



## **PROYECTO DE ESTANCIA II**

# **SOFTWARE DE APOYO TRANSEÚNTE DENTRO DE ZONAS URBANAS**

**PRESENTADO POR:**

*Ángel Alan González Escalona*

**ASESORADO POR:**

*DR. PORFIRIO ESPEJEL FLORES*

## RESUMEN

El uso de la tecnología y las aplicaciones móviles está cambiando el entorno en el que nos desarrollamos y convivimos en varios ámbitos, incluida la forma en que nos transportarnos y viajamos de un lugar a otro.

Existen servicios de transporte público con rutas determinadas y tarifas fijas, llamados combis, pero presenta un gran inconveniente ya que no proporcionan mucha información que puede llegar a ser relevante para las personas que hacen uso del servicio día con día, haciendo que se vean opacadas por aplicaciones como Citymapper y Uber, que gracias al uso de la tecnología pueden llegar al público y facilitan el manejo de la información. Sin embargo, estas empresas tienen delimitaciones geográficas en donde solo funcionan siempre y cuando te encuentres en rutas ya fueron definidas como lo es el Tuzobus, lo que hace imposible que se puedan aplicar en otras zonas y municipios los cuales no se tenga una información clara de los recorridos que hacen los transportes.

Es por esto que se desarrollará una aplicación móvil para el sistema de transporte público más enfocado a las combis, el cual permita incorporar algunos beneficios, derivados de la ubicación en tiempo real del transporte y así brindar un servicio más actualizado y óptimo en donde a través de un Smartphone se pueda visualizar el tiempo de llegada y la distancia que hay entre la combi a la parada.

Cabe mencionar que el alcance del prototipo no abarcara todas las rutas del municipio de Tizayuca si no se centrara en una ruta específica y en un solo vehículo del transporte público ya que no se cuenta con el tiempo para desarrollar el sistema en más de una ruta y un vehículo, así mismo a través del tramo definido se ira notificando la ubicación del vehículo tanto al conductor como al pasajero.

El sistema contara con 2 pantallas principales una para el conductor y la otra para el pasajero en la cual la pantalla de conductor tendrá las opciones de compartir su ubicación y actualizar en tiempo real, así como poder visualizar la ruta que le toca recorrer y contar con un botón para desconectarse en el momento que termine su jornada; De igual manera las opciones que tendrá el pasajero es poder visualizar tanto su ubicación como la ubicación del conductor para así poder monitorear y saber en dónde se encuentra exactamente, también podrá solicitar por medio de un botón la distancia y el tiempo de llegada del vehículo a su localización.

En un mundo cada vez más interconectado, donde la tecnología avanza día con día y se integra en la vida cotidiana de las personas, nuestra relación con lo que nos rodea cambia. Hoy, gracias al uso de la tecnología, Internet y las aplicaciones móviles se nos facilitan tareas y procesos.



## CONTENIDO

RESUMEN.....	2
ÍNDICE DE FIGURAS.....	4
ÍNDICE DE TABLAS .....	4
MARCO CONTEXTUAL .....	5
INTRODUCCIÓN.....	5
CAPÍTULO 1 .....	6
<i>Definición Del Problema A Resolver.....</i>	6
1.1 Planteamiento Del Problema.....	6
1.1.2 Pregunta De Investigación .....	6
1.1.3 Solución Propuesta .....	6
1.2 Objetivos .....	7
1.2.1 Objetivo General .....	7
1.2.2 Objetivos Específicos .....	7
1.3 Cronograma .....	8
CAPITULO 2 .....	9
<i>Marco Teórico .....</i>	9
2.1 La Importancia Del Transporte Público.....	9
2.2 Antecedentes.....	9
2.2.1 Aplicaciones Móviles para el Transporte Público .....	9
2.2.2 Bases teóricas.....	11
2.2.3 Bases Legales.....	11
2.2.4 Variables .....	12
CAPITULO 3 .....	13
<i>Desarrollo De Actividades .....</i>	13
3.1 Salud Del Proyecto .....	13
3.2 Desarrollo de la aplicación.....	15
3.2.1 Pantallas Principales .....	15
3.2.2 Código .....	20
CAPITULO 4 .....	38



Conclusiones Y Perspectivas .....	38
BIBLIOGRAFÍA.....	42

## ÍNDICE DE FIGURAS

Figura 1 Cronograma.....	8
Figura 2 Pantalla Principal .....	15
Figura 2.1 Pantalla de Acceso.....	15
Figura 2.2 Pantalla de Registro del Pasajero .....	15
Figura 2.3 Pantalla de Login .....	15
Figura 2.4 Pantalla de Registro del Conductor .....	16
Figura 2.5 Pantalla de Navegación del Conductor .....	16
Figura 2.6 Pantalla de Navegación del Pasajero .....	16
Figura 2.7 Pantalla de Notificación al usuario .....	16
Figura 3 Base de Datos FireBase .....	17
Figura 4 Creación de carpetas y organización del código .....	18
Figura 5 Android Studio-Mapa conductor .....	19

## ÍNDICE DE TABLAS

Tabla 1 Tabla de Avance.....	14
Tabla 2 Tabla de Código Para Registrar Conductor.....	22
Tabla 2.1 Tabla de Código Para Registrar Pasajero .....	25
Tabla 2.2 Tabla de Código Para Iniciar Cesión.....	28
Tabla 2.3 Tabla de Código Para la Pantalla de Navegación del Conductor .....	37

## MARCO CONTEXTUAL

### INTRODUCCIÓN

A través de una problemática elaborada por el alumno, se desarrollará un prototipo el cual será la solución real al problema, así mismo el planteamiento del problema contará con los procesos que se llevarán a cabo para la construcción del sistema, en el cual, de acuerdo al desarrollo se adjuntarán los objetivos, marco teórico y el cronograma de actividades.

Esta aplicación va enfocada a las rutas de transporte público del municipio de Tizayuca, pero debido a que no se cuenta con el tiempo para abarcar todas estas rutas se enfocará esencialmente en un tramo y trayecto de una ruta ya establecida en donde se proporcionará la información del vehículo, es decir que la aplicación mostrará los datos de una sola ruta del transporte público.

El transporte público es un servicio primordial e importante para la comunidad, ya que es un servicio utilizado por miles de personas en el municipio de Tizayuca Hidalgo. Por lo tanto, se brindará un prototipo de software en donde se apliquen los conocimientos y habilidades adquiridos en los anteriores cuatrimestres, el sistema busca mejorar la calidad del servicio informando a los usuarios el tiempo y la distancia del transporte público al posicionarse en la parada.

En este caso, la aplicación informará la ubicación en tiempo real del vehículo, permitiendo a los usuarios visualizar en donde se encuentra posicionado el transporte de acuerdo a la ruta que recorre e indicar el tiempo y distancia del transporte a la parada en donde nos encontramos posicionados.

La aplicación es una herramienta digital que brindará información para que las personas puedan movilizarse de una mejor manera permitiendo ofrecer un mejor servicio ya que se podrá monitorear el transporte y de ahí se obtendrá el conjunto de datos con los cuales trabajará la aplicación. Así mismo les brindará la opción de monitorear en tiempo real el vehículo y facilitará el manejo de la información de ahí la importancia del desarrollo de este prototipo.

Existen personas que necesitan conocer el tiempo y la distancia del vehículo a las paradas, ya que hacen uso de este medio. Por este motivo se diseñará una aplicación la cual brinde esta información y permita satisfacer al usuario. Para lograrlo, se utilizará el entorno de desarrollo integrado (IDE) de Android Studio, que es el software oficial utilizado para desarrollar aplicaciones de Android y utiliza un lenguaje de programación orientado a objetos.

## CAPÍTULO 1

### *Definición Del Problema A Resolver*

#### 1.1 Planteamiento Del Problema

Actualmente, muchas ciudades o identidades no tienen aplicaciones que brinden información básica sobre transporte público y el tiempo de llegada. Esto plantea un problema para los residentes ya que el transporte no brinda ningún tipo de información que pueda ser de utilidad, ocasionado que se desconozca hora de llegada y la distancia del vehículo a la parada, Mencionado lo anterior, Tizayuca es uno de esos municipios que no cuentan con una aplicación la cual proporcione esa información, ocasionado que los habitantes no puedan transitar e inspeccionar su tiempo libremente, es por ello que pienso que es de suma importancia tener algún medio el cual indique esa información, para poder optimizar y mejorar de manera notoria el servicio de transporte público, y así este tenga un mejor manejo de la información sobre el tiempo de llegada y la distancia que hay entre el vehículo y el pasajero.

#### 1.1.2 Pregunta De Investigación

¿Qué tipo de sistema se necesita realizar para monitorear la ubicación del transporte público en tiempo real y permitir brindar datos de esto mismo?

#### 1.1.3 Solución Propuesta

Informar a los habitantes de Tizayuca Hidalgo cual es el tiempo de llegada y la distancia que hay entre el transporte público y la parada esto a través de una aplicación la cual permitirá visualizar dicha información por medio de la constante monitorización de la ubicación en tiempo real; Esta aplicación será un prototipo funcional y habrá 2 vistas, una para el cliente y otra para el conductor. En la vista del cliente podremos visualizar la ubicación del transporte público y a qué distancia se encuentra de la del pasajero, así mismo en la vista de conductor se visualizará la ruta y se compartirá la ubicación en tiempo real.

Así mismo se especifican que solo va englobada a una sola ruta y un solo vehículo ya que por cuestiones de tiempos es complicado desarrollar la aplicación para todas las rutas y todos los vehículos de transporte público del municipio, es por ello que se trabajara sobre un trayecto ya definido el cual recorre el vehículo.

La aplicación busca enfocarse en dispositivos móviles para brindar un mejor acceso e interacción, de igual manera el software contará con una interfaz fácil de operar y visualmente atractiva para que su uso sea intuitivo. Dado que actualmente no existe una aplicación que brinde la información anterior, tengo la intención de desarrollar esta aplicación como un recurso de ayuda para los usuarios del transporte público en Tizayuca, que tiene como objetivo informar a las personas.

