

**UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ**

**Colegio de Ciencias e Ingenierías**

**Diseño e Implementación de Sistema de  
Vehículo Compartido  
Propuesta tecnológica**

**Esteban Mauricio Flores Andrade**

**Ingeniería en Sistemas**

Trabajo de titulación presentado como requisito  
para la obtención del título de  
Ingeniero en Sistemas

Quito, 15 de mayo de 2019

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ  
COLEGIO DE CIENCIAS E INGENIERÍAS

**HOJA DE CALIFICACIÓN  
DE TRABAJO DE TITULACIÓN**

**Diseño e Implementación de Sistema de Vehículo Compartido**

**Esteban Mauricio Flores Andrade**

Calificación:

Nombre del profesor, Título académico

Fausto Pasmay, M.Sc.

Firma del profesor

---

Quito, 15 de mayo de 2019

## Derechos de Autor

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Firma del estudiante:

---

Nombres y apellidos:

Esteban Mauricio Flores Andrade

Código:

00128218

Cédula de Identidad:

1002812962

Lugar y fecha:

Quito, 15 de mayo de 2019

## RESUMEN

El presente trabajo describe un prototipo de un producto mínimo viable de una aplicación de auto compartido intercantonal e interprovincial que puede escalar a todas las ciudades del Ecuador. El problema identificado son las limitaciones e inconvenientes de las adaptaciones tecnológicas, como grupos de Facebook y WhatsApp, que los usuarios ecuatorianos están utilizando para publicar información con el fin de compartir viajes. El prototipo es una aplicación nativa de Android, *serverless* y que tiene entre sus principales funcionalidades un sistema integrado de mensajería instantánea, sistema de notificaciones, integración con Facebook para proporcionar registro en la aplicación e información de amigos en común, así como integración con Google Maps para visualizar mapas y calcular rutas y tarifas.

Palabras clave: auto compartido, viajes compartidos, Android, Facebook Graph API, NoSQL, Directions API, Firebase, serverless.

## ABSTRACT

This document describes a prototype of a minimum viable product of a ridesharing application that can scale to all cities in Ecuador. The identified problem is the limitations and drawbacks of technological adaptations, such as Facebook and WhatsApp groups, that Ecuadorians are using to post information in order to share rides across the country. The prototype is a *serverless* native Android application with features such as instant messaging, notification system, Facebook integration to provide login, registration capabilities and Facebook common friends information as well as Google Maps integration for visualizing maps and calculate routes and fares.

Key words: carpooling, ridesharing, Android, Facebook Graph API, NoSQL, Directions API, Firebase, serverless.

## TABLA DE CONTENIDO

Introducción.....	8
Desarrollo del tema.....	10
Formulación del problema.....	10
Facebook.....	10
WhatsApp.....	12
Descripción de la solución .....	12
Casos de uso.....	13
Tecnologías .....	14
Android.....	14
Facebook.....	14
Firebase.....	14
Google Maps.....	15
Sistema de mensajería.....	15
Sistema de notificaciones.....	15
Arquitectura.....	15
Servidor.....	15
Base de datos.....	15
MVP.....	16
Funcionamiento .....	16
Actividad de inicio.....	17
Actividad de inicio de sesión.....	17
Actividad de publicación de viaje.....	18
Actividad de búsqueda de viaje.....	19
Actividad de resultados de viaje.....	19
Actividad de detalles del viaje.....	20
Actividad de confirmación de reserva.....	20
Actividad de perfil de usuario.....	21
Actividad de mis viajes.....	22
Actividad de mensajes.....	22
Trabajo Futuro .....	23
Conclusiones .....	25
Referencias Bibliográficas .....	26
Anexo A: Estructura de la Base de Datos.....	28
Anexo B: Interfaz Gráfica de la Aplicación.....	36

## ÍNDICE DE FIGURAS

Figura 1: Casos de uso de la aplicación .....	13
Figura 2: Modelo simplificado del funcionamiento de la aplicación .....	16
Figura 3: Diagrama de flujo de inicio de la aplicación.....	17
Figura 4: Diagrama de flujo de inicio de sesión .....	18
Figura 5: Diagrama de flujo de la actividad de publicación de viajes. Funciones principales	19
Figura 6: Diagrama de flujo de la actividad de publicación de viajes. Funcionalidades de servicio puerta a puerta .....	20
Figura 7: Diagrama de flujo de actividades involucradas en la publicación de viajes .....	21
Figura 8: Diagrama de flujo de interacciones parciales de la actividad de detalles de viaje ...	22
Figura 9: Diagrama de flujo de las actividades detalles de viaje y confirmación de reserva ..	23

## INTRODUCCIÓN

A nivel nacional existe un flujo considerable de migración interna (Moreta, 2015), lo que se traduce en numerosos viajes intercantonales e interprovinciales en los que el usuario se enfrenta a múltiples problemas. La localización de enfoque de este trabajo es la provincia de Imbabura. En ella, los datos de Moreta, tomados del censo de poblacional del 2010, muestran, por ejemplo, que el 39 por ciento de los hogares tiene al menos 1 integrante que emigra a otra provincia con fines educativos o laborales. Sobre la cantidad de viajes, el terminal terrestre de la ciudad de Ibarra, capital de la provincia de Imbabura, registra en promedio 15 000 pasajeros diarios (Bravo, 2019). Para colocar el segundo dato en contexto, la población de la parroquia Ibarra fue de 139 721 habitantes según el último censo (Montero, 2010). Los medios de transporte intercantonales en Imbabura son autobuses, taxis, servicios ejecutivos y plataformas como Cabify y Uber. Las opciones de transporte son limitadas y cada una conlleva problemas para el pasajero. El transporte público interprovincial e intercantonal es ineficiente, tiene frecuencias rígidas, horarios del sistema integrado limitantes, incomodidad para el pasajero y tiempos de viaje mayores que los de un vehículo particular. Los vehículos particulares son costosos e ineficientes con tarifas entre 15USD y 100USD. Por ello los usuarios de los medios de transporte intercantonales e interprovinciales han buscado alternativas como la creación de grupos de conductores y pasajeros en plataformas tecnológicas como WhatsApp y Facebook. Estos grupos suelen estar limitados a conocidos y amigos y su objetivo es conectar a conductores que van a realizar un viaje con los pasajeros que necesitan movilizarse en aquel trayecto. Sin embargo, las plataformas mencionadas también tienen limitaciones pues no fueron creadas para el propósito mencionado.

Dadas las limitaciones y problemas de los medios de transporte interprovincial e intercantonales, así como los de las adaptaciones tecnológicas a las que han recurrido los usuarios se propone el desarrollo de un producto mínimo viable tecnológico que dé solución a

parte de los problemas que experimentan los usuarios en su movilización entre ciudades y se encuentran comunicándose con las adaptaciones tecnológicas mencionadas.

El producto mínimo viable consiste en una aplicación cliente-servidor. Del lado del cliente en forma de una aplicación nativa Android y del lado servidor como base de datos y funciones cloud de la plataforma Firebase. Entre las principales funcionalidades de la aplicación se encuentran módulos como los de reservación y publicación de viajes, mensajería instantánea, integración con API de Facebook para proveer información de amigos en común entre usuarios y API de Google Maps para la planificación y cálculo de rutas. A continuación, se procede a describir el problema y el funcionamiento del producto mínimo viable con más detalle.

## DESARROLLO DEL TEMA

### Formulación del problema

El problema radica en las limitaciones e incomodidades que pueden generar las adaptaciones tecnológicas que los usuarios imbabureños están utilizando para la publicación y reserva de viajes intercantonales e interprovinciales. En la presente sección se describe el proceso de publicación, búsqueda y reserva en plataformas sociales como Facebook y WhatsApp, así como los problemas que se generan al utilizar estas plataformas.

**Facebook.** El escenario descrito corresponde a los casos en donde un grupo de la red social es utilizado para intercambiar información sobre viajes. Los grupos de Facebook que se describen son grupos que sirven a pasajeros que se movilizan entre las ciudades de Ambato, Ibarra, Tena y Quito.

El intercambio de información inicia cuando un usuario, sea conductor o pasajero, publica en el muro de un grupo un texto en donde comunica su intención de viajar y su necesidad de un puesto o viajar y disponer de puestos libres. Esta publicación suele estar acompañada de información sobre la hora y la fecha en la que se pretende realizar el viaje. Rara vez se incluye información sobre precios o aportaciones. Cuando un pasajero realiza un post, un conductor tiene que visualizar el post y comentar o enviar un mensaje al pasajero indicándole su intención de invitarlo a formar parte de su viaje. El proceso de revisar los grupos para conocer si alguien desea un puesto libre es tedioso y con varios pasos intermedios que son una pérdida de tiempo para el conductor pues la publicación de viajes se realiza en un solo hilo del muro de un grupo de la red social y el conductor tendrá que leer y navegar por varios posts del muro del grupo para encontrar un pasajero que desee viajar a su destino en una hora determinada. Por ello, los posts de los pasajeros suelen ser irrelevantes ya que los conductores no se dan el trabajo de buscar personas que deseen viajar con ellos, sino que lo que sucede en

la mayoría de ocasiones es que el conductor realiza una publicación en el muro de un grupo y espera un mensaje de pasajeros o notificaciones de la red social que le comuniquen que algún pasajero ha respondido a su publicación. En algunos casos sucede que un conductor y un pasajero no cuentan con un estado de amistad en la red social por lo que el primer mensaje que envíe uno de los actores será recibido por la bandeja de mensajería de solicitudes de mensajes. Los mensajes que llegan a esta bandeja del actor que recibe el mensaje no disponen de notificaciones por lo que pueden no ser leídos por el receptor. Para visualizar estos mensajes es necesario ingresar a la bandeja de entrada de la mensajería y explícitamente entrar a la sección de solicitudes de mensajes. En dispositivos móviles el proceso de visualización de las solicitudes de mensajes más complicado. El proceso mencionado genera un problema en la conexión de usuarios que no tienen el estado de amigos en la red social pues muchas veces la comunicación por mensajería no llega a efectivizarse.

