



# CheckMyRoute

Alumno: **Álvaro Martínez Arbas**

Plan de estudios: *Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles*

Consultor: **Eduard Matín Lineros**

Fecha de entrega: 6 de Junio de 2018

## A) Creative Commons:



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## B) GNU Free Documentation License (GNU FDL)

Copyright © 2018 Álvaro Martínez Arbas

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

## C) Copyright

© (Álvaro Martínez Arbas)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	CheckMyRoute
<b>Nombre del autor:</b>	Álvaro Martínez Arbas
<b>Nombre del consultor:</b>	Eduard Martín Lineros
<b>Fecha de entrega (mm/aaaa):</b>	06/2018
<b>Área del Trabajo Final:</b>	Aplicación Android Tema propuesto por el alumno
<b>Titulación:</b>	<i>Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles</i>

### Resumen del Trabajo (máximo 250 palabras):

Actualmente hay disponibles aplicaciones móviles con las que es muy fácil viajar y estar informado del tráfico en todo momento. Pero si se quiere compartir el viaje con otro usuario o comprobar en qué momento del mismo se encuentra, es necesario comunicarse con esa persona a través de otra aplicación. Pero cuando alguien viaja, es habitual que haya momentos prolongados en los que no hay suficiente cobertura y queda incomunicada, lo que puede generar que otras personas se preocupen.

Por estas y otras razones, la aplicación CheckMyRoute, disponible para dispositivos Android, ofrece compartir rutas de viaje en tiempo real con otros usuarios. La ventaja que ofrece esta *app* es que cualquier persona puede estar informada cuando un familiar o amigo realiza un viaje por carretera. Además, muestra la última ubicación y los puntos de la ruta en el mapa, junto con el momento exacto en el que esa persona pasó por esos puntos. Así se evitan preocupaciones innecesarias.

Debido a su intuitiva interfaz no resulta difícil aprender a utilizarla. Por ejemplo, para compartir una ruta sólo hay que especificar el origen y el destino del viaje, junto con el número de *ckeckpoints* que se desea registrar y seleccionar los contactos con los que compartirlo. Pero si lo que se desea es ver el viaje de otra persona, la aplicación ofrece el detalle de todas las rutas que compartieron los contactos con el usuario.

Por último, no olvidar que está disponible en Google Play Store de forma totalmente gratuita.

**Abstract (in English, 250 words or less):**

Nowadays, with the huge number of available mobile applications, it is very easy to travel and be informed about traffic at all times. But if you want to share the trip with another user or to check the place where this person is currently located, it is necessary to communicate with that person through another application. But when someone is travelling, it is common not to have connection for a long time, which can cause other people to worry about them.

For these and other reasons, the CheckMyRoute application, available for Android devices, offers you to share the route of your trips in real time with other users. The advantage offered by this app is that anyone can be informed when a relative or friend makes a road trip. In addition, it shows the last location and checkpoints of the route on a map, along with the exact moment in which that person went through those points. This advantage avoids unnecessary worries.

Due to its intuitive interface it is not difficult to learn how to use it. For example, to share a route you only have to specify the origin and destination of the trip, along with the number of checkpoints that you want to register and select the contacts you want to share your route with. But if what you want is to see another person's trip, the application offers the detail of all the routes that your contacts shared with you.

Finally, do not forget that it is available in the Google Play Store for free.

**Palabras clave (entre 4 y 8):**

Ruta, viaje, checkpoint, compartir, control, carretera, mapa

## Índice

<b>1. Introducción</b> .....	1
1.1 Contexto y justificación del Trabajo .....	1
1.2 Objetivos del Trabajo.....	2
1.3 Enfoque y método seguido.....	3
1.4 Planificación del Trabajo .....	4
1.5 Breve resumen de productos obtenidos .....	5
1.6 Breve descripción de los otros capítulos de la memoria.....	6
<b>2. Análisis de usuarios</b> .....	7
2.1 Perfil de usuario .....	7
2.2. Necesidades a cubrir.....	7
2.3 Contextos de uso .....	8
2.4 Escenarios de uso de la aplicación .....	8
<b>3. Casos de uso</b> .....	10
<b>4. Diseño de la aplicación</b> .....	19
4.1 Árbol de navegación.....	19
4.2 Prototipo de alto nivel .....	19
<b>5. Arquitectura del sistema</b> .....	35
5.1 Diseño físico .....	35
5.2 Servidor .....	36
5.3 Cliente .....	38
5.4 Google APIs .....	40
5.5 GPS.....	41
5.6 Arquitectura de la aplicación .....	41
<b>6. Desarrollo</b> .....	44
6.1 Diagrama de clases.....	44
<b>7. Pruebas</b> .....	61
7.1 SharedWithMeTest.....	61
7.2 MyRoutesTest .....	62
<b>8. Conclusiones</b> .....	63
8.1 Lecciones aprendidas.....	63
8.2 Objetivos alcanzados .....	63
8.3 Seguimiento de la planificación. ....	64
8.4 Líneas de trabajo futuro.....	64
<b>9. Glosario</b> .....	65
<b>10. Bibliografía</b> .....	66
10.1 Referencias .....	66
10.2 Documentación .....	68

**Lista de figuras**

Figura 1: Diagrama de Gantt	5
Figura 2: Caso de uso – Crear cuenta de usuario	10
Figura 3: Caso de uso – Login en la aplicación	11
Figura 4: Caso de uso – Acceder al menú	12
Figura 5: Caso de uso – Acceder a mis rutas	12
Figura 6: Caso de uso – Acceder a rutas compartidas	13
Figura 7: Caso de uso – Acceder a contactos	13
Figura 8: Crear nueva ruta	14
Figura 9: Caso de uso – Mostrar detalles de una ruta	15
Figura 10: Caso de uso - Cambiar email	16
Figura 11: Caso de uso – Acceder a la ayuda	16
Figura 12: Caso de uso – Finalizar ruta	17
Figura 13: Caso de uso – Eliminar ruta	17
Figura 14: Caso de uso - Logout	18
Figura 15: Caso de uso – Reestablecer contraseña	18
Figura 16: Árbol de navegación	19
Figura 17: Pantalla de Login	20
Figura 18: Pantalla para crear una nueva cuenta	21
Figura 19: Pantalla para reestablecer contraseña	22
Figura 20: Menú de navegación	23
Figura 21: Pantalla de rutas de usuario	24
Figura 22: Pantalla de detalle de una ruta	25
Figura 23: Pantalla del mapa de una ruta	26

---

Figura 24: Pantalla de rutas compartidas y detalle de una ruta	27
Figura 25: Pantalla crear ruta – Origen y destino	28
Figura 26: Pantalla crear ruta – Checkpoints y contactos	29
Figura 27: Pantalla crar ruta - Mensajes	30
Figura 28: Pantallas contactos y ayuda	31
Figura 29: Pantalla cambiar email	32
Figura 30: Finalizar ruta activa	33
Figura 31: Eliminar ruta	34
Figura 32: Diseño Físico	35
Figura 33: Base de datos en Firebase	38
Figura 34: Base de datos Sugar ORM	40
Figura 35: Patrón Model – View - Controller	41
Figura 36: Diagrama de clase – SingleNavigationActivity.java	44
Figura 37: Diagrama de clase – LoginActivity.java – ForgottenActivity.java	45
Figura 38: Diagrama de clase – NewAccountActivity.java	46
Figura 39: Diagrama de clase – MyRoutesActivity.java - MyRoutesFragment.java	47
Figura 40: Diagrama de clase – MyDetailActivity.java – MyDetailFragment.java	48
Figura 41: Diagrama de clase – MyMapActivity.java – MapFragment.java	49
Figura 42: Diagrama de clase – SharedWithMeActivity.java – SharedWithMeFragment.java	50
Figura 43: Diagrama de clase – SharedDetailActivity.java – SHaredDetailFragment.java	51
Figura 44: Diagrama de clase – NewRouteActivity.java – NewMapFragment.java	52
Figura 45: Diagrama de clase – ContactsNewActivity.java – ContactsNewFragment.java	53



Figura 46: Diagrama de clase – ContactsMenuActivity.java – ContactsMenuFragment.java	54
Figura 47: Diagrama de clase – ChangeEmailActivity.java – ChangeEmailFragment.java – HelpActivity.java – HelpFragment.java	55
Figura 48: Diagrama de clase – SwipeHelperActivity.java	56
Figura 49: Diagrama de clase – Route.java – Checkpoint.java	57
Figura 50: Diagrama de clase – RouteFactory.java – LocalContacts.java	58
Figura 51: Diagrama de clase – FetchUrl.java – DataParserJSON.java – DataRequestConnection.java	59
Figura 52: Diagrama de clase – DrawRoute.java – DrawRouteMaps.java – DrawMarker.java - RouteDrawerTask	60

# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

En la actualidad, los dispositivos móviles se han convertido en una herramienta indispensable para la mayoría de las personas por las múltiples funcionalidades que ofrecen. Gracias a estos dispositivos, las personas pueden comunicarse al instante con otras personas, ya sea por llamadas de voz, por mensajes, o por otras plataformas disponibles, como las redes sociales. Pero su uso va mucho más allá.

Considerando sólo algunas de las opciones disponibles, los dispositivos móviles permiten consultar cualquier información al instante, monitorizar la actividad física o guiar a una persona cuando se dispone a realizar un desplazamiento. En este último caso, los teléfonos móviles incluyen un dispositivo GPS que les permite estar geolocalizados en todo momento. Son muchas las aplicaciones que aprovechan este dispositivo para ofrecer funcionalidades que mejoran la vida de las personas. El mejor ejemplo es la aplicación de "Google Maps", quizá la más extendida a la hora de realizar viajes o simplemente desplazarse por la ciudad. Se ha convertido en una herramienta indispensable para orientar y guiar a las personas en sus desplazamientos. Pero no sólo eso, sino que, además, permite conocer el estado del tráfico o innumerables sitios de interés que vienen marcados en sus mapas.

Aunque Google incluye cada vez más funcionalidades a su aplicación de mapas [1], como compartir en tiempo real la ubicación del usuario durante la navegación, el consumo excesivo de datos y batería, junto con la falta de cobertura durante un viaje, puede llevar a que la persona que comparte su ubicación cierre la aplicación y se olvide de volver a abrirla, lo que puede producir que las personas con las que compartía su ubicación se preocupen por si a esa persona le ha podido pasar algo durante el viaje, cuando simplemente es que se olvidó de abrir la aplicación de nuevo.

Para evitar preocupaciones innecesarias sería conveniente que los usuarios de la aplicación puedan visualizar en su mapa ciertos puntos de control por los que pasa la persona que viaja y la hora aproximada a la que lo hizo, además de su última ubicación. Aunque pueda parecer lo contrario, esto no se traduce en un uso excesivo de datos y batería, pues la aplicación puede trabajar en segundo plano y actualizar la ubicación cada cierto intervalo de tiempo, reduciendo considerablemente el consumo de datos y de batería.

Aunque pueda parecer una funcionalidad demasiado obvia para la situación actual de las aplicaciones de navegación, aun no existe en el mercado ninguna aplicación que ofrezca este servicio.

Por otra parte, las empresas de reparto también se podrían beneficiar de esta aplicación, ya que monitorizarían en tiempo real la ruta y puntos de control que

siguen sus repartidores. Aunque actualmente tienen a su disposición servicios parecidos, se ofrecen a un coste elevado y con la necesidad de instalar un dispositivo adicional en sus vehículos.

## **1.2 Objetivos del Trabajo**

El objetivo principal de este Trabajo de Fin de Máster es poner a disposición de los usuarios de un dispositivo Android una aplicación gratuita que les permita compartir su viaje y los puntos intermedios por los que pasan, en tiempo real con otros usuarios, sin que el consumo de datos y batería sea excesivo.

Para alcanzar este objetivo se proponen otros objetivos más específicos en los que se basan las tareas para el desarrollo de proyecto:

- Diseñar la aplicación e interfaz adecuadas para facilitar su uso y que sea atractiva para el usuario final.
- Que el usuario sea capaz de crear la ruta, con un punto de inicio y otro de destino, y establecer y editar los puntos de control que considere oportunos.
- Que el usuario pueda cancelar en cualquier momento la ruta activa que esté compartiendo.
- Que el usuario que comparte su ruta pueda visualizar en cualquier momento en el mapa, la ruta, los puntos de control y la posición de la última vez que se actualizó su posición. Que al pulsar sobre un punto de control aparezca el detalle de la hora a la que pasó por ese punto.
- Que los usuarios con los que se comparte una ruta activa puedan visualizar en cualquier momento en el mapa, la ruta, los puntos de control y la posición de la última vez que se actualizó la posición del usuario que comparte su viaje. Que al pulsar sobre un punto de control aparezca el detalle de la hora a la que el usuario que comparte su ruta pasó por ese punto.
- Facilitar a los usuarios un historial de las rutas que compartieron y que fueron compartidas con ellos. Además de poder borrar registros de estos historiales.
- Facilitar la forma de compartir la ruta con otros usuarios a partir de la lista de contactos que guarde el usuario en su dispositivo. Sólo estarán disponibles los contactos que estén dados de alta en la aplicación.
- Facilitar el acceso a la aplicación con una cuenta de usuario y una contraseña. La primera vez que el usuario abre la aplicación tendrá que rellenar un pequeño formulario que será validado para que pueda logarse en la aplicación.

- Optimizar el uso de datos y de la batería del dispositivo sin pérdida de rendimiento y fiabilidad de la aplicación.
- Disponer de la aplicación, de forma gratuita, en la tienda oficial de aplicaciones para dispositivos Android, "Google Play Store" [2].

### 1.3 Enfoque y método seguido

Aunque en el mercado existen numerosas aplicaciones relacionadas con la navegación y uso de mapas, además de ofrecer innumerables funcionalidades que pueden resultar complejas para usuarios inexpertos, uno de los objetivos principales de la aplicación objeto de este TFM es que su diseño y uso sean lo suficientemente simples para que el usuario tenga que hacer el menor esfuerzo posible para ser capaz de comprender y utilizar la aplicación.

La aplicación hará uso de los datos de acceso a internet y del GPS del dispositivo móvil para funcionar. Además, solicitará permiso para acceder a la lista de contactos del usuario.

Gracias a la API de "Google Maps", la aplicación podrá mostrar en todo momento el mapa, la ruta y los puntos de control, a los usuarios de la ruta activa que compartan o de las rutas activas que se comparten con ellos.

Gracias al servicio "Google Firebase" [3], la aplicación podrá transmitir información entre el usuario que comparte la ruta y los usuarios con los que se comparte. La aplicación se encargará de actualizar los puntos de control y la última posición del usuario y transmitirá esta información a "Firebase". Será la aplicación la que solicite a "Firebase" la información necesaria para actualizar datos de las rutas activas que se comparten. Así dispondrá en todo momento, en el dispositivo del usuario, de toda la información necesaria para "pintar" en el mapa los puntos de control actualizados. De esta forma, cada vez que la aplicación del usuario actualiza su posición o los puntos de control por los que pasa, se transmite la información al resto de usuarios para que en sus dispositivos aparezcan los puntos de control y la última posición actualizados. Es este servicio el que actúa de base de datos remota de la aplicación.

También es importante considerar que la aplicación tendrá que hacer uso de la memoria del dispositivo para guardar la información del usuario y de las rutas activas que se comparten con él, de la ruta activa que comparta y del historial de las rutas. Para ello se hará uso de la librería "Sugar ORM" [4].

La aplicación se desarrolla en el lenguaje de programación Java a través del entorno de desarrollo Android Studio.

## 1.4 Planificación del Trabajo

*Febrero 2018*

- Matriculación
- Búsqueda y análisis de proyectos
- Propuesta TFM

*Marzo 2018 – Abril 2018*

- PEC1 – Plan de Trabajo – 07/03/2018 - 14/03/2018 (20 horas)
  1. Contexto y justificación del trabajo. (5 horas)
  2. Definición de objetivos. (5 horas)
  3. Enfoque y método seguido. (5 horas)
  4. Planificación de las tareas a realizar. (5 horas)
- PEC2 – Diseño – 15/03/2018 - 04/04/2018 (40 horas)
  1. Definición de casos de uso. (5 horas)
  2. Diseño de la arquitectura. (20 horas)
    - Diagrama UML del diseño de la base de datos.
    - Diagrama UML de clases y métodos
    - Diagrama de la arquitectura del sistema
    - Flujo de la aplicación: Activities.
  3. Navegaciones y prototipado: (15 horas)
    - Diagrama UML de pantallas y navegación horizontal.
    - Prototipo en alta definición de todas las pantallas que incluya la navegación completa entre ellas.

*Abril 2018 – Mayo 2018*

- PEC3 – Desarrollo – 05/04/2018 - 16/05/2018 (120 horas)
  1. Implementación. (90 horas)
    - Implementación del diseño de la aplicación: diseño de todas las pantallas.
    - Definición de las estructuras de datos.
    - Estudio e implementación de las comunicaciones con la base de datos en el servidor: servicio FireBase.
    - Estudio e implementación de las comunicaciones con la base de datos del dispositivo: Librería SugarORM.
    - Estudio e implementación del manejo de mapas a través de la API de Google Maps.
    - Implementación de la sincronización y flujo.
  2. Pruebas. (28 horas)

- Implementación de pruebas para comprobar el funcionamiento de la aplicación a través de la librería Espresso.

### 3. Publicación de la aplicación en Google Play Store (2 horas)

Mayo 2018 – Junio 2018

- PEC4 – Entrega final – 17/05/2018 - 06/06/2018 (50 horas)
  1. Memoria. (30 horas)
  2. Presentación y vídeo. (15 horas)
  3. Preparar ficheros de la entrega final. (5 horas)

Junio 2018

- Defensa ante tribunal virtual – 18/06/2018 - 22/06/2018

Diagrama de Gantt

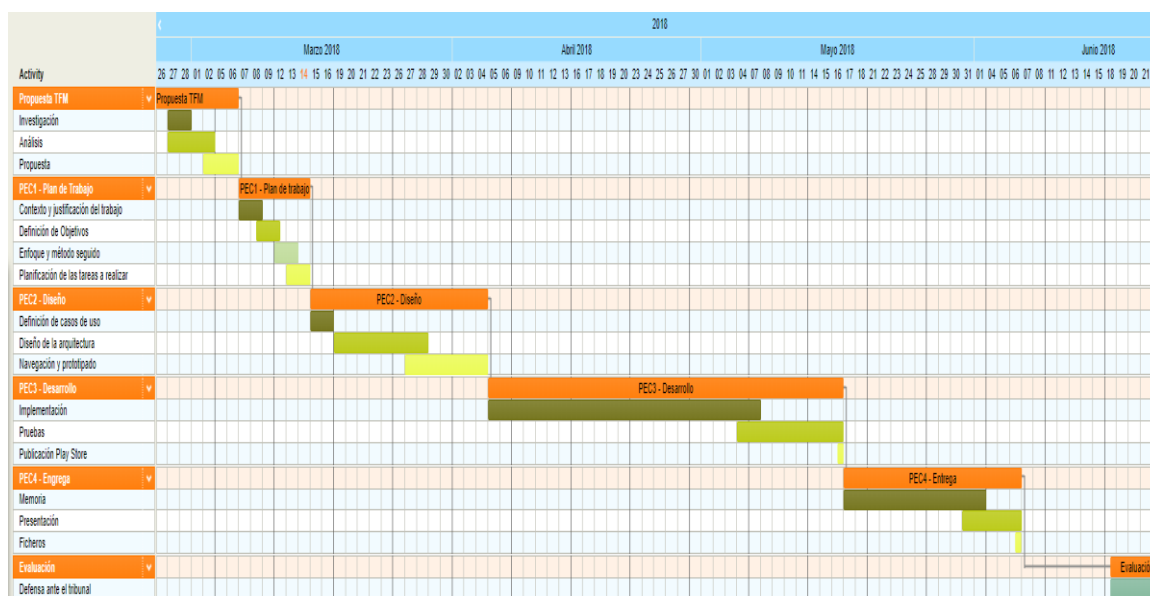


Figura 1: Diagrama de Gantt

## 1.5 Breve resumen de productos obtenidos

Documentos oficiales para la Universitat Oberta de Catalunya:

- PEC1 – Plan de trabajo.
- PEC2 – Diseño conceptual y en alta definición de la aplicación. Memoria (progreso hasta el momento).
- PEC3 – Aplicación (apk), código fuente y pruebas.

Memoria (progreso hasta el momento).

- PEC4 – Memoria final. Vídeo presentación. Aplicación (apk). Código fuente y pruebas. Manual de usuario.

Publicación de la aplicación en la tienda oficial Google Play Store.

## **1.6 Breve descripción de los otros capítulos de la memoria**

Capítulo 1: Planificación del trabajo y de las tareas a realizar

Capítulo 2: Análisis de los usuarios. Analiza los usuarios finales y escenarios de uso de la aplicación.

Capítulo 3: Casos de uso. Se definen todas las funcionalidades de la aplicación.

Capítulo 4: Diseño de la aplicación. Muestra el prototipo de alto nivel de las pantallas de la aplicación tal y como se ven en el producto final.

Capítulo 5: Arquitectura del sistema. Se define el diseño de la arquitectura de la aplicación, los servicios que utiliza para gestionar la información y la estructura de las bases de datos en servidor y en cliente.

Capítulo 6: Desarrollo. Muestra el diagrama de clases y los métodos que implementa la aplicación.

Capítulo 7: Pruebas. Se definen y analizan las pruebas llevadas a cabo para comprobar el funcionamiento de la aplicación.

Capítulo 8: Conclusiones. Resumen de las lecciones aprendidas, objetivos alcanzados y líneas futuras de desarrollo.

Capítulo 9: Glosario. Definición de términos utilizados en el proyecto que pueden no ser conocidos por un público general.

Capítulo 10: Bibliografía. Referencias y documentación consultada para el desarrollo del proyecto.

## 2. Análisis de usuarios

En este capítulo se aborda el análisis del diseño de la aplicación basado en el perfil, necesidades y contexto de uso de los usuarios finales de la aplicación. Además, se elaboran varios escenarios que sirven de ejemplo de uso de la aplicación.

### 2.1 Perfil de usuario

Al tratarse de una aplicación para dispositivos Android, las personas que harán uso de la aplicación son las que están acostumbradas a la utilización de este tipo de dispositivos. Aunque en un principio se puede pensar que son las personas más jóvenes las que suelen utilizar estos dispositivos, actualmente su uso se extiende a personas de mediana edad y, cada vez más, a personas de avanzada edad. Por lo tanto, se puede considerar que cualquier persona con un dispositivo Android está capacitada para utilizar esta aplicación.

Por otra parte, al tratarse de una aplicación cuya temática es la monitorización de un viaje, principalmente por carretera, se pueden distinguir claramente dos tipos de usuario de la aplicación.

- **Perfil 1: personas que viajan con frecuencia.** Son las que comparten su viaje con otros usuarios. Dentro de este perfil encajan personas que viajen con cierta frecuencia, sin importar el nivel de estudios. Lo que sí marca una segmentación dentro de este perfil es el motivo del viaje: por ocio/vacaciones o por motivos profesionales/negocios. Casi todas las personas en España, sean de la edad que sean, suelen viajar más en vacaciones. Pero la frecuencia de los viajes de este tipo es claramente mayor en personas jóvenes. Por otra parte, las personas que viajan por motivos profesionales son los transportistas de mercancías o las personas de negocios/vendedores.
- **Perfil 2: personas que se interesan por el estado del viaje que realizan otras personas.** Son las que visualizan el viaje de otros usuarios. En este perfil encajan personas no tan jóvenes, habitualmente personas con hijos en edad de viajar de forma autónoma, pero también personas que tienen alguna empresa de transporte con empleados a su cargo.

### 2.2. Necesidades a cubrir

Las personas del primer perfil necesitan informar en tiempo real de la hora exacta por la que pasan por ciertos puntos a lo largo de su viaje, sin necesidad de estar pendientes de su teléfono móvil, puesto que pueden estar realizando otra actividad, como conducir un vehículo. Además, deben tener disponible en



todo momento la información detallada de la ruta que llevan a cabo, así como su posición actual. Por motivos profesionales es posible que necesiten tener un histórico de las rutas que realizan.

Las personas del segundo perfil necesitan estar informadas en tiempo real de la hora exacta de los puntos por los que pasa otra persona al realizar un viaje, además de la información de la última posición de esa persona. En todo momento es necesario que puedan visualizar el histórico de los viajes que otras personas realizaron.

En ambos perfiles, si una persona cambia de dispositivo es necesario que pueda acceder a los datos que tenía guardados en la aplicación desde el nuevo dispositivo.

## 2.3 Contextos de uso

Siguiendo los perfiles de usuario y necesidades descritos en los 2 apartados anteriores, se pueden definir varios contextos de uso de la aplicación.

Habitualmente los usuarios que comparten su ruta harán uso de la aplicación desde el vehículo desde el que realizan el viaje. Principalmente son viajes por carretera. Pero también se puede considerar que una persona haga el camino a pie o en bicicleta.

