



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN MÓVIL JALÓN PARA FACILITAR EL
USO COMPARTIDO DE VEHÍCULO ENTRE CIUDADES EN EL TERRITORIO NACIONAL**

Erick Fernando Sac Nimatuj
Asesorado por el Ing. Ángel Eduardo Sic Morales

Guatemala, enero de 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN MÓVIL JALÓN PARA FACILITAR EL
USO COMPARTIDO DE VEHÍCULO ENTRE CIUDADES EN EL TERRITORIO NACIONAL**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

ERICK FERNANDO SAC NIMATUJ

ASESORADO POR EL ING. ÁNGEL EDUARDO SIC MORALES

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, ENERO DE 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Ing. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton De León Bran
VOCAL IV	Br. Christian Moisés de la Cruz Leal
VOCAL V	Br. Kevin Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Marlon Francisco Orellana López
EXAMINADOR	Ing. José Manuel Ruiz Juárez
EXAMINADOR	Ing. Marlon Antonio Pérez Turk
SECRETARIO	Inga. Lesbia Magalí Herrera López

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN MÓVIL JALÓN PARA FACILITAR EL USO COMPARTIDO DE VEHÍCULO ENTRE CIUDADES EN EL TERRITORIO NACIONAL

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 09 de agosto de 2019.



Erick Fernando Sac Nimatuj

Guatemala, 29 de Octubre de 2019

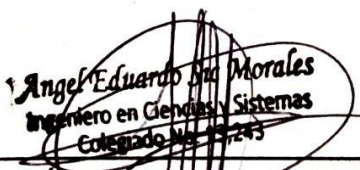
Señor
Ing. Carlos Azurdia
Facultad de Ingeniería
Universidad de San Carlos de Guatemala
Guatemala, Ciudad

Estimado Ing. Azurdia:

Me complace saludarle, el motivo de la presente es para informarle que como asesor del estudiante Erick Fernando Sac Nimatuj con CUI 1910935640901 y carnet 200730410 he procedido a supervisar y asesor el trabajo de graduación titulado **“Diseño e implementación de la aplicación móvil “Jalón” para facilitar el uso compartido de vehículo entre ciudades en el territorio nacional.”** y que de acuerdo a mi criterio este se encuentra concluido y revisado en su totalidad.

Sin otro particular me suscribo a usted,

Atentamente,


Angel Eduardo Sac Morales
Ingeniero en Ciencias y Sistemas
Colegiado No. 13243

Angel Sac Morales
Ingeniero en Ciencias y Sistemas
Colegiado No. 13243



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 15 de noviembre de 2019

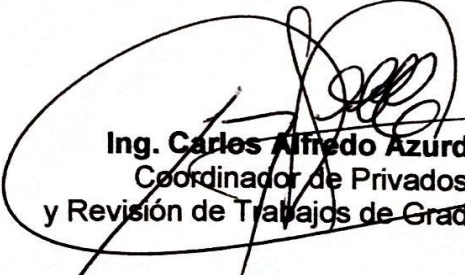
Ingeniero
Carlos Gustavo Alonzo
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **ERICK FERNANDO SAC NIMATUJ** con carné **200730410** y CUI **1910 93564 0901** titulado **“DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN MÓVIL –JALÓN- PARA FACILITAR EL USO COMPARTIDO DE VEHÍCULO ENTRE CIUDADES EN EL TERRITORIO NACIONAL”** y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



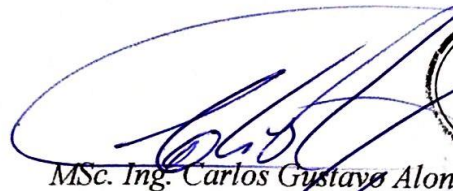
UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA




FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN
CIENCIAS Y SISTEMAS
TEL: 24188000 Ext. 1534

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación, **“DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN MÓVIL JALÓN PARA FACILITAR EL USO COMPARTIDO DE VEHÍCULO ENTRE CIUDADES EN EL TERRITORIO NACIONAL”** realizado por el estudiante, ERICK FERNANDO SAC NIMATUJ, aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”


MSc. Ing. Carlos Gustavo Alonzo
Director
Escuela de Ingeniería en Ciencias y Sistemas



Guatemala, 06 de febrero de 2020



DTG. 043.2020

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN MÓVIL JALÓN PARA FACILITAR EL USO COMPARTIDO DE VEHÍCULO ENTRE CIUDADES EN EL TERRITORIO NACIONAL**, presentado por el estudiante universitario: **Erick Fernando Sac Nimatuj**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

Inga. Anabela Cordova Estrada
Decana

Guatemala, enero de 2020

/gdech



ACTO QUE DEDICO A:

Mi padre

José Sac y Judith Nimatuj, por ser parte central y esencial en mi vida, fuentes de consejo y sabiduría y de un infinito amor.

Mis hermanos

José Alexander, William Francisco y Josué Oscar Enrique Sac Nimatuj, por compartir buenos y malos momentos.

Mi familia

Del lado paterno y materno, por estar siempre allí siempre en los buenos y malos momentos.

AGRADECIMIENTOS A:

- Mis padres** Por su eterno amor, por todo su apoyo incondicional, por ser una influencia importante en mi carrera, ser ejemplo y fuente de inspiración en los momentos difíciles.
- Mis amigos de la Facultad de ingeniería** Por ser grandes compañeros y amigos al inicio de la carrera, estando en las buenas y en las malas.
- Mis amigos de Sistemas** Por su amistad, compañerismo y apoyo a lo largo de la carrera, por compartir tristezas y alegrías en cada curso que llevamos.
- Mis amigos de francés y alemán** Por su apoyo y amistad demostrados a lo largo de mi carrera, por los grandes momentos que compartimos.
- Mis amigos de trabajo** Por su constante apoyo y motivación a seguir adelante.
- Ing. Ángel Sic** Por haber confiado en mí, por brindarme su apoyo como asesor y por ser también, un buen amigo.

**Universidad de San
Carlos de Guatemala**

Por haberme abierto las puertas y darme la oportunidad de crecer personalmente y superarme.

Facultad de Ingeniería

Por brindarme el conocimiento necesario para ser un profesional competente y de éxito.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
GLOSARIO	IX
RESUMEN.....	XIII
OBJETIVOS.....	XV
INTRODUCCIÓN	XVII
1. ESTUDIO DE LA TECNOLOGÍA Y SU IMPACTO EN GUATEMALA	1
1.1. Identificación de la teoría que soporta la investigación	1
1.1.1. Modelo de aceptación tecnológica.....	1
1.1.2. Ludificación.....	3
1.2. Teoría y la relación con la tecnología escogida.....	7
2. IDENTIFICACIÓN DE PROBLEMA Y SOLUCIÓN QUE LA APLICACIÓN REALIZARÁ.....	11
2.1. Antecedentes.....	11
2.1.1. Transporte extraurbano en Guatemala	12
2.1.2. Contaminación por CO2	13
2.1.3. Aplicaciones móviles	14
2.2. Benchmarking.....	15
2.2.1. Mercado nacional	15
2.2.1.1. TraeGuate.....	15
2.2.1.2. Guate jalón Facebook.....	16
2.2.2. Mercado internacional	18
2.3. Mercado objetivo	19

3.	ANÁLISIS Y DISEÑO DE LA APLICACIÓN.....	21
3.1.	Arquitectura de la solución	21
3.1.1.	API	21
3.1.2.	REST/RESTFul	21
3.1.3.	JSON Web Token	22
3.1.4.	Aplicaciones híbridas	23
3.1.5.	Arquitectura MEAN.....	23
3.1.6.	Descripción de la arquitectura.....	24
3.2.	Casos de uso	26
3.2.1.	Crear usuario.....	26
3.2.2.	Ingresar a la aplicación	27
3.2.3.	Crear viaje	28
3.2.4.	Buscar viaje.....	29
3.2.5.	Responder solicitud de viaje de usuario	30
3.2.6.	Solicitar unirse a un viaje.....	31
3.2.7.	Puntuar viaje	32
3.2.8.	Crear mensaje.....	33
3.2.9.	Responder mensaje	34
3.2.10.	Dar medallas	35
3.3.	Diagramas de secuencias	36
3.4.	Diseño de base de datos.....	39
3.4.1.	Modelo de datos.....	39
3.4.2.	Diccionario de datos	40
3.5.	Prototipo.....	43
3.5.1.	Login	43
3.5.2.	Inicio	45
3.5.3.	Búsqueda de viajes	46
3.5.4.	Creación de viaje.....	47
3.5.5.	Información de un viaje	48

3.5.6.	Finalización de viaje	49
3.5.7.	Chat	50
3.5.8.	Actualizar perfil	51
3.6.	Mapa de navegación de la aplicación móvil	52
4.	ANÁLISIS Y DISEÑO DE LA APLICACIÓN	53
4.1.	Herramientas	53
4.1.1.	Hardware	53
4.1.1.1.	Heroku	54
4.1.1.2.	mLab.....	54
4.1.2.	Software	55
4.1.2.1.	GIT.....	55
4.1.2.2.	Github	56
4.1.2.3.	Visual Studio Code	56
4.1.2.4.	Robo 3T.....	56
4.1.2.5.	Chrome developer <i>Tools</i>	57
4.2.	Requerimientos de software	57
4.2.1.	Javascript.....	58
4.2.2.	NodeJS.....	58
4.2.3.	ExpressJS.....	58
4.2.4.	NPM.....	59
4.2.5.	MongoDB.....	59
4.2.6.	Mongoose.....	59
4.2.7.	Angular	59
4.2.8.	Ionic Framework	60
4.2.8.1.	HTML5.....	60
4.2.8.2.	SASS	60
4.2.8.3.	Cordova	61
4.3.	Manual de la aplicación	61

4.4.	Manual de publicación de servicios en Heroku	76
4.5.	Consideraciones de implementación.....	82
CONCLUSIONES.....		85
RECOMENDACIONES		87
BIBLIOGRAFÍA.....		89
ANEXO		91

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Modelo conceptual para la aceptación de una tecnología.....	2
2.	TAM original propuesto por Fred Davis	2
3.	Teoría de la efectividad de la gamificación.....	5
4.	Pantallas de la aplicación móvil de TraeGuate.GT	16
5.	Página inicial del grupo Guate jalón	17
6.	Pantallas de creación de viajes y de solicitudes de viaje	18
7.	Perfil de usuarios de la aplicación móvil BlaBlaCar	19
8.	Arquitectura de la solución	25
9.	Diagrama, login	36
10.	Diagrama, creación de viajes	36
11.	Diagrama, búsqueda de viajes	37
12.	Diagrama, finalizar viaje	37
13.	Diagrama, chat	38
14.	Diagrama, medallas.....	38
15.	Diagrama de modelo de datos	39
16.	Prototipo, pantalla de login	44
17.	Prototipo, pantalla de inicio	45
18.	Prototipo, pantallas de búsqueda y resultados	46
19.	Prototipo, pantallas de creación de viaje	47
20.	Prototipo, pantalla información del viaje	48
21.	Prototipo, pantalla de finalización de viaje.....	49
22.	Pantalla prototipo chat.....	50
23.	Pantalla, prototipo actualización de perfil	51

24.	Esquema de navegación de la app.....	52
25.	Pantalla de login.....	62
26.	Pantalla de registro	63
27.	Pantalla de inicio	64
28.	Pantalla, viajes del usuario	65
29.	Pantalla de información de viaje y finalización.....	66
30.	Pantalla de búsqueda	67
31.	Pantallas de resultados y solicitar viaje.....	68
32.	Pantalla de perfil de usuario.....	69
33.	Pantalla de creación de viaje, paso de lugar de partida y llegada.....	70
34.	Pantalla de creación de viaje, paso de ingresar paradas.....	71
35.	Pantalla de creación de viaje, paso de ingresar fecha	72
36.	Pantalla de creación de viaje, paso final de pasajeros.....	73
37.	Pantallas de chat.....	74
38.	Pantalla, actualizar perfil	75
39.	Página de login de Heroku	76
40.	Página de login de mLab	77
41.	Crear un cluster en mLab.....	78
42.	Cómo conectarse a la aplicación de servicios	79
43.	Cadena de conexión de la base de datos	80
44.	Crear app nueva en Heroku	81
45.	Página de <i>deploy</i> de una app en Heroku	81
46.	Variables de configuración	82

TABLAS

I.	Protocolos HTTP.....	22
II.	Usuario	40

III.	Comentario.....	40
IV.	Viaje	41
V.	Mensaje.....	41
VI.	Notificación.....	42
VII.	Jalón	42
VIII.	Imagen	42
IX.	Medalla	43
X.	Puntuación	43
XI.	Especificaciones de la computadora de desarrollo	53
XII.	Dispositivo Móvil	54
XIII.	Heroku y mLab.....	54
XIV.	Versiones de las herramientas de software	55

GLOSARIO

API	Interfaz de programación de aplicaciones, conjunto de servicios que pueden ser utilizados por otras aplicaciones o sistemas de software.
App	Es una aplicación de software comúnmente instalado en dispositivos móviles o <i>tablets</i> .
Android	Sistema operativo para dispositivos móviles propiedad de Google.
Arquitectura	Estructura lógica y física de los componentes de un sistema informático.
<i>Benchmarking</i>	Estudio que consiste en comparar las características y funcionalidades de un sistema con otro como referencia.
<i>Carpooling</i>	Vehículo compartido. Es la práctica que consiste en compartir un viaje en automóvil con otras personas.
CO2	Dióxido de carbono. Es un importante gas de efecto invernadero.
<i>Commit</i>	Se refiere a la idea de confirmar un conjunto de cambios provisionales de forma permanente.

CSS	Hojas de estilo en cascada utilizadas para el diseño de páginas web.
<i>Framework</i>	Es un entorno de trabajo para desarrollo que integra herramientas que facilitan el desarrollo de aplicaciones.
<i>Front-end</i>	Es la aplicación del lado del cliente o usuario.
HTTP	Protocolo de transferencia de hipertexto.
HTML	Lenguaje de marcas de hipertexto estándar en la elaboración de páginas web.
IOS	Sistema operativo para dispositivos móviles propiedad de Apple.
Jalón	Sinónimo de aventón, acción de llevar a un pasajero en un vehículo de manera gratuita.
JSON	Es un estándar en la notación de objetos JavaScript.
<i>Login</i>	Es el proceso con el cual se controla el acceso a un sistema informático mediante credenciales de usuario.
<i>Logout</i>	Es el proceso de salida de un sistema informático a través de la identificación personal.
MIT	Siglas del Instituto de Tecnología de Massachusetts.

NoSQL	Es un sistema de gestión de base de datos que difiere del sistema clásico de datos relacionales; también es conocido como 'no solo SQL'.
PaaS	Plataforma como servicio.
Package	Es un programa con un conjunto de funcionalidades específicas.
Plugins	Complemento de una aplicación que añade funcionalidades extras.
REST	Transferencia de estado representacional. Es un estilo de arquitectura de software.
SQL	Es un lenguaje de programación utilizado para administrar y recuperar información de sistemas de gestión de base de datos relacionales.
Token	Es un componente electrónico que se le proporciona a un usuario para la gestión de acceso a servicios de manera segura.
URL	Siglas de localizador uniforme de recursos. Es una secuencia de caracteres que se utilizan para nombrar y localizar recursos en internet.

RESUMEN

La aplicación Jalón es una solución de software que permite conectar a personas que desean compañía o ayuda económica para realizar un viaje entre ciudades y personas que buscan la manera de transportarse de una ciudad a otra de forma segura, cómoda y barata.

En el primer capítulo se aborda la teoría que se aplicó para desarrollar la aplicación que toma como base dos puntos importantes: el primero el modelo de aceptación tecnológica que explica cómo una nueva tecnología y sus características son recibidas y utilizadas por el usuario final. El segundo punto es la ludificación, es decir, cómo hacer que un usuario, a través de los elementos de un juego, utilice de una manera más entretenida y amena la aplicación o sistema.

En el segundo capítulo se plantea la problemática que se desea resolver en Guatemala y la situación actual. Uno de los problemas es la falta de transporte seguro y flexible para desplazarse entre municipios y el aumento de contaminación por CO₂ causante del calentamiento global. Así mismo, se plantea la solución por medio de medios informáticos tales como lo son las aplicaciones móviles.

En el tercer capítulo se presenta el análisis y el diseño de la aplicación que incluye la arquitectura, los casos de uso, los diagramas de secuencia, el diagrama de datos y un prototipo inicial. En el cuarto capítulo se describen las características finales de la aplicación así como las consideraciones de la implementación, que incluye manuales, herramientas utilizadas y versiones de software que fueron utilizadas para su desarrollo.

OBJETIVOS

General

Diseñar e implementar una aplicación móvil multiplataforma escalable capaz de facilitar e incentivar el uso compartido de vehículo entre ciudades y municipios de Guatemala de una manera interactiva y agradable a los usuarios.

Específicos

1. Incentivar el uso compartido de vehículo como medio seguro, rápido, económico y cómodo para realizar viajes.
2. Incentivar el turismo nacional al crear un ambiente agradable de movilidad para los viajeros.
3. Centralizar la información de viajes ofrecidos para una búsqueda fácil y cómoda para los usuarios que quieran optar por un trayecto compartido.
4. Reducir a largo plazo el tránsito vehicular en las carreteras nacionales y por ende las emisiones de dióxido de carbono.
5. Proveer la información necesaria para usarla como guía para el desarrollo de aplicaciones móviles parecidas.

INTRODUCCIÓN

Actualmente, los dispositivos móviles se han convertido en herramientas de uso diario y muchas veces necesarios para el vivir cotidiano de las personas; hacen que se requieran idear soluciones a problemas actuales del país a través del software.

La movilidad en Guatemala es actualmente uno de los problemas importantes tanto a nivel urbano como extraurbano debido principalmente a la falta de seguridad ciudadana y seguridad vial en los medios de transporte. Esto crea un clima de desconfianza por parte de la población en el transporte público ofrecido y crea la necesidad de buscar alternativas que sean más seguras y puedan satisfacer sus necesidades.

Una de las soluciones que actualmente se proponen para una movilidad segura es el uso compartido del automóvil, que además reduce la emisión de dióxido de carbono ya que existen menos autos emitiendo este gas de efecto invernadero. Muchas veces varias personas comparten el mismo destino en cierta fecha y a cierto rango de hora; la solución acá planteada provee una forma de comunicar a estas personas por medio de una aplicación móvil que centraliza la información de estos viajes y los conecta para compartir un viaje. Además, la aplicación móvil implementa conceptos de teoría de juegos para que las personas se sientan motivadas a compartir el automóvil con más frecuencia.

1. ESTUDIO DE LA TECNOLOGÍA Y SU IMPACTO EN GUATEMALA

1.1. Identificación de la teoría que soporta la investigación

La investigación y la aplicación móvil se basan en dos puntos fundamentales: el modelo de aceptación tecnológica (TAM) y la ludificación o más conocida en inglés como *gamification*. Ambos conceptos son necesarios para crear una aplicación móvil que se adapte a las necesidades de los usuarios de una mejor manera, que hace que la utilicen de una forma regular en su día a día.

