

**UNIVERSIDAD POLITÉCNICA SALESIANA**

**SEDE QUITO**

**CARRERA:**

**INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:**

**Ingenieros de Sistemas**

**TEMA:**

**DESARROLLO DE UN PROTOTIPO DE APLICACIÓN MÓVIL PARA  
MONITOREAR EL TRANSPORTE PÚBLICO EN QUITO.**

**AUTORES:**

**CHRISTIAN ESTEBAN ARGUELLO PAEZ**

**DANNY ADRIAN QUISHPE FARINANGO**

**TUTOR**

**HOLGER RAÚL ORTEGA MARTÍNEZ**

**Quito, marzo del 2021**

## CESIÓN DE DERECHOS DE AUTOR

Nosotros, Christian Esteban Arguello Paez, con documento de identificación N° 1726207671 y Danny Adrian Quishpe Farinango, con documento de identificación N° 1725079568, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: DESARROLLO DE UN PROTOTIPO DE APLICACIÓN MÓVIL PARA MONITOREAR EL TRANSPORTE PÚBLICO EN QUITO., mismo que ha sido desarrollado para optar por el título de INGENIEROS DE SISTEMAS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada.

En concordancia, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



ARGUELLO PAEZ  
CHRISTIAN ESTEBAN  
CI:1726207671



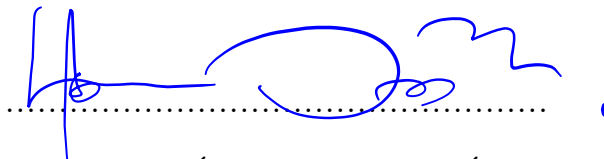
QUISHPE FARINANGO  
DANNY ADRIAN  
CI: 172507956

Quito, marzo 2021

## **DECLARATORIA DE COAUTORÍA DEL TUTOR**

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: DESARROLLO DE UN PROTOTIPO DE APLICACIÓN MÓVIL PARA MONITOREAR EL TRANSPORTE PÚBLICO EN QUITO., realizado por Christian Esteban Arguello Paez y Danny Adrian Quishpe Farinango, obteniendo un producto que cumple con todos los requisitos por la Universidad Politécnica Salesiana para ser considerado como trabajo final de titulación.

Quito, marzo 2021



**HOLGER RAÚL ORTEGA MARTÍNEZ**

CI: 1708182728

## **DEDICATORIA**

Dedico este trabajo de titulación en primer lugar a mis padres que con su esfuerzo y dedicación me enseñaron a que con trabajo duro todas las metas que me proponga se pueden lograr.

A mis hermanos que supieron darme su apoyo y confianza, por alegrar mis días más oscuros y hacerme reír con cada una de sus locuras.

A mis Docentes por brindarme toda su sabiduría y técnicas para mi desarrollo profesional.

Finalmente, a todos mis amigos y compañeros a los que conocí en esta etapa de mi vida, ya que con ellos tuve muchas experiencias agradables.

*Christian Esteban Arguello Paez*

## **DEDICATORIA**

Dedico este trabajo en primer lugar a Dios por haber guiado en la etapa de estudio, sobre todo en los más difíciles y darme fuerza para poder seguir adelante en todo el proceso universitario.

A mis docentes que me inculcaron el conocimiento y me supieron ayudar en las dudas que se presentaban.

A mi familia, por todo el apoyo incondicional, por darme fuerzas para seguir y alentarme en culminar mi carrera.

A mis hermanos, por acompañarme y apoyarme en todo lo que se requiera por darme los recursos cuando más lo necesitaba.

A mis compañeros y amigos que fueron quienes me supieron apoyar en las aulas y en toda la vida universitaria.

*Danny Adrian Quishpe Farinango*

## **AGRADECIMIENTOS**

Agradecemos a la Universidad Politécnica Salesiana por darnos la oportunidad de tener una carrera, por contribuir en nuestra formación profesional y personal a nuestro tutor el Físico Holger Ortega por habernos motivado y guiado en todo el proceso para poder realizar nuestro trabajo de titulación.

A nuestros amigos quienes nos ayudaron y apoyaron en esta etapa de crecimiento profesional.

**Christian Esteban Arguello Paez**  
**Danny Adrian Quishpe Farinango**

## ÍNDICE DE CONTENIDO

<b>INTRODUCCIÓN</b> .....	1
Antecedentes.....	1
Justificación.....	3
<b>OBJETIVOS</b> .....	5
Objetivo General .....	5
Objetivo Específico .....	5
<b>CAPÍTULO I</b> .....	6
<b>MARCO CONCEPTUAL</b> .....	6
1.1 Metodologías Ágiles.....	6
1.2 Scrum.....	6
1.2.1 Definición .....	6
1.2.2 Actividades de Scrum .....	7
1.3 Patrón Modelo-Vista-Controlador (MVC).....	7
1.4 Modelo Cliente-Servidor.....	8
1.5 BPMN (Business Process Model and Notation) .....	8
1.6 Aplicación móvil .....	8
1.7 Android.....	8
1.8 Base de Datos .....	9
1.9 Herramientas de Desarrollo.....	9
1.9.1 Firebase.....	9
1.9.2 Facebook Developers.....	9
1.9.3 Google Maps.....	10
1.9.4 Api Keys .....	10
1.9.5 Android Studio.....	10
1.9.6 Bizagi Modeler .....	10
1.9.7 UML.....	11
1.9.8 Java .....	11
1.9.9 XML.....	11
1.9.10 Servicios REST.....	11
1.9.11 JWT Token .....	11
<b>CAPÍTULO II</b> .....	13
<b>METODOLOGÍA</b> .....	13
2.1 Análisis del sistema.....	13
2.1.1 Diagrama de bloques .....	13

2.1.2 Alcance del proyecto .....	14
2.1.3 Análisis de requerimientos.....	15
2.1.4 Análisis de factibilidad .....	16
2.2 Implementación de Metodología.....	19
2.2.1 Roles .....	19
2.2.2 Cronograma de Actividades.....	20
2.2.3 Sprints .....	21
2.2.4 Mapa de la Aplicación .....	24
<b>CAPÍTULO III.....</b>	<b>29</b>
<b>DESARROLLO E IMPLEMENTACIÓN.....</b>	<b>29</b>
3.1 Aplicación Móvil.....	29
3.2 Diseño de la aplicación.....	29
3.3 Arquetipo.....	29
3.4 Configuraciones iniciales .....	33
3.5 Inicio de sesión.....	35
3.6 Creación de las clases.....	36
3.7 Desarrollo de Módulos .....	37
<b>CAPÍTULO IV .....</b>	<b>43</b>
<b>PRUEBAS Y RESULTADOS .....</b>	<b>43</b>
4.1 Pruebas Unitarias.....	43
4.1.1 Pruebas Login .....	43
4.1.2 Pruebas Base de Datos.....	44
4.1.3 Pruebas Buster .....	46
4.1.4 Pruebas Ruters .....	48
4.2 Pruebas de Sistema.....	49
4.3 Pruebas de Aceptación .....	53
4.3.1 ¿Qué tan factible sería usar la aplicación BusterApp en el Distrito Metropolitano de Quito?.....	53
4.3.2 ¿Cree que es necesario que se realice un monitoreo de las unidades de autobuses en la ciudad de Quito?.....	54
4.3.3 ¿Según su experiencia que tan fácil fue usar la aplicación? .....	55
4.3.4 ¿Qué colores le gustaría en la aplicación? .....	55
4.3.5 ¿Qué icono le gusta más?.....	56
4.3.6 ¿Cree que los íconos utilizados en esta aplicación son los adecuados?.....	58
4.3.7 ¿El lenguaje utilizado en la aplicación es el idóneo o entendible en cuanto a los mensajes de ayuda al momento de calificar? .....	58



4.3.8 ¿Qué elementos cambiaría en el diseño de la aplicación? .....	59
4.3.9 ¿Usted estaría dispuesto a compartir su información personal al momento de registrarse dentro de la aplicación? .....	60
4.3.10 Resultado del diseño basados en los resultados de la encuesta .....	60
<b>CONCLUSIONES</b> .....	68
<b>RECOMENDACIONES</b> .....	69
<b>GLOSARIO</b> .....	70
<b>REFERENCIAS</b> .....	71

## ÍNDICE DE TABLAS

Tabla 1 Requerimientos mínimos .....	17
Tabla 2 Recursos Operacionales .....	18
Tabla 3 Patrocinador .....	19
Tabla 4 Scrum Master .....	19
Tabla 5 Equipo de desarrollo .....	20
Tabla 6 Cronograma de Actividades.....	20
Tabla 7 Pruebas de Login.....	43
Tabla 8 Pruebas módulo Buster .....	46
Tabla 9 Pruebas en el módulo de Rutgers.....	48
Tabla 10 Funcionamiento de sistema.....	50

## ÍNDICE DE FIGURAS

Figura 1 Se puede apreciar las fases de evolución de la aplicación.....	13
Figura 3 Diseño preliminar del prototipo.....	21
Figura 4 Medios que se integrarán a la aplicación.....	22
Figura 5 Inicio de sesión por medio de redes sociales y correo electrónico.....	22
Figura 6 Usabilidad del módulo de login.....	23
Figura 7 Requerimientos básicos del módulo Buster.....	23
Figura 8 Capacitación y requerimientos del módulo Ruters.....	24
Figura 9 Fase de pruebas y mejoras integrando todos los módulos.....	24
Figura 10 Mapa jerárquico de las clases del prototipo.....	25
Figura 11 Procesos del módulo login.....	26
Figura 12 Procesos del módulo Buster.....	27
Figura 13 Procesos del módulo Ruters.....	28
Figura 14 Funcionamiento de la arquitectura del sistema.....	28
Figura 15 Boceto de la pantalla inicial.....	30
Figura 16 Boceto de la pantalla de principal.....	30
Figura 17 Boceto del módulo Buster con sus categorías.....	31
Figura 18 Boceto módulo ruters.....	32
Figura 19 Boceto actualizado del módulo Ruters.....	33
Figura 20 Nuevo proyecto dentro de Firebase.....	34
Figura 21 Comunicación inicial entre Firebase y Android Studio.....	35
Figura 22 Servicios de autenticación para el módulo login.....	35
Figura 23 Configuración inicial para la autenticación con Facebook.....	36
Figura 24 Clases del prototipo usando MVC.....	37
Figura 25 Vista previa de la interfaz del módulo.....	38
Figura 26 Simulación de la interfaz del login.....	39
Figura 27 Ventana de bienvenida para los usuarios.....	40
Figura 28 Primera versión del módulo Ruters en funcionamiento.....	41
Figura 29 Implementación del sistema de calificaciones por medio de estrellas.....	42
Figura 30 Almacenamiento de usuarios en la BD de Firebase.....	44
Figura 31 Servicio de autenticación de usuarios registrados.....	45
Figura 32 Almacenamiento de calificación enviadas desde el módulo Buster.....	47
Figura 33 Almacenamiento de las coordenadas de los usuarios Ruters.....	49
Figura 34 Muestra de usuarios que usan la aplicación luego de la instalación del aplicativo.....	51
Figura 35 Usuarios activos en la aplicación.....	52
Figura 36. Uso de cada pantalla del prototipo por parte de los usuarios.....	53
Figura 37 Porcentajes de la factibilidad de la aplicación.....	54
Figura 38 porcentajes sobre el monitoreo en la ciudad de Quito.....	54
Figura 39 Manejo y entendimiento del prototipo por parte de los usuarios-.....	55
Figura 40 Selección de colores del prototipo por parte de los usuarios.....	56
Figura 41 Íconos llamativo por los usuarios.....	57
Figura 42 Porcentajes del ícono mas llamativo por parte de los encuestados.....	57
Figura 43 Selección de íconos para las categorías del módulo Buster.....	58
Figura 44 Porcentajes sobre el entendimiento del lenguaje implementado.....	59
Figura 45 Resultados sobre cambiar el diseño de los módulos según el usuario.....	59

Figura 46 Resultados sobre compartir la información de usuarios.....	60
Figura 47 Mejora de la pantalla de inicio.....	61
Figura 48 Resultado final del módulo Login .....	62
Figura 49 Resultado de la pantalla de registro de usuarios por medio de correo electrónico.....	63
Figura 50 Prototipo final, pantalla principal con sus módulos implementados .....	64
Figura 51 Resultado final del módulo Buster con las sugerencias de los usuarios encuestados. ....	65
Figura 52 Mensajes de ayuda para los usuarios.....	66
Figura 53 Resultado final del módulo Rutgers .....	67

## **RESUMEN**

El objetivo principal del presente proyecto es la construcción de un prototipo de aplicación móvil para el monitoreo del transporte público en la ciudad de Quito, utilizando la metodología de desarrollo ágil SCRUM.

Dicho prototipo se plantea como una herramienta para poder calificar y visualizar a las unidades de transporte público; además de contar con la opción de opinar sobre las mismas. El sistema operativo sobre el cual se desarrolla el prototipo de aplicación es Android, el cual permite un inicio de sesión vinculado a redes sociales o tradicionalmente con el registro por medio de una dirección de correo electrónico.

En la primera parte del proyecto se detalla toda la información acerca de las diferentes herramientas que se utilizaron para el desarrollo conceptual, los requerimientos, el diseño y la arquitectura del prototipo planteado. Dentro de la segunda parte se explica la construcción, implementación, y las pruebas de funcionamiento, tanto unitarias como conjuntas. De trece usuarios encuestados, el 69.2% consideró viable al proyecto en la ciudad de Quito.

## **ABSTRACT**

The main objective of this project is the development of a prototype of a mobile application for monitoring public transport in Quito city, using the SCRUM agile development methodology.

The prototype is proposed as a tool to grade and visualize public transport units; in addition to having the option to comment on them. The operating system upon which this application prototype was developed is Android, which allows a session sign up linked to social network or traditionally with the use of an e-mail address.

In the first part of the project, all the information about the different tools that were used for conceptual development, the requirements, the design, and the architecture of the proposed prototype is detailed. Within the second part the explanation about the construction, implementation, and functional tests, both unit and joint, is given. Of thirteen users surveyed, this prototype was considered as functional in Quito city by 69.2% of them.

## INTRODUCCIÓN

### Antecedentes

El sistema de transporte público de la ciudad de Quito, especialmente aquel que no pertenece a los sistemas integrados a cargo del municipio (Trolebús, Ecovía, Metrovía), adolece de un insuficiente sistema de gestión de la información, lo cual ha dado como resultado una mala calidad del servicio (Núñez Manzano & Pino Riera, 2019).

Por ejemplo, no existe información sobre tiempos de espera en paradas, o sobre la calidad de servicio que brinda determinada unidad. De igual manera, las autoridades reguladoras (e incluso los mismos propietarios de las unidades y los directivos de las cooperativas de transporte) no cuentan con un medio que les permita conocer el desempeño de los conductores, el cometimiento de infracciones, el estado de las unidades o la voz de los usuarios insatisfechos con el sistema (Quito Informa, 2017).

Obviamente, se han realizado esfuerzos por solucionar esta problemática mediante la implementación de herramientas tecnológicas; desde mensajes para denunciar el acoso hasta aplicaciones móviles como MOVILIZATE UIO; promocionada por el municipio de Quito como una herramienta que cuenta con tecnología de Google, que permitirá al usuario de Transporte Público conocer la información del servicio como rutas y horarios del transporte de autobús (Quito Informa, 2017).

Sin embargo, estas soluciones no han sido efectivas para el problema que describimos. Una rápida inspección del aplicativo MOVILIZATE UIO, por ejemplo, devela algunos vacíos como, por ejemplo: en la sección “mapas” se presenta solo imágenes PDF de los mapas del sistema integrado de transporte (ignorando por completo al resto de rutas y paradas), y el sistema de reportes de incidentes e

irregularidades requiere que el usuario ingrese una cantidad de información tal, que vuelve al sistema poco práctico. Si hay alguna información que se termine gestionando efectivamente, no lo sabemos, porque dicha información está oculta para el usuario común.

Muchas de estas problemáticas han sido solucionadas, en su mayoría, por aplicaciones móviles; sin necesidad de aprobación de leyes con alto costo político ni de instalación de hardware (sensores, cámaras, etc.) que resultaría a la postre muy costoso, aplicaciones como Waze han logrado sistemas que prestan información muy relevante al usuario, “usando” al resto de usuarios como agentes generadores de esa información (Quito Informa, 2017). Otra aplicación que nos sirve de inspiración en el presente proyecto es Uber, que “utiliza” a los usuarios para brindar a un nuevo pasajero la información que necesita para tomar una unidad; ejerciendo, con ello, una presión social capaz de incrementar en mucho la calidad del servicio de transporte en este ámbito.



## **Justificación**

Siguiendo el modelo de las aplicaciones mencionadas (y otras, probablemente) la aplicación que se plantea en el presente proyecto permitirá “usar” a los usuarios del transporte público de Quito para generar información que será relevante a los próximos usuarios y a las autoridades y público en general. Esta información incluirá: promedios y número de calificaciones para cada unidad, reportes de incidentes registrados por cada unidad, rutas efectuadas, velocidad y estimaciones acerca de los tiempos de llegada de unidades a una determinada parada, dado un código o ID de dicha parada. Esta información, se intuye, será una herramienta importante en la mejora del servicio de transporte urbano de Quito.

Algunos de los problemas a los que se enfrenta al momento de implementar la aplicación en la práctica son la (potencial) baja capacidad de los dispositivos móviles de los usuarios, en promedio; el porcentaje de usuarios que no cuenta con datos móviles; y la poca utilidad del sistema en un inicio, al carecer de la información que “alimente” su base de datos.

Estos problemas pueden solucionarse haciendo un aplicativo ligero, que no consuma mucha batería y que no consuma una gran cantidad de datos, con un modo offline en donde los reportes se podrán enviar cuando los usuarios estén conectados a una red WI-FI. Por otra parte, se contempla la posibilidad de extender el proyecto a un emprendimiento comercial (apoyados por el sistema de Coworking de la universidad), cuyos fondos iniciales permitirían hacer la publicidad necesaria para que más y más usuarios descarguen la aplicación y vayan así alimentando la base de datos, volviendo más cada vez más útil y poderoso al sistema.

Para la implementación del aplicativo tenemos el conocimiento de manejo de sistemas de información de georreferenciación (*GIS*), programación de aplicaciones

móviles (Android Studio) y manejo de base de datos, que nos permitirá su desarrollo de forma óptima y ligera, apoyado con un web service donde se almacene toda la información proporcionada por los usuarios.

## **OBJETIVOS**

### **Objetivo General**

Diseñar e implementar un prototipo de aplicación móvil para coleccionar, gestionar y presentar información sobre las unidades del transporte público de Quito.

### **Objetivo Específico**

Diseñar la arquitectura del sistema, definiendo los módulos que conforma el mismo.

Implementar la aplicación móvil según la arquitectura diseñada.

Realizar las pruebas de funcionamiento, utilizando un público de control constituido por usuarios voluntarios.

# **CAPÍTULO I**

## **MARCO CONCEPTUAL**

### **1.1 Metodologías Ágiles**

Las metodologías ágiles hacen referencia a las técnicas de desarrollo de software con el fin de que el proyecto se ajuste al triángulo de hierro en gestión de proyectos de software (Cadavid 2013).

### **1.2 Scrum**

#### **1.2.1 Definición**

La metodología Scrum es una metodología ágil que provee de pasos para la gestión e implementación de desarrollo de software. Scrum ofrece una manera personalizada de trabajar un proyecto con varios requerimientos, lo cual le hace ventajosa debido a la flexibilidad de organizar las tareas por su dificultad. A dichas tareas se les denomina “Sprint”. Los Sprint en el proyecto permiten tener un inicio y un fin estable con los cuales se puede ir avanzando en el proceso (Srivastava, Bhardwaj, y Saraswat 2017).

Scrum es un conjunto de buenas prácticas y roles de las personas que conformarán el proyecto. Los roles son:

Scrum Master: Figura que lidera los equipos en la gestión ágil del proyecto.

Product Owner (Dueño del Producto): Es la figura que prioriza las características del producto.

Equipo de Desarrollo: Por lo general está conformado dos a nueve personas se encargan de desarrollar e implementar el software.

Scrum ofrece ciertas características que son:

- Alinear la gestión de requisitos del Cliente
- Flexibilidad.

- Adaptabilidad.
- Mitigación de riesgos.
- Productividad y calidad.
- Alineamiento entre el cliente y el equipo de desarrollo.

### **1.2.2 Actividades de Scrum**

**Sprint Planning:** Es la actividad en donde todo el equipo decide qué actividades tendrá el proyecto a desarrollar; así, teniendo una proyección del resultado final, este puede ir experimentando modificaciones en el transcurso del desarrollo (Diaz Ortiz y Romero Suarez 2017).

**Sprint:** Es cada uno de los bloques que se va a realizar. A estos se los denomina como entregable que en su total viene a ser el proyecto final (Srivastava et al. 2017).

**Product Backlog:** Es una lista de objetivos o requisitos en los que representan la visión y expectativas del cliente. Este se encarga de la gestión de tiempo y de entregables para el cliente (Diaz Ortiz y Romero Suarez 2017; Srivastava et al. 2017).

**Reuniones:** Son programadas para que el Scrum Master presente avances del proyecto al cliente, si existen modificaciones pequeñas por parte del cliente se las realiza en ese momento y si las modificaciones tienen un grado de dificultad mayor se lo deja hasta la próxima reunión (Diaz Ortiz y Romero Suarez 2017; Srivastava et al. 2017).

### **1.3 Patrón Modelo-Vista-Controlador (MVC)**

Es un conjunto de buenas prácticas que consiste en dividir el desarrollo de software para tener claridad al identificar las partes del software y que su funcionalidad está bien determinada (Bascón Pantoja 2004).

**Modelo:** El modelo es responsable de acceder a la capa de almacenamiento, comunicar las reglas del negocio y notificar los cambios del usuario (Anón 2012).

Vista: La vista es donde el usuario manda las peticiones y la aplicación presentara los resultados visuales. Un ejemplo puede ser el inicio de sesión de un usuario (López S. 2009).