Los grupos mencionados son moderados por administradores que tienen roles como el de aceptación de miembros nuevos o mediación y eliminación de post con contenidos no relacionados. La seguridad y confiabilidad del grupo depende de esta tarea pues el administrador debe decidir los parámetros sobre los cuales se acepten miembros nuevos. El ambiente de seguridad del grupo suele estar definido con el sentido de pertenencia entre usuarios a grupos o círculos sociales similares. La tarea de moderación del grupo es tediosa y podría estar automatizada.

Otro problema en la adaptación de Facebook es que algunas notificaciones no son útiles y otras llegan a ser fastidiosas para los usuarios. Las notificaciones de Facebook ocurren cuando un amigo de la red social publica contenido en un grupo en común. Aunque no exista interés en el contenido publicado, se producirá una notificación.

**WhatsApp.** Los usuarios también han adaptado WhatsApp para crear grupos de viajes entre ciudades. Una limitación de los grupos de WhatsApp es la cantidad de usuarios que pueden pertenecer a un grupo. Un grupo puede tener un máximo de 256 participantes (WhatsApp, 2019). Otro problema son las notificaciones, ya que son ineficientes. Cada publicación en el grupo genera una notificación para todos los usuarios del grupo. En el campo de seguridad y confianza de usuarios, la información de un perfil de un usuario es bastante limitada y en muchos casos llega a ser solo un número de teléfono.

Un problema común de los grupos de Facebook y WhatsApp es que los usuarios pueden utilizarlos para enviar spam, publicaciones no relacionadas o recolecten datos de usuarios.

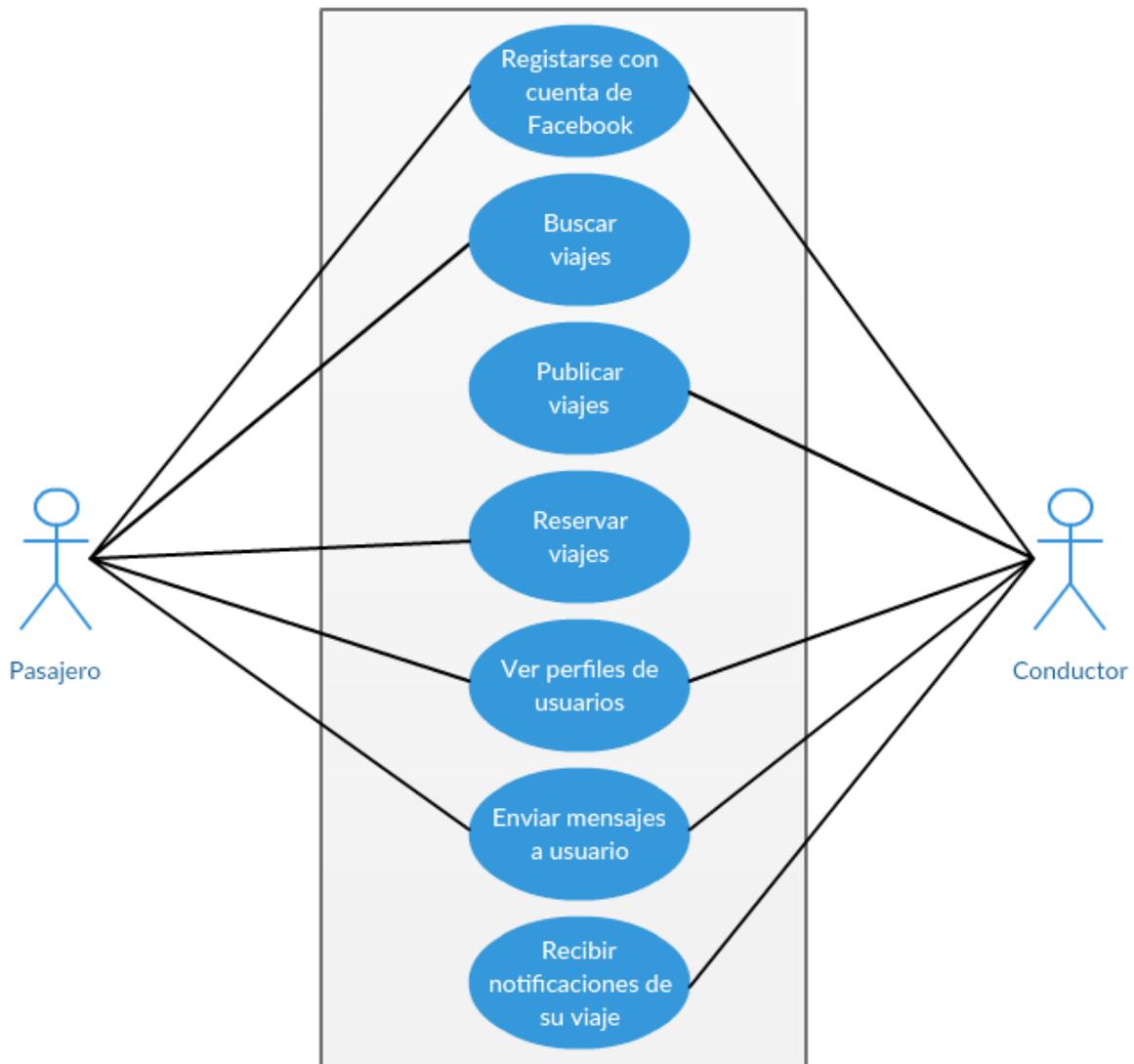
En esta sección se ha mencionado algunos problemas que tienen los grupos de Facebook y WhatsApp cuando son usados como medio de publicación y reservación de viajes. A continuación, se presenta la propuesta de solución a varios de estos problemas.

### **Descripción de la solución**

El proyecto a desarrollar consiste en el desarrollo e implementación de un prototipo de un producto mínimo viable que sirva como plataforma en la que conductores y pasajeros puedan publicar y reservar viajes, solucionando los problemas que se mencionan en la sección anterior. Este prototipo tiene funcionalidades de reserva y publicación de viajes, sistema de notificaciones, mensajería instantánea, cálculo de importe y rutas en servicio de puerta a puerta, información de amigos en común de Facebook entre usuarios e interfaz gráfica multilinguaje (español e inglés).

## Casos de uso

En la Figura 1 se muestran los casos de uso de la aplicación. Los usuarios de la plataforma pueden ser pasajeros y conductores. Cuando un usuario está bajo un rol tienen funcionalidades limitadas que se describen en la figura mencionada. Así, por ejemplo, solo los conductores reciben notificaciones sobre actualizaciones de su viaje.



*Figura 1:* Casos de uso de la aplicación

## Tecnologías

En la presente sección se describen las principales tecnologías que utiliza la plataforma. Se ha decidido realizar una aplicación móvil debido a la cantidad de usuarios en Ecuador con un teléfono inteligente conectado a internet. En Ecuador, tres de cada diez personas cuentan con un teléfono inteligente y se espera que esta cifra se duplique hasta el 2021 (El Universo, 2018). A su vez, ARCOTEL registra 9 millones de cuentas de internet móvil al 2017 (Arcolet, 2018).

**Android.** El sistema es una aplicación nativa que corre bajo la plataforma Android. La versión mínima de Android requerida para instalar la aplicación es API 23, conocida como Marshmallow 6.0 (Android Open Source Project, 2019). Los lenguajes de programación que usa la aplicación en el lado cliente son JAVA y XML. Se ha elegido Android como plataforma del producto mínimo viable porque Android lidera con el 80% el mercado de sistemas operativos de teléfonos móviles en Ecuador (StatCounter, 2018).

**Facebook.** De Facebook se utilizan dos tecnologías, la primera es *Facebook Login* que sirve para proveer a los usuarios una forma conveniente y segura de iniciar sesión en la aplicación con Facebook. La segunda tecnología es *Graph API* que permite, mediante el uso de un token, consultas al grafo social de Facebook. Entre las consultas que se utiliza en la aplicación están el nombre de usuario, foto de perfil y amigos de la red social que utilizan la aplicación.

**Firestore.** Firestore es una plataforma de desarrollo de aplicaciones que funciona como *Backend-as-a-Service*. Los productos de Firestore que se utilizan en el sistema son *Firestore Realtime Database* como base de datos alojada en la nube, *Cloud Functions for Firestore* como funciones servidor que responden a llamadas HTTP, que sirve para programar los *triggers* de la base de datos o enviar notificaciones a usuarios en específico y finalmente *Firestore Cloud Messaging* para manejar notificaciones globales.

**Google Maps.** La aplicación utiliza dos componentes de Google Maps, *Maps SDK for Android* y *Directions API*. El primer componente permite visualizar e interactuar con mapas en un dispositivo Android. El segundo componente es un servicio que calcula rutas entre dos ubicaciones y al cual se accede mediante una solicitud HTTP.

**Sistema de mensajería.** La aplicación hace uso de la característica de tiempo real de la *Real Time Database* de Firebase para configurar un *listener* a cambios en la base de datos y mostrar en tiempo real mensajes recibidos.

**Sistema de notificaciones.** Utilizando *Firebase Cloud Functions*, *Firebase Cloud Messaging* y *Android Services* es posible ensamblar un sistema de notificaciones en los dispositivos Android. *Android Services* permite configurar en el dispositivo un componente de ejecución en segundo plano y de larga duración que escucha recibe los mensajes construidos, dirigidos y enviados por *Firebase Cloud Functions* y *Firebase Cloud Messaging*.