## 1.2 Objetivos

### 1.2.1 Objetivo General

Brindar Información sobre la distancia y el tiempo que hay entre la parada y el transporte público de una sola ruta del municipio de Tizayuca hidalgo; con el fin de ayudar a los habitantes para que estos puedan movilizar y administrar sus tiempos de mejor manera, todo esto por medio de un prototipo el cual mostrara la información de una de las tantas rutas y un vehículo del transporte público, en el cual se visualizara la información de manera rápida por medio de la geolocalización en un dispositivo móvil, así mismo se desarrollara el proyecto en un lapso no mayor a 2 meses.

### 1.2.2 Objetivos Específicos

1. Diseñar las pantallas principales de la aplicación, la cual será la pantalla de bienvenida ,selección de login o registro la cual permitirá escoger entre ambas opciones y la otra vista será una pantalla de estas misma las cuales dará acceso a las funciones de acceder y registra, de la aplicación; Esta vista se desarrollará en un lapso no mayor a 2 semanas.
2. Crear una base de datos para almacenar la información introducirá en los formularios (Login, Registro) y así poder verificar la información para acceder a la aplicación, en una semana.
3. Asegurar la calidad del código para realizar las funciones de la aplicación, y establecer un orden en la sintaxis al programar, para evitar errores en la ejecución de la aplicación, todo esto en un lapso no mayor a una semana.
4. Diseñar la plantilla del mapa del municipio de Tizayuca para la pantalla del conductor y establecer sus funciones las cuales son visualizar la ruta en el trayecto del viaje y compartir la ubicación en tiempo real en un lapso no mayor a 2 semanas.
5. Diseñar una plantilla del mapa municipal para la pantalla del pasajero y establecer las funciones las cuales son visualizar el vehículo y obtener la distancia que hay entre el transporte público y el pasajero, al igual que el tiempo de llegada a esta misma, en un lapso no mayor a 2 semanas.
6. Ejecutar el prototipo de la aplicación para aplicar pruebas y verificar el correcto funcionamiento y observar si presenta algún tipo de error mientras está funcionando, para poder corregirlo a tiempo en una semana.

### 1.3 Cronograma

Es una representación gráfica, ordenada y esquemática de eventos próximos. A menudo se utiliza como herramienta de gestión de proyectos, roles o eventos. Su uso es muy común en diversos campos, especialmente en el ámbito laboral y empresarial.

Al permitir una visualización ordenada de eventos futuros, la programación es ideal para planificar, monitorear y pronosticar. Por lo tanto, son una herramienta común y popular. Su actuación visual puede tener muchas formas y estilos, pero suele constar de un conjunto de espacios delimitados correspondientes al tiempo: días, horas, etc., algunos de los cuales son más o menos detallados.

Es por eso que se hace uso de esta herramienta para representar la actividades a realizar en las 8 semanas así mismo se menciona que se entregara un prototipo el cual cumpla con los objetivos especificados y presenta la función la cual consiste en notificar el tiempo de llegada del transporte público y la distancia del el mismo a la parada.

N.	Actividades	SEMANA 1 DEL 15-10-2021 AL 21-10-2021	SEMANA 2 DEL 22-10-2021 AL 28-10-2021	SEMANA 3 DEL 29-10-2021 AL 04-11-2021	SEMANA 4 DEL 29-10-2021 AL 04-11-2021	SEMANA 5 DEL 29-10-2021 AL 04-11-2021	SEMANA 6 DEL 29-10-2021 AL 04-11-2021	SEMANA 7 DEL 29-10-2021 AL 04-11-2021	SEMANA 8 DEL 29-10-2021 AL 04-11-2021
1	Diseño principal de pantalla de Login y de registro para que el usuario pueda hacer uso de la aplicación		Pantallas						
2	Creación de base de datos la cual permitirá almacenar la información del usuario			Creacion de BD					
3	Revisión y corrección de sintaxis en el código, esto con la finalidad de evitar errores				Diseño de vista del conductor				
4	Diseño de vista para el conductor en el cual sus principales funciones es visualizar la ruta y compartir su ubicación					Diseño de vista para el pasajero			
5	Diseño de vista para el pasajero el cual sus principales funciones es visualizar la distancia del vehículo y saber el tiempo de llegada de este								Aplicación de pruebas
6	Ejecución del prototipo para realizar pruebas y verificar el correcto funcionamiento								

Figura 1 Cronograma de Actividades



## CAPITULO 2

### *Marco Teórico*

#### 2.1 La Importancia Del Transporte Público

En las acciones propuestas para controlar la propagación de la epidemia, la mayoría de los gobiernos han adoptado medidas de distanciamiento social para restringir el movimiento de ciudadanos en un estado de emergencia representado por la rápida propagación del virus. Sin embargo, algunos servicios se han declarado esenciales. Por ejemplo, priorizar los servicios de transporte público ya que son aquellos que permiten asegurar que los trabajadores de otros sectores importantes puedan viajar hacia y desplazarse de un lugar a otro;

Esto con el fin de movilizar la economía ya que múltiples trabajos dependen de la movilización a través del transporte público algunos ejemplos de esto son los mercados, farmacias, bancos, instituciones gubernamentales y escuelas, que es este último fue en donde mayor impacto se tuvo ya que los establecimientos dependían de la movilización de los alumnos y gran parte de los traslados eran alumnos que viajaban para ir o regresar a las instituciones educativas.

Una vez más el Covid-19 nos demuestra la suma importancia del transporte público ya que proporciona bienestar a las personas, porque permite el acceso a bienes, servicios y empleos. Para todos aquellos que no cuentan o no tienen acceso a un vehículo propio o privado: El sistema de transporte público es sin duda indispensable. Decenas de trabajadores de la salud y otros sectores importantes necesitan servicios de transporte confiables para llegar al trabajo.

#### 2.2 Antecedentes

##### 2.2.1 Aplicaciones Móviles para el Transporte Público

Existen algunas aplicaciones que están en el mercado actual, cabe destacar que estas aplicaciones solo cuentan con los transportes que se encuentran en la ciudad de México y algunos Estados y ciudades, pero se han establecido rutas, como en el caso del Tuzobús lo que imposibilita aplicarlas en identidades

Entre la aplicación que se centran en el transporte público las más utilizadas y mencionada por los usuarios son:

##### **Moovit**

(Moovit, 2021) Es una aplicación de transporte público y un servicio de mapeo desarrollado por la compañía de software israelí Moovit Inc. La app incluye planificación de viajes, horarios de salida y llegada en vivo, itinerarios actualizados,

mapas de paradas y estaciones locales, alertas de servicios, y avisos que podrían afectar algún viaje. Actualmente está disponible para dispositivos móviles Android, iOS, y Windows Phone y tiene una versión web. Moovit es la aplicación de transporte público más utilizada en el mundo, con más de 140 millones de usuarios en todo el mundo, y presente más de 2000 ciudades, en 80 países, y 44 idiomas.

Pero estas rutas no cuentan las rutas de los municipios ajenos ya que no se tiene un control, Por lo que aun siendo una aplicación que cuenta con respaldo del gobierno es insuficiente ya que ni siquiera puede proporcionar datos tan básicos como lo son las rutas del municipio de Tizayuca, Así mismo para hacer uso de esta aplicación debe de conocer los componentes básicos ya que sin la aplicación te confundirá ya que requiere de un manual.

### Citymapper

(Citymapper, 2021) Es una aplicación móvil de web mapping y transporte público. Integra información de todos los medios de transporte urbano, en especial transporte público; aunque también añade opciones para ir a pie, en bici, o en sistemas de transporte compartido (bicis, motos, coches o incluso patinetes). Funciona de manera gratuita como aplicación móvil o como sitio web en ordenadores.

Esta aplicación estaba enfocada en dispositivos iOS y solo se enfocaba en rutas de bus cerca de Londres, pero tiene una limitación porque solo se aplica a 30 ciudades. Su función principal era que los usuarios pueden utilizar el transporte público para planificar sus rutas,

A lo largo de los años ha cambiado la forma y el algoritmo, pero todavía hay deficiencias las cuales las hacen imposibles de adaptar en municipios o ciudades más pequeñas es por ello que se necesita una aplicación propia del municipio de Tizayuca Hidalgo la cual satisfaga esas necesidades.

### Uber

(¿Qué es Uber?, 2021) Uber es una empresa privada cuya principal actividad es la de transportar personas mediante un servicio de transporte seguro y responsable y funciona únicamente en los teléfonos inteligentes, es en esa plataforma donde los conductores y usuarios pueden comunicarse abiertamente y, de hecho, los usuarios tienen la facultad de poder ver el recorrido que toma el auto, así sabrán si el servicio será prestado en poco lapso de tiempo, conocer la ubicación exacta del automóvil y sentirse seguros.

Estas aplicaciones fueron creadas para promover las rutas de transporte público, que depende del gobierno y las conexiones entre ellos. Similar a otros y brindan los mismos servicios que otros Aplicaciones, por lo tanto, carecen de funcionalidad al

momento de solicitar algún dato en específico ya que se centran principalmente en mostrar las rutas: El municipio de Tizayuca, no cuenta con una guía como tal de los transportes urbanos que no sean parte de las redes manejadas por el gobierno, mientras que en otros países se pueden encontrar estos sitios web, que han evolucionado adaptándose a las necesidades de los pobladores.

### 2.2.2 Bases teóricas

**Geolocalización:** Es la capacidad de rastrear el paradero de un dispositivo utilizando, torres de telefonía celular, puntos de acceso WiFi o una combinación de estos. Dado que estos dispositivos son utilizados por individuos," la geolocalización utiliza sistemas de posicionamiento para rastrear el paradero de las personas en la latitud y Coordenadas de longitud. Incluso la dirección física.

