



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA
CARRERA DE INGENIERÍA MECÁNICA AUTOMOTRIZ

**DISEÑO DE UNA APP PARA SISTEMA OPERATIVO ANDROID, PARA EL
MONITOREO EN TIEMPO REAL Y RASTREO DE UN AUTOMÓVIL**

Trabajo de titulación previo a la obtención del
título de Ingeniero Mecánico Automotriz

AUTORES: JAVIER DAVID LEÓN CALLE

HENRY ISRAEL YUNGA PAÑI

TUTOR: ING. CHRISTIAN OMAR PULLA MOROCHO, MSc.

Cuenca - Ecuador

2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Javier David León Calle con documento de identificación N° 0302263868 y Henry Israel Yunga Pañi con documento de identificación N° 0104756846; manifestamos que:

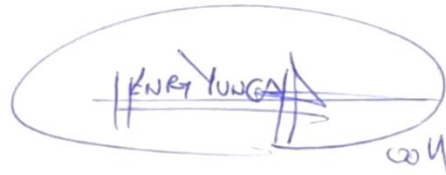
Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 18 de abril del 2022

Atentamente,



Javier David León Calle
0302263868



Henry Israel Yunga Pañi
0104756846

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO
DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, Javier David León Calle y León Calle con documento de identificación N° 0302263868 y Henry Israel Yunga Pañi con documento de identificación N° 0104756846, expresamos nuestra voluntad y por medio de este documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: “Diseño de una APP para sistema operativo Android, para el monitoreo en tiempo real y rastreo de un automóvil”, el cual ha sido desarrollado para optar el título de: Ingeniero Mecánico Automotriz, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer los derechos cedidos anteriormente.

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

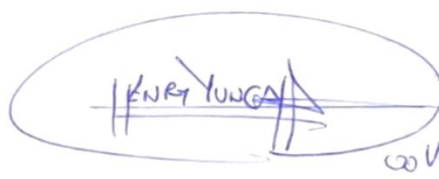
Cuenca, 18 de abril del 2022

Atentamente,



Javier David León Calle

0302263868



Henry Israel Yunga Pañi

0104756846

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Christian Omar Pulla Morocho con documento de identificación N° 0103570602, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO DE UNA APP PARA SISTEMA OPERATIVO ANDROID, PARA EL MONITOREO EN TIEMPO REAL Y RASTREO DE UN AUTOMÓVIL, realizado por Javier David León Calle con documento de identificación N° 0302263868 y por Henry Israel Yunga Pañi con documento de identificación N° 0104756846, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 18 de abril del 2022

Atentamente,



Ing. Christian Omar Pulla Morocho, M.Sc.

0103570602

AGRADECIMIENTO

En primer lugar, los autores del presente trabajo de investigación expresan sus agradecimientos a Dios por otorgarles la sabiduría necesaria para culminar con la carrera, en segundo lugar, a todos los profesores de la Universidad Politécnica Salesiana por brindar todos sus conocimientos, de forma especial al Ing. Christian Pulla como tutor de este trabajo por su apoyo incondicional y su predisposición.

LEÓN CALLE JAVIER DAVID

DEDICATORIA

En primera instancia dedico mi trabajo de titulación a Dios, a mis padres por brindarme todo su apoyo, amor, trabajo y sacrificio en todos estos años, ya que, gracias a ustedes he llegado a culminar esta etapa tan importante de mi vida. A mi abuelita, por siempre confiar en mí y apoyarme de manera incondicional a pesar de todas las circunstancias, a todos mis familiares y amigos.

Finalmente, de manera especial, este proyecto va dedicado a mi tía abuela quien me incentivo a no decaer, a mi padre por ser el pilar fundamental para todo este proceso, mi hermano menor y mi novia quienes son el motor de mi vida.

LEÓN CALLE JAVIER DAVID

DEDICATORIA

Este trabajo de titulación lo dedico a mis padres quienes con su amor, paciencia y esfuerzo han permitido cumplir hoy con un sueño más, gracias por enseñarme el esfuerzo, sacrificio y valentía para no temer a ninguna adversidad con la ayuda del Todo poderoso.

A mis hermanos, amigos y demás familiares que han estado conmigo en este arduo camino y me han apoyado.

YUNGA PAÑI HENRY ISRAEL

RESUMEN

El presente trabajo de investigación se basó en el desarrollo, diseño e implementación de una aplicación móvil para el rastreo y monitoreo de un vehículo en tiempo real desde un dispositivo móvil como, celulares o tablets para el sistema operativo Android, para que, los propietarios tengan el control de su vehículo, debido al incremento de la delincuencia, en el Ecuador según los datos proporcionados por el Instituto Nacional de Censos, que se da de dos modos operandi, robo total o parcial.

En primer lugar, se realizó un estudio basado en la revisión bibliográfica sobre todas las definiciones y conceptualizaciones de las herramientas utilizadas para el desarrollo de la aplicación móvil.

En segundo lugar, se desarrolló la aplicación móvil para el rastreo y monitoreo en tiempo real en un dispositivo móvil, que se implementó las cámaras de vigilancia en el vehículo Aveo Emotion 2015, además de ejecutar todas las modificaciones y correcciones necesarias.

En tercer lugar, se realizó las pruebas de la aplicación móvil desde diferentes dispositivos móviles con sistema operativo Android, pudiendo verificar su funcionamiento.

En la presente investigación se utilizó diferentes metodologías de esta forma: revisión bibliográfica para obtener la información necesaria para la realización de la parte teórica, método inductivo, que permitió proporcionar la información necesaria para comprender el desarrollo adecuado de los sistemas antirrobo, el método analítico que facilitó estudiar los componentes necesarios para el desarrollo de la aplicación, el método tecnológico lo que permitió dar una solución viable a la problemática planteada, es decir, la aplicación de la teoría en la práctica lo que beneficia a los propietarios de los automotores y por último la metodología experimental que nos ayudó para el desarrollo e implementación de la app en el vehículo.

SUMMARY

This research work was based on the development, design and implementation of a mobile application for tracking and monitoring of a vehicle in real time from a mobile device such as cell phones or tablets for the Android operating system, so that owners have control of their vehicle, due to the increase in crime in Ecuador according to data provided by the National Census Institute, which occurs in two modes of operation, total or partial theft.

First, a study was conducted based on a literature review of all definitions and conceptualizations of the tools used for the development of the mobile application.

Secondly, the mobile application was developed for real-time tracking and monitoring on a mobile device, which was implemented surveillance cameras in the vehicle Aveo Emotion 2015, in addition to executing all the necessary modifications and corrections.

Thirdly, the testing of the mobile application was performed from different mobile devices with Android operating system, being able to verify its operation.

In the present research, different methodologies were used in this way: bibliographic review to obtain the necessary information for the realization of the theoretical part, inductive method, which allowed to provide the necessary information to understand the proper development of anti-theft systems, the analytical method that facilitated the study of the necessary components for the development of the application, the technological method which allowed to give a viable solution to the problem posed, that is, the application of the theory in practice which benefits the owners of the automobiles and finally the experimental methodology that helped us for the development and implementation of the app in the vehicle.

ÍNDICE GENERAL

<i>CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA</i>	<i>I</i>
<i>CESIÓN DE DERECHOS DE AUTOR</i>	<i>II</i>
<i>CERTIFICADO</i>	<i>III</i>
<i>AGRADECIMIENTO</i>	<i>IV</i>
<i>DEDICATORIA</i>	<i>V</i>
<i>DEDICATORIA</i>	<i>VI</i>
<i>RESUMEN</i>	<i>VII</i>
<i>SUMMARY</i>	<i>VIII</i>
<i>ÍNDICE GENERAL</i>	<i>IX</i>
<i>ÍNDICE DE FIGURAS</i>	<i>X</i>
<i>ÍNDICE DE ANEXOS</i>	<i>XII</i>
<i>INTRODUCCIÓN</i>	<i>1</i>
<i>PROBLEMA</i>	<i>2</i>
<i>IMPORTANCIA Y ALCANCES</i>	<i>3</i>
<i>OBJETIVOS</i>	<i>4</i>
<i>Objetivo General</i>	<i>4</i>
<i>Objetivos Específicos</i>	<i>4</i>
<i>1 Capítulo 1: Introducción</i>	<i>5</i>
<i>1.1 Estado de arte</i>	<i>5</i>
<i>1.1.1 La inseguridad de la Industria automotriz</i>	<i>5</i>
<i>1.2 La Inseguridad de la Industria automotriz en el Ecuador</i>	<i>6</i>
<i>1.3 Sistemas Antirrobo</i>	<i>7</i>
<i>1.4 Sistema de Vigilancia</i>	<i>7</i>
<i>1.4.1 GPS</i>	<i>7</i>
<i>1.4.2 Cámaras de Vigilancia</i>	<i>8</i>
<i>1.5 Aplicación de monitoreo y rastreo en tiempo real de un vehículo</i>	<i>8</i>
<i>1.6 Raspberry Pi</i>	<i>10</i>
<i>1.7 Componentes de la aplicación</i>	<i>10</i>
<i>1.7.1 GPS</i>	<i>11</i>
<i>1.7.2 Mifi</i>	<i>14</i>
<i>2 Capítulo 2</i>	<i>15</i>
<i>Desarrollo de la aplicación móvil e instrumentación</i>	<i>15</i>

2.1 Desarrollo de la aplicación	15
2.2 Implementación de cámaras y Gps	35
3 Capítulo 3	39
Pruebas de rastreo y monitoreo de la aplicación en diferentes tipos de dispositivos móviles con sistema operativo Android	39
CONCLUSIONES	45
RECOMENDACIONES	46
BIBLIOGRAFÍA	60
ANEXOS	63

ÍNDICE DE TABLAS

Tabla 1: Herramientas para la elaboración de la app	9
--	---

ÍNDICE DE FIGURAS

Figura 1.1: Sistema Antirrobo	7
Figura 1.2: Esquema Básico del GPS	8
Figura 1.3: Cámara de Vigilancia	8
Figura 1.4: Características del Raspberry Pi	10
Figura 1.5: Características del GPS	11
Figura 1.6: Características del GSM	12
Figura 1.7: Características del GPRS	12
Figura 1.8: Características del Bluetooth	13
Figura 1.9: Características del MISIC	13
Figura 1.10: Funcionamiento del Mifi	14
Figura 2.1: Android Studio	15
Figura 2.2: Com.tesis.tesis	16
Figura 2.3: Layout	16
Figura 2.4: Private fun login	17
Figura 2.5: Main Activity	17
Figura 2.6: Menu_inicio.kt	18
Figura 2.7: Menú Inicio	18
Figura 2.8: Activity menú cámaras	19
Figura 2.9: Menú cámaras	20
Figura 2.10: Activity video view 1	20
Figura 2.11: Activity video view 1	21
Figura 2.12: Aplicación Móvil	22
Figura 2.13: Api Key	22
Figura 2.14: Api Key	23
Figura 2.15: Api Key Creación	23
Figura 2.16: Api Key Menú principal	24
Figura 2.17: Habilidad de la Api Key	24

Figura 2.18: Api Key Maps	25
Figura 2.19: Maps SDK	25
Figura 2.20: API, servicios y credenciales	26
Figura 2.21: Creacion de la credencial y la clave de la Api	26
Figura 2.22: Clave de la Api.....	27
Figura 2.23: Creacion de una librería (dependencia)	27
Figura 2.24: Values Resource File	27
Figura 2.25: AndroidManifest.....	28
Figura 2.26: Creación del Fragment blank.....	28
Figura 2.27: Creación de Archivos	29
Figura 2.28: MapsFragment.....	29
Figura 2.29: Realización del Código	29
Figura 2.30: Código de fragment	30
Figura 2.31: ID de la API maps	30
Figura 2.32: ID de la API maps	30
Figura 2.33: Instalación de la motion	32
Figura 2.34: Instalación de la motion	32
Figura 2.35: Fichero Principal	33
Figura 2.36: Configuración de las Cámaras.....	33
Figura 2.37: Target.....	34
Figura 2.38: Cámaras	34
Figura 2.39: Modificación de las Cámaras	35
Figura 2.40: WEB de las cámaras.....	35
Figura 2.41: Desmontaje de mascarilla	36
Figura 2.42: Desmontaje del Tablero	36
Figura 2.43: Cableado de las cámaras	37
Figura 2.44: Ubicación de las cámaras.....	37
Figura 2.45: Instalacion del GPS	37
Figura 2.46: Instalacion de la Raspberry Pi.....	37
Figura 3.1: Usuario y contraseña de Autocontigo	39
Figura 3.2: Menú Inicio	40
Figura 3.3: Menú Camaras.....	40
Figura 3.4: Geolocalización del vehículo.....	41
Figura 3.5: Geolocalización precisa del vehículo	41
Figura 3.6: Geolocalización del vehículo.....	41
Figura 3.7: Cámara 3	40
Figura 3.8: Cámara 4	40
Figura 3.9: Geolocalización del vehiculo.....	40
Figura 3.10: Cámara 3	40
Figura 3.11: Cámara 1	40

ÍNDICE DE ANEXOS

Anexo 1: Dispositivo Mifi	63
Anexo 2: Batería Externa	63
Anexo 3: Batería de alimentación y Raspberrypi	64
Anexo 4: Instalación de la Aplicación en un dispositivo Xiaomi	64
Anexo 5: Diferentes modelos de Smartphone utilizados en las pruebas	65
Anexo 6: Diagrama de flujo de la programación	65
Anexo 7: Programacion del MainActivity	66
Anexo 8: Programación de MapsActivity	67
Anexo 9: Programación del Menú Cámaras	70
Anexo 10: Programación del Menú Inicio	71
Anexo 11: Programación del VideoView	73
Anexo 12: Programación del Android Manifest	74
Anexo 13: Programación del Activity_main	76
Anexo 14: Programación del Activity_maps	78
Anexo 15: Programación de activity_menu_camaras	79
Anexo 16: Programación del Activity_menu_inicio	81
Anexo 17: Activity_video_view	83

INTRODUCCIÓN

La importancia de este proyecto de investigación; radica en desarrollar una aplicación que permita, monitorear y rastrear en tiempo real el automóvil, registrando su ubicación y su condición de funcionamiento, el mismo que puede ser utilizado en cualquier tipo de vehículo, y solo será necesario: instrumentar, descargar la App en un dispositivo móvil, crear un usuario, contraseña y enlazarla con el vehículo.

La App brindará otros servicios adicionales a los usuarios que no solo protejan la integridad del vehículo, sino también permitirá un seguimiento en tiempo real del automóvil, para evidenciar su velocidad, y no se infrinja la normativa de tránsito, otra ventaja de esta aplicación es que, en el supuesto caso de secuestro o robo del vehículo en su totalidad, permitirá reconocer al autor para procesos de investigación.

Además, como se instalarán las cámaras de vigilancia en el interior y exterior del automóvil, en un posible siniestro o accidente, se podrá entregar a las autoridades competentes el video registrado por las cámaras que ayudarían en el proceso legal, al aprovechar la era tecnológica actual.

PROBLEMA

Según el Instituto Nacional de Estadísticas y Censos, en los últimos tres años en Ecuador, ha existido un incremento del 23% de robos de vehículos parciales y en su totalidad, lo que implica un riesgo inminente que se da en la mayoría de los casos en ausencia de su propietario o en horarios nocturnos, siendo la vía pública el medio propicio para su ejecución. (INEC, 2020)

En consecuencia, las víctimas de estos delitos han optado por implementar ciertos mecanismos que impidan el cometimiento de dichos actos, tales como: sistemas de alarma, sistema de bloqueo, láminas de seguridad, GPS, etc., las cuales no han podido inhabilitar que se sigan cometiendo los hechos ilícitos, ya que, los antisociales han desarrollado métodos para eludir dichos sistemas de seguridad.

Las herramientas tecnológicas constituyen la base fundamental para las interacciones sociales, en este contexto, las medidas o mecanismos de seguridad que existen en el mercado no son adecuadas, porque no han sido desarrolladas en su totalidad para satisfacer los requerimientos del propietario.

La utilización de dispositivos inteligentes Smartphone o tablets han permitido que exista la disponibilidad de varias aplicaciones con diversas finalidades como: sociales, culturales, económicas o entretenimiento, sin enfocarse en la seguridad de los vehículos, ya que, no existe una aplicación que satisfaga la necesidad y permita al usuario conectarse con su automotor, en todo momento, aprovechando de los beneficios que la tecnología brinda y sobre todo que resulte económico y autónomo (Báez y Cabrera, 2010).

Se debe recalcar, que existen diversas aplicaciones de seguridad ya desarrolladas en el mercado, como, por ejemplo: rastreo satelital Ecuador, rastreo satelital vehicular, auto conectado smartag, la mayoría de estas trabajan con seguros, lo cual eleva los costos para adquirirla, en consecuencia, existen varios problemas, ya que, es necesario contratar un seguro, donde les permite tener toda la información del vehículo, y únicamente reportándole al usuario la ubicación.

IMPORTANCIA Y ALCANCES

En este orden de ideas, el diseño y la aplicación que se desarrollará parte de tres premisas fundamentales que son:

1) La creación de una App móvil intuitiva, invulnerable, autónoma e indescifrable entre el dispositivo móvil y el microcontrolador.

2) Para un ambiente de uso más amigable; permite aumentar la rapidez de comunicación entre el teléfono celular o tablet con el automóvil, accediendo a ello a través un simple manejo, utilizando un usuario y contraseña, y

3) Será una aplicación gratuita, sin embargo, vale recalcar que existirán costos para su implementación, los que deberán ser cubiertos por el beneficiario, disponible para el sistema operativo Android y apta para cualquier tipo de vehículo, que se irá actualizando en función a la evolución tecnológica.

4) La propuesta de solución es desarrollar e implementar una App para el sistema operativo Android, para ser utilizado en cualquier tipo de vehículo, permitiendo así obtener una interacción en tiempo real con el automóvil, la cual contará con las siguientes características:

- La aplicación será autónoma, es decir, que sea manipulada solo por el dueño del automotor.
- Se implementará un sistema de cámaras de vigilancia que permita monitorear el interior y exterior del vehículo en tiempo real, utilizando el wifi.
- La aplicación contará con un sistema de usuario y contraseña para evitar que pueda ser hackeado y sea solo utilizado por el dueño del automóvil.
- Sistema con alimentación independiente a la batería del vehículo disponible las 24 horas, la cual garantizará el funcionamiento del sistema de seguridad, en el caso de que exista un corte de alimentación de la batería

OBJETIVOS

Objetivo General

Diseñar una App para sistema operativo Android, para el monitoreo en tiempo real y rastreo de un automóvil.