1.1.1. Modelo de aceptación tecnológica

También conocido como *Technology Acceptance Model (TAM)*, explica como una nueva tecnología y varios de sus aspectos, son recibidos y usados por el usuario. Aunque varios modelos han sido propuestos con anterioridad en el campo de los sistemas de la información para describir las relaciones anteriores, es este modelo el que ha sido ampliamente usado y aclamado.

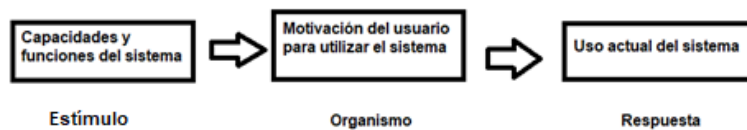
El modelo fue propuesto desarrollando la teoría de acción razonada, la cual propone que la aceptación de una nueva tecnología por parte del usuario está basada en dos factores:

- Utilidad percibida: se refiere a cuanto cree el usuario que la tecnología le ayudará a mejorar la eficiencia o el rendimiento.

- Facilidad de uso: se refiere a hasta qué punto el usuario está cómodo usando las funciones y características de la tecnología.

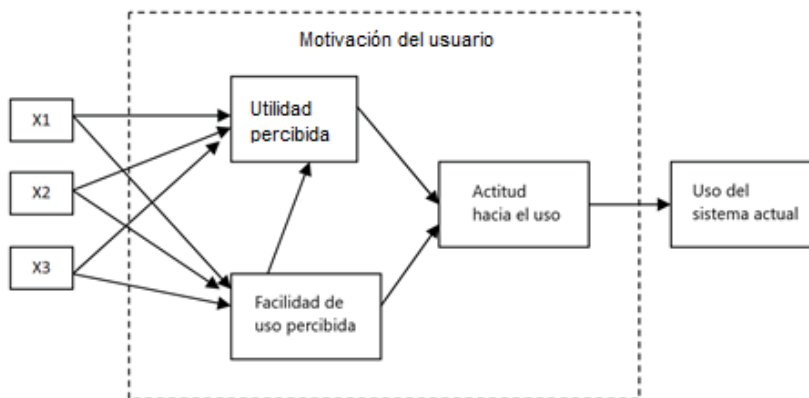
Estos dos factores entonces determinan la actitud del usuario hacia la tecnología. El modelo dice también que la utilidad percibida influenciará el comportamiento de la intención de uso. La actitud determina el comportamiento que a su vez influye en la aceptación real.

Figura 1. **Modelo conceptual para la aceptación de una tecnología**



Fuente: DAVIS, Fred. *A technology acceptance model for empirically testing new end-user informations systems: theory and results*. p. 10.

Figura 2. **TAM original propuesto por Fred Davis**



Fuente: DAVIS, Fred. *A technology acceptance model for empirically testing new end-user informations systems: theory and results*. p. 24.

En la figura 2, Davis sugiere que la motivación de los usuarios puede ser explicada por tres factores: facilidad de uso percibida, utilidad percibida y la actitud hacia el uso del sistema. Él formuló la hipótesis que la actitud de un usuario hacia un sistema era un determinante importante para saber si el usuario usaría o rechazaría el sistema. La actitud del usuario, a su vez, fue considerada influenciada por dos creencias mayores: la utilidad percibida y la facilidad de uso percibida; con esta última teniendo una influencia directa sobre la utilidad percibida. Finalmente, ambas creencias fueron hipotetizadas ser directamente influenciadas por las características diseñadas del sistema, representadas esta por X1, X2 y X3.

Durante etapas posteriores de experimentación, Davis refinaría su modelo para incluir otras variables y modificar las relaciones que él inicialmente formuló. Similarmente, otros investigadores aplicarían y propondrían más elementos al modelo de aceptación tecnológica, tanto así que con el tiempo el TAM evolucionó a un modelo líder en la explicación y predicción del uso de un sistema. Hoy, la investigación en la aceptación de una tecnología continua, y por lo tanto, el entendimiento en los supuestos, las fortalezas y las limitaciones del TAM es esencial para cualquiera que desee la aceptación de una tecnología por parte de un usuario.

1.1.2. Ludificación

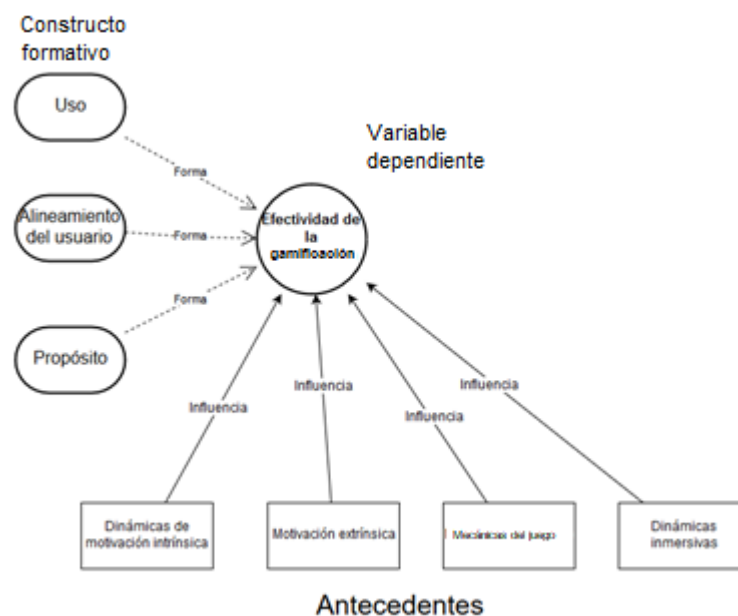
Ludificación o más conocido como *Gamefication*, término en inglés, es el uso de elementos del diseño de juegos en contextos fuera de los juegos para hacer sistemas o procesos más divertidos y atractivos. Informalmente, se puede decir que, hace un sistema más parecido a un juego a través de sus mecánicas. En la práctica, la gamificación o ludificación está más relacionada a la ingeniería de software por medio de aplicaciones web o aplicaciones móviles.

A medida que se obtienen puntos y otras recompensas, se alcanza nuevos niveles con nuevos desafíos, ver el nombre del usuario en un marcador y explorar nuevas misiones, el cerebro se excita y libera dopamina lo cual crea mayor motivación y hace perseguir una recompensa. En su libro *Actionable Gamification: Beyond Points, Badges, and Leaderboards*, uno de los expertos en gamificación de productividad más conocidos del mundo, Yu-kai Chou, señala los siguientes factores que hacen que los juegos y las tareas gamificadas sean increíblemente atractivas:

- Significado épico y vocación: los juegos le hacen creer al usuario que está participando en algo más grande que él mismo al darle una misión.
- Desarrollo y realización: se está preparado para amar un desafío. Si al desafío le sigue una recompensa, genera una sensación de logro que a menudo falta en muchas tareas de la 'vida real'. La sensación de acercarse al objetivo y lograrlo puede brindarse dividiendo el desafío en etapas y mostrando su progreso (puntos, gemas, niveles y más).
- Empoderamiento de la creatividad y retroalimentación: como seres creativos, los humanos se inspiran en las oportunidades de dar vida a su imaginación.
- Propiedad y posesión: los bienes virtuales y el dinero apelan al deseo intrínseco de acumular riqueza.
- Influencia social y parentesco o envidia: los seres humanos tienen una necesidad de conectarse y compararse con los demás. Los juegos apelan a esta necesidad, por ejemplo, permitiendo a los jugadores invitar a amigos, crear misiones grupales, usar mentores y ganar tesoros sociales.

- La escasez y la impaciencia: se desean cosas que no se pueden tener. Se está conectado para buscar de forma intuitiva todo lo que escasea o se desvanece en la disponibilidad.
- Impredecibilidad y curiosidad: las recompensas variables motivan a un jugador a avanzar y buscar nuevas recompensas, principalmente para satisfacer su curiosidad.

Figura 3. **Teoría de la efectividad de la gamificación**



Fuente: elaboración propia, empleando draw.io.

La figura 3 muestra la teoría de la efectividad de gamificación propuesta por Bilal Amir y Paul Ralph, el cual es un marco de referencia que propone que la gamificación es efectiva en la medida que el sistema gamificado es usado, contribuye a cualquier objetivo explícito del sistema y contribuye a los objetivos de sus usuarios. El marco postula cuatro factores principales de eficacia: las

motivaciones intrínseca y extrínseca, la mecánica del juego y la dinámica inmersiva.

Los constructos formativos son las dimensiones de la efectividad de la gamificación. El uso se define como el consumo de una salida del sistema. Para interactuar eficazmente con el sistema gamificado, el desarrollador de software debe recibir y procesar los comentarios, por ejemplo, los informes proporcionados al usuario después de ciertas entradas. La mala retroalimentación puede llevar a reducciones en la participación del usuario e incluso al fracaso de una plataforma de gamificación. Si el sistema tiene un propósito explícito, la gamificación es efectiva en la medida en que contribuye a este propósito.

Los antecedentes son los factores que causan la efectividad de la gamificación. Los antecedentes pueden organizarse de muchas maneras diferentes (por ejemplo, a diferentes niveles de abstracción). El marco propuesto, sin embargo, presenta cuatro antecedentes principales de gamificación efectiva: dinámica de motivación intrínseca, motivadores extrínsecos, mecánica de juego y dinámica de inmersión.

La motivación intrínseca depende de los sentimientos de autonomía, competencia y afinidad. Por ejemplo, los juegos digitales emplean numerosos motivadores extrínsecos, incluidos puntos, niveles, insignias o trofeos.

Se pueden vincular los motivadores extrínsecos del juego con los motivadores extrínsecos del mundo real, por ejemplo, pero otorgando premios tangibles a los desarrolladores que reciben trofeos virtuales.

La mecánica del juego, incluidos el espacio, los objetos, las acciones, las reglas y las habilidades, es el núcleo de un juego. El espacio se define como área de juego, ya sea físico, virtual o ambos. Los objetos son las herramientas utilizadas por el jugador. Las acciones son formas de interactuar con objetos dentro del espacio del juego. Las reglas son leyes que gobiernan el espacio. Las habilidades son un conjunto de habilidades requeridas por el jugador para cumplir un objetivo. Como la gamificación implica la adición de la mecánica del juego, la efectividad de la gamificación depende de la efectividad de esta mecánica.

Las dinámicas de inmersión son factores que afectan la inmersión del jugador en el sistema o actividad gamificada. Dos ejemplos son la historia y la estética. La estética se refiere a las emociones engendradas en un jugador por un sistema, mientras que la historia se refiere a una narración del progreso del jugador.

1.2. Teoría y la relación con la tecnología escogida

Dentro de la creación de la aplicación Jalón se busca que el usuario tenga una aplicación agradable, amigable y de fácil uso para pedir y dar aventones entre ciudades de la república guatemalteca con centralización de la información obtenida y siendo ésta de fácil y rápida búsqueda.

Con esto se asegura que el usuario la perciba como una excelente opción para consultar viajes entre ciudades desde su dispositivo móvil con acceso a internet.

Se debe tomar en cuenta también que la aplicación tenga una alta probabilidad de ser usada, ya que esto hará que se cuente con más viajes por

ofrecer y que el usuario la perciba como útil en su día a día. Esto además, hará que el público objetivo crezca exponencialmente.

La hipótesis anterior propone que la aplicación sea una herramienta de beneficio hacia el usuario, pero que también sienta que es cómoda por medio de los fáciles accesos a los menús sin necesidad de cargar de objetos las pantallas que utiliza. Además, limitar el número de pantallas a las cuales debe acceder para llegar a su objetivo, esto concordando con la facilidad de uso. Dentro de las funcionalidades en donde se aplicará la facilidad de uso están:

- Creación de usuario e ingreso de usuario: estas pantallas requerirán poca información ya que el usuario lo que desea es hacer esto de una manera rápida y sin tantas vueltas.
- Publicar un viaje: el usuario simplemente desea comunicar que hará un viaje en cierta fecha, hora, destino y salida. Adicionalmente, si desea ser más específico, se podrán llenar campos extras para que el viaje sea más cómodo.
- Pedir jalón: poder buscar viajes que se hayan publicado, esta búsqueda debe ser rápida y concisa ofreciendo muchas posibilidades a escoger.
- Dar jalón: simplemente aprobar a las personas que acompañarán al conductor en el viaje creado.

La utilidad percibida se verá reflejada al usuario cuando la información que se le presente sea relevante y de interés para su búsqueda, es decir, el viaje que necesita o personas interesadas en el viaje que ofrece. Por lo tanto, se debe tener

información actualizada y no ambigua para cumplir con sus expectativas. Dentro de las funcionalidades que el usuario puede percibir como útiles están:

- Mensajes entre usuarios: una forma de comunicación entre usuarios para ponerse de acuerdo sobre temas del viaje.
- Manejo de cancelación: tener la posibilidad de cancelar el viaje o la participación en un viaje ya que se podrían dar imprevistos personales.
- Calcular precios entre puntos de ruta: esto ayudará al conductor y pasajeros a cobrar o dar un aporte en forma de dinero, si fuese el caso, ya que esto no es obligatorio, pero es un buen gesto de gratificación por el aventón.
- Puntuación de viajes y usuarios: esta parte es importante en el tema de seguridad lo cual es un factor obligatorio en Guatemala, ya que así los pasajeros podrán escoger con quien ir y los conductores a quienes autorizar el viaje, debido a que es importante realizar el viaje con personas que tengan mejor reputación y poco conflictivas.

La teoría de ludificación o gamificación es importante dentro de la aplicación ya que será una parte fundamental en la participación y compromiso del usuario. Dentro del significado épico y desarrollo están los diferentes niveles con los cuales se podrán identificar y las diferentes medallas que pueden ganar al realizar las diferentes acciones en la aplicación: ser platicadores durante el viaje, realizar viajes largos, compartir comida durante el viaje, música durante el trayecto o simplemente ser buen pasajero.

Los bienes virtuales que pueden ganar los pasajeros y pilotos son las diferentes medallas y niveles; esto crea una influencia social en las demás

personas que quieran tomar un viaje con ellos, motivándolos a crecer en reputación dentro de la comunidad; además de crear curiosidad por ver hasta dónde llegan, realizando acciones positivas en los viajes con las demás personas. Una funcionalidad que puede incluirse como aplicación de la gamificación es el ganar medallas y puntos al realizar o tomar viajes.

2. IDENTIFICACIÓN DE PROBLEMA Y SOLUCIÓN QUE LA APLICACIÓN REALIZARÁ

2.1. Antecedentes

La movilidad sostenible es definida como los desplazamientos que las personas hacen por motivos laborales, sociales, culturales, entre otros; es capaz de satisfacer las necesidades de la sociedad de moverse libremente sin sacrificar valores humanos o ecológicos básicos.

En Guatemala, el modelo de movilidad es de tipo insostenible ya que de ser sostenible sería por medios de transporte que utilicen la luz solar o electricidad, transporte público, compartiendo vehículo o movilizarse en bicicleta o caminando.

Es importante adoptar en el país paradigmas de movilidad sostenible ya que constituye un beneficio para toda la población; contribuye a mejorar la calidad de vida de las personas, reduce el estrés social, se cuida el medio ambiente y se ahorra tiempo y dinero en el costo de los viajes.

Otras de las formas como se mejora la calidad de vida es que se viaja de una manera cómoda, segura y satisfaciendo una necesidad; por último, la parte ambiental al reducir las emisiones de dióxido de carbono hacia la atmósfera.

El uso compartido del vehículo es una de las soluciones de movilidad sostenible que ha tenido bastante aceptación en los últimos años alrededor del mundo; Guatemala es uno de los países en donde actualmente su uso es menos

frecuente debido al factor confianza y seguridad, esto derivado de las condiciones sociales del país.

Poco a poco las aplicaciones de uso compartido de vehículo han empezado a ganar terreno en las ciudades importantes del país como la ciudad de Guatemala, Antigua Guatemala y Quetzaltenango, lo cual hace que este mercado se vaya abriendo paso en los últimos años.

Una característica del sistema de uso compartido de vehículo es que debe ser dinámico y generar confianza en él a un cien por ciento por parte de los usuarios; actualmente, existen grupos en la red social Facebook que cuentan con la confianza de muchas personas; sin embargo, la información se encuentra dispersa en la plataforma, lo cual no la hace totalmente eficiente.

2.1.1. Transporte extraurbano en Guatemala

El transporte público en Guatemala es uno de los grandes problemas que afecta al país, solo el urbano y el extraurbano; las principales características de este problema es la baja calidad del servicio que incluye vehículos en mal estado, excesiva velocidad, poca comodidad al ir en muchas ocasiones sobrecargados, precios que no corresponden a las rutas; otra característica es la violencia que se sufre en el transporte: verbal, física y sexual; asaltos, robos y asesinatos son hechos que suceden con alta frecuencia.

Según datos de Edgar Guerra, defensor de usuarios del transporte público de la PDH, en el país se cometen entre 90 y 100 asaltos diarios en unidades de transporte urbano y extraurbano, y es considerado solo el 40 % de la realidad cotidiana.

Guerra también confirma que los hechos de tránsito pueden prevenirse con diferentes acciones. Para empezar, muchos pilotos no están capacitados. Casi cualquiera puede obtener una licencia. Luego está el mantenimiento de los vehículos. Llantas reencauchadas varias veces, sistemas eléctricos en mal estado. En el caso del transporte, especialmente el extraurbano, existen talleres que remodelan unidades diseñadas para ser transportes escolares en EE.UU. y los transforman en buses extraurbanos. Estas modificaciones, hechas por personas no especializadas, pueden generar daños que provoquen, los mal llamados, 'accidentes'.

La aplicación contribuye a que los usuarios puedan optar por otro medio de transporte que pueda ofrecerles seguridad y comodidad que son los dos grandes factores de los que carece actualmente el sistema de transporte extraurbano.

2.1.2. Contaminación por CO2

El calentamiento global es uno de los temas de gran preocupación a nivel mundial; el CO2 es uno de los gases más contaminantes a nivel medioambiental; actualmente, se han fijado marcas históricas en sus niveles debido a diferentes políticas adoptadas por países entre 2017 y 2019, se estima que a finales de 2018 se habían emitido 37 gigatoneladas de este gas.

Se prevé que las emisiones de dióxido de carbono aumenten hasta un 130 % desde el periodo actual hasta 2050, según las cifras que aporta la Agencia Internacional de Energía, y que harían falta uno 45 000 millones de dólares para acometer las medidas necesarias para reducir estas emisiones a la mitad. Las principales causas del aumento de las emisiones de CO2 son las actividades humanas. Al haber incrementado la población mundial, cada vez son mayores

los requerimientos en energía, que se producen fundamentalmente por el uso de los combustibles fósiles, que producen importantes emisiones.

Al compartir el vehículo se evita que las personas utilicen las unidades obsoletas del transporte público, el cual solo genera más contaminación al tener motores antiguos que producen más gases de efecto invernadero. Además, al compartir el vehículo se reduce la emisión de dióxido de carbono debido a que se evita el uso de uno, dos o tres vehículos extras conducidos por personas individuales.

Añadido a lo anterior, la flexibilidad de este tipo de viajes hace que los usuarios no estén atados a los horarios convencionales del transporte; esto es un gran problema para las personas que necesitan realizar un viaje fuera de estos horarios. Conforme crezca la cantidad de usuarios, la flexibilidad de horario se hará mayor, ya que existirán más opciones de viaje entre los destinos que se desea movilizarse.