**Firestore:** Es una plataforma en la nube para desarrollar aplicaciones web y móviles. Es adecuado para diferentes plataformas (iOS, Android y Web), lo que permite un desarrollo más rápido.

**Android Studio:** Es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones de Android, basado en IntelliJ IDEA. Además del potente editor de código y las herramientas para desarrolladores de IntelliJ, Android Studio también proporciona más funciones para mejorar la eficiencia de su trabajo al desarrollar aplicaciones de Android.

### 2.2.3 Bases Legales

Como se hará uso de la ubicación del usuario para permitir visualizar el mapa del municipio de Tizayuca Hidalgo y las opciones de la aplicación que se va desarrollar, es importante tener en cuenta el siguiente marco legal con el fin de evitar problemas ya que se trabajar con API las cuales permitirán visualizar el mapa y posteriormente utilizar la aplicación.

- Prevalencia de las Condiciones de las API de Google Maps. En caso de existir algún conflicto entre las Condiciones de las API de Google Maps y otros documentos relacionados con estas API, incluida la Documentación de las API de Google Maps, entonces tendrán prioridad las Condiciones.
- Determinación de Cumplimiento. Google se reserva el derecho de determinar, a su entera discreción, si el uso que usted hace del Servicio, el Contenido y las Características de Marca satisface estas Condiciones.

- No debe quitar, distorsionar ni alterar ningún elemento de una Característica de Marca de Google (esto incluye no comprimirlo, estirarlo, invertirlo ni decolorarlo).

#### 2.2.4 Variables

El prototipo de la aplicación contara con los siguientes factores los cuales son medibles

**Conductor:** Al iniciar cesión podrá compartir su ubicación en tiempo real del transporte público y podrá visualizar la ruta que debe recorrer en el municipio de Tizayuca Hidalgo

**Cliente:** será capaz de observar la ubicación del transporte público cuando este se ubique en una para y así mismo ver a la distancia y el tiempo de llegada del transporte a la parada

**Ruta:** Al ser un prototipo el sistema solo trabajara con un solo ruta ya que debido al tiempo es imposible introducir todas las rutas de todos los transportes públicos del municipio de Tizayuca, en este caso la ruta será de una combi que tendrá un punto de partida y un punto de destino.

**Optimización del servicio:** La finalidad del sistema es brindar un mejor servicio a los clientes en donde se les proporcione información del transporte público en tiempo real

#### **Bibliografías Marco Teórico**

- Rodolfo Perez (2021c, septiembre 24). Moovit. Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/Moovit>
- Rafael Pineda Tapia (2021, 2 abril). Citymapper. Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/Citymapper>
- D. (2021, 6 febrero). ¿Qué es Uber? Su Definición y Significado [2021]. Definición.xyz. <https://definicion.xyz/uber/>
- López, S. (2020, 27 mayo). *Firestore: qué es, para qué sirve, funcionalidades y ventajas*. DIGITAL55. <https://www.digital55.com/desarrollo-tecnologia/que-es-firebase-funcionalidades-ventajas-conclusion>

## CAPITULO 3

### Desarrollo De Actividades

#### 3.1 Salud Del Proyecto

Atreves del primer avance se busca que la aplicación se visualmente atractiva y fácil de manipular así mismo que cumpla con las métricas de calidad de desarrollo de una aplicación las cuales dictamina la calidad de la aplicación y si verdaderamente se cumplen con los requisitos y objetivos previamente establecidos

Nombre de la Actividad	Descripción de la Actividad	Porcentaje Representativo
Diseño de pantallas iniciales de la aplicación	<p>A través del entorno de Android studio se hizo el diseño total de 4 pantallas las cuales son</p> <p><b>-Pantalla 1:</b> Es la pantalla inicial de la aplicación en donde se muestran 2 botones los cuales permiten escoger si eres conductor del transporte público o si solo eres un pasajero.</p> <p><b>-Pantalla 2:</b> Es una pantalla que despliega 2 opciones las cuales permiten crear una nueva cuenta o iniciar cesión, ya sea un conductor o pasajero.</p> <p><b>-Pantalla 3:</b> Es una pantalla de registro en donde se solicitan llenar los campos para poder almacenarlos en una base de datos y posteriormente inicializar cesión.</p> <p><b>-Pantalla 4:</b> Es una pantalla de inicio de cesión en donde se va introducir el correo y la contraseña siempre y cuando el usuario este dado de alta.</p>	25%
Diseño de la base de datos	<p>Por medio de la plataforma digital FireBase se hizo la conexión con Android studio y se importaron las librerías para poder hacer uso de estas mismas funciones.</p> <p>Una vez realizada la conexión se establecieron líneas de código para que se fueran almacenando los datos</p>	

<p>Asegurar la calidad del código</p>	<p>Se adaptó el código y se crearon distintas carpetas en el proyecto para clasificar y establecer un orden en la sintaxis para poder cumplir con los estándares de calidad, en donde de acuerdo al contenido se establecía un orden, para evitar, corregir y localizar fallos de una manera más precisa.</p> <p>así mismo se crearon clases para el modelo de base de datos en donde se establecieron los parámetros y atributos los culés se mandaría a llamar en las diferentes clases; para así evitar posibles errores de compilación ya que anteriormente no existía este orden.</p>	<p><b>25%</b></p>
<p>Diseño de Pantalla de Navegación del conductor</p>	<p>Se empezó a programar y a diseñar las principales funcionalidades de la aplicación en donde por medio de la pantalla del conductor esto podrá visualizar y compartir su ubicación en tiempo real, para posteriormente esta ubicación puede ser monitoreada y visualizada a través de la pantalla de navegación del pasajero;</p> <p>De igual manera se estableció la ruta la cual recorrerá el transporte público y se colocaron marcadores en donde se muestra el lugar de partida del vehículo y el lugar de destino</p>	<p><b>25%</b></p>
<p>Diseño de Pantalla de Navegación del Pasajero</p>	<p>Se diseñó y se programó la última la pantalla la cual es la del pasajero esta presenta funcionalidades parecidas a las del conductor en donde también se puede visualizar el trayecto que recorre el vehículo y la ubicación de uno mismo en tiempo real.</p> <p>Así mismo en el mapa de pasajero también se muestra la ubicación del en conductor para poder monitorearla este diseño aún no está concluido ya que el desarrollo;</p> <p>De igual manera esta pantalla contara con un botón la cual permita solicitar al pasajero el tiempo y la distancia que hay del vehículo del transporte publico hasta du ubicación.</p>	<p><b>25%</b></p>

Tabla 1 Tabla de Avance

Pantalla de Notificación al usuario

### 3.2 Desarrollo de la aplicación

Por medio del entorno de Android studio se empezó a realizar el prototipo de la aplicación, dicha aplicación consiste en informar a los pasajeros el tiempo aproximado de llegada del transporte público y para ello se necesitarán 2 tipos de usuario uno el cual proporcione la información sobre la ubicación del transporte que en esta caso será el conductor del transporte publico y el otro será el que reciba y visualice la información que en este caso será el cliente que dese abordar el transporte, así mismo esta prototipo se desarrollara en una sola ruta del colectivo y el pasajero al posicionarse en una determinada área podrá observar la información antes mencionada.

primero se desarrollaron 4 pantallas, la primera consiste en ser el menú el cual elegirás el tipo de usuario (conductor o un pasajero), la segunda pantalla es para escoger si iniciaras cesión o crearas una cuenta, la tercera es la pantalla de registro para poder darte de alta en la aplicación y la cuarta es para poder inicializar cesión