## Arquitectura

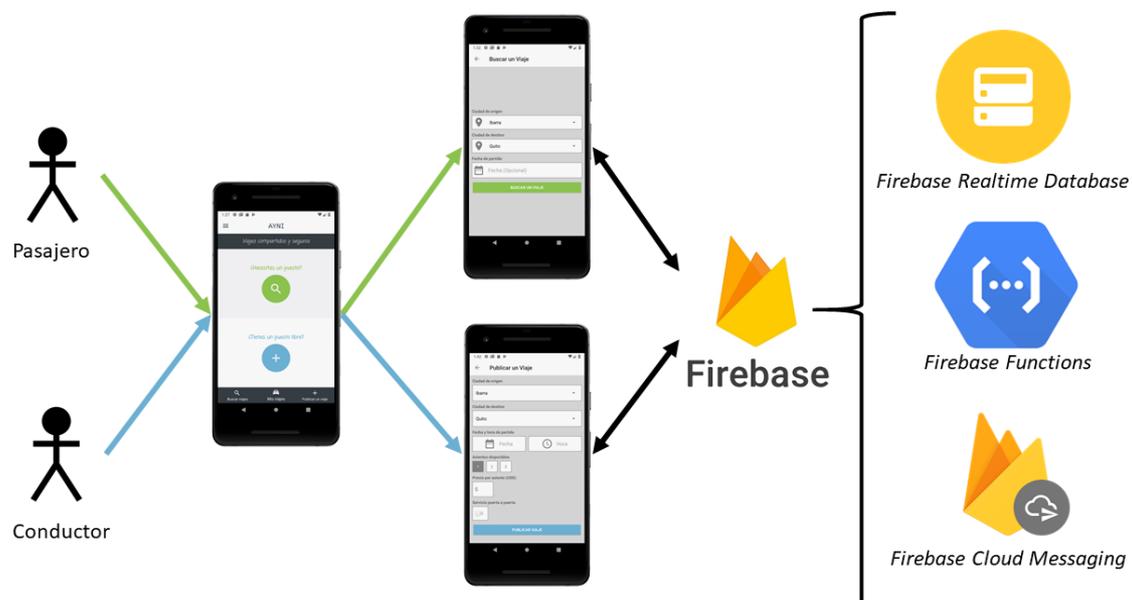
**Servidor.** El lado servidor de la aplicación funciona bajo la plataforma Firebase de Google. Tiene las funciones de almacenamiento de datos (*Real Time Database*), lectura y escritura de datos cuando bajo solicitud, llamadas a eventos (*listeners*) que se registren a cambios en la base de datos, llamados HTTP en forma de API y triggers de la base de datos que se ejecutan cuando se insertan datos. Existen dos triggers en la base de datos. Los dos sirven para enviar notificaciones. En el primer caso cuando un usuario recibe un mensaje y cuando un pasajero reserva el viaje de un conductor.

**Base de datos.** La base de datos de la aplicación es manejada por Firebase bajo el producto de *Real Time Database*. La base de datos es una base de datos no relacional, *document oriented* con la capacidad de mantener los datos sincronizados continuamente entre todos los clientes. Los documentos de la base de datos son objetos JSON. La estructura de la base de datos se presenta en el Anexo A: Estructura de la base de datos.

**MVP.** En el lado del cliente, la aplicación utiliza el patrón de arquitectura *Model View Presenter* (MVP). Algunas clases de java funcionan como *Models*, controlando los datos de la aplicación. Los *layouts* de Android, algunos fragmentos y actividades ejercen el rol de *Views*, presentando información y datos de la forma que decida el *Presenter*. Finalmente, el *Presenter* es el actor intermedio entre *Models* y *Views*. Toda la lógica de presentación se encuentra ejercida por este actor. Este actor está representado en la aplicación por las clases de java que son actividades de Android. Estas clases ejercen el rol de mantener concordancia entre los datos y la información que presentan las vistas, así como también interceptar y reaccionar ante las acciones del usuario.

## Funcionamiento

En la Figura 2 se presenta un modelo simplificado del funcionamiento de la aplicación.



*Figura 2:* Modelo simplificado del funcionamiento de la aplicación

Existe una sola interfaz gráfica tanto para conductores como para pasajeros. Los roles se definen cuando un usuario decide publicar o reservar un viaje. En la interfaz gráfica se intenta que cada rol se identifique con un color como se puede observar en la Figura 2, verde para acciones de búsqueda, reserva y vinculadas a los pasajeros y azul para acciones de

publicación y vinculadas al conductor. A continuación, se detalla las principales actividades de Android y su función en la aplicación. La interfaz gráfica de cada actividad se presenta en la sección Anexo B.

**Actividad de inicio.** Denominada en el sistema como *Home Activity*, representa el punto de entrada de la aplicación. El flujo en el que participa esta actividad se presenta en la Figura 3.

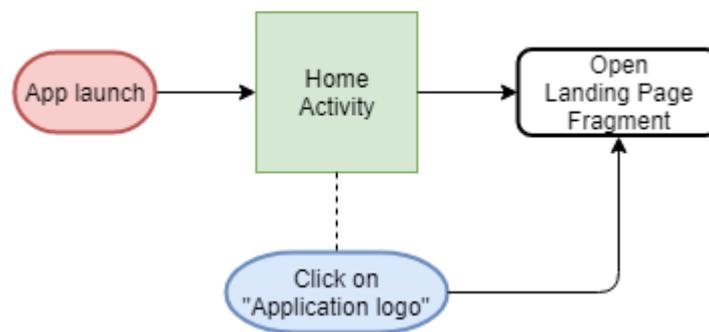


Figura 3: Diagrama de flujo de inicio de la aplicación

**Actividad de inicio de sesión.** Denominada en el sistema como *Sign In Activity*. La principal función de esta actividad es vincular la aplicación con *Facebook Login*, solicitar datos a Facebook y registrar al usuario en la base de datos de la aplicación. Esta actividad maneja y configura un botón de inicio de sesión con Facebook el cual retorna un token de Facebook si el inicio de sesión fue correcto. Este *token* de Facebook es guardado en el disco del dispositivo Android del usuario y no se almacena ni se transfiere a la base de datos por temas de seguridad. Con este token es posible realizar solicitudes de información a *Facebook Graph API*. Luego de un correcto inicio de sesión con Facebook, la actividad solicita información del usuario como el nombre completo, el primer nombre, el id de Facebook, el email del usuario y la lista de amigos de Facebook que también utilizan la aplicación. Con estos se crea el usuario de la aplicación y se carga esta información a la base de datos de la aplicación si los datos del usuario todavía no están registrados en la base de datos. El flujo en donde esta actividad ejerce el rol principal esta descrito en la Figura 4.

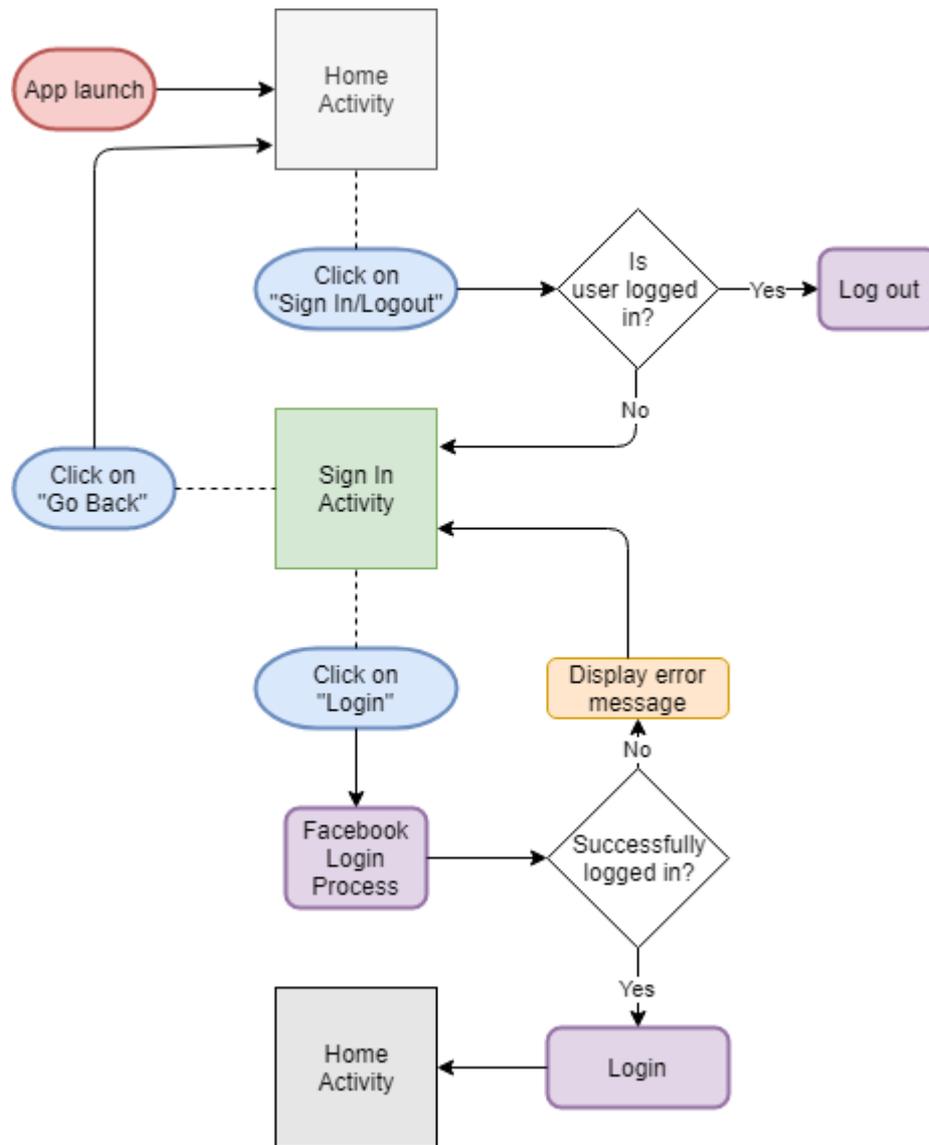


Figura 4: Diagrama de flujo de inicio de sesión

**Actividad de publicación de viaje.** Denominada *Post Activity* en la aplicación. Esta actividad es un formulario en donde el usuario ingresa los datos del viaje que desee publicar. Esta actividad tiene un componente visual en forma de *switch* que habilita la opción de publicar un viaje puerta a puerta. Para publicar un viaje puerta a puerta la actividad requiere que usuario ingrese su punto de partida y su punto de destino. La actividad *Post Activity* llama a otra actividad denominada *Select Map Location Activity* para que el usuario pueda seleccionar puntos en el mapa. Los flujos principales de la actividad *Post Activity* se presentan en la Figura 5 y Figura 6.