Por otra parte, el uso de la aplicación por parte de las personas que monitorizan el viaje de otra se puede realizar desde cualquier parte, ya sea desde su casa, el trabajo o desde un lugar público.

En todo caso el único requisito para poder hacer uso de la aplicación es que se tenga conexión a internet y, sólo en caso de las personas que comparten su ruta, la función de GPS activada.

El tiempo que el usuario destina a la aplicación varía en función del tipo de uso que se haga. Si se comparte una ruta, es posible que se mantenga la aplicación abierta durante la duración del viaje, aunque no es necesario. Para una persona que monitoriza la ruta de otra, abrirá la aplicación de forma intermitente para echar un vistazo.

## 2.4 Escenarios de uso de la aplicación

### 2.4.1 Escenario 1: Estudiante en viaje de fin de curso

Carlos Muñoz, de 17 años, es un estudiante de 1º de bachillerato que acaba de terminar el curso y está de viaje con sus compañeros de clase. Después de dos semanas de exámenes y mucho estrés, por fin le dieron la última nota, Matemáticas, aunque no las tenía todas consigo ha sacado un 6, por lo que no tendrá que recuperar ninguna asignatura en Septiembre y podrá disfrutar de unas merecidas vacaciones.

Los padres de Carlos, María y Julián, un matrimonio de clase media de Madrid, están preocupados por el viaje de Carlos. Es la primera vez viaja sin ellos y no se fían del conductor del autobús que los está llevando hasta Lisboa. Al ser un viaje por carretera y de tantas horas les preocupa que puedan tener un accidente. Pero Carlos, entusiasta de la tecnología y, como casi todos los adolescentes, no se separa de su teléfono móvil. Para tranquilizar a sus padres, les hizo descargarse una aplicación para que supieran en todo momento la ruta que sigue el autobús y el punto en el que se encuentran.

A las 09:00 de la mañana Carlos salió del instituto rumbo a Lisboa. Aunque María y Julián sienten cierta tristeza por ver cómo su hijo se hace mayor, Carlos no se olvidó de compartir su viaje con ellos. Menos mal que pueden ver en el móvil que a las 11:15 pasó por Trujillo y que, ahora mismo, a las 13:30, acaban de pasar por Badajoz. Aunque la angustia es un poco menor, no se quedarán tranquilos hasta que llegue a Lisboa en un par de horas.

### **2.4.2 Escenario 2: Propietario de empresa de transportes**

Fernando Estévez, de 45 años, es un empresario de Madrid que tiene una empresa de transporte de mercancías. No le va mal, ya que el último año pudo comprar el quinto camión de la compañía y contratar a otro empleado más.

Para Fernando es muy importante que cada camión llegue a su destino a la hora planificada, aunque bien sabe que muchas veces no depende de él ni de sus camioneros llegar a tiempo. Casi siempre hay imprevistos que no puede controlar. Pero lo que más le estresaba es que no sabía dónde estaban sus camiones ni si habían alcanzado ya ciertos puntos del viaje. Nervioso, solía llamar a cada uno de sus camioneros para que le dieran explicaciones y, si no conseguía hablar con alguno su nerviosismo alcanzaba límites preocupantes.

Son las 18:00 horas y sus camiones ya deberían haber llegado a su destino. Fernando mira su teléfono y observa que 3 de ellos han llegado a la hora planificada. Sin embargo, hay 2 camiones que aún no han llegado. Empieza a ponerse nervioso. Abre de nuevo la aplicación para ver qué ha pasado y se queda un poco más tranquilo al ver que los dos camiones han pasado por todos los puntos de control y sólo les queda llegar al destino. Como la ruta de ambos camiones era similar, observa que en ambos casos se produjo el retraso entre el cuarto y el quinto punto de control, debe ser que había algo de tráfico. Fernando guarda su teléfono y ya, algo más tranquilo, se va a casa después de una larga jornada.

### 3. Casos de uso

En este capítulo se describe con detalle todos los casos de uso [5] o funcionalidades que el usuario puede llevar a cabo en la aplicación.

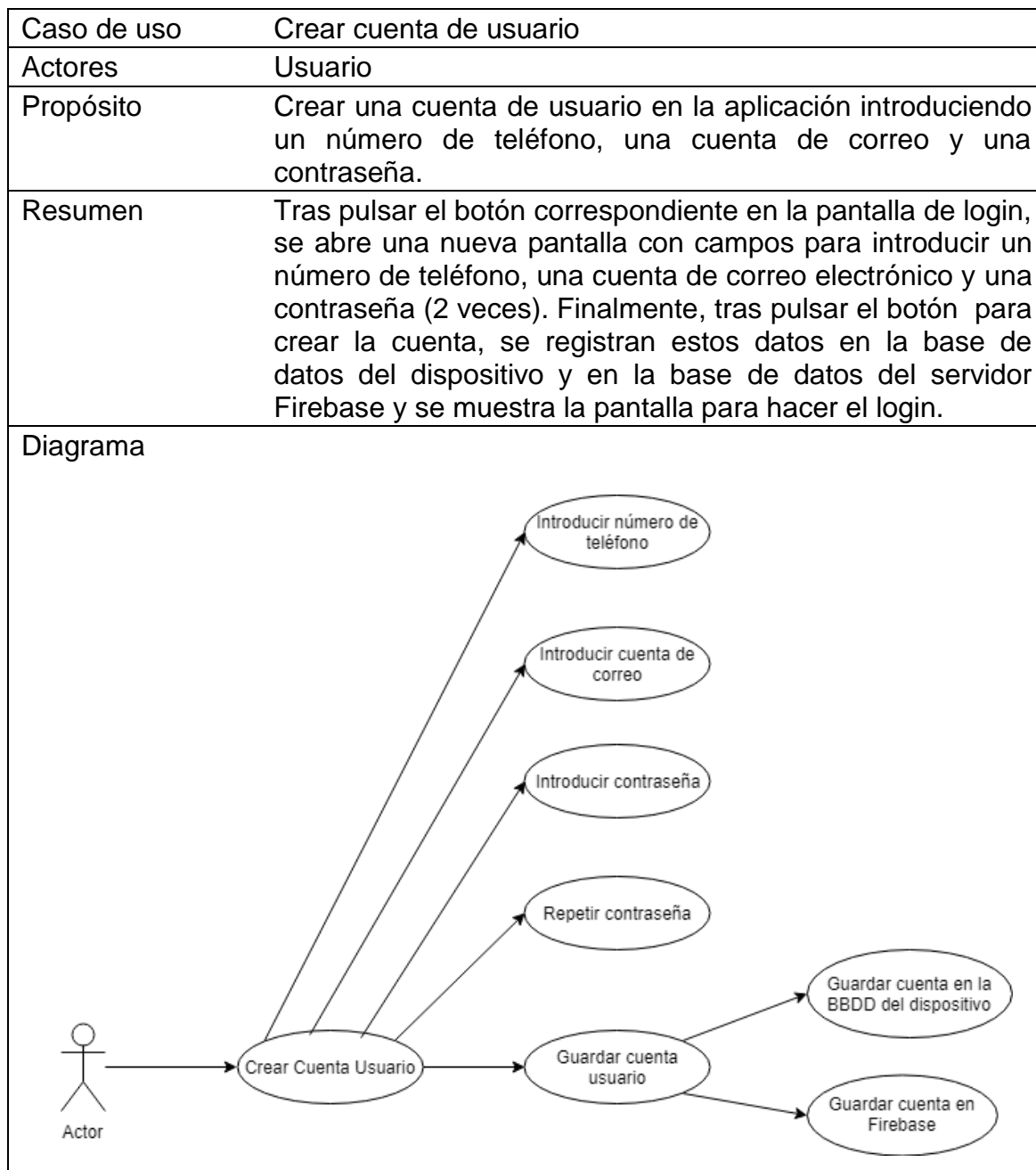


Figura 2: Caso de uso – Crear cuenta de usuario

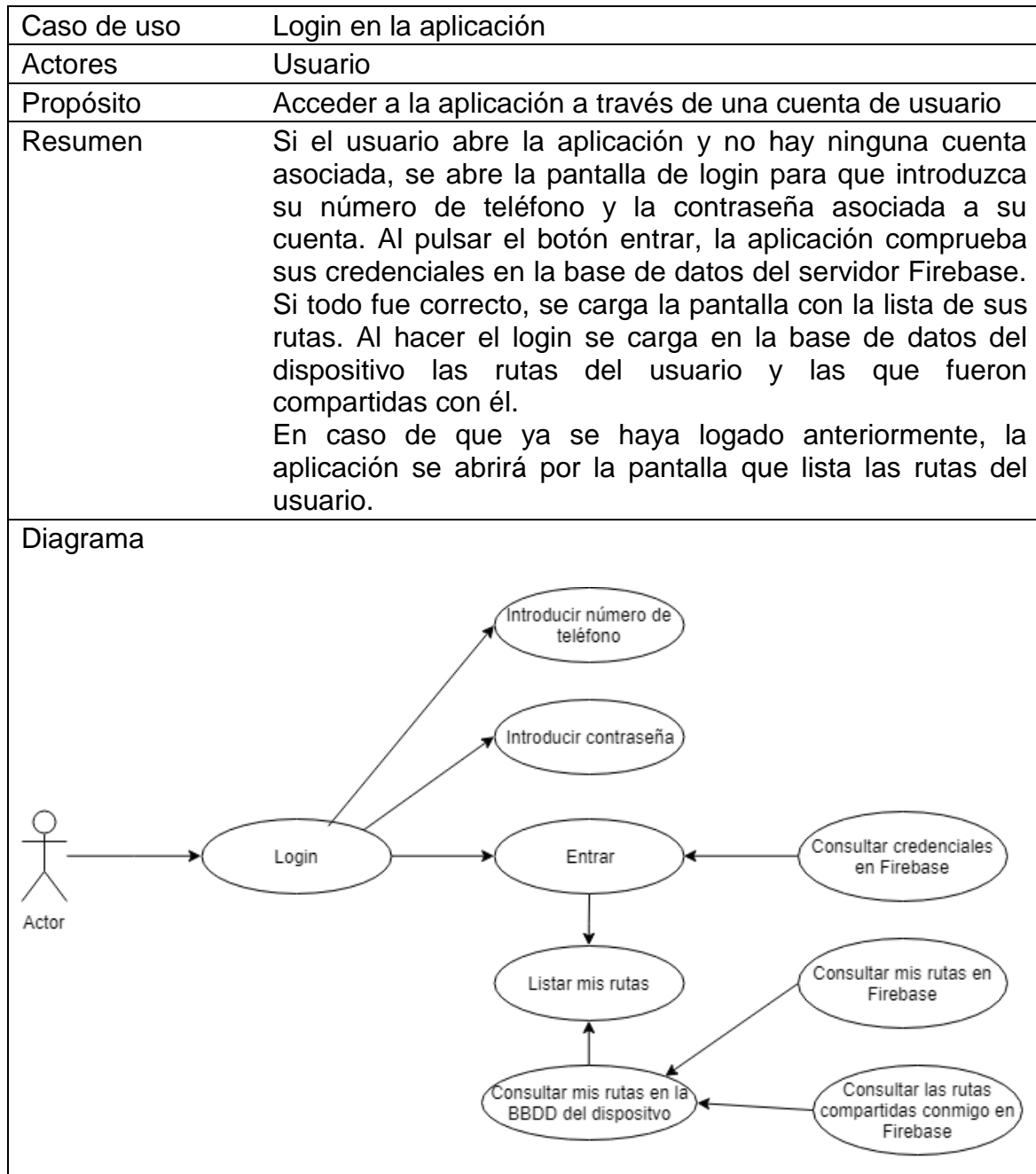


Figura 3: Caso de uso – Login en la aplicación

Caso de uso	Acceder al menú de la aplicación
Actores	Usuario
Propósito	Acceder a las pantallas disponibles en la aplicación desde un menú.
Resumen	El usuario puede acceder a cualquier pantalla de la aplicación desde la pantalla en la que se encuentre (salvo pantallas de detalle y mapa). Desde casi todas las pantallas está disponible el botón del menú. Al pulsarlo, se despliega un menú lateral donde puede seleccionar la pantalla a la que desea moverse.
Diagrama	<p>The diagram shows an actor labeled 'Actor' with an arrow pointing to a central use case 'Acceder al menú'. From this central use case, seven arrows point to other use cases: 'Acceder a mis rutas', 'Acceder a rutas compartidas conmigo', 'Crear nueva ruta', 'Acceder a contactos', 'Acceder a la ayuda', 'Acceder a cambiar contraseña', and 'Hacer logut'.</p>

Figura 4: Caso de uso – Acceder al menú

Caso de uso	Acceder a mis rutas
Actores	Usuario
Propósito	Disponer de una lista con todas las rutas del usuario
Resumen	En esta pantalla se cargan las rutas que el usuario ha creado en la aplicación. La aplicación carga las rutas del usuario desde la base de datos del dispositivo. Cada entrada de la lista muestra el destino, mes y año del viaje.
Diagrama	<p>The diagram shows an actor labeled 'Actor' with an arrow pointing to a use case 'Acceder a mis rutas'. An arrow points from 'Acceder a mis rutas' to 'Listar mis rutas', which in turn points to 'Consultar mis rutas en la BBDD del dispositivo'.</p>

Figura 5: Caso de uso – Acceder a mis rutas

Caso de uso	Acceder a rutas compartidas con el usuario
Actores	Usuario
Propósito	Disponer de una lista con todas las rutas compartidas con el usuario por parte de sus contactos.
Resumen	En esta pantalla se cargan las rutas compartidas con el usuario. La aplicación carga la lista con las rutas compartidas con el usuario desde la base de datos del dispositivo. Cada entrada de la lista muestra el contacto que comparte la ruta, el destino, el mes y el año.
Diagrama	<pre> graph LR     Actor((Actor)) --&gt; UC1([Acceder a rutas compartidas conmigo])     UC1 --&gt; UC2([Listar rutas compartidas conmigo])     UC3([Consultar rutas compartidas conmigo en la BBDD del dispositivo]) --&gt; UC2             </pre>

**Figura 6: Caso de uso – Acceder a rutas compartidas**

Caso de uso	Acceder a contactos
Actores	Usuario
Propósito	Disponer de una lista de los contactos disponibles en la aplicación con los que compartir una ruta.
Resumen	En esta pantalla se carga la lista de contactos del usuario que están dados de alta en la aplicación. Esta lista de contactos se carga desde la base de datos del dispositivo.
Diagrama	<pre> graph LR     Actor((Actor)) --&gt; UC1([Acceder a Contactos])     UC1 --&gt; UC2([Listar contactos disponibles])     UC3([Consultar contactos en la BBDD del dispositivo]) --&gt; UC2             </pre>

**Figura 7: Caso de uso – Acceder a contactos**

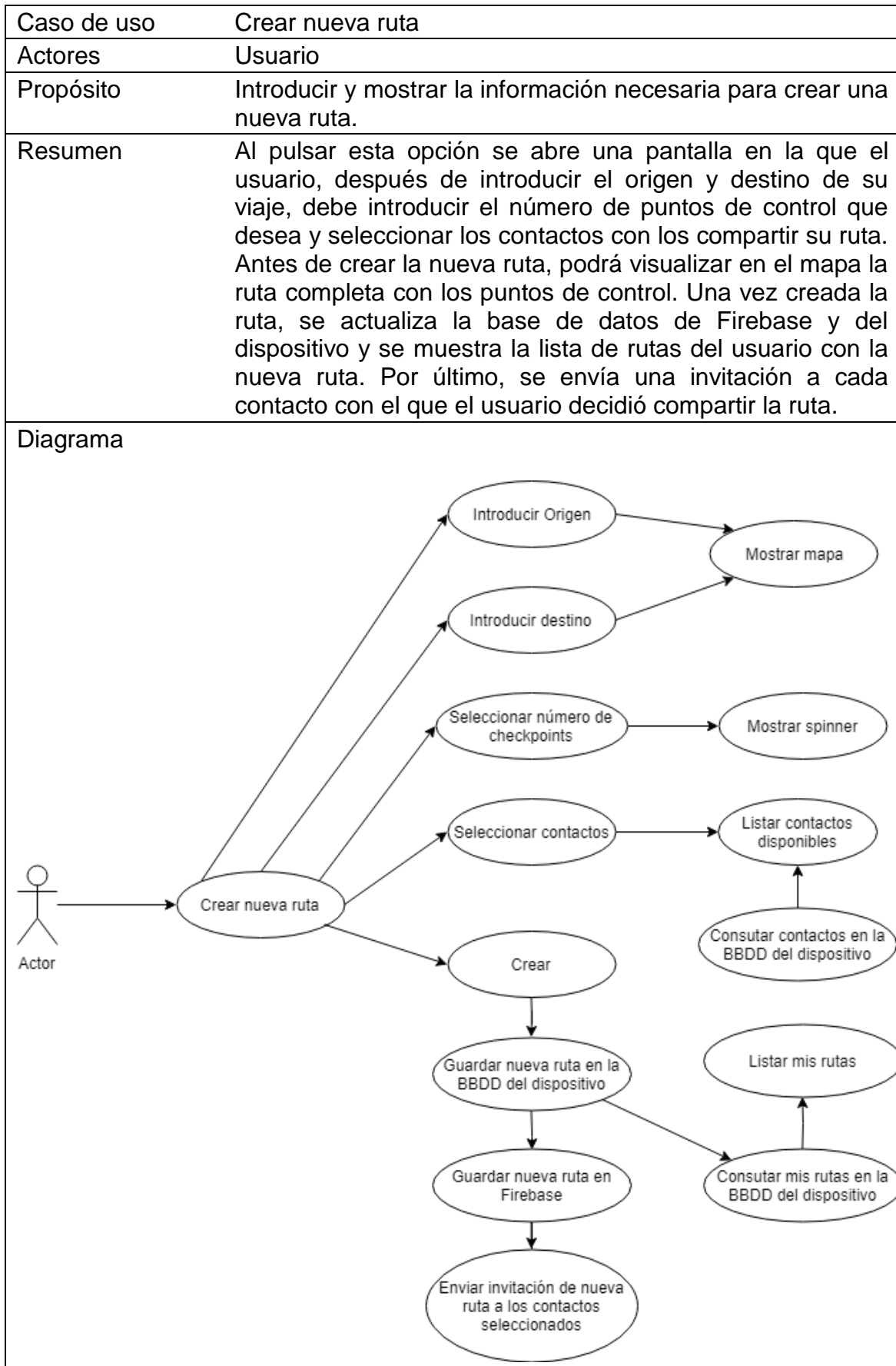


Figura 8: Crear nueva ruta

Caso de uso	Mostrar detalles de una ruta
Actores	Usuario
Propósito	Ver el detalle completo de una ruta.
Resumen	Desde la lista de rutas, tanto las del usuario como las compartidas, el usuario puede ver el detalle de una ruta pulsando sobre la entrada en la lista de esa ruta. Se abre una pantalla que muestra la información de la ruta (origen, destino, distancia total, fecha y hora de inicio y fin y número de puntos de control), una lista con los puntos de control (que muestra la distancia desde el origen, la ciudad, la hora y la fecha por la que se pasó por ese punto) y la lista de los contactos con los que se comparte la ruta (en caso de que el usuario comparta la ruta) o el contacto que la comparte (en caso de que sea una ruta compartida por otro usuario). También está disponible un botón para mostrar una nueva pantalla en la que aparece el mapa con la ruta y los puntos de control marcados.
Diagrama	<pre> graph TD     Actor[Actor] --&gt; UC1[Mostrar detalles de una ruta]     UC1 --&gt; UC2[Información de la ruta]     UC1 --&gt; UC3[Lista de checkpoints]     UC1 --&gt; UC4[Lista de contactos]     UC1 --&gt; UC5[Mostrar mapa]     UC5 --&gt; UC6[Mapa]     UC7[Consultar base de datos del dispositivo] --&gt; UC2     UC7 --&gt; UC3     UC7 --&gt; UC4     UC7 --&gt; UC6     </pre> <p>The diagram is a UML Use Case Diagram. It features an actor labeled 'Actor' on the left. An arrow points from the actor to the use case 'Mostrar detalles de una ruta'. From this use case, four arrows point to 'Información de la ruta', 'Lista de checkpoints', 'Lista de contactos', and 'Mostrar mapa'. From 'Mostrar mapa', an arrow points to 'Mapa'. On the right side, a use case 'Consultar base de datos del dispositivo' has four arrows pointing to 'Información de la ruta', 'Lista de checkpoints', 'Lista de contactos', and 'Mapa'.</p>

Figura 9: Caso de uso – Mostrar detalles de una ruta



Caso de uso	Acceder a cambiar el email
Actores	Usuario
Propósito	Cambiar el email asociado a una cuenta de usuario
Resumen	En esta pantalla el usuario introduce la nueva dirección de email que se asocia a su cuenta en la aplicación.
Diagrama	<pre> graph LR     Actor[Actor] --&gt; UC1([Acceder a cambiar email])     UC1 --&gt; UC2([Cambiar])     UC2 --&gt; UC3([Cambiar email en Firebase])     UC2 --&gt; UC4([Cambiar email en la BBDD del dispositivo])             </pre> <p>The diagram shows a stick figure actor labeled 'Actor' with an arrow pointing to an oval use case 'Acceder a cambiar email'. From this use case, an arrow points to another oval use case 'Cambiar'. From 'Cambiar', two arrows branch out to two more oval use cases: 'Cambiar email en Firebase' and 'Cambiar email en la BBDD del dispositivo'.</p>

Figura 10: Caso de uso - Cambiar email

Caso de uso	Acceder a la ayuda
Actores	Usuario
Propósito	Aportar la información al usuario en caso de duda.
Resumen	En esta pantalla se carga la información referida a cada opción del menú de la aplicación.
Diagrama	<pre> graph LR     Actor[Actor] --&gt; UC1([Acceder a la ayuda])     UC1 --&gt; UC2([Cargar ayuda])             </pre> <p>The diagram shows a stick figure actor labeled 'Actor' with an arrow pointing to an oval use case 'Acceder a la ayuda'. From this use case, an arrow points to another oval use case 'Cargar ayuda'.</p>

Figura 11: Caso de uso – Acceder a la ayuda

Caso de uso	Finalizar ruta
Actores	Usuario
Propósito	Finalizar una ruta activa
Resumen	Desde la lista de las rutas del usuario, en caso de que el usuario tenga una ruta activa (en ese momento comparte en tiempo real una ruta) y desea finalizar el viaje, tiene disponible esta opción al desplazar hacia la izquierda la entrada de la ruta en la lista. Aparece el botón correspondiente para finalizar el viaje y, en ese momento, ya no se actualizarán más los datos de su posición ni de los checkpoints de esa ruta y pasará a ser una ruta no activa.
Diagrama	<pre> graph LR     Actor((Actor)) --&gt; UC1([Finalizar ruta])     UC1 --&gt; UC2([Listar mis rutas])     UC3([Actualizar base de datos del dispositivo]) --&gt; UC2     UC4([Actualizar registro en Firebase]) --&gt; UC2     </pre>

Figura 12: Caso de uso – Finalizar ruta

Caso de uso	Eliminar ruta
Actores	Usuario
Propósito	Eliminar tanto rutas del usuario como compartidas por sus contactos
Resumen	Esta opción está disponible tanto en las rutas del usuario como en las que comparten sus contactos. El usuario sólo tiene que desplazar a la izquierda la entrada de la ruta en la lista y pulsar el botón de borrar. En ese momento se actualiza la lista de las rutas en Firebase y en la base de datos del dispositivo. En la lista de las rutas ya no aparecerá la entrada de la ruta borrada.
Diagrama	<pre> graph LR     Actor((Actor)) --&gt; UC1([Eliminar ruta])     UC1 --&gt; UC2([Listar rutas])     UC3([Eliminar ruta en la base de datos del dispositivo]) --&gt; UC2     UC4([Eliminar ruta en Firebase]) --&gt; UC2     </pre>

Figura 13: Caso de uso – Eliminar ruta

Caso de uso	Hacer logout
Actores	Usuario
Propósito	Salir de la cuenta que el usuario tenga cargada en la aplicación.
Resumen	Al pulsar esta opción del menú, el usuario navega a la pantalla de login, donde podrá logarse nuevamente en la aplicación con la misma cuenta o con otra si así lo desea. La pantalla de login se carga después de que la aplicación borre los datos de la cuenta y de todas las rutas de la base de datos del dispositivo.
Diagrama	<pre> graph LR     Actor[Actor] --&gt; HacerLogout(Hacer Logout)     HacerLogout --&gt; Login(Login)     BorrarDatos(Borrar todos los datos de la cuenta y rutas en la base de datos del dispositivo) &lt;--&gt; Login     </pre>

Figura 14: Caso de uso - Logout

Caso de uso	Reestablecer contraseña
Actores	Usuario
Propósito	Reestablecer la contraseña asociada a una cuenta de usuario
Resumen	Al pulsar este botón en la pantalla de login se carga otra pantalla en la que el usuario debe introducir el correo electrónico asociados a su cuenta. La aplicación enviará un correo electrónico a esa dirección con un link para que el usuario introduzca su nueva contraseña.
Diagrama	<pre> graph LR     Actor[Actor] --&gt; Reestablecer(Reestablecer contraseña)     Reestablecer --&gt; Enviar(Enviar)     Enviar --&gt; Envio(Envío de correo)     Reestablecer &lt;--&gt; Introducir(Introducir correo electrónico)     </pre>

Figura 15: Caso de uso – Reestablecer contraseña

## 4. Diseño de la aplicación

En este capítulo se muestra el árbol de navegación horizontal junto con el diseño de alto nivel de cada pantalla de la aplicación.