#### 3.2.1 Pantallas Principales

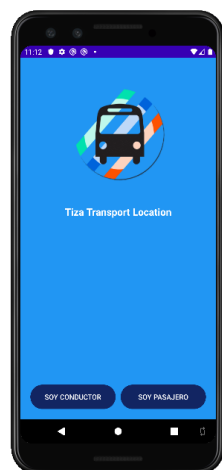


Figura 2 Pantalla Principal

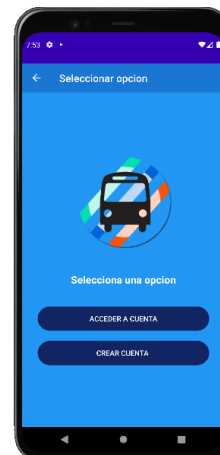


Figura 2.1 Pantalla de Acceso

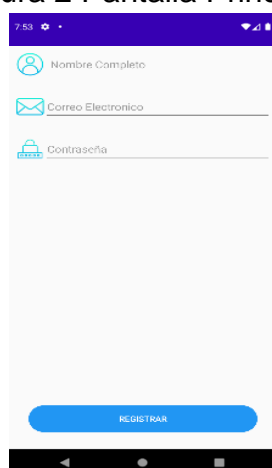


Figura 2.2 Pantalla de Registro del Pasajero

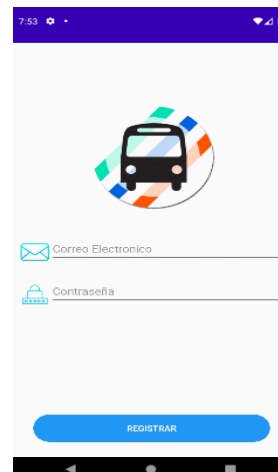


Figura 2.3 Pantalla de Login



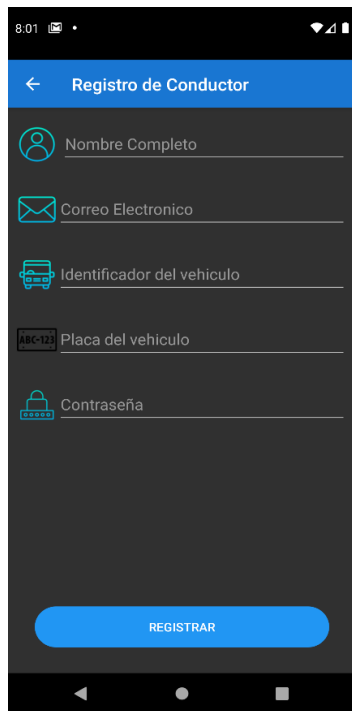


Figura 2.4 Pantalla de Registro del Conductor

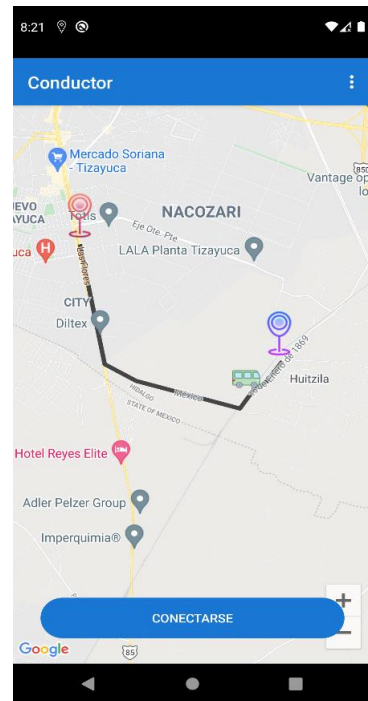


Figura 2.5 Pantalla de Navegación del Conductor

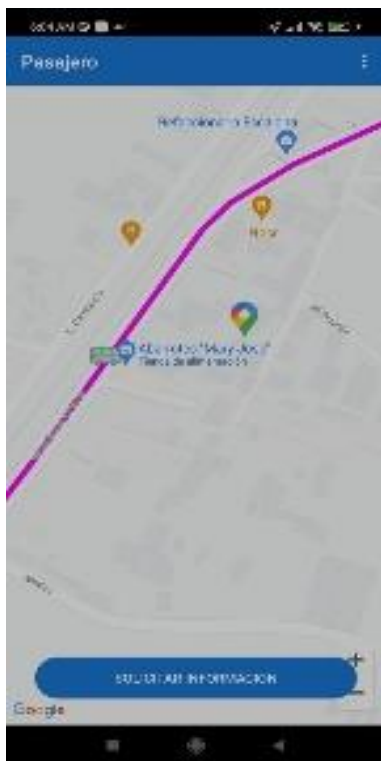


Figura 2.6 Pantalla de Navegación del Pasajero

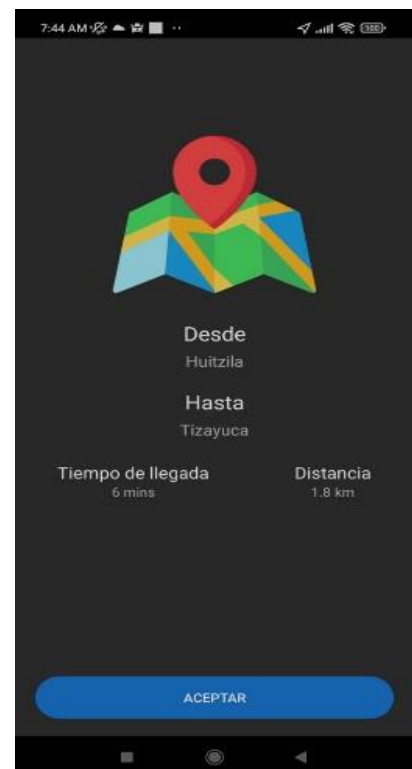


Figura 2.7 Pantalla de Notificación al usuario



una vez creada las pantallas se procedió agregar la funcionalidad de cada una e interconectarlas por medio de botones así mismo se realizó la conexión de Android studio con FireBase de acuerdo a los pasos y las indicaciones que mostraba esta plataforma

Al terminar la conexión de Android studio se procedió a inicializar las variables las cuales guardarían los datos introducidos a los campos, al momento de registrarse; así mismo cabe aclarar que tanto el usuario de conductor como pasajero comparten las mismas pantallas de Login y de Registro, pero al momento de introducir la información esta se almacenará y clasificara por separado.

De igual manera si pusieron las condiciones las cuales permitían evitar que los campos no estuvieran vacíos, que el mismo correo se repita 2 veces y que el número de caracteres de la contraseña fuera mayor a 6; Por último se verifico que si almacenara los datos introducidos por el usuario y que de acuerdo a estos datos se iniciara la cesión. (ver figura 3)

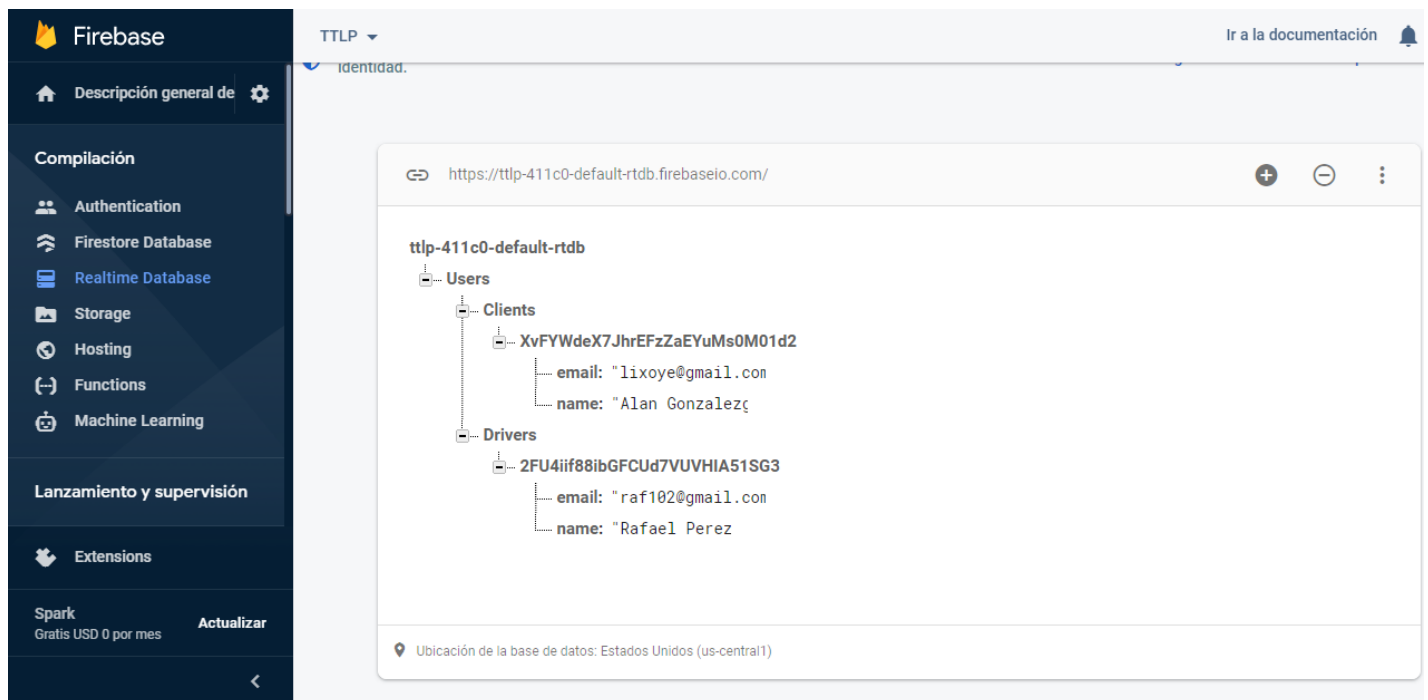


Figura 3 Base de Datos FireBase

Para evitar errores de programación y establecer un orden en la sintaxis al momento de programar y compilar el programa se crearon distintas carpetas las cuales de acuerdo al contenido de las plantillas y de las clases creadas se clasificarían, esto

con el motivo de brindar un sistema el cual cumpla con las normas de calidad, al igual que se fácil de manipular; Estas carpetas son las siguientes:

**Activities:** Contiene las pantallas principales de la aplicación las cuales son al momento de elegir la opción de registrarse e iniciar cesio y la pantalla de bienvenida de la aplicación, así mismo contiene otras 2 carpetas la cuales son la carpeta de Driver y Client en donde se hacen los procesos de registro de usuarios y las validaciones.

**Includes;** Contiene la función del Toolbar para hacer uso de ello sin la necesidad de repetir el código ya que por medio de esta clase se puede llamar.

**Models:** Contiene la clase de base de datos en donde se establecen los parámetros de cada usuario.

**Providers:** Como su mismo nombre lo indica por medio de estas clases hacemos uso de los servicios de base de datos Firebase para la autenticación de correos y almacenamiento de estos mismos.

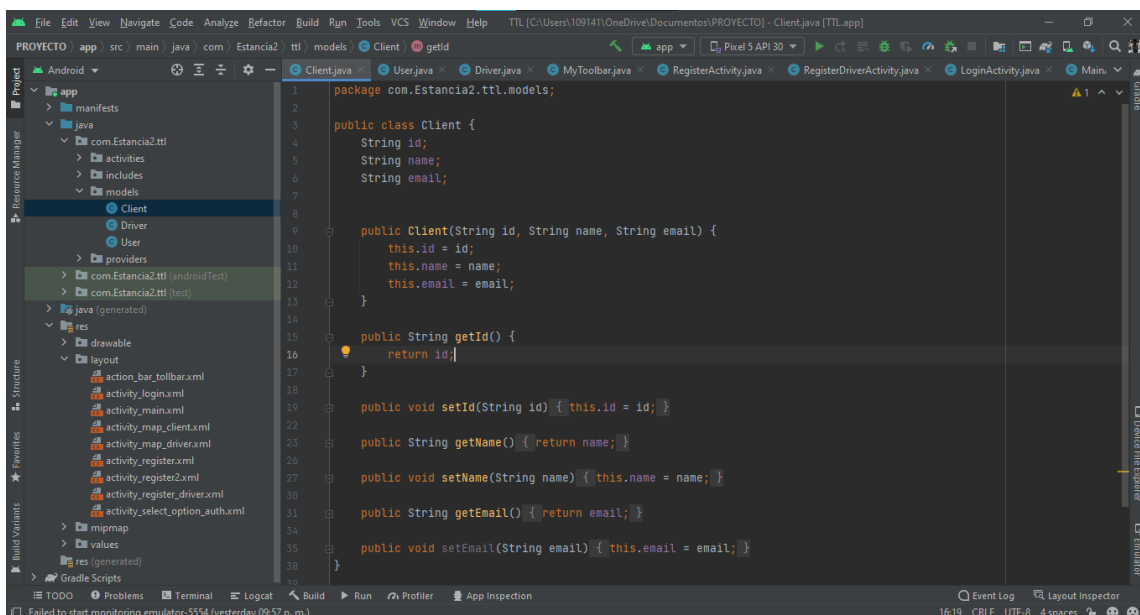


Figura 4 Creación de carpetas y organización del código

Al concluir este proceso se empezó a diseñar la pantalla principal de navegación del conductor en donde por medio de la geolocalización el podrá visualizar su ubicación en tiempo real y en caso de desplazar el vehículo esta se actualizara, de modo que estas acciones se puedan visualizar en la pantalla de navegación de pasajeros, así mismo se estableció la ruta en la cual navegara el vehículo y se