2.1.3. Aplicaciones móviles

Una de las soluciones viables para atacar los problemas anteriormente descritos es por medio de las aplicaciones móviles que se han convertido en sistemas fundamentales de las tecnologías de la información. El fácil uso y acceso a la información por medio de un dispositivo móvil con conexión a internet hacen que sea accesible a una gran cantidad de público.

La seguridad y centralización de la información de viajes y personas es una de las ventajas que proponen las aplicaciones móviles, esto ofrece al usuario confianza y comodidad al realizar las consultas que puedan satisfacer su necesidad.

Al incorporar el factor de ludificación se podrá hacer de la experiencia de los usuarios más amena y divertida dentro de la aplicación ya que se darán recompensas virtuales a aquellas personas que la utilicen frecuentemente.

2.2. Benchmarking

A continuación, se describen diferentes aspectos relacionados a la comparación de la aplicación con respecto a otras ya existentes.

2.2.1. Mercado nacional

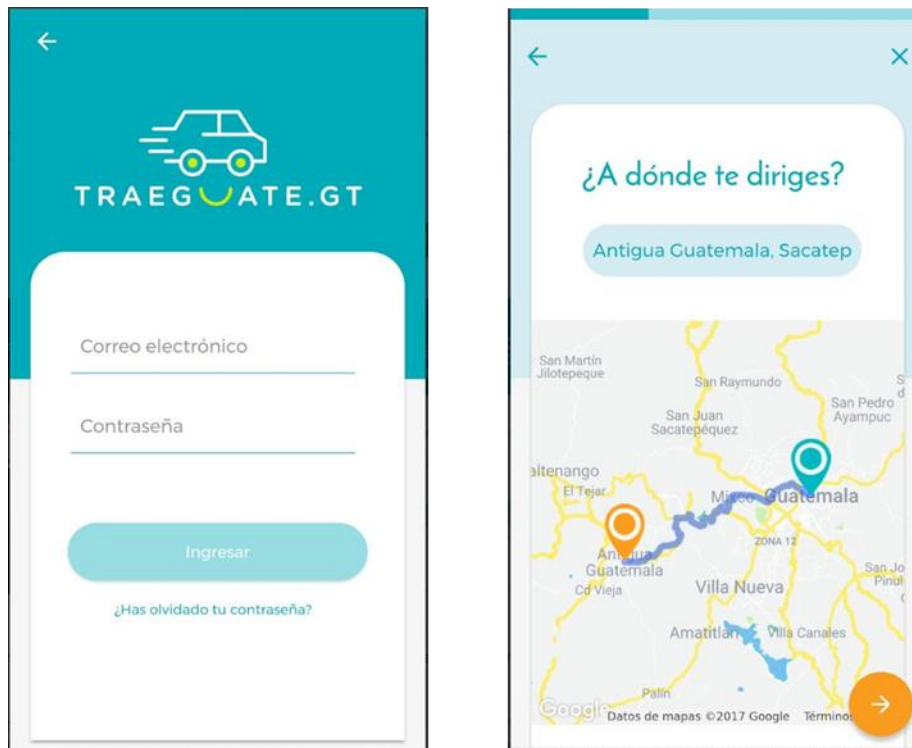
Dentro del mercado nacional no existe alguna aplicación móvil con las mismas características o con el mismo enfoque que la solución móvil aquí propuesta; sin embargo, existen referencias actuales que hacen un trabajo parecido.

2.2.1.1. TraeGuate

Es un sistema de auto compartido o *carpooling*, que funciona con base en *tokens* los cuales se adquieren comprándolos en la misma aplicación y el servicio de viajes se paga por medio de estos *tokens*.

Al ser un servicio de paga ofrecen descuentos en empresas afiliadas, asistencia por medio de aseguradoras.

Figura 4. **Pantallas de la aplicación móvil de TraeGuate.GT**



Fuente: Aplicación TraeGuate.GT.

https://play.google.com/store/apps/details?id=app.traeguate.com&hl=es_GT. Consulta: 30 de mayo de 2019.

2.2.1.2. Guate jalón Facebook

Esta comunidad de la red social Facebook funciona como un grupo de personas que viajan y ofrecen viajes entre ciudades de Guatemala; la información se hace por medio de publicaciones; las crean para indicar que necesitan un 'jalón' o que ofrecen jalón a otras personas de un lugar a otro a cierta hora.

Figura 5. **Página inicial del grupo Guate jalón**



Fuente: Página del grupo Guate jalón. <https://www.facebook.com/groups/616581051697715/>.

Consulta: 30 de mayo de 2019.

Dentro del grupo existen reglas a través de etiquetas o *hashtags* para elaborar las publicaciones ya que de esta manera la información se encuentra segmentada de una mejor manera; sin embargo, estas reglas no siempre se cumplen y es difícil que exista un moderador todo el tiempo en el grupo para supervisarlos. Algunas etiquetas son:

- Doy jalón
- Pido jalón
- Paquetes
- Denuncia
- Noticias

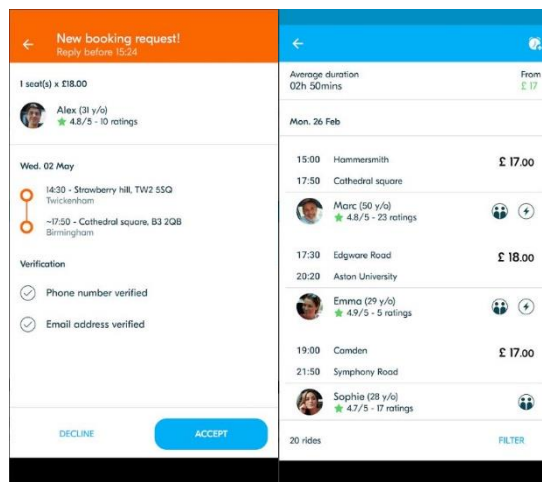
- Eventos

2.2.2. Mercado internacional

En el mercado internacional existen opciones que cubren la necesidad de transporte extraurbano en algunos países; pero la herramienta más conocida a nivel mundial por su alcance y tiempo en el mercado es BlaBlaCar.

Esta aplicación funciona por medio de reservaciones pagadas a viajes publicados entre ciudades; tiene la capacidad de escoger ciudades intermedias que puedan ser útiles a otros usuarios. Estos precios se calculan con base en el kilometraje entre ciudades que crean un precio base y luego el conductor decide cuánto más podría llegar a cobrar. Las formas de pago son por medio de una tarjeta de crédito o débito y PayPal.

Figura 6. Pantallas de creación de viajes y de solicitudes de viaje



Fuente: Aplicación

BlaBlaCar. https://play.google.com/store/apps/details?id=com.comuto&hl=es_419. Consulta: 31 de mayo de 2019.

La aplicación incluye perfiles de los conductores y pasajeros para verificar la valoración que han hecho otros usuarios, sus gustos de viajes y qué tipo de verificación de identidad se ha hecho a través de la aplicación.

Figura 7. **Perfil de usuarios de la aplicación móvil BlaBlaCar**



Fuente: Perfil de usuario de BlaBlaCar.

https://play.google.com/store/apps/details?id=com.comuto&hl=es_419. Consulta: 31 de mayo de 2019.

2.3. Mercado objetivo

Esta aplicación está dirigida a todas las personas que viajan a través del territorio nacional, nacionales o extranjeros; tiene los perfiles de conductor y viajero; según el viaje el usuario puede tomar uno u otro perfil. Ambos perfiles tienen como característica que disfrutan de la compañía y que realicen viajes a menudo.

- Conductor: es la persona que está al mando del vehículo, crea los viajes con sus lineamientos y aprueba a las personas que podrán realizar el viaje con él, luego de que haya sido solicitado.
- Viajero: es quien solicita unirse a un viaje que coincida con su destino y partida; si lo desea, una contribución al conductor. Este debe estar sujeto a los lineamientos descritos en el viaje por el conductor.

Ambas personas deben ponerse de acuerdo con el lugar donde se encontrarán para realizar el viaje y el lugar específico para dejar a los viajeros en la ciudad.

Los recursos que necesitan las personas para utilizar la aplicación móvil serán una conexión a internet y un móvil en donde instalar la aplicación; es importante saber que deberán crear una cuenta primero para disponer de los servicios anteriormente mencionados.

3. ANÁLISIS Y DISEÑO DE LA APLICACIÓN

3.1. Arquitectura de la solución

La arquitectura propuesta es robusta y flexible para poder adaptarse a futuros cambios en el entorno completo.

3.1.1. API

Las siglas provienen de la definición en inglés de *Application Programming Interface*, o interfaz de programación de aplicaciones. Una API es un conjunto de comandos, funciones, protocolos y objetos que los desarrolladores pueden usar para crear software e interactuar con sistemas externos.

3.1.2. REST/RESTFul

Significa transferencia de estado representacional (*Representational State Transfer*), conocido como RESTful y es un estilo de arquitectura de software para recursos distribuidos que utiliza HTTP como vía de comunicación. Entre sus principios se pueden mencionar:

- Utiliza los verbos o protocolos HTTP
- No mantiene estados
- Expone las URLs en forma de directorios
- Puede transferir distintos lenguajes de marcado

Tabla I. **Protocolos HTTP**

Protocolo	Funcionalidad
GET	Método utilizado para obtener información con base en las peticiones con formato JSON hacia el servicio RESTful.
POST	Método utilizado para realizar actualizaciones o modificaciones de datos, se realiza la petición con formato JSON hacia el servicio RESTful.
PUT	Método utilizado para realizar inserciones de datos, realiza la petición con formato JSON hacia el servicio RESTful.
DELETE	Método utilizado para eliminar datos, se realiza la petición con formato JSON hacia el servicio RESTful.

Fuente: elaboración propia.

3.1.3. JSON Web Token

Json Web Token es un conjunto de medios de seguridad para peticiones http y así representar demandas para ser transferidos entre dos partes (cliente y servidor).

La mejor alternativa es llevar a cabo la autenticación haciendo uso de *tokens* que vayan del servidor al cliente; un usuario hace *login* (no necesita enviar *token* porque no lo tiene); una vez el servidor lo valida, retorna un *token* como respuesta y el usuario debe enviar dicho *token* en las siguientes peticiones para acceder a los recursos del servicio.

3.1.4. Aplicaciones híbridas

Son aplicaciones que combinan el desarrollo nativo con distintas tecnologías web, teniendo en cuenta la multiplataforma de estas últimas y su comunicación con APIs nativas. Estas son escritas en lenguajes HTML, CSS y JavaScript. Entre las ventajas que tienen las aplicaciones híbridas se pueden enumerar:

- Se pueden visualizar en cualquier teléfono móvil
- Permitir la reutilización de código ahorrando tiempo de desarrollo
- Tienen buen rendimiento en cualquier plataforma
- El costo de inversión es más bajo que el de las aplicaciones nativas

3.1.5. Arquitectura MEAN

Las tecnologías que conforman MEAN son: MongoDB, Express, Angular y NodeJs. Todas estas tecnologías usan JavaScript como su *core* y que juntas se convierten en una herramienta poderosa para el desarrollo de aplicaciones web y móviles.

Las cuatro tecnologías unidas permiten desarrollar aplicaciones escalables, cercanas al mundo móvil y a la necesidad de tiempo real.

MongoDB es la base de datos; NodeJs permite ejecutar Javascript desde el servidor; Express es el *framework* que se utilizará para NodeJS y Angular es el *framework* JavaScript del lado del cliente o, en este caso, la aplicación móvil híbrida.

3.1.6. Descripción de la arquitectura

La arquitectura planteada se realizó pensando principalmente en la escalabilidad del sistema, tanto a nivel de funcionalidades como de nuevas implementaciones; incluye el aumento en la cantidad de usuarios y viajes a través del tiempo.

Otra característica que hay que tomar en cuenta es la concurrencia, es decir, el número de peticiones que se realizan al API y que todas estas puedan ser respondidas de la manera correcta con la información necesaria.

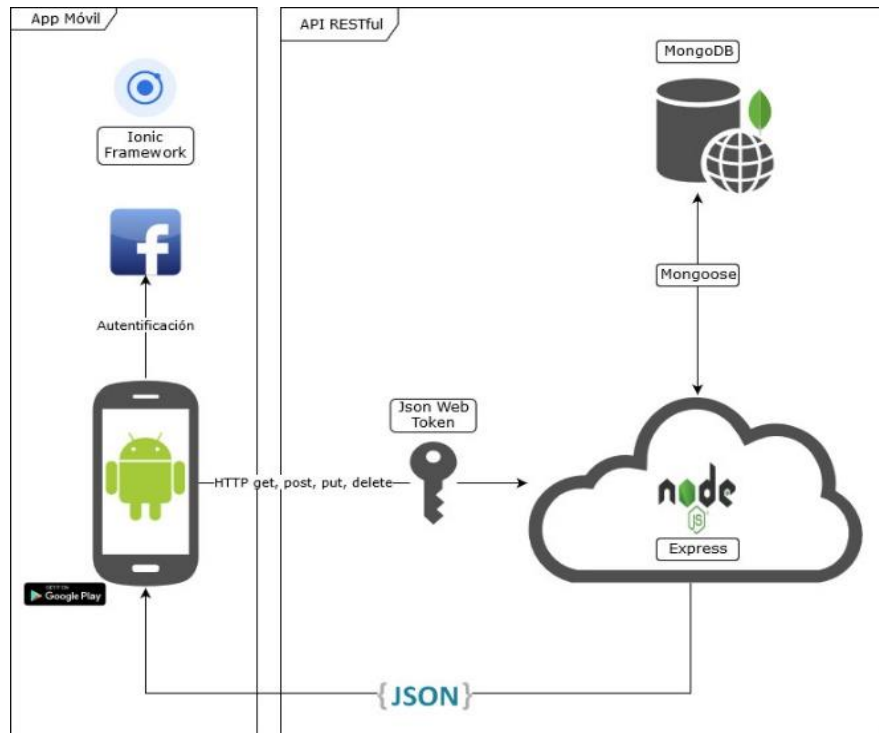
La arquitectura del sistema como se muestra en la figura siguiente consta de dos subsistemas relacionados, los cuales pueden aumentar dependiendo las futuras implementaciones:

- Aplicación Móvil
- API RESTFul

La relación entre la App móvil y la API RESTful es mediante el protocolo HTTP utilizando la convención REST. Como medida de seguridad para la comunicación entre la aplicación móvil y la API se utiliza un Json Web Token que se construye a partir del correo del usuario y con cierto tiempo de expiración. Este *token* es necesario para acceder a los servicios prestados por la API; si la API no logra validar el *token*, esta regresa un error de autenticación, de lo contrario, devuelve la información requerida en formato JSON.

La autenticación de un usuario se puede hacer por medio de los usuarios registrados en la plataforma o por medio de la autenticación de la red social Facebook.

Figura 8. **Arquitectura de la solución**



Fuente: elaboración propia, empleando el programa draw.io.

La API realizada con el *framework* Express para NodeJs se comunicará con la base de datos haciendo uso de la librería Mongoose para el modelado de información hacia la base de datos MongoDB.

Esta arquitectura está basada en la arquitectura MEAN, la cual se basa en tener solamente un lenguaje de programación para construir sitios web; el lenguaje de programación en común es JavaScript.

Dentro de la adaptación de esta nueva arquitectura se utiliza Ionic Framework para la elaboración de la aplicación móvil que a su vez utiliza el *framework* Angular inherente a la arquitectura MEAN.

3.2. Casos de uso

A continuación, se presentan los casos de uso para los usuarios finales.

3.2.1. Crear usuario

Realiza la creación de un nuevo usuario para el sistema ya que para utilizar las funcionalidades se necesita ser un usuario registrado.

- Flujo principal
 - El usuario instala la aplicación móvil.
 - En la pantalla inicial elige la opción de nueva cuenta.
 - El usuario llena los campos obligatorios del formulario con sus datos.
 - El usuario llena los campos no obligatorios en el formulario con sus datos.
 - El usuario termina el registro al elegir la opción de registrar.

- Flujo alternativo
 - El usuario instala la aplicación móvil.
 - En la pantalla inicial elige la opción de asociar con Facebook.

- El usuario ingresa sus credenciales de Facebook.
- Termina de llenar los campos obligatorios.
- El usuario termina el registro al elegir la opción registrar.

3.2.2. Ingresar a la aplicación

Se describe el ingreso a la aplicación móvil al tener ya creado un usuario en ella y las diferentes formas de acceder a ella.

- Flujo principal
 - El usuario abre la aplicación móvil.
 - En la pantalla inicial ingresa su nombre y contraseña.
 - Luego selecciona la opción entrar.
 - De coincidir el nombre y la contraseña en el sistema le mostrará la pantalla de bienvenida.
 - De no coincidir el nombre y la contraseña se le indicará que se ha equivocado en las credenciales.
- Flujo alternativo
 - El usuario abre la aplicación móvil.

- En la pantalla inicial elige la opción de entrar con Facebook.
- Ingresa sus credenciales de Facebook.
- De acceder con sus credenciales acceder automáticamente a la pantalla de bienvenida.
- De no coincidir las credenciales, podrá seguir intentando o ingresar por medio del ingreso de la aplicación.

3.2.3. Crear viaje

Describe los pasos para crear o publicar un viaje. El usuario que dispara este evento es el usuario conductor o que realizará el viaje y que dará 'jalón'.

- Flujo principal
 - El usuario se encuentra en la pantalla de bienvenida y en el menú deberá ingresar a la opción publicar viaje.
 - En la primera pantalla de publicar viaje deberá llenar los campos de fecha y hora de partida como el de llegada, el lugar o ciudad de donde se partirá y cantidad de pasajeros como datos obligatorios.
 - Como datos complementarios una donación sugerida e información complementaria o comentarios sobre el viaje.

- En la segunda pantalla se ingresarán de manera opcional las paradas y los puntos de encuentro donde se podrá recoger a una persona.
- Al final se escogerá la opción de publicar viaje.

3.2.4. Buscar viaje

Esta opción es utilizada por las personas que buscan una oferta de viaje compartido que concuerde con su punto de partida, punto de llegada, fecha y hora. De no encontrar un viaje que se ajuste a sus necesidades podrá programar una alarma si en caso se crea un viaje con sus especificaciones.

- Flujo principal
 - El usuario se encuentra en la pantalla de bienvenida y en el menú deberá ingresar a la opción buscar jalón.
 - En la pantalla de búsqueda ingresa su punto de partida y su destino.
 - En la misma pantalla de búsqueda ingresa la fecha y hora de partida.
 - Luego elije la opción guardar y buscar.
 - En la siguiente pantalla se le mostrará un listado de viajes que coinciden con los parámetros de búsqueda, desde el que más coincide al que menos coincide.

- El usuario puede o no escoger un viaje.
- Flujo alternativo
 - El usuario se encuentra en la pantalla de bienvenida y en el menú deberá ingresar a la opción buscar jalón.
 - En la pantalla de búsqueda ingresa su punto de partida y su destino.
 - En la misma pantalla de búsqueda ingresa la fecha y hora de partida.
 - Luego elije la opción guardar y buscar.
 - El usuario no encuentra ningún viaje que coincida con sus parámetros o decide que algún viaje no le convence.
 - El usuario puede elegir la opción programar una alarma que le avisará cuando haya un nuevo viaje que coincida con sus parámetros de búsqueda.