### 4.1 Árbol de navegación

El siguiente diagrama muestra la navegación que el usuario puede realizar en la aplicación. Es importante destacar que el menú no es una pantalla independiente, sino que está integrado en las pantallas a las que da acceso, en este caso, los hijos de “Menu” en el árbol de navegación [6].

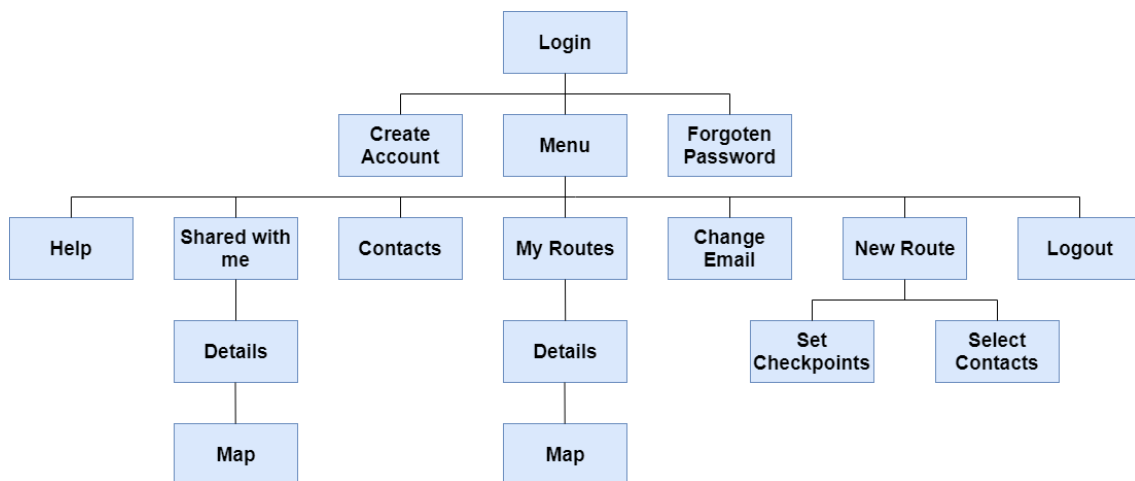


Figura 16: Árbol de navegación

### 4.2 Prototipo de alto nivel

En esta sección se muestra con detalle cada pantalla de la aplicación tal como se verá en el producto final [7].

## Pantalla Login

Cada vez que el usuario abre la aplicación, en caso de que no se haya logado previamente, aparecerá la pantalla “Login”. También se mostrará cuando el usuario pulse la opción “Logout” del menú.

Desde la pantalla “Login” el usuario puede ir a la pantalla para crear una nueva cuenta, “Create New Account”, y a la pantalla para reestablecer su contraseña, “Forgotten password”.

Una vez que el usuario introduce unas credenciales válidas y pulse el botón “Login”, se abrirá la pantalla “My Routes”. En caso de que las credenciales no sean válidas, aparecerá un mensaje indicando que la cuenta no es válida.

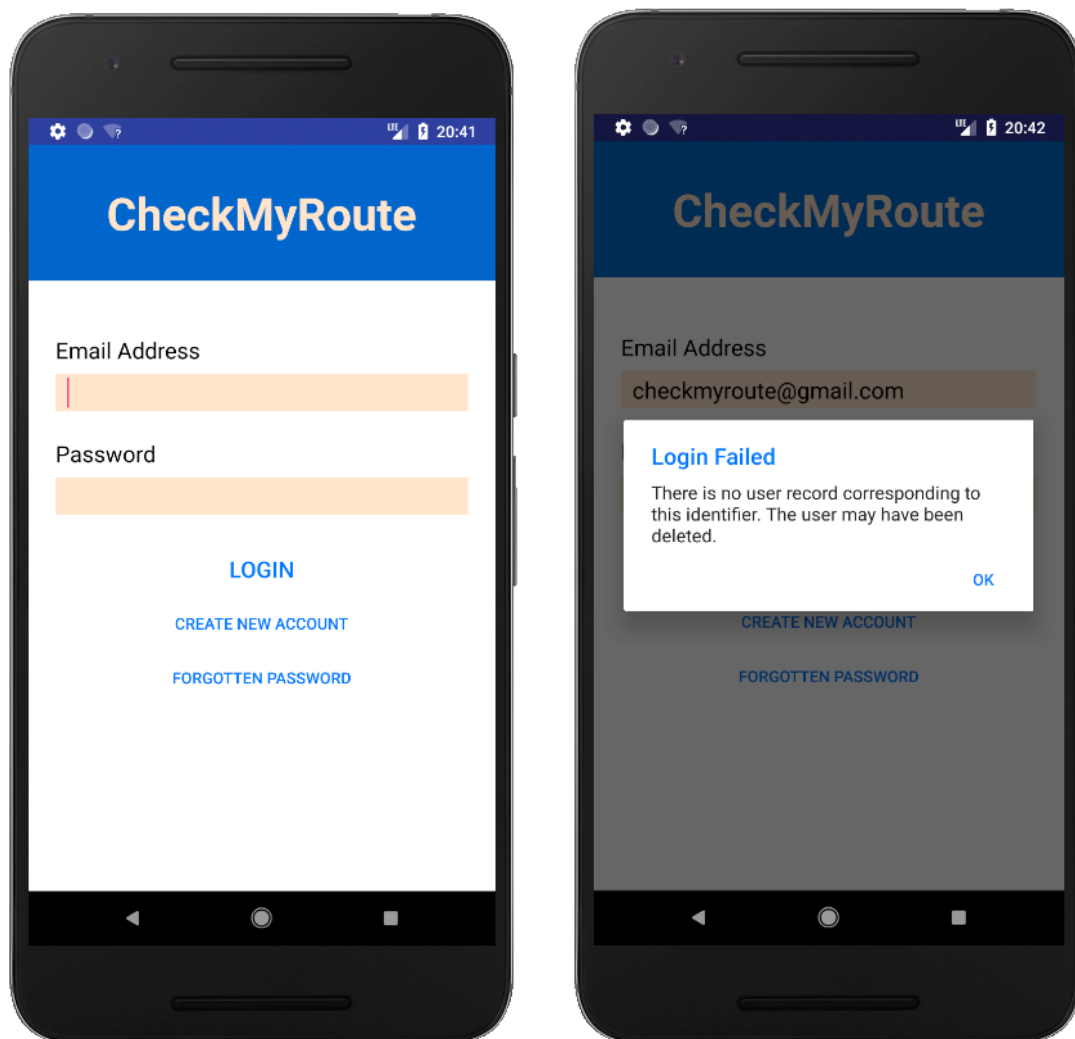


Figura 17: Pantalla de Login

## Pantalla Create New Account

En esta pantalla el usuario debe introducir un número de teléfono con el prefijo de su país, una dirección de correo electrónico y una contraseña (2 veces). En caso de que los datos introducidos sean correctos aparecerá un mensaje indicando que su cuenta ha sido creada. En caso de que se produzca algún error aparecerá el mensaje de error correspondiente.

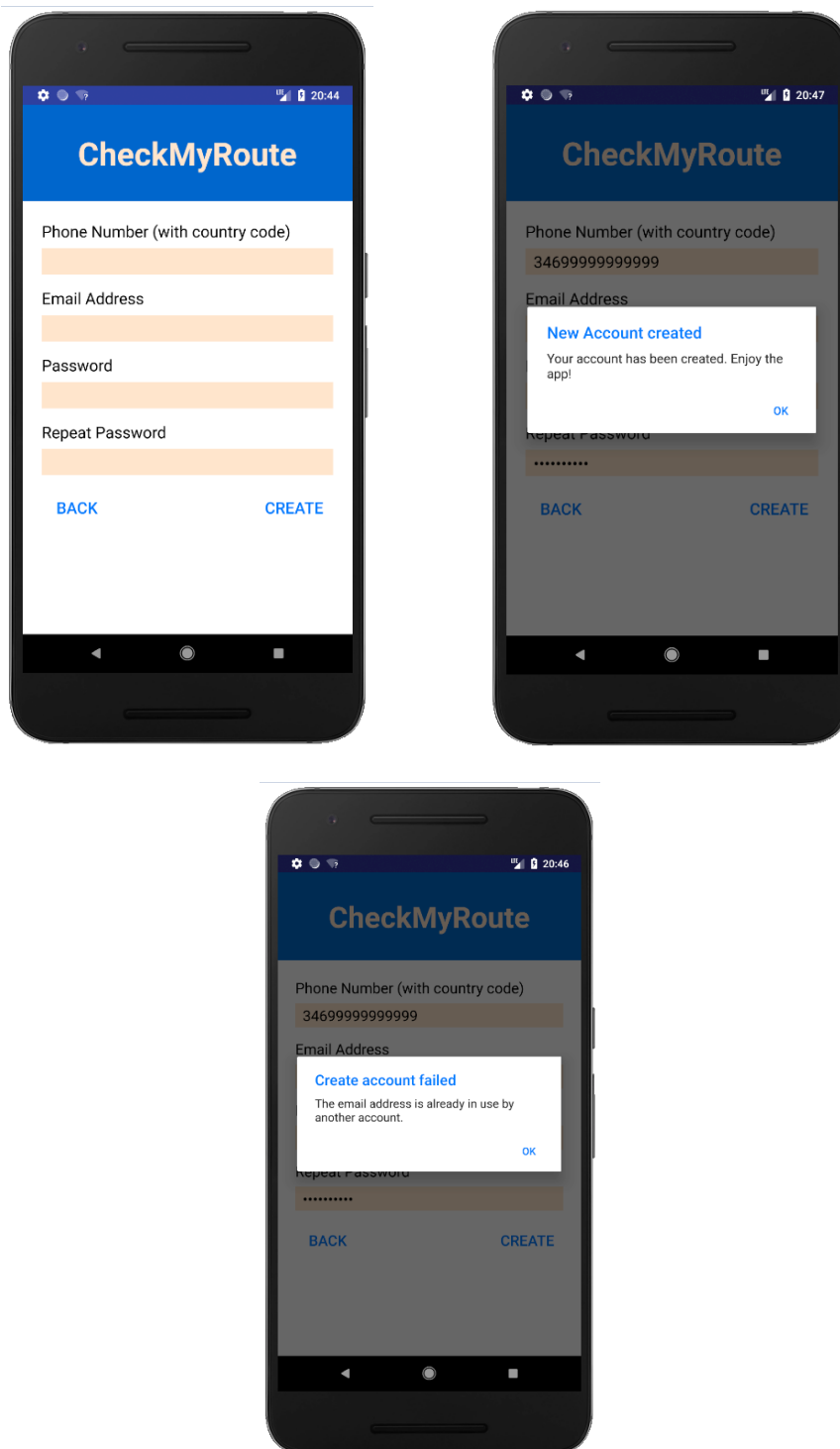


Figura 18: Pantalla para crear una nueva cuenta

## Pantalla Forgotten password

En esta pantalla el usuario puede reestablecer la contraseña de su cuenta. Para ello tiene que introducir el email asociado a su cuenta. Si es correcto, se le enviará un mensaje con un link para introducir la nueva contraseña de su cuenta. En todo caso, aparecerán los mensajes correspondientes.



Figura 19 Pantalla para reestablecer contraseña

## Menú de navegación

El menú se despliega pulsando el botón con icono “hamburger”, que está disponible en todas las pantallas a las que este menú da acceso. Estas pantallas son: “My Routes”, “Shared with me”, “New Route”, “Contacts”, “Change password” y “Help”. La última opción del menú, “Logout”, no accede a una pantalla, sino que permite al usuario salir de la cuenta con la que hizo “Login” en la aplicación.

El botón que da acceso al menú se sitúa en la parte superior izquierda de las pantallas mencionadas anteriormente.

Si el usuario abre el menú pero pulsa en cualquier zona que no sea una opción, el menú se cierra.

En la parte superior del menú aparecen el número de teléfono y la dirección de correo electrónico del usuario.

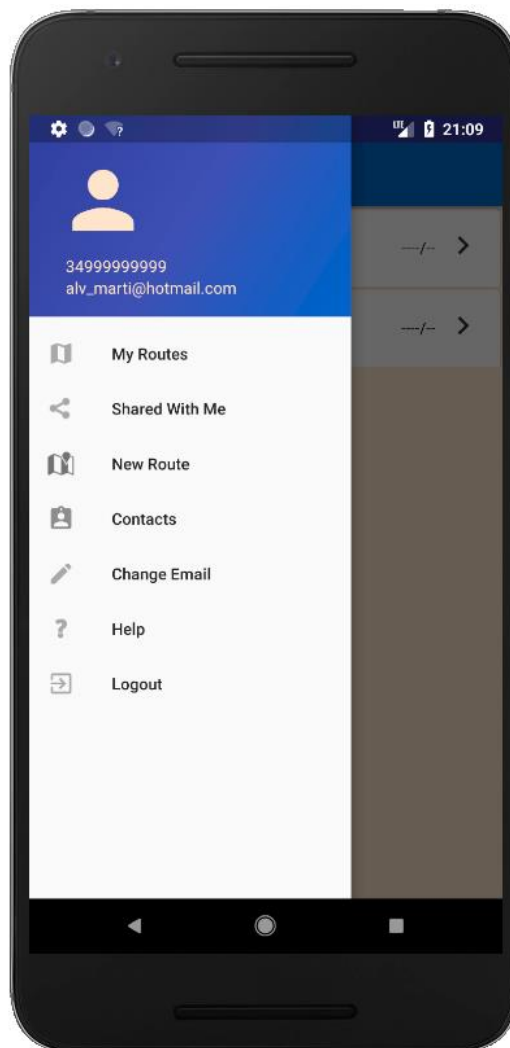


Figura 20: Menú de navegación



## Pantalla My Routes

En esta pantalla se muestra al usuario una lista con todas las rutas que haya creado. En caso de que esté actualmente realizando un viaje y tenga una ruta activa, ésta aparecerá con las letras e iconos de color azul.

Cada entrada de la lista muestra la dirección y el país de destino de la ruta, además de la fecha en la que llegó a ese destino. En caso de que sea una ruta activa, aparecerá el texto “On the road” en lugar de la fecha.

Si el usuario pulsa sobre algún elemento de la lista, se cargará la pantalla “My Detail” con los detalles de la ruta correspondiente.

En la parte superior izquierda de la pantalla, el usuario dispone del botón del menú (icono “hamburguer”) para navegar a otras pantallas de la aplicación.

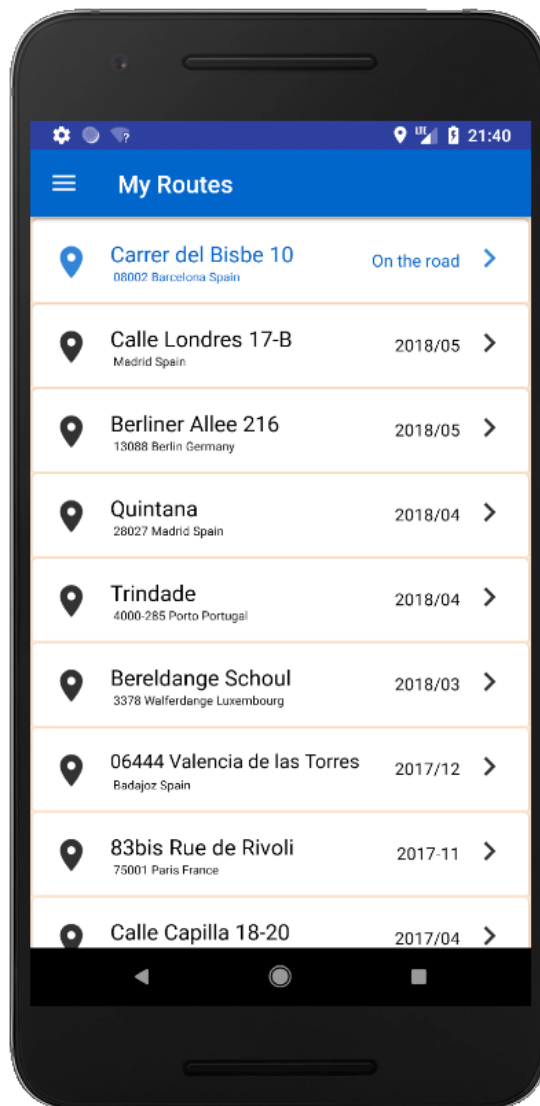


Figura 21: Pantalla de rutas de usuario

## Pantalla *MyDetail*

Muestra el detalle de la ruta seleccionada en la pantalla anterior. Se distinguen 2 secciones:

- Lista de puntos de control: Muestra todos los puntos de control definidos en la ruta junto con la última localización, el origen y el destino. Aporta información sobre la distancia desde el origen, la localización y la hora y fecha a la que se pasó por ese punto.
- Lista de contactos con los que el usuario comparte la ruta.

En caso de que se visualice una ruta activa, algunos datos referidos anteriormente aparecerán con guiones. A medida que se vaya avanzando en el viaje y se alcancen puntos de control por los que no se había pasado, esta información se irá actualizando en tiempo real. Para ello, el usuario debe aceptar los permisos de localización que la aplicación le pedirá la primera vez cargue la pantalla “My Routes”.

Desde esta pantalla el usuario puede volver a la pantalla anterior pulsando en el botón con icono “Left Arrow” en la esquina superior izquierda.

Para visualizar en el mapa la ruta del viaje, los puntos de control y la última localización, el usuario sólo tiene que pulsar en el botón “Map” de la esquina superior derecha.

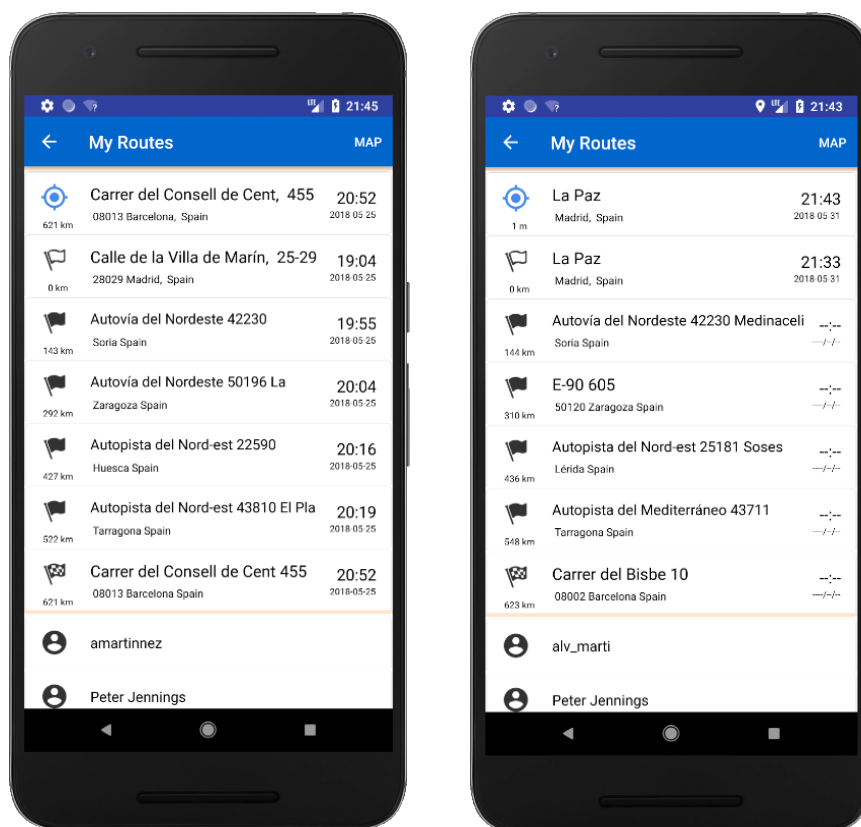


Figura 22: Pantalla de detalle de una ruta

## Pantalla Map

Muestra en el mapa la ruta del viaje, los puntos de control y la última localización.

Si el usuario selecciona un punto de control sobre el mapa, se muestra la distancia al origen de ese punto de control.

El usuario puede volver a la pantalla anterior, de detalles de la ruta, pulsando en la flecha de la esquina superior izquierda.

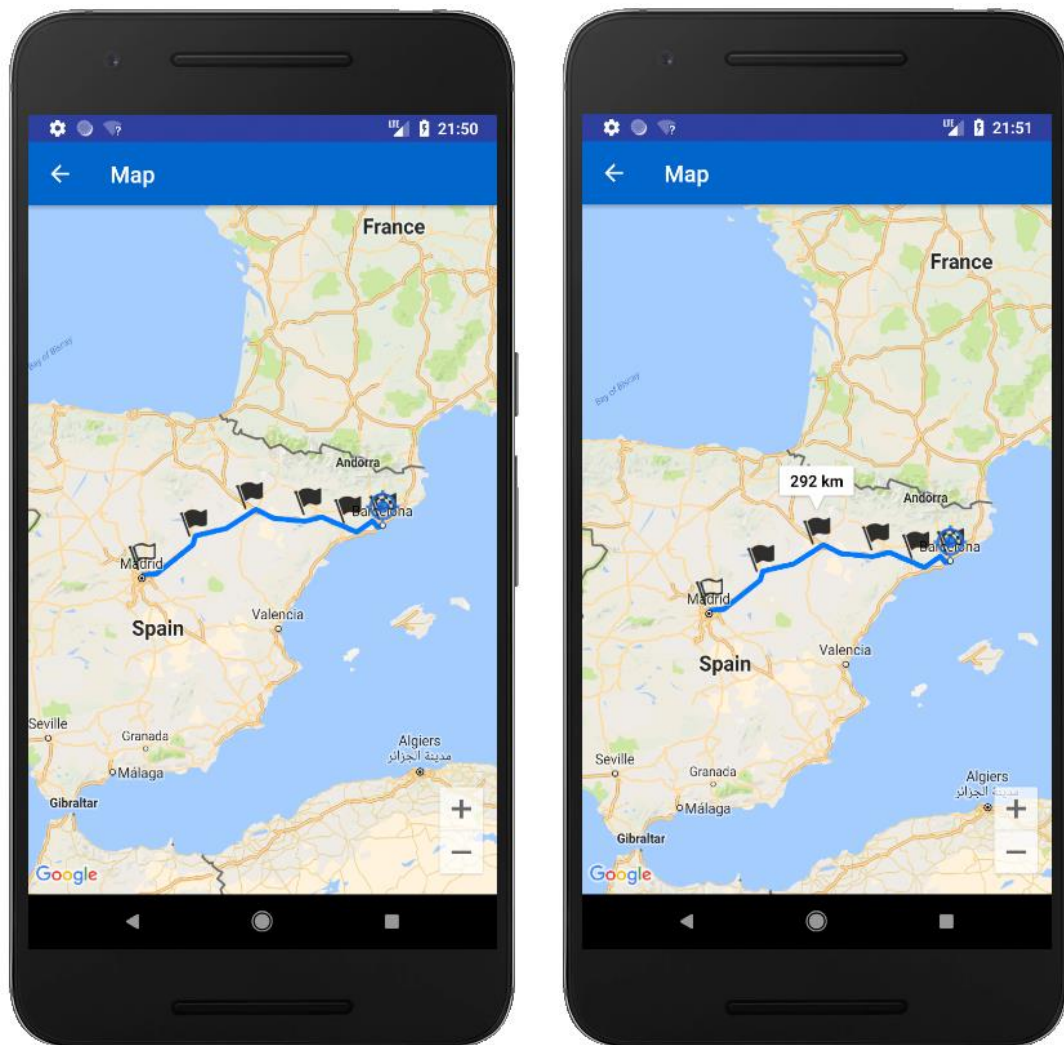


Figura 23: Pantalla del mapa de una ruta

## Pantalla Shared with me

Muestra las rutas compartidas por los contactos del usuario.

Esta pantalla es idéntica a “My Routes” con la única diferencia que cada entrada de la lista muestra, además, el nombre del contacto que comparte la ruta.

Si el usuario pulsa sobre alguna entrada de la lista, se cargará la pantalla “Shared Detail” con la información de la ruta seleccionada. Esta pantalla es idéntica a la mencionada anteriormente, con la única diferencia de que en vez de aparecer la lista de contactos con los que se compartió la ruta, aparece el nombre del usuario que comparte la ruta.