Controlador: Son todas las clases que controlarán las interacciones de manera que solicitan la información al modelo y los presentan por medio de la vista haciendo que el flujo de la información sea legible (Bahit 2011).

#### **1.4 Modelo Cliente-Servidor**

Es un modelo de aplicación distribuida en donde se tiene todos los servicios que se comparten entre los distintos recursos denominados servidores. A los solicitantes se les denomina clientes. Estos clientes mandan varias peticiones con el fin de completar las tareas de la aplicación (Marini 2012).

Este modelo nos permite trabajar sin que los clientes se sobrecarguen de peticiones, ya que ellos solo solicitan al servidor y se comunican entre sí para poder dar el resultado requerido de manera rápida y concisa.

#### **1.5 BPMN (Business Process Model and Notation)**

Es una notación legible y entendible, cuyo objetivo es dar a conocer a todos los usuarios del negocio el flujo del proceso (Vasquez, Giandini, & Bazán, 2010).

#### **1.6 Aplicación móvil**

Es un programa de software que puede llegar a realizar una o varias funciones determinadas que el usuario necesite. Generalmente este tipo de programas son pequeños y específicos, particularmente usados en dispositivos móviles (Milenium, 2020).

#### **1.7 Android**

Android es un S.O. que proporciona interfaces para desarrollar aplicaciones móviles que accedan a las funciones del celular (como el GPS, las llamadas, la agenda,

etc.) de una forma muy sencilla, bajo el lenguaje de programación Java (Nieto Gonzales, 2020).

## **1.8 Base de Datos**

Es un almacén de datos con información que puede o no estar relacionada, comúnmente este agrupamiento de información quiere representar aspectos del mundo real, para tenerlos accesibles cuando el usuario necesite hacer uso de la información (Gutiérrez Díaz, 2020). Cada registro dentro de las bases de datos se las conoce como tablas y estas estas contienen “Datos” como puede ser ID, números telefónicos, direcciones, nombres, edades, años, etc.

## **1.9 Herramientas de Desarrollo**

### **1.9.1 Firebase**

Es una plataforma web que ayuda a desarrolladores a construir aplicaciones de alta calidad y sigue un modelo de desarrollo Backend-as-a-Service. Administrado por el formato JavaScript Object Notation (JSON) que ayuda a la manipulación de datos en tiempo real (Khawas y Shah 2018).

Firebase cloud system es la combinación de varios servicios de Google en la nube en los cuales incluye Análisis de datos, Autenticación de cuentas, Base de datos en tiempo real, Repositorio de almacenamiento, Hosting web y un sistema de mensajería (Khawas y Shah 2018; Li et al. 2018).

### **1.9.2 Facebook Developers**

Es una plataforma de desarrollo que provee las herramientas para el inicio de sesión, las cuales permiten además analizar el índice de usuarios que ingresan, se registran y están conectados en un periodo de tiempo, gracias a sus gráficas (Graph API) (Facebook, 2020).

### **1.9.3 Google Maps**

Es un servicio que presta a los usuarios información dinámica para encontrar su ubicación en tiempo real, haciendo que los usuarios puedan organizar o planificar una trayectoria de viaje, esto por medio de imágenes de geolocalización ya que cuenta con dos componentes que son Maps SDK for Android y Directions API (Google, 2020).

### **1.9.4 Api Keys**

Es un servicio gratuito que nos proporciona Google Console. Proporciona acceso a servicios como Google Maps, Google Cloud e inicio de sesión de manera automática. La API Key sirve como identificador para garantizar la seguridad del usuario.

También sirve para agregar marcadores al mapa en donde se puede obtener información del objeto seleccionado. Estos objetos proporcionan información adicional de las ubicaciones en el mapa y permiten la interacción del usuario con el mapa (Google, 2020).

### **1.9.5 Android Studio**

Según la guía del usuario de la página oficial de desarrolladores de Android la definen como entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android basado en IntelliJ IDEA (Android Studio, 2020).

### **1.9.6 Bizagi Modeler**

Es el conjunto de servicios en donde el equipo de desarrollo de software puede colaborar de manera visual, al realizar el modelado o simulaciones de un proceso del negocio mediante BPMN (Bizagi Ltd, 2020).



### **1.9.7 UML**

UML es considerado como los planos de desarrollo de una aplicación para los arquitectos de software para proporcionar una estructura, donde se puede ver cómo la información viaja a través de la aplicación, esto se debe a que su objetivo es proporcionar herramientas de análisis, diseño e implementación para el ciclo de vida de desarrollo de software (Group 2017).

### **1.9.8 Java**

Es un Lenguaje de programación creado por Sun Microsystems, cuyo propósito general es crear aplicativos basados en programación hacia objetos. Java es rápido, seguro y es fiable (Oracle, 2021).

### **1.9.9 XML**

XML se define como un lenguaje de marcado extensible que fue diseñado para almacenar y transportar la información entre aplicación y bases de datos. Este tipo de documentos es útil para comunicarse con las BD por su estándar de segmentación de información, el cual la hace eficaz para que no lleguen datos o información incompleta (Álvarez Pérez, 2017).

### **1.9.10 Servicios REST**

Los servicios REST se puede definir como el uso de servicios web bajo protocolos HTTP, este tipo de servicios se activan al escuchar operaciones CRUD "Create Read Update Delete" (Amodeo, 2013). Y generalmente este tipo de servicios funcionan en formato JSON y XML(Mascheroni, Maximiliano Irrazábal 2016).

### **1.9.11 JWT Token**

El JWT es una cadena codificada en base64 que permite la autenticación y el intercambio de información, por medio de sus tres partes: el encabezado, que determina el algoritmo de codificación; la carga útil, que son los datos de

configuración para el token de acceso o de identidad; y, la firma, que permite saber la validez del token (Gunawan y Rahmatulloh 2019).

## CAPÍTULO II

### METODOLOGÍA

#### 2.1 Análisis del sistema

En el capítulo dos se tratará sobre la evolución, alcance, requerimientos y la factibilidad del proyecto.

##### 2.1.1 Diagrama de bloques

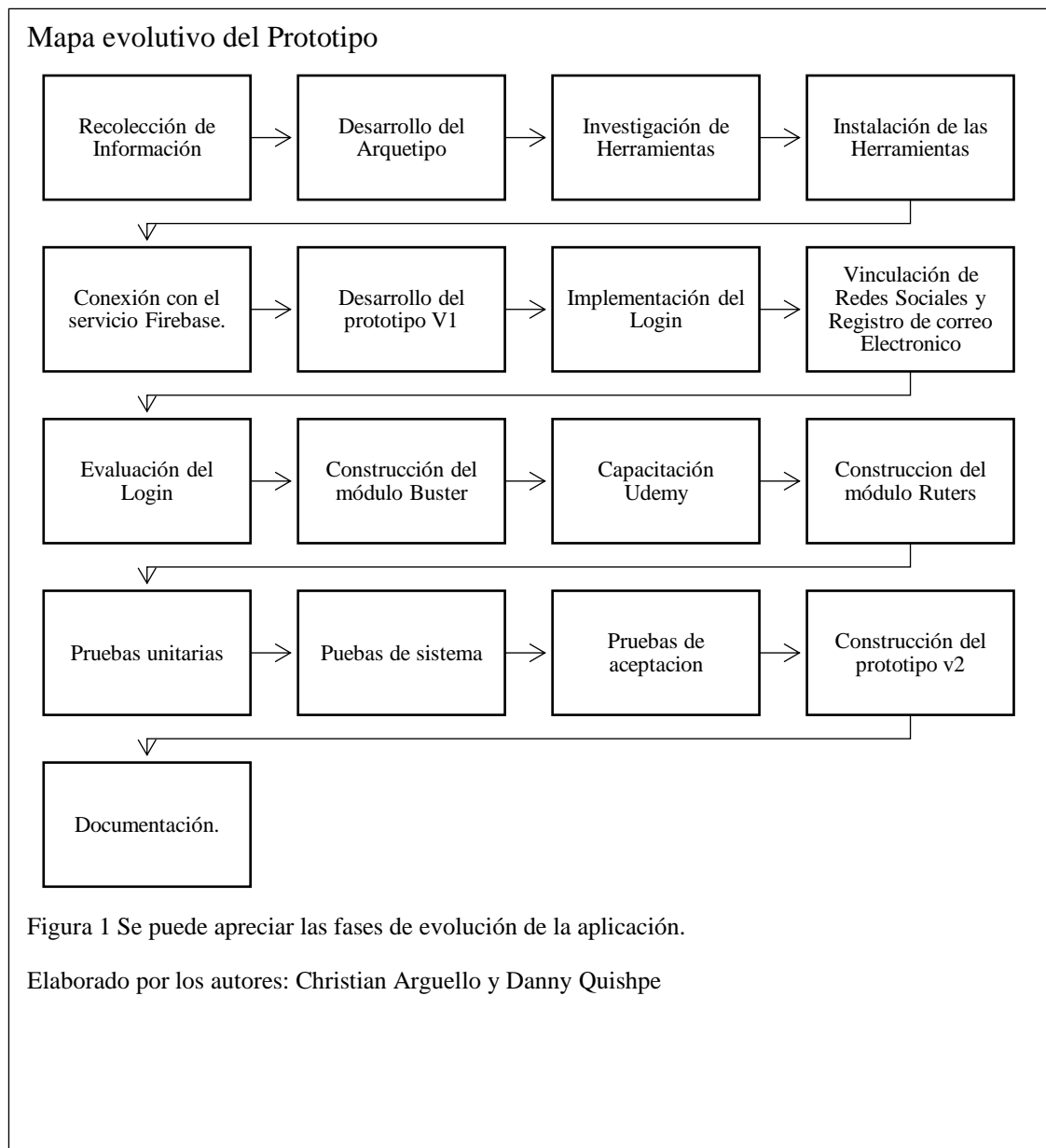


Figura 1 Se puede apreciar las fases de evolución de la aplicación.

Elaborado por los autores: Christian Arguello y Danny Quishpe

### **2.1.2 Alcance del proyecto**

La aplicación para el monitoreo de transporte público en la ciudad de Quito permitirá a los ciudadanos evaluar a las unidades y visualizar a los usuarios que se encuentren en dichas unidades, los cuales servirán como medio para proporcionar la información del estado, trato, servicio y entre otros aspectos relevantes para una posible mejora de la calidad del servicio.

La evaluación se realizará mediante un sistema de calificación por estrellas, que servirá para evaluar cada aspecto ya mencionado de la unidad. Además, se podrá poner de manera opcional un comentario. Por el momento, el prototipo no contará con la visualización de los comentarios dentro de la aplicación. El usuario además tendrá la opción de ver la ubicación de las unidades de transporte en tiempo real cuando un usuario de la aplicación se encuentre conectado dentro del módulo y que se encuentre en movimiento.

Al igual que en muchas aplicaciones, es requerido un método de inicio de sesión, que hará uso de redes sociales como Facebook y Google. Para ello se usó el Modelo Cliente-Servidor, para que el usuario no tenga que realizar varias peticiones de acceso externo a las del prototipo.

En el caso de no tener dichas cuentas, se podrá hacer el registro mediante un correo electrónico. Este inicio de sesión dará acceso a la pantalla inicial del prototipo en donde los usuarios podrán evaluar a las unidades (a estos usuarios los denominaremos Busters) y podrán ver a los pasajeros en movimiento en el mapa (a estos los denominaremos Ruters).

Para controlar a la aplicación se utilizó el modelo MVC que sirve para dividir la aplicación por módulos y estos a su vez hacen uso de controladores de

almacenamiento y de ubicación, los mismos que se encuentran en los módulos de Inicio de sesión, Buster y Ruter.

Al ser un prototipo no cumplirá con las acciones como recuperación de la contraseña en el caso de que el usuario se haya creado una cuenta con correo electrónico, el mapa no se encuentra sincronizado con el módulo Buster para visualizar la calificación general de las unidades; estas actividades se las tiene planteadas para investigaciones futuras. La aplicación no se encontrará en Play Store hasta que toda su investigación culmine y tenga todas las funcionalidades implementadas.

### **Origen de las palabras Buster y Ruter**

Estas palabras son creación de los autores y surgen como un juego de palabras a partir de los vocablos bus, ruta/route y user.

#### **Buster**

Es el usuario que tendrá la opción de calificar a las unidades de transporte al igual que poner un comentario opcional que se subirá a la base de datos para su respectiva evaluación. Este usuario podrá ingresar el número de la unidad emitida por la ANT, también podrá marcar de 1 a 5 estrellas siendo 1 la más baja y 5 la más alta calificación.

#### **Ruter**

Es el usuario que podrá hacer uso del mapa, en el cual se visualizará las unidades de transporte en un rango de 5 Km a la redonda, en tiempo real.

### **2.1.3 Análisis de requerimientos**

El análisis de requerimientos nos permite comprender el problema del cual estamos partiendo para poder establecer la mejor solución posible. Además, nos permitirá realizar las pruebas una vez que la aplicación esté desarrollada. Este proceso permite comprobar cómo se segmentará la solución.

A continuación, se detallarán los módulos de la aplicación móvil que solventarán las necesidades de los usuarios.

#### Módulo de login y registro de usuarios

Dicho módulo se realizó en base a la comparación de varias aplicaciones con inicio de sesión y registro de usuario, estas opciones fueron usar el servicio de Google, Facebook y por medio de correo electrónico.

#### Módulo de Calificación

En este módulo los usuarios podrán ingresar el número de la unidad y calificarla en cinco categorías: seguridad, estado de la unidad, comodidad, trato del conductor y tiempo.

#### Módulo de Geolocalización

En este módulo los usuarios pueden ver las unidades en movimiento en un rango de 5 km.

### **2.1.4 Análisis de factibilidad**

Para evaluar las probabilidades de éxito del proyecto se realizó el estudio tanto de la factibilidad técnica como operacional, mismo que se detalla a continuación.

#### **2.1.4.1 Factibilidad técnica**

La aplicación se ejecutaría en un dispositivo móvil, la cual se conectará directamente a Firebase de una manera cliente-servidor para poder almacenar la información y ocupar los servicios tanto para el inicio de sesión como para la analítica de los datos.

#### **2.1.4.2 Hardware:**

Para poder garantizar un mejor funcionamiento de la aplicación se tomará en cuenta las siguientes características.

- Dispositivo móvil con GPS integrado.

- Sistema Operativo (S.O) Android 4.1.
- Memoria interna 16 GB.
- Memoria RAM 2 GB.

Tomando en cuenta los requerimientos antes mencionados se trabajará para el desarrollo con un emulador y dos dispositivos móviles.

### 2.1.4.3 Software:

Para el desarrollo de la aplicación se utilizó IDE (Interfaz de desarrollo) Android Studio la cual trabaja con XML y un lenguaje de programación en JAVA/Kotlin debido a que es un lenguaje fácil de manejar y contiene varios repositorios en donde se puede solventar cualquier error o duda que se tenga durante el desarrollo de la aplicación.

La aplicación móvil va a ir sujeta a herramientas de base de datos y de geolocalización para las cuales se hace uso del conjunto de servicios Firebase y herramientas de Google API keys. Estas herramientas son ideales gracias a su gran variedad de servicios y a que no ocupan demasiados recursos, ya que son servicios que se encuentran de manera online.

Para la instalación de las herramientas de desarrollo se describe a continuación los requerimientos mínimos:

Tabla 1 Requerimientos mínimos

Software	Hardware	Descripción
Android Studio 4.0.1	2GB RAM. 500 MB de almacenamiento. 1GB para SDK de Android. Java Development Kit (JDK) 7 o superior.	Para el desarrollo de la aplicación en XML y Java es un IDE básico el cual se adapta a estos dos lenguajes de programación.
Node Js	2GB RAM. 500 MB de almacenamiento.	Entorno JavaScript que nos permite ejecutar en el servidor, de manera asíncrona, con una arquitectura orientada a eventos.

Nota: Requerimientos para desarrollar el proyecto

Con este estudio se pueden observar las características mínimas del computador y el dispositivo donde se realizará el desarrollo y las pruebas de la aplicación.

#### 2.1.4.4 Factibilidad operacional

Para un correcto funcionamiento de la aplicación es requerido realizar el análisis operacional el cual nos servirá para poder identificar tanto a los desarrolladores como a quienes va dirigida la aplicación, ya que para el uso de esta es necesario que el usuario tenga control sobre los dispositivos móviles y sepa configurar la aplicación en su primer uso, ya que debe otorgar permisos de ubicación al igual que introducir texto donde se lo solicite.

Tabla 2 Recursos Operacionales

Recurso	Cantidad	Módulo	Descripción
Desarrollador	Se dispone de dos desarrolladores con conocimiento intermedio sobre el desarrollo de aplicaciones móviles, que conocen el lenguaje de programación JAVA, XML y diseño y arquitectura de aplicaciones.	Módulo de ingreso al sistema. Módulo de Calificación (Busters). Módulo de Geolocalización (Ruters).	Crear una ventana en donde se obtenga los métodos de inicio de sesión al igual que el registro de una cuenta nueva para poder acceder al sistema. Crear un módulo donde esté una forma de ingreso del número de la unidad acompañado con un sistema de calificación por estrellas las cuales se redireccionarán a un comentario para que se guarde en la base de datos. Crear un mapa donde se obtenga la ubicación precisa, generar marcadores que se guardarán en la base de datos para poder visualizar a los usuarios que hagan uso de ese módulo.
Curso UDEMY (Capacitante)	Se requiere un curso para el desarrollo de la aplicación.	Módulo de Geolocalización.	Se debe tomar un curso para poder capacitarse acerca del uso del mapa al igual que sus configuraciones para un control mediante marcadores.

Nota: Información de los recursos utilizados



## 2.2 Implementación de Metodología

Para la dirección correcta del proyecto se optó por seguir la metodología ágil SCRUM, ya que es una de las más utilizadas en el desarrollo y la gestión de proyectos ya sean estos grandes o pequeños. Esta metodología ayuda a los miembros del equipo a estar en continuo contacto con el o los interesados del proyecto. Gracias a su capacidad de trabajar en segmentos llamados “sprints”, la metodología SCRUM hace posible realizar cambios o modificaciones durante el proyecto, permitiendo al equipo de trabajo añadir los cambios a tiempo sin afectar el cronograma de desarrollo y continuar con las tareas pendientes de manera regulada por los sprints.

Una buena práctica de SCRUM es la definición de roles, entre los cuales se puede tener un auspiciante externo, que podría opinar sobre el activo final del proyecto. Se definen a los siguientes involucrados del presente proyecto:

### 2.2.1 Roles

#### Patrocinador

Tabla 3 Patrocinador

Nombre	Cargo	Función
Holger Ortega	Patrocinador.	Se encargará de que se cumplan los objetivos del proyecto.

Nota: Información del patrocinador del proyecto

#### Scrum Master

Tabla 4 Scrum Master

Nombre	Cargo	Función
Christian Arguello	Scrum Master.	Se encarga de ser intermediario con todos los interesados.

Nota: Información detallada del Scrum Master

## Equipo de desarrollo:

Tabla 5 Equipo de desarrollo

Nombre	Cargo	Área	Función
Christian Arguello	Desarrollador.	Sistemas.	Se encargan de la implementación del software.
Danny Quishpe			

Nota: Información del equipo de desarrollo del proyecto

## 2.2.2 Cronograma de Actividades

Son todas las actividades que se tienen que cumplir en un determinado tiempo para el avance del proyecto.

Tabla 6 Cronograma de Actividades

Nombre de Tarea	Duración	Comienzo	Fin
Prototipo aplicación móvil para el monitoreo en la ciudad de Quito.	151 días	vie 03/4/20	vie 30/10/20
Diseño del Arquetipo de la Aplicación.	20 días	vie 03/4/20	jue 30/4/20
Configuraciones iniciales.	8 días	vie 01/5/20	mar 12/5/20
Opciones de Inicio de sesión.	22 días	jue 14/5/20	vie 12/6/20
Pruebas de Inicios de sesión.	6 días	vie 12/6/20	vie 19/6/20
Desarrollo módulo Busters.	11 días	vie 19/6/20	vie 03/7/20
Desarrollo módulo Ruters.	11 días	lun 06/7/20	sáb 18/7/20
Pruebas y Mejoras.	25 días	lun 20/7/20	vie 21/8/20

Nota: Información de las Actividades a realizarse en el proyecto

### 2.2.3 Sprints

Mediante el cronograma de actividades se decidió realizar los entregables por las fechas señaladas en la tabla No. 4. Dichos entregables tendrán subtareas que el equipo de desarrollo establece para la revisión con el patrocinador y para dar un seguimiento constante a las tareas por entregar. Los sprints se detallan a continuación segmentando a la aplicación por módulos.

#### 2.2.3.1 Sprint 1

En las reuniones, el Scrum Master y el Patrocinador deciden como primera tarea realizar el diseño de la aplicación, a la cual se le llamará el “arquetipo” de la aplicación. En este se definirá cuáles son las ventanas que tendrá la aplicación al finalizar y sus respectivas funciones. Este sprint se puede visualizar en la figura No 2.