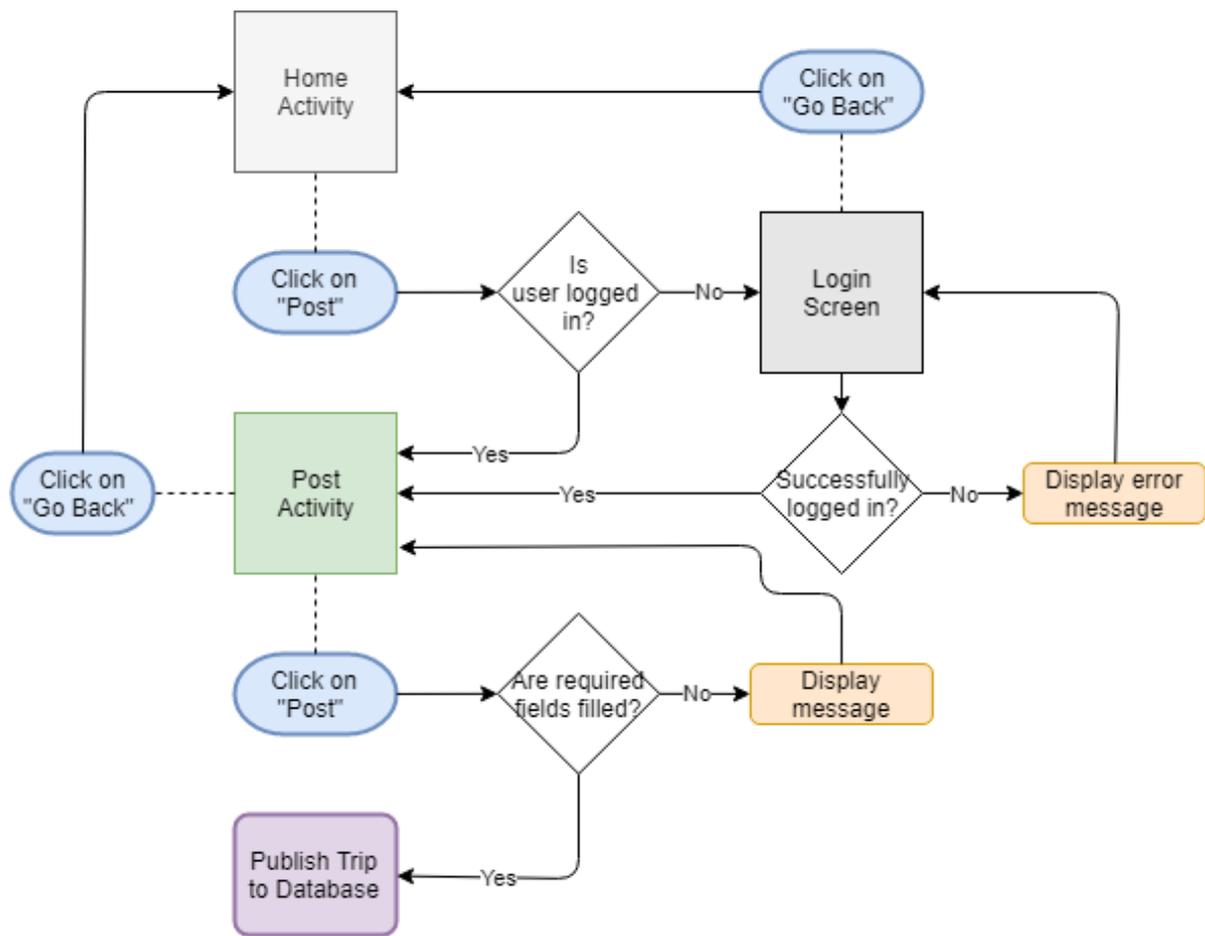


Figura 5: Diagrama de flujo de la actividad de publicación de viajes. Funciones principales

**Actividad de búsqueda de viaje.** Denominada *Search Activity* en la aplicación. Esta actividad consiste en un formulario de 3 campos en donde el usuario selecciona la ciudad de origen, destino y la fecha en la que desea realizar un viaje. Esta actividad entonces invoca a la actividad de resultados de viaje, como se muestra en la Figura 7.

**Actividad de resultados de viaje.** Denominada *Search Results Activity*, presenta los resultados de búsqueda al usuario. Esta actividad realiza una solicitud a Firebase de los viajes que se encuentran registrado bajo el nodo origen-destino, año, mes y día solicitados. Debido a la forma en que está estructurada la base de datos, la respuesta de Firebase contiene los viajes ordenados por fecha y hora de forma ascendente por lo que esta actividad no tiene que realizar operaciones de ordenamiento de viajes. Esta actividad involucrada en el funcionamiento que se describe en la Figura 7.

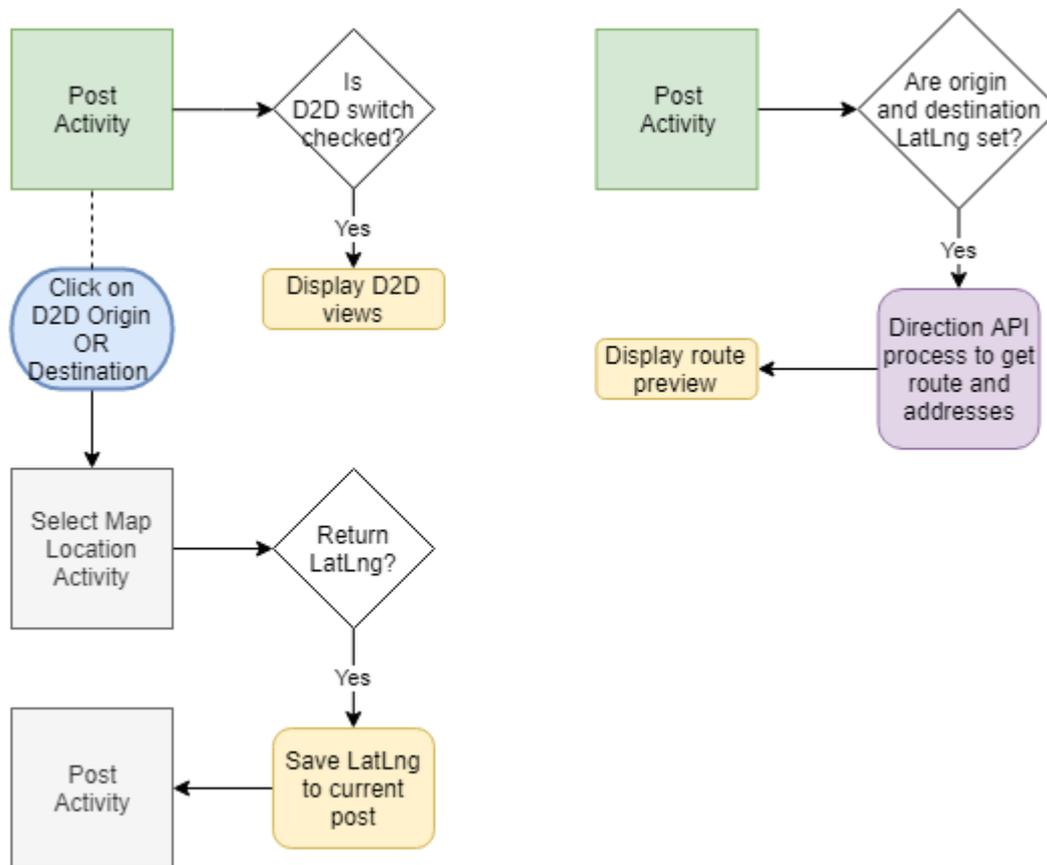


Figura 6: Diagrama de flujo de la actividad de publicación de viajes. Funcionalidades de servicio puerta a puerta

**Actividad de detalles del viaje.** Denominada *Trip Details Activity*. Esta actividad presenta detalles de un viaje. Sus interacciones con otras actividades se describen en la Figura 8 y Figura 9.

**Actividad de confirmación de reserva.** Denominada *Booking Confirmation Activity*. Esta actividad tiene la función de reservar el viaje, así como calcular la tarifa de un viaje puerta a puerta si se habilita esta opción. El cálculo de la tarifa puerta a puerta se lo realiza tomando en cuenta la distancia y el tiempo que tiene que desviarse el conductor de su trayecto original. Cuando se verifica la reserva, la aplicación vuelve a consultar si existen asientos disponibles para evitar problemas de concurrencia. Solo después de la segunda consulta y si existen asientos libres, la aplicación procede a reservar el viaje. Esta actividad participa en el flujo que se describe en la Figura 9.