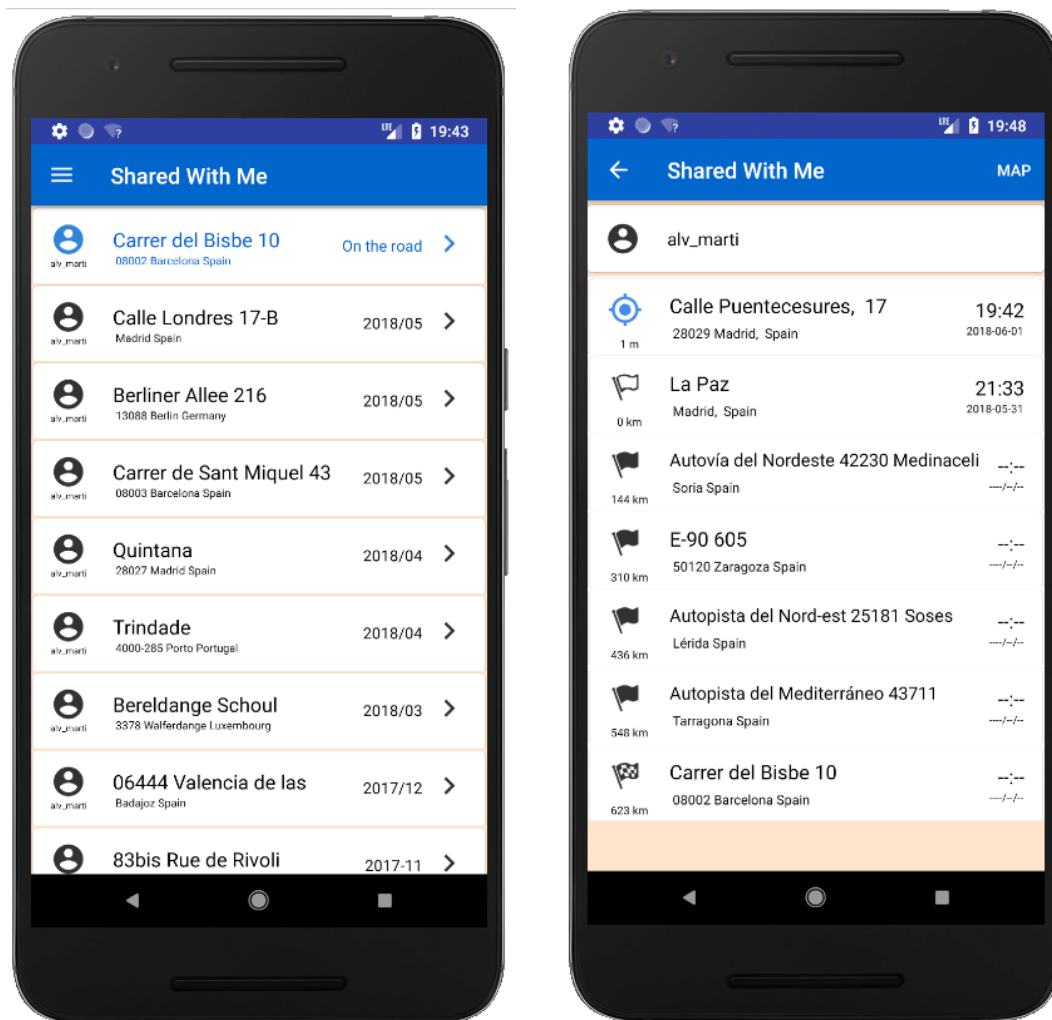


Figura 24: Pantalla de rutas compartidas y detalle de una ruta

El mapa de cada ruta es idéntico al mencionado para la pantalla “My Detail”.

## Pantalla *New Route*

Desde esta pantalla el usuario crea la ruta que desea compartir.

Al pulsar en el campo para introducir el origen, aparece un editor donde se cargan las opciones disponibles a medida que escribe una localización. Para seleccionar el destino aparece el mismo editor. Después de seleccionar los puntos de origen y destino, el mapa se refresca con estos puntos.

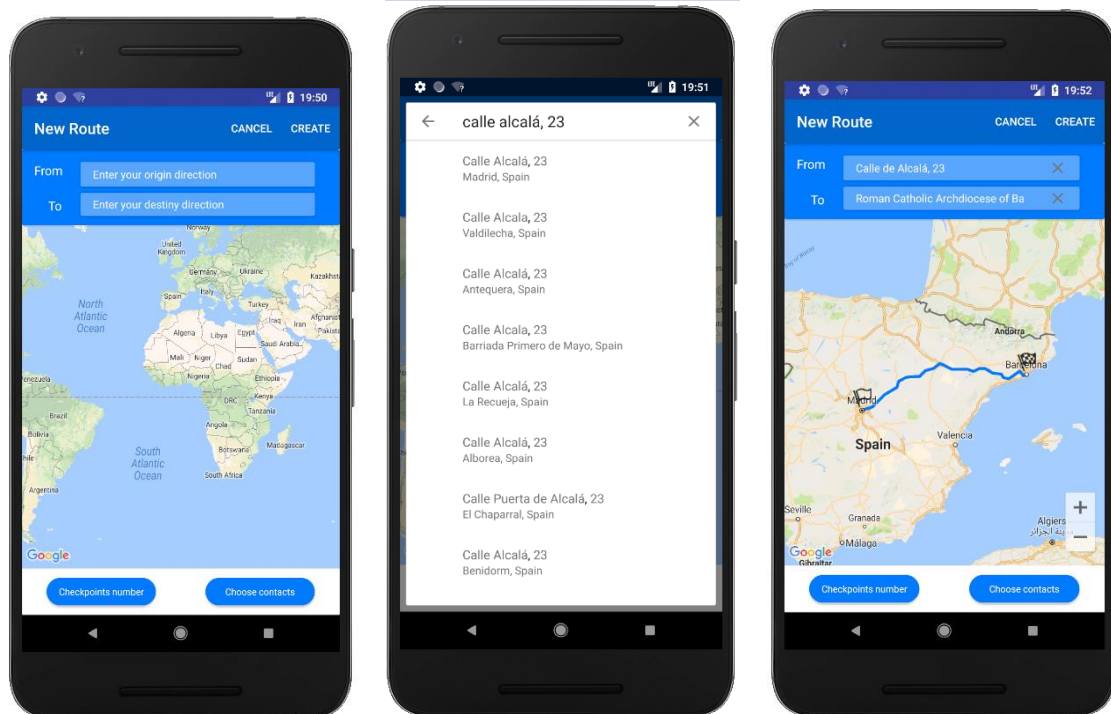


Figura 25: Pantalla crear ruta – Origen y destino

El siguiente paso es que el usuario establezca el número de puntos de control. Para ello debe pulsar el botón “Checkpoints Nuber”, que aparece en la parte inferior de la pantalla. Posteriormente se despliega un diálogo con un “Spinner” donde podrá seleccionar el número exacto de puntos de control. Al pulsar en el botón “Done” del diálogo, se cargará el mapa con los “checkpoints” marcados en el mapa. El número máximo de puntos de control que se ofrece al usuario para una ruta es 20.

El último paso para poder crear una ruta es seleccionar al menos un contacto con el que compartirla. Para ello dispone del botón “Choose Contacts” en la parte inferior de la pantalla. A pulsarlo se despliega la lista de contactos. En este punto solo tiene que marcar la casilla de cada contacto que quiera añadir y pulsar el botón “Done” en la esquina superior derecha.

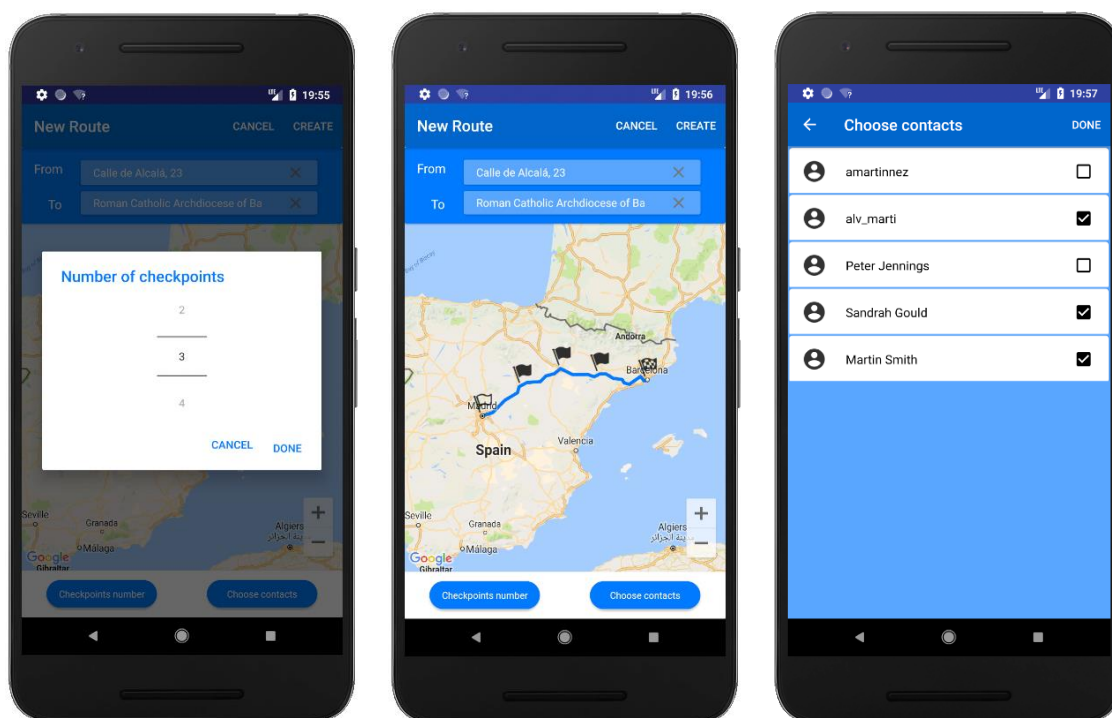


Figura 26: Pantalla crear ruta – Checkpoints y contactos

Una vez que el usuario completa los pasos anteriores, si pulsa en el botón “Create”, en la esquina superior derecha, aparecerá un mensaje de diálogo indicando que la ruta se creó satisfactoriamente. Tras aceptar este mensaje, se abrirá la pantalla “My Routes” con el registro de la ruta que se acaba de crear. En caso de que no se pueda crear la ruta, se mostrará el diálogo con el mensaje correspondiente al error que se haya producido.

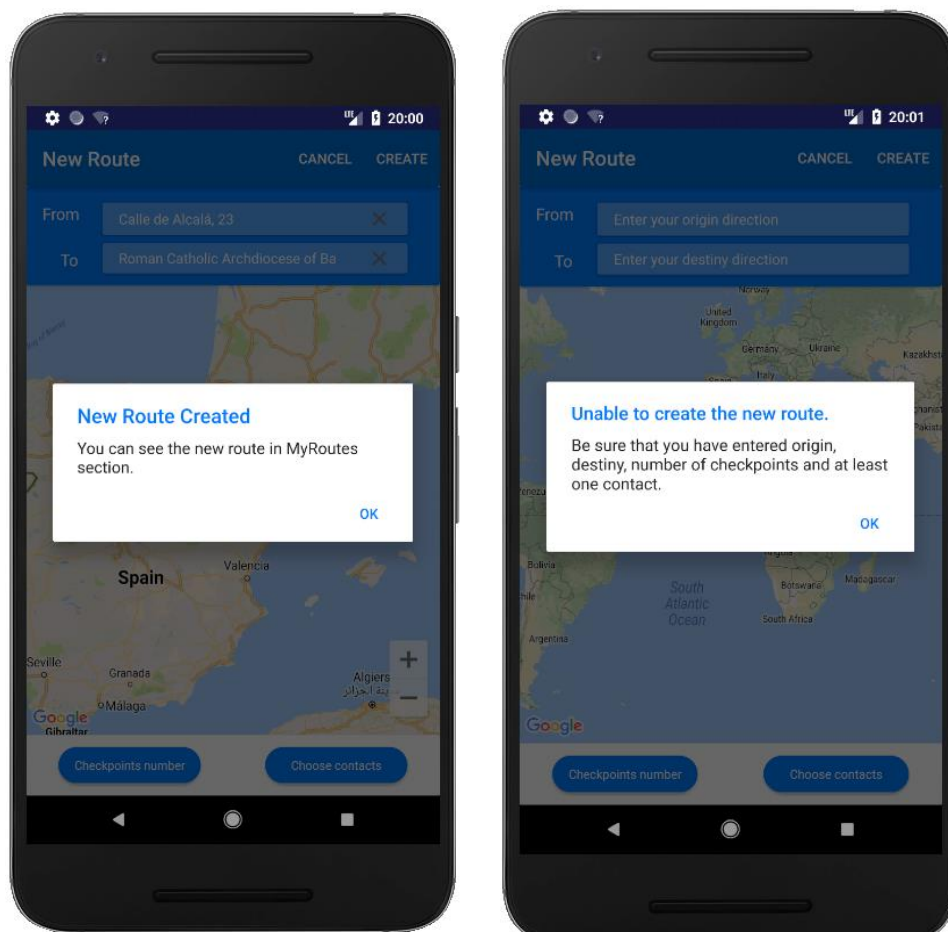


Figura 27: Pantalla crar ruta - Mensajes

Una vez creada la ruta, los contactos seleccionados tendrán un nuevo registro, correspondiente a la ruta que acaba de compartir el usuario, en sus pantallas “Shared with me”.

## Pantalla **Contacts**

Muestra la lista de contactos del usuario que tienen una cuenta en la aplicación.

## Pantalla **Help**

Muestra la información de ayuda para el usuario.

Esta pantalla tiene varios apartados con la información de cada sección del menú de opciones.

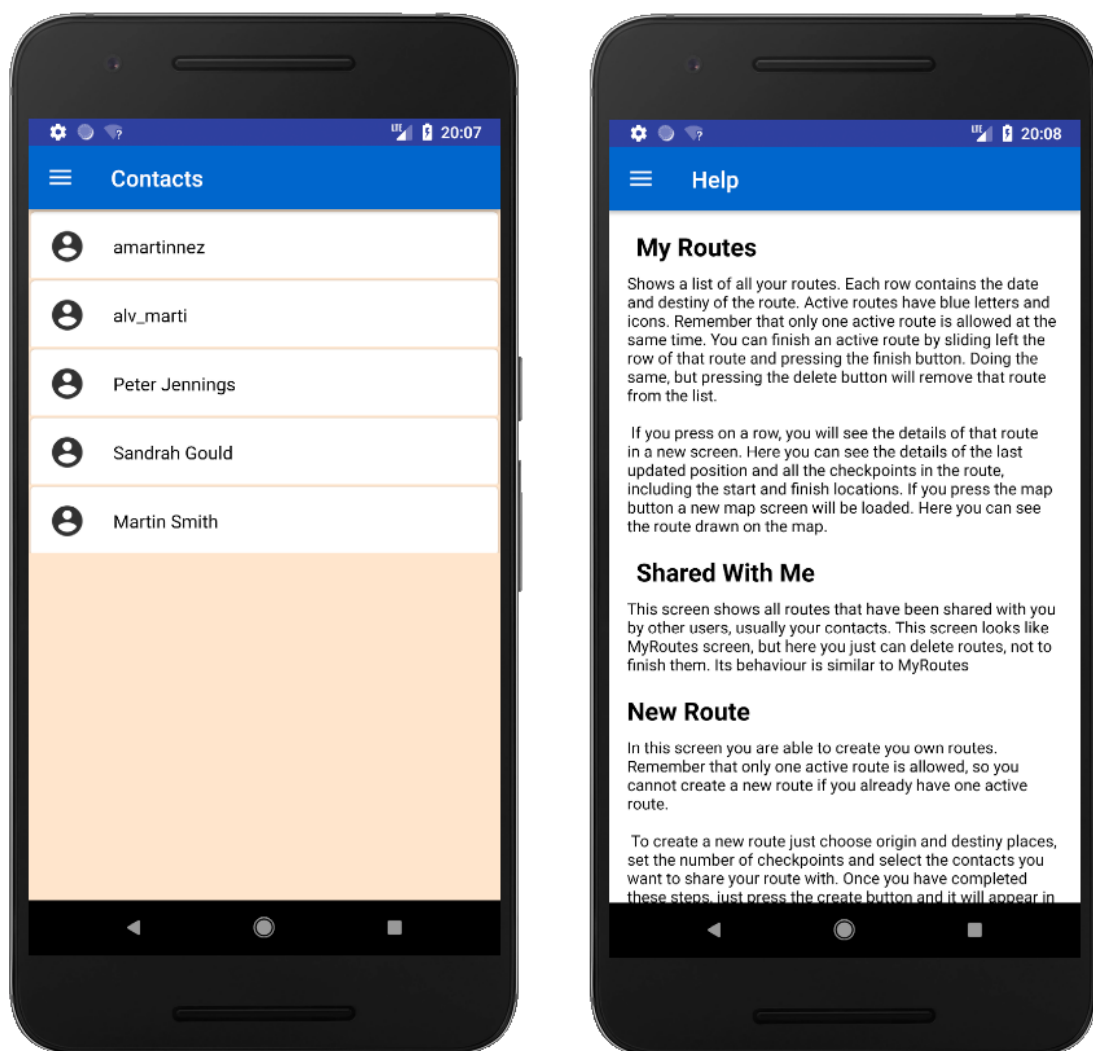


Figura 28: Pantallas contactos y ayuda



## Pantalla Change Email

Desde esta pantalla el usuario puede cambiar la contraseña de su cuenta. Tras pulsar el botón “Change” aparecerá un mensaje en la parte inferior informando al usuario de si se pudo realizar el cambio o no.

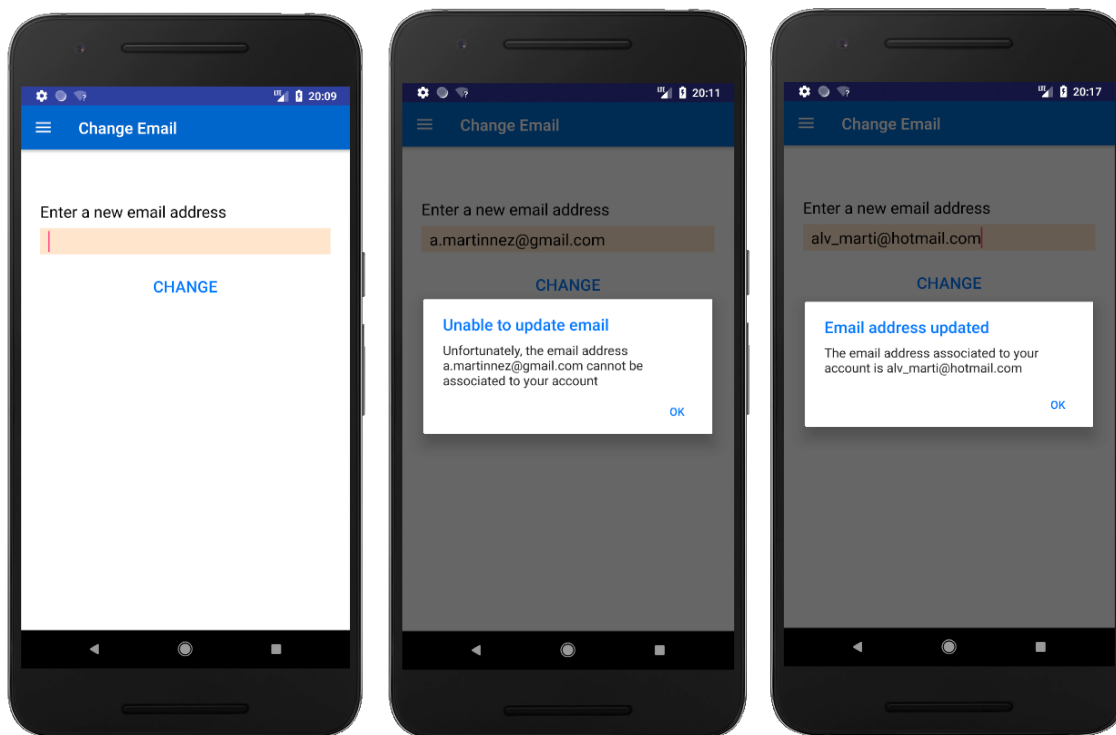


Figura 29: Pantalla cambiar email

## Finalizar una ruta activa

Esta acción está disponible en la pantalla “My Routes” para que el usuario pueda dar por finalizada una ruta activa.

Para ello sólo tiene que deslizar a la izquierda la entrada en la lista de la ruta que desee finalizar y pulsar en el botón “Finish”. A continuación podrá comprobar que el color del registro cambia de azul a negro.

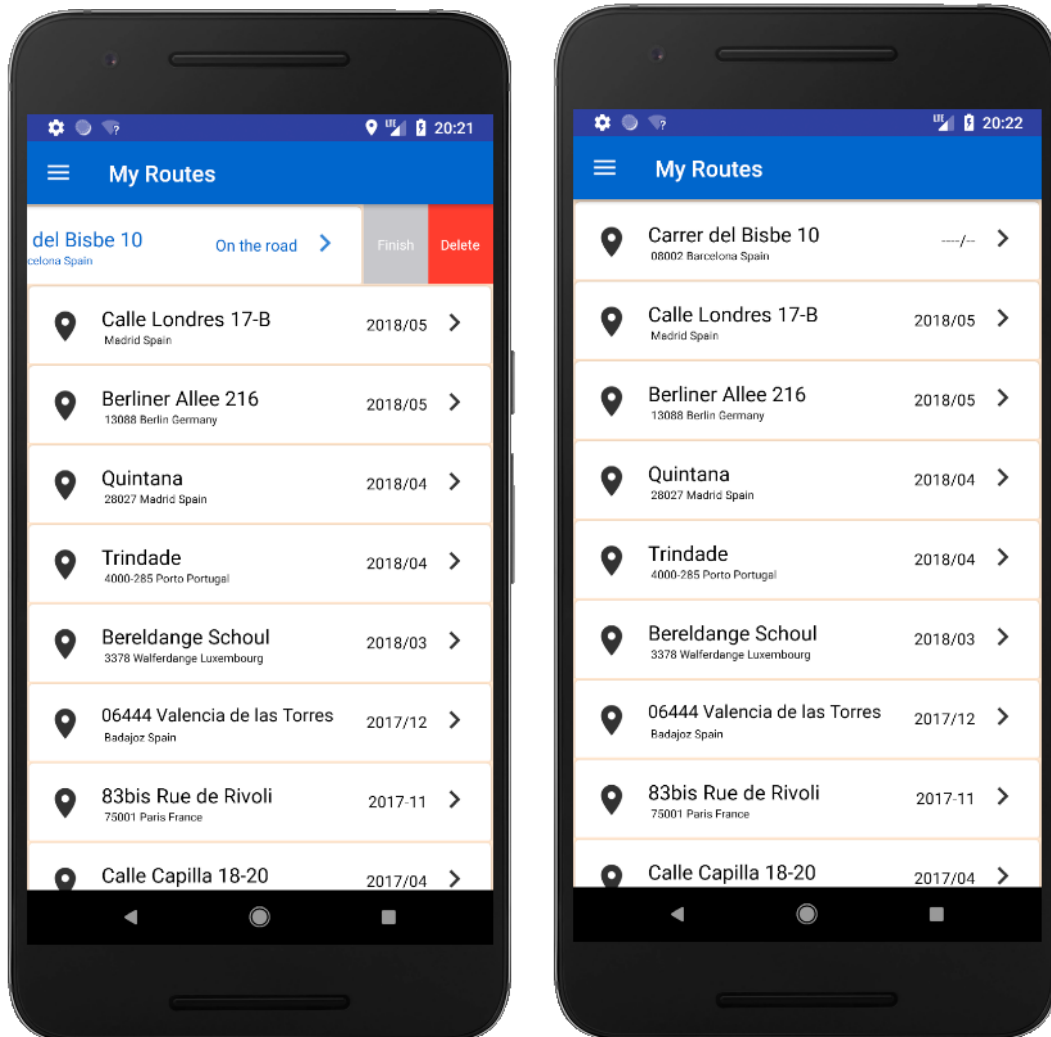


Figura 30: Finalizar ruta activa

## Eliminar una ruta

Esta acción está disponible en las pantallas “My Routes” y “Shared with me” para que el usuario pueda eliminar una ruta.

Para ello sólo tiene que deslizar a la izquierda la entrada en la lista de la ruta que desee eliminar y pulsar en el botón “Delete”. A continuación podrá comprobar que se elimina la entrada de la ruta en la lista.