Objetivos Específicos

- Redactar un marco conceptual a través de revisión bibliográfica y de estudios previos, para la realización del proyecto de investigación.
- Diseñar la aplicación móvil e instrumentación de equipos necesarios.
- Realizar pruebas de rastreo y monitoreo de la aplicación en diferentes tipos de dispositivos móviles con sistema operativo Android.

Estudio bibliográfico de los componentes para el desarrollo de una aplicación móvil

1.1 Estado de arte

1.1.1 La inseguridad de la Industria automotriz

La palabra inseguridad automotriz se refiere al delito cometido contra los vehículos, es decir, el robo, tráfico con vehículos, venta ilícita de diversas piezas, que afectan directamente a los propietarios de los automotores y a la seguridad pública en general, que es un fenómeno que se da en todos los Estados.

La delincuencia organizada de vehículos, no solo afecta a los propietarios, también implica una responsabilidad de las compañías de seguros, siempre que los hayan contratado, que en la mayoría de los casos no sucede, este tipo de delitos por lo general se vinculan otros de mayor incidencia a nivel social, como por ejemplo: terrorismo, venta de sustancias estupefacientes, etc., a nivel económico, ya que, la comercialización de las piezas es una de las formas más comunes de obtener dinero de manera fácil, en denominado, mercado negro.

Por otro lado, la era tecnológica en la que nos encontramos ha sido una herramienta fundamental en la reventa de los repuestos sustraídos de los vehículos, que causa preocupación a los entes competentes como a la sociedad en general. (INTERPOL, 2020)

El área de seguridad de la industria automotriz debido al incremento de robos de vehículos ha optado por implementar sistemas inteligentes de control remoto del vehículo como las de antirrobo que va desde alarmas semi-electrónicas hasta las controladas por GPS, bloqueo de puestas, corte de suministro de combustible, y aplicaciones móviles para precautelar la seguridad de los vehículos. (Calderón, 2011)

La necesidad de desarrollar e implementar una aplicación móvil para un sistema Android para prevenir el hurto o robo de los vehículos, nace desde la carencia

de sistemas que permita tener un control y seguridad de los automotores, el sistema contara con un Raspberry pi que es esencial para la interacción con la aplicación que se desarrollará para los dispositivos, la cual ayudará a la verificación de los videos de las cámaras, ubicación del vehículo y el estado del mismo.

1.2 La Inseguridad de la Industria automotriz en el Ecuador

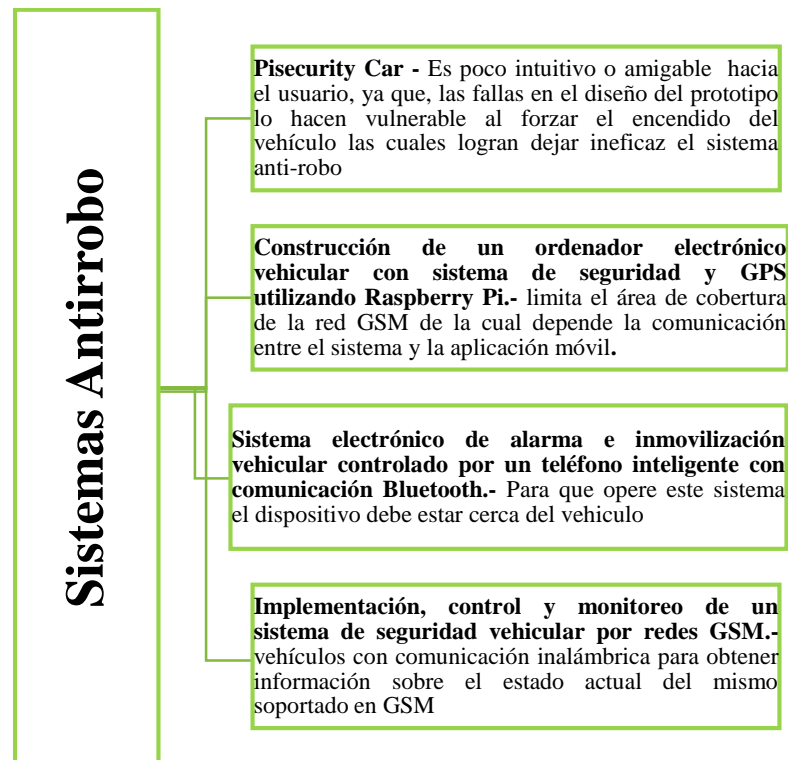
Según el Instituto Nacional de Estadísticas y Censos, en los últimos tres años en Ecuador, ha existido un incremento del 23% de robos de vehículos parciales y en su totalidad, lo que implica un riesgo inminente que se da en la mayoría de los casos en ausencia de su propietario o en horarios nocturnos, siendo la vía pública el medio propicio para su ejecución. (INEC, 2020). Para entender los elementos de la cuestión vehicular, es necesario considerar lo que afirman los autores Abril y Tupiza (2009):

Un elemento clave para entender la cuestión vehicular es la evolución del parque automotor. En este sentido, vemos que en el Ecuador a partir del año 2002 –una vez instalada la dolarización– existe un incremento importante de vehículos, sostenido principalmente por el crecimiento de créditos de consumo que posibilita la adopción de una moneada dura, y por la agresiva inversión en bienes muebles e inmuebles debido a la desconfianza en el sistema bancario. En términos generales se puede afirmar que el mercado vehicular posee una estabilidad creciente. (p. 4)

Según los datos estadísticos que la Fiscalía General del Estado ha proporcionado, podemos identificar que se trata cuando una persona, a través de la utilización de la fuerza, amenazas o intimidación, sustraiga totalmente o parcialmente un carro, ya sea en la vía pública como privada, es necesario entender que un carro puede ser: camión, automóviles, cabezales, equipos camioneros, etc. (Fiscalía General del Estado, 2021)

1.3 Sistemas Antirrobo

Es un programa que permite el acceso desde un dispositivo móvil a través de la cual se puede ejecutar diferentes acciones, en diversos campos como social, político, económico, cultural y en el automotriz no es la excepción, ya que, se han implementado varias herramientas o métodos que permiten precautelar la seguridad de los vehículos.



*Figura 1.1: Sistema Antirrobo
Fuente: Elaboración Propia*

1.4 Sistema de Vigilancia

Los sistemas de vigilancia que mayor eficacia tienen para fortalecer la seguridad de los vehículos y evitar hurtos o robos totales o parciales son:

1.4.1 GPS

Permite posicionar un objeto en la superficie de la Tierra a través de las señales emitidas en forma de ondas de radio por los satélites. Se determina la posición con una precisión en función del tipo de información recibida, tiempo de recepción y condiciones de la emisión. (Hernández, 2001, p. 6)



Figura 1.2: Esquema Básico del GPS¹

1.4.2 Cámaras de Vigilancia

El incremento de la delincuencia ha llevado a tomar alternativas diferentes con la finalidad de proteger el vehículo, por ende, el uso de las cámaras de vigilancia en el interior y exterior del vehículo permite que se registre el acto ilícito cometido, y así identificar al autor.



Figura 1.3: Cámara de Vigilancia²

1.5 Aplicación de monitoreo y rastreo en tiempo real de un vehículo







Es un programa que permite el acceso desde un dispositivo móvil mediante la cual se puede ejecutar diferentes acciones, la aplicación antirrobo tiene por objeto monitorear en tiempo real el estado del vehículo y su ubicación utilizando internet.

Por todo lo antes mencionado, el presente trabajo de investigación se desarrollará e implementará una aplicación móvil para sistema Android, la que será compatible con cualquier tipo de vehículo, su elaboración requerirá de las siguientes herramientas y elementos, descritas a continuación:

¹ <http://orestemendozalosteques.blogspot.com/2015/03/estructura-del-equipo-gps.html?m=1>

² <https://camarasdevigilancia10.info/camara-para-coche/vigilancia/>

Tabla 1: Herramientas para la elaboración de la app

<p>Redes de datos³</p> 	<p>Se denomina red de datos a aquellas infraestructuras o redes de comunicación que se ha diseñado específicamente a la Transmisión de información mediante el intercambio de datos. (Casas, 2019)</p>
<p>Internet⁴</p> 	<p>Internet es una red de computadoras que se encuentran interconectadas a nivel mundial para compartir información. (Raffino, 2020)</p>
<p>Banda ancha móvil⁵</p> 	<p>La banda ancha brinda acceso a Internet de alta velocidad a través de múltiples tipos de tecnologías, incluidas la fibra óptica, el servicio móvil, el servicio de conexión por cable, por DSL y por satélite.</p>
<p>Tecnología 4g⁶</p> 	<p>Esta tecnología requiere de infraestructura especial que actualmente está siendo desplegada por las operadoras del Servicio Móvil Avanzado para que pueda estar cubierta a escala nacional. (Agencia de Regulación y Control de Telecomunicaciones, 2020)</p>
<p>Mifi⁷</p> 	<p>"Es un router inalámbrico que admite la posibilidad de enlazar diferentes dispositivos, posee la ventaja de tener una batería interna por lo que no necesita estar conectado a una alimentación externa, acepta cualquier tarjeta SIM, nacional o internacional." (Alvarez, 2018)</p>
<p>Raspberry Pi⁸</p> 	<p>"Es un microordenador de tamaño de tarjeta de crédito, o una placa que soporta varios componentes necesarios en un ordenador común." (Pascual, 2015)</p>

Fuente: Elaboración Propia.

³ <http://alejandrofariña.com/redes-de-datos-para-negocios-y-pymes/>

⁴ <https://www.edx.org/es/course/introduccion-al-internet-de-las-cosas>

⁵ <https://www.edx.org/es/course/introduccion-al-internet-de-las-cosas>

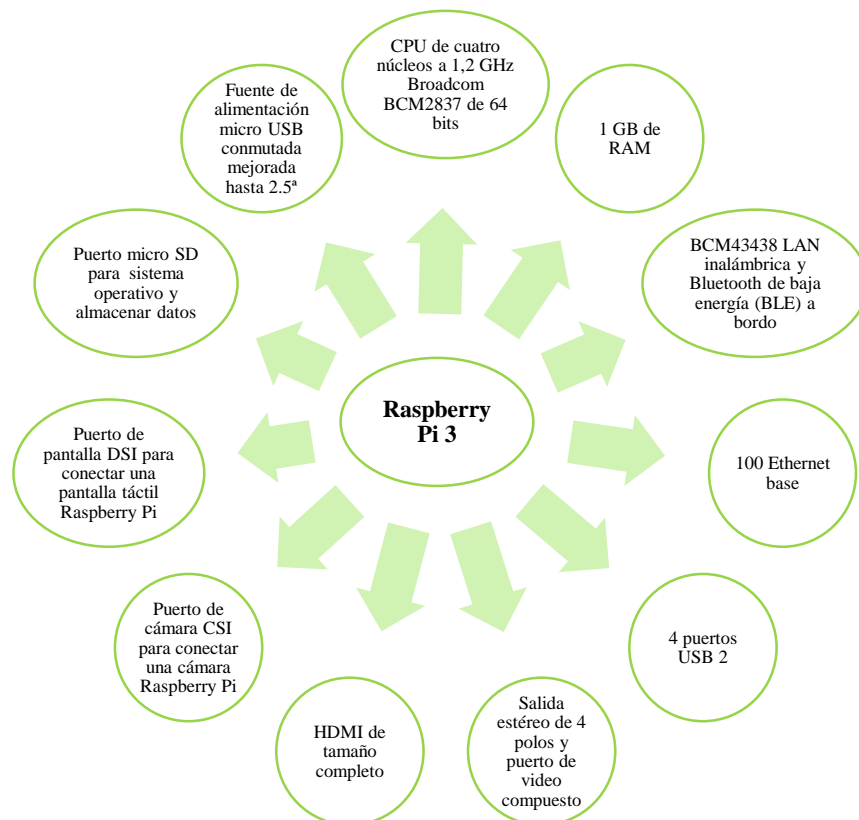
⁶ <https://www.google.com/amp/s/nacho.com.ar/que-es-4g-y-como-aprovecharlo/amp/>

⁷ <https://www.embou.com/blog/que-es-mifi-y-como-funciona>

⁸ <https://www.raspberrypi.org/>

1.6 Raspberry Pi

Un pilar en el mundo de los fabricantes y la electrónica, es la Raspberry pi definido como una computadora de una sola placa, de bajo costo y alto rendimiento, fue creada en Reino Unido por la fundación Raspberry pi cuya principal función es originar y promover la electrónica y la programación de computadoras para fines académicos, que ha ido evolucionando conforme el tiempo, puesto que, se han realizado diversas modificaciones para mejorar su capacidad, como por ejemplo: pantallas táctiles, cámaras y placas, logrando así infinidad de posibilidades cuando se trata de ideas de proyectos. (Raspberry Pi, 2021)



*Figura 1.4: Características del Raspberry Pi
Fuente: Elaboración Propia.*

1.7 Componentes de la aplicación

Para el desarrollo de la aplicación se necesita de los siguientes instrumentos, que hay que describirlos:

1.7.1 GPS

Es un práctico Raspberry Pi HAT de bajo consumo que cuenta con funciones de comunicación múltiple: GSM, GPRS, GNSS y Bluetooth, que permite a su Pi realizar sencillamente llamada telefónica, enviar mensajes, conectarse a Internet inalámbrico, posicionarse globalmente, transferir datos a través de Bluetooth, etc. (Raspberry Pi, 2021)

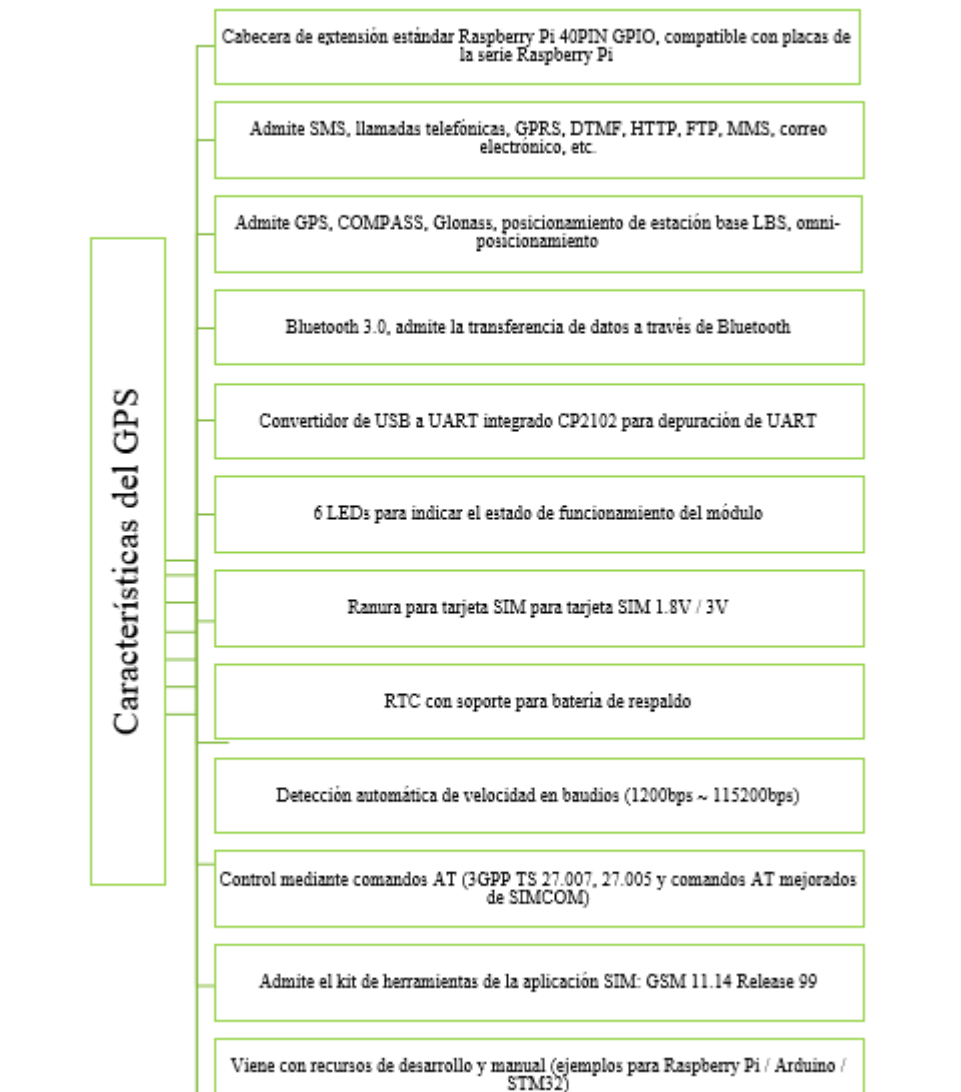


Figura 1.5: Características del GPS

Fuente: Elaboración Propia.

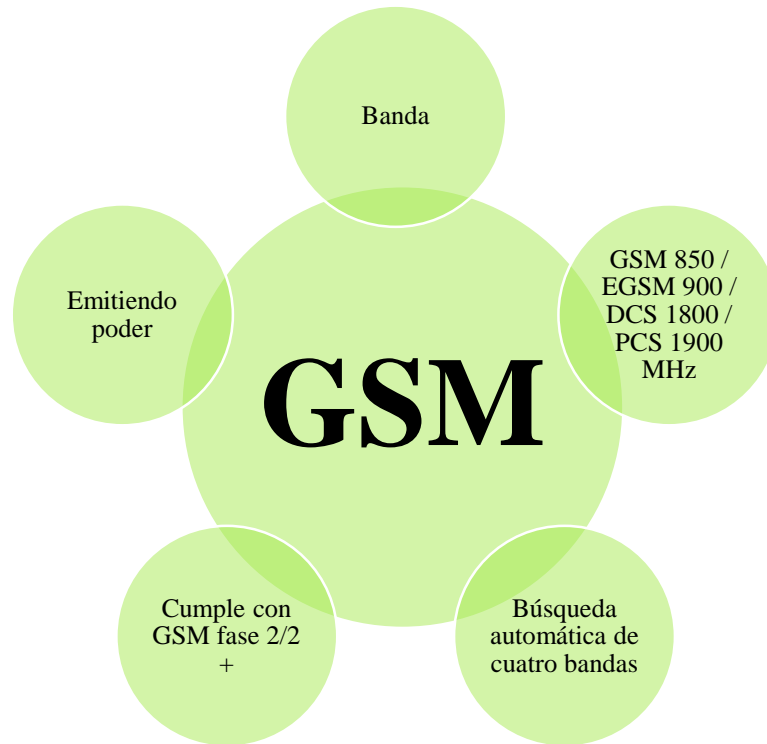


Figura 1.6: Características del GSM
Fuente: Elaboración Propia.