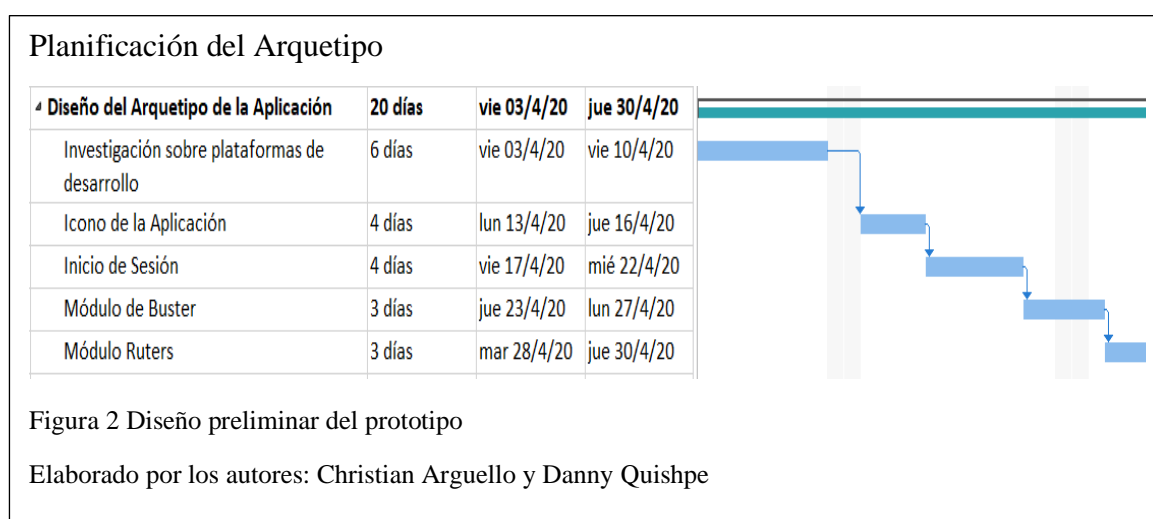


Figura 2 Diseño preliminar del prototipo

Elaborado por los autores: Christian Arguello y Danny Quishpe

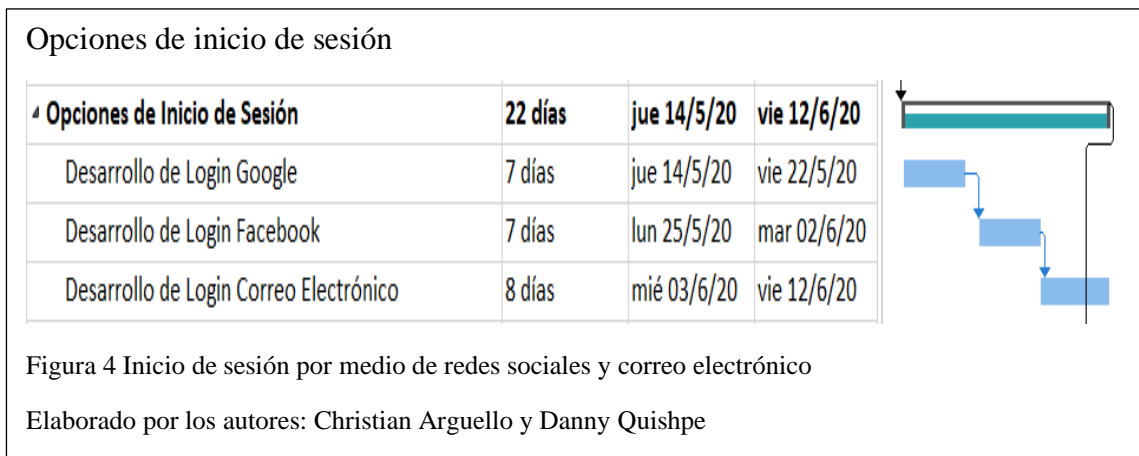
#### 2.2.3.2 Sprint 2

Una vez realizado el sprint 1 se pidió en la siguiente reunión realizar la aplicación en Android Studio. Se comenzó con las instalaciones y configuraciones de la aplicación del desarrollo y con los medios que se integrarán a la aplicación para el inicio de sesión, tal como se muestra en la figura No. 3.



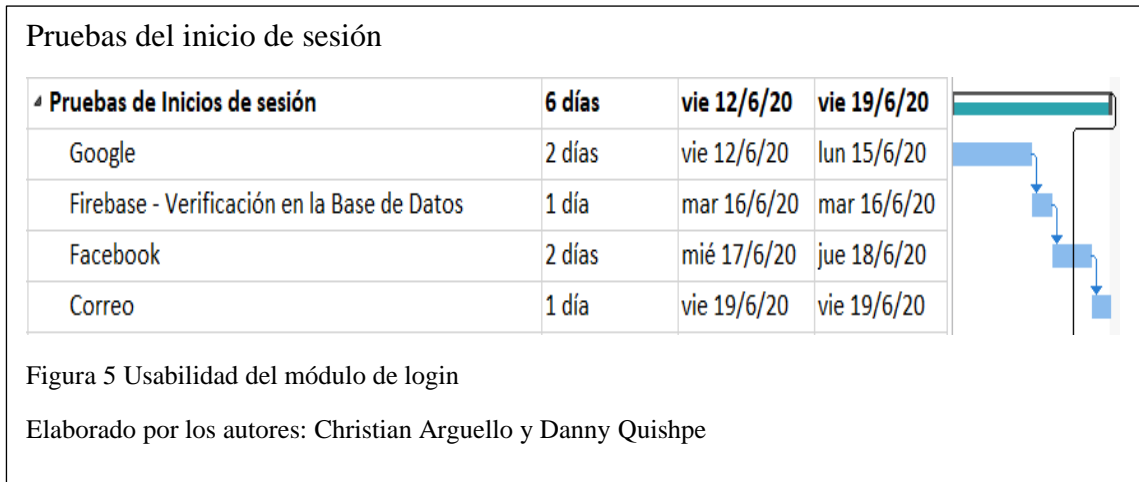
### 2.2.3.3 Sprint 3

En el sprint 2 se logró realizar la primera ventana, la del inicio de sesión. A continuación, el patrocinador pidió que se realice las opciones de login más conocidas, como son las redes sociales o, alternativamente, por correo electrónico. Esto se detalla en la figura No. 4.



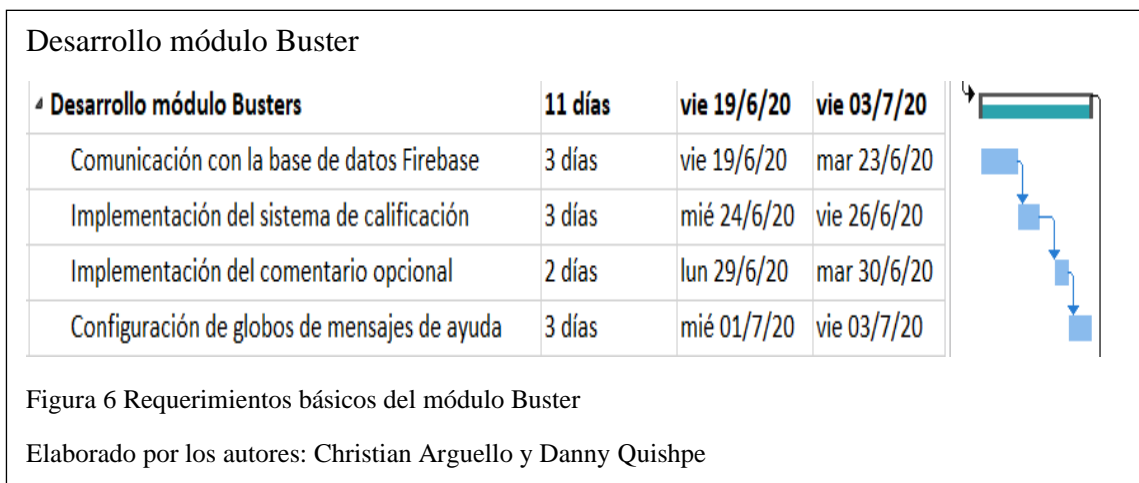
### 2.2.3.4 Sprint 4

Una vez culminado con la implementación de los métodos de inicio de sesión descrita en el sprint 3, se realizaron las pruebas demostrando que se ingresa con las cuentas de las redes sociales y con el correo. Esto se revisó además en la base de datos, donde se observa el método por el cual los usuarios ingresan a la aplicación. Este sprint puede visualizarse en la figura No. 5.



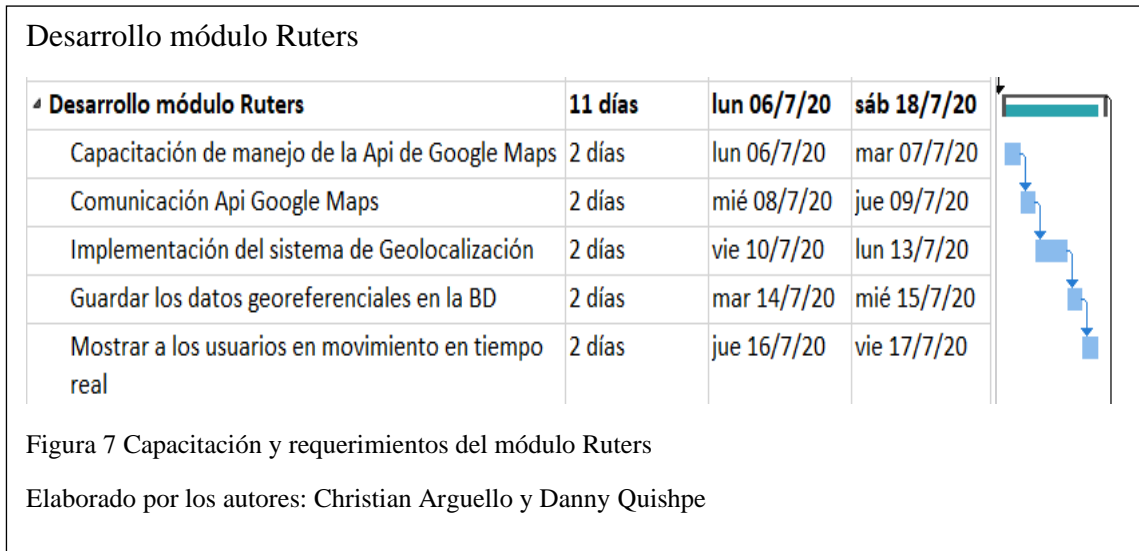
### 2.2.3.5 Sprint 5

A continuación se acordó realizar el módulo de Busters; en la figura No. 6 se muestran las funciones que debe cumplir, según los requerimientos que el patrocinador pide como básico para poder realizar la calificación.



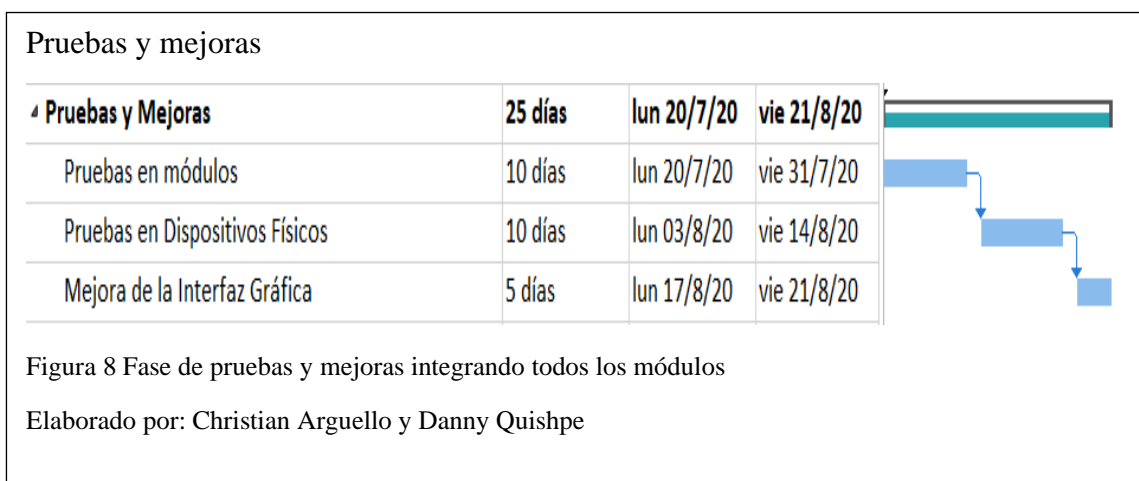
### 2.2.3.6 Sprint 6

Para la siguiente reunión se planteó realizar la ventana de Rutgers, tarea para la cual se tuvo que tomar un curso, dada la complejidad del uso del mapa y las API de Google. El curso fue indispensable para saber cuáles son los métodos requeridos para implementar las funciones y configuraciones básicas de un mapa, al igual que el manejo de los métodos para implementar los marcadores que serán de referencia a cada usuario. Este sprint puede ser visualizado en la figura No. 7.



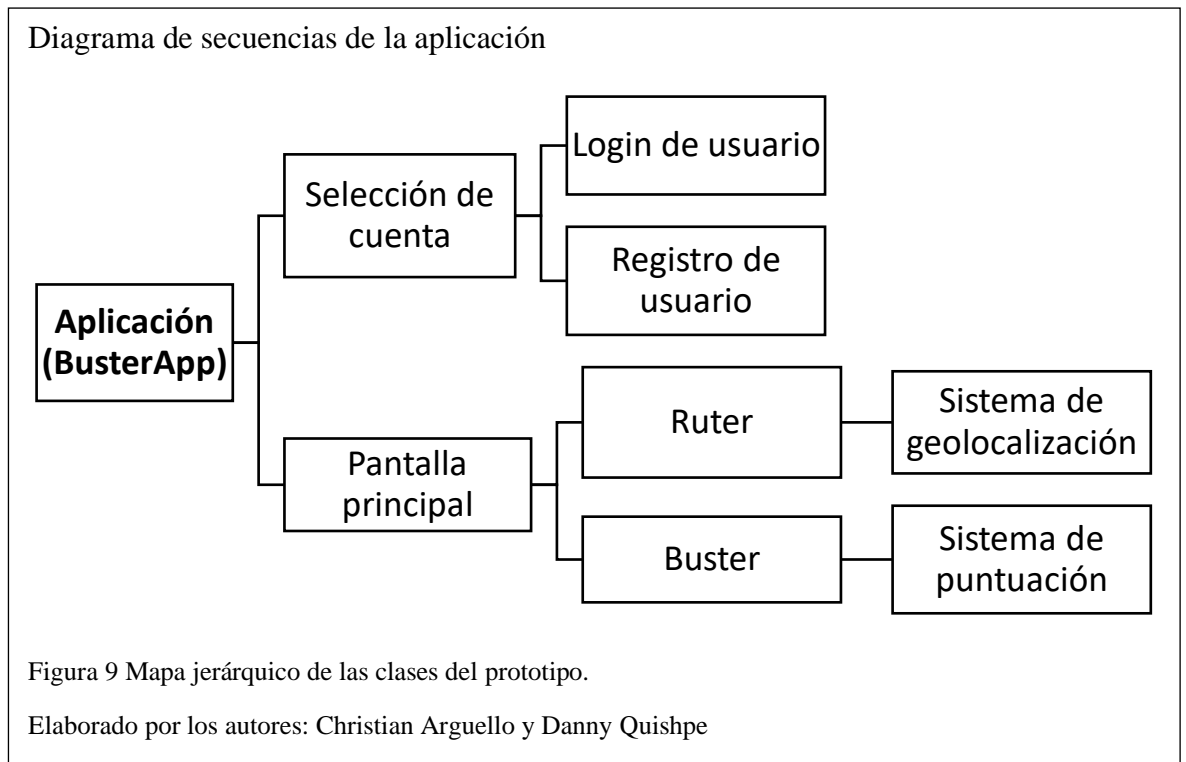
### 2.2.3.7 Sprint 7

Una vez culminadas las funciones básicas del prototipo, que eran los métodos de inicio de sesión y los módulos Ruters y Busters, se pasó a la parte de mejora de la interfaz gráfica y a las pruebas de la aplicación por módulos . Para estas y futuras pruebas, se instaló la aplicación en dispositivos físicos. Este sprint se muestra en la figura No. 8.



### 2.2.4 Mapa de la Aplicación

Siguiendo los requerimientos de alto nivel se realizó un esquema en donde se presentó al patrocinador, cómo sería la conexión y flujo de la aplicación, esto se describe en la figura No. 9.



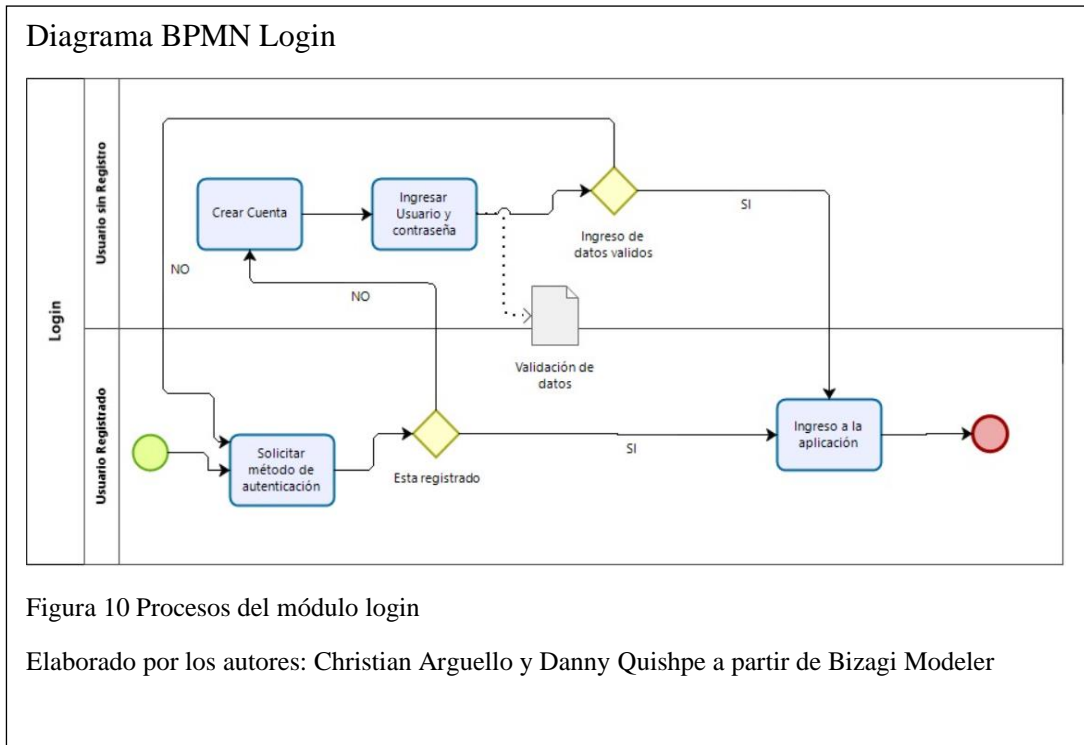
#### 2.2.4.1 Módulo Login

##### Objetivo

Realizar el inicio de sesión para acceder a la aplicación mediante el uso de redes sociales o creación mediante correo electrónico.

##### Descripción

El proceso del login consiste en la adquisición de datos del usuario como nombre, correo electrónico, ubicación, etc. Esta información se almacenará en la BD de Firebase y será usada luego por el servicio de Firebase Analytics para apreciar la cantidad de usuarios registrados en la aplicación.



### 2.2.4.2 Modulo Busters

#### Objetivo

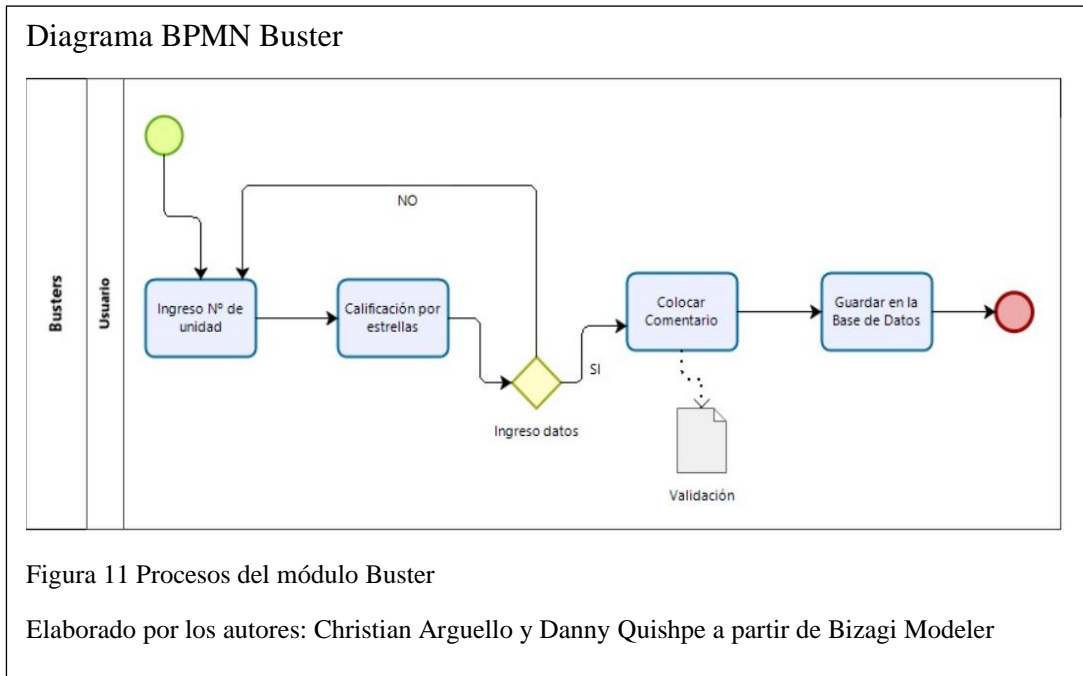
Calificar a las unidades mediante un sistema de valoración de estrellas para poder revisar la información proporcionada y hacer análisis de la misma.

#### Descripción

El proceso de Busters consiste en la calificación de las unidades de transporte, donde se ingresará el número de la unidad al igual que la calificación de cada uno de los aspectos del transporte.

En el caso que el usuario desee, puede publicar un comentario, el mismo que se guardará junto a las calificaciones previas.





### 2.2.4.3 Módulo Ruters

#### Objetivo

Mostrar los usuarios disponibles en el mapa mediante el uso de API de Google Maps y el GPS incorporado en el dispositivo para obtener la información de cada usuario e indicar dónde se encuentran en tiempo real.

#### Descripción

Para acceder a la función de Ruters, el usuario deberá conceder los permisos de ubicación y tener activo el GPS.