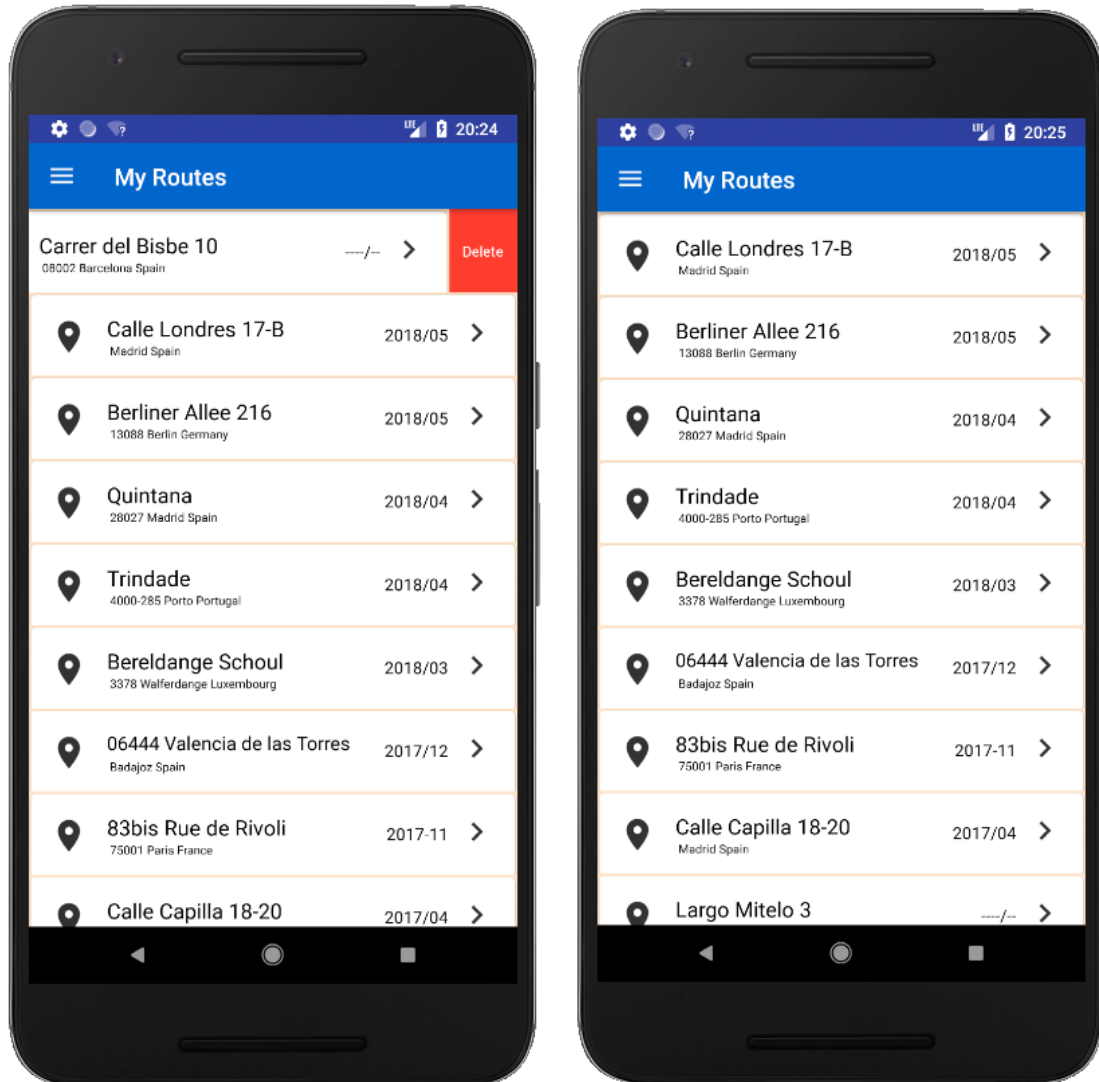


Figura 31: Eliminar ruta

## 5. Arquitectura del sistema

La aplicación se basa en una arquitectura básica *cliente-servidor* [8]. Funciona en modo *online*, por lo que es fundamental que el dispositivo tenga conexión a internet, ya sea por redes de telefonía móvil o por redes inalámbricas.

### 5.1 Diseño físico

La siguiente figura muestra la representación gráfica del diseño físico de la arquitectura del sistema. Se pueden observar los componentes principales que forman la arquitectura y los protocolos de comunicación que utiliza la aplicación para cargar y actualizar datos.

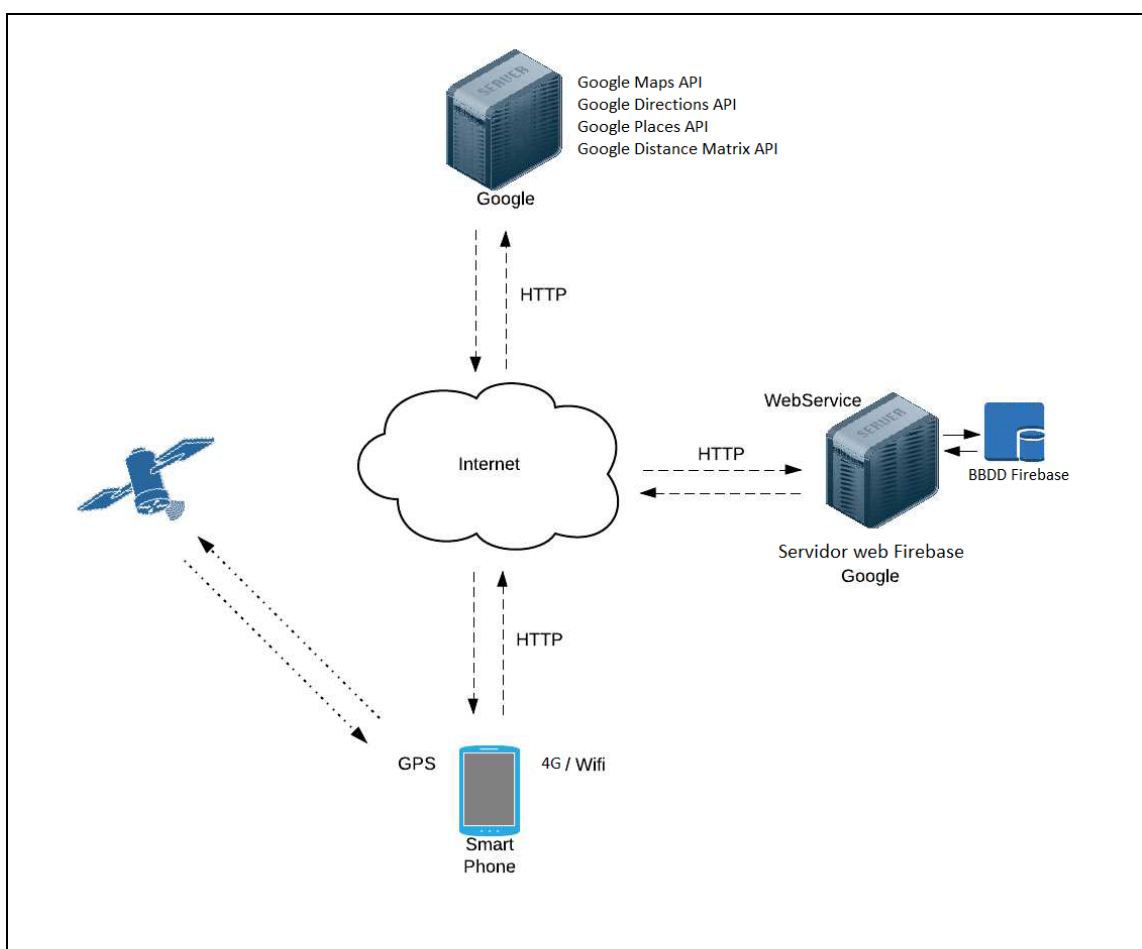


Figura 32: Diseño Físico

La aplicación se instala en el dispositivo móvil (cliente) y solicita información, mediante peticiones HTTP, a una base de datos alojada en el servicio web Firebase (servidor). Este servicio devuelve la información de las consultas al cliente en formato JSON. La aplicación se encarga de gestionar las peticiones al servidor y la información recibida. Los datos que solicita y gestiona la

aplicación se refieren principalmente a información de los usuarios y de las rutas.

## 5.2 Servidor

### 5.2.1 Plataforma Firebase

Firebase es un servicio de Google para el desarrollo de aplicaciones web y aplicaciones para móviles. Su función principal es ofrecer un servidor en línea para que las aplicaciones puedan gestionar los datos de sus usuarios y permitir ciertas funcionalidades. Entre los servicios que ofrece, la aplicación *CheckMyRoute* utiliza *Realtime Database* [9] como base de datos en línea de la aplicación. Además, utiliza el servicio *Authentication* [10] para gestionar las cuentas de los usuarios en la aplicación.

### 5.2.2 Firebase Realtime Database (BBDD Firebase)

La aplicación utiliza este servicio como base de datos *online*, para gestionar los datos de los usuarios y de las rutas y poder mostrar la información actualizada de cada ruta en tiempo real.

Cada vez que se actualiza la información de una ruta se envía a esta base de datos para que el resto de usuarios reciban en la aplicación los datos actualizados. También se actualiza la base de datos cada vez que se crea un usuario nuevo, un usuario hace “login” o se borra una ruta.

La base de datos creada para la aplicación se llama *CheckMyRouteDB*, y está compuesta por las tablas “Users”, “My Routes”, “Shared with me”, “Routes”, “Other Users”, “Checkpoints”.

#### Tabla Users

Almacena el número de teléfono, dirección de correo electrónico y contraseña de los usuarios que han creado una cuenta en la aplicación. Además, incluye un contador que aumenta una unidad cada vez que crea una ruta. La clave primaria de la tabla es el número de teléfono del usuario (11 dígitos, incluyendo el prefijo del país).

También incluye la tabla “My Routes” con los identificadores de las rutas que crea el usuario y la tabla “Shared with me” que, al igual que la anterior, guarda los identificadores de las rutas que sus contactos comparten con el usuario.

#### Tabla My Routes

Sus únicos registros son los identificadores de las rutas. Cada cuenta de usuario incluye una de estas tablas. Si el usuario borra en la aplicación alguna de sus rutas creadas, también se borra el registro de esa ruta en esta tabla.

## Tabla Shared with me

Sus únicos registros son los identificadores de las rutas. Cada cuenta de usuario incluye una de estas tablas. Si el usuario borra en la aplicación alguna de las rutas que otro contacto compartió, también se borra el registro de esa ruta en esta tabla.

## Tabla Routes

Almacena la información de las rutas. Los datos que incluyen son: el identificador de la ruta, la información general de la ruta (origen, destino, distancia total, fecha y hora de inicio, fecha y hora de fin, número de puntos de control), la tabla de los puntos de control de la ruta ("*Checkpoints*"), el identificador del contacto que comparte la ruta, la última posición del usuario durante el viaje, incluye una tabla de los usuarios con los que se comparte la ruta ("*Other Users*") y el número de usuarios que no han borrado la ruta en sus dispositivos. Este último dato se utiliza para borrar la tabla en caso de que todos los usuarios hayan borrado la ruta en sus dispositivos.

El identificador de cada ruta se compone por la concatenación del número de teléfono y el contador de rutas creadas del usuario.

## Tabla Other Users

Almacena el identificador de los usuarios con los que se comparte una ruta. Está incluida como un registro en la tabla "*Routes*".

## Tabla Checkpoints

Almacena los puntos de control de una ruta. Se incluye como parte del registro de una ruta en la tabla "*Routes*".

Los datos que guarda esta tabla son: la distancia al origen (en kilómetros), el lugar (dirección), la fecha y la hora por la que se pasa por este punto (en caso de no haber pasado todavía, este dato estará en blanco).

## Modelo relacional de la base de datos de Firebase

La siguiente figura muestra la relación entre las tablas de la base de datos en el servidor Firebase. Las flechas de color gris indican la relación entre dos tablas para reflejar la consistencia de datos.

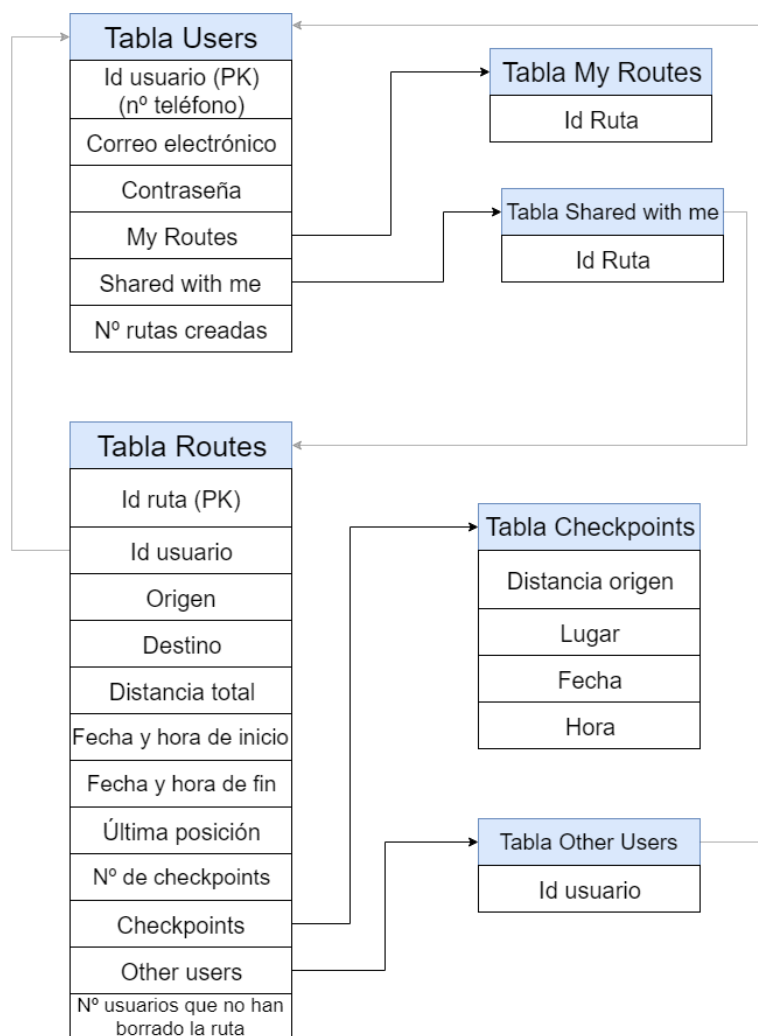


Figura 33: Base de datos en Firebase

## 5.3 Cliente

El lado del cliente, de acuerdo a la arquitectura del sistema definida anteriormente, es el dispositivo móvil que ejecuta la aplicación.

### 5.3.1 Aplicación CheckMyRoute

La aplicación se desarrolla para dispositivos con sistema operativo Android con versiones 4.0 o superior.

### 5.3.2 Base de datos Sugar ORM

La base de datos que utiliza la aplicación para manejar la información en el propio dispositivo viene definida por la librería *Sugar ORM*.

La información que la aplicación guarda en el dispositivo son los datos del usuario y de las rutas asociadas a ese usuario (las que crea y las que sus contactos comparten con el usuario).

Cuando el usuario entra en la aplicación con su cuenta, se carga la información almacenada en Firebase a esta base de datos.

Cada vez que el usuario sale de su cuenta en la aplicación, la información de esta base de datos se borra.

La base de datos que crea y maneja la aplicación en el dispositivo móvil del usuario se compone de las tablas “*User*”, “*My Routes*”, “*Other Users*”, “*Shared with me*”.

### **Tabla User**

Almacena el número de teléfono, dirección de correo electrónico y contraseña del usuario. Además, incluye un registro para saber si el usuario está logado al abrir la aplicación y un contador que aumenta una unidad cada vez que crea una ruta.

### **Tabla My Routes**

Almacena la información de las rutas que crea el usuario. Los datos que incluyen son la información general de la ruta (origen, destino, distancia total, fecha y hora de inicio, fecha y hora de fin, número de puntos de control), la tabla de los puntos de control de la ruta (“*Checkpoints*”), la última posición del usuario durante el viaje y, además, incluye una tabla con el nombre de los usuarios con los que se comparte la ruta (“*Other Users*”). El nombre de esos usuarios es el que aparece en su lista de contactos.

### **Tabla Shared with me**

Almacena la información de las rutas que los contactos del usuario comparten con él/ella. Los datos que incluyen son la información general de la ruta (origen, destino, distancia total, fecha y hora de inicio, fecha y hora de fin, número de puntos de control), la tabla de los puntos de control de la ruta (“*Checkpoints*”), la última posición (durante el viaje) y el nombre del usuario que comparte la ruta.

### **Tabla Other Users**

Almacena el nombre de los contactos con los que se comparte una ruta. Está incluida como un registro en la tabla “*Routes*”.

### **Tabla Checkpoints**

Almacena los puntos de control de una ruta. Se incluye como parte del registro de una ruta en la tabla “*Routes*”.



Los datos que guarda esta tabla son: la distancia al origen (en kilómetros), el lugar (dirección), la fecha y la hora por la que se pasa por este punto (en caso de no haber pasado todavía, este dato estará en blanco).

## Modelo relacional de la base de datos del dispositivo

La siguiente figura muestra la relación entre las tablas de la base de datos que la aplicación crea y gestiona en el dispositivo.

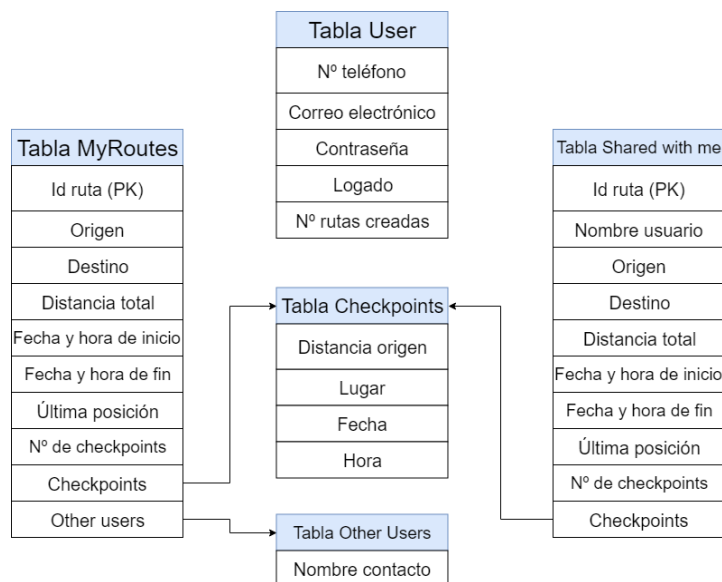


Figura 34: Base de datos Sugar ORM

## 5.4 Google APIs

La aplicación utiliza las siguientes APIs de Google para visualizar el mapa, obtener la codificación geográfica de una dirección y la ruta entre dos posiciones geográficas.

- Google Maps API [11]  
Servicio para incluir y gestionar mapas interactivos en las aplicaciones.
- Google Directions API [12]  
Servicio web para obtener la ruta desde una posición de origen a otra de destino.
- Google Places API [13]  
Servicio web para solicitar información sobre localizaciones.
- Google Distance Matrix API [14]  
Servicio web para obtener distancias entre puntos de origen y destino.

## 5.5 GPS

La aplicación utiliza el sistema de posicionamiento global (GPS) del dispositivo para obtener la posición geográfica del usuario. En caso de no estar disponible, utiliza la red de telefonía móvil como alternativa.

## 5.6 Arquitectura de la aplicación

### 5.6.1 Patrón Model-View-Controller

La aplicación se basa en el patrón de arquitectura de software MVC (Modelo-Vista-Controlador) [15]. Este patrón propone la creación de 3 capas para gestionar todos los elementos que componen la aplicación. El *Modelo* hace referencia a los componentes para representar la información, mientras que *Vista* se refiere a los elementos visuales con los que el usuario interactúa. La capa *Controlador* se encarga de toda la lógica de la aplicación.

En una aplicación estas capas deben estar claramente identificadas para garantizar el funcionamiento óptimo de la misma y facilitar su mantenimiento.

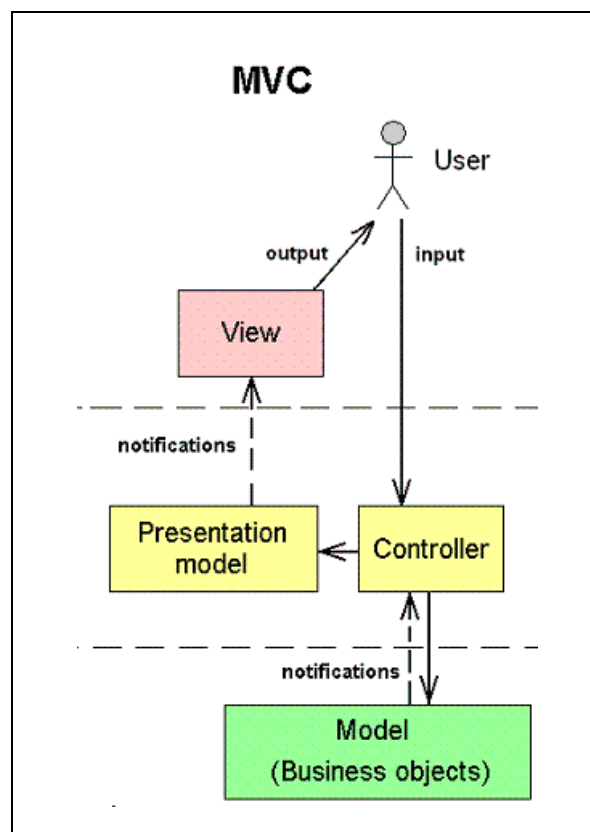


Figura 35: Patrón Model – View – Controller

## 5.6.2 Estructura de la aplicación

La aplicación se implementa basándose en la siguiente estructura de directorios, propia de una aplicación Android [16]:

### **Carpeta /src**

Contiene la lógica de la aplicación o controladores. Bajo una estructura de paquetes, se almacenan las diferentes clases de la aplicación. En esta misma carpeta se guardan también las clases modelo y las clases auxiliares.

### **Carpeta /libs**

Contiene librerías adicionales necesarias para la implementación de funcionalidades específicas.

### **Carpeta /res**

Contiene los layouts o archivos de las vistas de la aplicación así como los recursos gráficos utilizados, definición de menús, cadenas de texto constantes...

Este directorio se divide a su vez en otras carpetas o directorios:

- *Layout/hdpi*. Contiene los archivos de la vista o layouts de cada una de las pantallas de la aplicación para dispositivos móviles con una pantalla de alta densidad (hdpi).
- *Layout/xhdpi*. Contiene los archivos de la vista o layouts de cada una de las pantallas de la aplicación para dispositivos móviles con una pantalla de extra alta densidad (xhdpi).
- *Drawable-hdpi*. Contiene los recursos gráficos (imágenes, etc.) que utiliza la aplicación.
- *Drawable-xhdpi*. Contiene los recursos gráficos (imágenes, etc.) que utiliza la aplicación.
- *Values*. Contiene la definición de valores constantes que se utilizan en el aplicativo como la definición de estilos, colores o cadenas de texto, etc.

### **AndroidManifest.xml**

Archivo que contiene información esencial sobre la aplicación. Entre otros datos, contiene la siguiente información:

- Nombre del paquete de la aplicación
- Versión SDK mínima requerida
- Versión SDK objetivo
- Definición de permisos necesarios para la aplicación:
  - Acceso a Internet.

- Acceso a localización.
  - Acceso a los mapas de Google.
  - Acceso a los contactos del usuario.
- 
- Definición de las actividades que componen la aplicación.

## 6. Desarrollo

La aplicación se ha implementado en el entorno de desarrollo Android Studio [17], utilizando el lenguaje de programación Java.

Se distinguen 4 paquetes que contienen todas las clases de la aplicación:

- *com.tiglath.android.checkmyroute* -> contiene las activities, fragments y clases auxiliares que definen las pantallas y controladores.
- *com.tiglath.android.checkmyroute.model* -> contiene las clases que representan el modelo de datos.
- *com.tiglath.android.checkmyroute.googleapis* -> contiene las clases que gestionan las comunicaciones con las APIs de Google.
- *com.tiglath.android.checkmyroute.drawroutelib* -> librería de apoyo para dibujar la ruta y los marcadores en el mapa.

### 6.1 Diagrama de clases

En este apartado se describen todas las clases que conforman la aplicación.