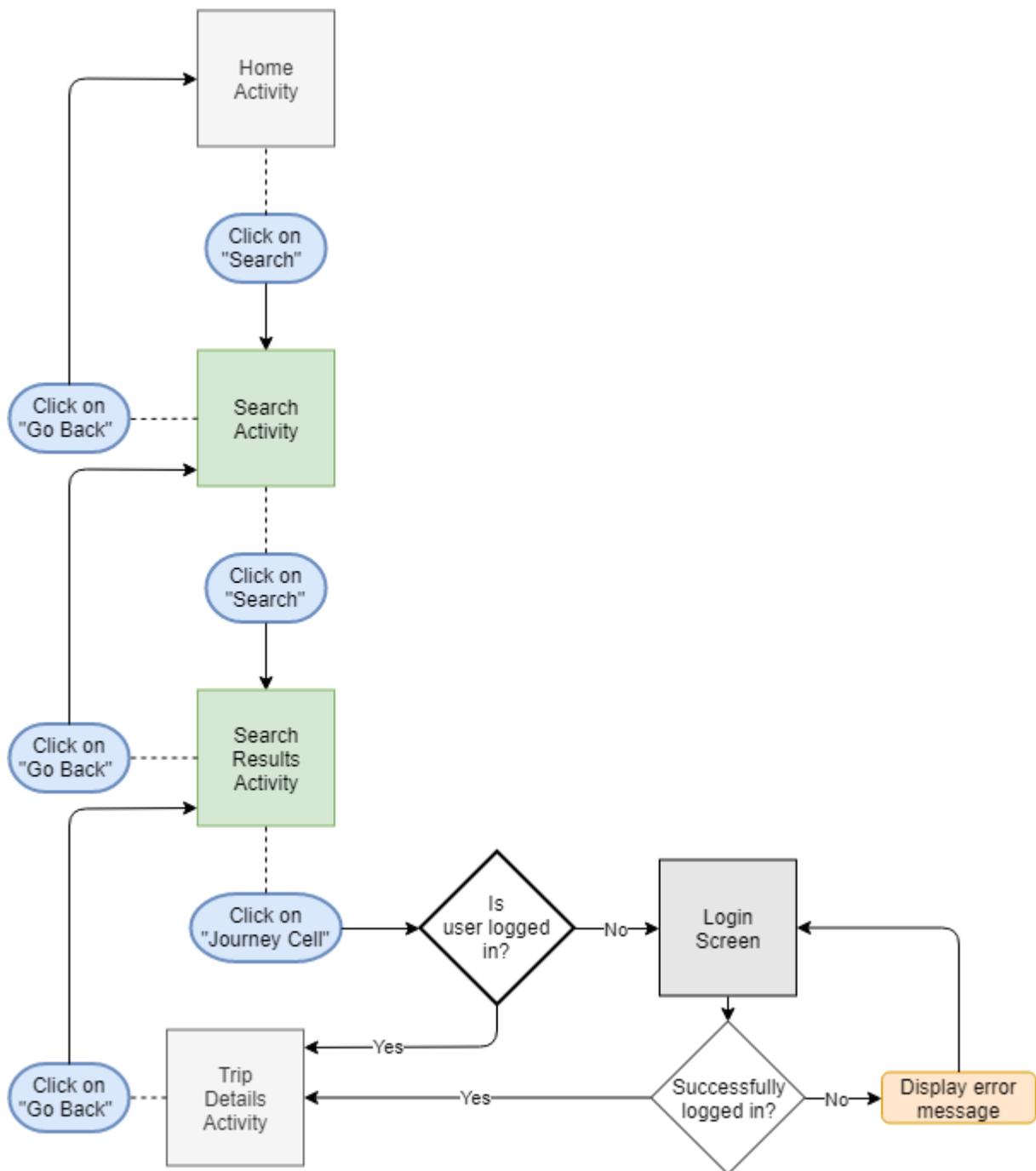


Figura 7: Diagrama de flujo de actividades involucradas en la publicación de viajes

**Actividad de perfil de usuario.** Denominada *User Profile Activity*, esta actividad muestra el perfil de un usuario solicitando la información del mismo a la base de datos. Esta actividad sirve tanto como para mostrar perfiles de pasajeros y perfiles de conductores. La información de usuario que esta actividad expone es una foto de perfil, el nombre, el número

de amigos en común, así como imágenes miniaturas de las fotos de perfil de los amigos en común. Esta actividad también contiene el botón de enviar mensaje a al usuario. Si se da clic en este botón se abre la actividad *Message Activity*.

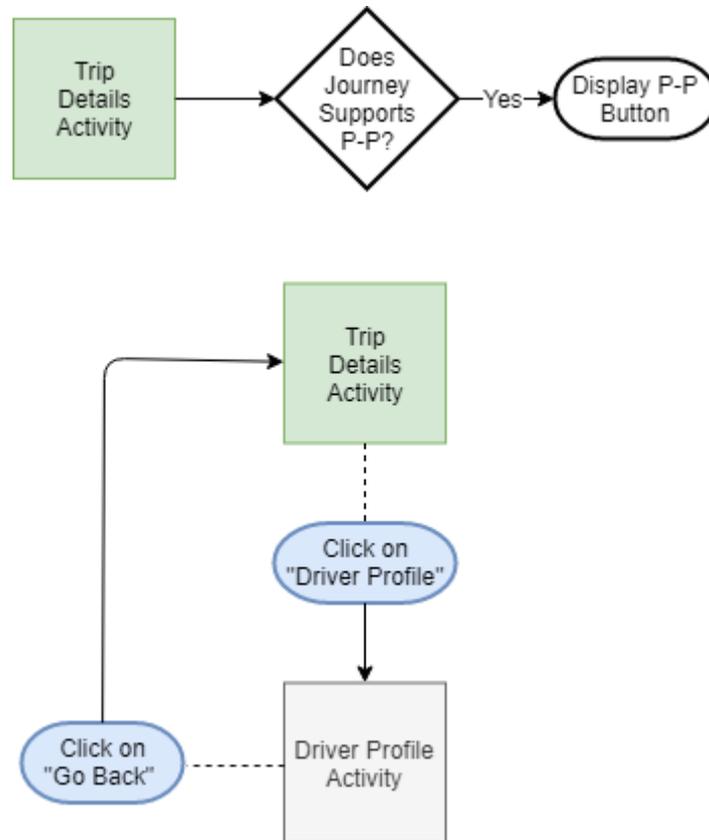
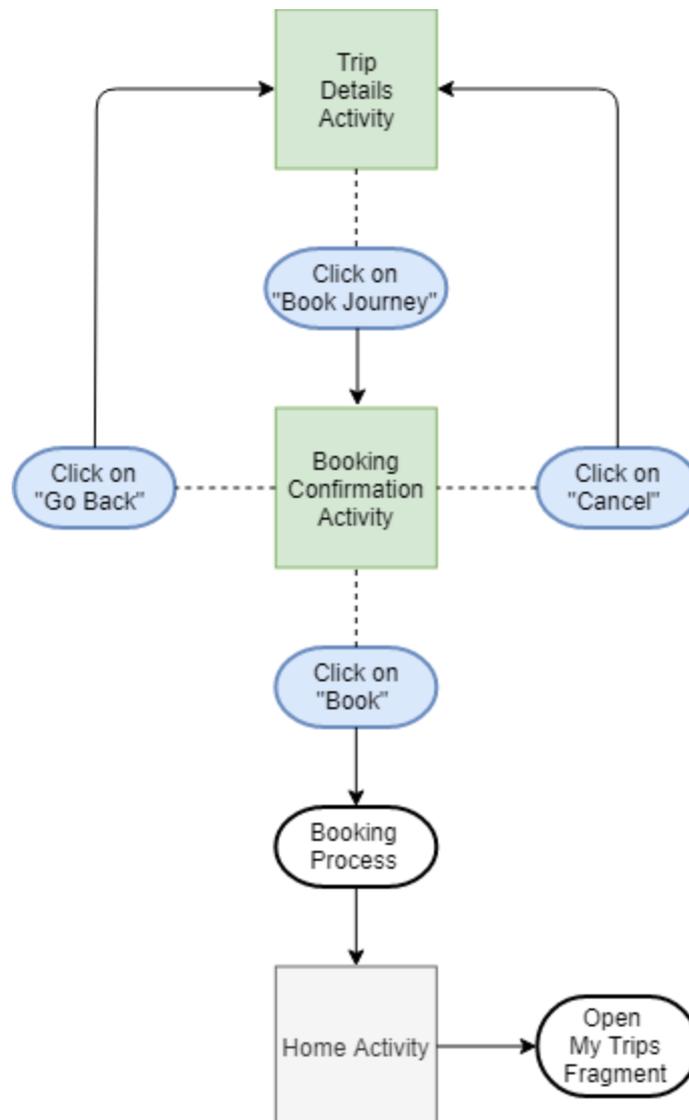


Figura 8: Diagrama de flujo de interacciones parciales de la actividad de detalles de viaje

**Actividad de mis viajes.** Denominada *My Trips Activity*. Esta actividad muestra los viajes que corresponden al usuario del dispositivo. Como se muestra la sección anexos B, su interfaz gráfica señala al usuario su rol en el viaje, este puede ser conductor o pasajero. Cada vez que un usuario reserve un viaje de un conductor, el conductor recibirá una notificación.