3.2.5. Responder solicitud de viaje de usuario

Al crear un viaje nuevo, el creador tiene la opción de aceptar o rechazar las solicitudes de otros usuarios de compartir el viaje, su decisión puede basarse en la puntuación de los usuarios u otros factores como espacio, equipaje, entre otros.

- Flujo principal

- El usuario se encuentra en la pantalla de bienvenida y en el menú deberá ingresar a la opción peticiones.
- En la pantalla de datos del viaje le aparecerá la lista peticiones a su viaje, pudiendo aceptarlas o rechazarlas.
- Flujo alternativo
 - El usuario se encuentra en la pantalla de bienvenida y en el menú deberá ingresar a la opción peticiones.
 - El usuario no cuenta con un viaje creado, le aparecerá simplemente que no cuenta con viajes creados.

3.2.6. Solicitar unirse a un viaje

Luego de buscar el viaje y tener resultados con la búsqueda se podrá solicitar unirse a un viaje y dependerá del conductor o creador del viaje de aceptarlas solicitudes.

- Flujo principal
 - El usuario previamente tuvo que haber buscado un viaje y tener resultados con su búsqueda.
 - El usuario elige el mejor viaje que se ajuste a sus necesidades.
 - En la pantalla de información de viaje puede ingresar comentarios hacia el conductor para solicitar unirse al viaje.

- Solicita por medio de la opción llévame.
- El usuario creador del viaje acepta la petición.
- Flujo alternativo
 - El usuario previamente tuvo que haber buscado un viaje y tener resultados con su búsqueda.
 - El usuario elige el mejor viaje que se ajuste a sus necesidades.
 - En la pantalla de información de viaje puede ingresar comentarios hacia el conductor para solicitar unirse al viaje.
 - Solicita por medio de la opción llévame.
 - El usuario creador del viaje rechaza la petición.
 - El usuario deberá buscar otra opción de viaje para solicitar.

3.2.7. Puntuar viaje

Al terminar un viaje tanto el conductor como los acompañantes podrán puntuar el viaje, es decir puntuarse entre usuarios. Esto representa una reputación ante la comunidad.

- Flujo principal

- El usuario que creó el viaje deberá dar por finalizado el viaje en la parte del menú de la pantalla principal.
- Elige la opción finalizar viajes.
- Le aparecerá una pantalla con los viajes que tiene pendiente de finalizar siempre y cuando sea el creador.
- Podrá puntuar a cada usuario que viajó con él.
- Flujo alternativo
 - El usuario creador finaliza el viaje.
 - A los viajeros les llegará un mensaje en el cual ya podrán puntuar al conductor y los otros pasajeros.

3.2.8. Crear mensaje

Enviar mensajes entre usuarios es importante ya que de esta manera se podrán poner de acuerdo sobre el viaje y los detalles de él. Dentro de la aplicación se podrán enviar mensajes entre usuarios siempre y cuando haya un viaje en el que irán ambos.

- Flujo principal
 - Un usuario crea un viaje para poder compartirlo.
 - Otros usuarios piden unirse al viaje.

- El usuario creador acepta la petición o peticiones de unirse al viaje.
- En la pantalla de información de viaje aparecerá un listado de usuarios que compartirán el viaje.
- Elige a qué usuario desea enviarle un mensaje y escribe el mensaje.
- Lo envía.
- Flujo alternativo
 - Un usuario crea un viaje para poder compartirlo.
 - Otros usuarios piden unirse al viaje.
 - El usuario creador acepta la petición o peticiones de unirse al viaje.
 - En la pantalla de información de viaje aparecerá la opción comunicarse con el conductor.
 - Elige esta opción escribe el mensaje.
 - Lo envía.

3.2.9. Responder mensaje

Para responder mensajes se debe hacer por medio del menú de la pantalla principal para poder ver el historial de la conversación de una manera más fácil.

- Flujo principal
 - El usuario se encuentra en la pantalla de bienvenida y del lado izquierdo se encuentra la opción de mensajes.
 - Elige la opción de mensajes.
 - Elige alguna conversación para responder.
 - En la conversación escribe algún mensaje y escoge la opción enviar.

3.2.10. Dar medallas

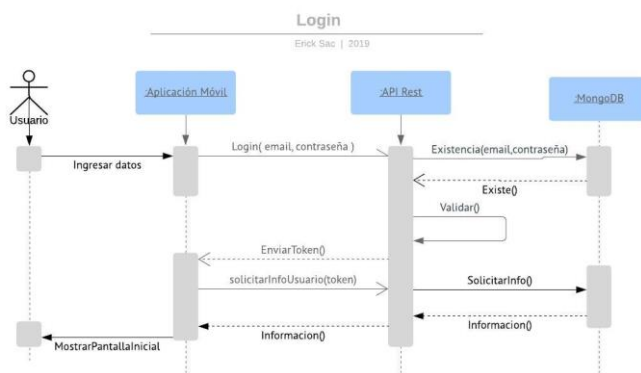
Esta acción la realiza el sistema ya que se basa en las actividades y recomendaciones de los usuarios, dependiendo de la cantidad de viajes, distancias, comportamiento en el viaje, entre otros.

- Flujo principal
 - El usuario realiza un viaje ya sea como conductor o pasajero.
 - Entre los usuarios se calificarán y se agregarán los datos del viaje al perfil de los usuarios.
 - El sistema verifica los datos para comparar y dar las medallas que se hayan ganado, de lo contrario no dará ninguna medalla.

3.3. Diagramas de secuencias

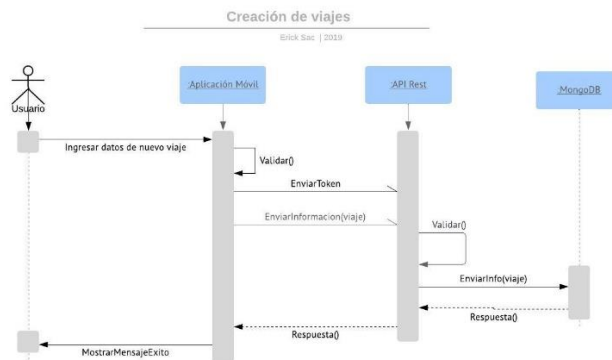
Estos diagramas representan las actividades en el sistema a través del tiempo.

Figura 9. Diagrama, login



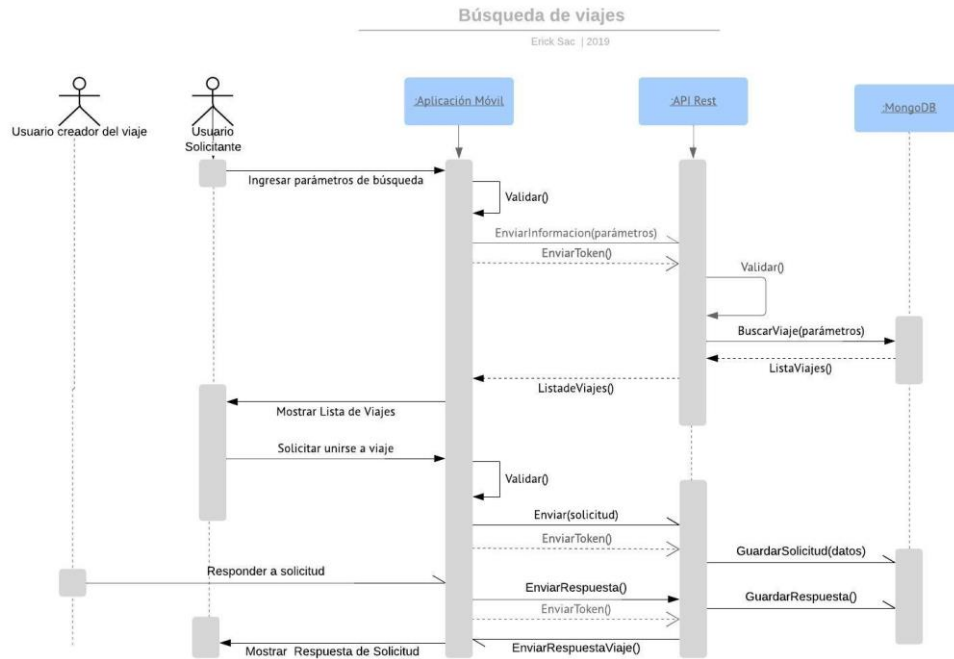
Fuente: elaboración propia, empleando Lucidchart.

Figura 10. Diagrama, creación de viajes



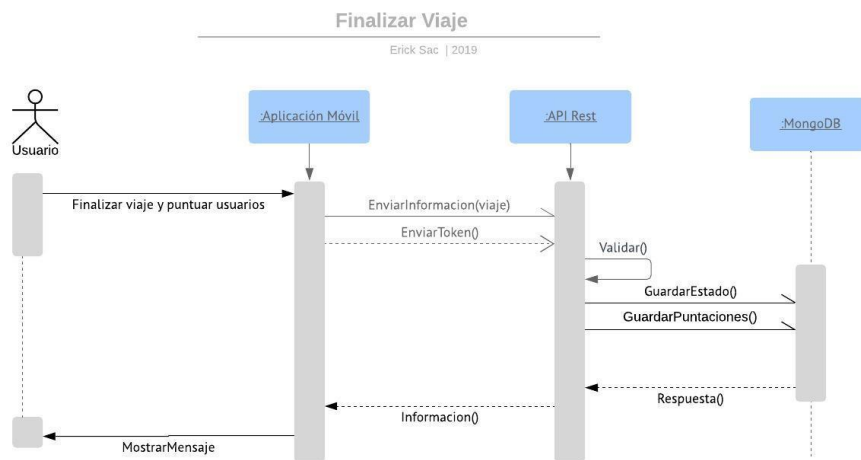
Fuente: elaboración propia, empleando Lucidchart.

Figura 11. Diagrama, búsqueda de viajes



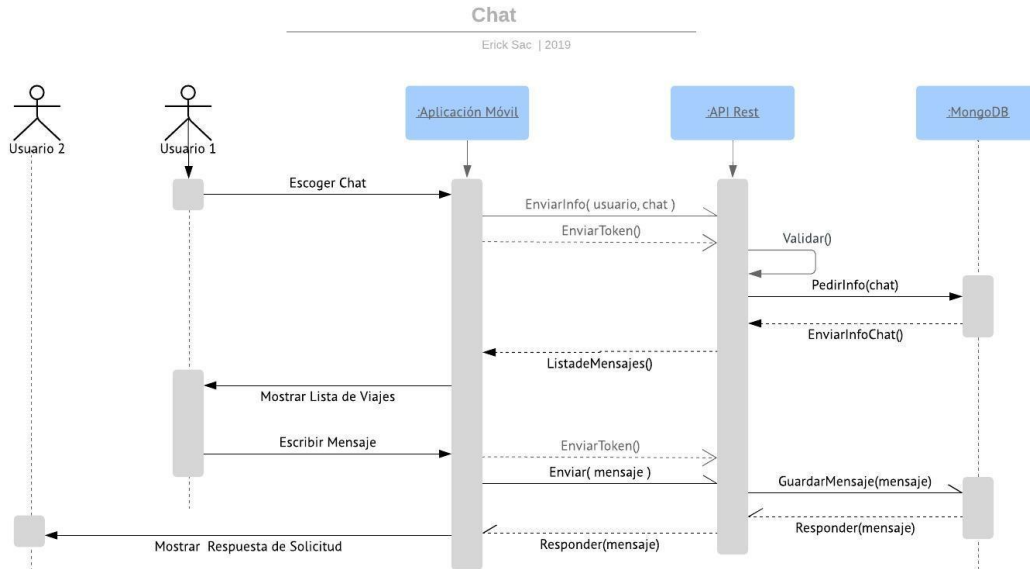
Fuente: elaboración propia, empleando Lucidchart.

Figura 12. Diagrama, finalizar viaje



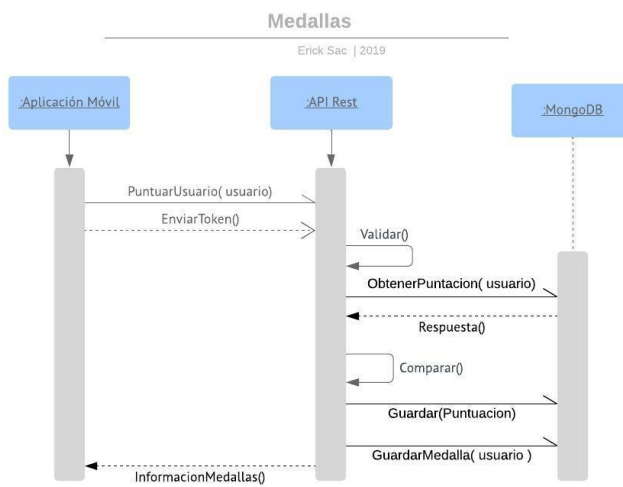
Fuente: elaboración propia, empleando Lucidchart.

Figura 13. Diagrama, chat



Fuente: elaboración propia, empleando Lucidchart.

Figura 14. Diagrama, medallas



Fuente: elaboración propia, empleando Lucidchart.

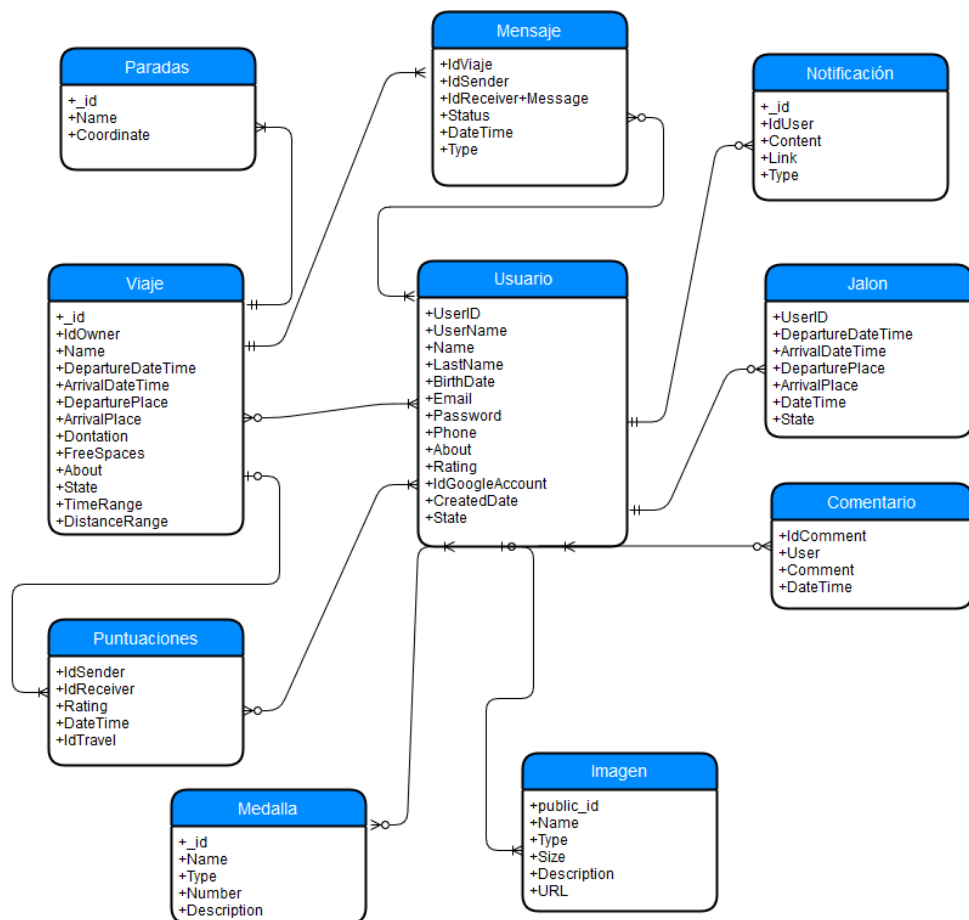
3.4. Diseño de base de datos

A continuación, se presentan los puntos relacionados con la base de datos de la aplicación.

3.4.1. Modelo de datos

En la figura 15 se muestra el modelo de datos de forma gráfica.

Figura 15. Diagrama de modelo de datos



Fuente: elaboración propia, empleando draw.io.

3.4.2. Diccionario de datos

Esta sección describe los nombres y tipos de datos con los que cuenta la base datos.

Tabla II. **Usuario**

<i>Usuario</i>	
UserID	ObjectId
UserName	String
Name	String
LastName	String
BirthDate	Date
Email	String
Password	String
Phone	String
About	String
Rating	Number
IdGoogleAccount	String
CreatedDate	Date
State	String
Imagen	
_Id	REF_ObjectId
Medalla	
_Id	REF_ObjectId

Fuente: elaboración propia.

Tabla III. **Comentario**

<i>Comentario</i>	
_Id	ObjectId
RefUser	ObjectId
Comment	String
DateTime	Date

Fuente: elaboración propia.

Tabla IV. **Viaje**

<i>Viaje</i>	
_Id	ObjectId
IdOwner	REF_ObjectId
Name	String
DepartureDate	Date
ArrivalDate	Date
DeparturePlace	String
ArrivalPlace	String
Donation	Number
FreeSpaces	Number
About	String
State	String
TimeRangeMin	Number
DistanceRangeMts	Number
<i>Paradas</i>	
Name	String
Coordinate	String
<i>Viajeros</i>	
User_Id	REF_ObjectId
Username	String

Fuente: elaboración propia.

Tabla V. **Mensaje**

<i>Mensaje</i>	
_Id	ObjectId
IdViaje	REF_ObjectId
IdSender	REF_ObjectId
IdReceiver	REF_ObjectId
Message	String
Status	String
Date	Date
Type	String

Fuente: elaboración propia.

Tabla VI. **Notificación**

Notificación	
_id	ObjectId
idUser	REF_ObjectId
Content	String
Link	String
Type	String

Fuente: elaboración propia.

Tabla VII. **Jalón**

Jalón	
_id	ObjectId
DepartureDate	Date
ArriavalDate	Date
DeparturePlace	String
ArrivalPlace	String
DateTime	Date

Fuente: elaboración propia.

Tabla VIII. **Imagen**

Imagen		
public_id		String
Name		String
Type		String
Size		String
Description		String
URL		String

Fuente: elaboración propia.

Tabla IX. **Medalla**

Medalla	
_Id	ObjectId
Name	String
Type	String
Number	Number
Description	String

Fuente: elaboración propia.

Tabla X. **Puntuación**

Puntuacion	
IdSender	REF_ObjectId
IdReceiver	REF_ObjectId
Rating	Number
DateTime	Date
IdTravel	REF_ObjectId

Fuente: elaboración propia.

3.5. **Prototipo**

Se presentan las pantallas preliminares de la aplicación y una breve explicación sobre ellas.