colocaron marcadores del lugar de origen y destino del transporte, También se agregó un botón en caso de que el conductor del transporte público quiera dejar de compartir la ubicación, esto con la finalidad de que al terminar la jornada su ubicación ya no sea monitoreada; de igual manera se hicieron las validación para que el usuario aceptara los términos de uso de la aplicación ya que se hará uso de su ubicación para poder utilizar estas funcionalidades y en caso de no aceptar o de no activar el GPS esta no funcionara ya que requiere de los términos antes mencionados.

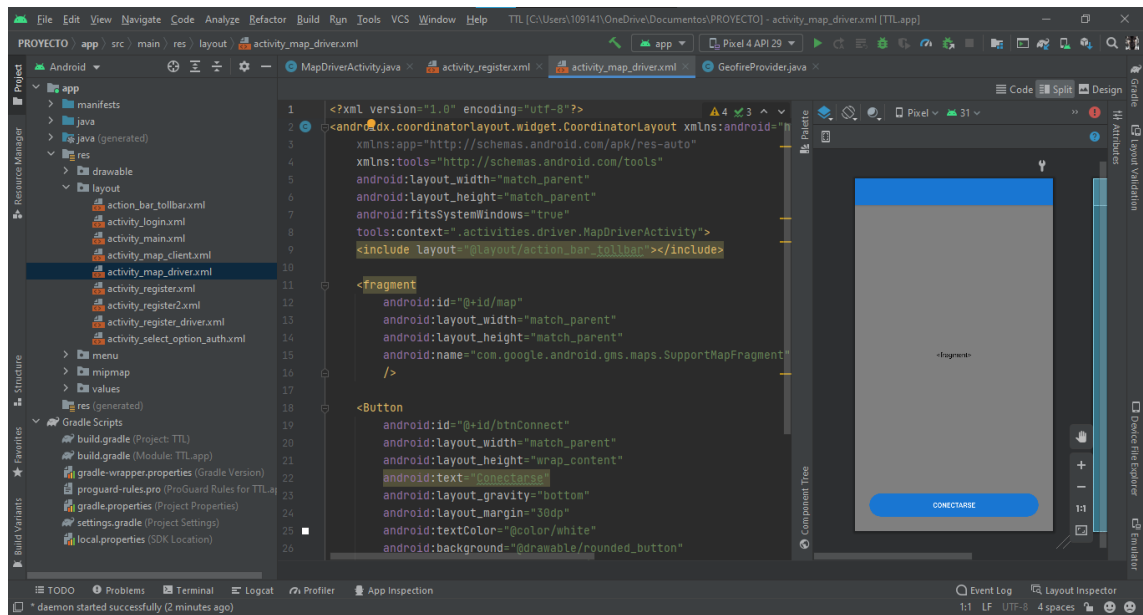


Figura 5 Android Studio-Mapa conductor

En el desarrollo de la pantalla de pasajero no solo se puede visualizar la ubicación propia de uno, sino que también se puede monitorear la ubicación del transporte público, en el cual por medio del mapa se observar como el vehículo se está desplazando; al igual que el pasajero también puede visualizar la ruta que recorre el transporte público para así poder estar informado de los lugares que recorre el transporte el cual abordara; así mismo se contara con un botón en donde al clickearlo este abrirá otra pantalla que proporcione información acerca del trayecto del vehículo de transporte público, en donde se indique el tiempo de llegada y la distancia que hay entre la ubicación actual del pasajero y la ubicación del transporte público, esto con la finalidad de poder brindar un mejor servicio al igual que poder proporcionar información al pasajero para así administrar de mejor manera su tiempo.

### 3.2.2 Código

#### Registro de Conductor

```
package com.Estancia2.ttl.activities.driver;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import com.Estancia2.ttl.R;
import com.Estancia2.ttl.activities.MainActivity;
import com.Estancia2.ttl.activities.client.RegisterActivity;
import com.Estancia2.ttl.includes.MyToolbar;
import com.Estancia2.ttl.models.Client;
import com.Estancia2.ttl.models.Driver;
import com.Estancia2.ttl.providers.AuthProvider;
import com.Estancia2.ttl.providers.ClientProvider;
import com.Estancia2.ttl.providers.DriverProvider;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.textfield.TextInputEditText;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

import dmax.dialog.SpotsDialog;

public class RegisterDriverActivity extends AppCompatActivity {

    AuthProvider mAuthProvider;
    DriverProvider mDriverProvider;
    //views
    Button mButtonRegister;
    TextInputEditText mTextInputEmail;
    TextInputEditText mTextInputName;
    TextInputEditText mTextInputVehicleName;
    TextInputEditText getmTextInputVehiclePlate;
    TextInputEditText mTextInputPassword;
    AlertDialog mDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register_driver);
    }
}
```

```
MyToolbar.show(this, "Registro de Conductor", true);

 mAuthProvider=new AuthProvider();
 mDriverProvider= new DriverProvider();

 mDialog = new
 SpotsDialog.Builder().setContext(RegisterDriverActivity.this).setMessage("Espere un
 momento").build();

 mButtonRegister=findViewById(R.id.btnRegister);
 mTextInputEmail=findViewById(R.id.textInputEmail);
 mTextInputName=findViewById(R.id.textInputName);
 mTextInputVehicleName=findViewById(R.id.textInputVehicleName);
 getmTextInputVehiclePlate=findViewById(R.id.textInputVehiclePlate);
 mTextInputPassword=findViewById(R.id.textInputPassword);

 mButtonRegister.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        clickRegister();
    }
});

void clickRegister(){
    final String name=mTextInputName.getText().toString();
    final String email=mTextInputEmail.getText().toString();
    final String vehicleName=mTextInputVehicleName.getText().toString();
    final String vehiclePlate=getmTextInputVehiclePlate.getText().toString();
    final String password=mTextInputPassword.getText().toString();

    if (!name.isEmpty() && !email.isEmpty() && !password.isEmpty() &&
    !vehicleName.isEmpty() && !vehiclePlate.isEmpty()){
        if (password.length() >= 6){
            mDialog.show();
            register(name,email,password,vehicleName,vehiclePlate);
        }
        else {
            Toast.makeText(this, "La contraseña debe de tener minimo 6 caracteres",
            Toast.LENGTH_SHORT).show();
        }
    }
    else{
        Toast.makeText(this, "Ingresa todos los campos", Toast.LENGTH_SHORT).show();
    }
}

void register(final String name, String email,String password,String vehicleName,String
vehiclePlate){
    mAuthProvider.register(email,password).addOnCompleteListener(new
    OnCompleteListener<AuthResult>() {
```

```
@Override
public void onComplete(@NonNull Task<AuthResult> task) {
    mDialog.hide();
    if (task.isSuccessful()) {
        String id= FirebaseAuth.getInstance().getCurrentUser().getUid();
        Driver driver=new Driver(id,name,email,vehicleName,vehiclePlate);
        create(driver);
    }
    else {
        Toast.makeText(RegisterDriverActivity.this,"Error al registrar
usuario",Toast.LENGTH_SHORT).show();
    }
}

});

}

void create(Driver driver){
    mDriverProvider.create(driver).addOnCompleteListener(new OnCompleteListener<Void>()
{
    @Override
    public void onComplete(@NonNull Task<Void> task) {
        if(task.isSuccessful()){
            //Toast.makeText(RegisterDriverActivity.this, "el registro fue exitoso",
Toast.LENGTH_SHORT).show();
            Intent intent=new Intent(RegisterDriverActivity.this,
MapDriverActivity.class);
            intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
            startActivity(intent);
        }else {
            Toast.makeText(RegisterDriverActivity.this, "No se pudo registrar el
cliente", Toast.LENGTH_SHORT).show();
        }
    }
});
}
}
```