**Actividad de mensajes.** Denominada *Messages Activity*. La actividad cumple la función de mostrar los mensajes entre dos usuarios y proveer la capacidad de enviar los mensajes entre los usuarios. El momento que un usuario inicia esta actividad, la actividad registra un *listener* al nodo correspondiente en la base de datos. Cada vez que se escriba un

mensaje en este nodo, este mensaje será recibido por el *listener* mencionado y será mostrado al usuario. Cada vez que un usuario recibe un mensaje también recibirá una notificación.



*Figura 9:* Diagrama de flujo de las actividades detalles de viaje y confirmación de reserva

## Trabajo Futuro

El prototipo de la aplicación tiene mejoras pendientes, este trabajo futuro se describe en esta sección.

La primera mejora es proveer funcionalidades de edición y eliminación de viajes. Otra mejora de alta prioridad es reemplazar la selección de ubicación en un mapa por ingreso de una dirección mediante texto y autocompletado para mejorar la experiencia de usuario. Se

recomienda también implementar la creación de un servicio que actualice los amigos de Facebook de un usuario cada cierto período de tiempo pues los amigos de Facebook de un usuario, en la versión que se presenta de la aplicación, solo se extraen cuando el usuario se registra por primera vez. Ya que los amigos de Facebook solo son aquellos que también usan la aplicación, conforme más usuarios usen la aplicación, más usuarios estarán disponibles como respuesta a la solicitud de amigos al *Graph API* de Facebook. Otra mejora a la experiencia de usuario que se recomienda es utilizar la existente librería Shimmer de Facebook que muestra animaciones personalizadas mientras el contenido de un layout se carga de internet. Sobre la organización de los datos de viajes que se muestran al usuario, una recomendación de mejora es modificar los resultados de búsqueda para que los primeros resultados sean publicaciones de amigos o de amigos de amigos en Facebook. Así mismo, mostrar un mensaje cuando no se encuentren resultados y habilitar los resultados de búsqueda para que estos sigan desplegándose sin límite a la fecha de búsqueda. Finalmente, una mejora a la experiencia de usuario puede ser limitar y mejorar el sistema de notificaciones pues estas no están optimizadas, si un usuario recibe un mensaje mientras está en la actividad de mensajes igual recibirá una notificación. Así mismo, los pasajeros deben recibir notificaciones si su viaje es modificado o eliminado.

## CONCLUSIONES

La idea del prototipo del producto mínimo viable fue desarrollada de una forma exitosa. Se obtuvo una aplicación nativa de Android con funcionalidades de mensajería instantánea, sistema de notificaciones, integración con Facebook y Google Maps, así como una interfaz de usuario sencilla, intuitiva y funcional en dos idiomas: inglés y español.

La aplicación logró solucionar algunos problemas de las adaptaciones tecnológicas mencionadas. Los usuarios son capaces de publicar y reservar viajes simples o viajes de puerta a puerta. Se ha logrado un sistema eficiente y conveniente de comunicación y de notificaciones. Entre las pruebas realizadas se concluye que la aplicación es robusta y ha funcionado en múltiples dispositivos con diferentes versiones de sistema operativo y varios tamaños y resoluciones de pantalla. Una característica final de la aplicación desarrollada es la facilidad de expansión y escalamiento para ser usada en otras zonas del país o a nivel nacional.

Entre los problemas y retos encontrados se encuentran inconvenientes como cuando un usuario dispone de una conexión de internet limitada o de mala calidad, así como el posible consumo en exceso de los datos de internet de un usuario debido a la no optimización de cantidad de datos que la aplicación descarga de la base de datos.

Entre los riesgos y amenazas hacia la funcionalidad y escalabilidad de la aplicación se encuentran la modificación de políticas de datos de la red social Facebook, así como el cambio de tarifas de Google Maps o de Firebase.

Finalmente, se espera que esta aplicación pueda ser cargada a la tienda de aplicaciones de Google y empiece a funcionar como un producto mínimo viable enfocado en la población objetivo inicial. Así mismo, seguir trabajando en las modificaciones, entre otras, como las que se mencionan en la sección de trabajo futuro.

## REFERENCIAS BIBLIOGRÁFICAS

- AlyuDeen, M. (16 de Marzo de 2019). *Serverless notifications with Cloud Functions for Firebase*. Obtenido de Medium: AndroidPub: <https://android.jlelse.eu/serverless-notifications-with-cloud-functions-for-firebase-685d7c327cd4>
- Android Open Source Project. (2019). *Codenames, Tags, and Build Numbers*. Obtenido de Android: <https://source.android.com/setup/start/build-numbers>
- Arcolet. (Marzo de 2018). *ARCOTEL registró alrededor de 9 millones de cuentas de internet móvil hasta 2017*. Recuperado el Diciembre de 2018, de ARCOTEL: <http://www.arcotel.gob.ec/arcotel-registro-alrededor-de-9-millones-de-cuentas-de-internet-movil-hasta-2017/>
- Bravo, M. P. (28 de Abril de 2019). Nueva tarifa para ingresar a la terminal desde en mayo. *El Norte*. Recuperado el 28 de Abril de 2019, de <https://www.elnorte.ec/ibarra/nueva-tarifa-para-ingresar-a-la-terminal-rige-en-mayo-NM385924>
- Cervone, F. (27 de Febrero de 2017). *Model-View-Presenter: Android guidelines*. Obtenido de Medium: <https://medium.com/@cervonefrancesco/model-view-presenter-android-guidelines-94970b430ddf>
- El Universo. (Agosto de 2018). Tres de cada diez personas cuentan con smartphone en Ecuador. *El Universo*. Recuperado el Diciembre de 2018, de <https://www.eluniverso.com/noticias/2018/08/06/nota/6893255/tres-cada-diez-personas-cuentan-smartphone>
- Facebook. (2015). *Shimmer for Android*. Obtenido de Github: <https://github.com/facebook/shimmer-android>
- Facebook. (2019). *Facebook Login*. Obtenido de Facebook for developers: <https://developers.facebook.com/docs/facebook-login/>
- Facebook. (2019). *Graph API*. Obtenido de Facebook for developers: <https://developers.facebook.com/docs/graph-api/>
- Firebase. (2019). *Cloud Functions for Firebase*. Obtenido de Firebase Documentation: <https://firebase.google.com/docs/functions>
- Firebase. (2019). *Firebase Cloud Messaging*. Obtenido de Firebase Documentation: <https://firebase.google.com/docs/cloud-messaging>
- Firebase. (2019). *Firebase overview*. Obtenido de Firebase: <https://firebase.google.com/>
- Firebase. (2019). *Firebase Realtime Database*. Obtenido de Firebase Documentation: <https://firebase.google.com/docs/database>
- Google. (2019). *Directions API*. Obtenido de Google Maps Platform: <https://developers.google.com/maps/documentation/directions/intro>
- Google. (2019). *Maps SDK for Android: Overview*. Obtenido de Google Maps Platform: <https://developers.google.com/maps/documentation/android-sdk/intro>
- Google. (2019). *Services*. Obtenido de Developers Documentation: <https://developer.android.com/guide/components/services>

- Google. (2019). *SharedPreferences*. Obtenido de Android Developers Documentation:  
<https://developer.android.com/reference/android/content/SharedPreferences>
- Montero, W. (2010). *Datos Demográficos*. Obtenido de Gobierno Autónomo Descentralizado Municipal San Miguel de Ibarra:  
<https://www.ibarra.gob.ec/web/index.php/ibarra1234/informacion-general/660-datos-demograficos>
- Moreta, M. (14 de Abril de 2015). Tres provincias tienen mayor emigración. *El Comercio*. Recuperado el 5 de Diciembre de 2018, de  
<https://www.elcomercio.com/actualidad/provincias-mayor-emigracion-ecuador.html>
- StatCounter. (Noviembre de 2018). *Mobile Operating System Market Share Ecuador*. Recuperado el 5 de Diciembre de 2018, de StatCounter GlobalStats:  
<http://gs.statcounter.com/os-market-share/mobile/ecuador>
- WhatsApp. (2019). *Adding and removing group participants*. Obtenido de WhatsApp:  
<https://faq.whatsapp.com/en/web/26000157/?category=5245251>

## ANEXO A: ESTRUCTURA DE LA BASE DE DATOS

La base datos de la aplicación es una base de datos no relacional, *document oriented*. Los documentos son objetos JSON. Los principales documentos de la base de datos se describen a continuación. En los ejemplos de documentos JSON se utiliza texto en negrilla para vincular el elemento con la descripción del documento. Este texto no forma parte del JSON.

### Available Cities

Este documento almacena las ciudades que son opciones de selección de origen y destino de la aplicación.

Nombre del campo	Tipo	Descripción
<b>city_name</b>	Boolean	El nombre de la ciudad

Ejemplo:

```
"AvailableCities": {
  "Ambato" (city_name): true,
  "Ibarra": true,
  "Quito": true,
  "Tena": true
}
```

### Chats

Este documento almacena la información de mensajería de la aplicación.

Nombre del campo	Tipo	Descripción
<b>chat_id</b>	Node	Identificador de cada chat. El identificador es producto de una concatenación de los Strings identificadores de cada usuario

<b>chat_id: message_id</b>	Node	Identificador de mensaje
<b>chat_id: message_id: message</b>	String	Texto del mensaje
<b>chat_id: message_id: receiver</b>	String	Emisor del mensaje
<b>chat_id: message_id: sender</b>	String	Receptor del mensaje

Ejemplo:

```

"Chats": {
  "100152531151603101223101036334" (chat_id): {
    "-LeCpa94EaNK6NMnEk7x" (message_id): {
      "message": "Hola!",
      "receiver": "100152531151603",
      "sender": "101223101036334"
    }
  }
}

```

## Notification Tokens

Este documento almacena los tokens de notificación de *Firebase Cloud Messaging*.