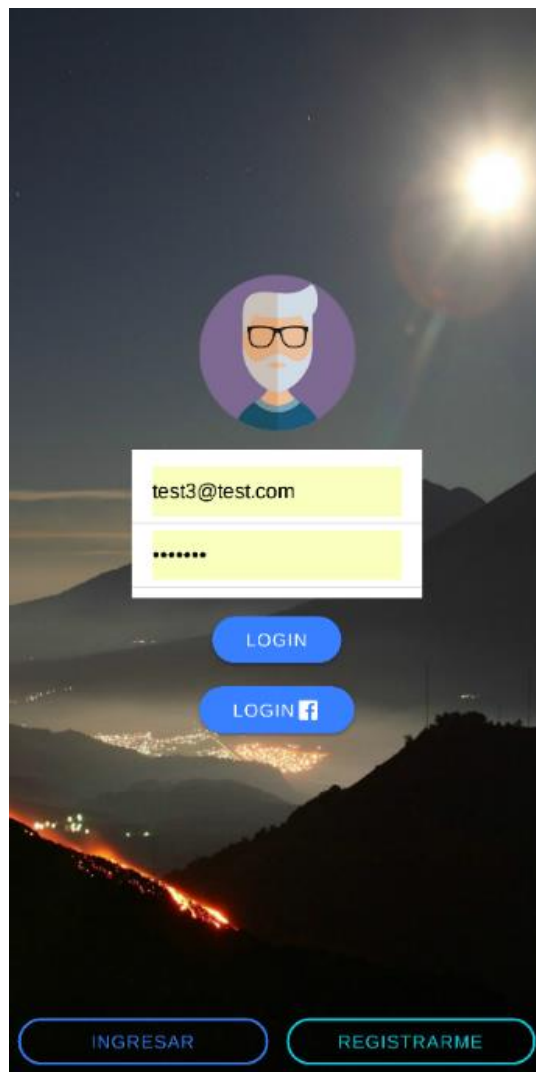
3.5.1. **Login**

La pantalla de ingreso contendrá un pequeño avatar y los campos de email y contraseña cuando el usuario quiera ingresar por medio de la aplicación. Tendrá la opción de crear un nuevo usuario si en caso es un nuevo usuario.

También, tendrá el botón de ingresar por medio de Facebook; si fuese la primera vez que ingresa se le creará un usuario al ingresar por este medio.

En los campos de registro de nuevo usuario se tendrá la validación de estos para verificar la integridad de datos.

Figura 16. **Prototipo, pantalla de login**



Fuente: elaboración propia, empleando Ionic *Framework*.

3.5.2. Inicio

La pantalla de inicio contendrá una imagen de fondo con las opciones de crear un viaje nuevo o de buscar alguno. Al deslizar podrá encontrar los viajes recientes y del lado derecho un botón para ver su historial.

Figura 17. Prototipo, pantalla de inicio

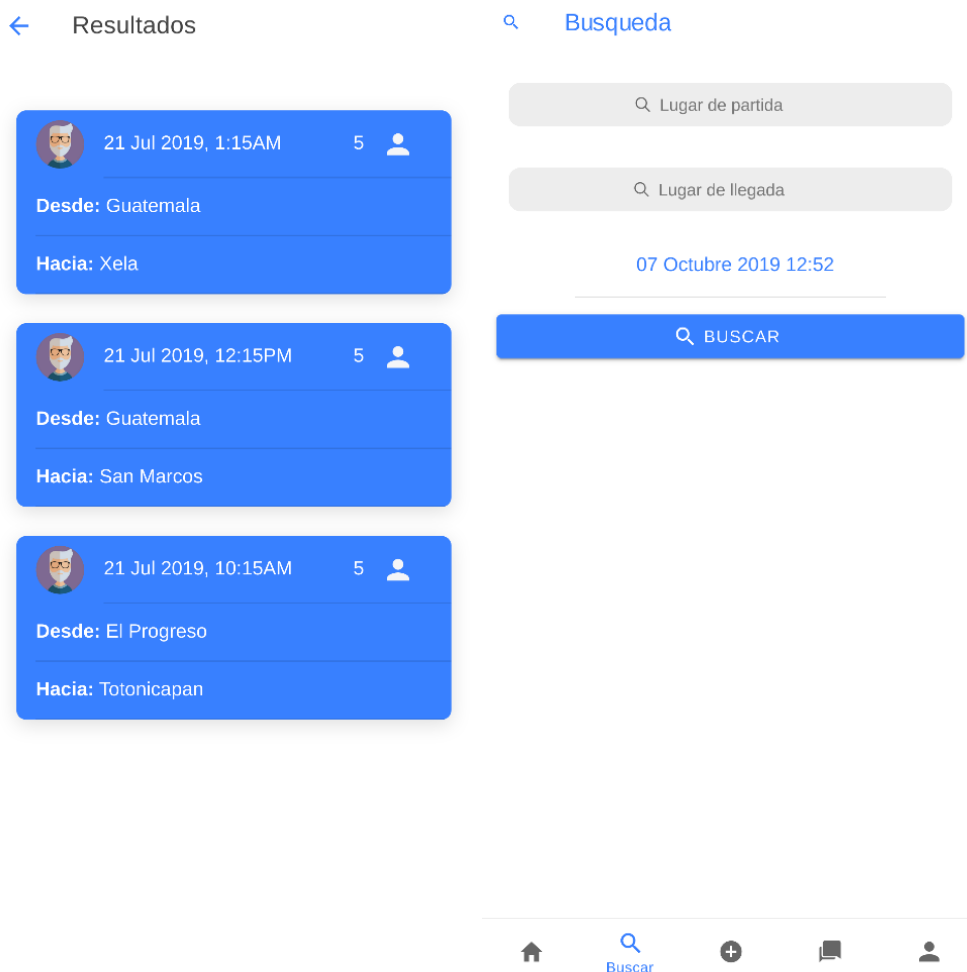


Fuente: elaboración propia, empleando Ionic Framework.

3.5.3. Búsqueda de viajes

En la búsqueda el usuario podrá buscar algún viaje por medio del lugar de partida, el lugar de llegada y la hora que desea salir. Se le mostrará un listado de resultados con los viajes que se aproximan a su búsqueda.

Figura 18. Prototipo, pantallas de búsqueda y resultados



Fuente: elaboración propia, empleando Ionic *Framework*.

3.5.4. Creación de viaje

La creación de viajes se realizará por pasos; se indican los lugares de partida y destino, las paradas, la hora y fecha del viaje, la cantidad de pasajeros, alguna donación opcional y una pequeña descripción de lo que se espera del viaje.

Figura 19. Prototipo, pantallas de creación de viaje

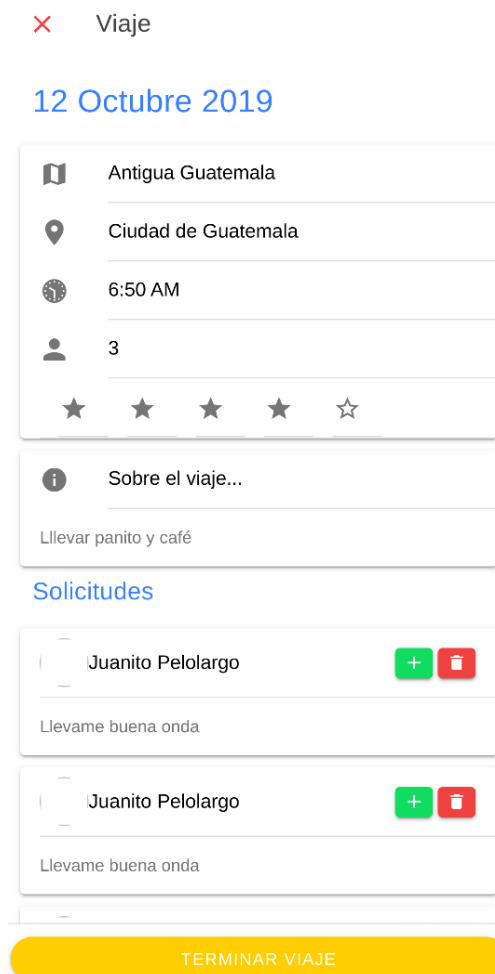
El prototipo muestra una interfaz de usuario para la creación de un viaje, dividida en dos pantallas. La pantalla de la izquierda, titulada 'Jalón' con un icono de cerrar (X), contiene tres secciones: '¿De dónde sales?' con un campo de texto que muestra 'Ex. Antigua', '¿A dónde vas?' con un campo de texto que muestra 'Ex. Panajachel', y '¿Pedir alguna donación?' con el número '0'. La pantalla de la derecha, titulada 'Pasajeros' con un icono de retroceder (←), contiene una sección '¿Alguna especificación sobre el viaje?' con el texto 'Ex. Sin gatos, no fumadores, si comida. Nos encontramos en X lugar'. En la parte inferior de la pantalla de la derecha hay un botón azul con un icono de avanzar (→) y el texto 'CREAR EL VIAJE =)'.

Fuente: elaboración propia, empleando Ionic *Framework*.

3.5.5. Información de un viaje

En la información del viaje se mostrará un resumen del viaje; se podrán agregar o eliminar peticiones de otros usuarios y terminar el viaje cuando desde el perfil del usuario creador.

Figura 20. Prototipo, pantalla información del viaje

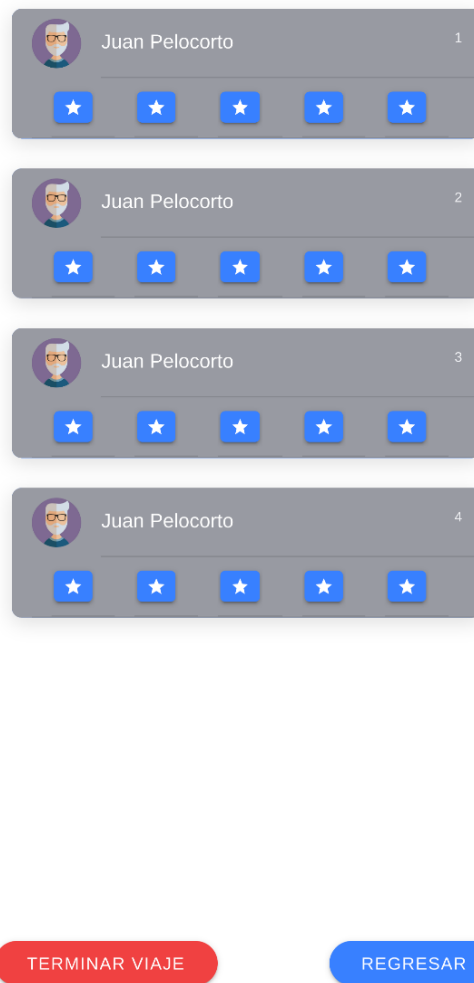


Fuente: elaboración propia, empleando Ionic *Framework*.

3.5.6. Finalización de viaje

En esta pantalla el usuario dueño de un viaje podrá finalizar un viaje y darles opcionalmente una puntuación a los usuarios que viajaron con él.

Figura 21. Prototipo, pantalla de finalización de viaje



Fuente: elaboración propia, empleando Ionic *Framework*.

3.5.7. Chat

Un chat será creado con los usuarios solamente si estos han sido aceptados en el viaje; cada chat abrirá una pantalla diferente para poderse enviar mensajes.

Figura 22. **Pantalla prototipo chat**

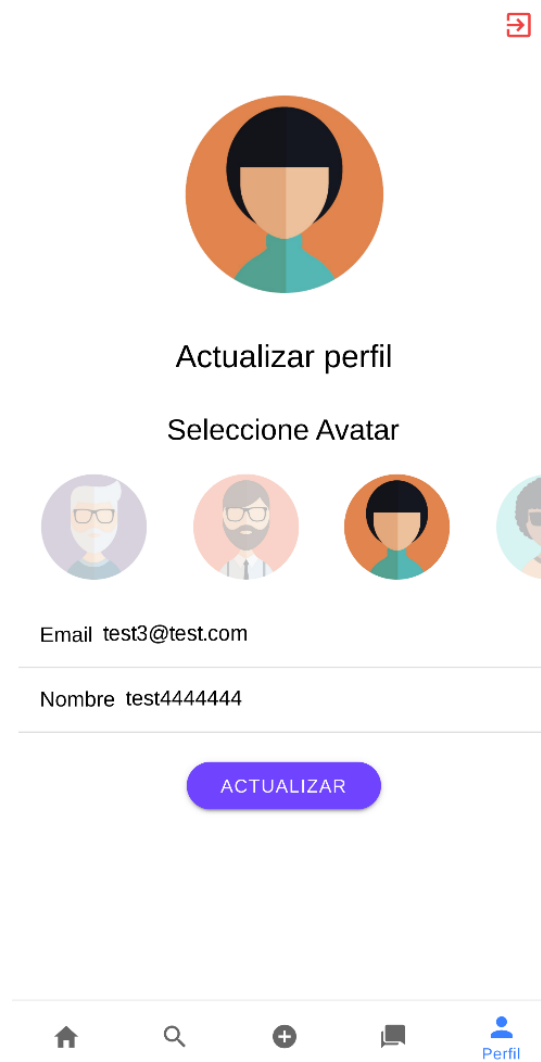


Fuente: elaboración propia, empleando Ionic *Framework*.

3.5.8. Actualizar perfil

En esta pantalla se podrá actualizar la información del usuario conectado y del lado superior derecho el botón de logout.

Figura 23. Pantalla, prototipo actualización de perfil

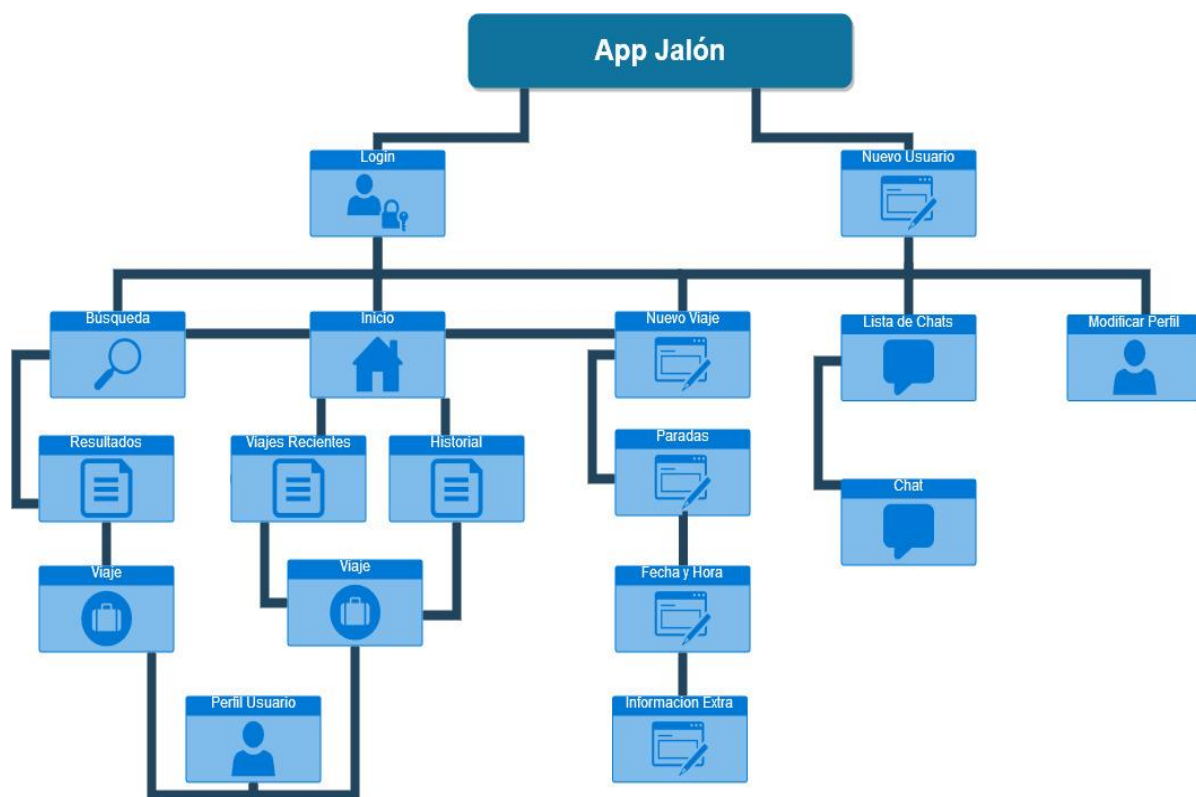


Fuente: elaboración propia, empleando Ionic *Framework*.

3.6. Mapa de navegación de la aplicación móvil

Se presenta un mapa general de la navegación a través de la aplicación móvil basado en las pantallas.

Figura 24. Esquema de navegación de la app



Fuente: elaboración propia, empleando Ionic *Framework*.

4. ANÁLISIS Y DISEÑO DE LA APLICACIÓN

4.1. Herramientas

A continuación, se describen las herramientas que se utilizaron en la elaboración de la aplicación móvil y la API RESTful.

4.1.1. Hardware

En la elaboración de la aplicación móvil como de los servicios RESTful se utilizó una computadora portátil, un móvil con sistema operativo android, la plataforma Heroku para entornos en la nube y la plataforma mLab para la base de datos.

Tabla XI. Especificaciones de la computadora de desarrollo

Característica	Computadora portátil
Sistema operativo	Debian 9 Stretch
CPU	Procesador I5 6ta generación
Memoria RAM	8 GB
Disco duro	1 TB
Herramienta de desarrollo	Nodejs v10, Ionic framwerok v4.7, Java development kit (JDK), Android Development Kit (SDK).
Modelo	Asus X55U

Fuente: elaboración propia.

Tabla XII. **Dispositivo Móvil**

Característica	Redmi Note 7
Sistema operativo	Android 9.18
CPU	Qualcomm SDM660 Snapdragon 660 octa-core 2.2GHz 64 bits
Procesador gráfico	Adreno 512
Almacenamiento interno	128 GB
Modo desarrollador	Activado

Fuente: elaboración propia.

Tabla XIII. **Heroku y mLab**

Característica	
Heroku capacidad	500 MB Dyno
mLab capacidad	500 MB

Fuente: elaboración propia.

4.1.1.1. Heroku

Es una plataforma como servicio en la nube o PaaS, que soporta distintos tipos de lenguaje de programación. Esta ejecuta las aplicaciones en contenedores virtuales que se ejecutan en un entorno de tiempo real; además, estos contenedores pueden crecer dinámicamente y permitir al desarrollador escalar la aplicación de manera instantánea.

4.1.1.2. mLab

Es un servicio de base de datos en la nube que aloja bases de datos MongoDB. Este corre en servidores provistos por Amazon, Google y Microsoft

Azure y se ha asociado con proveedores PaaS (*plataform as a service*), tales como Heroku.

4.1.2. Software

Esta sección incluye todas las herramientas de software utilizadas en el desarrollo de los servicios REST y la aplicación móvil.

Tabla XIV. **Versiones de las herramientas de software**

Software	Versión
GIT	2.11.0
Visual Studio Code	1.38.1
Robo 3t	1.3.1
Chrome Developer Tools	Chrome 79
Android Studio	3.5
Postman	7.6.0

Fuente: elaboración propia.

4.1.2.1. GIT

Un sistema de control de versiones va a servir para trabajar en equipo de una manera mucho más simple y óptima cuando se está desarrollando software.

Con Git se pueden controlar todos los cambios que se hacen en nuestra aplicación y en nuestro código y se va a tener control absoluto de todo lo que pasa en el código; puede volver atrás en el tiempo; puede abrir diferentes ramas de desarrollo, entre otros. También, proporciona un listado de los cambios (*commits*) y se puede volver atrás en el tiempo a cualquiera de esos cambios o *commits*.