Diagrama BPMN proceso Ruters

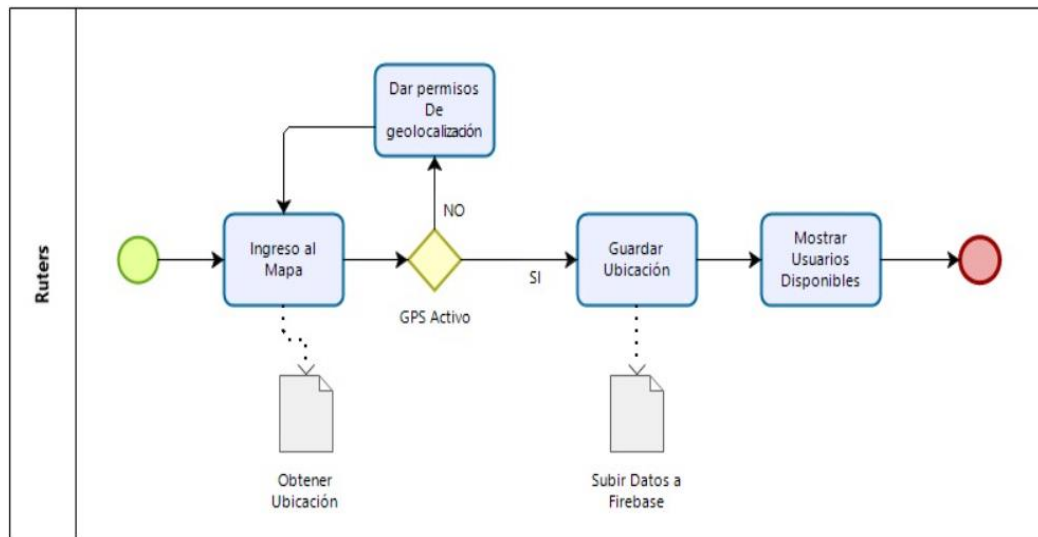


Figura 12 Procesos del módulo Ruters

Elaborado por los autores: Christian Arguello y Danny Quishpe a partir de Bizagi Modeler

#### 2.2.4.4 Diagrama de la arquitectura general del prototipo

En la figura No. 13 se puede observar la arquitectura general del sistema con los servicios que la aplicación utiliza. Con esto detallamos que se está usando la arquitectura cliente-servidor y la arquitectura MVC.

Diagrama de la arquitectura general del sistema

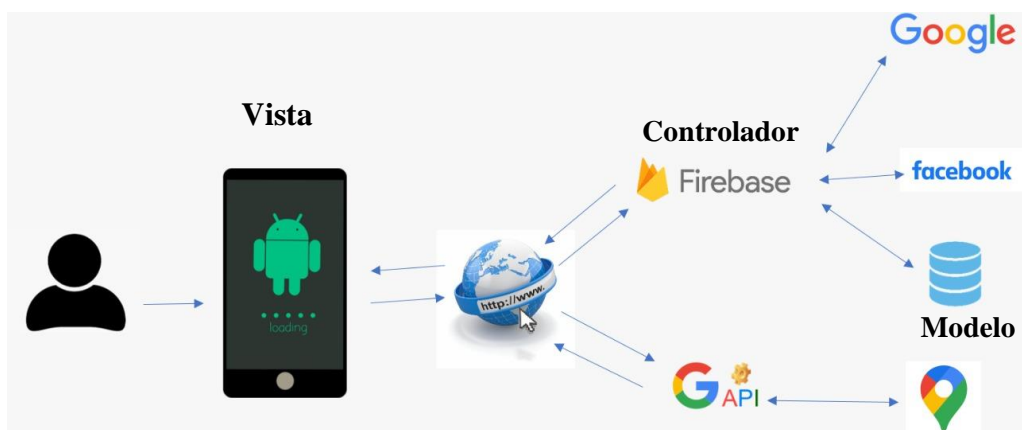


Figura 13 Funcionamiento de la arquitectura del sistema.

Elaborado por los autores: Christian Arguello y Danny Quishpe

## **CAPÍTULO III**

### **DESARROLLO E IMPLEMENTACIÓN**

#### **3.1 Aplicación Móvil**

En esta sección se describe el procedimiento que se utilizó para el desarrollo, configuraciones y las funciones principales, así como obtención de datos del usuario, guardar información, geolocalización y cómo se comunican entre ellos.

#### **3.2 Diseño de la aplicación**

Para el desarrollo de la aplicación se procedió con el diseño del arquetipo de la aplicación, para lo cual se hizo uso de la aplicación Pencil, que permite generar las posibles pantallas de carga. Esto permite tener una visión de cómo estará realizada la aplicación sin ninguna función, de manera que el Dueño del producto pueda dar su punto de vista y disponer cambios en el diseño antes de ser realizada la aplicación.

#### **3.3 Arquetipo**

En el primer boceto tenemos la pantalla principal en donde se muestra el nombre de la aplicación, los métodos de inicio de sesión y el botón que permitirá hacer el registro en el caso de ser necesario. El boceto inicial de la pantalla principal se muestra en la figura No. 14.

### Pantalla de inicio del arquetipo

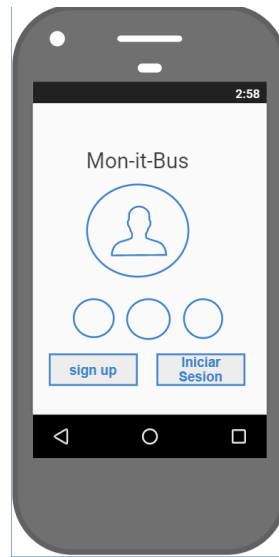


Figura 14 Boceto de la pantalla inicial

Elaborado por los autores: Christian Arguello y Danny Quishpe a partir de pencil

Una vez iniciada la sesión, el usuario podrá elegir qué función va a utilizar (Busters o Ruters). Existirá un botón de ayuda que despliega información acerca de cada función. La pantalla correspondiente se muestra en la figura No. 15.

### Pantalla principal del arquetipo

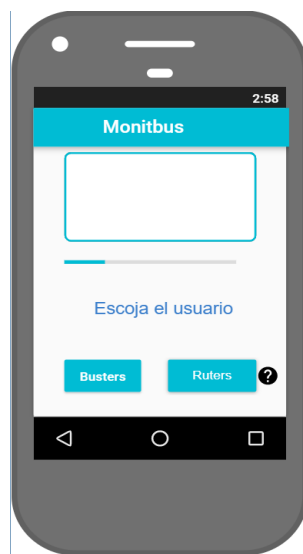


Figura 15 Boceto de la pantalla de principal

Elaborado por los autores: Christian Arguello y Danny Quishpe a partir de pencil

Si el usuario optó por Buster, se le mostrará la opción de calificación por estrellas y un cuadro de texto para introducir un comentario (opcional). Al igual que en la pantalla principal, en esta pantalla habrá globos de ayuda con la descripción de cada aspecto a calificar.



En el módulo de Ruters el usuario tendrá la opción de ver su ubicación actual junto con la de los usuarios de la aplicación que se encuentren alrededor de alguna unidad de transporte.

## Arquetipo de la pantalla del Ruter



Figura 17 Boceto módulo ruters

Elaborado por los autores: Christian Arguello y Danny Quishpe a partir de pencil

Para que el usuario tenga una mejor experiencia se decide cambiar la presentación de información en forma de texto por la presentación de la misma en un mapa. Esto le permitirá ver la ubicación de las unidades de transporte, tiempos de llegada y tomar decisiones en base a la calificación de las unidades (el promedio de las calificaciones de todos los Busters previos).

## Arquetipo de la pantalla del Ruter, versión 1



Figura 18 Boceto actualizado del módulo Ruters

Elaborado por los autores: Christian Arguello y Danny Quishpe a partir de pencil

### 3.4 Configuraciones iniciales

En primer lugar, se crea el proyecto en Firebase para poder hacer uso de los métodos de inicio de sesión. Además, esta plataforma ofrece la opción de base de datos de la cual se va a hacer uso más adelante.

## Configuración de Firebase

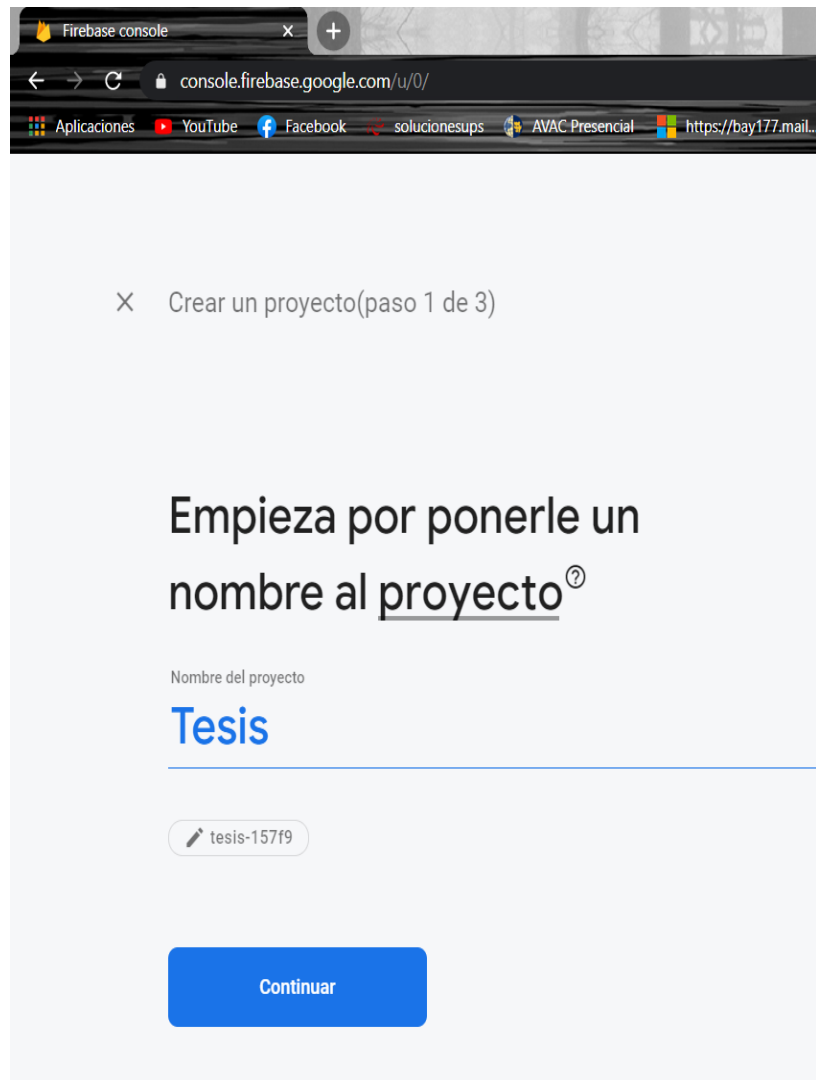


Figura 19 Nuevo proyecto dentro de Firebase

Elaborado por los autores: Christian Arguello y Danny Quishpe

Para poder vincular el proyecto con Firebase, Android Studio tiene la herramienta que se muestra en la figura No. 20.



## Conexión Firebase - Android Studio

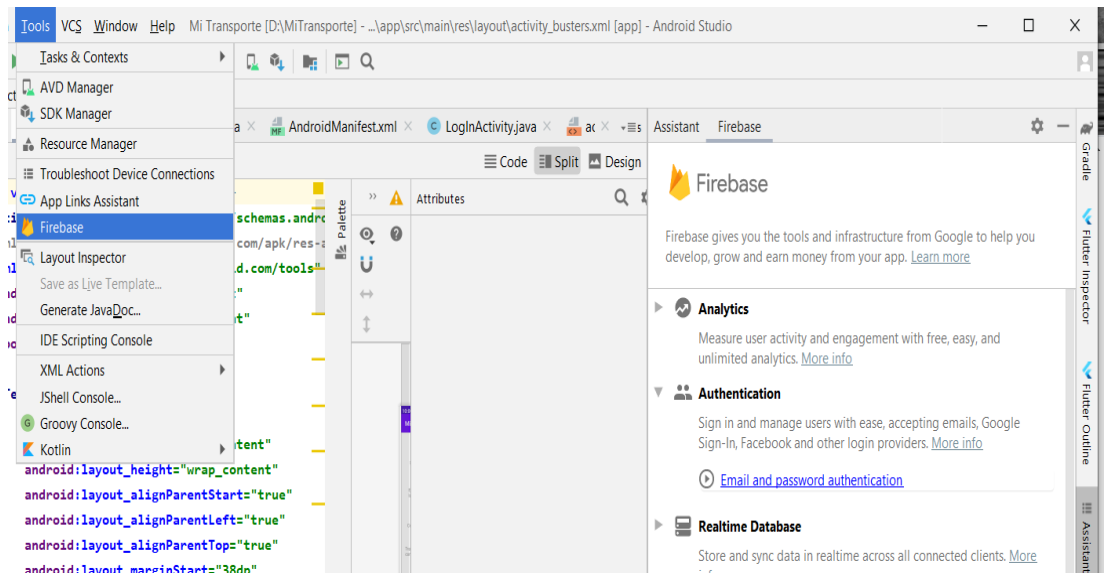


Figura 20 Comunicación inicial entre Firebase y Android Studio

Elaborado por los autores: Christian Arguello y Danny Quishpe

## 3.5 Inicio de sesión

Una vez integrado Android Studio con Firebase procedemos al uso de los métodos de inicio de sesión. En este caso activamos: correo electrónico, Google y Facebook. Estos métodos se deben configurar para hacer uso de la aplicación.

## Servicios de Inicio de Sesión

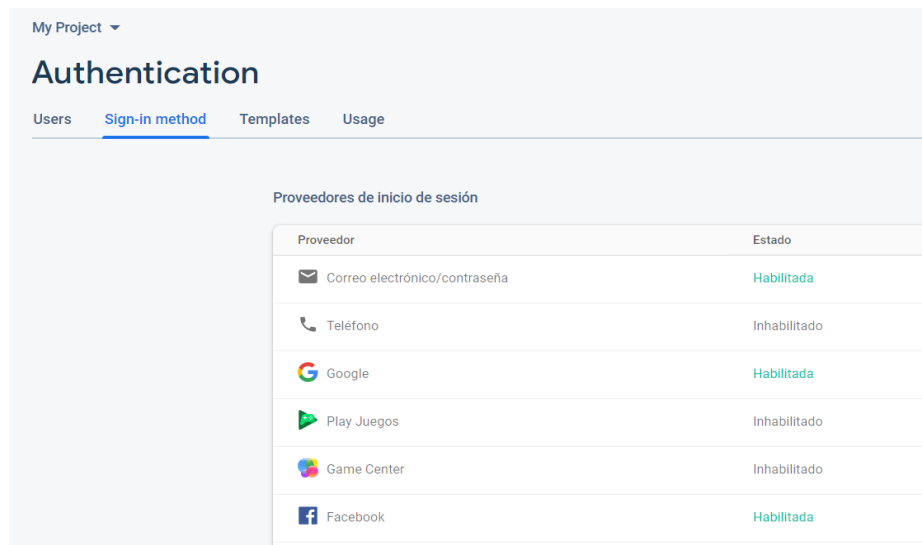


Figura 21 Servicios de autenticación para el módulo login

Elaborado por los autores: Christian Arguello y Danny Quishpe

Para usar el método de inicio de sesión con Facebook es indispensable entrar a Facebook Developer y realizar las configuraciones básicas. Al igual que en Firebase, se crea un proyecto y se configura según el S.O (Sistema Operativo) que se va a utilizar (en nuestro caso, Android).

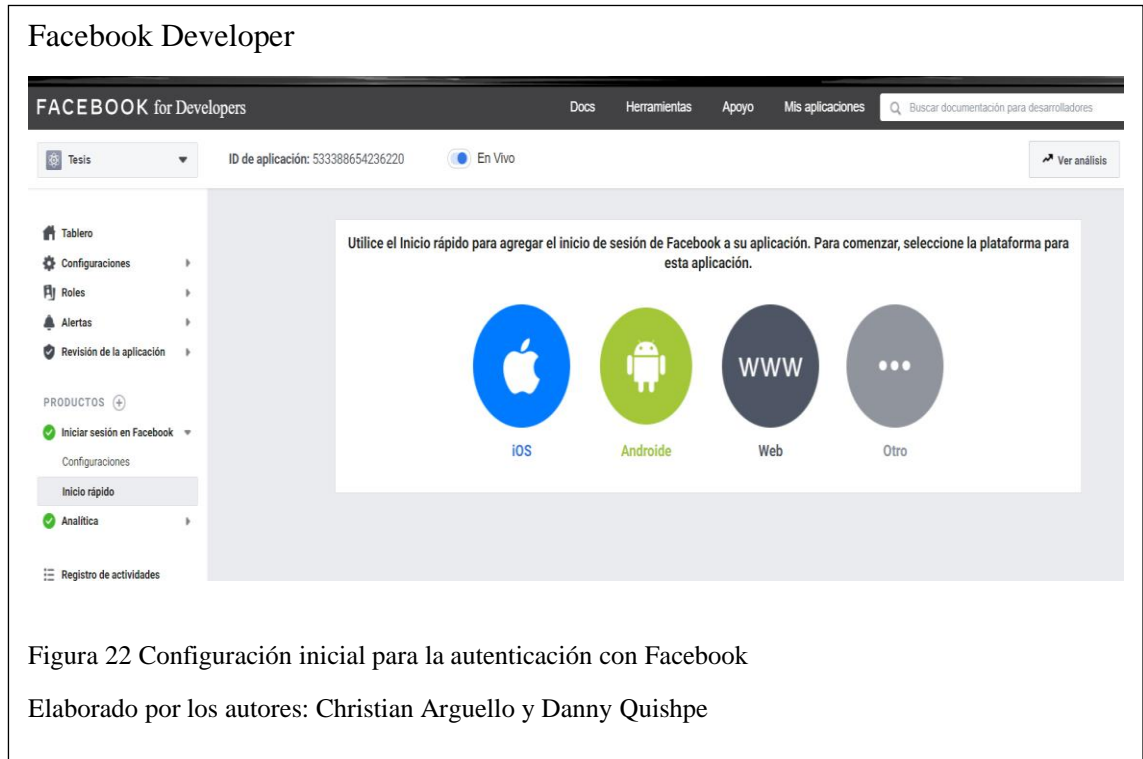


Figura 22 Configuración inicial para la autenticación con Facebook

Elaborado por los autores: Christian Arguello y Danny Quishpe

### 3.6 Creación de las clases

Con la ayuda del arquetipo se pueden crear las clases, las cuales se pueden ver en la figura No. 23. Android Studio nos permite tener una vista previa de cómo está la aplicación, además, nos ofrece un emulador en donde se procedió a instalar la aplicación.

## Clases de la aplicación

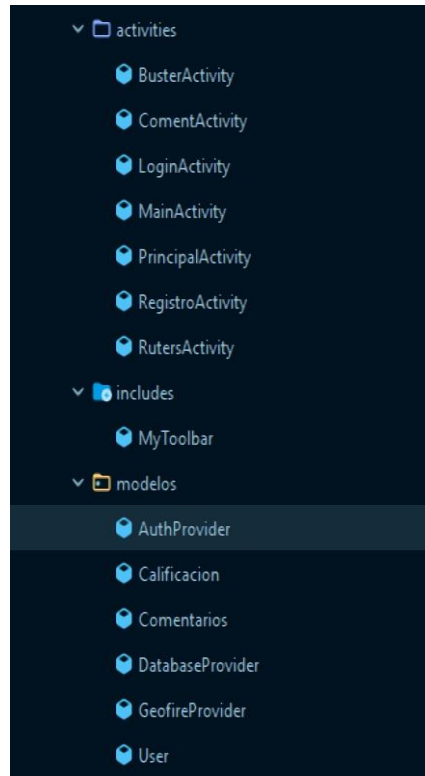


Figura 23 Clases del prototipo usando MVC

Elaborado por los autores: Christian Arguello y Danny Quishpe

### 3.7 Desarrollo de Módulos

En la figura No. 24, se muestra un ejemplo de uno de los módulos, en el cual se observa cómo se desarrolló cada interfaz. En este caso tenemos la ventana de calificación con un fragmento del código, junto a una ventana en donde se puede ubicar cada componente, y una vista previa de la aplicación en un dispositivo.

## Desarrollo de Módulos

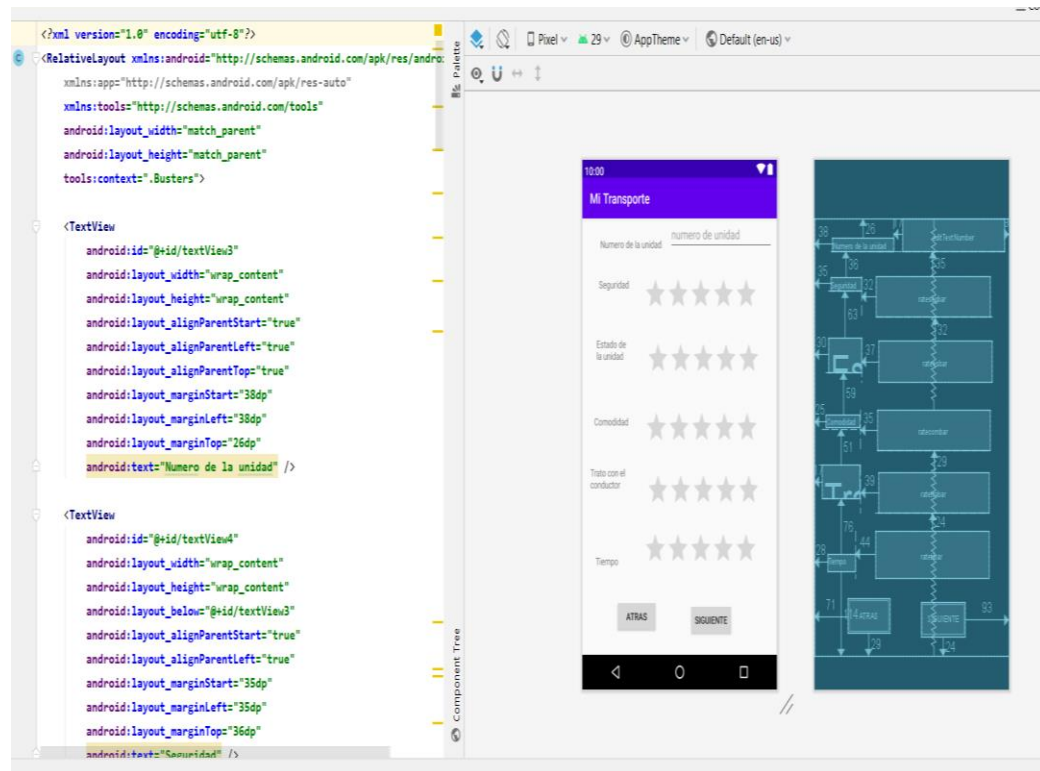


Figura 24 Vista previa de la interfaz del módulo

Elaborado por los autores: Christian Arguello y Danny Quishpe

Siguiendo el diseño visual del arquetipo y los requerimientos, se presenta en la figura No. 25 una vista de la aplicación ya desarrollada, específicamente la pantalla de inicio de sesión.