Nombre de la clase	SingleNavigationActivity.java
Descripción	Activity que define el menú de navegación de la aplicación. Las activities con menú extienden de esta clase.
Paquete	com.tiglath.android.checkmyroute

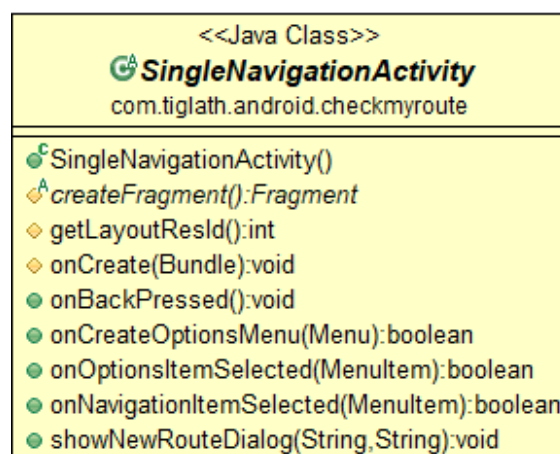


Figura 36: Diagrama de clase – SingleNavigationActivity.java

Nombre de la clase	LoginActivity.java
Descripción	Representa la pantalla de login y gestiona la entrada de datos y la comunicación con Firebase para logarse en la cuenta del usuario.
Paquete	com.tiglath.android.checkmyroute

Nombre de la clase	ForgottenActivity.java
Descripción	Representa la pantalla para que el usuario introduzca su email para cambiar su contraseña.
Paquete	com.tiglath.android.checkmyroute

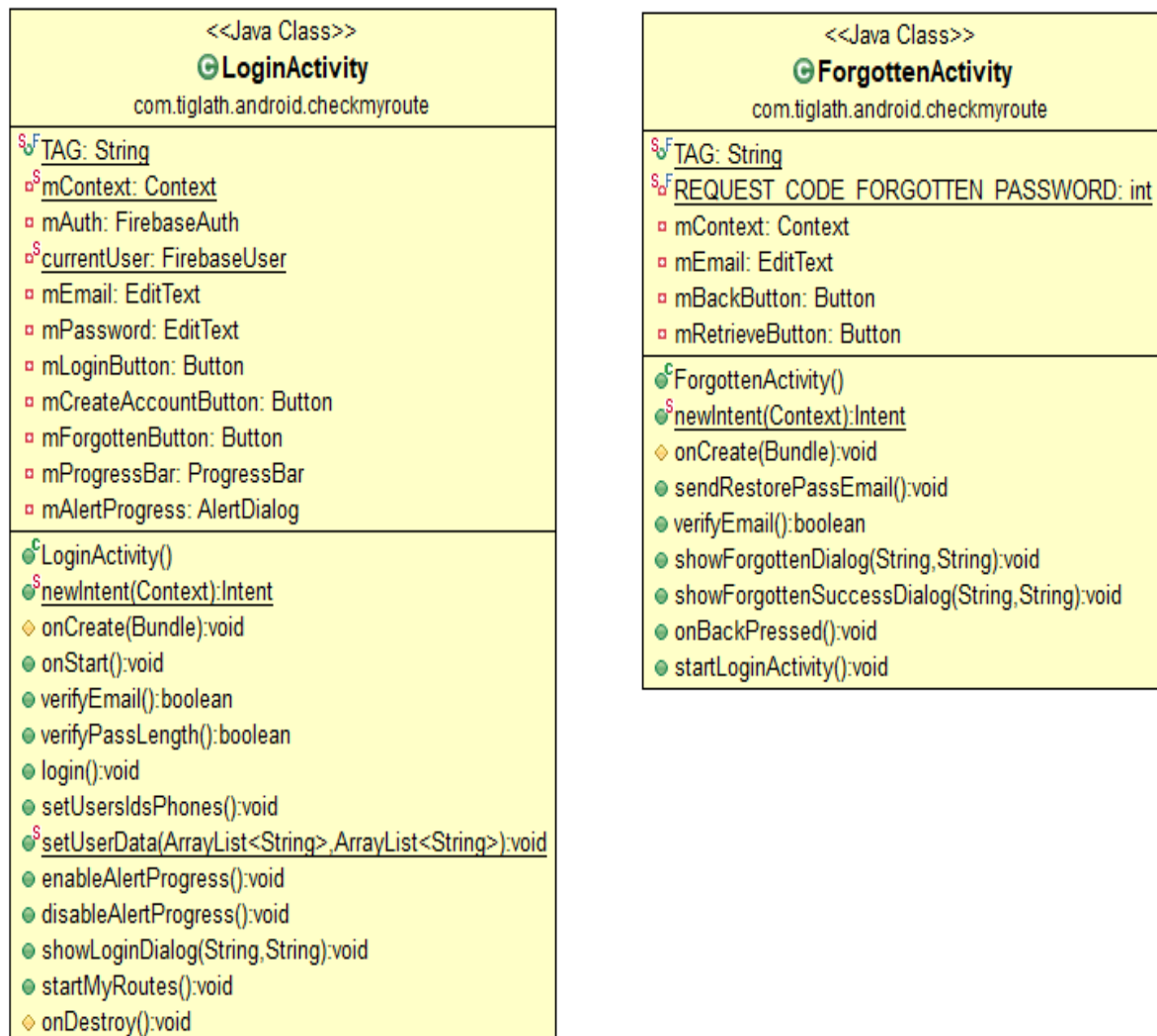


Figura 37: Diagrama de clase – LoginActivity.java – ForgottenActivity.java

Nombre de la clase	NewAccountActivity.java
Descripción	Representa la pantalla para crear una cuenta en la aplicación. Gestiona la entrada de datos y la comunicación con Firebase para crear la cuenta.
Paquete	com.tiglath.android.checkmyroute

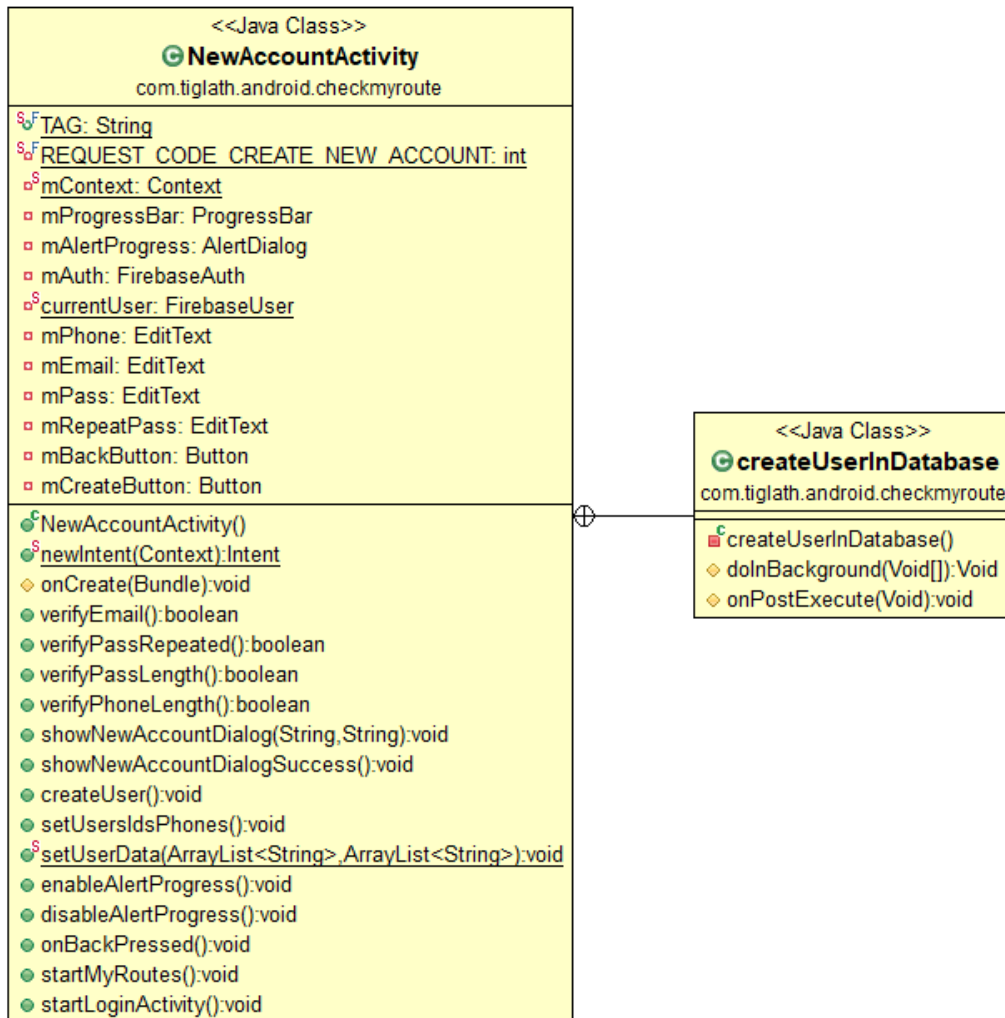


Figura 38: Diagrama de clase – NewAccountActivity.java

Nombre de la clase	MyRoutesActivity.java
Descripción	Activity que contiene el fragmento que gestiona la pantalla principal de la aplicación.
Paquete	com.tiglath.android.checkmyroute

Nombre de la clase	MyRoutesFragment.java
Descripción	Representa la lista con las rutas del usuario. Gestiona la información que se muestra, la comunicación con el servidor Firebase y la actualización de la localización.
Paquete	com.tiglath.android.checkmyroute

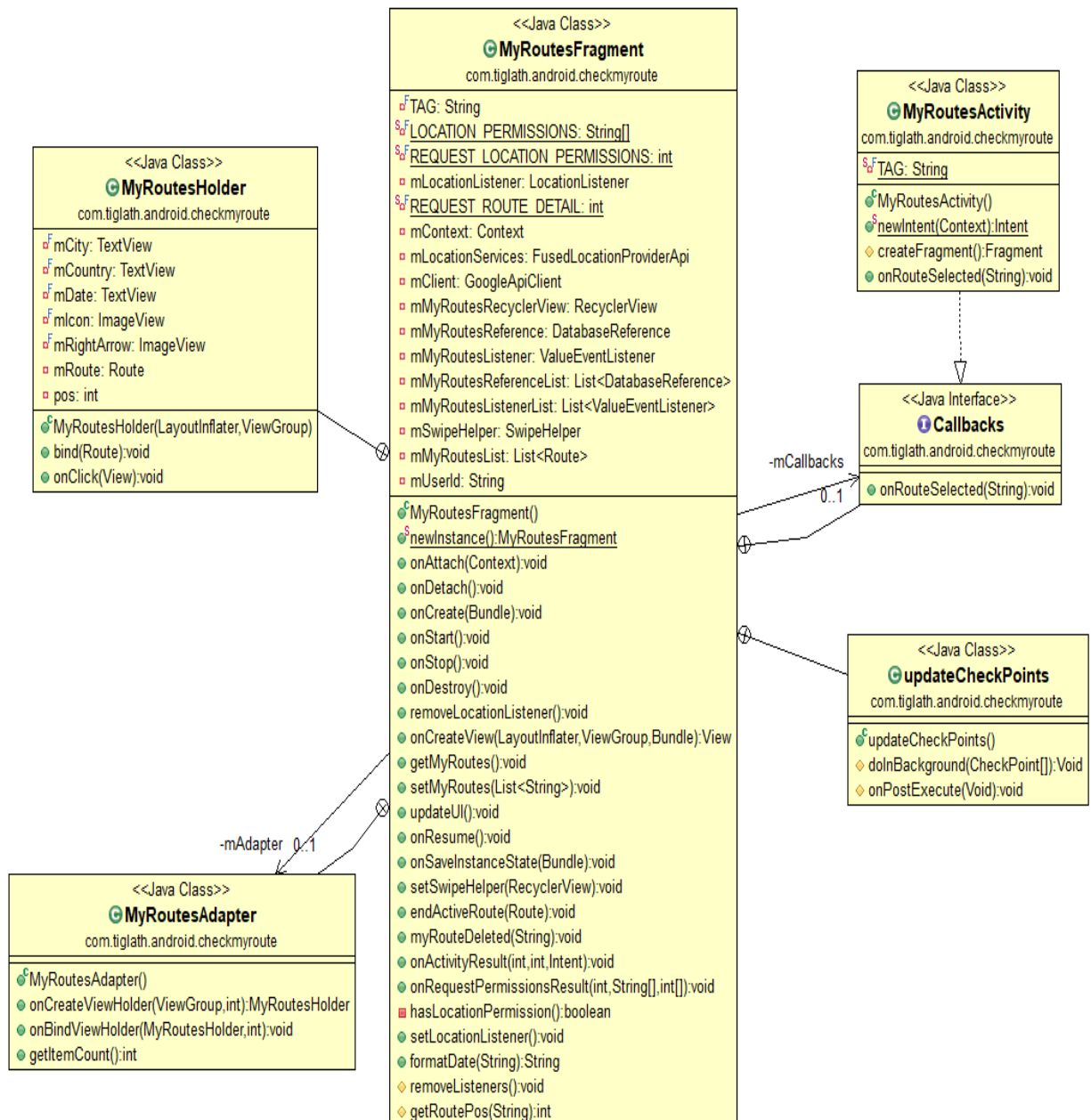


Figura 39: Diagrama de clase – MyRoutesActivity.java - MyRoutesFragment.java



Nombre de la clase	MyDetailActivity.java
Descripción	Activity que contiene el fragmento que gestiona la pantalla del detalle de una ruta del usuario.
Paquete	com.tiglath.android.checkmyroute

Nombre de la clase	MyDetailFragment.java
Descripción	Representa la lista con los detalles de la ruta del usuario y la actualización de la localización y puntos de control con Firebase.
Paquete	com.tiglath.android.checkmyroute

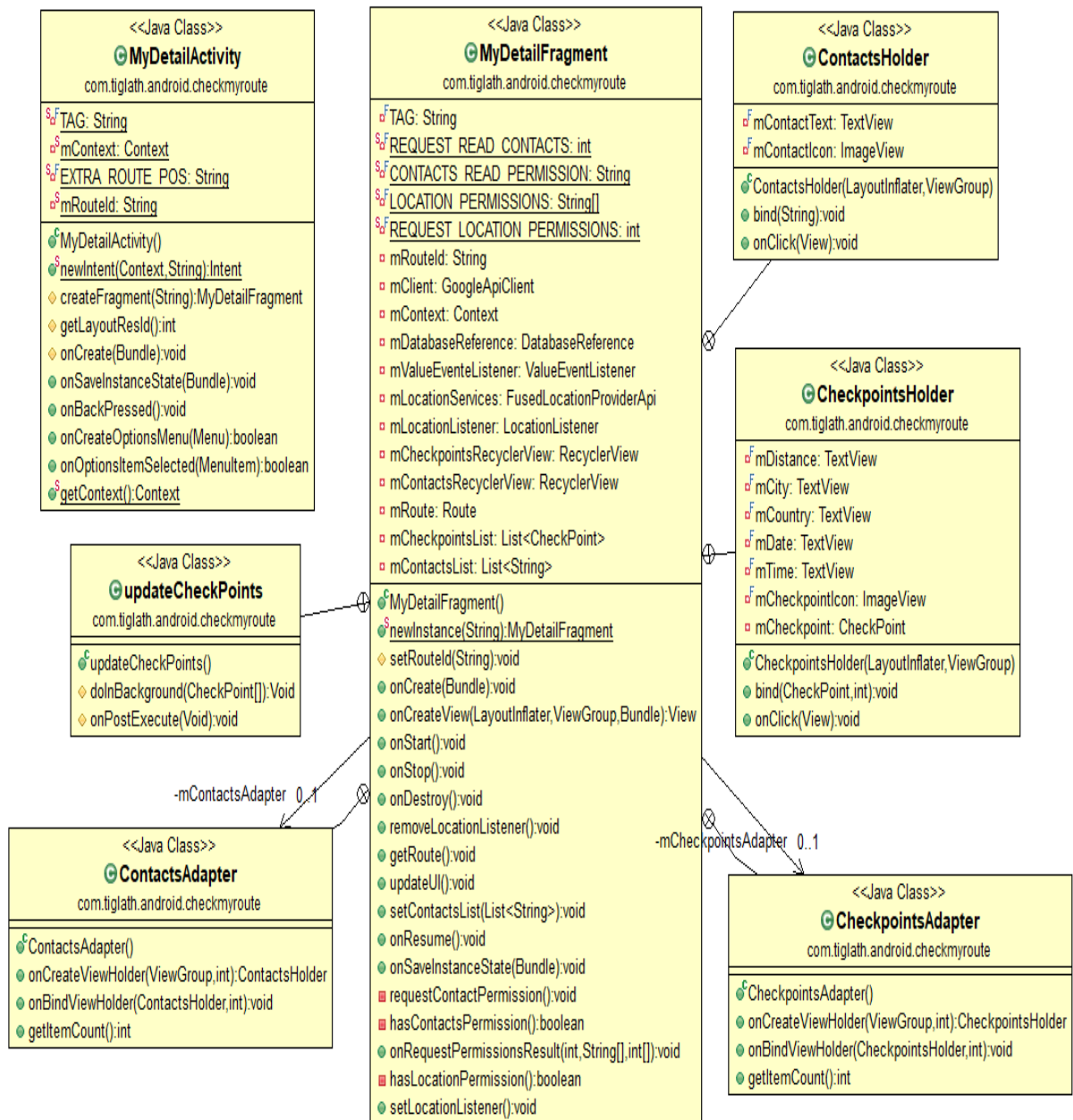


Figura 40: Diagrama de clase – MyDetailActivity.java – MyDetailFragment.java

Nombre de la clase	MyMapActivity.java
Descripción	Activity que contiene el fragmento que gestiona la pantalla del mapa de la ruta.
Paquete	com.tiglath.android.checkmyroute

Nombre de la clase	MapFragment.java
Descripción	Fragmento que carga el mapa de Google Maps con la ruta y última localización.
Paquete	com.tiglath.android.checkmyroute

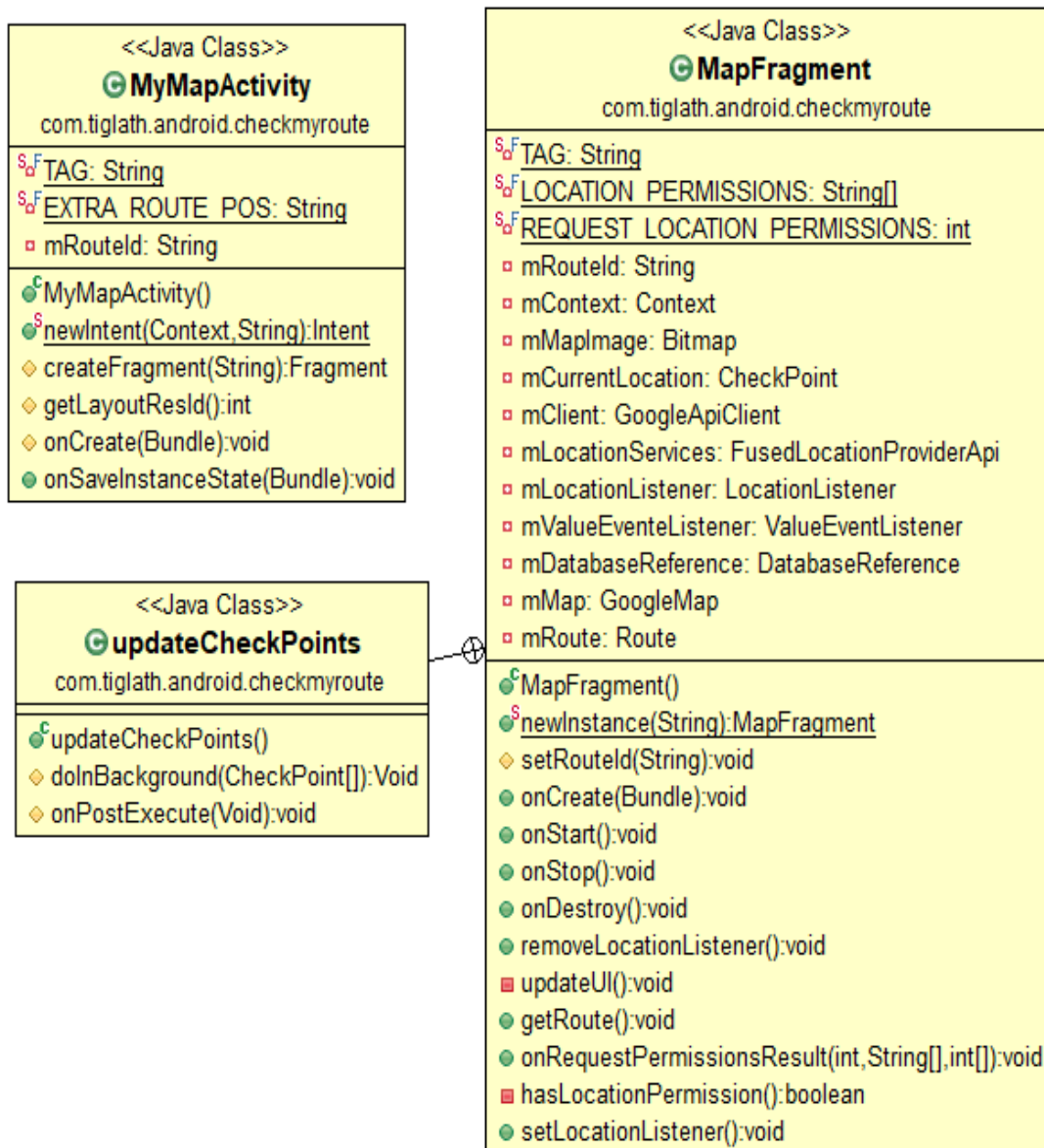


Figura 41: Diagrama de clase – MyMapActivity.java – MapFragment.java

Nombre de la clase	SharedWithMeActivity.java
Descripción	Activity que contiene el fragmento que gestiona la pantalla de las rutas compartidas con el usuario.
Paquete	com.tiglath.android.checkmyroute

Nombre de la clase	SharedWithMeFragment.java
Descripción	Representa la lista con los detalles de las rutas compartidas con el usuario y la actualización de los puntos de control con Firebase..
Paquete	com.tiglath.android.checkmyroute

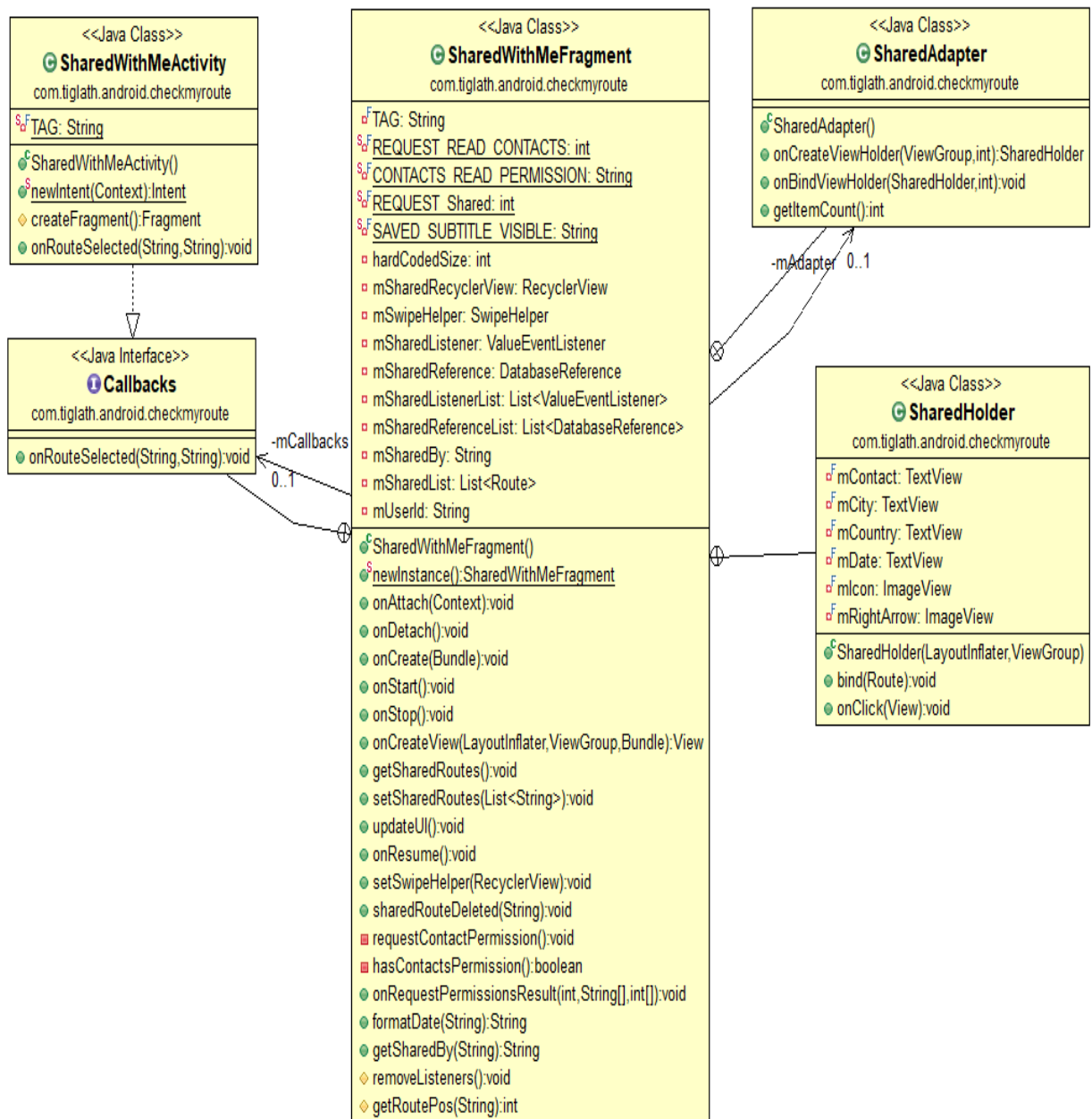


Figura 42: Diagrama de clase – SharedWithMeActivity.java – SharedWithMeFragment.java