4.1.2.2. Github

Es una plataforma web que permite almacenar proyectos de desarrollo de software por medio del controlador de versiones GIT. El código de los proyectos se puede almacenar de manera pública o privada.

4.1.2.3. Visual Studio Code

Visual Studio Code es un editor de programación multiplataforma desarrollado por Microsoft. Es un proyecto de software libre que se distribuye bajo la licencia MIT, aunque los ejecutables se distribuyen bajo una licencia gratuita no libre.

Desde su aparición, Visual Studio Code ha mantenido un ritmo de desarrollo muy rápido, y se publica una nueva versión a principios de cada mes (salvo en enero). Además, casi todos los meses se publican versiones secundarias que corrigen fallos de última hora. Actualmente (octubre de 2019), la última versión publicada de Visual Studio Code es la versión 1.38, publicada el 4 de septiembre de 2019.

4.1.2.4. Robo 3T

También conocido como Robomongo, Robo 3T es un gestor de bases de datos para MongoDB que nació en 2013 y que el 14 de Marzo de 2017 fue adquirido por la empresa 3T Software. Desde entonces es completamente gratuito y se llama Robo 3T y permite trabajar con un entorno gráfico muy amigable con las bases de datos MongoDB.

4.1.2.5. Chrome developer *Tools*

Es un conjunto de herramientas de creación y depuración de sitios web integrado en Google Chrome y en su versión abierta, Chromium. Las secciones que se encuentran en el panel dentro del navegador son:

- *Elements*: muestra los diferentes componentes de la página.
- *Console*: permite ejecutar diagnósticos durante el desarrollo o interactuar con el código Javascript como lo harías con la terminal de Linux.
- *Sources*: facilita depurar el código Javascript y trabajar con archivos locales.
- *Network*: ayuda a monitorizar y mejorar el rendimiento de la página.
- *Timeline*: permite la grabación y la exploración de los diferentes eventos que ocurren durante la visita a un sitio.
- *Application*: investiga todos los recursos que se cargan; entre otros, bases de datos IndexedDB o Web SQL, almacenamiento local y de sesión, cookies, caché de la app, imágenes, fuentes y hojas de estilos.
- *Security*: detecta problemas de seguridad en el código.

4.2. Requerimientos de software

A continuación, se presentan las tecnologías de software base de la aplicación móvil y del entorno de los servicios.

4.2.1. Javascript

Es un lenguaje de programación interpretado, es decir, no se compila para poder ejecutarse. Es liviano, multiplataforma ya que se puede utilizar en Windows, Linux o Mac; es imperativo, estructurado y prototipado debido a que utiliza prototipos en vez de clases para el uso de la herencia; es orientado a objetos y eventos. El uso principal es del lado cliente o *front-end*; sin embargo, ha demostrado ser un lenguaje muy potente del lado del servidor a través de NodeJs.

4.2.2. NodeJS

Es un entorno de ejecución multiplataforma para el lenguaje de programación JavaScript hecho con el motor V8 de Google Chrome; mayormente se utiliza para el desarrollo de aplicaciones del lado del servidor utilizando una arquitectura dirigida por eventos capaz de realizar procesos asíncronos. Su distribución es gratuita y su código es abierto, pudiéndose instalar por medio del gestor de paquetes NPM o desde su página principal.

4.2.3. ExpressJS

Es una infraestructura para aplicaciones de NodeJS compacta, flexible, rápida, simple y robusta. Proporciona por medio de su enrutador métodos HTTP útiles para la creación de APIs rápidas y sencillas que incluye un middleware mínimo y sólido.

Es la infraestructura más popular entre la comunidad de NodeJS, por lo que existe bastante documentación para su utilización.

4.2.4. NPM

Node Package Manager es un gestor de paquetes que permite la instalación de librerías y paquetes de forma pública y sencilla a través de una línea de código. Este repositorio cuenta con más de 800 000 paquetes.

4.2.5. MongoDB

Es una base de datos NoSQL distribuida basada en documentos de código abierto y de uso general. Los datos se guardan dinámicamente con un esquema BSON que es una especificación similar a JSON que hace que la integración con ciertas aplicaciones sea fácil y rápida.

4.2.6. Mongoose

Es un convertidor de objeto a documento, es decir, permite definir objetos con un esquema de datos fuertemente tipados y mapearlos a documentos de MongoDB

4.2.7. Angular

Es un *framework* desarrollado y mantenido por Google para el desarrollo de aplicaciones web del lado del cliente, cuya estructura está hecha con base en TypeScript, la cual es una versión avanzada de JavaScript.

Uno de los objetivos principales del *framework* es fortalecer la estructura modelo-vista-controlador que se encarga de separar el funcionamiento de una aplicación, en donde la vista es la interfaz gráfica del usuario y esta cambia de acuerdo a un controlador que gestiona el contenido en ella. De este mismo

concepto derivan las aplicaciones de una sola página o SPAs que cargan únicamente un archivo HTML en el cliente y que su contenido cambia de manera dinámica mediante la interacción con ella; siendo estas la finalidad de Angular.

4.2.8. Ionic Framework

Es un marco de trabajo gratuito y de código abierto para desarrollar aplicaciones híbridas multiplataforma que utiliza HTML5, CSS generado por SASS, Angular y se ejecuta sobre Cordova.

Mediante Cordova se puede empaquetar cualquier web; el estándar web permite crear una *web progressive app* de manera sencilla.

4.2.8.1. HTML5

Es la quinta versión del lenguaje de diseño de páginas de web o HTML. En esta versión existen etiquetas nuevas, una estandarización de soporte en multiexploradores, una cantidad considerable de API's que simplifican el desarrollo y soporte para diferentes tipos de contenido multimedia, que elimina la rigurosa dependencia de software de terceros.

4.2.8.2. SASS

Es un preprocesador para CSS y proporciona componentes que hacen que una vista HTML tenga funcionalidades y efectos; estas son muy similares a las interfaces que una aplicación nativa posee. Este preprocesador hace que la escritura de código CSS sea más sencilla para diseños potentes.

4.2.8.3. Cordova

Es un entorno de desarrollo de aplicaciones móviles que permite a los desarrolladores de software construir aplicaciones para dispositivos a partir de herramientas web tales como CSS3, HTML5 y JavaScript en vez de utilizar las herramientas específicas de cada plataforma como Android o iOS.

Cordova ofrece una gran cantidad de *plugins* que son aprovechados para las aplicaciones nativas que se desarrollan en *frameworks* como Ionic.

Algunos *plugins* que se incluyen son los de Facebook, mapas, GPS, almacenamiento interno del dispositivo móvil, entre otros.

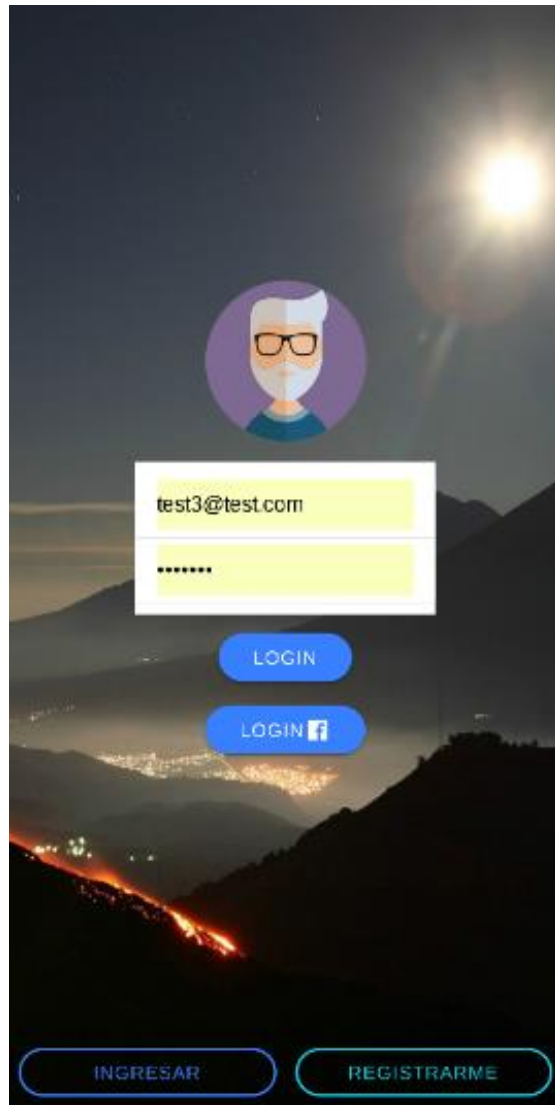
4.3. Manual de la aplicación

En la pantalla inicial de *login* consta de dos campos de ingreso de datos: el primero para ingresar el correo con el cual se encuentra registrado y el segundo para escribir la contraseña.

Seguido se encuentra el botón de *login* con el cual puede ingresar a la aplicación y debajo de este el *login* para ingresar a la aplicación por medio Facebook.

En la parte inferior se encuentra los botones para cambiar entre la pantalla de *login* y la pantalla de registro.

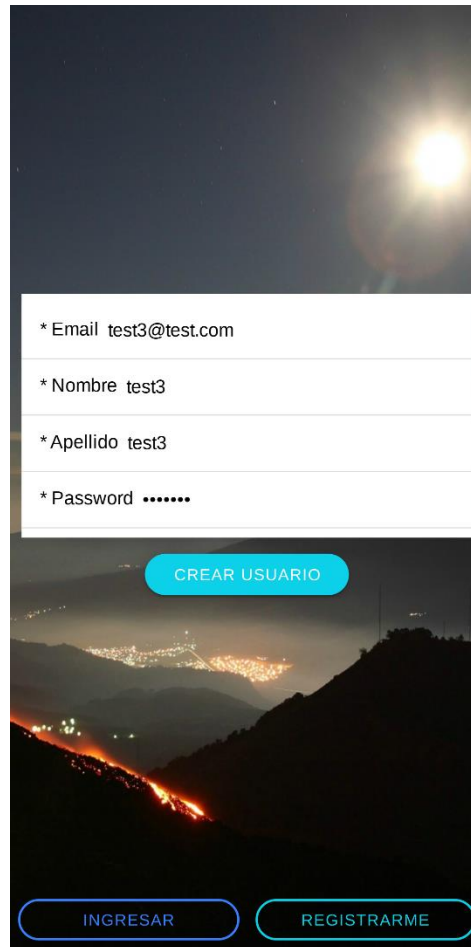
Figura 25. **Pantalla de login**



Fuente: elaboración propia, empleando Ionic *Framework*.

En la pantalla de registro se crean los nuevos usuarios con los datos básicos de correo electrónico, nombre, apellido y contraseña. Luego, se presiona el botón de crear usuario. Se ingresará automáticamente a su sesión.

Figura 26. **Pantalla de registro**



Fuente: elaboración propia, empleando Ionic *Framework*

Luego de ingresar a la aplicación se encuentra la pantalla de inicio en la cual se encuentran los botones de proponer un viaje, para crear un viaje nuevo, y buscar, para buscar un viaje y pedir aventón.

En la parte inferior se encuentran los accesos de inicio-recientes, buscar, creación de viaje, chat y perfil.

Figura 27. **Pantalla de inicio**

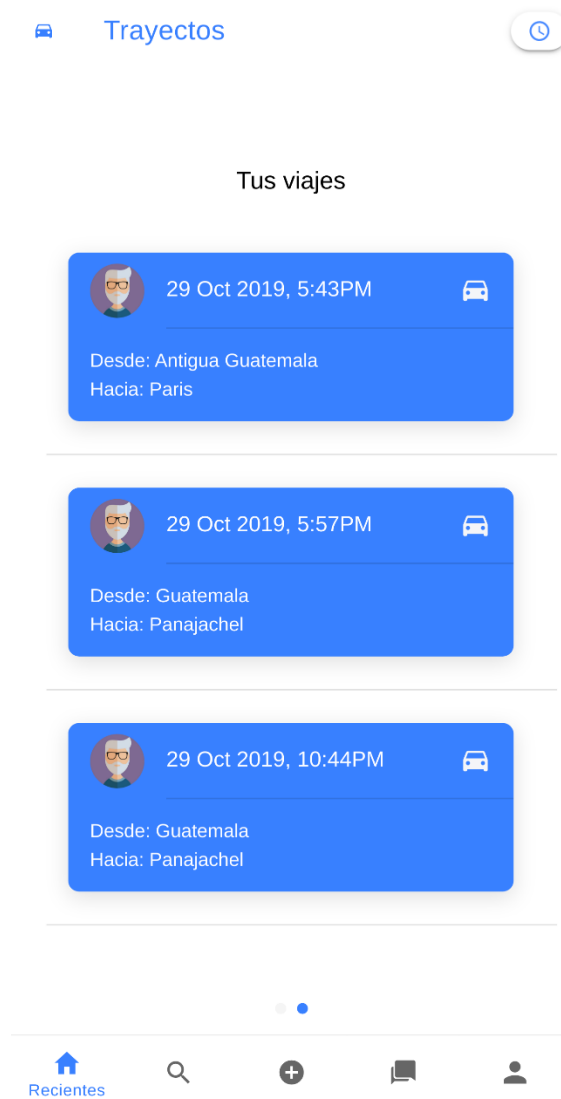


Fuente: elaboración propia, empleando Ionic *Framework*.

Desde la pantalla de inicio al deslizar hacia la izquierda se encuentran los viajes recientes del usuario. Al presionar sobre cada viaje se mostrarán sus detalles.

Cada tarjeta incluye los datos más importantes del viaje y por medio del ícono del lado superior derecho si se es pasajero o conductor.

Figura 28. **Pantalla, viajes del usuario**

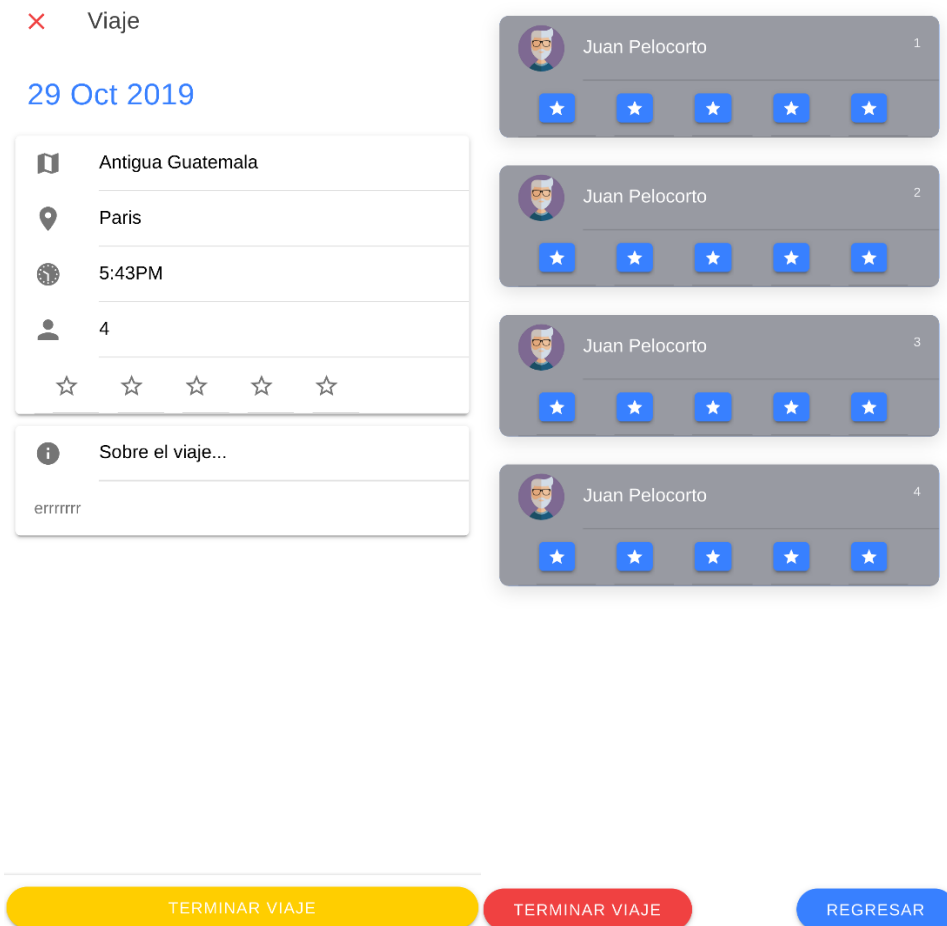


Fuente: elaboración propia, empleando Ionic *Framework*

Al seleccionar un viaje del usuario si este se encuentra vigente podrá ver la información y podrá terminarlo con el botón en la parte inferior llamado 'terminar viaje' o podrá regresar por medio del símbolo de x en la parte superior izquierda.

Al seleccionar terminar el viaje se podrá evaluar a los pasajeros por medio de una lista, no es obligatorio. Con los botones en el inferior de la pantalla se podrá terminar el viaje o regresar a la pantalla anterior.

Figura 29. **Pantalla de información de viaje y finalización**

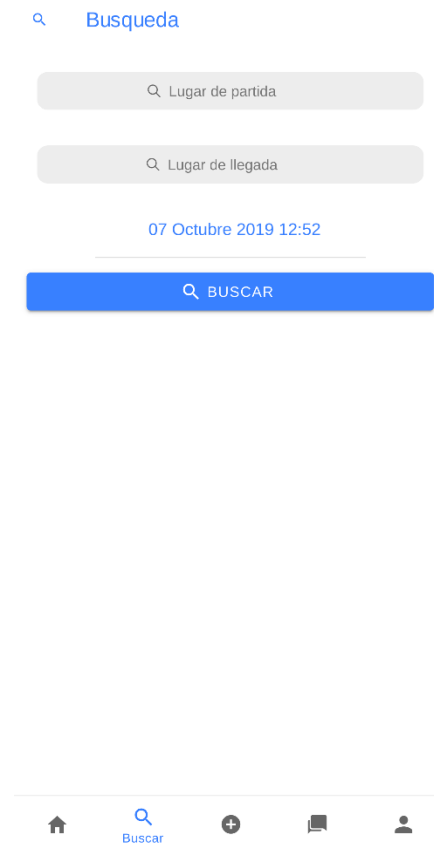


Fuente: elaboración propia, empleando Ionic *Framework*.

En la pantalla de búsqueda se podrán buscar viajes por lugar de partida, lugar de llegada y la fecha. Al buscar los lugares estos se desplegarán por medio

de una lista autocompletada y se deberá seleccionar un elemento de la lista. Luego, se deberá presionar el botón buscar para buscar las coincidencias.