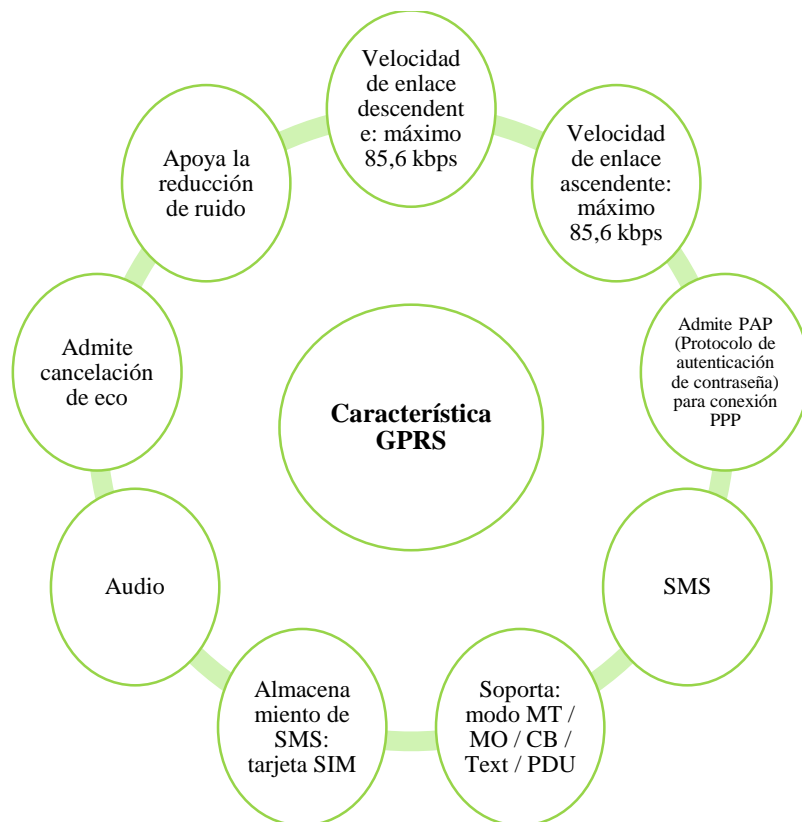
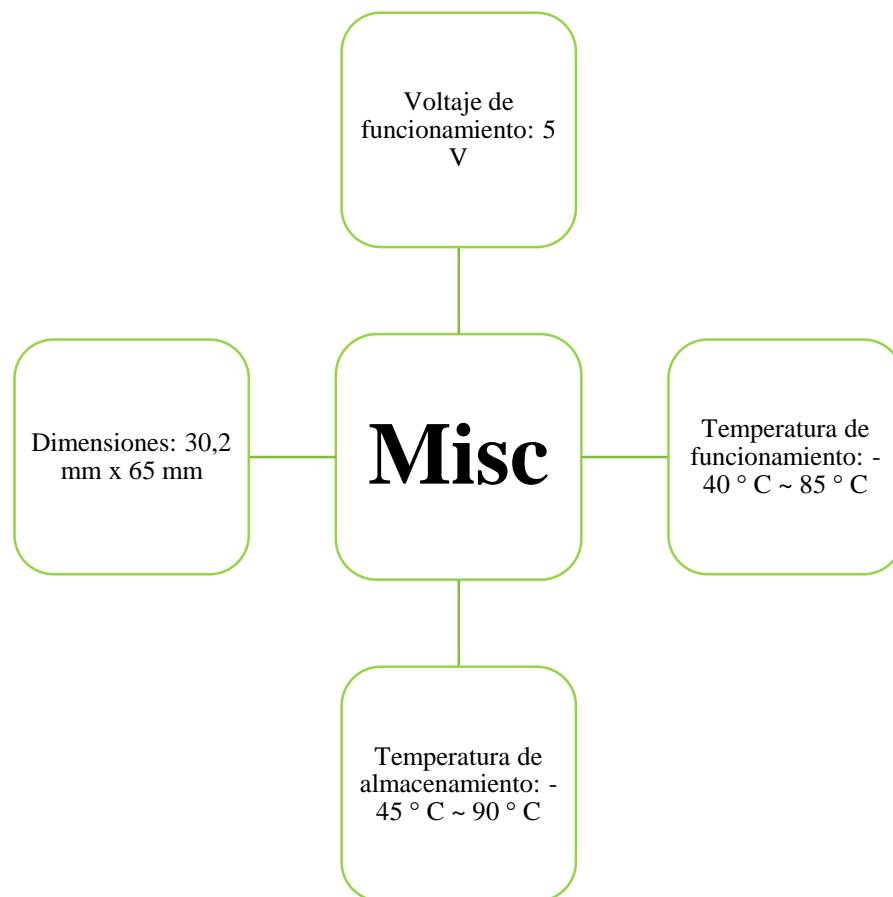


Figura 1.7: Características del GPRS
Fuente: Elaboración Propia.



*Figura 1.8: Características del Bluetooth
Fuente: Elaboración Propia.*



*Figura 1.9: Características del MISIC
Fuente: Elaboración Propia.*

1.7.2 Mifi

Es un router con conexión a Internet móvil 4G o 3G, es decir, con tarjeta SIM, puede ser utilizado como un router portátil para conectarte a Internet en cualquier lugar a precios asequibles. (Alvarez, 2018)

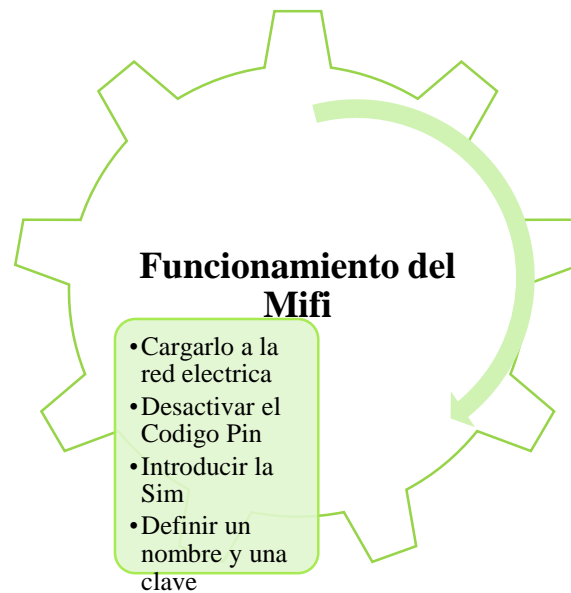


Figura 1.10: Funcionamiento del Mifi
Fuente: Elaboración Propia.

Capítulo 2

Desarrollo de la aplicación móvil e instrumentación

2.1 Desarrollo de la aplicación

El desarrollo de la aplicación se realizó en Android Studio, que es una herramienta gratuita considerada como una fuente de información que brinda varios servicios para obtener la creación de las aplicaciones, utilizando el lenguaje java y kotlin, en este caso se trabajó con kotlin; ya que, el 80% de las aplicaciones subidas a la plataforma de Googleplay están en ese lenguaje de programación.



Figura 2.1: Android Studio
Fuente: Android Studio

En la carpeta com. tesis. tesis existen los archivos MainActivity que son las funciones main como en el lenguaje java, estos archivos son los que permiten declarar las variables, funciones, métodos entre otros, con la finalidad de realizar los procesos como cambiar de Activity, o la adquisición de datos de textView.

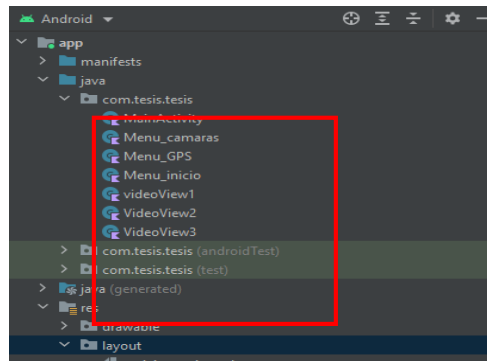


Figura 2.2: Com.tesis.tesis
Fuente: Elaboración Propia.

En la carpeta layout se encuentra toda la parte gráfica, como los botones, cajas de texto entre otros; hay que tener claro que por cada MainActivity hay un layout.

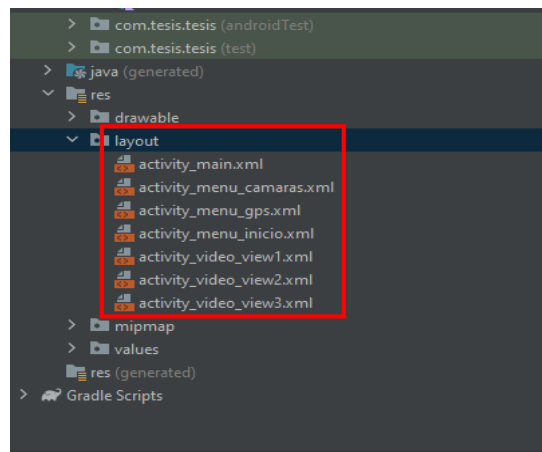


Figura 2.3: Layout
Fuente: Elaboración Propia.

En el primer fichero de com. tesis.tesis/MainActivity es el encargado de realizar el login de la aplicación, cuenta con dos funciones; la primera función “private fun login ()” permite tomar los valores de los textview que están en los layout en este caso activity_main.xml, también proporciona mensajes de notificación cuando inicies sesión o cuándo se ingresan mal los datos.

El usuario por defecto es admin@autogps.com y la contraseña es “123456”, estos valores se colocaron por defecto, debido a que en el proyecto es un prototipo no se realizó una base de datos en la cual se guarden los usuarios.

La función “private fun openMenuInicio ()” permite el cambio de activity, esto ocurre cuando el usuario y contraseña se verifique inmediatamente se cambia a

activity_menu_inicio. El Intent () permite solicitar una acción de otro componente de una app, gracias a ese objeto podemos cambiar de pantalla.

```

private fun login (){
    // se toman los valores de los textView del activity_main.xml
    val email = binding.editTextTextEmailAddress.editableText?.toString()
    val password = binding.editTextTextPassword.editableText?.toString()

    // mensajes de ayuda para verificar el login con éxito
    val toast = Toast.makeText( context: this, text: "Ingreso Exitoso",Toast.LENGTH_LONG)
    val toast1 = Toast.makeText( context: this, text: "Ingreso Denegado",Toast.LENGTH_LONG)

    // funcion logica para comprar los datos -- tipo base de datos--
    if(email == "admin@autogps.com" && password == "123456"){
        toast.show()
        opentMenuInicio()
    }else{
        toast1.show()
    }
}

// funcion en la cual se realiza el cambio de activity de MainActivity --> Manu_inicio
private fun opentMenuInicio() {
    val intent = Intent( packageContext: this,Menu_inicio::class.java)
    startActivity(intent)
}

```

Figura 2.4: Private fun login
Fuente: Elaboración Propia.

El layout del MainActivity se presenta con dos TextView que son utilizados para el usuario y contraseña, además de un button para poder realizar la acción de verificación

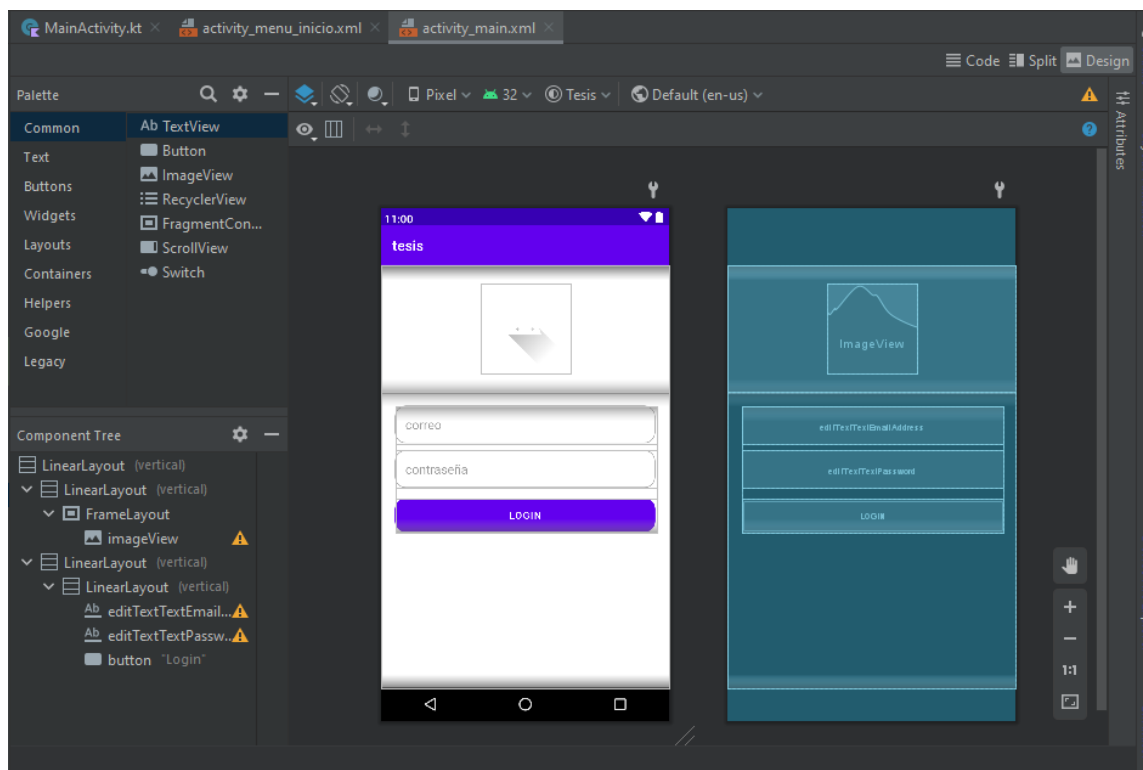


Figura 2.5: Main Activity
Fuente: Elaboración Propia.

En el Menu_inicio.tk que es una activity se presenta el menú principal donde el usuario puede acceder a las cámaras y a la geolocalización; esta activity consta de una función “private fun btnMenu ()” que es la encargada de realizar la solicitud de una acción de otro componente de la app.

```

y_menu_inicio.xml x Menu_inicio.kt x
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = ActivityMenuInicioBinding.inflate(layoutInflater)
    setContentView(binding.root)
    // se cambia el titulo de barra principal
    val actionBar = supportActionBar
    actionBar!!.title = "Menu Inicio"
    // se declara la funcion de botones
    btnMenu()
}
//actionBar.setDisplayHomeAsUpEnabled(true)
// funcion en la cual se declaran los metodos Intends para cambiar de activity
private fun btnMenu(){
    // el binding permite tomar la accion del button para ejecutar
    //permite cambiar de activity especificamente a Menu_camaras
    binding.camara.setOnClickListener { it: View!
        val intent = Intent(packageContext, Menu_camaras::class.java)
        startActivity(intent)
    }
    // permite cambiar de activity especificamente a GPS
    binding.gps.setOnClickListener { it: View!
    }
}
//permite salir de la app
binding.salir.setOnClickListener{ it: View!
    exitProcess(-1)
}
    
```

Figura 2.6: Menu_inicio.kt
Fuente: Elaboración Propia.

El layout del Menú_inicio se presenta con tres buttons, uno para acceder al menú de las cámaras, el segundo para acceder a la geolocalización y el tercero para salir de la aplicación.

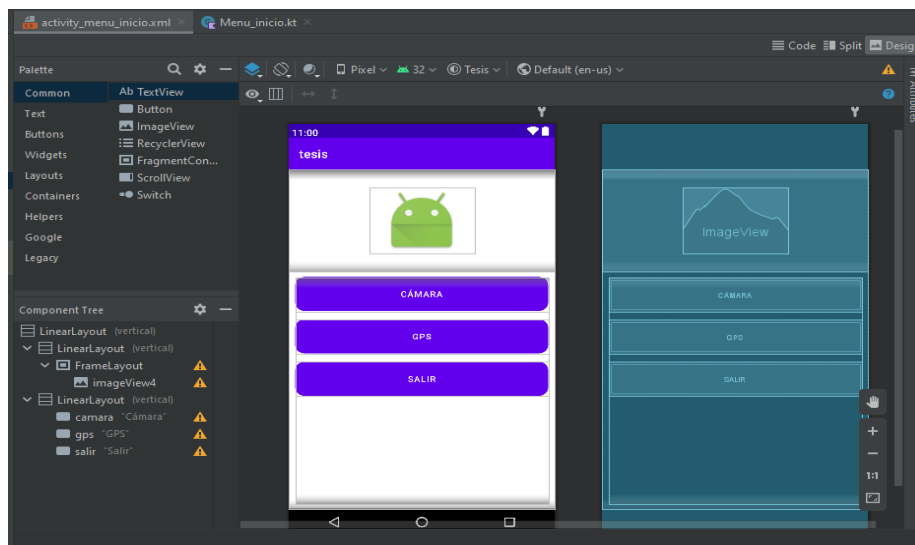


Figura 2.7: Menú Inicio
Fuente: Elaboración Propia.

Al momento que se presiona el botón salir, de manera inmediata la aplicación se cierra y aparece en la ventana principal de login.

En el caso de presionar el botón Cámaras, se dirige a una nueva activity que presenta el menú para acceder a las cámaras.