Nombre de la clase	SharedDetailActivity.java
Descripción	Activity que contiene el fragmento que gestiona la pantalla del detalle de la una ruta compartida con el usuario
Paquete	com.tiglath.android.checkmyroute

Nombre de la clase	SharedDetailFragment.java
Descripción	Representa la lista con los detalles de la ruta y la actualización de los puntos de control con Firebase.
Paquete	com.tiglath.android.checkmyroute

Nombre de la clase	SharedMapActivity.java
Descripción	Activity que contiene el fragmento que gestiona la pantalla del mapa de la ruta. Extiende de MapFragment.java
Paquete	com.tiglath.android.checkmyroute

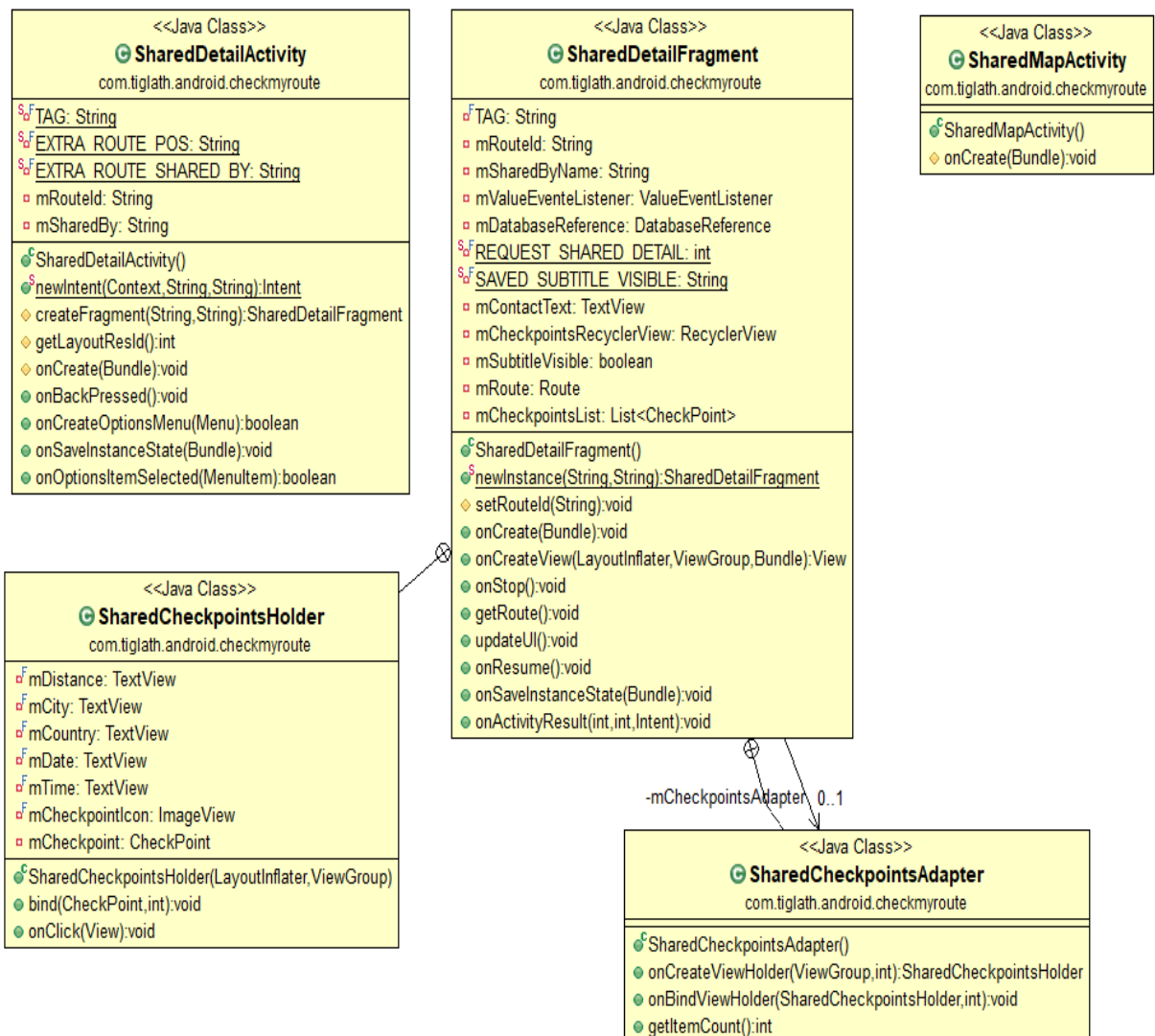


Figura 43: Diagrama de clase – SharedDetailActivity.java – SharedDetailFragment.java

Nombre de la clase	NewRouteActivity.java
Descripción	Activity que gestiona la entrada de datos del usuario, las peticiones a las APIs de Google y la comunicación con Firebase para crear una ruta.
Paquete	com.tiglath.android.checkmyroute

Nombre de la clase	NewMapFragment.java
Descripción	Fragmento para cargar el mapa con la ruta seleccionada por el usuario.
Paquete	com.tiglath.android.checkmyroute

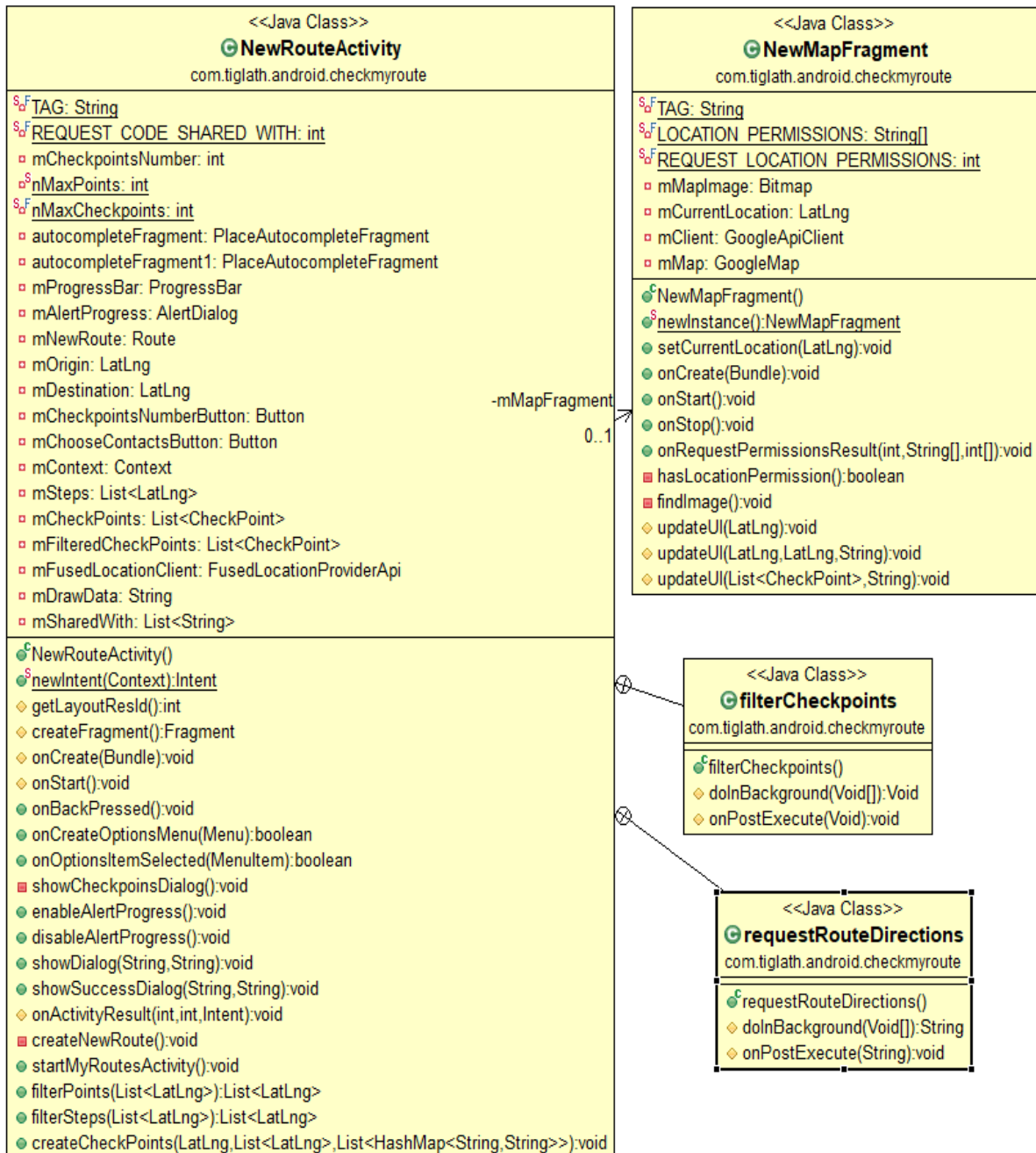


Figura 44: Diagrama de clase – NewRouteActivity.java – NewMapFragment.java

Nombre de la clase	ContactsNewActivity.java
Descripción	Activity que contiene el fragmento para mostrar la lista de contactos con los que el usuario puede compartir la nueva ruta.
Paquete	com.tiglath.android.checkmyroute

Nombre de la clase	ContactsNewFragment.java
Descripción	Fragmento que muestra la lista de contactos que el usuario puede seleccionar para compartir la nueva ruta.
Paquete	com.tiglath.android.checkmyroute

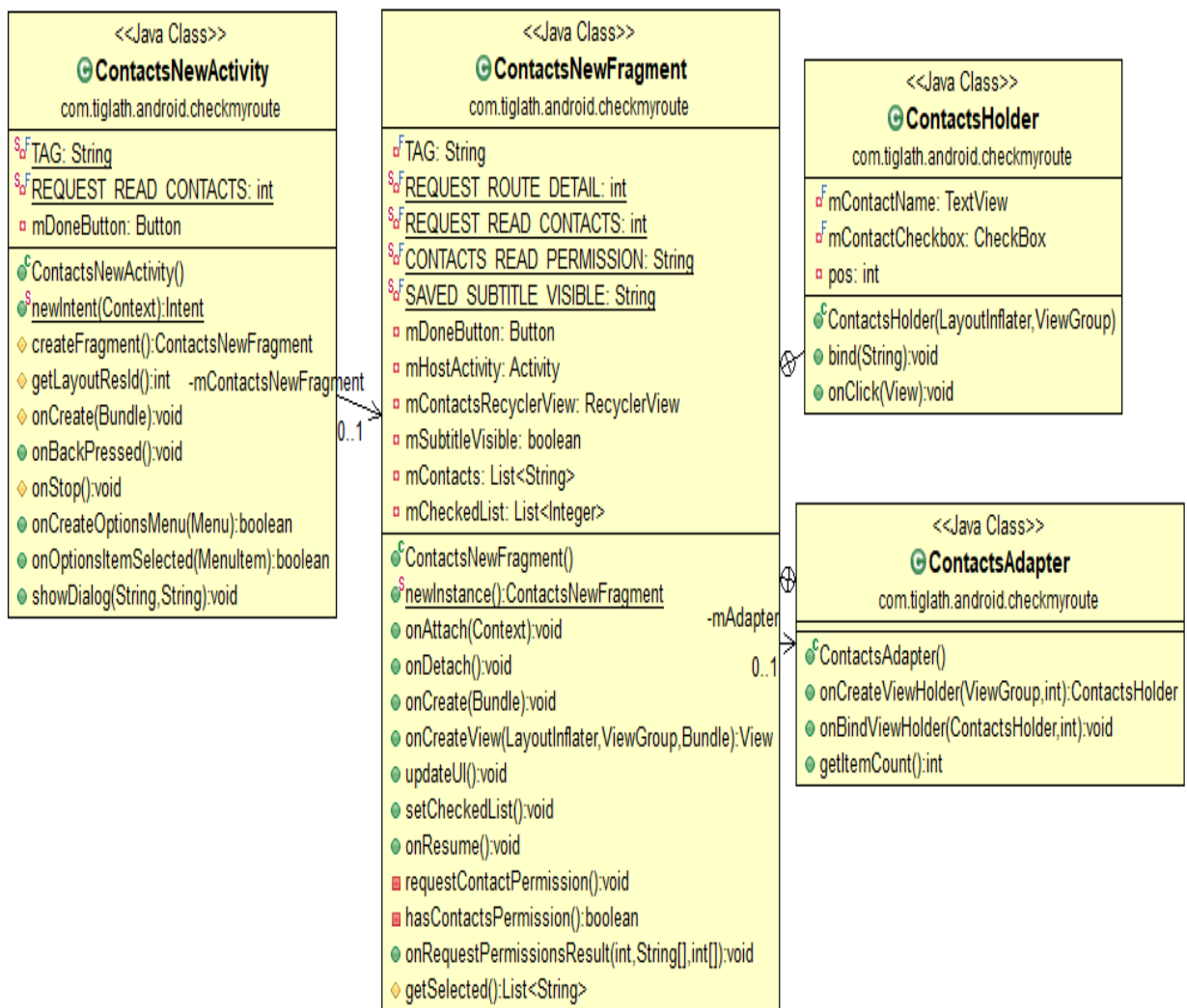


Figura 45: Diagrama de clase – ContactsNewActivity.java – ContactsNewFragment.java

Nombre de la clase	ContactsMenuActivity.java
Descripción	Activity que contiene el fragmento para mostrar la lista de contactos con los que el usuario puede compartir rutas.
Paquete	com.tiglath.android.checkmyroute

Nombre de la clase	ContactsMenuFragment.java
Descripción	Fragmento que muestra la lista de contactos con cuenta activa en la aplicación.
Paquete	com.tiglath.android.checkmyroute

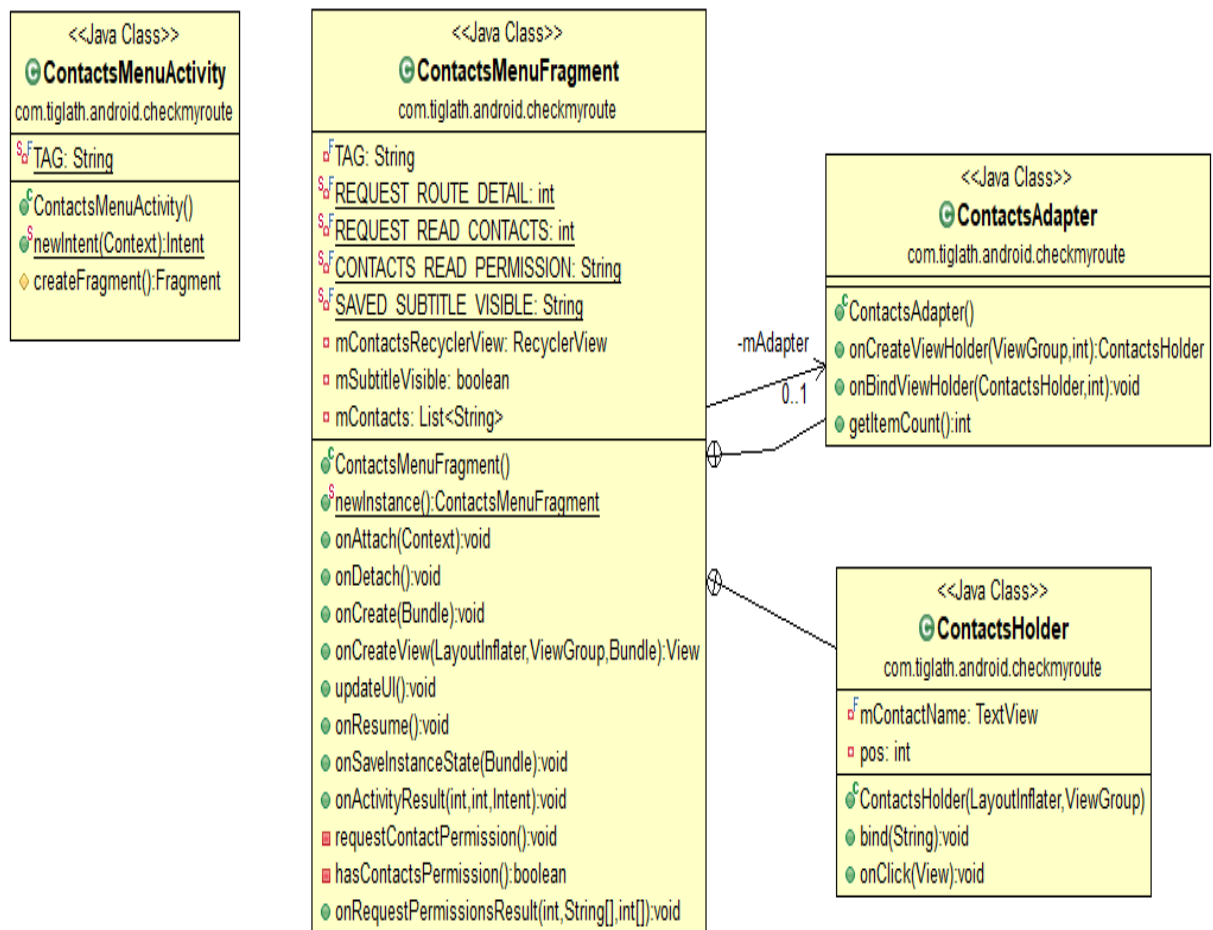


Figura 46: Diagrama de clase – ContactsMenuActivity.java – ContactsMenuFragment.java

Nombre de la clase	ChangeEmailActivity.java
Descripción	Activity para que el usuario pueda cambiar el email asociado a su cuenta en la aplicación.
Paquete	com.tiglath.android.checkmyroute
Nombre de la clase	ChangeEmailFragment.java
Descripción	Fragmento que muestra al usuario la casilla para introducir su nueva dirección de correo.
Paquete	com.tiglath.android.checkmyroute
Nombre de la clase	HelpActivity.java
Descripción	Activity que contiene el fragmento para mostrar la ayuda al usuario
Paquete	com.tiglath.android.checkmyroute
Nombre de la clase	HelpFragment.java
Descripción	Fragmento que muestra la ayuda.
Paquete	com.tiglath.android.checkmyroute

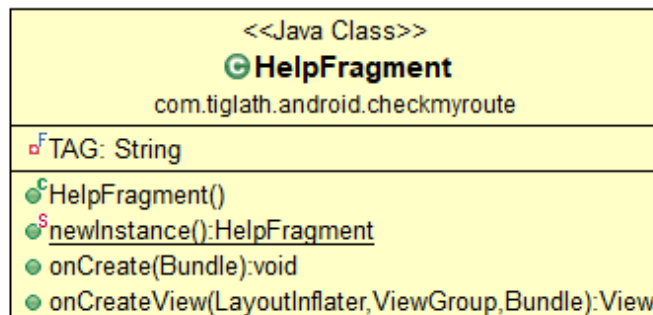
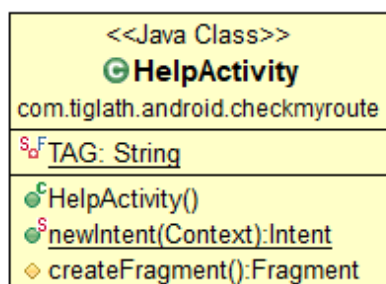
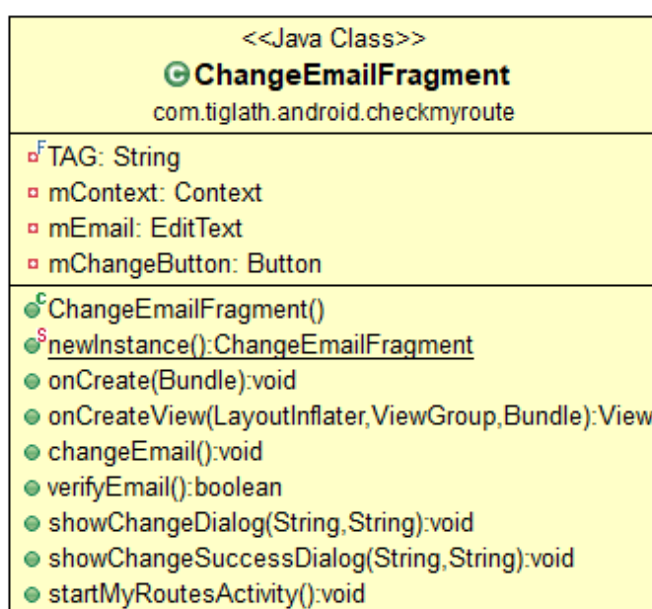
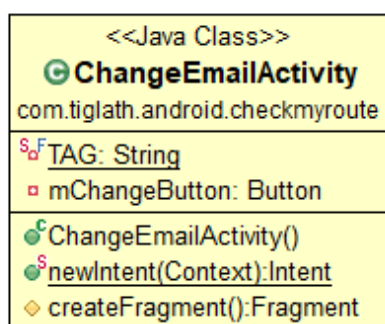


Figura 47: Diagrama de clase – ChangeEmailActivity.java – ChangeEmailFragment.java – HelpActivity.java – HelpFragment.java



Nombre de la clase	SwipeHelper.java
Descripción	Clase que representa y gestiona los botones para finalizar y/o borrar una ruta en las listas de rutas del usuario en las pantallas “MyRoutes” y “SharedWithMe”.
Paquete	com.tiglath.android.checkmyroute

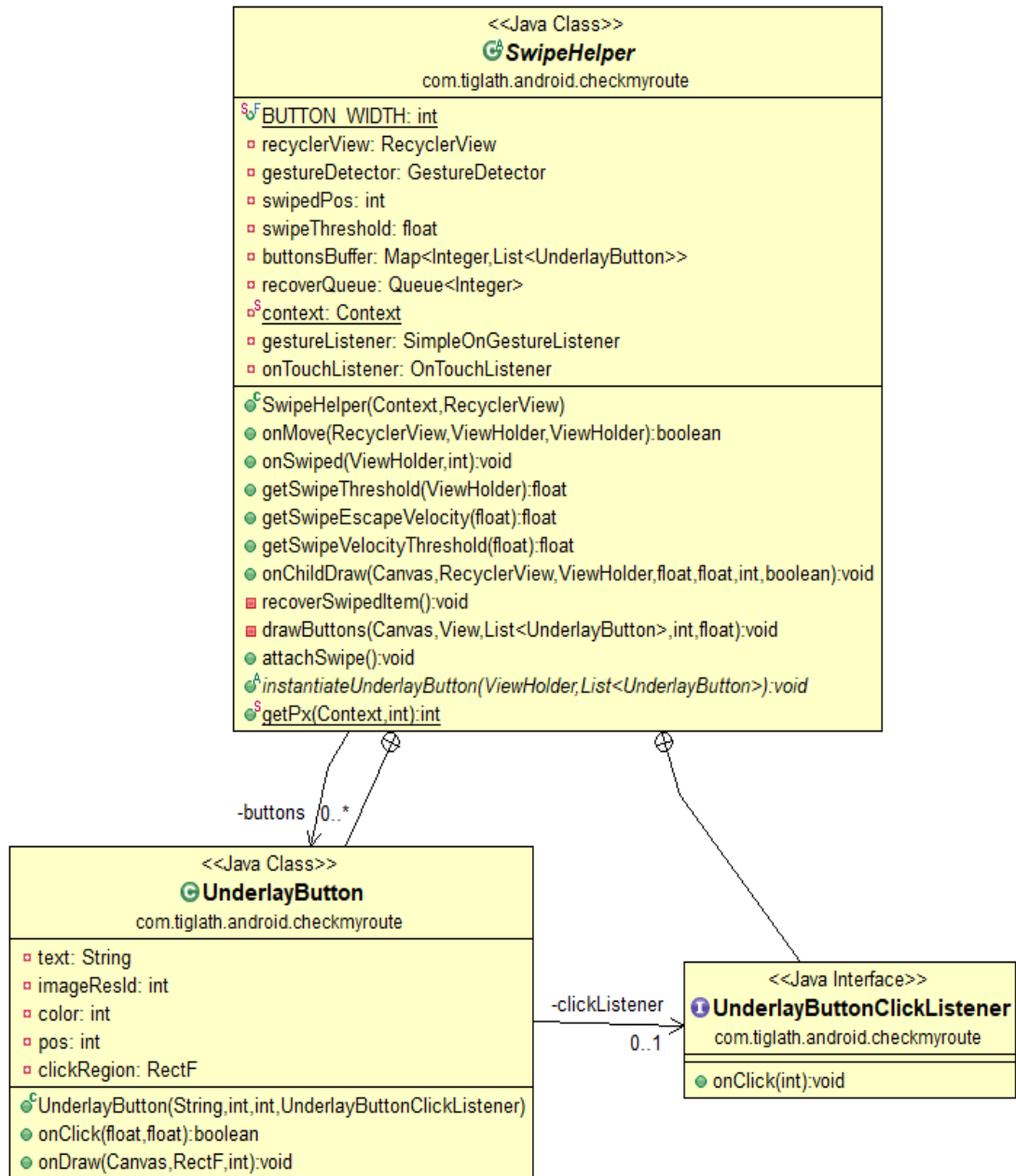


Figura 48: Diagrama de clase – SwipeHelperActivity.java

Nombre de la clase	Route.java
Descripción	Clase que representa el modelo de datos de las rutas.
Paquete	com.tiglath.android.checkmyroute.model