Tabla 2 Tabla de Código Para Registrar Conductor

## Registro de Pasajero

```
Package com.Estancia2.ttl.activities.client;

import android.app.AlertDialog;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.Estancia2.ttl.R;
import com.Estancia2.ttl.activities.driver.MapDriverActivity;
import com.Estancia2.ttl.activities.driver.RegisterDriverActivity;
import com.Estancia2.ttl.includes.MyToolbar;
import com.Estancia2.ttl.models.Client;
import com.Estancia2.ttl.providers.AuthProvider;
import com.Estancia2.ttl.providers.ClientProvider;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.textfield.TextInputEditText;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

import dmax.dialog.SpotsDialog;

public class RegisterActivity extends AppCompatActivity {
    AuthProvider mAuthProvider;
    ClientProvider mClientProvider;
    //views
    Button mButtonRegister;
    TextInputEditText mTextInputEmail;
    TextInputEditText mTextInputName;
    TextInputEditText mTextInputPassword;
    AlertDialog mDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        MyToolbar.show(this, "Registro de Usuario", true);

        mAuthProvider=new AuthProvider();
        mClientProvider= new ClientProvider();
    }
}
```

```
mDialog = new
SpotsDialog.Builder().setContext(RegisterActivity.this).setMessage("Espere un
momento").build();

mButtonRegister=findViewById(R.id.btnRegister);
mTextInputEmail=findViewById(R.id.textInputEmail);
mTextInputName=findViewById(R.id.textInputName);
mTextInputPassword=findViewById(R.id.textInputPassword);

mButtonRegister.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        clickRegister();
    }
});

void clickRegister(){
    final String name=mTextInputName.getText().toString();
    final String email=mTextInputEmail.getText().toString();
    final String password=mTextInputPassword.getText().toString();

    if (!name.isEmpty() && !email.isEmpty() && !password.isEmpty()){
        if (password.length() >= 6){
            mDialog.show();
            register(name,email,password);
        }
        else {
            Toast.makeText(this, "La contraseña debe de tener minimo 6 caracteres",
Toast.LENGTH_SHORT).show();
        }
    }
    else{
        Toast.makeText(this, "Ingresa todos los campos", Toast.LENGTH_SHORT).show();
    }
}

void register(final String name, String email,String password){
    mAuthProvider.register(email,password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            mDialog.hide();
            if (task.isSuccessful()){
                String id=FirebaseAuth.getInstance().getCurrentUser().getUid();
                Client client=new Client(id,name,email);
                create(client);
            }
            else {
```



```
        Toast.makeText(RegisterActivity.this, "Error al registrar
usuario", Toast.LENGTH_SHORT).show();
    }

    });

}

void create(Client client){
    mClientProvider.create(client).addOnCompleteListener(new OnCompleteListener<Void>()
{
    @Override
    public void onComplete(@NonNull Task<Void> task) {
        if(task.isSuccessful()){
            Intent intent=new Intent(RegisterActivity.this,
MapClientActivity.class);
            intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
            startActivity(intent);
        }else {
            Toast.makeText(RegisterActivity.this, "No se pudo registrar el cliente",
Toast.LENGTH_SHORT).show();
        }
    }
});
}
```

Tabla 2.1 Tabla de Código Para Registrar Pasajero

## Inicio de cesión

```
Package com.Estancia2.ttl.activities.client;

import android.app.AlertDialog;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.Estancia2.ttl.R;
import com.Estancia2.ttl.activities.driver.MapDriverActivity;
import com.Estancia2.ttl.activities.driver.RegisterDriverActivity;
import com.Estancia2.ttl.includes.MyToolbar;
import com.Estancia2.ttl.models.Client;
import com.Estancia2.ttl.providers.AuthProvider;
import com.Estancia2.ttl.providers.ClientProvider;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.textfield.TextInputEditText;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

import dmax.dialog.SpotsDialog;

public class RegisterActivity extends AppCompatActivity {
    AuthProvider mAuthProvider;
    ClientProvider mClientProvider;
    //views
    Button mButtonRegister;
    TextInputEditText mTextInputEmail;
    TextInputEditText mTextInputName;
    TextInputEditText mTextInputPassword;
    AlertDialog mDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        MyToolbar.show(this, "Registro de Usuario", true);

        mAuthProvider=new AuthProvider();
        mClientProvider= new ClientProvider();
    }
}
```

```
mDialog = new
SpotsDialog.Builder().setContext(RegisterActivity.this).setMessage("Espere un
momento").build();

mButtonRegister=findViewById(R.id.btnRegister);
mTextInputEmail=findViewById(R.id.textInputEmail);
mTextInputName=findViewById(R.id.textInputName);
mTextInputPassword=findViewById(R.id.textInputPassword);

mButtonRegister.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        clickRegister();
    }
});

void clickRegister(){
    final String name=mTextInputName.getText().toString();
    final String email=mTextInputEmail.getText().toString();
    final String password=mTextInputPassword.getText().toString();

    if (!name.isEmpty() && !email.isEmpty() && !password.isEmpty()){
        if (password.length() >= 6){
            mDialog.show();
            register(name,email,password);
        }
        else {
            Toast.makeText(this, "La contraseña debe de tener minimo 6 caracteres",
Toast.LENGTH_SHORT).show();
        }
    }
    else{
        Toast.makeText(this, "Ingresa todos los campos", Toast.LENGTH_SHORT).show();
    }
}

void register(final String name, String email,String password){
    mAuthProvider.register(email,password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            mDialog.hide();
            if (task.isSuccessful()){
                String id=FirebaseAuth.getInstance().getCurrentUser().getUid();
                Client client=new Client(id,name,email);
                create(client);
            }
            else {
```

```
        Toast.makeText(RegisterActivity.this, "Error al registrar
usuario", Toast.LENGTH_SHORT).show();
    }
}

});

}

void create(Client client){
    mClientProvider.create(client).addOnCompleteListener(new OnCompleteListener<Void>()
{
    @Override
    public void onComplete(@NonNull Task<Void> task) {
        if(task.isSuccessful()){
            Intent intent=new Intent(RegisterActivity.this,
MapClientActivity.class);
            intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
            startActivity(intent);
        }else {
            Toast.makeText(RegisterActivity.this, "No se pudo registrar el cliente",
Toast.LENGTH_SHORT).show();
        }
    }
});
}
```