La activity Menu_Camaras consta de una sola función que consiste en realizar las llamadas a las activitys donde se muestra el video que las cámaras obtienen

```
camaras.kt x activity_menu_camaras.xml x
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = ActivityMenuCamarasBinding.inflate(layoutInflater)
    setContentView(binding.root)

    val actionBar = supportActionBar
    actionBar!!.title = "Menu Cámaras"
    //actionBar.setDisplayHomeAsUpEnabled(true)

    botones()
}
private fun botones(){
    binding.btncam1.setOnClickListener { it: View!
        val intent = Intent( packageContext: this,VideoView1::class.java)
        startActivity(intent)
    }
    binding.btncam2.setOnClickListener { it: View!
        val intent2 = Intent( packageContext: this,VideoView2::class.java)
        startActivity(intent2)
    }
    binding.btncam3.setOnClickListener { it: View!
        val intent3 = Intent( packageContext: this,VideoView3::class.java)
        startActivity(intent3)
    }
}
```

Figura 2.8: Activity menú cámaras
Fuente: Elaboración Propia.

El layout de esta activity consta de tres botones que son para acceder a la cámara 1, cámara 2 y cámara 3.

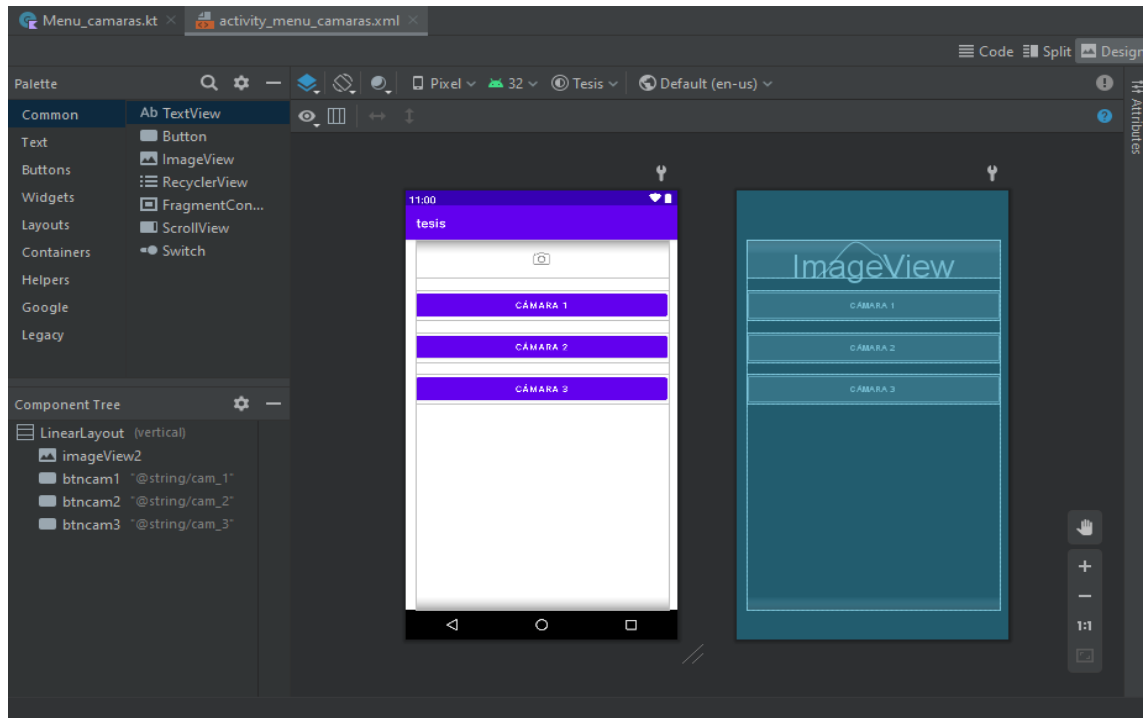


Figura 2.9: Menú cámaras
Fuente: Elaboración Propia.

En la activity VideoView X, muestra un WebView el cual permite mostrar el video que se está generando en la raspberry para lograr que el video se muestre en la app se debe cargar la URL donde se está corriendo el video, para la cámara 2 es 192.168.1.2:8082 y para la cámara 3 es 192.168.1.2:8083

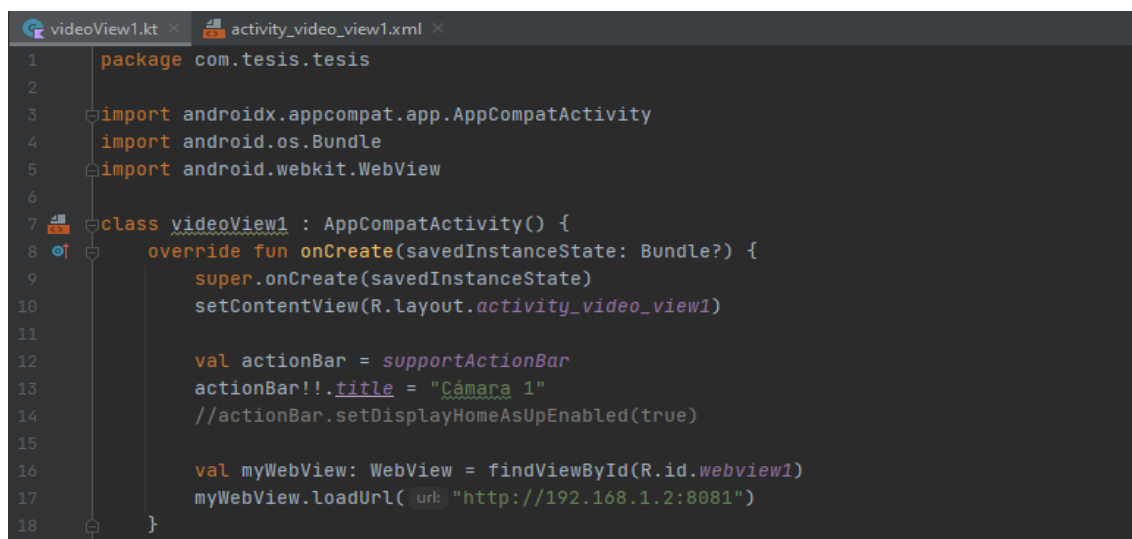


Figura 10: Activity video view 1
Fuente: Elaboración Propia.

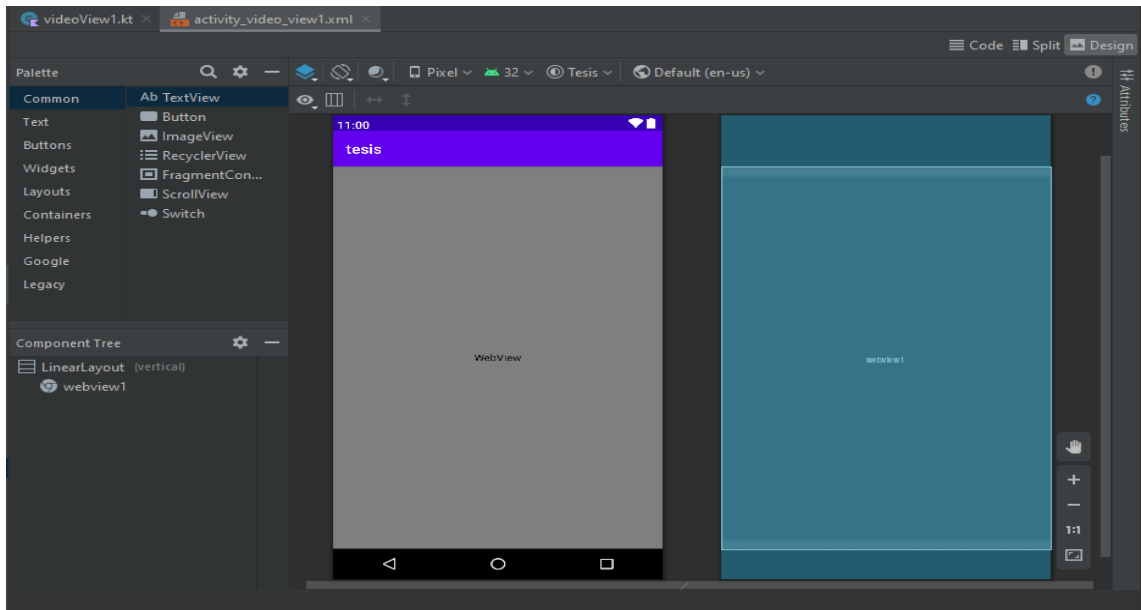
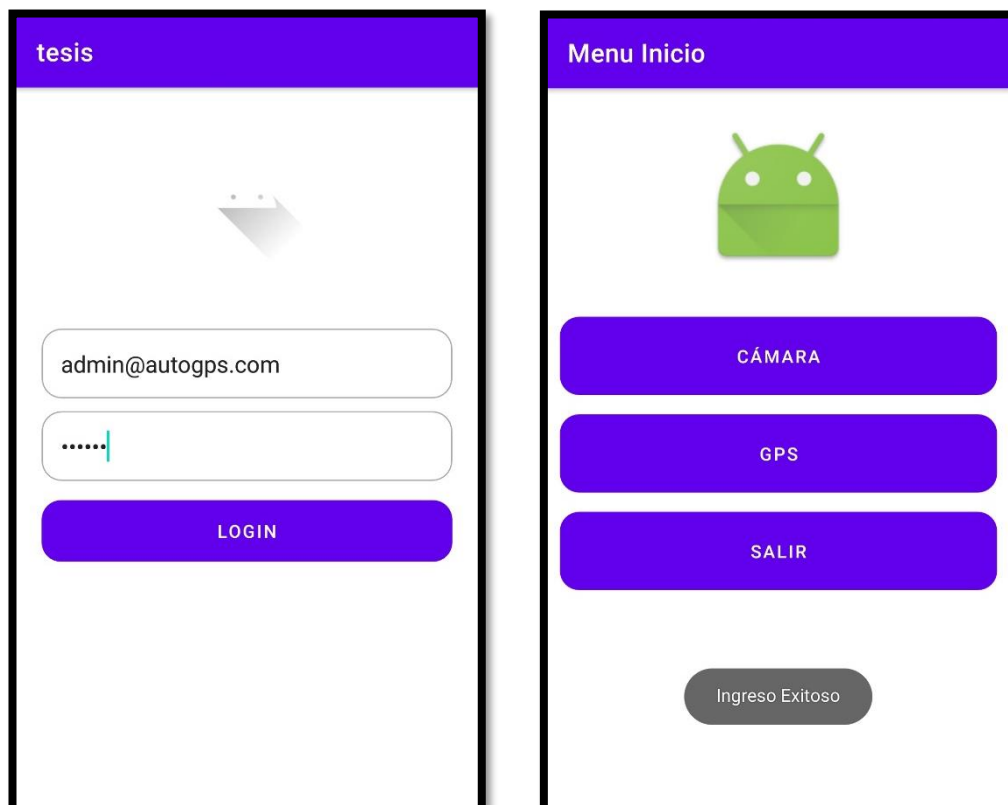


Figura 11: Activity video view 1
Fuente: Elaboración Propia.

Ahora instalando la APP en un sistema Android se presenta de la siguiente manera



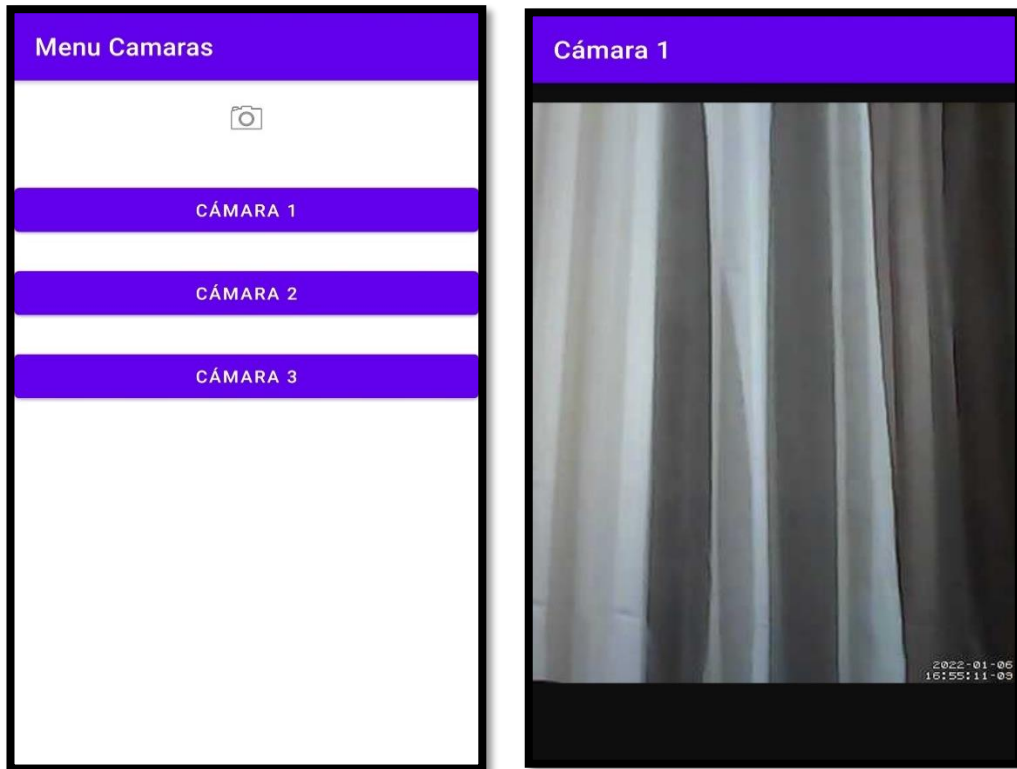


Figura 12: Aplicación Móvil
Fuente: Elaboración Propia.

Para el desarrollo de aplicaciones Android con Google maps, es necesario una **api key** la cual se obtiene mediante la URL: <https://console.cloud.google.com/>, en la cual debemos ir a la esquina superior derecha donde dice Proyecto Nuevo.

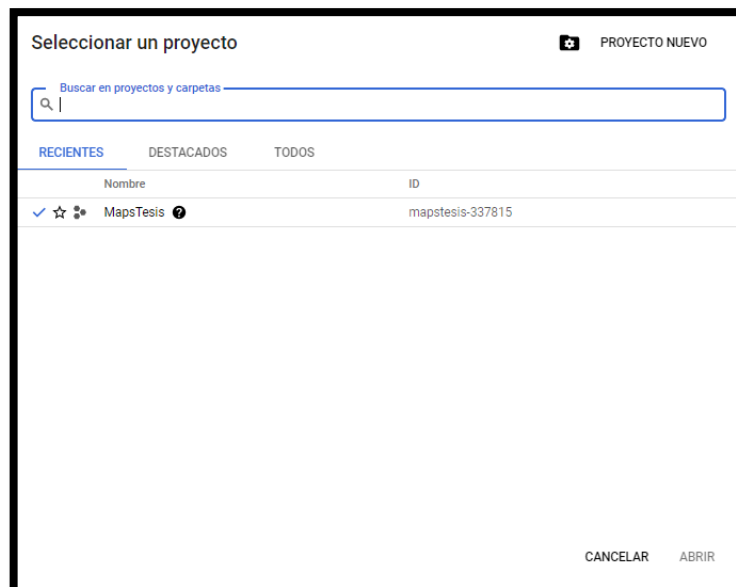


Figura 13: Api Key
Fuente: Elaboración Propia.

Esta opción redirige a otra página donde se coloca el nombre del proyecto en este caso “GpsTracking”, para este caso será “sin organización”; ya que, es una prueba del sistema, damos clic en crear.

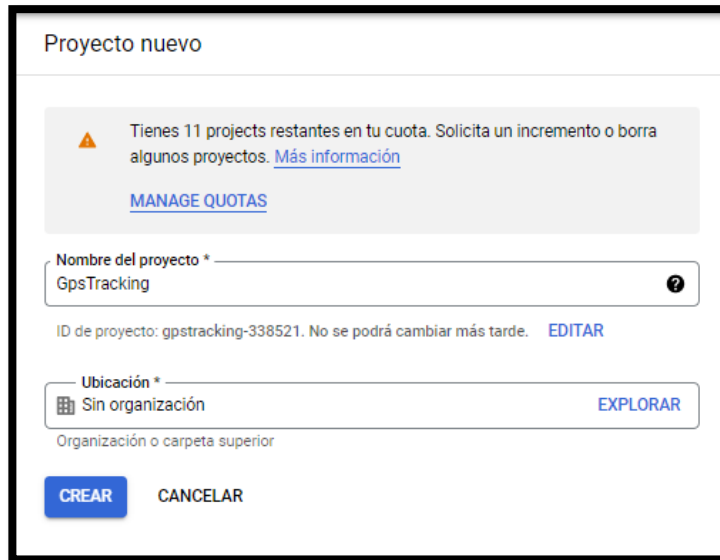


Figura 14: Api Key
Fuente: Elaboración Propia.

En la barra superior el sistema nos mostrara una notificación, seleccionaremos el proyecto.

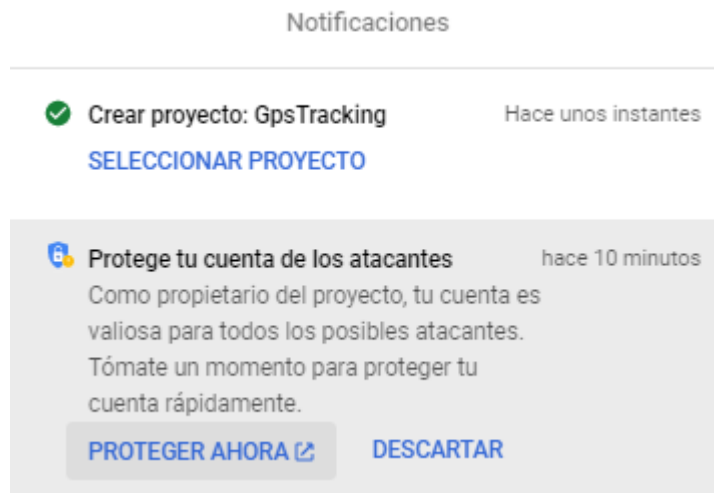


Figura 15: Api Key Creación
Fuente: Elaboración Propia.