## Pagina Login v1

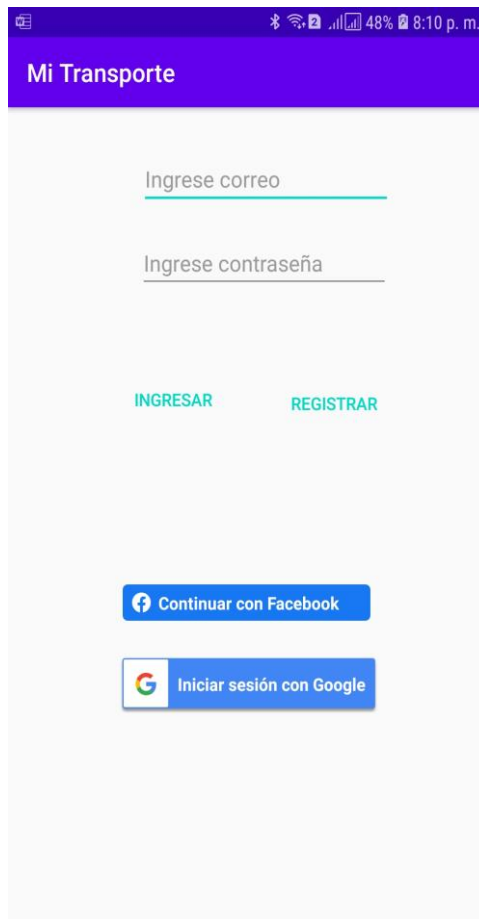


Figura 25 Simulación de la interfaz del login.

Elaborado por los autores: Christian Arguello y Danny Quishpe

Dentro de la aplicación se encuentra la pantalla de bienvenida, donde están los botones de Buster, Ruters y cierre de sesión. Dicha pantalla se muestra en la figura No. 26.

## Pantalla de bienvenida

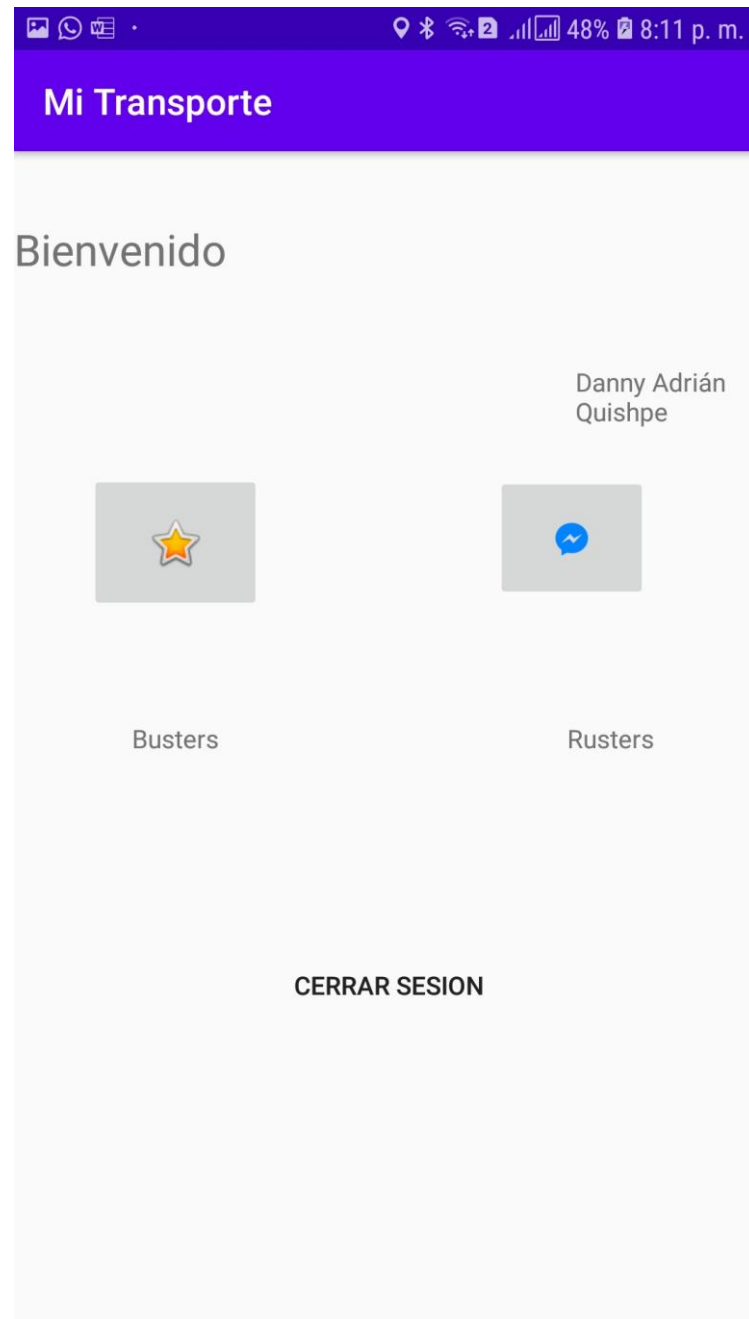
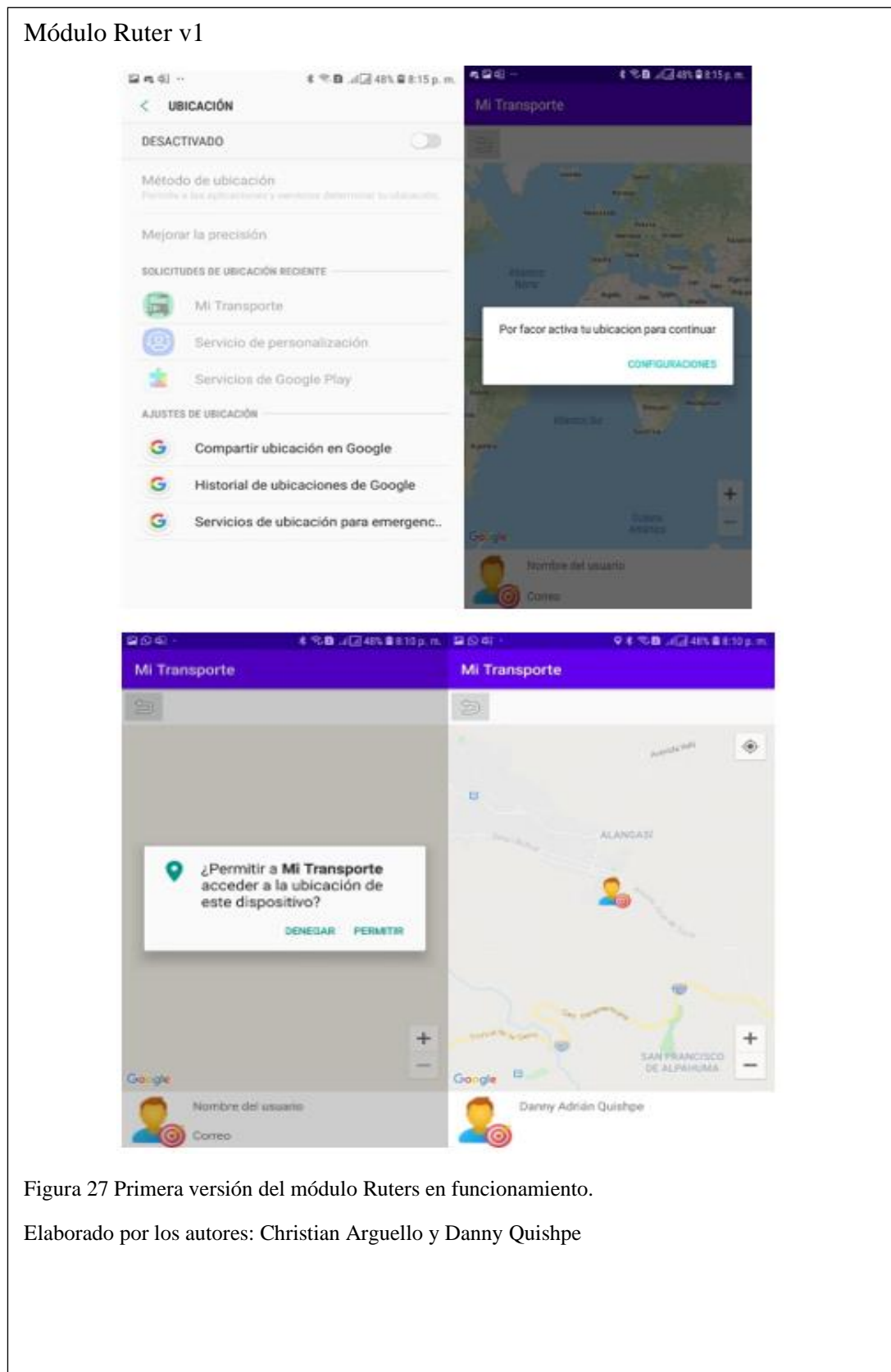


Figura 26 Ventana de bienvenida para los usuarios.

Elaborado por los autores: Christian Arguello y Danny Quishpe

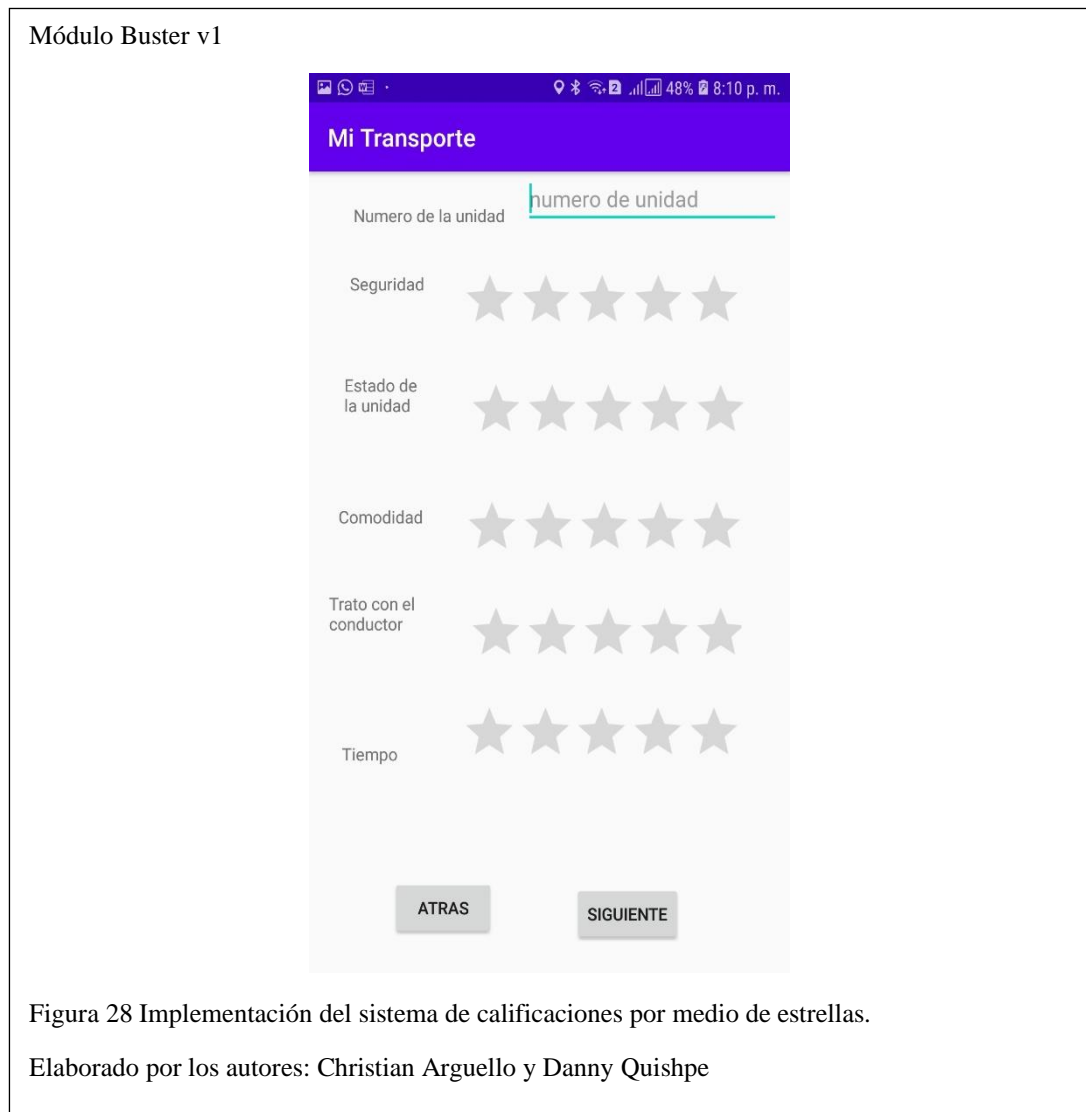
## Ruter

En la figura No. 27 se muestra una pantalla del módulo de Ruter.



## Buster

Dentro de este módulo se solicita el número de la unidad y se presenta un sistema de calificación por estrellas y por categorías. En la figura No. 28 se puede ver la pantalla correspondiente.





## CAPÍTULO IV

### PRUEBAS Y RESULTADOS

#### 4.1 Pruebas Unitarias

Las pruebas unitarias serán utilizadas para verificar que el código y la aplicación estén funcionando según como se tiene esperado, en caso de que no se cumpla, o se produzcan errores, se deben realizar modificaciones para su buen funcionamiento.

##### 4.1.1 Pruebas Login

Las primeras pruebas que se realizaron a la aplicación contaron con la ayuda de simuladores virtuales proporcionados por Android Studio.

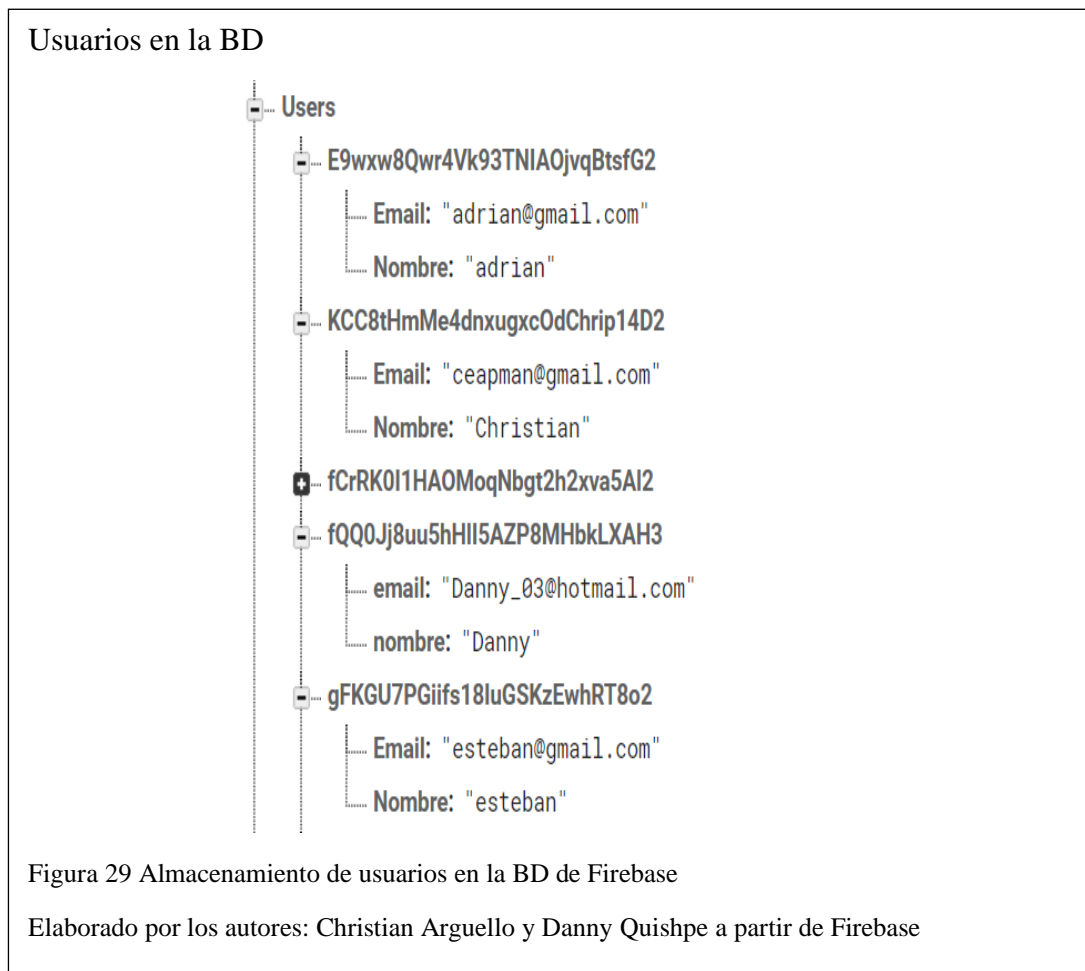
Tabla 7 Pruebas de Login

Código	Escenario	Responsable	Fecha	Resultados	
				Esperado	Obtenido
PRL01	Registro de usuarios mediante Google.	Danny Quishpe	15/5/20	El usuario se registra y la aplicación se comunica con Firebase y le permite la entrada a la aplicación.	La aplicación pasó, pero no se registró por el método de autenticación de Google.
PRL02	Registro de usuarios mediante Facebook.	Christian Arguello	15/5/20	La aplicación se comunica con la API de Facebook y Firebase y crea a un usuario.	La aplicación no se comunicaba con Facebook y no registraba a ningún usuario.
PRL03	Segunda prueba de registro mediante Google.	Danny Quishpe	19/5/20	Resolución con la comunicación de Firebase y Google.	La aplicación pasó con éxito el registro de usuario y mostró el método de autenticación.
PRL04	Segunda prueba de registro mediante Facebook.	Christian Arguello	22/5/20	Resolución con la comunicación con la API de Facebook.	La aplicación exitosamente se comunicó con Facebook y Firebase, registrando al usuario.
PRL05	Registro de usuarios mediante correo electrónico.	Danny Quishpe	25/5/20	Comunicación con Firebase e ingreso de datos del usuario.	Falla con el ingreso de los datos del usuario.
PRL06	Segunda prueba del registro mediante correo electrónico.	Christian Arguello/ Danny Quishpe	29/5/20	Resolución de ingreso de datos.	Exitosamente se registra el usuario.

Nota: Escenarios establecidos para las pruebas en el Login

#### 4.1.2 Pruebas Base de Datos

En la figura No. 29 se muestran los usuarios que se registraron mediante correo electrónico, estos usuarios son guardados por un identificador único en donde, para mayor seguridad, se guarda solo el nombre de usuario y correo electrónico. En el caso de requerir cambio de contraseña, se deberá hacer mediante un administrador de la base de datos. En una versión futura, este proceso debería poderse realizar desde la aplicación.



En la figura No. 30 se muestra los inicios de sesión que se reportaron mediante el uso de las redes sociales y correo electrónico, generando un identificador único para cada usuario.

### Usuarios registrados.











Identificador ↓	Proveedores	Fecha de creación	Inicio de sesión	UID de usuario ↑
danny0adrian@gmail.com		2 jul. 2020	18 jul. 2020	0HNcU491apQ2g1qXryefrLWJqJF2
ad@gmail.com		4 jun. 2020	4 jun. 2020	55Bp4R9KyCdaBDOn8oTGKoaFd0j1
cris@gmail.com		5 jun. 2020	5 jun. 2020	6CHikmBfdyUwSoG05q7e530g5zJ2
ceapman.12@gmail.com		7 jun. 2020	18 jul. 2020	9CKBY5qhwaRdOKXHfGrMhFj23t32
ownerspogo@gmail.com		23 jun. 2020	18 jul. 2020	AIF3wOZXvcgM0uNiEzL8aTV9Lm...
dannyquishpe05@gmail.com		28 jun. 2020	13 jul. 2020	FlnA68dZsdhAkj0jWejqFtoVf7C3
–		18 jul. 2020	18 jul. 2020	GciwA0gC2yWoagMGVVJGCJxsg...
–		21 may. 2020	18 jul. 2020	IrrVN7U8uTRKf2CZ7IPloKOYvpV2
christian_03@hotmail.com		7 jun. 2020	28 jun. 2020	Kn5L59LoVpfw5ocAF5VK6wJpAsv2
ani@gmail.com		11 jun. 2020	12 jun. 2020	MXN72AsYZnNNGLUd3A16jqqt6P...

Figura 30 Servicio de autenticación de usuarios registrados.