Tabla 2.2 Tabla de Código Para Iniciar Cesión

## Pantalla de navegación del conductor

```
package com.Estancia2.ttl.activities.driver;

import android.Manifest;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.location.Location;
import android.location.LocationManager;
import android.os.Build;
import android.os.Bundle;
import android.os.Looper;
import android.provider.Settings;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.Estancia2.ttl.R;
import com.Estancia2.ttl.activities.MainActivity;
import com.Estancia2.ttl.activities.client.MapClientActivity;
import com.Estancia2.ttl.includes.MyToolbar;
import com.Estancia2.ttl.providers.AuthProvider;
import com.Estancia2.ttl.providers.GeofireProvider;
import com.Estancia2.ttl.providers.GoogleApiProvider;
import com.Estancia2.ttl.utils.DecodePoints;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
```

```
import com.google.android.gms.maps.model.CameraPosition;
import com.google.android.gms.maps.model.JointType;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.maps.model.PolylineOptions;
import com.google.android.gms.maps.model.SquareCap;
import com.google.firebase.auth.FirebaseAuth;

import org.json.JSONArray;
import org.json.JSONObject;

import java.util.List;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class MapDriverActivity extends AppCompatActivity implements
    OnMapReadyCallback {
    private GoogleMap mMap;
    private SupportMapFragment mMapFragment;
    private AuthProvider mAuthProvider;
    private GeofireProvider mGeofireProvider;
    private LocationRequest mLocationRequest;
    private FusedLocationProviderClient mFusedLocation;
    private final static int LOCATION_REQUEST_CODE = 1;
    private final static int SETTINGS_REQUEST_CODE = 2;
    private Marker mMarker;
    private Button mButtonConnect;
    private boolean mIsConnect=false;
    private LatLng mCurrentLatLng;
    private LatLng mOriginLatLng;
    private LatLng mDestinationLatLng;
    private List<LatLng> mPolylineList;
    private PolylineOptions mPolylineOptions;
    private GoogleApiProvider mGoogleApiProvider;

    LocationCallback mLocationCallback = new LocationCallback() {
        @Override
        public void onLocationResult(LocationResult locationResult) {
            for (Location location : locationResult.getLocations()) {
                if (getApplicationContext() != null) {
                    mCurrentLatLng=new
                    LatLng(location.getLatitude(),location.getLongitude());

                    if (mMarker !=null){
                        mMarker.remove();
                    }
                }
            }
        }
    }
}
```

```
mMarker=mMap.addMarker(new
MarkerOptions().position(
    new
LatLng(location.getLatitude(),location.getLongitude())
)
    .title("Tu Posicion")
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.transport))
);
//ubicacion en tiempo real

mMap.moveCamera(CameraUpdateFactory.newCameraPosition(
    new CameraPosition.Builder()
        .target(new
LatLng(location.getLatitude(), location.getLongitude()))
        .zoom(15f)
        .build()
));
updateLocation();

}
}
};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_map_driver);
    MyToolbar.show(this, "Conductor", false);

    mAuthProvider = new AuthProvider();
    mGeofireProvider=new GeofireProvider();

    mFusedLocation =
    LocationServices.getFusedLocationProviderClient(this);
    mOriginLatLng=new LatLng(19.81358581103487,-98.95870057590021);
    mDestinationLatLng=new LatLng(19.8248712062147,-
    98.97743782479284);

    mMapFragment = (SupportMapFragment)
    getSupportFragmentManager().findFragmentById(R.id.map);
    mMapFragment.getMapAsync(this);
    mButtonConnect=findViewById(R.id.btnConnect);
    mGoogleApiProvider=new
    GoogleApiProvider(MapDriverActivity.this);
```

```
mButtonConnect.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if (mIsConnect) {  
            disconnect();  
        }  
        else {  
            startLocation();  
        }  
    }  
});  
  
}  
  
public void updateLocation(){  
    if(mAuthProvider.existSession() && mCurrentLatLng !=null){  
mGeofireProvider.saveLocation(mAuthProvider.getId(),mCurrentLatLng);  
    }  
}  
  
private void drawRoute() {  
    mGoogleApiProvider.getDirections(mOriginLatLng,  
mDestinationLatLng).enqueue(new Callback<String>() {  
        @Override  
        public void onResponse(Call<String> call, Response<String>  
response) {  
            try {  
  
                JSONObject jsonObject = new  
JSONObject(response.body());  
                JSONArray jsonArray =  
jsonObject.getJSONArray("routes");  
                JSONObject route = jsonArray.getJSONObject(0);  
                JSONObject polyLines =  
route.getJSONObject("overview_polyline");  
                String points = polyLines.getString("points");  
                mPolylineList = DecodePoints.decodePoly(points);  
                mPolylineOptions = new PolylineOptions();  
                mPolylineOptions.color(Color.DKGRAY);  
                mPolylineOptions.width(13f);  
                mPolylineOptions.startCap(new SquareCap());  
                mPolylineOptions.jointType(JointType.ROUND);  
                mPolylineOptions.addAll(mPolylineList);  
                mMap.addPolyline(mPolylineOptions);  
  
            } catch (Exception e) {  
                Log.d("Error", "Error encontrado " +  
e.getMessage());  
            }  
        }  
    }  
});  
}
```



```
    }  
    }  
    @Override  
    public void onFailure(Call<String> call, Throwable t) {  
    }  
    }  
});  
}  
  
@Override  
public void onMapReady(GoogleMap googleMap) {  
    mMap = googleMap;  
    mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);  
    mMap.getUiSettings().setZoomControlsEnabled(true);  
    if (ActivityCompat.checkSelfPermission(this,  
Manifest.permission.ACCESS_FINE_LOCATION) !=  
PackageManager.PERMISSION_GRANTED &&  
ActivityCompat.checkSelfPermission(this,  
Manifest.permission.ACCESS_COARSE_LOCATION) !=  
PackageManager.PERMISSION_GRANTED) {  
    }  
  
    mLocationRequest = new LocationRequest();  
    mLocationRequest.setInterval(1000);  
    mLocationRequest.setFastestInterval(1000);  
  
    mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);  
    mLocationRequest.setSmallestDisplacement(5);  
    mMap.addMarker(new  
MarkerOptions().position(mOriginLatLng).title("ORIGEN").icon(BitmapDesc  
riptorFactory.fromResource(R.drawable.icon_pinblue)));  
    mMap.addMarker(new  
MarkerOptions().position(mDestinationLatLng).title("Destino").icon(Bitm  
apDescriptorFactory.fromResource(R.drawable.icon_pinred)));  
    drawRoute();  
}  
  
@Override  
public void onRequestPermissionsResult(int requestCode, @NonNull  
String[] permissions, @NonNull int[] grantResults) {  
    super.onRequestPermissionsResult(requestCode, permissions,  
grantResults);  
    if (requestCode == LOCATION_REQUEST_CODE) {  
        if (grantResults.length > 0 && grantResults[0] ==  
PackageManager.PERMISSION_GRANTED) {  
            if (ContextCompat.checkSelfPermission(this,  
Manifest.permission.ACCESS_FINE_LOCATION) ==  
PackageManager.PERMISSION_GRANTED) {  
                if (gpsActivated()) {
```

```
mFusedLocation.requestLocationUpdates(mLocationRequest,
mLocationCallback, Looper.myLooper());
        mMap.setMyLocationEnabled(false);
    } else {
        showAlertDialogNOGPS();
    }
} else {
    checkLocationPermissions();
}
} else {
    checkLocationPermissions();
}
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == SETTINGS_REQUEST_CODE && gpsActivated()) {
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {

            return;
        }
        mFusedLocation.requestLocationUpdates(mLocationRequest,
mLocationCallback, Looper.myLooper());
        mMap.setMyLocationEnabled(false);
    }
    else {
        showAlertDialogNOGPS();
    }
}

private void showAlertDialogNOGPS() {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setMessage("Por favor activa tu ubicacion para
continuar")
        .setPositiveButton("Configuraciones", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface
dialogInterface, int i) {
```

```
startActivityForResult(new
Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS),
SETTINGS_REQUEST_CODE);
    }
    }).create().show();
}

private boolean gpsActivated() {
    boolean isActive = false;
    LocationManager locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);

    if
(locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)) {
        isActive = true;
    }
    return isActive;
}

private void disconnect(){

    if (mFusedLocation != null){
        mButtonConnect.setText("Conectarse");
        mIsConnect=false;
        mFusedLocation.removeLocationUpdates(mLocationCallback);
        if (mAuthProvider.existsSession()){
            mGeofireProvider.removeLocation(mAuthProvider.getId());
        }
    }
    else{
        Toast.makeText(this, "No te puedes desconectar",
Toast.LENGTH_SHORT).show();
    }

}

private void startLocation() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED) {
            if (gpsActivated()) {
                mButtonConnect.setText("Desconectarse");
                mIsConnect=true;

mFusedLocation.requestLocationUpdates(mLocationRequest,
mLocationCallback, Looper.myLooper());
                mMap.setMyLocationEnabled(false);
            }
        }
    }
}
```

```
        else {
            showAlertDialogNOGPS();
        }
    }
    else {
        checkLocationPermissions();
    }
} else {
    if (gpsActivated()) {
        mFusedLocation.requestLocationUpdates(mLocationRequest,
mLocationCallback, Looper.myLooper());
        mMap.setMyLocationEnabled(false);
    }
    else {
        showAlertDialogNOGPS();
    }
}
}

private void checkLocationPermissions() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
        if
(ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.ACCESS_FINE_LOCATION)) {
            new AlertDialog.Builder(this)
                .setTitle("Proporciona los permisos para
continuar")
                .setMessage("Esta aplicacion requiere de los
permisos de ubicacion para poder utilizarse")
                .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface
dialogInterface, int i) {

ActivityCompat.requestPermissions(MapDriverActivity.this, new String[]
{Manifest.permission.ACCESS_FINE_LOCATION}, LOCATION_REQUEST_CODE);
                    }
                })
                .create()
                .show();
        }
        else {

ActivityCompat.requestPermissions(MapDriverActivity.this, new String[]
{Manifest.permission.ACCESS_FINE_LOCATION}, LOCATION_REQUEST_CODE);
        }
    }
}
```

```
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.driver_menu, menu);

    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    if (item.getItemId() == R.id.action_logout) {
        logout();
    }
    return super.onOptionsItemSelected(item);
}

void logout() {
    disconnect();
    mAuthProvider.logout();
    Intent intent = new Intent(MapDriverActivity.this,
MainActivity.class);
    startActivity(intent);
    finish();
}
}
```

Tabla 2.3 Tabla de Código Para la Pantalla de Navegación del Conductor

## CAPITULO 4

### *Conclusiones Y Perspectivas*

En este trabajo se desarrolló una aplicación la cual busca brindar información sobre la distancia y el tiempo del transporte público, esto enfocada a una sola ruta del municipio de Tizayuca Hidalgo y aun solo vehículo, ya que es un prototipo el cual permitirá visualizar la información de manera rápida por medio de la geolocalización de un dispositivo móvil. Así mismo se menciona que para hacer uso de esta aplicación en todo momento se tiene que tener acceso a internet ya que se hace uso de los servicios de google y de Firebase.

Para desarrollar esta aplicación se ocupó el modelo vista controlador (MVC), el controlador es la aplicación móvil desarrollada, La vista es proporcionada por la plataforma de Google Maps y el modelo está implementado en la plataforma antes mencionada(Firebase);

Google Maps Platform proporciono todo lo que se necesitaba para procesar datos como lo son las visualizaciones de la ruta, la ubicación en tiempo real, el tiempo de llegada del vehículo, etc.

En lo que a Firebase se refiere, nos proporciona todo lo relacionado con el almacenamiento de datos y el almacenamiento de la ubicación de los pasajeros y del transporte público.-para así posteriormente hacer uso de estos datos y poder implementarlos en el código exitosamente; De igual manera se hace mención que para hacer uso de esta aplicación se necesita en todo momento internet debido a que se hace uso de los servicios de Google, y la base de datos se encuentra en la nube lo que ocasiona que las validaciones de los correos electrónicos, contraseñas y los tipos de usuario necesiten internet para que se dé el acceso a la aplicación o se indique si el correo o la contraseña son correctos, es por ello que se recomienda en todo momento contar con una conexión estable de internet en especial los conductores de vehículos del transporte público, ya que son ellos los principales actores los cuales a través de su ubicación se visualizara la información del tiempo y la distancia que hay de punto a punto (Pasajero-Vehículo) así mismo el pasajero igual debe de contar con una buena conexión de red ya que el será el principal receptor de la información proporcionada de la ubicación del conductor del transporte público.

El prototipo esta diseñado para poder instalarse en dispositivos móviles que cuenta con una versión de Android superior a 6 ya que versiones más antiguas pueden generar problemas al momento de compartir la ubicación o abrir el sistema ya que las librerías implementadas en el código requieren de una versión estable y compatible con los permisos de ubicación y de acceso a internet.

Lo más importante del proyecto fue notificar al usuario que en este caso es el pasajero, el tiempo de llegada y la distancia que hay entre su ubicación actual y la ubicación del vehículo de transporte público, esto por medio de una interfaz gráfica programada con el lenguaje de Java, Kotlin en el entorno Android Studio; esta interfaz cuenta con un diseño gráfico atractivo y fácil de manipular para que los usuarios hagan uso de ella intuitivamente. Así mismo las opciones del conductor son las siguientes:

- Visualizar la ruta que va a recorrer desde el lugar de origen hasta el lugar de destino.
- Visualizar el mapa del municipio de Tizayuca por medio de los servicios de Google Maps.
- Contar con un botón para activar o desactivar su ubicación en tiempo real.
- Poder Acceder o Registrar una nueva cuenta de conductor de transporte público.
- Visualizar por medio de un icono su ubicación en tiempo real.

Las Funciones para el pasajero son las siguientes:

- Visualizar el trayecto que hará el transporte público desde su origen hasta su destino.
- Visualizar el mapa del municipio de Tizayuca por medio de los servicios de Google Maps.
- Visualizar y monitorear por medio del mapa la posición del vehículo del transporte público.
- Visualizar la ubicación propia en tiempo real.
- Tener un botón para solicitar la información del trayecto, esto quiere decir que se nos abrirá una ventana emergente para poder saber la distancia y el tiempo que hay desde la posición actual de uno a la posición que tiene el vehículo.