Nos dirigimos al menú principal, Api y servicios, panel

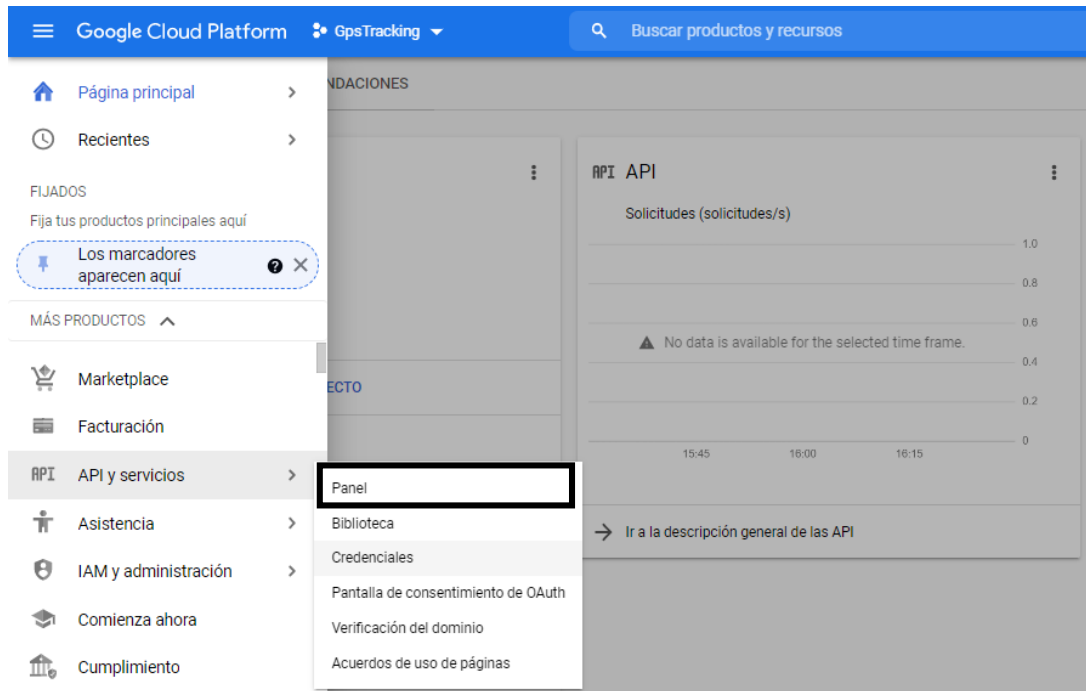


Figura 16: Api Key Menú principal

Fuente: Elaboración Propia.

Luego nos dirigimos a “HABILITAR API Y SERVICIO”

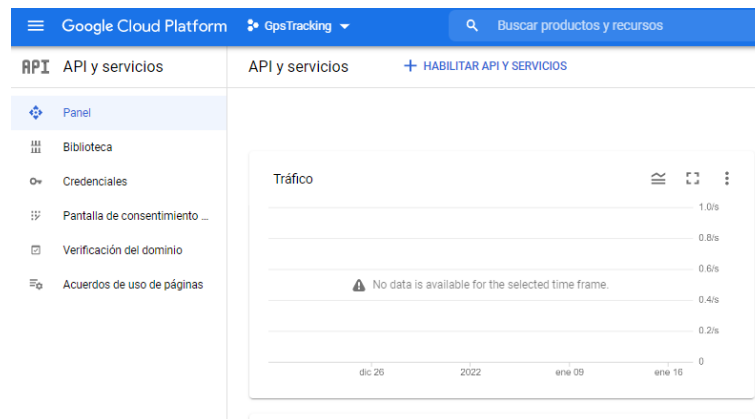
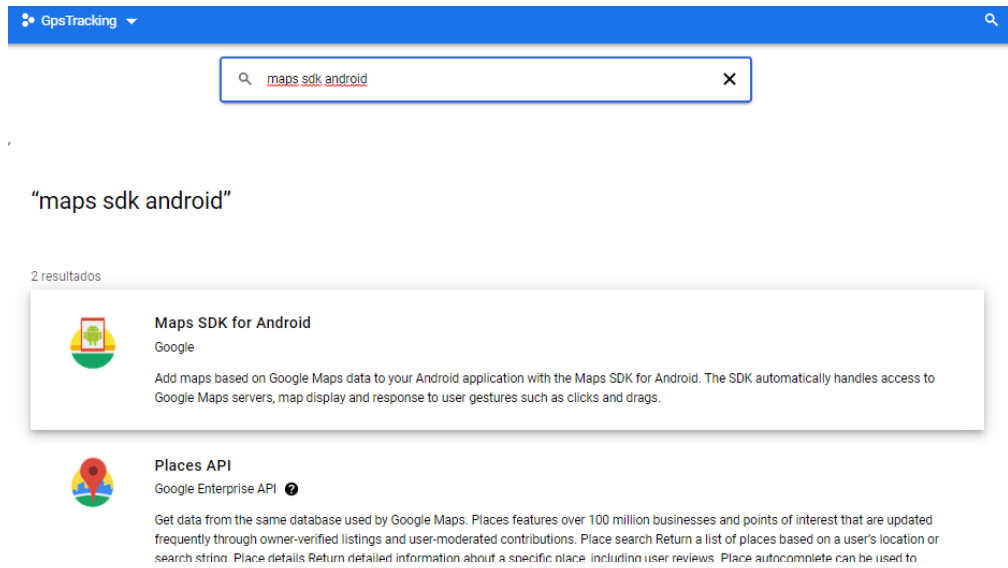


Figura 2.117: Habilitación de la Api Key

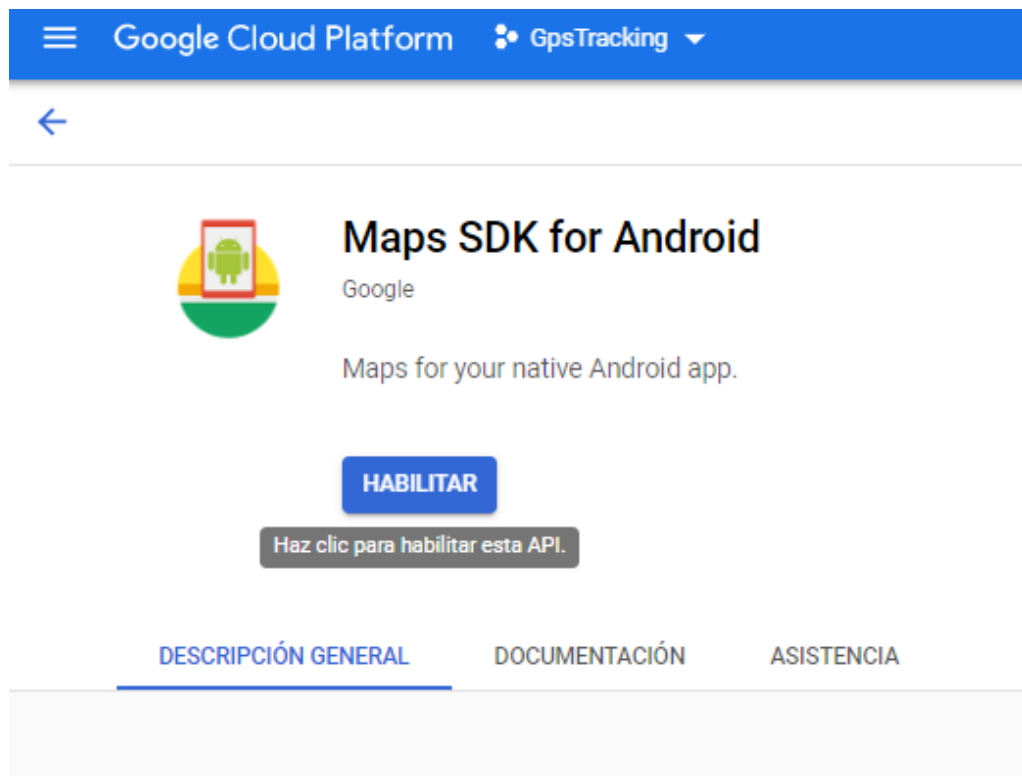
Fuente: Elaboración Propia.

Y buscamos maps SDK Android y cuando salga el resultado seleccionaremos el siguiente.



*Figura 2.118: Api Key Maps
Fuente: Elaboración Propia.*

Luego habilitamos en el sistema el SDK como en la figura 2.19:



*Figura 2.119: Maps SDK
Fuente: Elaboración Propia.*

Luego de estas acciones se encuentra habilitado maps para la aplicación, se procederá con el API KEY que es un código único que se genera para ello se regresa

al panel de control de nuestro proyecto de Google Cloud y esta vez hacemos clic en el menú lateral *API y servicios, credenciales*.

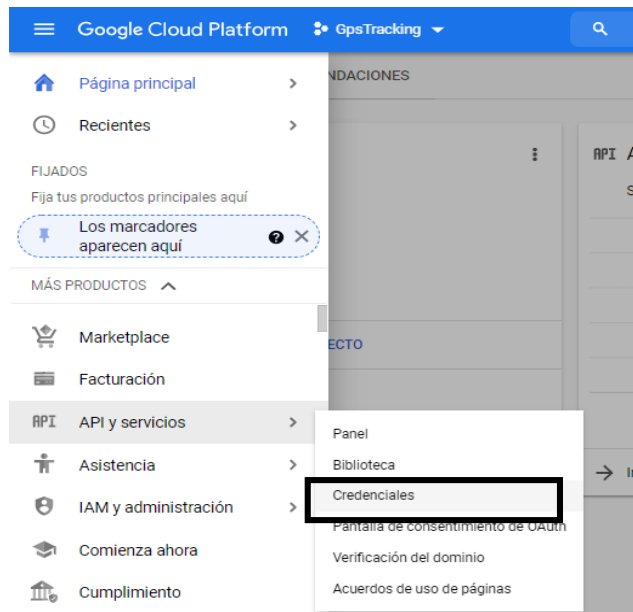


Figura 20: API, servicios y credenciales
Fuente: Elaboración Propia.

Posterior a ello, se crea las credenciales y el clave api

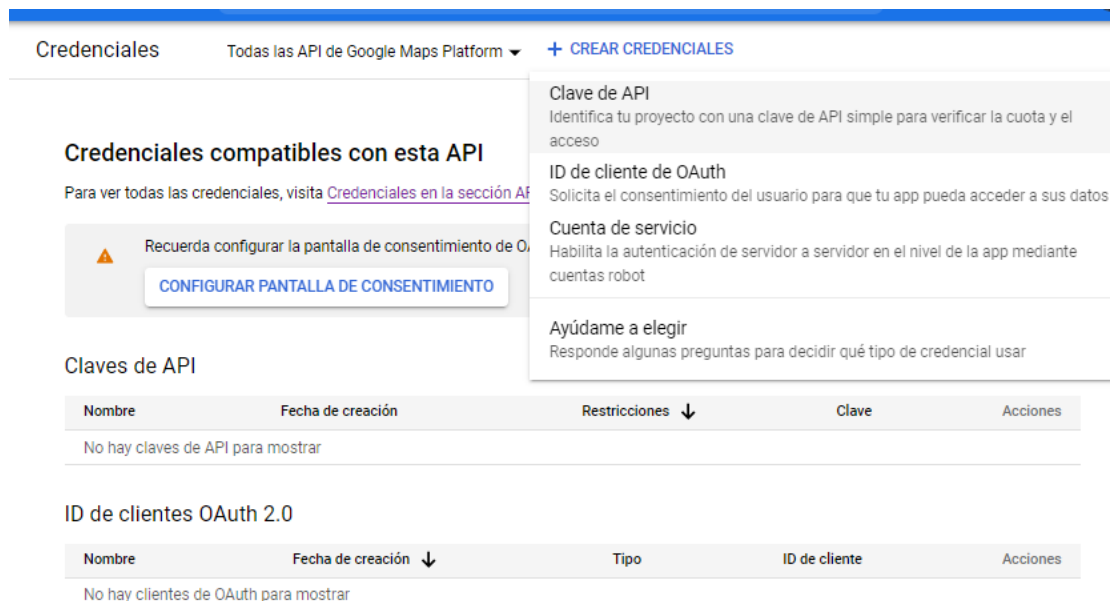


Figura 21: Creacion de la credencial y la clave de la Api
Fuente: Elaboración Propia.

Y después de unos segundos se crea la clave que se debe colocar dentro de la aplicación.



Figura 22: Clave de la Api
Fuente: Elaboración Propia.

Para poder utilizar Google Maps tenemos que añadir una dependencia (una librería), así que iremos a *Build.Gradle* y dentro de *dependencies{}* añadimos la siguiente línea y sincronizamos.

```
dependencies {
    implementation 'androidx.core:core-ktx:1.7.0'
    implementation 'androidx.appcompat:appcompat:1.4.0'
    implementation 'com.google.android.material:material:1.4.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.2'
    implementation 'com.google.android.gms:play-services-maps:18.0.2'
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
}
```

Figura 23: Creacion de una librería (dependencia)
Fuente: Elaboración Propia.

Con la vista en modo *project* vamos a *app > src > main > res > values* y haremos clic derecho en la carpeta *new > Values Resource File*, crearemos un fichero llamado **google_maps_api**. Y agregamos la API KEY que se genero

```
<string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">
    AIzaSyBGB3R0PeN04_s8425MXsGp7WKgRsqNONg</string>
</resources>
```

Figura 24: Values Resource File
Fuente: Elaboración Propia.

Como ya se creó el recurso, se procede a ejecutar en el proyecto Android que existe un api key válida, es por ello que se acude al fichero **AndroidManifest.xml** y dentro de la etiqueta `<application>` añadiremos nuestra propia etiqueta.

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="@string/google_maps_key" />
```

Figura 25: *AndroidManifest*
Fuente: *Elaboración Propia.*

Posterior a ello se crea, un fragment blank haciendo clic derecho com.nombre.nombredelproyecto.

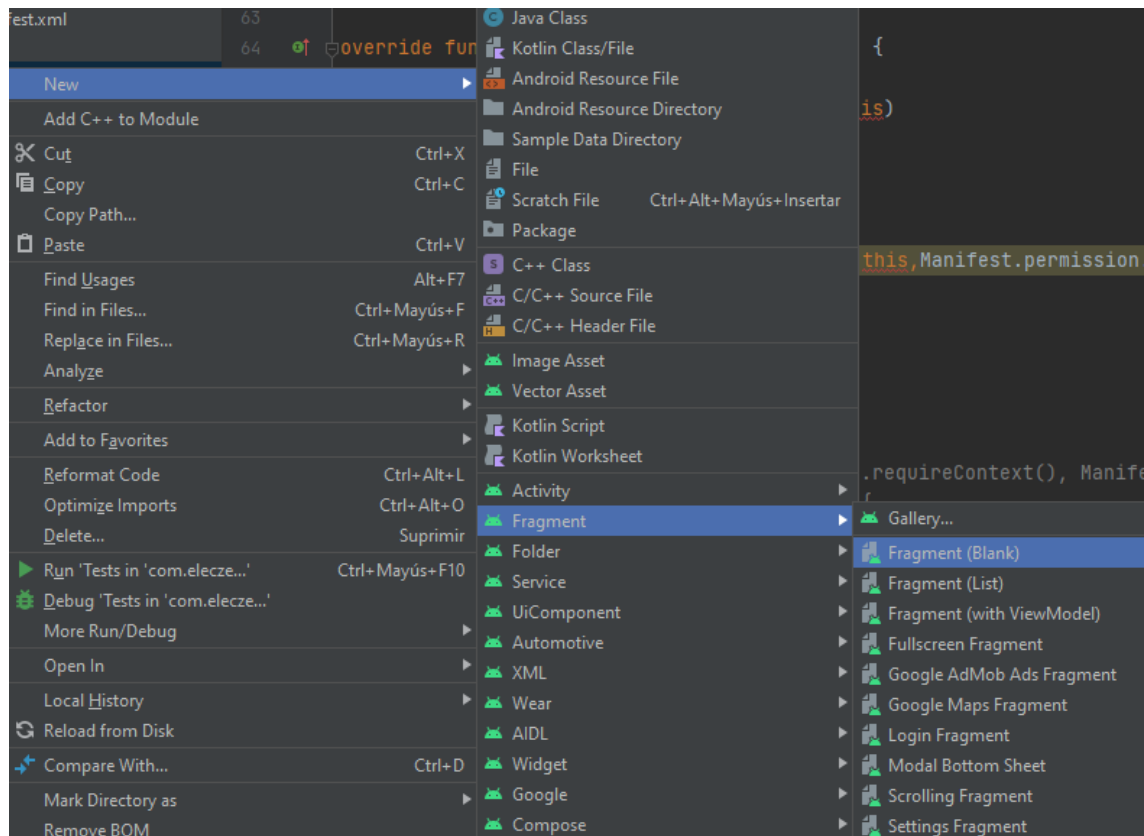
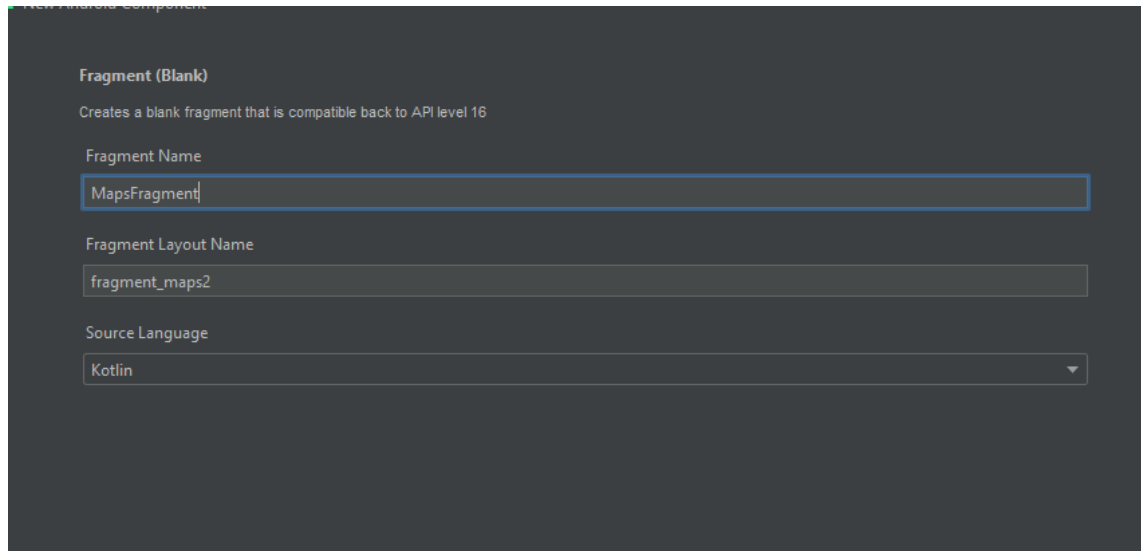


Figura 26: *Creación del Fragment blank*
Fuente: *Elaboración Propia.*

Colocamos el nombre en este caso se nombró como **MapsFragment** y luego creamos los archivos.



*Figura 27: Creación de Archivos
Fuente: Elaboración Propia.*

En el mapsfragment se procede a borrar todos los datos con excepción lo que se encuentra en la siguiente figura:

```
package com.eleczero.mapsapp

import ...

class MapsFragment : Fragment()
```

*Figura 28: MapsFragment
Fuente: Elaboración Propia.*

Y comenzamos a realizar el código, primero colocar los permisos en el manifest.xml

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
```

*Figura 29: Realización del Código
Fuente: Elaboración Propia.*

Nos dirigimos a fragments_map.xml y colocamos el siguiente código

```

1 | <?xml version="1.0" encoding="utf-8"?>
2 | <fragment xmlns:android="http://schemas.android.com/apk/res/android"
3 |         xmlns:tools="http://schemas.android.com/tools"
4 |         android:id="@+id/map"
5 |         android:name="com.google.android.gms.maps.SupportMapFragment"
6 |         android:layout_width="match_parent"
7 |         android:layout_height="match_parent"
8 |         tools:context=".MapsFragment" />

```

Figura 30: Código de fragment
Fuente: Elaboración Propia.