Elaborado por los autores: Christian Arguello y Danny Quishpe, generado por Firebase-autenticación

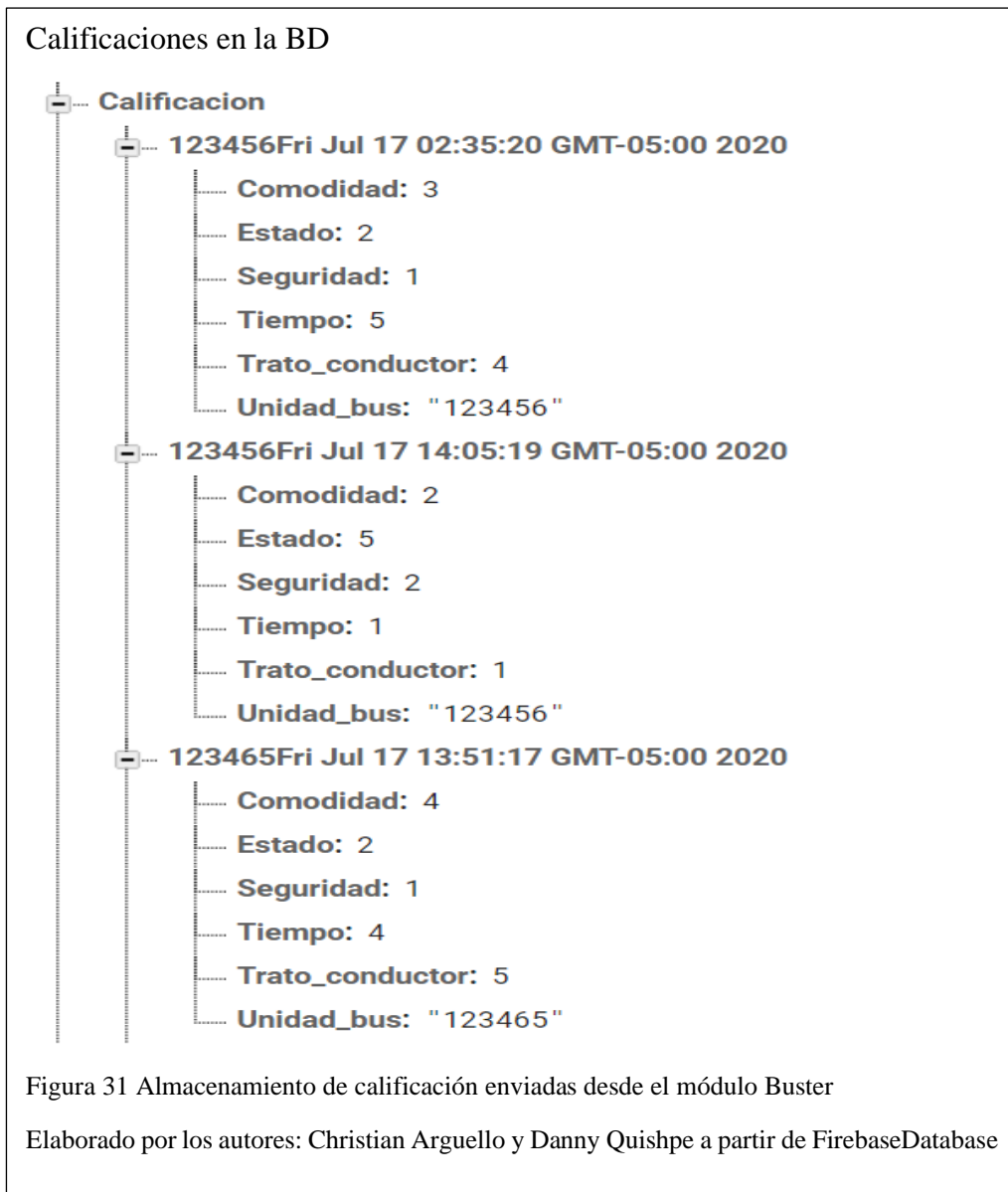
### 4.1.3 Pruebas Buster

Tabla 8 Pruebas módulo Buster

Código	Escenario	Responsable	Fecha	Resultados	
				Esperado	Obtenido
<b>PRB01</b>	Comunicación con la Base de datos.	Danny Quishpe	30/5/20	Comunicación con la Base de Firebase.	La aplicación se comunicó con éxito.
<b>PRB02</b>	Ingreso de los campos dentro de la base de datos.	Christian Arguello	02/6/20	La aplicación envía la información proporcionada por el usuario y la almacena.	Error de tipo de datos.
<b>PRB03</b>	Envío de datos correctos.	Christian Arguello	12/6/20	La aplicación almacenará la información proporcionada por el usuario.	Ingreso exitoso.
<b>PRB04</b>	Envío de comentarios.	Christian Arguello	14/6/20	Se almacena en la base de datos el comentario del usuario.	Ingreso exitoso.

Nota: Pruebas realizadas en el Módulo Buster.

Al igual que con los usuarios, podemos observar en la figura No. 31 que las calificaciones enviadas se guardaron exitosamente. Como se puede observar, la aplicación registra también la fecha en la que fueron enviadas y el número de la unidad. En un trabajo futuro, la aplicación podría verificar la existencia del número de unidad ingresado.



#### 4.1.4 Pruebas Ruters

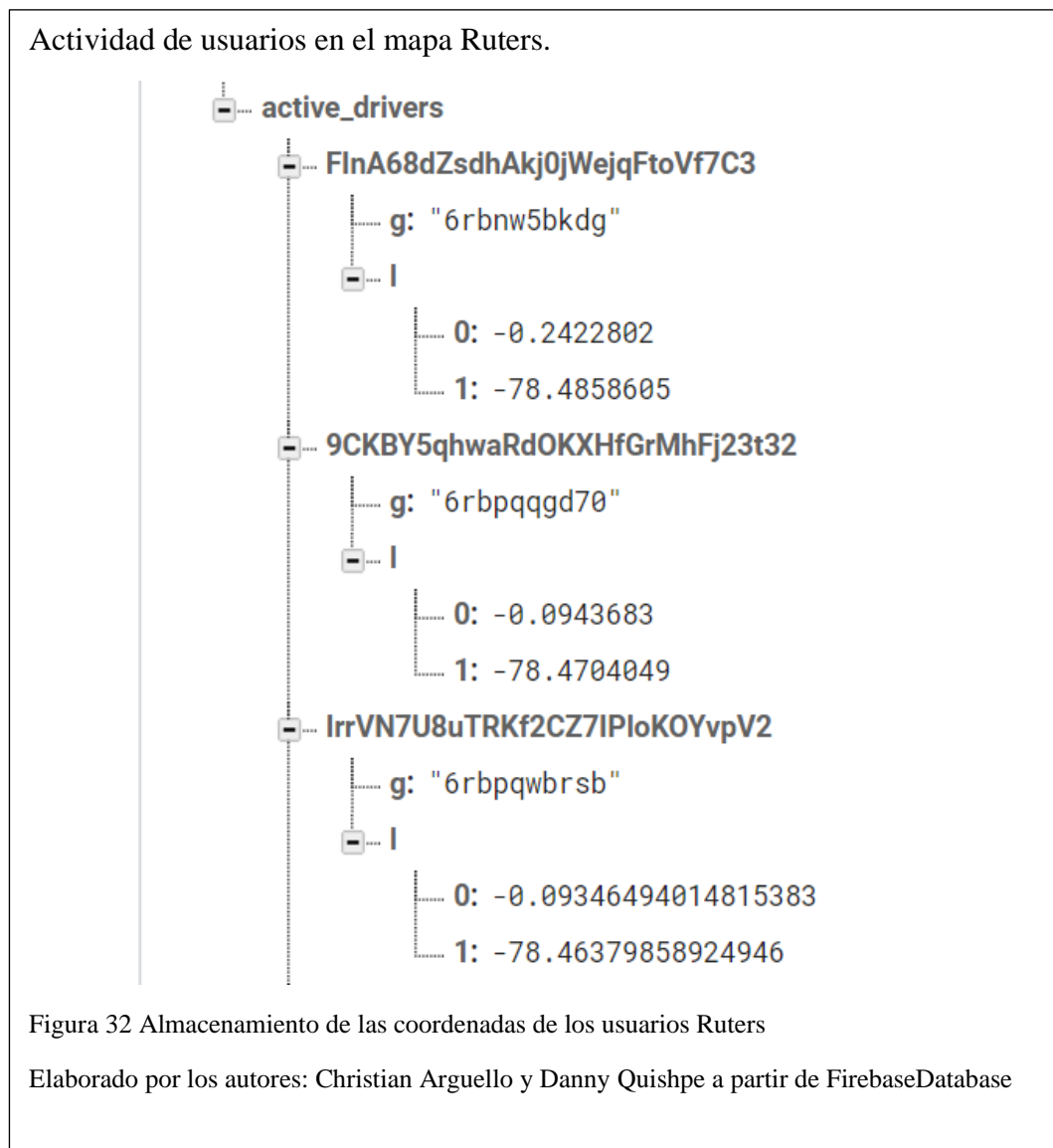
Tabla 9 Pruebas en el módulo de Ruters

Código	Escenario	Responsable	Fecha	Resultados	
				Esperado	Obtenido
<b>PRR01</b>	Comunicación del mapa con el API de Google.	Christian Arguello	12/6/20	Esperar respuesta de acceso al servicio de la API-Maps de Google	La aplicación se comunicó con éxito, mostrando un mapamundi.
<b>PRR02</b>	Ubicación del usuario.	Danny Quishpe	16/6/20	Detecta donde se encuentra el usuario.	Error, el controlador no funcionó.
<b>PRR03</b>	Segunda prueba para la ubicación del usuario	Danny Quishpe	22/6/20	Mostrar el botón de ubicación actual.	La aplicación ya se comunicaba con el GPS del dispositivo.
<b>PRR04</b>	Movimiento automático de la ubicación del usuario.	Danny Quishpe	28/6/20	El mapa de forma automática debe mostrar dónde se encuentra el usuario sin la necesidad del botón.	El mapa automáticamente detecta en donde está el usuario y lo posiciona en el mapa.
<b>PRR05</b>	Detección automática de otros usuarios dentro de un radio.	Danny Quishpe	11/7/20	Cambio de ícono de la ubicación del usuario y detecta al usuario en movimiento.	Error de detección de usuarios en movimiento y el mapa no funcionaba de forma bidireccional al mostrar otros usuarios.
<b>PRR06</b>	Segunda prueba de detección de usuarios en movimiento.	Danny Quishpe	18/7/20	Mostrar usuarios en movimiento y que funcione de forma bidireccional.	La detección fue un éxito en la búsqueda de usuarios en movimiento.

Nota: Pruebas realizadas en el módulo de Ruters

Las tablas. No. 8, No. 9 y No. 10 nos muestran la planificación de pruebas y resultados que se realizaron para los tres módulos de la aplicación, con el fin de ajustar el código de la aplicación mediante los errores que lanzaba el código de programación.

Para que los usuarios puedan ser visibles en el mapa se tomó tanto la latitud como la longitud para posicionarlos y de esta forma ir actualizando su ubicación; tal como se observa en la figura No. 32.



## 4.2 Pruebas de Sistema

Las pruebas de sistema se las realizó analizando el comportamiento de la aplicación cuando se integraron todos los módulos (Login, Buster, Ruter) en función de que cumplan todos los requisitos funcionales previamente ya especificados del prototipo. En la tabla No. 10 se puede visualizar el correcto funcionamiento de los módulos.

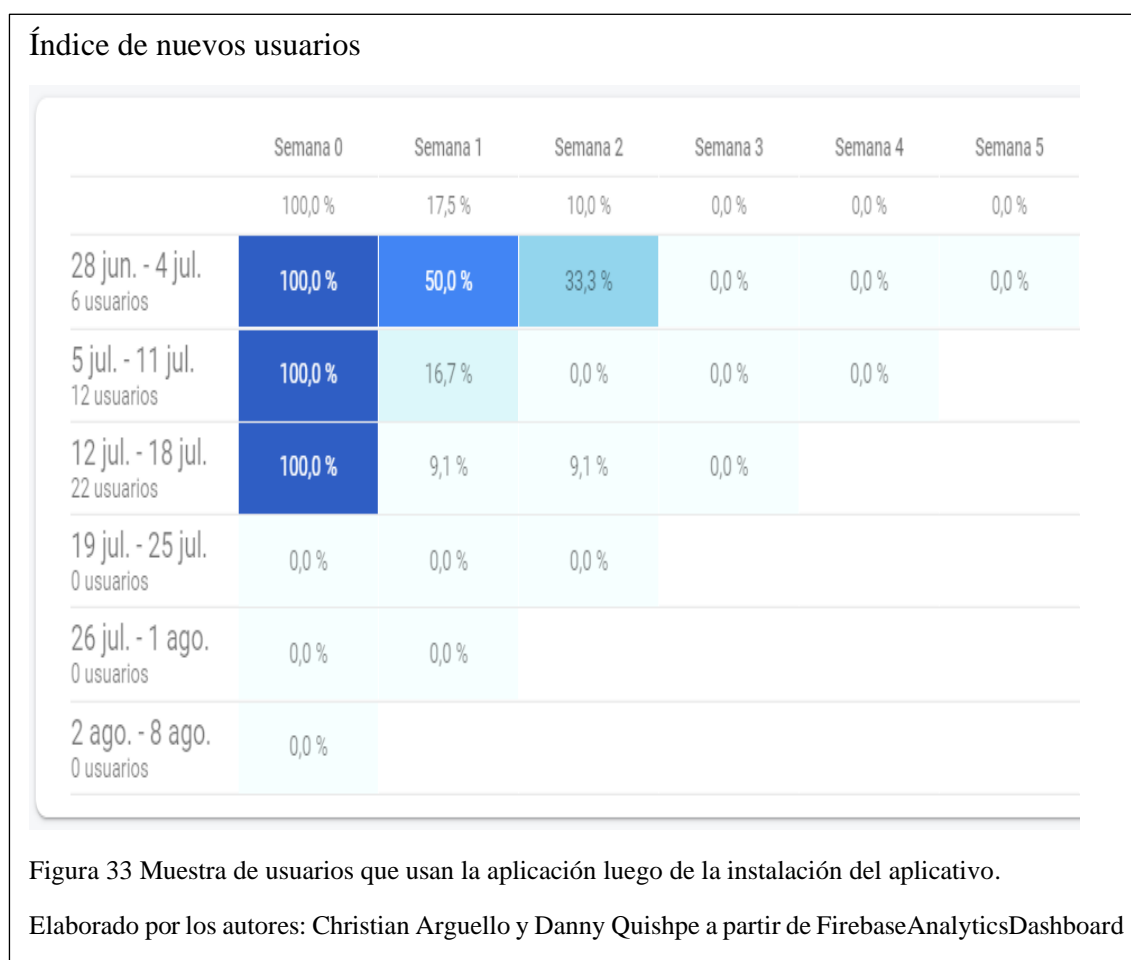
Tabla 10 Funcionamiento de sistema

<b>Código</b>	<b>Módulo</b>	<b>Funcionamiento</b>	<b>Observaciones</b>	<b>Estado</b>
PRF01	Login	<ol style="list-style-type: none"> <li>1. Iniciar la aplicación instalada.</li> <li>2. Ingresar las credenciales y presionar el botón “Iniciar Sesión”</li> </ol>	<p>La contraseña debe contener 6 o más caracteres.</p>	OK, pasó su funcionalidad.
PRF02	Firestore-autenticación	<ol style="list-style-type: none"> <li>1. Autenticar usuarios con Facebook, Google o Correo Electrónico.</li> </ol>	Se logra autenticar a los usuarios por los diferentes modos	OK, logra autenticar usuarios.
PRF03	Buster	<ol style="list-style-type: none"> <li>1. Ingreso del número de la unidad.</li> <li>2. Calificar con Estrellas.</li> <li>3. Comentar.</li> </ol>	<p>La unidad admite sólo 6 dígitos. Las estrellas no tienen una calificación media. Al comentar se admiten todo tipo de caracteres.</p>	Ok, logra realizar todas las funciones con éxito.
PRF04	Ruters	<p>Ingreso al módulo Ruters. Aparece la marca del usuario.</p> <ol style="list-style-type: none"> <li>3. Detectar movimiento de las unidades.</li> </ol>	El dispositivo obtiene las coordenadas del usuario y detecta con facilidad.	OK, cumple con sus funciones.
PRF05	Firestore-BD	Almacenar la información de cada módulo.	Es una base de datos no relacional.	OK, logra almacenar los datos enviados desde la aplicación.

Nota: Detalle de la integración de todos los módulos en la aplicación



A continuación, se detallan los resultados obtenidos en cada prueba que se realizó a los distintos módulos.



En la figura No. 33 se puede observar la interacción de nuevos usuarios, y qué usuarios se logran retener. El gráfico constituye un estudio de cohortes, y muestra el porcentaje de usuarios que continúan usando la aplicación luego de una semana, dos, etcétera, hasta la quinta semana.

### Actividad de los usuarios en la aplicación.

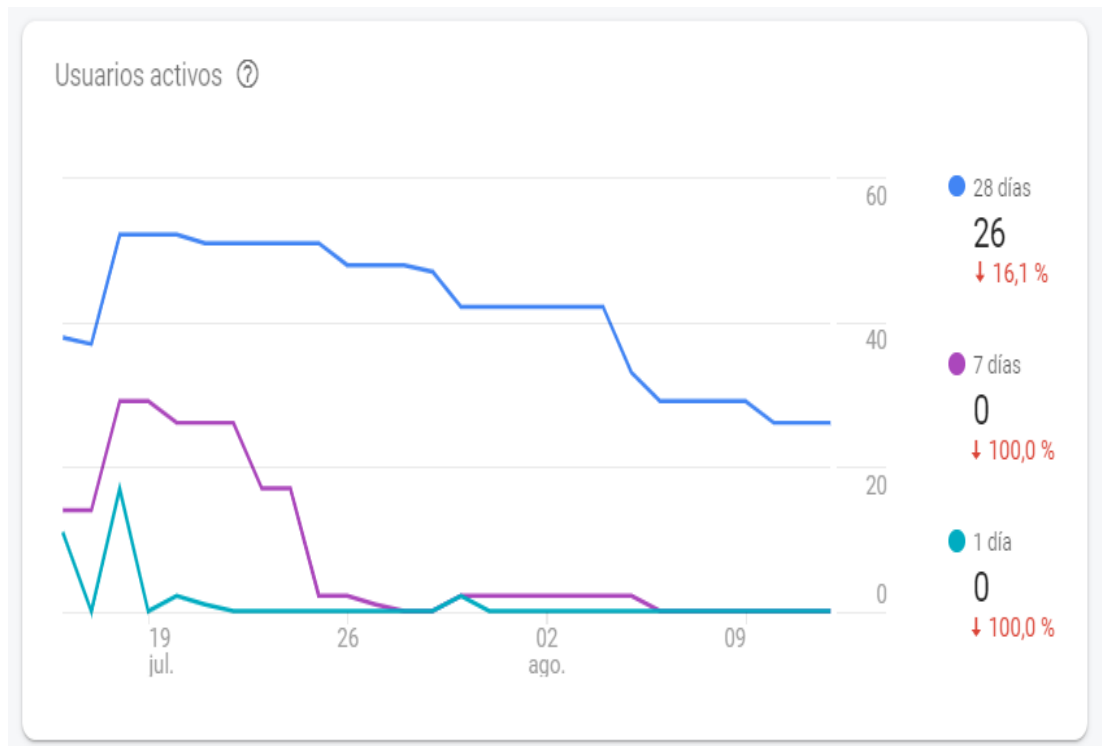


Figura 34 Usuarios activos en la aplicación.

Elaborado por los autores: Christian Arguello y Danny Quishpe a partir de FirebaseAnalytics

Adicionalmente, en la plataforma de Firebase se presenta una gráfica de la actividad que tienen los usuarios en la aplicación y se indica la interacción de los usuarios activos, como puede verse en la figura No. 34 y la figura No. 35.

El siguiente apartado es el análisis de actividad de interés de los usuarios, mostrando cuál es la sección que más se visita y qué tiempo se demora dentro de cada una de las pantallas.

Interacción de usuarios en la aplicación.



Figura 35. Uso de cada pantalla del prototipo por parte de los usuarios.

Elaborado por los autores: Christian Arguello y Danny Quishpe a partir de Firebase

### 4.3 Pruebas de Aceptación

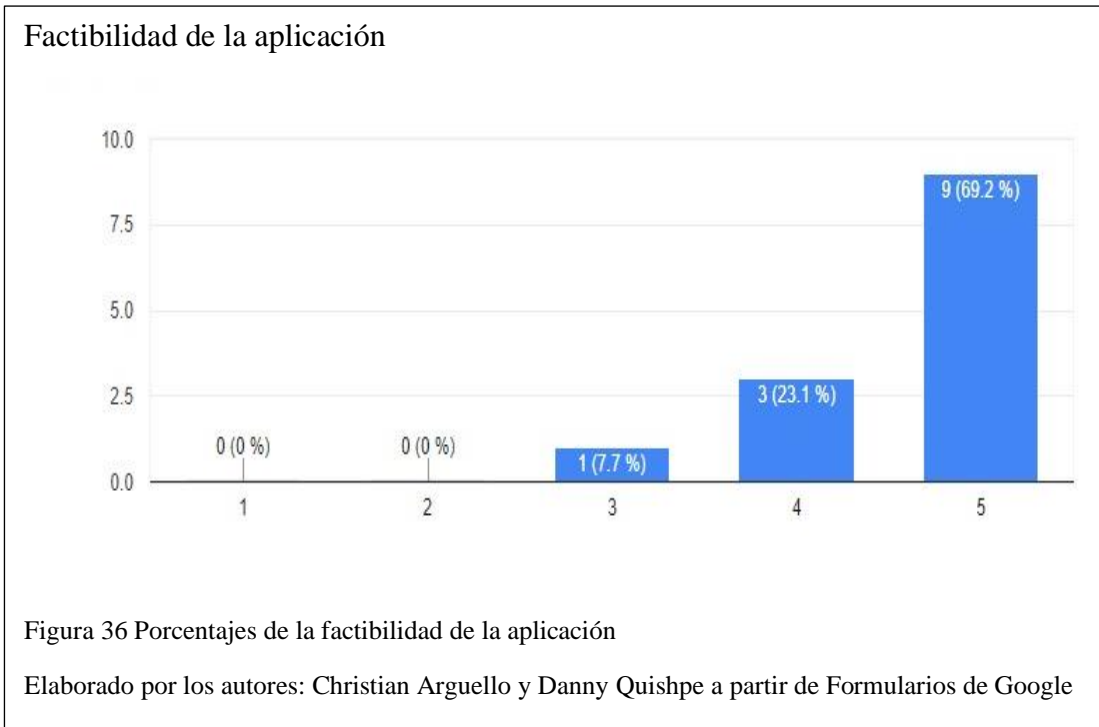
Para las pruebas de aceptación se solicitaron 3 sujetos de prueba con dispositivos Android. Los usuarios probaron la aplicación y expresaron que el diseño de la misma no era adecuado y que la funcionalidad no era clara, con excepción del módulo de “Login”.

