <https://www.youtube.com/watch?v=G27zMPLZm6o> [Accessed 8 Mar. 2018].