Donde se coloca un id para poder identificar, un nombre donde se define que es de la API maps.

```

override fun onCreateView(
    inflater: LayoutInflater,
    container: ViewGroup?,
    savedInstanceState: Bundle?
): View? {
    return inflater.inflate(R.layout.fragment_maps, container, attachToRoot: false)
}

```

Figura 31: ID de la API maps
Fuente: Elaboración Propia.

```

override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    createFracment()
}

private fun createFracment(){
    val mapFragment = childFragmentManager.findFragmentById(R.id.map) as SupportMapFragment
    mapFragment.getMapAsync( callback: this)
}

override fun onMapReady(googleMap: GoogleMap) {
    map = googleMap
    //map.setOnMyLocationButtonClickListener(this)
    //enableMyLocation()
}

```

Figura 32: ID de la API maps
Fuente: Elaboración Propia.

Hasta el momento lo que tenemos hecho es la estructura del mapa, en el cual podemos navegar por el mapa que se creó.

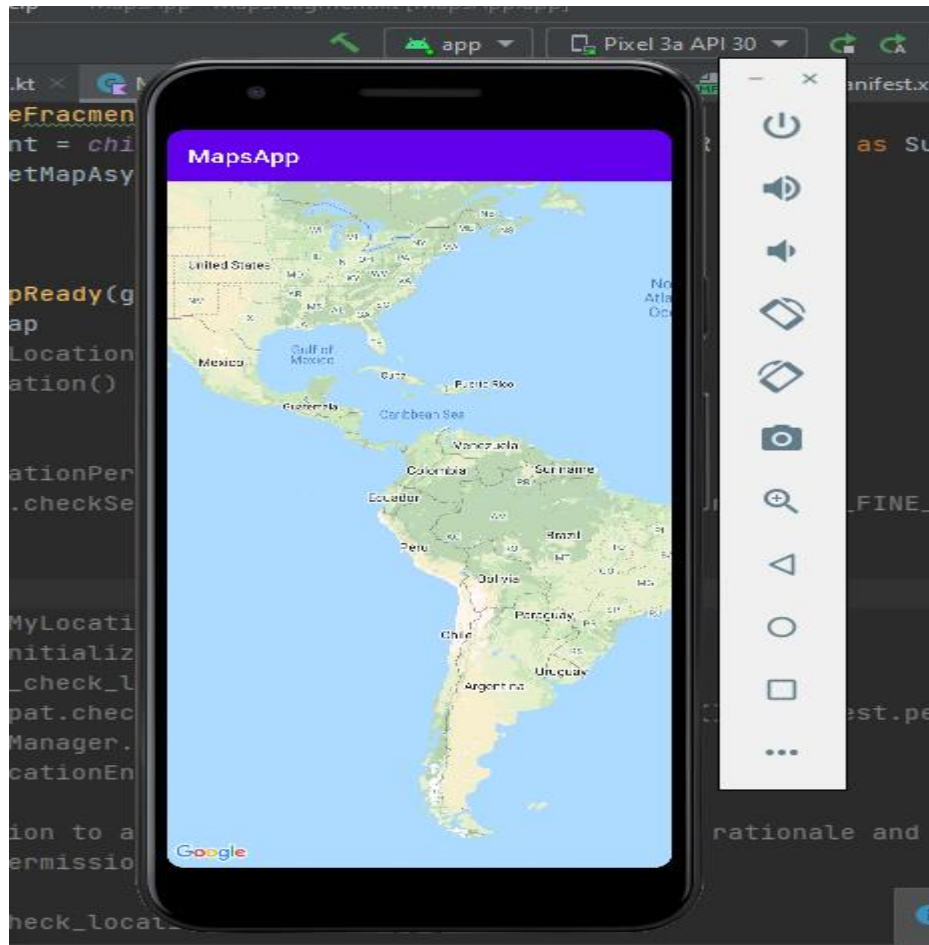


Figura 2.33: Estructura del Mapa
Fuente: Elaboración Propia.

Instalar motion

Para instalar motion se puede realizar de dos formas; la primera desde el terminal con el siguiente comando

```
sudo apt-get install motion
```

Y la segunda, desde las plataformas de desarrollo que se encuentran en <https://github.com/Motion-Project/motion/releases/>, en la cual copiamos el enlace del sistema que deseamos:

https://github.com/Motion-Project/motion/releases/download/release-4.4.0/pi_buster_motion_4.4.0-1_armhf.deb.

Con la herramienta PUTTY iniciamos una sesión ssh para controlar la raspberry pi desde el ordenador, posterior a ello señalamos:

1. sudo apt-get update
2. sudo apt-get ungrade

Se procede con la descarga del paquete de motion sudo wget https://github.com/Motion-Project/motion/releases/download/release-4.4.0/pi_buster_motion_4.4.0-1_armhf.deb cuando se termine el proceso de descarga se procede a colocar la siguiente línea para instalar el archivo .deb sudo dpkg -i pi_buster_motion_4.4.0-1_armhf.deb

```
2022-01-18 10:11:46 (95,9 KB/s) - «pi_buster_motion_4.4.0-1_armhf.deb» guardado
[489684/489684]

pi@raspberrypi:~ $ dpkg -i pi_buster_motion_4.4.0-1_armhf.deb
dpkg: error: la operación solicitada precisa privilegios de superusuario
pi@raspberrypi:~ $ sudo dpkg -i pi_buster_motion_4.4.0-1_armhf.deb
(Leyendo la base de datos ... 101125 ficheros o directorios instalados actualmen
te.)
Preparando para desempaquetar pi_buster_motion_4.4.0-1_armhf.deb ...
Desempaquetando motion (4.4.0-1) sobre (4.3.2-1) ...
Configurando motion (4.4.0-1) ...
Instalando una nueva versión del fichero de configuración /etc/motion/cameral-d
st.conf ...
Instalando una nueva versión del fichero de configuración /etc/motion/camera2-d
st.conf ...
Instalando una nueva versión del fichero de configuración /etc/motion/camera3-d
st.conf ...
Instalando una nueva versión del fichero de configuración /etc/motion/camera4-d
st.conf ...
Instalando una nueva versión del fichero de configuración /etc/motion/motion.co
nf ...
Procesando disparadores para man-db (2.9.4-2) ...
pi@raspberrypi:~ $
```

Figura 33: Instalación de la motion
Fuente: Elaboración Propia.

Antes de modificar el archivo de configuración de motion es necesario realizar una copia de seguridad

```
sudo cp /etc./motion/motion.conf /etc./motion/motion.conf.bak
```

Realizada la copia de seguridad se procede la configuración correspondiente, para ello hay que ir donde está el fichero principal

```
sudo nano /etc./motion/motion.conf
```

```

# Rename this distribution example file to motion.conf
#
# This config file was generated by motion 4.4.0
# Documentation: /usr/share/doc/motion/motion_guide.html
#
# This file contains only the basic configuration options to get a
# system working. There are many more options available. Please
# consult the documentation for the complete list of all options.
#
#####
# System control configuration parameters
#####
# Start in daemon (background) mode and release terminal.
daemon off

# Start in Setup-Mode, daemon disabled.
setup_mode off

# File to store the process ID.
; pid_file value

# File to write logs messages into. If not defined stderr and syslog is used.
; log_file value

# Level of log messages [1..9] (EMG, ALR, CRT, ERR, WRN, NTC, INF, DBG, ALL).
log_level 6
    
```

Figura 34: Fichero Principal
Fuente Elaboración Propia.

Los parámetros de configuración son los siguientes

- Deamon on
- Framerate 30
- Movie_quality 75
- Webcontrol_localhost off
- Webcontrol_parms 2
- Stream_localhost off

Para configurar varias cámaras se debe presionar los archivos de cámara

```

#####
# Camera config files - One for each camera.
#####
; camera /usr/etc/motion/camera1.conf
; camera /usr/etc/motion/camera2.conf
; camera /usr/etc/motion/camera3.conf
; camera /usr/etc/motion/camera4.conf
    
```

Figura 35: Configuración de las Cámaras
Fuente: Elaboración Propia.

Para trabajar con 3 camaras hay que modificar 4 ficheros:

1. El fichero /etc./motion/motion.conf
2. El fichero /etc./motion/camera1.conf
3. El fichero /etc./motion/camera2.conf

4. El fichero /etc/motion/camera3.conf

En el fichero 1 se debe comentar las líneas

- Videodevice /dev/video0
- Target_dir value

```
# Target directory for pictures, snapshots and movies
#; target_dir value

# Video device (e.g. /dev/video0) to be used for capturing.
#video_device /dev/video0

# Parameters to control video device. See motion_guide.html
```

Figura 36: Target
Fuente: Elaboración Propia.

Después desconectar las líneas de las cámaras

```
#####
# Camera config files - One for each camera.
#####
camera /usr/etc/motion/camera1.conf
camera /usr/etc/motion/camera2.conf
camera /usr/etc/motion/camera3.conf
; camera /usr/etc/motion/camera4.conf
```

Figura 37: Cámaras
Fuente: Elaboración Propia.

Ahora se debe definir la ubicación y configuración de los ficheros

- Sudo nano /etc/motion/camera1.conf
- Sudo nano /etc/motion/camera2.conf
- Sudo nano /etc/motion/camera3.conf

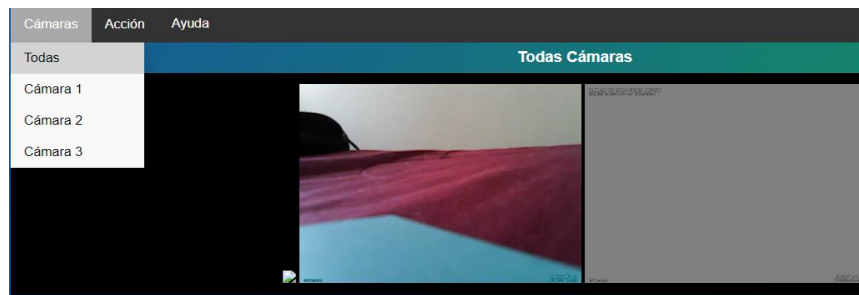
Posterior a ello, selecciona los directorios de las cámaras para ello se debe modificar de la siguiente manera:

```
#####
# Camera config files - One for each camera.
#####
camera /etc/motion/cameral.conf
camera /etc/motion/camera2.conf
camera /etc/motion/camera3.conf
; camera /usr/etc/motion/camera4.conf

#####
# Directory to read '.conf' files for cameras.
#####
; camera_dir /etc/motion/conf.d
█
```

*Figura 38: Modificación de las Cámaras
Fuente: Elaboración Propia.*

Al colocar la dirección IP de la raspberry con el puerto 8080, se dirigirá al panel de control donde se puede apreciar en la web las cámaras



*Figura 39: WEB de las cámaras
Fuente: Elaboración Propia.*

2.2 Implementación de cámaras y Gps.

Posterior a ello, se realizó la instalación e implementación de los dispositivos que se utilizaron para el rastreo y monitoreo del vehículo, como, por ejemplo: 3 cámaras de vigilancia ubicadas en la parte delantera y trasera, además el GPS, a continuación, se mostrara el proceso mediante las fotografías:



Figura 40: Desmontaje de mascarilla
Fuente: Elaboración Propia.

Para iniciar con el proceso, es necesario proceder con el desmontaje de la mascarilla del equipo.



Figura 41: Desmontaje del Tablero
Fuente: Elaboración Propia.



Figura 42: Cableado de las cámaras
Fuente: Elaboración Propia.

Una vez realizado el desmontaje se procede a incorporar el cableado de las cámaras de vigilancia.



Figura 2.44: Ubicación de las cámaras
Fuente: Elaboración Propia.



Figura 2.45: *Instalación del GPS*
Fuente: Elaboración Propia.

El GPS se ubicó en el parabrisas por motivo de prueba, ya que, permite una mejor interacción con el satélite.



Figura 2.46: *Instalación de la Raspberry*
Fuente: Elaboración Propia.

La Raspberry fue ubicada en la guardera del vehículo, la misma nos permite la interacción de los elementos con la aplicación.

Capítulo 3

Pruebas de rastreo y monitoreo de la aplicación en diferentes tipos de dispositivos móviles con sistema operativo Android

Una vez que se creó la aplicación móvil, posterior a ello se realizó la programación en este acápite se realizara las pruebas pertinentes en diferentes dispositivos móviles con sistema operativo Android que nos ayudara a verificar la validez de la app o lo cambios que se deben efectuar.

La primera prueba se llevó a cabo el día viernes 4 de febrero del 2022 en celular Samsung J7, en el que pudimos constatar el funcionamiento correcto de la aplicación denominada AUTOCONTIGO, para lo cual se realizó ciertos pasos:

1. Registrar el Usuario y la contraseña, cabe recalcar que estas credenciales fueron creadas para la prueba de la aplicación.

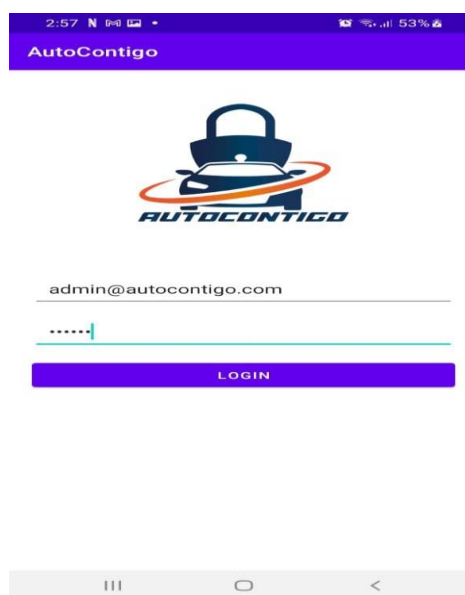


Figura 3.1: Usuario y Contraseña de Autocontrol
Fuente: Elaboración Propia.

2. Una vez que ingresamos a la aplicación accedemos al menú de AUTOCONTIGO, donde están disponibles las cámaras, GPS y la opción salir.

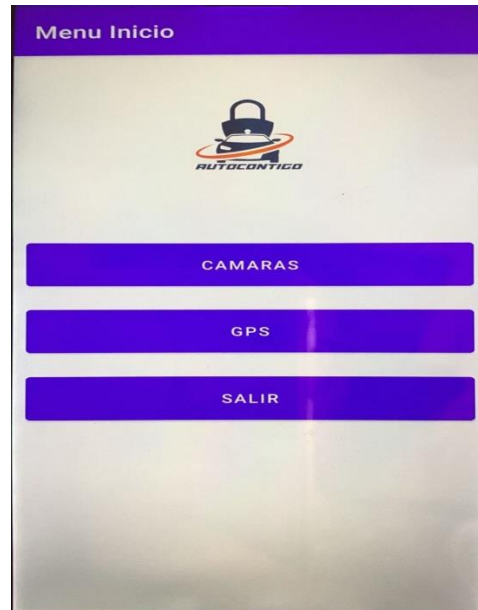


Figura 1: Menú Inicio
Fuente: Elaboración Propia.

3. Seleccionamos el Menú cámaras y podemos verificar en tiempo real cualquiera de las 3 cámaras disponibles en el vehículo.

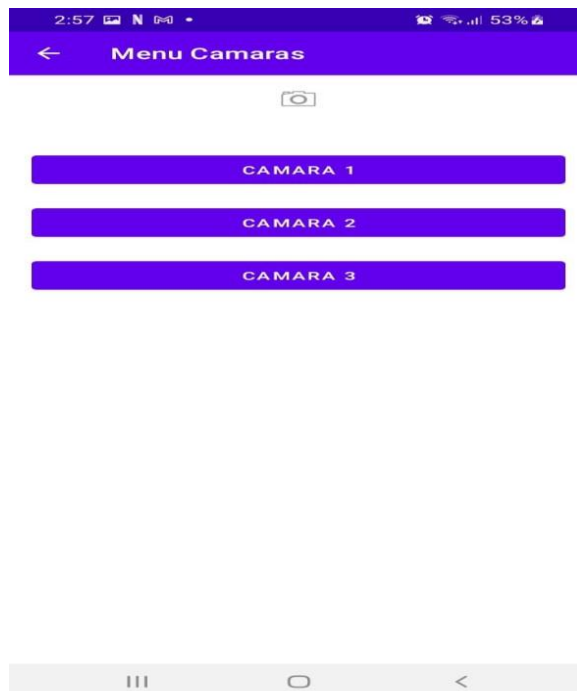


Figura 3.3: Menú Cámaras
Fuente: Elaboración Propia.