La finalidad de las funciones tanto de pasajero como las del conductor es brindar herramientas necesarias que pueda ayudar a compartir información enfocada a la ubicación en tiempo real ya que actualmente en el municipio de Tizayuca Hidalgo no se tiene un sistema o una aplicación que apoye a los Transeúntes, el prototipo desarrollado es el primero en satisfacer las necesidades antes mencionadas y sirve para establecer las bases para el desarrollo de una aplicación un poco más compleja la cual incorpore todas las rutas y todos los vehículos con los que cuenta el municipio y así poder dar la solución en su totalidad a la pregunta de investigación previamente desarrollada.

Lo que más me ayudo al programar el prototipo aparte de IDE de Android Studio y de la plataforma Firebase fueron las Apis de google las cuales son un conjunto de definiciones y protocolos que se utilizan para desarrollar e integrar software de aplicación y permiten que los productos y servicios se comuniquen con otros sin saber cómo se implementan.

Esto simplifica que las Apis permiten que el desarrollo de aplicaciones ahorre tiempo y dinero, ya que brindan flexibilidad; simplifican el diseño, la administración y el uso de aplicaciones, y brindan oportunidades de innovación, que son ideales al diseñar nuevas herramientas. A veces, una API se considera un contrato y su documentación representa un acuerdo entre las dos partes: si una de las partes envía una solicitud remota con una estructura específica, esa estructura determinará cómo responderá el software de la otra parte.

Las API de google me permitió conectar mi propia infraestructura a través del desarrollo de aplicaciones nativas de la nube, al igual que me ofrece opciones las cuales me permite compartir datos con distintos tipos de usuarios. haciendo que valor comercial sea único debido a que se simplifican permitiendo que los datos sean rentables, Es muy importante apoyar el rápido despliegue y desarrollo de servicios innovadores. El desarrollo de aplicaciones nativas de la nube es una forma identificable de aumentar la velocidad de desarrollo y se basa en la API para conectarse a la arquitectura de la aplicación de micro servicio.

En resumen, las API me permitieron habilitar el acceso a los recursos mientras mantenía la seguridad y el control, ya que al crear aplicaciones que utilicen datos o las funciones proporcionadas por la API, estas puede utilizar una plataforma de integración distribuida para conectar todos los elementos que en esta caso son los elementos de geolocalización de mi sistema. En este caso las Apis de google que si implementaron son las siguientes:

- API Directions: permite proporcionar rutas de transporte público, bicicleta, auto y a pié.
- API Distance Matrix: proporciona la distancia y el tiempo de viaje entre los puntos de origen y de destino de una ruta recomendada.
- SDK de Maps Android: Se puedes agregar mapas a tu app para Android, incluidas las apps para Wear OS que utilizan datos, mapas y respuestas gestuales de Google Maps.

Todos esos servicios fueron implementados en el sistema con la finalidad de que se cumplieran las funcionalidades previamente de finidas en los objetivos generales; Así mismo el uso de las Apis fueron de suma importancia en el desarrollo de la aplicación ya que sin ellas hubiera sido imposible generar y mostrar la ubicación en tiempo real.



Lo que más se me complicó fue la incompatibilidad con algunas librerías debido a que las versiones actuales generaban pequeños fallos lo que me imposibilitaban el poder implementar cierta línea de código para mostrar las ubicaciones en tiempo real, pero esto se soluciona ocupando librerías anteriores a las versiones actuales

De igual manera la versión actual del IDE de Android Studio que es la 3.1 presenta ciertas modificaciones en los ajustes y en las herramientas a comparación de versiones inferiores por lo que a veces los métodos implementados o sintaxis en algunas partes del código cambiaban. Ocasionando pequeños errores de compilación.

Algunas herramientas externas como drawable importer la cual permite seleccionar un formato de imagen, especificar su color, cambiar el nombre de destino y seleccionar todas las resoluciones que se desea usar. Presentaban incompatibilidad con la nueva versión por lo que se optó por buscar esta herramienta con ajustes adicionales por partes de usuario de internet para poder hacer uso de ella, ya que de otra manera hubiera sido más difícil el agregar las imágenes al prototipo diseñado.

También se dificultó al principio el proceso de programación del prototipo ya que no se tenían conocimientos previos del entorno Android Studio es por ello que se adquirió un curso el cual explicaba las bases de la programación en este entorno y cuál era la sintaxis que se ocupaba normalmente al observar el curso puede notar que era parecido a lenguaje de programación e java solo que este presentaba distinta sintaxis en algún comando.

Todos los problemas mencionados anteriormente ocasionaban que el proyecto sufriera pequeños retrasos si bien no excedían el tiempo definido es importante tomar en cuenta la versiones y la incompatibilidad con algunas herramientas en el entorno de Android Studio ya que puede generar error especialmente si se requiere retomar el prototipo para mejorarlo y escalarlo, Con la ayuda de la etapa de análisis y la consola de desarrolladores se pudieron solucionar estos errores de forma satisfactoria.

El aporte de esta investigación describe cada uno del proceso que se tomó en cuenta para resolver la pregunta de investigación planteada ya que se cuenta con un marco teórico y una extensa investigación sobre aplicaciones enfocadas a las rutas como lo es el tuzobus es por ello que si se desea continuar con el proyecto este documento permitirá poder visualizar el desarrollo de la aplicación ya que se anexa los códigos y las funciones principales para la ejecución del programa; Se utilizó una aplicación móvil ya que es más fácil de realizar no consume tanto tiempo como lo haría un servicio o página web.

Para concluir se menciona que el prototipo abre múltiples posibilidades para nuevas funciones, opciones, optimizaciones, ideas, etc. relacionadas con el mejoramiento de la aplicación del transporte público, algunos ejemplos podrían ser:

- Análisis de datos: la cantidad de datos obtenidos a través de la aplicación, como el número de usuario que podrían hacer uso de ella, ubicaciones más concurridas, etc. se pueden utilizar para generar informes, estadísticas, etc.
- Prioridad de usuario: puede establecer una cola de prioridad para los usuarios que tienen un tiempo de espera más largo para brindar una mejor calidad de servicio y oportunidades más justas.
- Mejorar la aplicación: puede mejorar los aspectos funciones, como visualizar la información acerca del trayecto del viaje la distancia que se va a recorrer, el tiempo que tomara llegar etc, de la misma forma se puede crear una versión compatible con IOS del sistema operativo.
- Separar la aplicación: Como el mismo nombre lo indica se podrían separar la aplicación en 2 para que una se enfoque principalmente en solo los conductores y otra se enfoque en la del pasajero sin la necesidad de que estén junta.
- Actualizar rutas: En esta función se podría agregar o modificar las rutas sin la necesidad de que estén definidas las cuales se podían visualizar en ambas pantallas
- Implementar opciones de pago: poder utilizar la aplicación para realizar el pago del viaje a través de dinero electrónico evitando el contacto físico o el intercambio de efectivo.

Estas son algunas ideas en las que se podría escalar el prototipo ya que al ser una aplicación enfocada al transporte público de alguna manera se puede implementar y adaptar la aplicación según sean las necesidades de los pasajeros del municipio de Tizayuca Hgo, ya que estas plataformas y herramientas en su conjunto proporcionaron una arquitectura que adaptativa según el cómo se ha desarrollado la aplicación móvil. De la misma forma, menciono la importancia de que el proyecto es innovador ya que por si solo ofrece funciones que resuelva una problemática en concreto la cual es informar al pasajero el tiempo y la distancia que hay entre el y el vehículo.

## BIBLIOGRAFÍA

- colaboradores de Wikipedia. (s. f.). Geolocalización - Wikipedia, la enciclopedia libre. Geolocalización. Recuperado 23 de septiembre de 2021, de <https://es.wikipedia.org/wiki/Geolocalizaci%C3%B3n>

- Servicio de Transporte Publico en Tizayuca, Hidalgo. (2021, 5 enero). Servicio de Transporte. <https://serviciodetransporte.net/transporte-publico/hidalgo/tizayuca/>
- Condiciones del Servicio de las API de Google Maps. (s. f.). Google Developers. Recuperado 7 de octubre de 2021, de <https://developers.google.com/maps/terms-20180207?hl=es-419>
- Introducción a Android Studio | Desarrolladores de Android. (s. f.). Android Developers. Recuperado 7 de octubre de 2021, de <https://developer.android.com/studio/intro?hl=es-419>
- Cardona, M. P. (2020, 8 enero). Firebase, qué es y para qué sirve la plataforma de Google. Thinking for Innovation. <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarroladores-google-seo-sem/>
- Modelo vista controlador (MVC). Servicio de Informática ASP.NET MVC 3 Framework. (s. f.). MVC. Recuperado 7 de octubre de 2021, de [https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html#:~:text=Modelo%20Vista%20Controlador%20\(MVC\)%20es,control%20en%20tres%20componentes%20distintos.&text=La%20Vista%2C%20o%20interfaz%20de,los%20mecanismos%20interacci%C3%B3n%20con%20C3%A9ste.](https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html#:~:text=Modelo%20Vista%20Controlador%20(MVC)%20es,control%20en%20tres%20componentes%20distintos.&text=La%20Vista%2C%20o%20interfaz%20de,los%20mecanismos%20interacci%C3%B3n%20con%20C3%A9ste.)
- del Medico, F. (2020, 29 mayo). ¿Qué son las API de Google? Descúbrelo ahora mismo. Maplink. Recuperado 7 de diciembre de 2021, de <https://maplink.global/blog/es/que-son-google-apis/>
- U. (2018, 10 agosto). *5 apps para moverte en transporte público por el nuevo #HoyNoCircula*. unocero. Recuperado 7 de diciembre de 2021, de <https://www.unocero.com/noticias/5-apps-para-moverte-en-transporte-publico-por-el-nuevo-hoynocircula/>
- Usa claves de API | Autenticación |. (s. f.). Google Cloud. Recuperado 7 de diciembre de 2021, de [https://cloud.google.com/docs/authentication/api-keys?hl=es&visit\\_id=637746499245363759-2378501096&rd=1](https://cloud.google.com/docs/authentication/api-keys?hl=es&visit_id=637746499245363759-2378501096&rd=1)