Figura 30. **Pantalla de búsqueda**

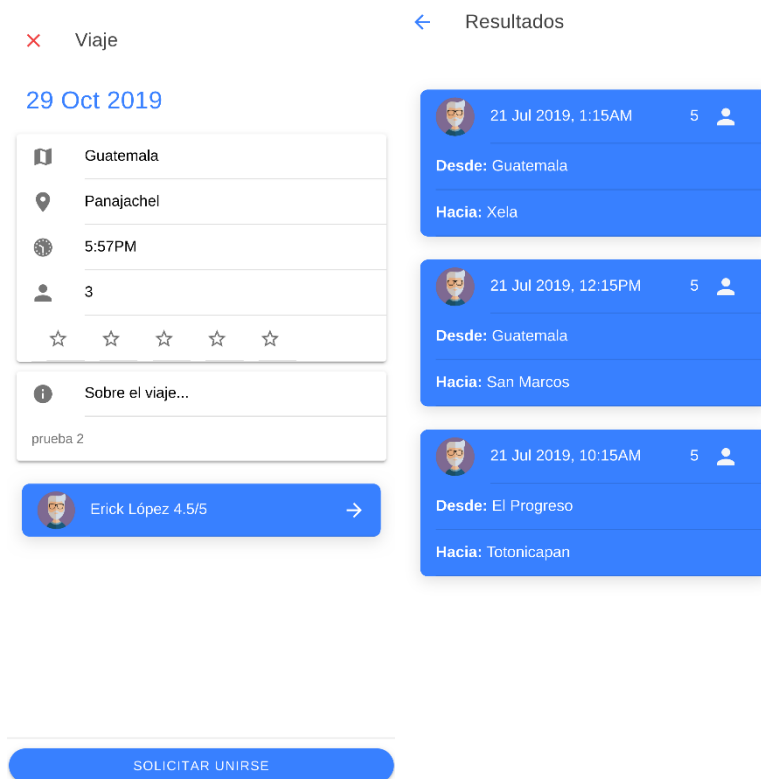


Fuente: elaboración propia, empleando Ionic *Framework*.

La pantalla de resultados se podrá elegir cualquier viaje y esto desplegará la pantalla de viaje con el cual podrá solicitar un jalón al dueño del viaje. La solicitud deberá llevar un mensaje que ingresará por medio de un modal que se desplegará al seleccionar el botón solicitar unirse en la parte inferior de la pantalla.

Al seleccionar el usuario se podrá acceder a su perfil y ver su información.

Figura 31. **Pantallas de resultados y solicitar viaje.**

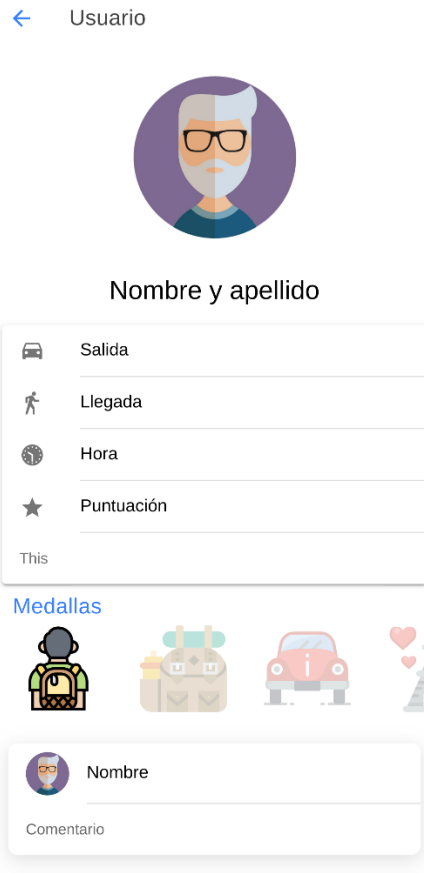


Fuente: elaboración propia, empleando Ionic *Framework*.

En la pantalla de perfil del usuario, se muestra la información general del usuario, los comentarios, medallas, como se muestra en la figura 32.

La pantalla de perfil de usuario es de visibilidad pública para otros usuarios registrados en la aplicación.

Figura 32. **Pantalla de perfil de usuario**



Fuente: elaboración propia, empleando Ionic *Framework*.

Al crear un nuevo viaje se deberá primero ingresar el lugar de partida y luego, el lugar de llegada. Ambos campos de texto tienen listas desplegables de autocompletado, se deberá elegir una opción de la lista. Luego, se deberá presionar el botón del lado inferior derecho para continuar con la creación hacia la pantalla de paradas; o si se desea salir presionar la x del lado superior izquierdo.

Figura 33. **Pantalla de creación de viaje, paso de lugar de partida y llegada**

The image shows a mobile application interface for creating a travel itinerary. At the top, there is a red 'X' icon followed by the text 'Jalón'. Below this, the question '¿De dónde sales?' is displayed above a search input field containing the text 'Ex. Antigua'. Further down, the question '¿A dónde vas?' is displayed above another search input field containing the text 'Ex. Panajachel'. At the bottom center of the screen, there is a blue circular button with a white right-pointing arrow.

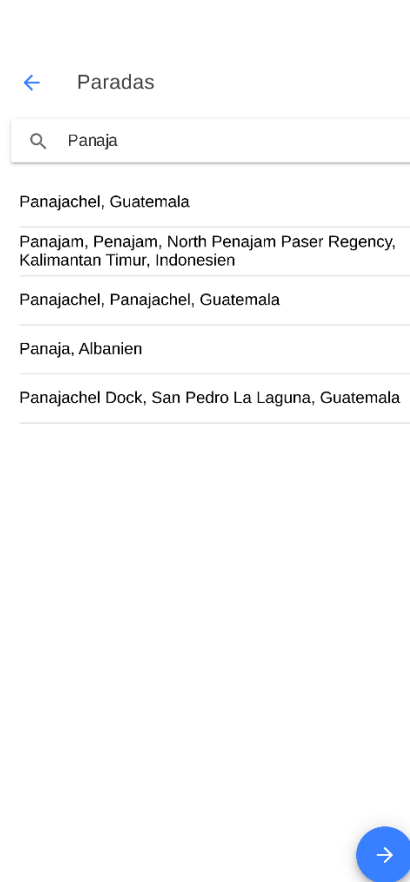
Fuente: elaboración propia, empleando Ionic *Framework*.

En la pantalla de paradas de la creación de un nuevo viaje se debe escribir en el campo de texto las paradas que se desean agregar al seleccionar un elemento de la lista desplegable.

De haber cometido un error, se podrá eliminar la parada al seleccionar el ícono rojo de basura al lado de la parada ingresada.

Luego, se deberá presionar el botón del lado inferior derecho para continuar con la creación hacia la pantalla de fecha o si se desea regresar a la pantalla anterior presionar la flecha del lado superior izquierdo.

Figura 34. **Pantalla de creación de viaje, paso de ingresar paradas**



Fuente: elaboración propia, empleando Ionic *Framework*.

Se debe elegir la fecha del viaje; luego la hora prevista de partida, la cantidad de pasajeros la cual no debe exceder de 4 por seguridad; por último, cuánto tiempo aproximadamente puede esperar a los pasajeros.

Luego, se deberá presionar el botón del lado inferior derecho para continuar con la creación hacia la pantalla de pasajeros; o si se desea regresar a la pantalla anterior presionar la flecha del lado superior izquierdo.

Figura 35. **Pantalla de creación de viaje, paso de ingresar fecha**

← Fecha

¿Cuándo sales?
Lunes, 07 Octubre 2019

¿A qué hora sales?
Hora de partida

¿Cuántos pasajeros?
1,2,3,4

¿Cuánto tiempo esperarás?
Minutos

→

Fuente: elaboración propia, empleando Ionic *Framework*.

En la pantalla final de creación de pasajeros se debe ingresar una especificación para el viaje y un monto opcional que podría cobrar en efectivo.

Luego, se deberá presionar el botón del lado inferior para finalizar con la creación del viaje; o si se desea regresar a la pantalla anterior presionar la flecha del lado superior izquierdo.

Figura 36. **Pantalla de creación de viaje, paso final de pasajeros**

← Pasajeros

¿Alguna especificación sobre el viaje ?

Ex. Sin gatos, no fumadores, sí comida. Nos encontramos en X lugar

¿Pedir alguna donación?

0

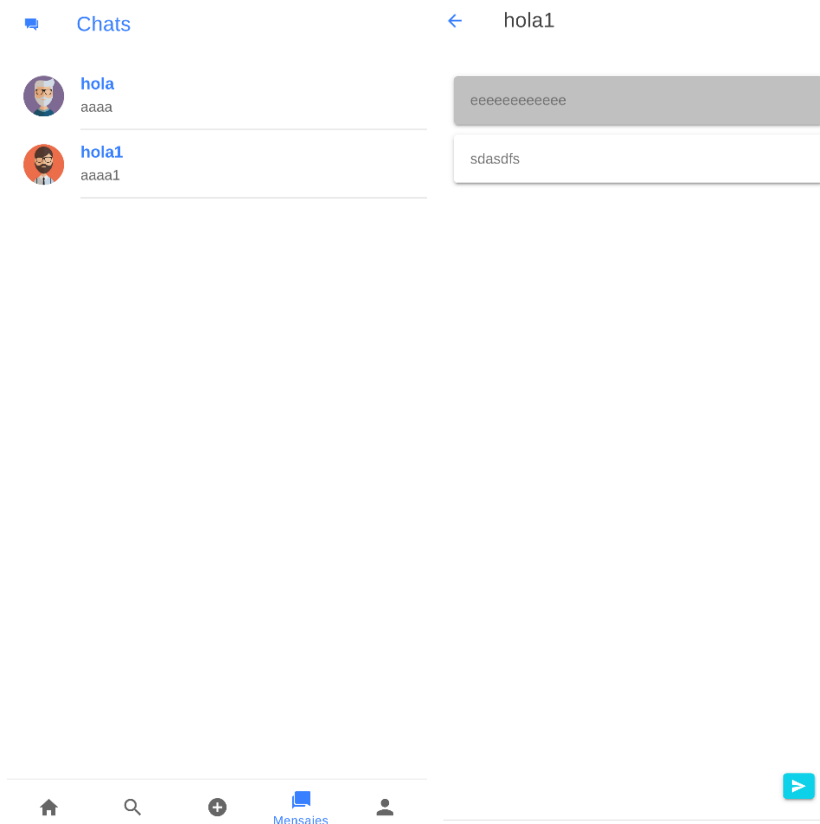
CREAR EL VIAJE =)

Fuente: elaboración propia, empleando Ionic *Framework*.

En la pantalla de chat se deberá seleccionar la conversación para ingresar a esta.

En la conversación se deberá escribir en la parte inferior; luego, enviar el mensaje por medio del icono del lado inferior derecho.

Figura 37. **Pantallas de chat**



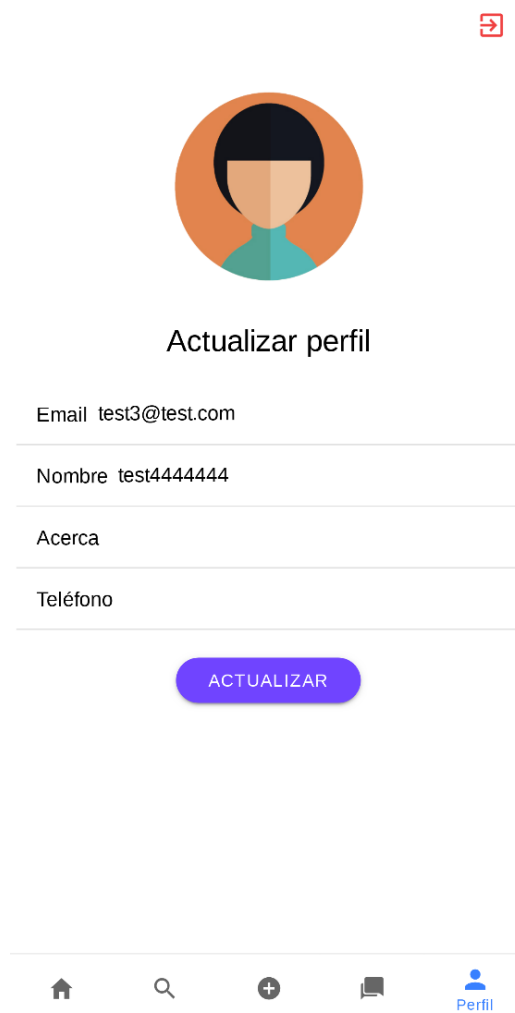
Fuente: elaboración propia, empleando Ionic *Framework*.

En la pantalla de perfil se podrán actualizar los datos del usuario al cambiar los campos y presionar el botón de actualizar.

Del lado superior derecho se encuentra el botón de salir o *logout* para cerrar la sesión de la aplicación.

Del lado superior izquierdo se encuentra el botón de ver el perfil público.

Figura 38. **Pantalla, actualizar perfil**



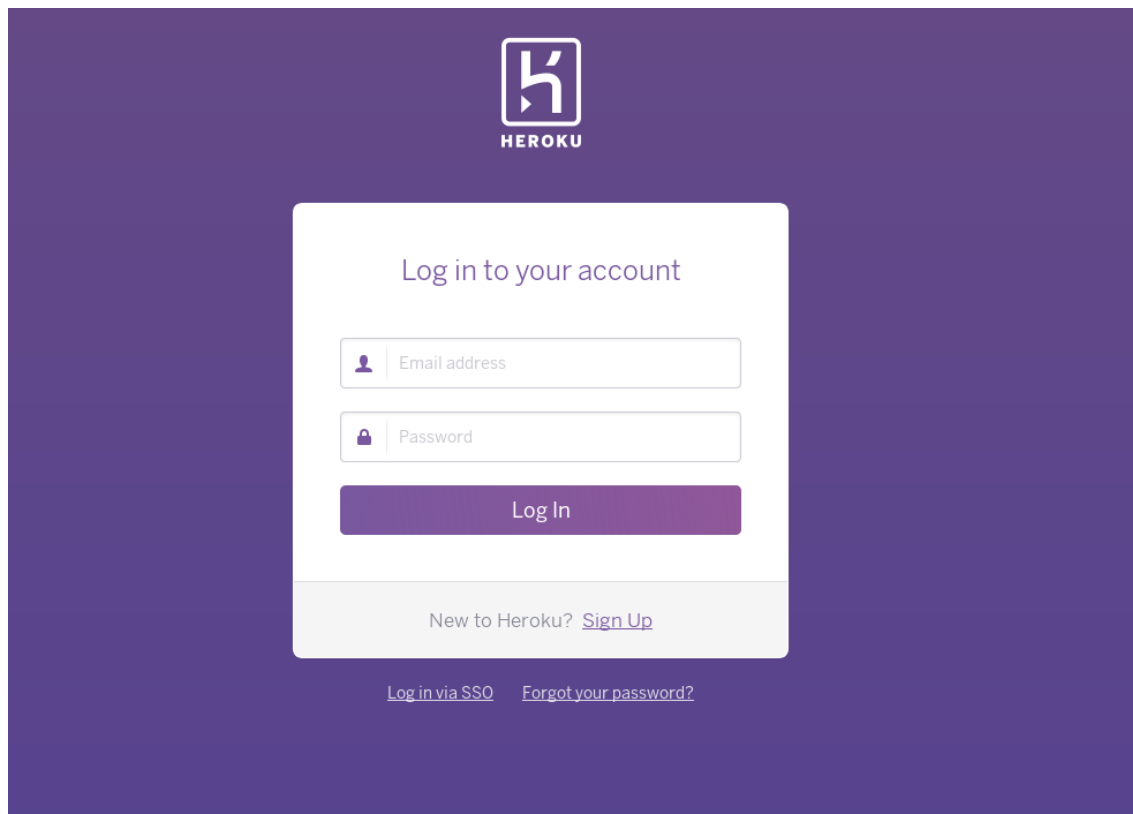
Fuente: elaboración propia, empleando Ionic *Framework*.

4.4. Manual de publicación de servicios en Heroku

A continuación, se presentan los pasos para publicar los servicios de la API en el servidor en Heroku; incluye la base de datos en mLab.

Antes de hacer las configuraciones se debe crear una cuenta en la página de www.heroku.com siguiendo los pasos que se indican, básicamente un correo válido.

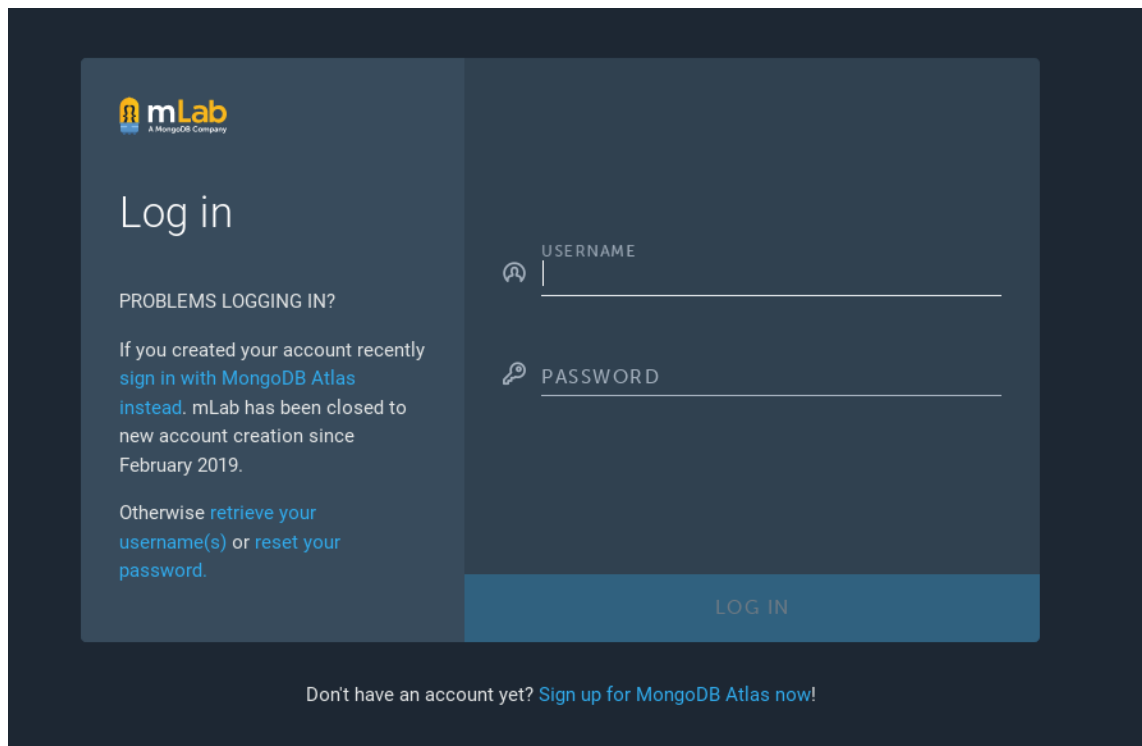
Figura 39. **Página de login de Heroku**



Fuente: *Página de login de Heroku*. www.heroku.com. Consulta: 20 de octubre de 2019.

Luego, crear una cuenta en www.mLab.com, siguiendo los sencillos pasos para crearla utilizando también un correo válido.

Figura 40. **Página de login de mLab**



Fuente: *Página de ingreso*. mlab.com. Consulta: 20 de octubre de 2019.

Al tener ambas cuentas creadas se debe instalar la interfaz de línea de comandos de Heroku; esto se hace de manera diferente dependiendo del sistema operativo; en este caso se utilizó el sistema operativo Debian y se hizo de la siguiente manera:

- `curl https://cli-assets.heroku.com/install-ubuntu.sh | sh`

En la cuenta de mLab se debe crear un *cluster* en la región que quede más cercana a nuestra ubicación; en este caso se eligió Virginia del Norte.