- Además, podemos optar por la opción de GPS que permitirá verificar la localización exacta de nuestro automóvil

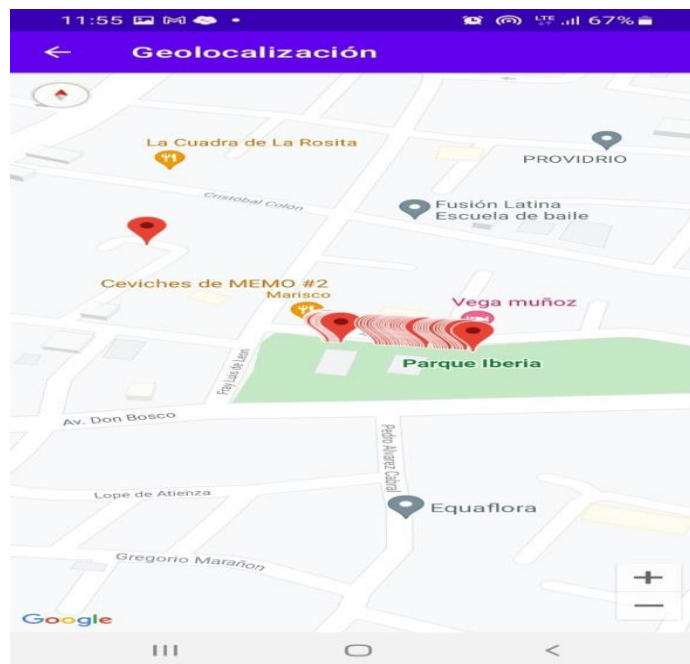


Figura 2: Geolocalización del vehículo
Fuente: Elaboración Propia.

En la segunda prueba se realizó el día 5 de febrero del 2022 en la cual se optimizo la relación del GPS con el satélite con el objetivo de observar el funcionamiento en otro dispositivo móvil, la trayectoria se queda almacenada en la nube lo que genera que en la aplicación en la opción GPS se registre solo la ubicación actual y exacta del automóvil.

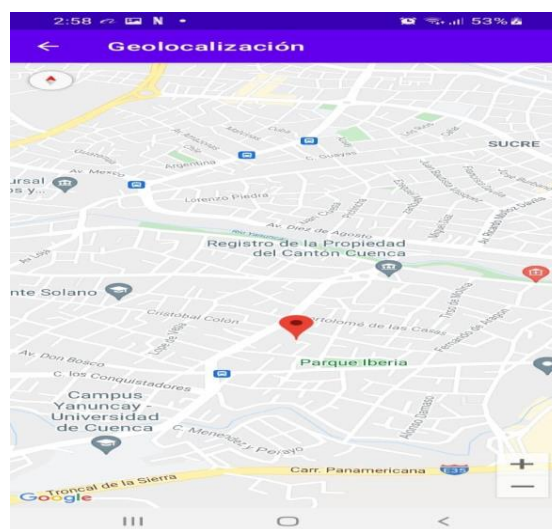


Figura 3.5: Geolocalización precisa del vehículo
Fuente: Elaboración Propia.

La tercera prueba se efectuó el día jueves 10 febrero en la cual se pudo realizar más ajustes a la aplicación móvil para poder tener mayor efectividad, en esta prueba se realizó en un dispositivo móvil de marca Xiaomi 10t, en esta prueba se verifico el funcionamiento de las cámaras y el GPS, recalcando que Android Studio realizo una actualización nueva por lo que fue necesario actualizarlo lo que produjo algunos errores que implica ajustar la aplicación móvil para una prueba posterior.

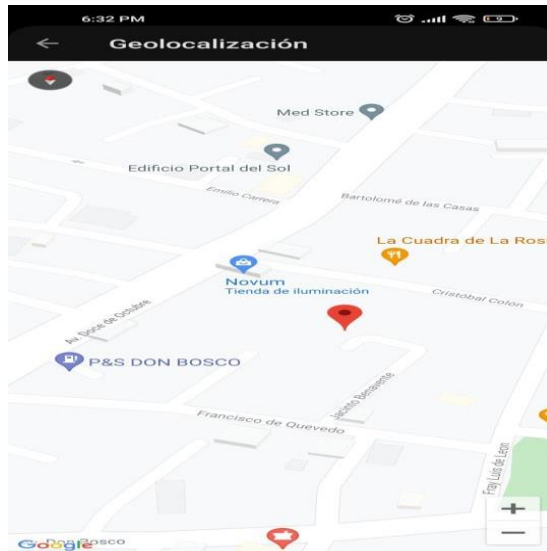


Figura 3.6: Geolocalización del vehículo
Fuente: Elaboración Propia.

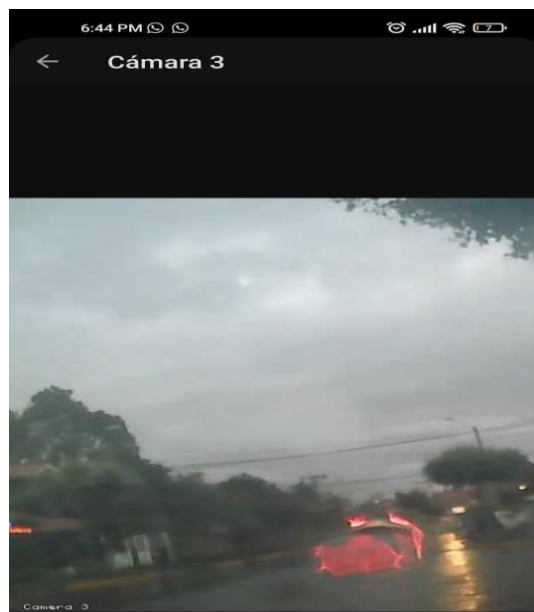


Figura 3.7: Cámara 3
Fuente: Elaboración Propia.

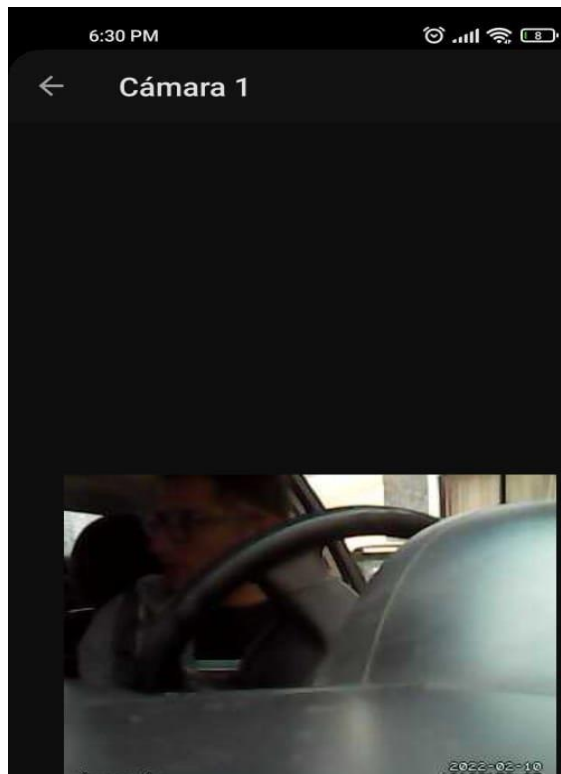


Figura 3.8: Cámara 1
Fuente: Elaboración Propia.

La última prueba se realizó el día 16 el febrero del 2022 en un Smartphone Huawei P20Lite en la cual la aplicación se encuentra en óptimas condiciones de funcionamiento tanto el geo localizador como las cámaras de vigilancia.

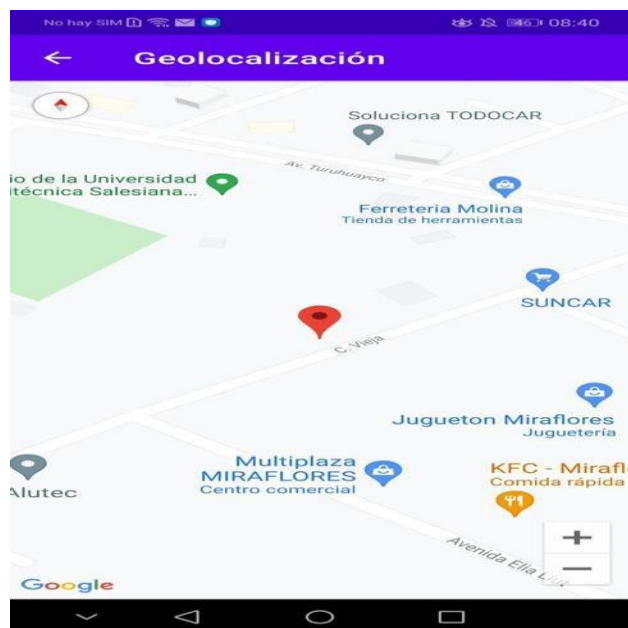


Figura 3.9: Geolocalización del Vehículo
Fuente: Elaboración Propia.

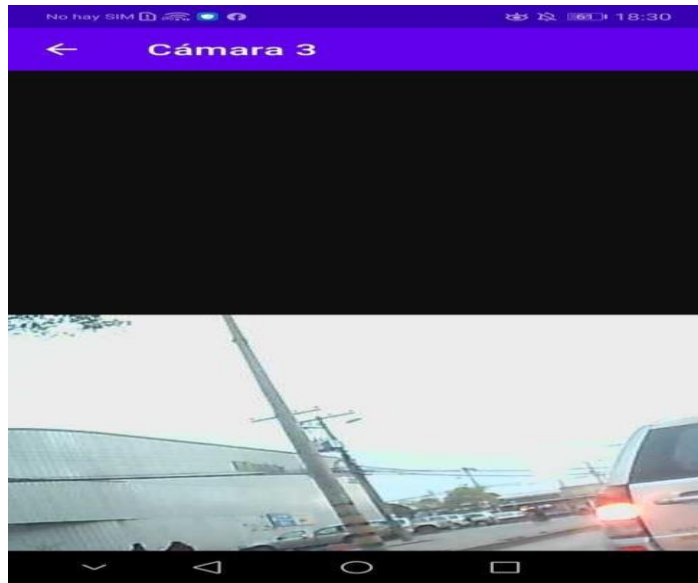


Figura 3.10: Cámara 3
Fuente: Elaboración Propia.

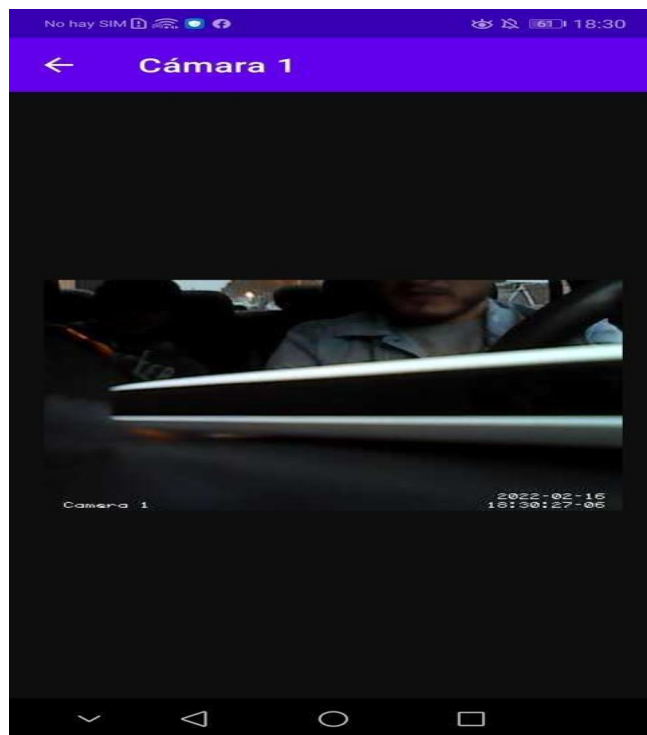


Figura 3.11: Cámara 1
Fuente: Elaboración Propia.

CONCLUSIONES

Una vez finalizado el presente trabajo de investigación, se permitió conceptualizar y definir los componentes que fueron utilizados para la realización de la aplicación móvil de seguridad, basado en la revisión bibliográfica.

En este contexto, vale recalcar que la inseguridad en el Ecuador ha incrementado notablemente incluyendo: el robo de vehículos ya sea de forma parcial o total, pues al existir el riesgo inminente se planteó como objetivo el diseño e implementación de una aplicación móvil para el sistema operativo Android que se ubicó en un vehículo Aveo Emotion año 2015, la misma que se realizó en dos enfoques:

El primero consistió en la programación y creación de la aplicación mediante el programa Android Studio con el lenguaje de Kotlin; ya que, al ser un lenguaje reducido; facilitó y contribuyó en la reducción del factor tiempo con la ayuda de sus librerías ya definidas. También existieron problemas entre la interacción de la Raspberry con la aplicación, porque se encontraba en el lenguaje Python, lo que conllevó a realizar un estudio más profundo sobre este tema, y vale recalcar que dentro de este campo no existe abundante información.

El desarrollo de la aplicación permite un mejor acceso para los usuarios y facilidad de utilización, para complementar el proyecto fue necesario la instalación de los elementos como: cámaras de vigilancia, el GPS, Raspberry, una batería externa para alimentar de corriente eléctrica cuando esta sea necesaria.

La batería externa sirvió para la autonomía del dispositivo que puede utilizarse para la localización del vehículo, además como control de objetos que se encuentran dentro del automotor.

Las cámaras utilizadas en el presente proyecto son cámaras web obteniendo un tiempo de retraso mínimo en la app, en las noches la nitidez de las mismas no es buena, ya que al utilizar cámaras web su calidad no es la esperada.

El segundo enfoque, se procedió a efectuar varias pruebas en diferentes dispositivos móviles con el sistema operativo Android, en las primeras pruebas se comprobó que existían errores en la aplicación, los cuales fueron resueltos hasta que la

aplicación funcione de manera correcta, el alcance dependerá de la red wifi en este caso a la que se encuentre conectado tanto las cámaras como el GPS.

En el mercado existen otras aplicaciones de seguridad, pero esta se diferencia, por su autonomía y el sistema de video vigilancia en tiempo real.

RECOMENDACIONES

Al presentar inconvenientes en el tiempo de transferencia de datos entre el GPS y el satélite se recomienda realizar un sistema interno propio para la aplicación y así garantizar la celeridad, además instalar cámaras de vigilancia de excelente calidad, con un dispositivo de grabación de video para que se complemente con el sistema del servidor.

Otro mejoramiento de la aplicación, consistiría en modificar el Hardware del dispositivo para que sea un dispositivo compacto, portátil, ya que, al utilizar el raspberry pi como elemento esencial para realizar las diferentes pruebas con los prototipos; es decir, se recomienda utilizar un Raspberry pi más actual; para aprovechar las bondades que nos brinda las actualizaciones de estos equipos.

Es factible, que el sistema propuesto, no solo se base en un sistema de video vigilancia; sino más bien vincularlo a un sistema de alarma antirrobo, con el objetivo de que, si alguna persona quisiera sustraer alguna parte del automotor, el sistema genere una alerta o alarma de un posible robo para que así garantizar el objetivo principal de este trabajo de titulación que es la seguridad del vehículo.

BIBLIOGRAFÍA

- Abreu, J. V. (2013). Business Processes Improvement on Maintenance Management. *Procedia technology*, 320-330.
- Alpala Guerrero, F. A., & Leon Chancusig, A. F. (2014). *Elaboración de un plan de mantenimiento automotriz con la implementación y diseño de un software especializado para el funcionamiento del parque automotor del ilustre municipio de la ciudad de Ibarra*. Riobamba, Ecuador: Escuela Superior Politecnica de Chimborazo.
- Alvarez, E. (Abril de 2018). *Computer Hoy*. Obtenido de Computer Hoy: <https://computerhoy.com/noticias/internet/que-es-router-mifi-3g-4g-que-sirve-78519>
- Alvarez, E. (2018). *Qué es un Router MiFi 3G o 4G y para qué sirve*. Obtenido de Qué es un Router MiFi 3G o 4G y para qué sirve
- Báez, L., & Cabrera, F. (2010). Diseño e Implementación de un sistema móvil anti-robo comandado por voz a un sistema de seguridad electrónica para vehículo. Quito, Ecuador: Universidad Internacional del Ecuador. Obtenido de <https://repositorio.uide.edu.ec/bitstream/37000/182/1/T-UIDE-0118.pdf>
- Ballester Bauset, S., Olmeda Gonzales, P. C., Macian Martinez, V., & Tormos Martinez, B. (2002). El mantenimiento de las flotas de transporte. *Tecnica Industrial*.
- Barreto Tenemea, A. S., & Sinchi Lojano, D. I. (2017). *Plan de mantenimiento asistido por computador para una flota de vehículos y maquinaria automotriz*. Cuenca, Ecuador: Universidad del Azuay.
- Bueno, D., Fumagall, L., Garetti, M., Pereira, C., Botelho, S., & Ventura, E. (2013). *Un modelo basado en el enfoque de integración de datos para mejorar el mantenimiento*. *Computers in industry*.
- Casas, P. (Noviembre de 2019). *Ecured*. Obtenido de Ecured: https://www.ecured.cu/Redes_de_datos

- Cerrada, M., Cardillo, J., Aguilar, J., & Faneite, R. (2007). *Diseño basado en agentes para sistemas de gestión de fallos en procesos industriales*. Computers in Industry.
- Eti, M. C. (2006). Strategic maintenance-management in Nigerian Industries. *Applied Energy*, 211-227.
- Godot, A., Villard, P., & Sournin, A. (1998). *The Implementation of a computerised maintenance management system*. Dallas, Texas: Society of Petroleum Engineers.
- González Fernández, F. J. (2011). *Teoría y práctica del mantenimiento industrial avanzado*. Madrid, España: Fundacion confemetal.
- INTERPOL. (octubre de 2020). *La delincuencia relacionada con los vehículos afecta a todas las regiones del mundo y tiene claros vínculos con la delincuencia organizada y el terrorismo*. Obtenido de <https://www.interpol.int/es/Delitos/Delincuencia-relacionada-con-los-vehiculos>
- López Jumbo, L. N., & Guaman Paucar, R. (2015). *Implementación de una gestión de mantenimiento asistido por ordenador (GMAO) para la flota vehicular del GAD municipal de Catamayo en la provincia de Loja*. Cuenca, Ecuador: Universidad Politecnica Salesiana.
- Luxhej, J. O. (1997). Trends and Perspectives in Industrial. *Journal of Manufacturing Systems*, 437-453.
- mapfre, C. e. (2008). *Gestión y logística del mantenimiento en automoción, 4ta Edición*. Valladolid, España: Cevismap S.A.
- Montilla Montaña, C. A. (2016). *Fundamentos de mantenimiento industrial*. Colombia: Universidad Tecnológica de Pereira.
- Pascual, J. (02 de 02 de 2015). *Computer Hoy*. Obtenido de Computer Hoy: <https://computerhoy.com/noticias/hardware/raspberry-pi-que-modelo-me-compro-23811>
- Raffino, M. (Octubre de 2020). *Concepto.de*. Obtenido de Concepto.de: <https://concepto.de/internet/#ixzz6of9eA7rK>

Raspberry Pi. (2021). *Que es la Raspberry Pi 3 Modelo B*. Reino Unido. Obtenido de <https://www.raspberrypi.com/products/raspberry-pi-3-model-b/>

Rey Sacristan, F. (2001). *Manual del mantenimiento integral en la empresa*. Madrid, España: Fundacion confemetal.