En virtud de estas opiniones, se planificó una encuesta para cambiar el diseño según el punto de vista de los usuarios. Los resultados fueron los siguientes:

#### 4.3.1 ¿Qué tan factible sería usar la aplicación BusterApp en el Distrito

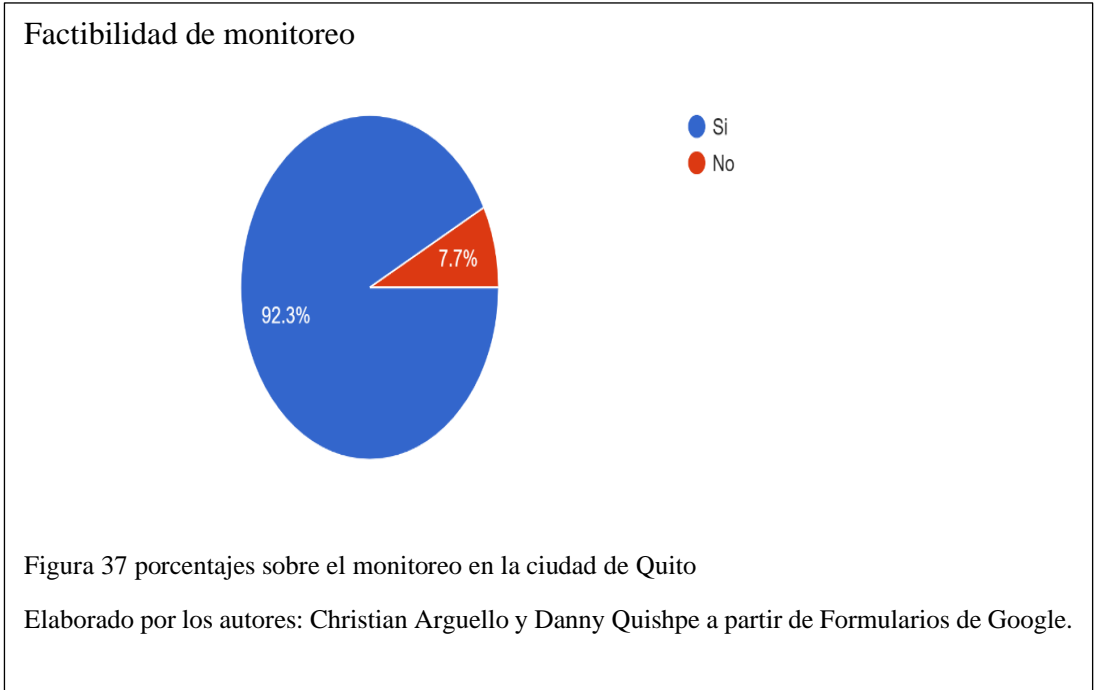
##### Metropolitano de Quito?

En la figura No. 36 se aprecia que 9 de los 13 encuestados, es decir 69,2%, opinan que la aplicación sería muy factible para el uso dentro del Distrito Metropolitano de Quito.



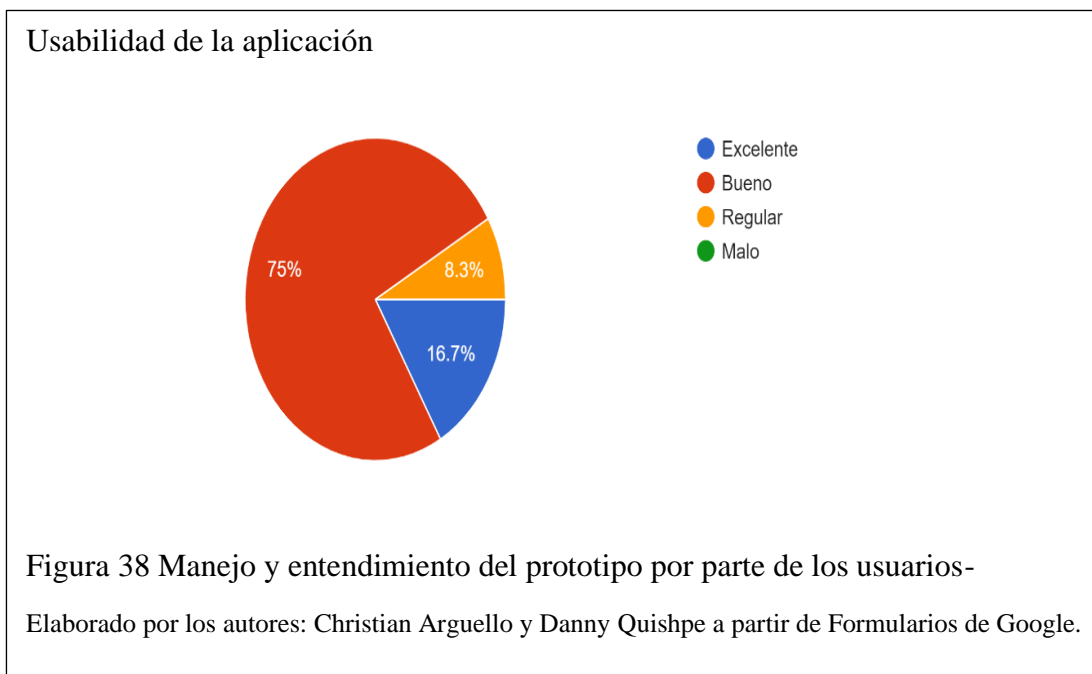
**4.3.2 ¿Cree que es necesario que se realice un monitoreo de las unidades de autobuses en la ciudad de Quito?**

Como se puede apreciar en la figura No. 37, el 92.3% piensa que sí se necesita un monitoreo de autobuses controlado dentro de la ciudad para mejorar la calidad de servicio.



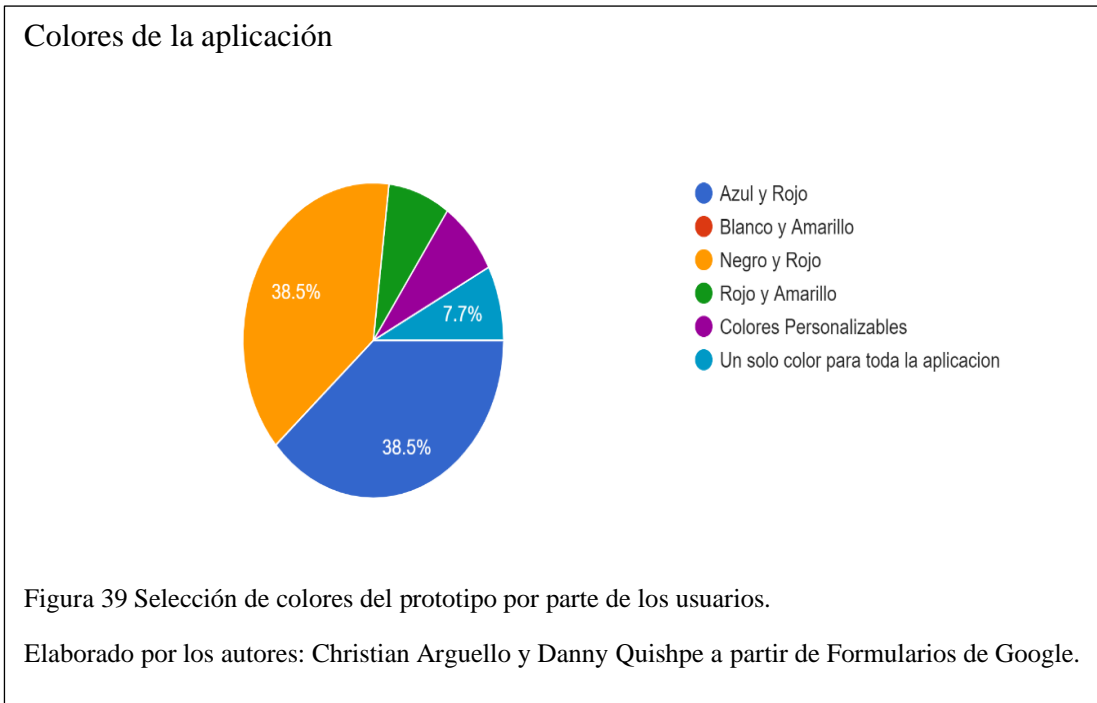
### 4.3.3 ¿Según su experiencia que tan fácil fue usar la aplicación?

El 75% de los encuestados opina que su uso fue bueno, el 16.7% excelente y el 8.3% regular. La aplicación en su mayor parte es entendible para los usuarios en cuanto su manejo como se presenta en la figura No. 38.



### 4.3.4 ¿Qué colores le gustaría en la aplicación?

Los encuestados se igualaron en su preferencia por los colores negro con rojo y azul con rojo, obteniendo ambas opciones un 38.5%, como se muestra en la figura No. 39. Ante este empate, el equipo de desarrollo optó por la combinación azul-rojo, que son los colores de la ciudad.



#### 4.3.5 ¿Qué icono le gusta más?

A los encuestados se les presentó las opciones que se muestran en la figura No. 40 para establecer un ícono para la aplicación, ya que se está realizando un cambio de diseño y en la primera versión la aplicación no contaba con dicho ícono.

### Opciones de los íconos

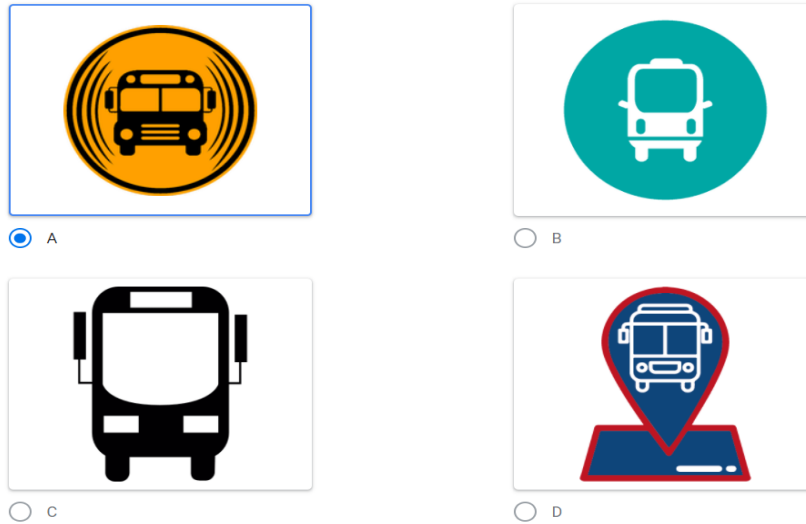


Figura 40 Íconos llamativo por los usuarios.

Elaborado por los autores: Christian Arguello y Danny Quishpe a partir de Formularios de Google.

El 76.9% de los usuarios encuestados decidieron que el icono más llamativo para la aplicación era la opción “D” y en segundo lugar la opción “A” con un 15.4% a favor. Estos resultados se pueden apreciar en la figura No. 41.

### Selección de ícono

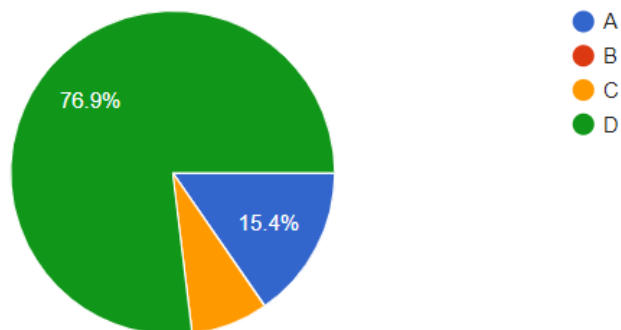
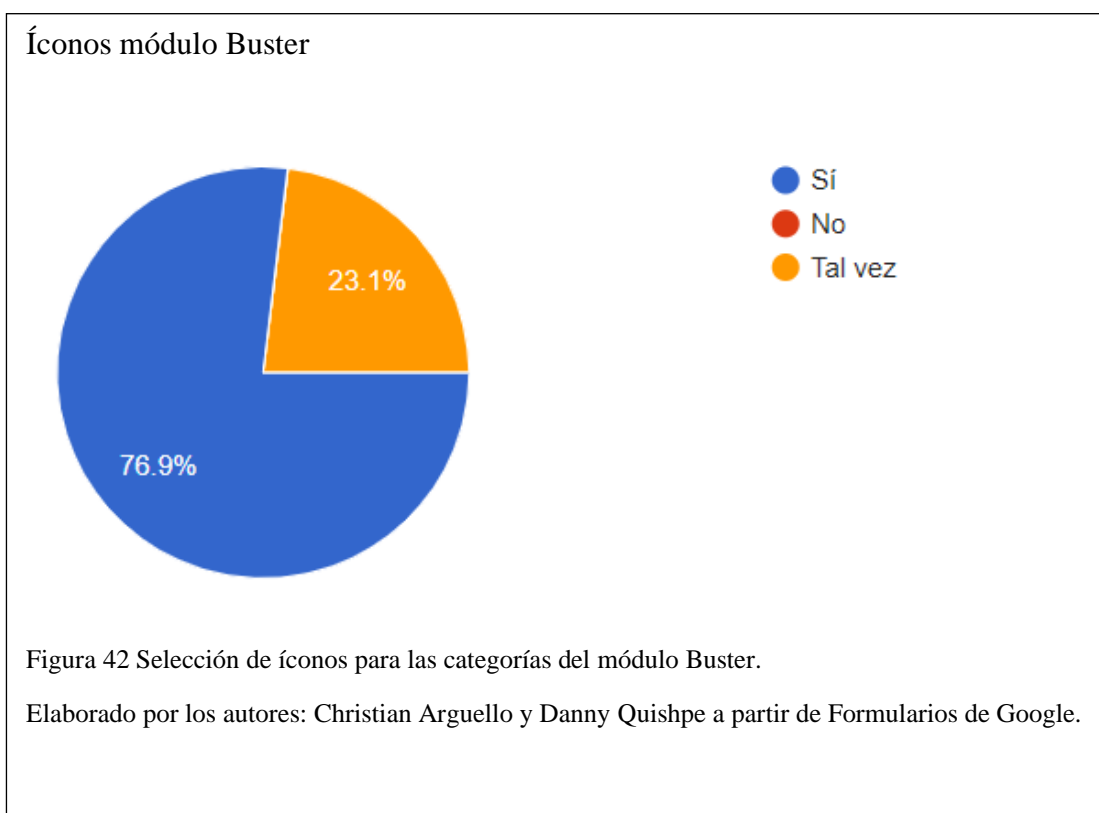


Figura 41 Porcentajes del ícono mas llamativo por parte de los encuestados.

Elaborado por los autores: Christian Arguello y Danny Quishpe a partir de Formularios de Google.

#### 4.3.6 ¿Cree que los íconos utilizados en esta aplicación son los adecuados?

La pregunta se refiere a los íconos usados para representar las diferentes categorías a ser calificadas por el usuario. Como se puede observar en la figura No. 42, el 23.1% de los encuestados no entiende el significado de dichos íconos. Por lo tanto, se cambió el diseño de la mayoría de ellos para que la experiencia sea más amigable con el usuario.



#### 4.3.7 ¿El lenguaje utilizado en la aplicación es el idóneo o entendible en cuanto a los mensajes de ayuda al momento de calificar?

Dentro del módulo Buster existen botones de ayuda que explican al usuario las diferentes categorías a calificar. El 15.4% de los encuestados no llegó a entender el mensaje que se estaba transmitiendo, como se muestra en la figura No. 43.



Texto entendible

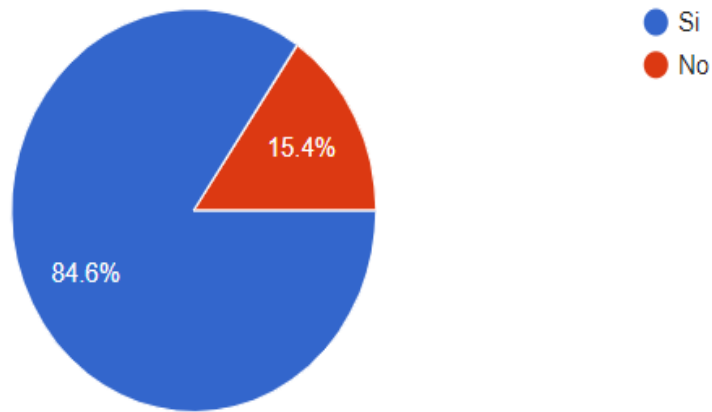


Figura 43 Porcentajes sobre el entendimiento del lenguaje implementado

Elaborado por los autores: Christian Arguello y Danny Quishpe a partir de Formularios de Google.

#### 4.3.8 ¿Qué elementos cambiaría en el diseño de la aplicación?

Cambios de diseño según el usuario

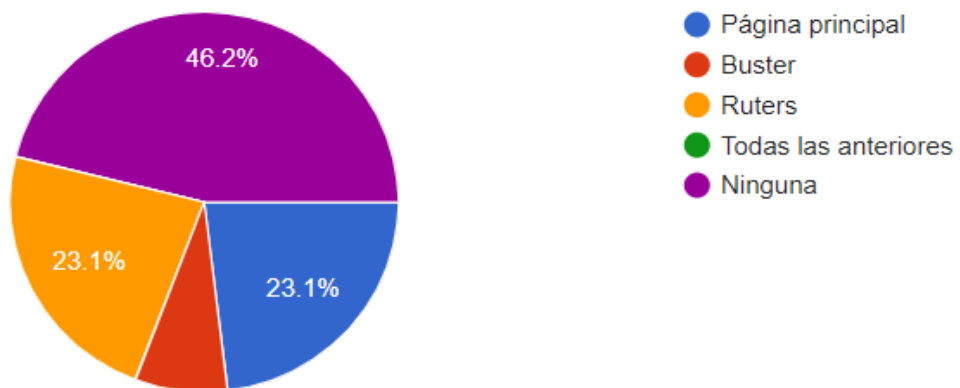
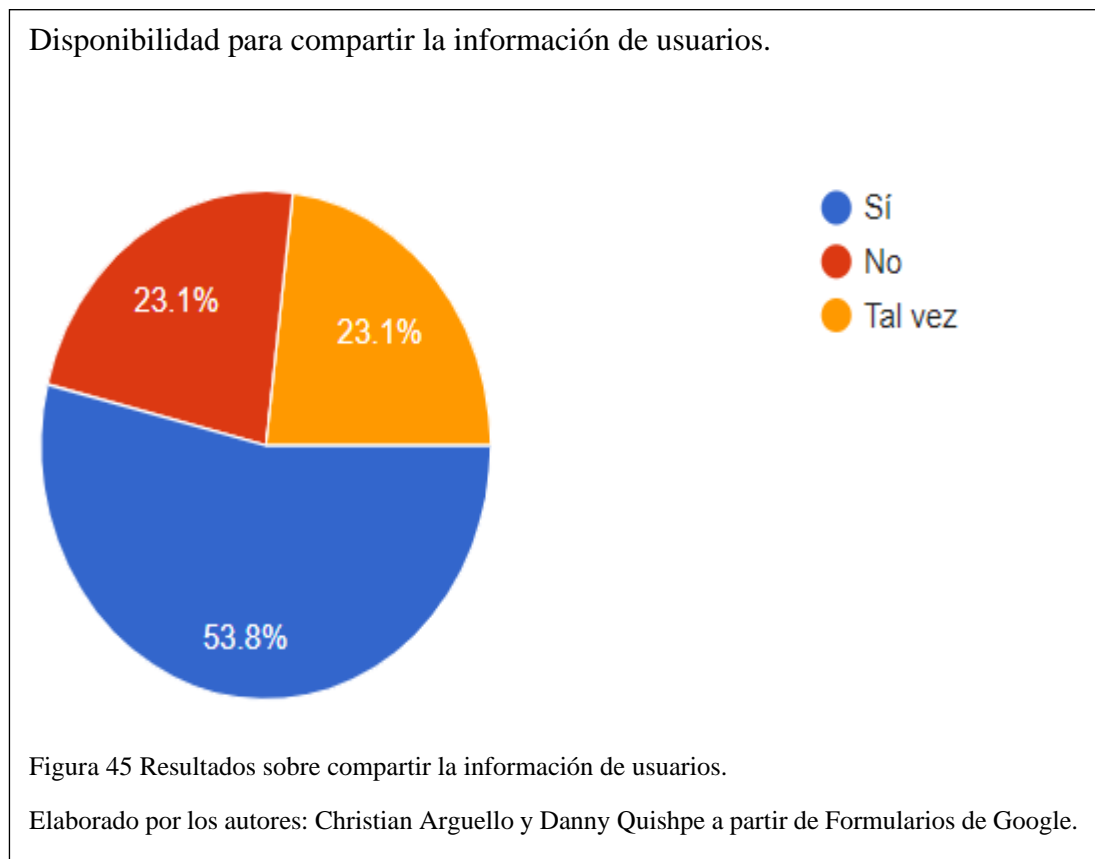


Figura 44 Resultados sobre cambiar el diseño de los módulos según el usuario

Elaborado por los autores: Christian Arguello y Danny Quishpe a partir de Formularios de Google.

En la Figura No 44 se puede ver que el 46.2% de los 13 sujetos opinaron que no necesitaba ningún cambio, pero dos partes de 23.1% nos mencionan que hay que realizar cambios en la página principal y el módulo Rutgers y el 7.6 % piensa que se debe de mejorar el diseño del módulo Buster.

#### 4.3.9 ¿Usted estaría dispuesto a compartir su información personal al momento de registrarse dentro de la aplicación?