Nombre de la clase	Checkpoint.java
Descripción	Clase que representa el modelo de datos de los puntos de control de la ruta, incluyendo el origen, el destino y la última localización.
Paquete	com.tiglath.android.checkmyroute.model

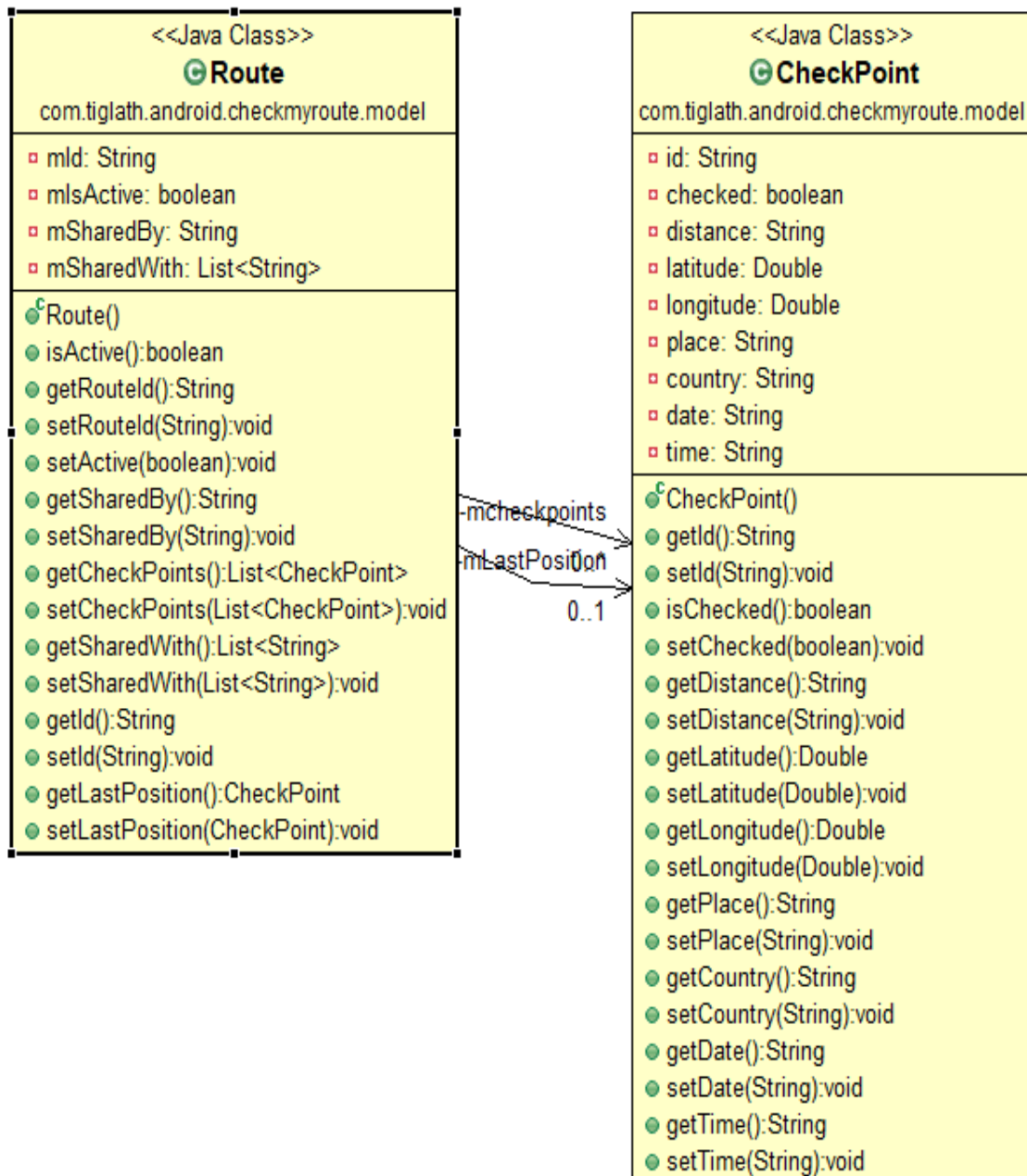


Figura 49: Diagrama de clase – Route.java – Checkpoint.java

Nombre de la clase	RouteFactory.java
Descripción	Clase que se utiliza para gestionar la información de las rutas y la actualización de los puntos de control con respecto a la última localización.
Paquete	com.tiglath.android.checkmyroute.model

Nombre de la clase	LocalContacts.java
Descripción	Clase que se utiliza para recuperar los contactos del usuario que tienen una cuenta activa en la aplicación.
Paquete	com.tiglath.android.checkmyroute.model

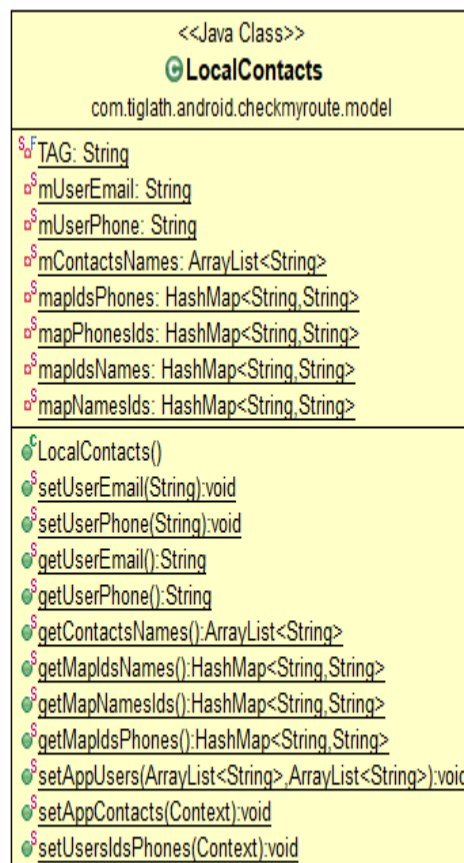


Figura 50: Diagrama de clase – RouteFactory.java – LocalContacts.java

Nombre de la clase	FetchURL.java
Descripción	Clase que se utiliza para contruir las peticiones a las APIs de Google.
Paquete	com.tiglath.android.checkmyroute.apis

Nombre de la clase	DataParserJSON.java
Descripción	Clase que se utiliza para recuperar la información que se obtiene de cada petición a las APIs de Google.
Paquete	com.tiglath.android.checkmyroute.apis

Nombre de la clase	DataRequestConnection.java
Descripción	Clase que se utiliza para realizar peticiones HTTP a las APIs de Google.
Paquete	com.tiglath.android.checkmyroute.apis

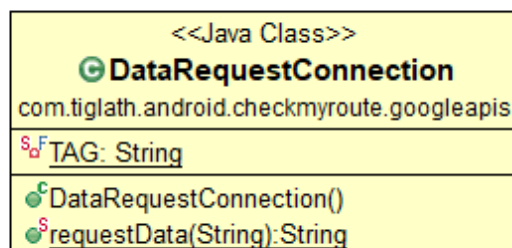
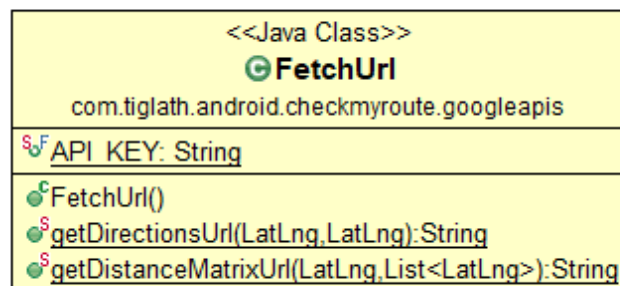


Figura 51: Diagrama de clase – FetchUrl.java – DataParserJSON.java – DataRequestConnection.java

Nombre de la clase	DrawRoute.java
Descripción	Clase que que getiona la representación de la ruta en el mapa.
Paquete	com.tiglath.android.checkmyroute.apis

Nombre de la clase	DrawRouteMaps.java
Descripción	Clase que gestiona el dibujo de la ruta en el mapa.
Paquete	com.tiglath.android.checkmyroute.apis

Nombre de la clase	DrawMarker.java
Descripción	Clase que se utiliza para pintar puntos concretos en el mapa.
Paquete	com.tiglath.android.checkmyroute.apis

Nombre de la clase	RouteDrawer.java
Descripción	Tarea que se ejecuta de forma asíncrona para dibujar la ruta y los marcadores en el mapa.
Paquete	com.tiglath.android.checkmyroute.apis

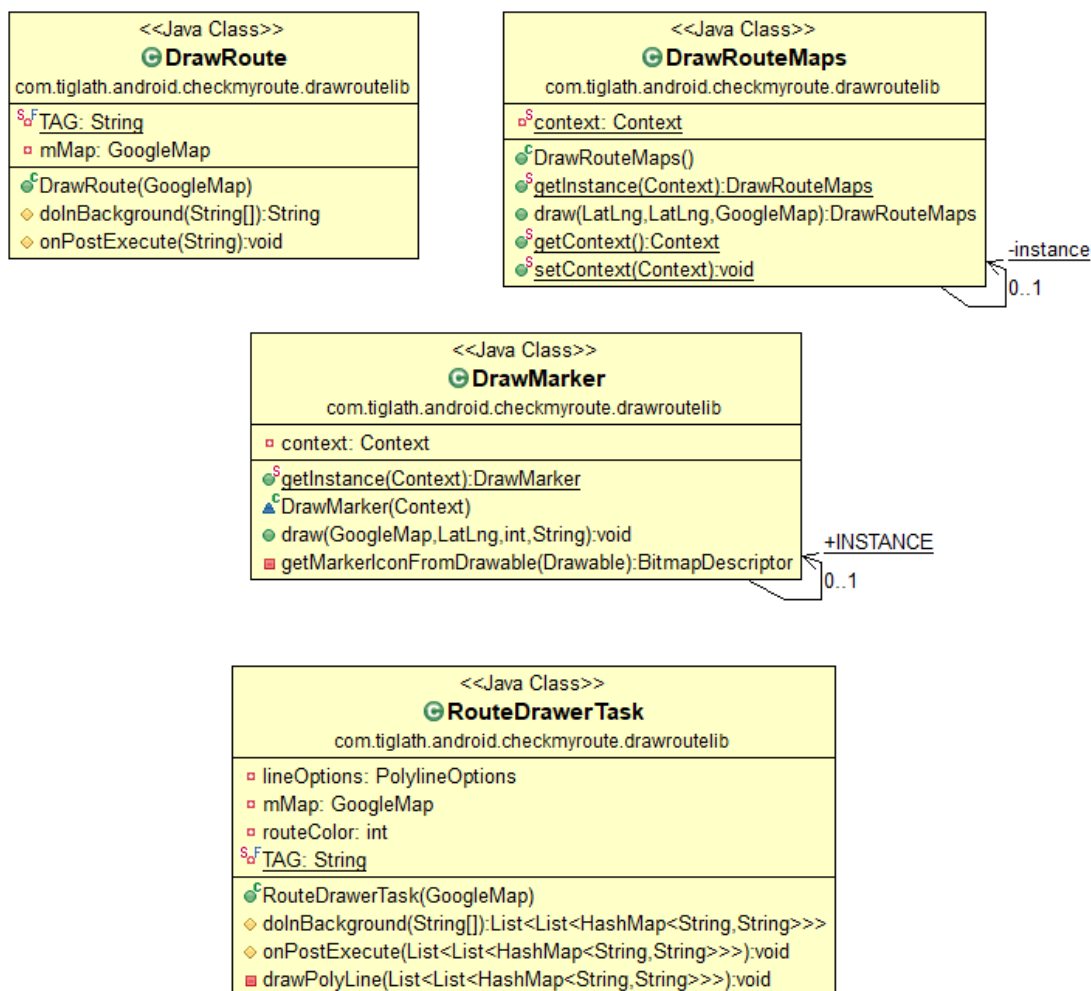


Figura 52: Diagrama de clase – DrawRoute.java – DrawRouteMaps.java – DrawMarker.java - RouteDrawerTask

## 7. Pruebas

Se han desarrollado dos pruebas principales que engloban las funcionalidades de la aplicación. Se ha hecho uso de la librería UIAutomator [18] de Android para simular las acciones del usuario.

### 7.1 SharedWithMeTest

Esta prueba consiste en automatizar el proceso que simula las acciones del usuario para visualizar la actualización de una ruta compartida por uno de sus contactos.

Las acciones que se visualizan al realizar la ejecución de esta prueba son:

1. Hacer login con una cuenta válida.
2. Desplegar el menú de navegación de la aplicación.
3. Pulsar sobre la opción “Shared With Me” para navegar a la pantalla de rutas compartidas con el usuario.
4. Esperar a que aparezca una nueva ruta en la lista de rutas compartidas con el usuario.
5. Pulsar sobre la primera ruta que acaba de aparecer.
6. Esperar a que se cargue la pantalla del detalle de la ruta.
7. Comprobar la actualización de todos los puntos de control.
8. Volver a la pantalla de rutas compartidas con el usuario.
9. Desplegar el menú de navegación.
10. Pulsar sobre la opción “Logout” para salir de la cuenta del usuario.
11. Esperar a que cargue la pantalla de Login.

El diagrama de la clase que implementa la prueba se muestra a continuación.

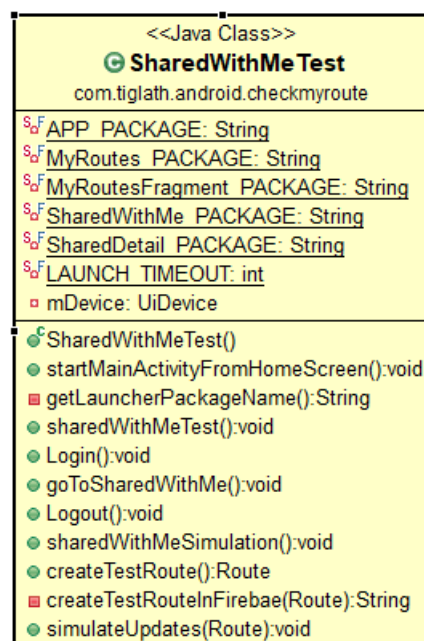


Figura 52: Diagrama de clase – SharedWithMeTest.java

## 7.2 MyRoutesTest

Esta prueba consiste en automatizar el proceso que simula las acciones del usuario para visualizar la actualización de una ruta activa que comparte con otros usuarios. Para comprobar que los puntos de control se actualizan al cambiar la última localización se envían varias localizaciones al dispositivo durante la ejecución de la prueba.

Las acciones que se visualizan al realizar la ejecución de esta prueba son:

1. Hacer login con una cuenta válida.
2. Esperar a que se cargue la pantalla “MyRoutes”.
3. Esperar a que se actualice la lista de rutas con una nueva ruta activa.
4. Pulsar sobre la ruta activa.
5. Esperar a que se cargue la pantalla del detalle de la ruta.
6. Comprobar cómo se actualizan los puntos de control y la última posición cada vez que se recibe una nueva localización.
7. Volver a la pantalla “MyRoutes”.
8. Desplegar el menú de navegación.
9. Pulsar sobre la opción “Logout” para salir de la cuenta del usuario.
10. Esperar a que cargue la pantalla de Login.

El diagrama de la clase que implementa la prueba se muestra a continuación.

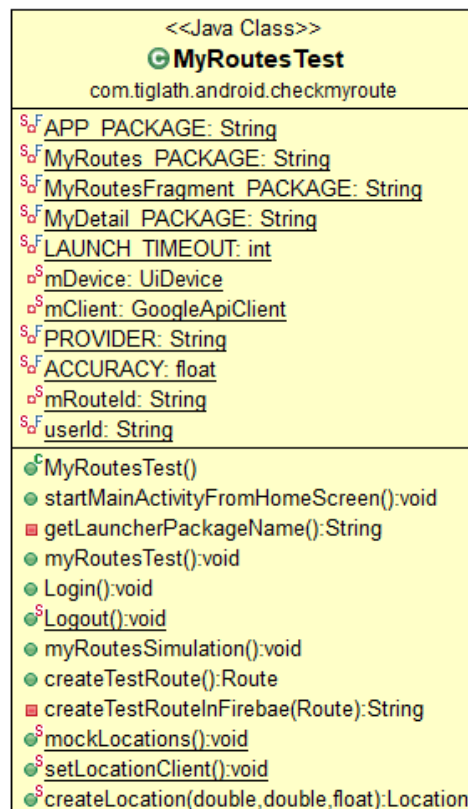


Figura 53: Diagrama de clase – MyRoutesTest.java

## 8. Conclusiones

Uno de los objetivos principales del trabajo final de máster es comprobar la capacidad para planificar, diseñar y desarrollar una aplicación para dispositivos móviles. La gestión del proyecto marca el éxito o fracaso del mismo. Para ello es fundamental concretar de forma eficiente y realista las fases que lo componen.

### 8.1 Lecciones aprendidas

- **Análisis:** Es el punto de partida de cualquier proyecto. Esta fase sirve para conocer el marco de un trabajo de investigación, así como para concretar el problema a resolver y las soluciones ya existentes. El estudio exhaustivo de servicios y aplicaciones similares ayuda a identificar los objetivos que el trabajo pretende alcanzar. Mediante el estudio de la audiencia se pretende conocer las necesidades, expectativas y motivaciones de los usuarios de la aplicación. Gracias a esta información se puede desarrollar un producto que se adapte mejor a los requisitos del usuario.
- **Definición del alcance:** Esta fase permite definir y acotar los productos que se pretenden conseguir al finalizar el proyecto, ayudando a concretar la planificación y las tareas a realizar.
- **Planificación:** Esta fase es esencial para estimar el tiempo que se va a dedicar a cada tarea. Además, permite realizar un seguimiento durante todo el proceso para comprobar si se están cumpliendo los plazos que se marcaron inicialmente.
- **Diseño:** El éxito del producto depende, en gran medida, de esta etapa. Permite conocer cómo debe ser el producto final, concretando todos los detalles sobre la interfaz de usuario, los casos de uso, las funciones y servicios que se ofrecen al usuario, además de cómo se define la arquitectura, el desarrollo y las pruebas para validar la aplicación.

### 8.2 Objetivos alcanzados

A título individual, la valoración global de los objetivos alcanzados es positiva. La mayoría de objetivos definidos al principio del proyecto se han podido alcanzar. El producto final es una aplicación que cumple con todas las funcionalidades definidas en el diseño. No obstante, hay aspectos que se pueden mejorar:

- Permitir al usuario editar los puntos de control.



- Mostar más detalles del punto de control cuando pulsa sobre el mismo en el mapa. Actualmente sólo se muestra la distancia al punto de origen.
- Publicar la aplicación en Google Play Store. La aplicación necesita más tiempo de depuración para garantizar la fiabilidad y robustez necesarios para poder ser publicada con garantías en la tienda oficial.
- Límites de las APIs de Google. Actualmente la aplicación soporta las peticiones de un número considerable de usuarios, pero, debido a los límites en las APIs de Google y al elevado coste que supone para un desarrollador independiente asumir la ampliación de estos límites, la aplicación no puede dar servicio a un número elevado de usuarios activos.

### 8.3 Seguimiento de la planificación.

En las primeras fases del proyecto se ha podido cumplir con la planificación de forma rigurosa. Sin embargo, la fase de desarrollo ha requerido mucho más tiempo del que estaba planificado. Este sobrecoste se debe a que la implementación de todas las funcionalidades requería más tiempo de desarrollo del que inicialmente se planificó. También a que se ha tenido que dedicar más tiempo al aprendizaje de Android para alcanzar el éxito en la implementación.

Por otra parte, la definición inicial del diseño ha facilitado que la fase de desarrollo fuera lo más fluida posible, permitiendo seguir con exactitud los pasos que se tenían que dar en cada momento. Los pocos cambios que ha habido que introducir con respecto al diseño inicial se deben a mejoras que había que abordar, sin que la dedicación se viera afectada por este motivo.

A pesar de que en la fase de desarrollo no se han podido cumplir los plazos marcados, en el resto de etapas del proyecto se han podido alcanzar los objetivos en el tiempo marcado inicialmente.

### 8.4 Líneas de trabajo futuro

Como se ha indicado anteriormente, queda pendiente subir la aplicación a la tienda oficial. Este objetivo se pretende alcanzar en las próximas semanas, después de haber podido probar la aplicación con más rigurosidad, solventar errores que todavía no se conocen e introducir mejoras después de haber recibido el “feedback” por parte de los usuarios que actualmente están haciendo uso de la aplicación.

Otro aspecto que queda pendiente es mejorar la experiencia del usuario en caso de que no tenga una conexión a internet fluida, ya que en estos momentos su experiencia se puede ver perjudicada al cargar mucha información en ciertas pantallas de la aplicación.

## 9. Glosario

**Activity.** En Android, cada una de las pantallas o vistas que forman una aplicación.

**API.** Application Programming Interface. Interfaz de programación de aplicaciones. Conjunto de funciones y métodos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

**APK.** Android Application Package. Paquete de aplicación para el sistema operativo Android. Normalmente es el instalador de una aplicación para dispositivos móviles cuyo sistema operativo es Android.

**Clase.** Paquete o fragmento de código Java que permite crear al menos una instancia (Objeto). Son las plantillas para la definición de objetos.

**Firestore.** Plataforma móvil creada por Google para facilitar el desarrollo de aplicaciones de alta calidad de una forma rápida y eficiente.

**Fragment.** En Android, un fragment representa un comportamiento o una parte de la interfaz de usuario en una activity.

**Geolocalización.** Localización geográfica de un dispositivo.

**GPS.** Global Positioning System. Sistema de Posicionamiento Global. Permite determinar la posición de un objeto, una persona o un vehículo con una alta precisión.

**HTTP.** HyperText Transfer Protocol. Protocolo de transferencia de hipertexto. Es el método más común de intercambio de información en Internet.

**JSON.** JavaScript Object Notation. Formato de texto ligero para el intercambio de datos.

**Layout.** En Android, estructura visual para una interfaz de usuario. Normalmente en formato XML.

**Servicio web.** Web service. Tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones a través de Internet.

**UML.** Unified Modeling Language. Lenguaje de modelado de sistemas software. Sirve para visualizar, especificar, construir y documentar un sistema.

**XML.** eXtensible Markup Language. Lenguaje de marcas o etiquetas para almacenar datos de forma legible.

## 10. Bibliografía

### 10.1 Referencias

[1] Google Maps.

<https://maps.google.es/>

[2] Google Play Store.

<https://play.google.com/store>

[3] Google Firebase.

<https://firebase.google.com/>

[4] Sugar ORM. Librería Android para crear y gestionar bases de datos.

<http://satyan.github.io/sugar/>

[5] Casos de uso.

<https://users.dcc.uchile.cl/~psalinas/uml/casosuso.html>

[6] Árbol de navegación.

<https://www.inboundcycle.com/blog-de-inbound-marketing/bid/195257/el-rbol-web-qu-es-y-para-qu-sirve>

[7] Proto.io. Herramienta para crear prototipos de alto nivel.

<https://proto.io/>

[8] Arquitectura cliente - servidor.

<http://somebooks.es/arquitectura-clienteservidor/>

[9] Firebse realtime database.

<https://firebase.google.com/docs/reference/android/com/google/firebase/database/package-summary>

[10] Firebase Authentication.

<https://firebase.google.com/docs/reference/android/com/google/firebase/auth/package-summary>

[11] Google Maps API.

<https://developers.google.com/maps/>

[12] Google Directions API.

<https://developers.google.com/maps/documentation/directions/>

[13] Google Places API.

<https://developers.google.com/places/>

[14] Google Distance Matrix API.

<https://developers.google.com/maps/documentation/distance-matrix/>

[15] Patrón de diseño Model – View – Controller.

<http://best-practice-software-engineering.ifs.tuwien.ac.at/patterns/mvc.html>

[16] Estructura de proyectos Android.

<https://developer.android.com/studio/projects/>

[17] Android Studio.

<https://developer.android.com/studio/>

[18] UiAutomator.

<https://developer.android.com/training/testing/ui-automator>

## 10.2 Documentación

- Bill Phillips, Chris Stewart, and Kristin Marsicano (2017). “Android Programming. THE BIG NERD RANCH GUIDE (3<sup>rd</sup> Edition)”.

Guia esencial para el desarrollo de aplicaciones en dispositivos Android.

- Robert Clarisó Viladrosa, “Introducción al trabajo final”.

Apuntes Universitat Oberta de Catalunya.

- Flamarich Zampalo, Jordi, “Diseño de productos interactivos multidispositivo”.

Apuntes Universitat Oberta de Catalunya.

- José Ramón Rodríguez, “El trabajo final como proyecto”.

Apuntes Universitat Oberta de Catalunya.

- Google LLC (2018), “Documentation for app developers”.  
<https://developer.android.com/docs/>

Documentación imprescindible para la implementación de la aplicación.

- Stack Exchange Inc. (2018), <https://stackoverflow.com/>

Foro de consulta esencial para cualquier programador.