Nombre del campo	Tipo	Descripción
<b>token_id</b>	Node	Identificador de token
<b>token_id: token</b>	String	Token de FCM

Ejemplo:

```

"NotificationTokens": {
  "100152531151603" (token_id): {
    "token": "ciCapENeVnE..."
  }
}

```

## Trips

Este documento almacena la información de los viajes

Nombre del campo	Tipo	Descripción
<b>origin_destination_id</b>	Node	Identificador de viajes con mismo origen y destino
<b>origin_destination_id: year</b>	Node	Año del viaje
<b>origin_destination_id: year: month</b>	Node	Mes del viaje
<b>origin_destination_id: year: month: day</b>	Node	Día del viaje
<b>origin_destination_id: year: month: day:</b> <b>trip_ip</b>	Node	Identificador del viaje
<b>origin_destination_id: year: month: day:</b> <b>trip_ip: date</b>	<b>DateTime</b> <b>Node</b>	Fecha y hora del viaje
<b>origin_destination_id: year: month: day:</b> <b>trip_ip: destination</b>	<b>Location</b> <b>Node</b>	Destino del viaje
<b>origin_destination_id: year: month: day:</b> <b>trip_ip: driver</b>	<b>User</b> <b>Node</b>	Conductor del viaje
<b>origin_destination_id: year: month: day:</b> <b>trip_ip: id</b>	String	Identificador del viaje
<b>origin_destination_id: year: month: day:</b> <b>trip_ip: numberOfSeats</b>	Integer	Número de asientos del viaje
<b>origin_destination_id: year: month: day:</b> <b>trip_ip: offersD2DService</b>	Boolean	¿El viaje ofrece servicio puerta a puerta?
<b>origin_destination_id: year: month: day:</b> <b>trip_ip: origin</b>	<b>Location</b> <b>Node</b>	Origen del viaje
<b>origin_destination_id: year: month: day:</b> <b>trip_ip: originalDistance</b>	Long	Distancia original del viaje en metros
<b>origin_destination_id: year: month: day:</b> <b>trip_ip: originalDuration</b>	Long	Duración original del viaje en segundos
<b>origin_destination_id: year: month: day:</b> <b>trip_ip: passengers</b>	Node	Arreglo de pasajeros que forman parte del viaje

<b>origin_destination_id: year: month: day: trip_ip: passengers: passenger_id</b>	Node	Identificador de usuario del pasajero
<b>origin_destination_id: year: month: day: trip_ip: passengers: passenger_id: d2dService</b>	Node	Información de viaje de puerta a puerta del pasajero
<b>origin_destination_id: year: month: day: trip_ip: passengers: passenger_id: d2dService: destination</b>	<b>Location Node</b>	Punto de destino del viaje puerta a puerta del pasajero. El nodo no contiene el campo <i>city</i>
<b>origin_destination_id: year: month: day: trip_ip: passengers: passenger_id: d2dService: distance</b>	Long	Distancia del viaje de puerta a puerta en metros
<b>origin_destination_id: year: month: day: trip_ip: passengers: passenger_id: d2dService: duration</b>	Long	Duración del viaje de puerta a puerta en segundos
<b>origin_destination_id: year: month: day: trip_ip: passengers: passenger_id: d2dService: fee</b>	Double	Tarifa del viaje de puerta a puerta en dólares
<b>origin_destination_id: year: month: day: trip_ip: passengers: passenger_id: d2dService: origin</b>	<b>Location Node</b>	Punto de recogida del viaje puerta a puerta del pasajero. El nodo no contiene el campo <i>city</i>
<b>origin_destination_id: year: month: day: trip_ip: passengers: passenger_id: firstName</b>	String	Primer nombre del conductor
<b>origin_destination_id: year: month: day: trip_ip: passengers: passenger_id: friendsInCommonWithTheDriver</b>	Integer	Numero de amigos en común del pasajero del servicio de puerta a puerta con el conductor
<b>origin_destination_id: year: month: day: trip_ip: passengers: passenger_id: id</b>	String	Identificador de usuario del pasajero
<b>origin_destination_id: year: month: day: trip_ip: passengers: passenger_id: profilePictureUrl</b>	String	Link de foto de perfil del pasajero
<b>origin_destination_id: year: month: day: trip_ip: price</b>	Double	Tarifa del viaje en dólares

Ejemplo:

```

"Trips": {
  "Ibarra-Quito" (origin_destination_id): {
    "2019" (year): {
      "05" (month): {
        "06" (day): {
          "201905061500-LeCzSDZ_D..." (trip_id): {
            "date": {date node item},
            "destination": {location node item},
            "driver": {user node item},
            "id": "201905061500-LeCzSDZ_D... ",
            "numberOfSeats": 2,
            "offersD2DService": true,
            "origin": {location node item},
            "originalDistance": 120756,
            "originalDuration": 8517,
            "passengers": {
              "100152531151603": {
                "d2DService": {
                  "destination": {},
                  "distance": 123504,
                  "duration": 9699,
                  "fee": 2.54,
                  "origin": {location node item}
                },
                "firstName": "Esteban",
                "friendsInCommonWithDriver": 2,
                "id": "1001525...",
                "profilePictureUrl": "https://..."
              }
            },
            "price": 5 }}}}

```

## UserTrips

Este documento almacena la información de viajes de un usuario. Esta información es la misma información que almacenan un nodo de viaje del documento “Trips”. Este documento existe para optimizar las consultas de “Mis viajes”.

Nombre del campo	Tipo	Descripción
<b>user_id</b>	Node	Identificador de usuario
<b>user_id: trip_id</b>	<b>Trip Nodes Array</b>	Arreglo de nodos de <i>Trip</i>

Ejemplo:

```

"Users": {
  "1081116..." (user_id): {
    Trip node item 1,
    ...
    Trip node item n
  }
}

```

## Users

Documento que contiene información sobre los usuarios de la aplicación.

Nombre	Tipo	Descripción
<b>Users</b>	Arreglo de nodos	Arreglo de nodos <b>User</b>

Ejemplo:

```

"Users": {
  User node item 1,
  ...
  User node item n
}

```

## Subnodos

Esta sección describe subnodos de la base de datos.

**DateTime.** Representa información de fecha y hora.

Nombre	Tipo	Descripción
<b>day</b>	Integer	Día
<b>hour</b>	Integer	Hora
<b>minute</b>	Integer	Minuto
<b>month</b>	Integer	Mes
<b>year</b>	Integer	Año

Ejemplo:

```

"date": {
  "day": 6,
  "hour": 12,
  "minute": 0,
  "month": 5,
  "year": 2019
}

```

**Location.** Representa una ubicación.

Nombre	Tipo	Descripción
<b>address</b>	String	Dirección
<b>city</b>	String	Ciudad
<b>lat</b>	Double	Latitud
<b>lng</b>	Double	Longitud

Ejemplo:

```
"location": {
  "address": "Calle A...",
  "city": "Quito",
  "lat": -0.2...,
  "lng": -78.4...,
}
```

**User.** Representa un usuario.

Nombre	Tipo	Descripción
<b>email</b>	String	Correo electrónico
<b>firstName</b>	String	Primer nombre
<b>friends</b>	Node	Arreglo de los identificadores de los amigos del usuario
<b>friends: friend_id</b>	Boolean	Identificador de usuario del amigo del usuario
<b>fullName</b>	String	Nombre completo
<b>id</b>	String	Identificador de usuario
<b>profilePictureUrl</b>	String	Link de foto de perfil

Ejemplo:

```
"user": {
  "email": "user@mail.com",
  "firstName": "Esteban",
  "friends": {
    "100152531151603": true,
    "103426700821335": true,
    "105168937312200": true
  },
  "fullName": "Esteban Mauricio...",
  "id": "108111643681397",
  "profilePictureUrl": https://...
}
```

## ANEXO B: INTERFAZ GRÁFICA DE LA APLICACIÓN

### Actividad de inicio (Home Activity)



### Actividad de sesión con y sin mensaje de restricción (Sign In Activity)



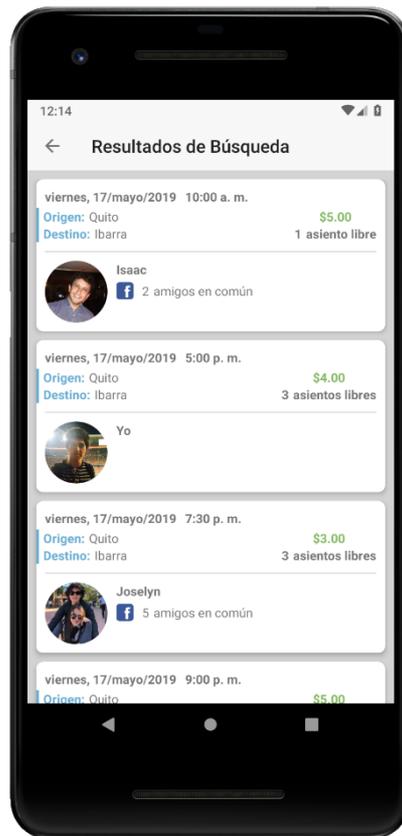
## Actividad de publicación de viaje (Post Activity)