A los encuestados se les preguntó si desearían compartían su información personal al momento de registrarse a la aplicación. El 53.8% están dispuestos, el 23.1% tal vez y el otro 23.1% definitivamente no, como se presenta en la figura No. 45.

#### 4.3.10 Resultado del diseño basados en los resultados de la encuesta

Con los resultados obtenidos de la encuesta la aplicación pasó por una fase de mejora la cual se detalla a continuación. Mostramos en esta sección el prototipo final, con mejoras como textos de ayuda y una interfaz más atractiva para el usuario.

## Pantalla de inicio v2

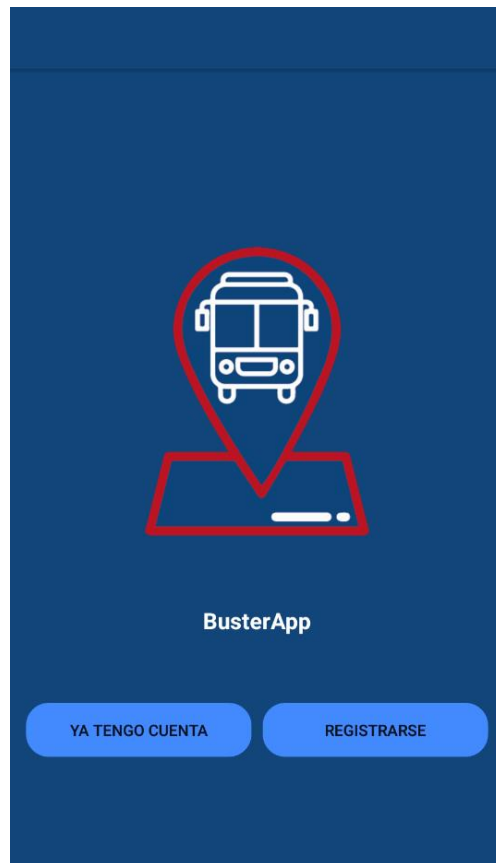


Figura 46 Mejora de la pantalla de inicio.

Elaborado por los autores: Christian Arguello, Danny Quishpe

En la figura No. 46 y figura No. 47, se puede observar la pantalla de inicio o principal, donde se implementan dos botones, que redireccionan al usuario al inicio de sesión. En la versión anterior no se contaban con dichos botones.

## Tres opciones de inicio de sesión

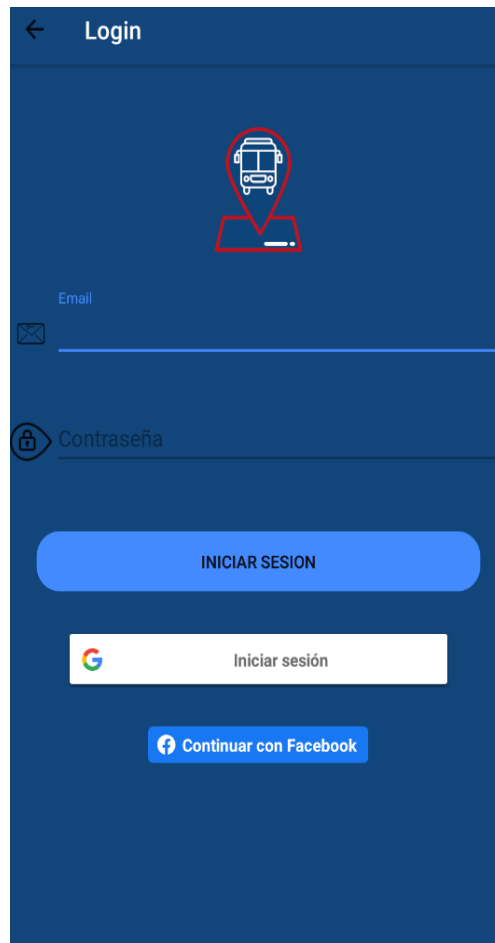
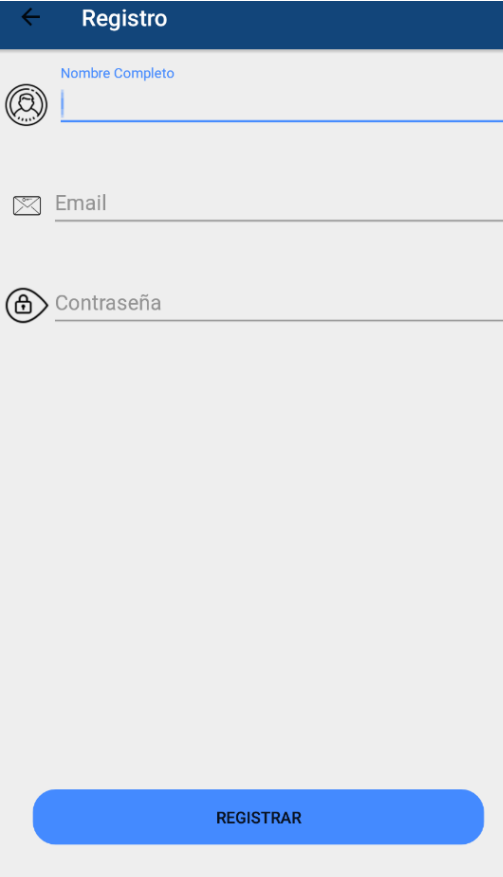


Figura 47 Resultado final del módulo Login

Elaborado por los autores: Christian Arguello & Danny Quishpe

En el caso de tener una cuenta, el usuario podrá seleccionar su opción de Login preferida. Caso contrario, tendrá que registrarse para poder usar la aplicación.

## Registro de usuarios



The image shows a mobile application screen for user registration. At the top, there is a dark blue header with a back arrow and the word "Registro". Below the header, there are three input fields: "Nombre Completo" with a person icon, "Email" with an envelope icon, and "Contraseña" with a lock icon. At the bottom of the form is a large blue button labeled "REGISTRAR".

Figura 48 Resultado de la pantalla de registro de usuarios por medio de correo electrónico

Elaborado por los autores : Christian Arguello, Danny Quishpe

En la figura No. 48 se detalla el registro del usuario, en este caso se implementaron íconos haciendo referencia a que es lo que se debe introducir, al igual que textos que fueron sugerencias de la encuesta para evitar que introduzcan el correo, nombre de usuario o la contraseña donde no corresponde.

## Pantalla Principal

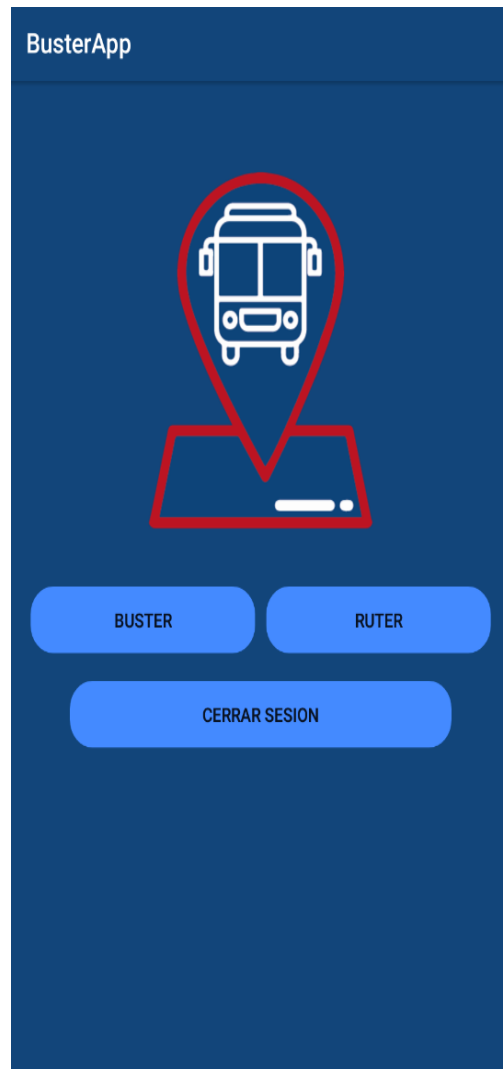


Figura 49 Prototipo final, pantalla principal con sus módulos implementados

Elaborado por: Christian Arguello, Danny Quishpe

Una vez ya iniciada la sesión tendrá tres botones a elegir, como se aprecia en la figura No. 49: podrá ingresar en el módulo de Buster, en el de Ruter o cerrar la sesión.

## Módulo Buster v2

Unidad de Transporte

Unidad de transporte ?

Seguridad ? ★ ★ ★ ★ ★

Estado de la Unidad ? ★ ★ ★ ★ ★

Confort ? ★ ★ ★ ★ ★

Conductor ? ★ ★ ★ ★ ★

Tiempo ? ★ ★ ★ ★ ★

ENVIAR CALIFICACIÓN

Comentario

Ingrese su comentario

OMITIR ENVIAR

Figura 50 Resultado final del módulo Buster con las sugerencias de los usuarios encuestados.

Elaborado por los autores: Christian Arguello, Danny Quishpe

Como se presenta en la figura No. 50, el módulo Buster tuvo mejoras en la adición de íconos y ayudas con mensajes informativos. Se muestra además la pantalla siguiente, que permite al usuario ingresar un comentario opcional.

## Ayudas para el usuario

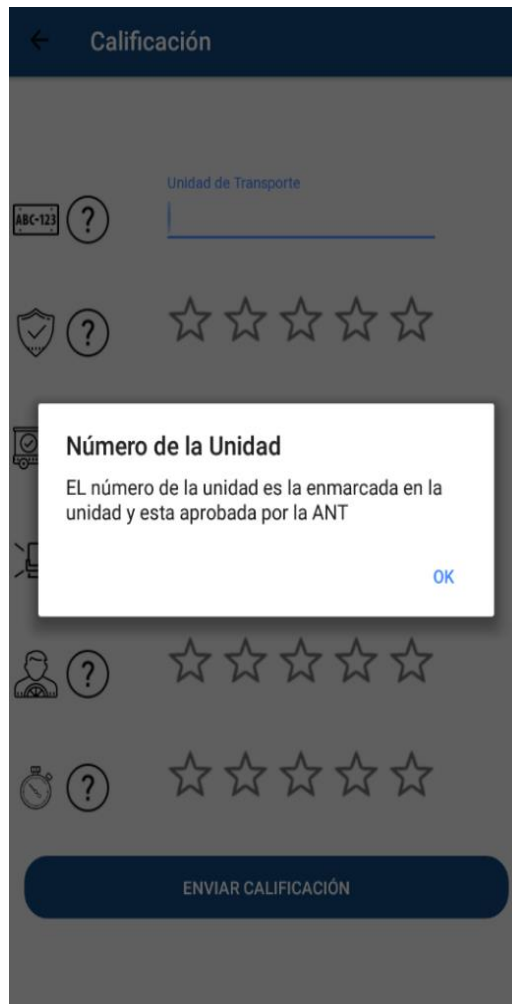


Figura 51 Mensajes de ayuda para los usuarios.

Elaborado por los autores: Christian Arguello, Danny Quishpe

En la figura No. 51, se tiene el ejemplo de los mensajes de ayuda que se implementaron, y que aparecen al presionar en el ícono de interrogación.



## Módulo Ruter v2

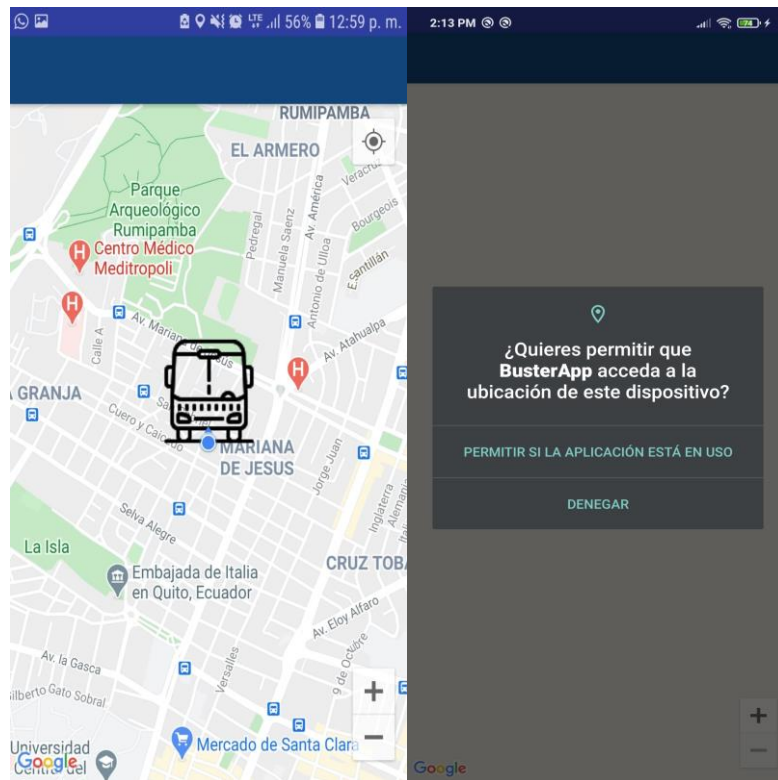


Figura 52 Resultado final del módulo Ruters

Elaborado por los autores: Christian Arguello, Danny Quishpe

La figura No. 52 muestra que el módulo Ruter presenta también ciertos cambios, como el ícono y la información del usuario, que ya no se encuentra obstruyendo parte del mapa. Los mensajes de permisos son los que están por defecto en cada dispositivo.

## CONCLUSIONES

Se cumplió el objetivo general que era crear el prototipo para el monitoreo del transporte público de la ciudad de Quito, y el uso de la metodología ágil SCRUM fue esencial para el desarrollo de la aplicación.

El arquetipo fue fundamental para la implementación, debido a que permitió al dueño del software tener un enfoque visual y a los desarrolladores tener una arquitectura base para su implementación.

El modelo MVC y la Arquitectura Cliente-Servidor permitió identificar los requerimientos y necesidades del prototipo.

Android Studio cumplió con los requerimientos para el desarrollo de la aplicación por tener la ventaja de la conectividad con las API's de Google, lo cual fue esencial para los módulos desarrollados.

Firestore logró ser de mucha utilidad en la fase de autenticación de usuarios con los tres servicios que se desarrollaron dentro de la aplicación, sin embargo, al ser una base de datos no relacional las entidades no están vinculadas.

La API "SDK de mapas para Android" de Google permitió la administración del mapa para el tratamiento de las coordenadas y acceso a la latitud y longitud, mostrando a cada usuario en tiempo real.

Se realizó una prueba de aceptación con una muestra de 13 usuarios en dos fases. Los resultados de la primera fase nos permitieron realizar cambios de funcionamiento y diseño. En la segunda fase, el 69.2% de los encuestados opinó que es factible el uso de la aplicación dentro de la ciudad de Quito.

## **RECOMENDACIONES**

Implementar un módulo dentro de una página web para manejar la información proporcionada por los usuarios, misma información que será utilizada por los administradores para generar reportes de cada unidad.

Para trabajos futuros se recomienda hacer una aplicación híbrida, es decir, que corra tanto en el sistema operativo Android como en iOS, para acoger a la mayor cantidad de usuarios.

Se recomienda hacer uso de los usuarios disponibles en el apartado de Rutgers para obtener otro tipo de información, como la unidad en la que se encuentra o las valoraciones que haya calificado dicho usuario.

El comentario opcional, que el usuario puede subir de manera complementaria a sus calificaciones, podría ser acompañado de una foto, igualmente opcional.

Se recomienda trabajar con cooperativas de transporte para obtener todas las unidades disponibles y que el usuario no requiera llenar el campo “xxxx”, sólo elegir la unidad de entre una lista que debería ser desplegada por la aplicación. Esto evitaría errores de ingreso.

## GLOSARIO

**Busters:** Usuario que calificará la unidad de transporte dentro de la aplicación.

**Ruters:** Usuario que visualizará las unidades de transporte cercanas en el mapa de la aplicación.

**BPMN:** Notación legible y entendible, cuyo objetivo es dar a conocer el flujo del proceso/actividad.

**Sprint:** Secuencia de actividades a realizar determinadas por un tiempo.

**JSON:** Es un formato ligero de intercambio de datos.

**API:** Es un conjunto de funciones y procedimientos que cumplen una o muchas funciones con el fin de ser utilizadas por otro software.

**SDK:** Es un conjunto de herramientas de desarrollo de software.

**Scrum:** Es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar.

**APK:** Es un paquete para el sistema operativo Android.

**KEY:** Es un identificador único utilizado para autenticar a un usuario.

## REFERENCIAS

### Artículos académicos

- Álvarez Pérez, J. A. (29 de Septiembre de 2017). *ENTORNO DE PROGRAMACIÓN INTENCIONAL BASADO EN XML*. Obtenido de [http://oa.upm.es/49793/1/PFC\\_JOSE\\_ANTONIO\\_ALVAREZ\\_PEREZ.pdf](http://oa.upm.es/49793/1/PFC_JOSE_ANTONIO_ALVAREZ_PEREZ.pdf)
- Amodeo, E. (2013). Principios de diseño de APIs REST. En E. Amodeo, *Principios de diseño de APIs REST* (pág. 1). Leanpub.
- Bascón Pantoja, Ernesto. 2004. “El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing”. *Acta Nova*.
- Cadavid, Andrés Navarro. 2013. “Revisión de metodologías ágiles para el desarrollo de software”. *Prospectiva*. doi: 10.15665/rp.v11i2.36.
- Díaz Ortiz, Jimmy Jhonon, y Mitcheli Anthony Romero Suarez. 2017. “Desarrollo E Implementación De Un Aplicativo Web, Utilizando La Metodología Scrum, Para Mejorar El Proceso De Atención Al Cliente En La Empresa Z Aditivos S.a”. 220.
- Group, Object Management. 2017. “OMG Unified Modeling Language TM ( OMG UML ), Version 2.5”. *InformatikSpektrum*. doi: 10.1007/s002870050092.
- Gunawan, Rohmat, y Alam Rahmatulloh. 2019. “JSON Web Token (JWT) untuk Authentication pada Interoperabilitas Arsitektur berbasis RESTful Web Service”. *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*. doi: 10.26418/jp.v5i1.27232.
- Khawas, Chunnu, y Pritam Shah. 2018. “Application of Firebase in Android App Development-A Study”. *International Journal of Computer Applications*. doi: 10.5120/ijca2018917200.
- Li, Wu Jeng, Chiaming Yen, You Sheng Lin, Shu Chu Tung, y Shih Miao Huang. 2018. “JustIoT Internet of Things based on the Firebase real-time database”. en *Proceedings - 2018 IEEE International Conference on Smart Manufacturing, Industrial and Logistics Engineering, SMILE 2018*.
- López S., Carlos Armando. 2009. “Cómo mantener el patrón modelo vista controlador en una aplicación orientada a la WEB”. *INVENTUM*. doi: 10.26620/uniminuto.inventum.4.7.2009.72-78.
- Marini, Ing Emiliano. 2012. “El Modelo Cliente / Servidor”. 1–11.

Srivastava, Apoorva, Sukriti Bhardwaj, y Shipra Saraswat. 2017. “SCRUM model for agile methodology”. en *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2017*.

Vasquez, P., Giandini, R., & Bazán, P. (2010). *Lenguajes Notacionales para Modelado de Procesos: un análisis comparativo*. Obtenido de [http://sedici.unlp.edu.ar/bitstream/handle/10915/19526/Documento\\_completo.pdf?sequence=1&isAllowed=y](http://sedici.unlp.edu.ar/bitstream/handle/10915/19526/Documento_completo.pdf?sequence=1&isAllowed=y)

### **Artículos de revistas y periódicos electrónicos**

Anón. 2012. “Patrón Modelo-Vista-Controlador.” *Revista Telemática*.

Quito Informa. (19 de Septiembre de 2017). *Quito Informa*. Obtenido de Municipio

### **Libro versión electrónica**

Bahit, Eugenia. 2011. *POO y MVC en PHP*.

### **Páginas web**

Android Studio. (09 de Julio de 2020). *Developers*. Obtenido de Guia de Usuario: <https://developer.android.com/studio/intro>

Bizagi Ltd. (19 de Julio de 2020). *Guia de Usuario de bizagi process*. Obtenido de Bienvenido a la documentación de Bizagi Modeler 3.7 y Modeler Services: <http://help.bizagi.com/process-modeler/es/>

Facebook. (22 de Septiembre de 2020). *Facebook for developer*. Obtenido de [https://developers.facebook.com/?no\\_redirect=1](https://developers.facebook.com/?no_redirect=1)

Google. (22 de Septiembre de 2020). *Google Cloud*. Obtenido de <https://cloud.google.com/maps-platform>

Gutiérrez Díaz, A. (18 de Febrero de 2020). *Centro Cultural Itaca*. Obtenido de <https://www.aiu.edu/cursos/base%20de%20datos/pdf%20leccion%201/lecci%C3%B3n%201.pdf>

Milenium. (18 de Febrero de 2020). *Milenium*. Obtenido de Temas: <https://www.informaticamilenium.com.mx/es/temas/que-es-una-app.html>

Nieto Gonzales, A. (9 de Febrero de 2020). *Xatakandroid*. Obtenido de ¿Qué es Android?: <https://www.xatakandroid.com/sistema-operativo/que-es-android>

Oracle. (5 de Febrero de 2021). *Java*. Obtenido de Ayuda:

[https://www.java.com/es/download/help/whatis\\_java.html](https://www.java.com/es/download/help/whatis_java.html)

### **Tesis y Trabajos de Grado**

Mascheroni, Maximiliano Irrazábal, Emanuel. 2016. “Framework para la creación y ejecución de pruebas automatizadas sobre servicios REST”. *Universidad de la Cuenca del Plata*.

Núñez Manzano, S. A., & Pino Riera, N. E. (2019). *Dspace PUCE*. Obtenido de Evaluacion: <http://repositorio.puce.edu.ec/handle/22000/17069>

## **Anexos**

Los anexos (que pueden ser encontrados en el CD) son:

- Manual de Usuario
- Manual de Desarrollador