Figura 41. Crear un cluster en mLab


CLUSTERS > CREATE A STARTER CLUSTER


Create a Starter Cluster

Welcome to MongoDB Atlas! We've recommended some of our most popular options, but feel free to customize your cluster to your needs. For more information, check our [documentation](#).

Cloud Provider & Region AWS, N. Virginia (us-east-1) ▾














Create a **free tier cluster** by selecting a region with **FREE TIER AVAILABLE** and choosing the **M0** cluster tier below.

★ Recommended region ⓘ

NORTH AMERICA	EUROPE	ASIA
<div style="border: 1px solid green; padding: 5px;"> N. Virginia (us-east-1) ★ FREE TIER AVAILABLE</div>	<div style="border: 1px solid gray; padding: 5px;"> Ireland (eu-west-1) ★ FREE TIER AVAILABLE</div>	<div style="border: 1px solid gray; padding: 5px;"> Singapore (ap-southeast-1) ★ FREE TIER AVAILABLE</div>
<div style="border: 1px solid gray; padding: 5px;"> Oregon (us-west-2) ★ FREE TIER AVAILABLE</div>	<div style="border: 1px solid gray; padding: 5px;"> Frankfurt (eu-central-1) ★ FREE TIER AVAILABLE</div>	<div style="border: 1px solid gray; padding: 5px;"> Mumbai (ap-south-1) FREE TIER AVAILABLE</div>
AUSTRALIA		
<div style="border: 1px solid gray; padding: 5px;"> Sydney (ap-southeast-2) ★ FREE TIER AVAILABLE</div>		

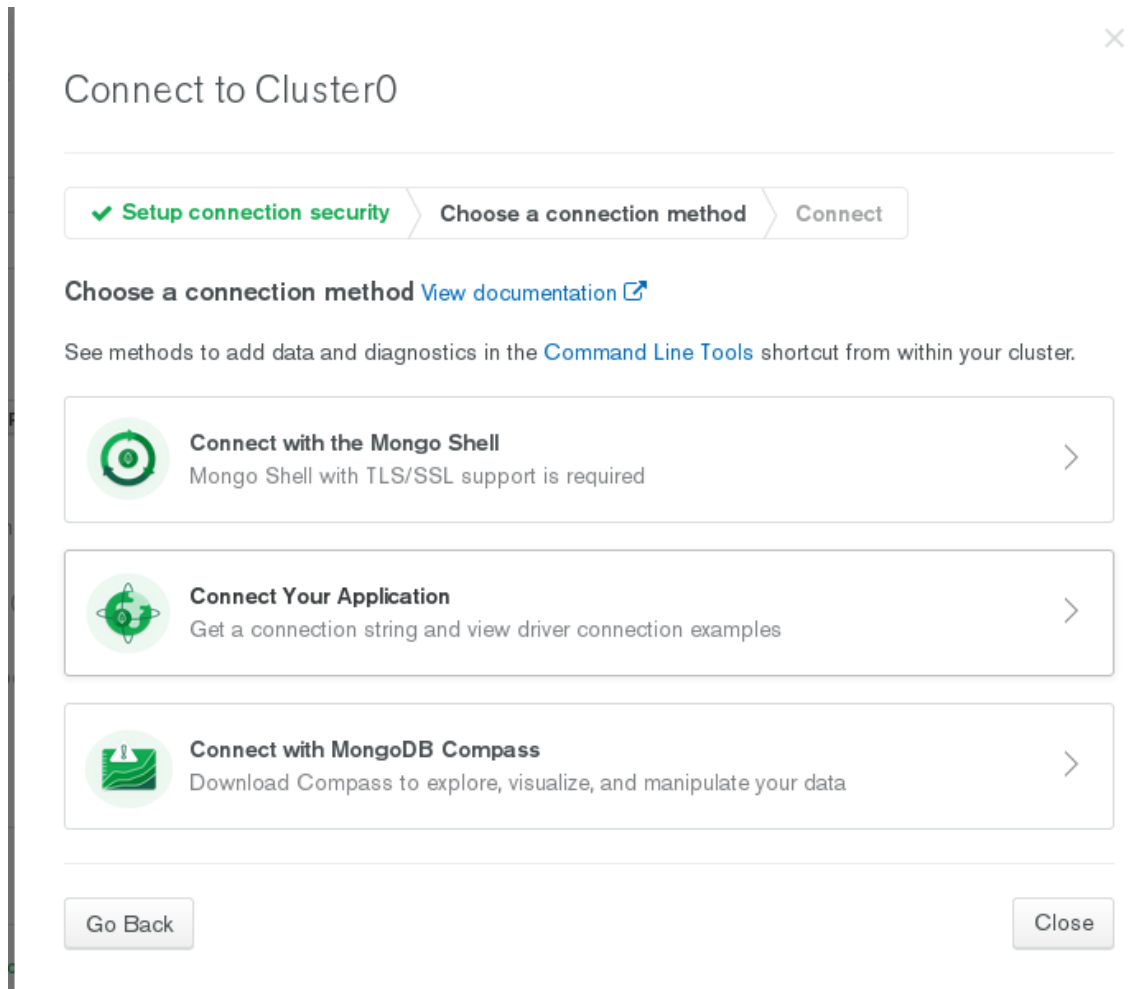
Cluster Tier M0 Sandbox (Shared RAM, 512 MB Storage) >
Encrypted

Additional Settings MongoDB 4.0, No Backup >

Fuente: *Página de región*. mlab.com. Consulta 21 de octubre de 2019.

Se debe elegir la opción de *Connect your application* para obtener la dirección de los servicios.

Figura 42. **Cómo conectarse a la aplicación de servicios**



Fuente: *Página de conexión*. mlab.com. Consulta 21 de octubre de 2019.

Luego, se obtendrá la dirección en donde se encuentra la base de datos la cual se colocará en una variable del aplicativo al subir el código a Heroku.

Esta dirección indica las credenciales y el lugar en la base de datos remota donde se guardará la información.

Figura 43. Cadena de conexión de la base de datos

Connect to Cluster0

✓ Setup connection security > ✓ Choose a connection method > Connect

1 Choose your driver version

DRIVER	VERSION
Node.js	3.0 or later

2 Add your connection string into your application code

Connection String Only Full Driver Example

```
mongodb+srv://ericksac:<password>@cluster0-bfnyt.mongodb.net/tes
```

Copy

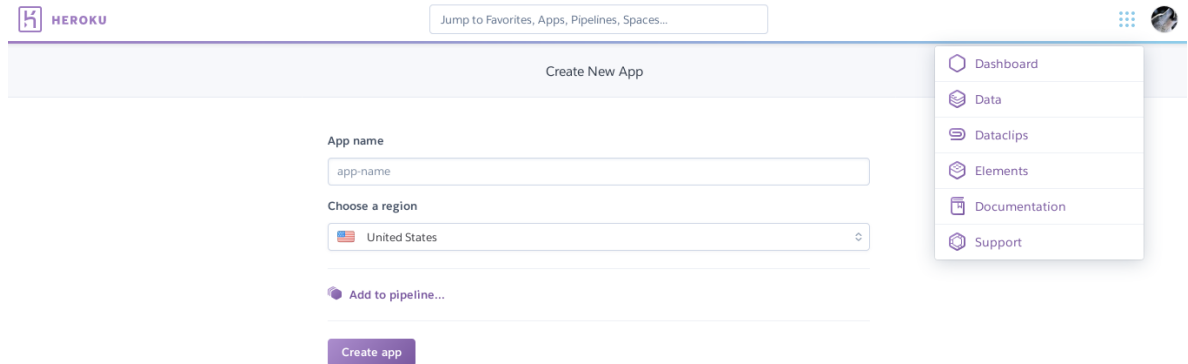
Replace <password> with the password for the ericksac user.
When entering your password, make sure that any special characters are [URL encoded](#).

Fuente: *Cadena de conexión*. mlab.com. Consulta 21 de octubre de 2019.

Ya en la página de heroku se debe crear una nueva app a través del *dashboard* o escritorio de la cuenta que se ha creado con anterioridad.

Esta aplicación es donde se sube el código del servicio REST, la aplicación se crea exclusivamente para ejecutar código javascript por medio de NodeJs.

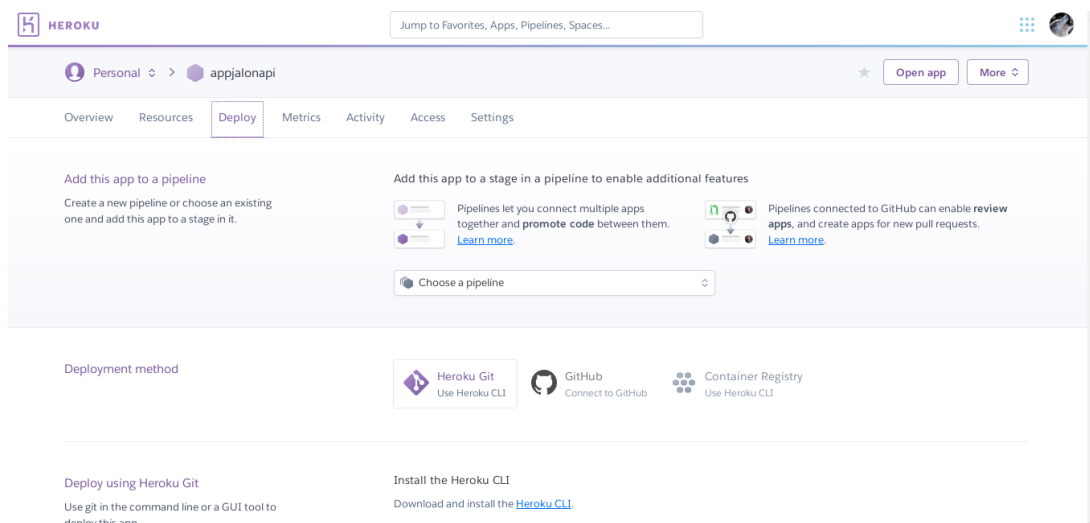
Figura 44. Crear app nueva en Heroku



Fuente: *Página de nueva aplicación*. www.heroku.com. Consulta 22 de octubre de 2019.

Luego, se deben seguir los pasos que se encuentran en la parte de *Deploy*; después elegir la opción Heroku Git.

Figura 45. Página de *deploy* de una app en Heroku



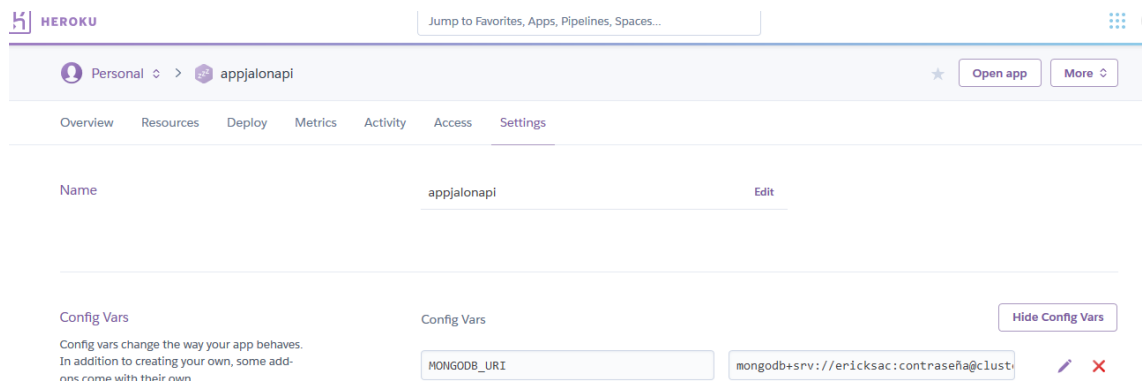
Fuente: *Página de despliegue*. www.heroku.com. Consulta 22 de octubre de 2019.

A continuación, se enlistan los comandos utilizados en la publicación por medio de la interfaz de comandos de Heroku instalada anteriormente:

- heroku login
- heroku git:clone -a appjalonapi
- git add .
- git commit -am "make it better"
- git push heroku master

Finalmente, se sustituye la dirección de mLab en las variables de configuración de Heroku en el apartado de *Settings*.

Figura 46. **Variables de configuración**



Fuente: *Variables de configuración*. www.heroku.com. Consulta: 22 de octubre de 2019.

4.5. Consideraciones de implementación

- Para subir una aplicación móvil a la Play Store que es la tienda para aplicaciones Android de Google, se debe tener una cuenta que cuesta 25 dólares.

- La aplicación móvil se desarrolló como aplicación híbrida; luego, se generó el código Android o APK para publicarla en la Play Store.
- Al generar el código Android desde Ionic a través de Cordova se debe tomar en cuenta en qué versión se hizo la implementación tendrá ya que influirá en la compatibilidad con los dispositivos móviles; en esta implementación se hizo para versiones Android 9 y menores.
- Si en un futuro los servicios de la API crecieran en concurrencia y tamaño de datos excedidos en las cuentas gratuitas, estas se deberán actualizar hacia cuentas de mayor capacidad; por lo tanto, tendrán un costo económico mayor.
- Se deben tomar en cuenta una versión 8 o superior de NodeJs en los servicios para hacer una implementación sin errores en Heroku.
- Para acceder a la documentación de la API a la dirección que proporcione Heroku se le debe adjuntar la dirección /api-docs.

CONCLUSIONES

1. La aplicación móvil facilita el acceso a la información sobre viajes compartidos, que favorece a las personas que desean moverse de un lugar a otro.
2. El uso compartido de vehículo ayuda a reducir las emisiones de dióxido de carbono en el ambiente evitando que más autos con solamente una persona circulen en la carretera.
3. La aplicación de esta solución puede expandirse en un futuro a una aplicación web basada en el código fuente de la aplicación móvil o creando una nueva, ya que la arquitectura implementada lo permite a través de su API.
4. La solución propuesta es la primera en ofrecer el servicio de viajes compartidos de manera gratuita; es de gran utilidad a la población que desee hacer de su uso.
5. Las herramientas de software así como el software con el que se desarrollaron la API y la aplicación móvil son instrumentos poderosos y de última generación que pueden ser utilizados para desarrollar mejores sistemas y mejorar la experiencia para los usuarios como para los desarrolladores.

RECOMENDACIONES

1. Es importante proponer soluciones basadas en software utilizando el modelo de aceptación tecnológica y la ludificación ya que así se aumentarán las posibilidades que la solución sea exitosa con los usuarios.
2. Hacer uso de los viajes compartidos en automóvil ya que es una manera de ahorrar en gastos, viajar seguro, conocer nuevas personas y reducir la emisión de dióxido de carbono en la atmósfera.
3. Apoyar a iniciativas e ideas cuyo objetivo sea resolver problemas locales a través de la tecnología de las aplicaciones móviles.
4. Al crear una aplicación móvil es necesario tomar el tiempo para preparar el diseño y la arquitectura ya que debe quedar abierto a mejoras o nuevas implementaciones ya que los sistemas siempre crecen con el tiempo.
5. Se insta a estudiantes y profesionales de la informática usar las soluciones técnicas aquí planteadas para que les sea de ayuda en desarrollar nuevas herramientas para solucionar otros problemas que puedan aquejar a su sociedad.

BIBLIOGRAFÍA

1. AMIR, Bilal y RALPH, Paul. *Proposing a Theory of Gamification Effectiveness*. [en línea]. <https://www.researchgate.net/profile/Paul_Ralph/publication/266656865_Proposing_a_theory_of_gamification_effectiveness/links/587bd21608ae9275d4e012e4/Proposing-a-theory-of-gamification-effectiveness.pdf> [Consulta: 19 de mayo de 2019].
2. ANDRASON, Ariana. *MEAN Stack Feature And Architecture*. [en línea]. <<https://medium.com/@ariana10andrason/mean-stack-feature-and-architecture-e70409d18337>> [Consulta: 30 de junio de 2019].
3. CHUTTUR, Mohammad. *Overview of the Technology Acceptance Model: Origins, Developments and Future Directions*. [en línea]. <<https://test.woland.me/pdfjs/pdf.js/test/pdfs/TAMReview.pdf>> [Consulta: 10 de abril de 2019].
4. DAVIS, Fred D. *Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology*. [en línea]. <https://www.jstor.org/stable/249008?origin=crossref&seq=1#page_scan_tab_contents> [Consulta: 10 de abril de 2019].
5. GONZÁLEZ, Diego Germán. *Para qué sirven las Chrome DevTools*. [en línea]. <<https://www.linuxadictos.com/para-que-sirven-las-chrome-devtools.html>> [Consulta: 20 de octubre de 2019].

6. IGLESIAS, Tomás. *Gestor de bases de datos para MongoDB*. [en línea]. <<http://programadordocumental.es/robo-3t-gestor-de-bases-de-datos-para-mongodb/>> [Consulta: 15 de agosto de 2019].
7. LEHENCHUK, Yaroslav. *Get the Most in Life and Work with Gamification*. [en línea] <<https://producttribe.com/productivity/life-gamification>> [Consulta: 19 de mayo de 2019].
8. ORANTES CÓRDOVA, Roxana. *Lo que no es cuantificado no puede ser administrado*. Revista *Perspectiva*. [en línea]. <<https://www.perspectiva.com.gt/noticias/lo-que-no-es-cuantificado-no-puede-ser-administrado/>> [Consulta: 10 de mayo de 2019].
9. PASZNIUK, Rodrigo. *¿Qué es Json Web Token (JWT)?* [en línea]. <<https://www.programacion.com.py/varios/que-es-json-web-token-jwt>> [Consulta: 30 de junio de 2019].
10. ROBLES, Victor. *¿Qué es Git y para qué sirve?* [en línea]. <<https://victorroblesweb.es/2018/04/28/que-es-git-y-para-que-sirve/>> [Consulta: 15 de agosto de 2019].
11. SÁNCHEZ, Javier. *El impacto medioambiental del dióxido de carbono*. [en línea]. <<https://www.ecologiaverde.com/el-impacto-medioambiental-del-dioxido-de-carbono-1334.html>> [Consulta: 11 de mayo de 2019].

ANEXO

Anexo 1. Fuentes del sistema completo en GitHub

El código de la aplicación híbrida y los servicios se encuentran en esta dirección: <https://github.com/ericksac/appJalon>.

ericksac / appJalon Private

Watch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Security Insights Settings

Repositorio de la aplicación app jalón tesis 2019 Edit

Manage topics

10 commits 1 branch 0 releases

⚠ We found a potential security vulnerability in one of your dependencies.
Only the owner of this repository can see this message. View security alert

Branch: master New pull request Create new file Upload files Find file Clone or download

File	Description	Latest commit
ericksac	Adicion de aplicacion movil	2470d3b 29 days ago
AppJalon	Adicion de aplicacion movil	29 days ago
AppJalonAPI	Cambios en Cors, swagger y config	29 days ago
Documentación	Servicios de login y de creación de usuarios	7 months ago
.directory	Adicion de aplicacion movil	29 days ago
.gitignore	Initial commit	8 months ago
README.md	Create README.md	8 months ago

README.md

appJalon

Repositorio de la aplicación app jalón tesis 2019

Fuente: Página de repositorio. www.github.com. Consulta: 28 de octubre de 2019.