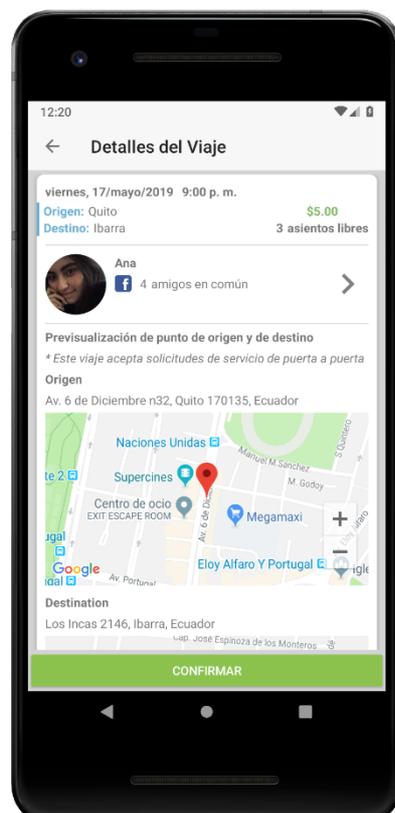
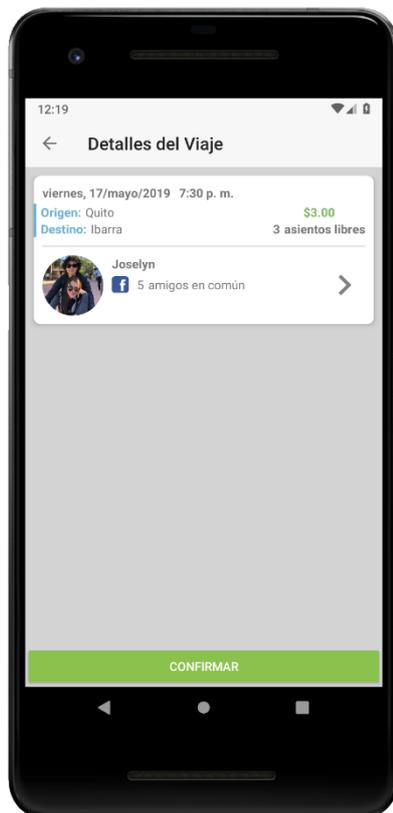
## Actividad de búsqueda de viaje (Search Activity)



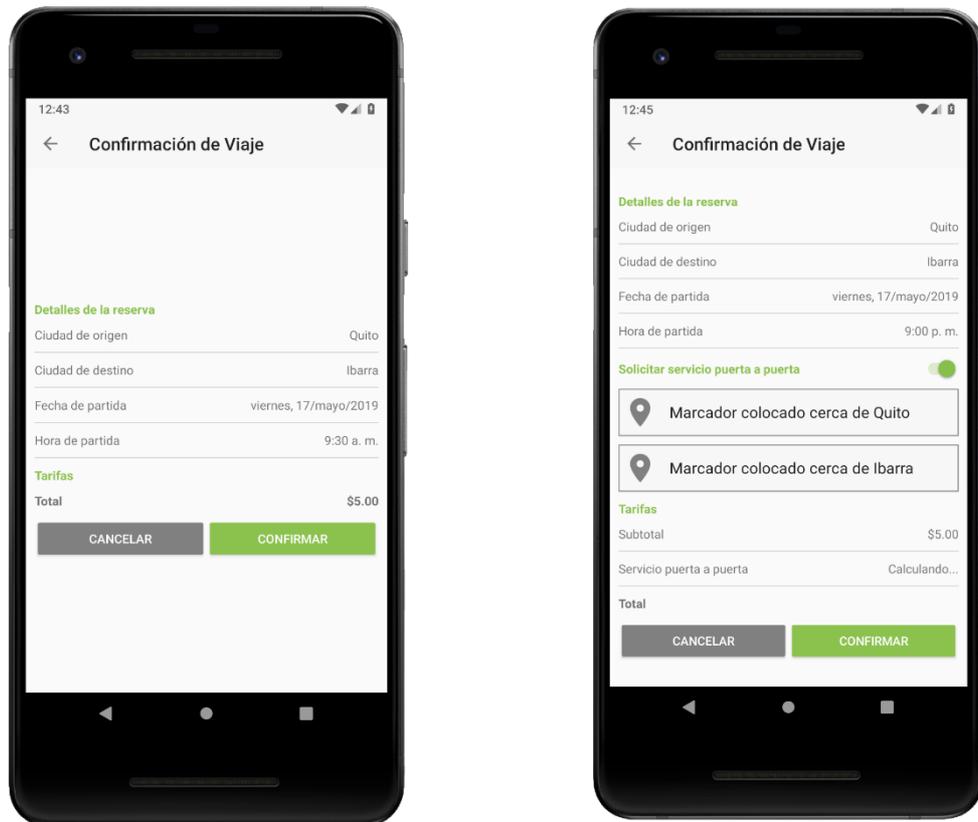
## Actividad de resultados de búsqueda (Search Results Activity)



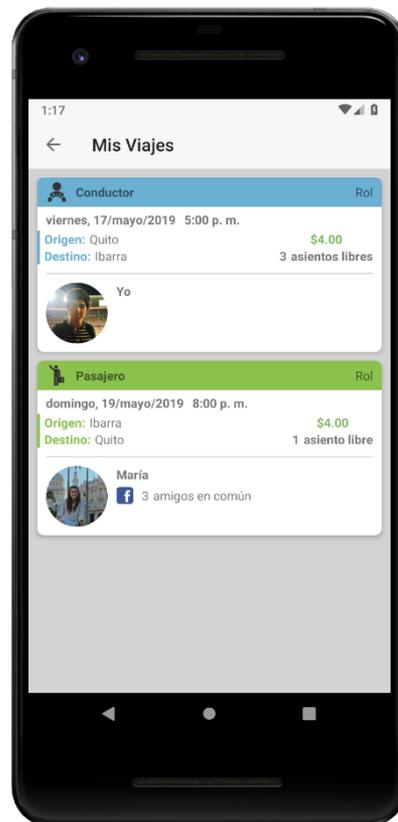
## Actividad de detalles del viaje (Trip Details Activity)



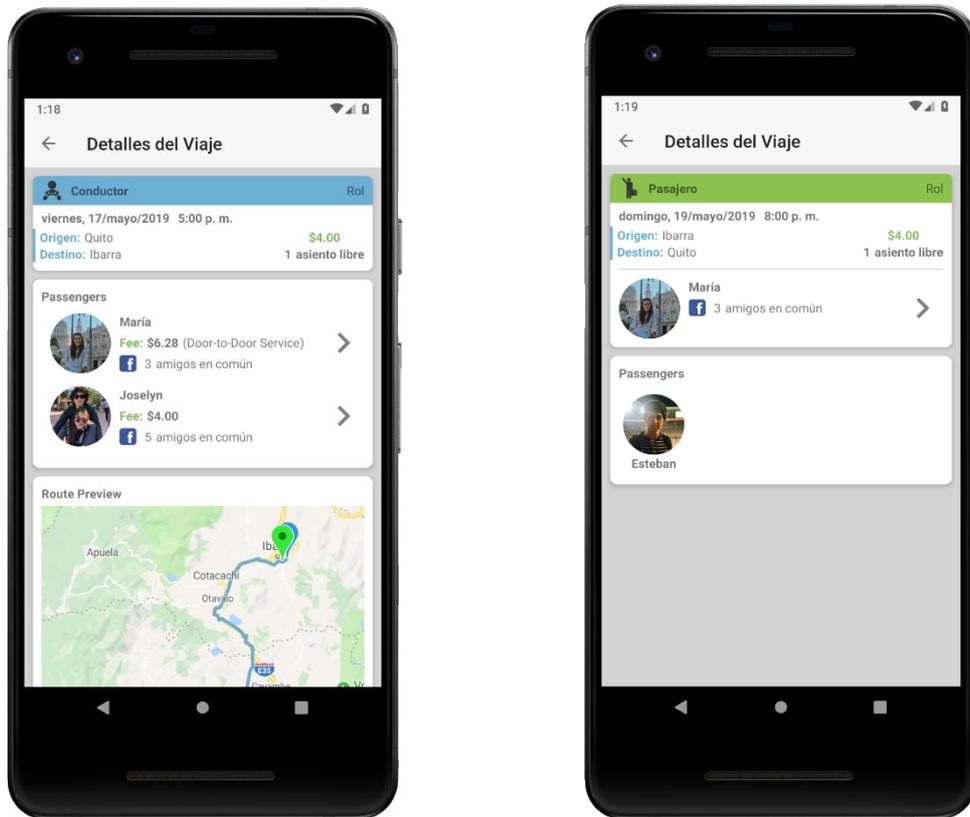
## Actividad de confirmación de reserva (Booking Confirmation Activity)



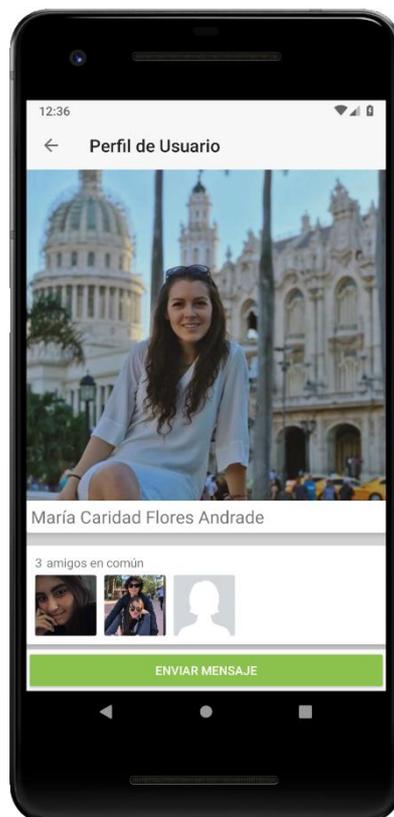
## Actividad de mis viajes (User Profile Activity)



### Detalles de mi viaje como conductor o como pasajero.



### Actividad de perfil de usuario (User Profile Activity)



## Actividad de mensajería (Message Activity)



## Selección de ubicación en mapa (Select Map Location Activity)