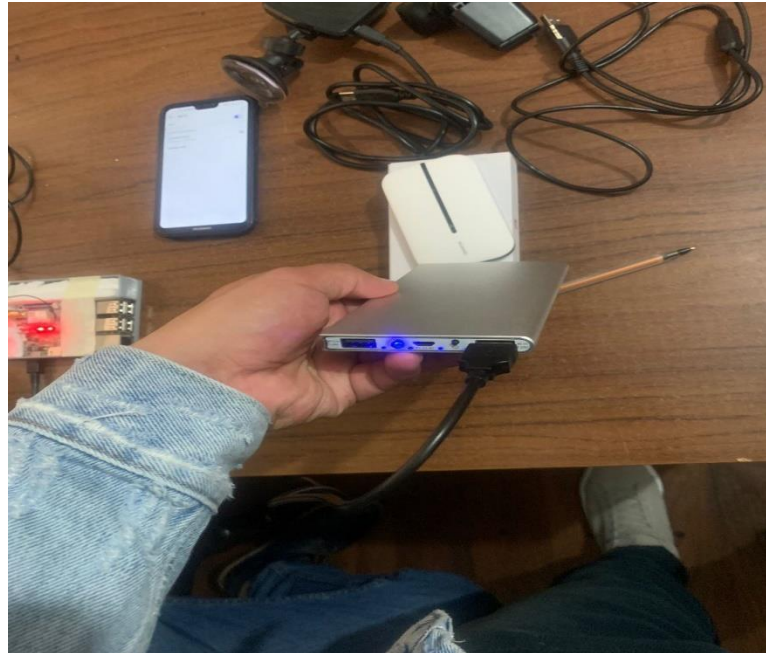
ANEXOS

Anexo 1: Dispositivo Mifi



Fuente: Realizado por los Autores

Anexo 2: Batería Externa



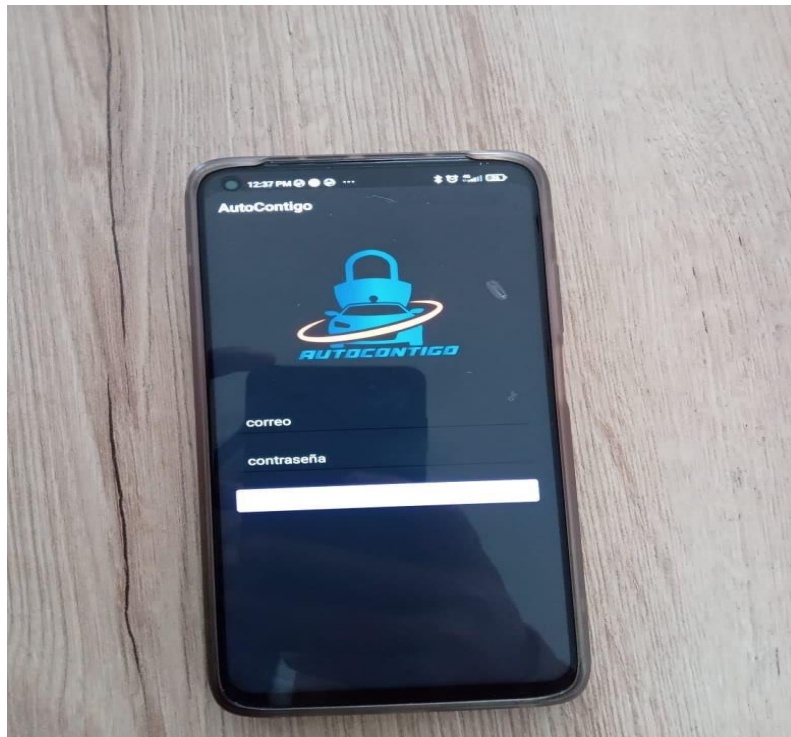
Fuente: Realizado por los Autores

Anexo 3: Batería de alimentación y Raspberrypi



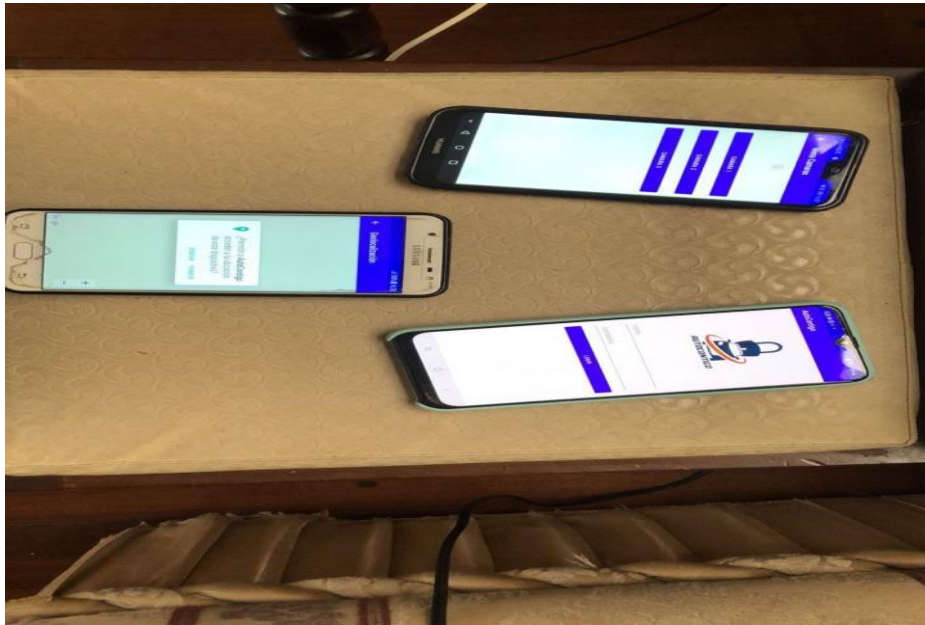
Fuente: Realizado por los Autores

Anexo 4: Instalación de la Aplicación en un dispositivo Xiaomi



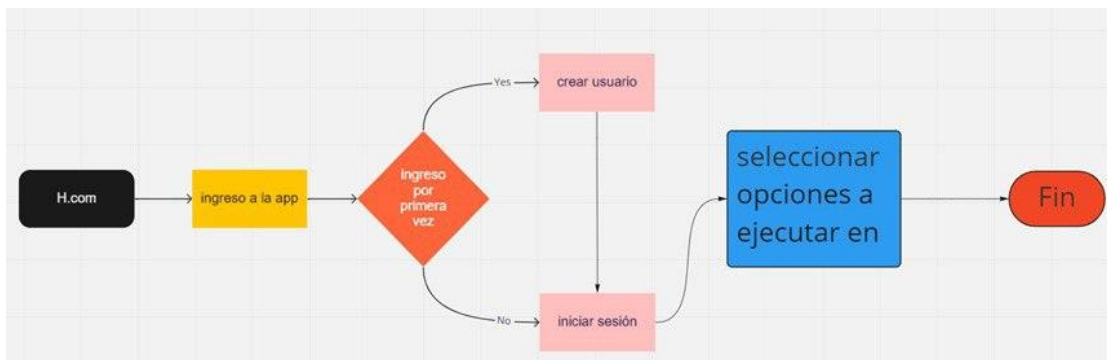
Fuente: Realizado por los Autores

Anexo 5 Diferentes modelos de Smartphone utilizados en las pruebas



Fuente: Realizado por los Autores

Anexo 6: Diagrama de flujo de la programación



Fuente: Realizado por los Autores

Anexo 7: Programacion del MainActivity

```

package com.eleczero.autocontigo

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Toast
import com.eleczero.autocontigo.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
        binding.btnlogin.setOnClickListener { login() }
    }

    private fun login () {
        // se toman los valores de los textView del
activity_main.xml
        val email =
binding.editTextTextEmailAddress.editableText?.toString()
        val password =
binding.editTextTextPassword.editableText?.toString()

        // mensajes de ayuda para verificar el login con exito
        val toast = Toast.makeText(this, "Ingreso Exitoso",
Toast.LENGTH_SHORT)
        val toast1 = Toast.makeText(this, "Ingreso Denegado",
Toast.LENGTH_SHORT)

        // funcion logica para comprar los datos -- tipo base de
datos--
        if(email == "admin@autocontigo.com" && password ==
"123456"){
            toast.show()
            openMenuInicio()
        }else{
            toast1.show()
        }
    }

    // funcion en la cual se realiza el cambio de activity de
MainActivity --> Manu_inicio
    private fun openMenuInicio() {
        val intent = Intent(this, MenuInicio::class.java)
        startActivity(intent)
    }
}

```

Anexo 8: Programación de MapsActivity

```
package com.eleczero.autocontigo

import android.Manifest
import android.content.ContentValues
import android.content.pm.PackageManager
import android.os.Bundle
import android.util.Log
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import com.eleczero.autocontigo.databinding.ActivityMapsBinding
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.LatLng
import com.google.android.gms.maps.model.MarkerOptions
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.DatabaseReference
import com.google.firebase.database.ValueEventListener
import com.google.firebase.database.ktx.database
import com.google.firebase.ktx.Firebase
import kotlinx.coroutines.*

class MapsActivity : AppCompatActivity(), OnMapReadyCallback {

    private lateinit var mMap: GoogleMap
    private lateinit var binding: ActivityMapsBinding

    private var REQUEST_CODE_LOCATION: Int = 0

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding = ActivityMapsBinding.inflate(layoutInflater)
        setContentView(binding.root)

        val actionBar = supportActionBar
        actionBar!!.title = "Geolocalización"
        createFragment() // Obtain the SupportMapFragment and get notified when the map is
        ready to be used.
    }

    private fun createFragment() {
```

```

        val mapFragment = supportFragmentManager.findFragmentById(R.id.map) as
SupportMapFragment
        mapFragment.getMapAsync(this)
    }

    override fun onMapReady(googleMap: GoogleMap) {
        mMap = googleMap // inicio el mapa
        enableLocation()
        val myRef = inicioFirebase()

        addMaker(myRef)
    }

    private fun enableLocation(){

        if (ActivityCompat.checkSelfPermission(
            this,
            Manifest.permission.ACCESS_FINE_LOCATION
        ) != PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(
            this,
            Manifest.permission.ACCESS_COARSE_LOCATION
        ) != PackageManager.PERMISSION_GRANTED
        ){
            ActivityCompat.requestPermissions(
                this,arrayOf(Manifest.permission.ACCESS_FINE_LOCATION),
                REQUEST_CODE_LOCATION)

        }else{
            requestLocalitationPermission()
        }
    }

    private fun requestLocalitationPermission() {
        if(ActivityCompat.shouldShowRequestPermissionRationale(
            this,Manifest.permission.ACCESS_FINE_LOCATION)){
            Toast.makeText(
                this,"Ve a ajustes y acepta los permisos",Toast.LENGTH_SHORT).show()
        }else{
            ActivityCompat.requestPermissions(
                this, arrayOf(Manifest.permission.ACCESS_FINE_LOCATION),
                REQUEST_CODE_LOCATION)
        }
    }

    private fun inicioFirebase(): DatabaseReference {
        val database = Firebase.database // conexion con la base de datos
        val myRef = database.getReference("GPS") // se refiere al hijo dentro de la base de

```

```
datos
    return myRef
}

private fun addMaker(myRef: DatabaseReference) {

    myRef.addValueEventListener(object : ValueEventListener {
        override fun onDataChange(dataSnapshot: DataSnapshot) {

            val latitud = // obtenemos los datos de latitud de la base de datos
                dataSnapshot.child("Georeferencia").child("latitud")
                    .value.toString().toDouble()
            val longitud = // obtenemos los datos de longitud de la base de datos
                dataSnapshot.child("Georeferencia").child("longitud")
                    .value.toString().toDouble()
            //val carro = LatLng(latitud, longitud) //asigna los valores de longitud latitud
            //Log.d(TAG, "Value is: $carro")

            mMap.addMarker(MarkerOptions().position(LatLng(latitud,longitud))
                .title("El vehiculo"))
            mMap.animateCamera(
                CameraUpdateFactory.newLatLngZoom(LatLng(latitud,longitud),18f),
                4000,null)
            mMap.clear()

        }
        override fun onCancelled(error: DatabaseError) {

            Log.w(ContentValues.TAG, "Failed to read value.", error.toException())
        }
    })
}
}
```

Anexo 9: Programación del Menú Cámaras

```
package com.eleczero.autocontigo

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import com.eleczero.autocontigo.databinding.ActivityMenuCamarasBinding

const val EXTRA_MESSAGE = "com.eleczero.autocontigo.MESSAGE"
class MenuCamaras : AppCompatActivity() {

    private lateinit var binding: ActivityMenuCamarasBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMenuCamarasBinding.inflate(layoutInflater)
        setContentView(binding.root)

        val actionBar = supportActionBar //cambio del titulo de las barra
        actionBar!!.title = "Menu Camaras"

        camera() // llama a las eventos de los botones
    }

    private fun camera () {
        var mensaje:String = ""

        binding.btncam1.setOnClickListener {
            val intentcam1 = Intent(this,VideoView::class.java).apply {
                putExtra(EXTRA_MESSAGE, "CAM1")
            }
            startActivity(intentcam1)
        }

        binding.btncam2.setOnClickListener {
            val intentcam2 = Intent(this,VideoView::class.java).apply {
                putExtra(EXTRA_MESSAGE,"CAM2")
            }
            startActivity(intentcam2)
        }

        binding.btncam3.setOnClickListener {
            val intentcam3 = Intent(this,VideoView::class.java).apply {
                putExtra(EXTRA_MESSAGE,"CAM3")
            }
            startActivity(intentcam3)
        }
    }
}
```

```
}
}
```

Anexo 10: Programación del Menú Inicio

```
package com.eleczero.autocontigo

import android.Manifest
import android.content.ContentValues
import android.content.Intent
import android.content.pm.PackageManager
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.view.View
import androidx.core.app.ActivityCompat
import com.eleczero.autocontigo.databinding.ActivityMenuInicioBinding
import kotlin.system.exitProcess

class MenuInicio : AppCompatActivity(), View.OnClickListener {

    private lateinit var binding: ActivityMenuInicioBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMenuInicioBinding.inflate(layoutInflater)
        setContentView(binding.root)

        val actionBar = supportActionBar // se cambia el titulo de barra principal
        actionBar!!.title = "Menu Inicio"

        btnMenu() // funcion de llamada a los menus
    }

    private fun btnMenu(){ //funcion en la cual se declaran los metodos Intends para cambiar
de activity

        // el binding permite tomar la accion del button para ejecutar

        binding.btncamara.setOnClickListener { //permite cambiar de activity especificamente
al menu camaras
            val intent = Intent(this,MenuCamaras::class.java)
            startActivity(intent)
        }

        binding.btngps.setOnClickListener(this) // permite ingresar al mapa
```

```
binding.btnsalir.setOnClickListener{ exitProcess(-1) } //permite salir de la activity  
  
}  
  
override fun onClick(p0: View) { // este metodo permite arrancar el MapsActivity en  
tiempo de ejecucion  
  
    val any = when (p0.id) {  
        R.id.btngps -> {  
            var intent:Intent = Intent(this, MapsActivity::class.java)  
            startActivity(intent)  
        }  
        else -> {  
            Log.e(ContentValues.TAG,"Error")  
        }  
    }  
}  
}
```


Anexo 11: Programación del VideoView

```
package com.eleczero.autocontigo

import android.os.Bundle
import android.webkit.WebView
import androidx.appcompat.app.AppCompatActivity

class VideoView : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_video_view)

        when(intent.getStringExtra(EXTRA_MESSAGE)){
            "CAM1" -> {
                val actionBar = supportActionBar
                actionBar!!.title = "Cámara 1"

                val myWebView: WebView = findViewById(R.id.webview1)
                myWebView.loadUrl("http://192.168.8.102:8081/1/stream")
            }
            "CAM2"->{
                val actionBar = supportActionBar
                actionBar!!.title = "Cámara 2"

                val myWebView: WebView = findViewById(R.id.webview1)
                myWebView.loadUrl("http://192.168.8.102:8084")
            }
            "CAM3"-> {
                val actionBar = supportActionBar
                actionBar!!.title = "Cámara 3"

                val myWebView: WebView = findViewById(R.id.webview1)
                myWebView.loadUrl("http://192.168.8.102:8083")
            }
        }
    }
}
```

Anexo 12: Programación del Android Manifest

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.eleczero.autocontigo">

    <permission android:name="com.eleczero.autocontigo.permission.MAPS_RECEIVE"
        android:protectionLevel="signature"/>
    <uses-permission
android:name="com.eleczero.autocontigo.googlemapstest.permission.MAPS_RECEIVE"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.AutoContigo"
        android:usesCleartextTraffic="true"
    >
        <activity
            android:name=".VideoView"
            android:exported="false" >
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value=".MenuCamaras"
            />
        </activity>
    <!--

```

TODO: Before you run your application, you need a Google Maps API key.

To get one, follow the directions here:

<https://developers.google.com/maps/documentation/android-sdk/get-api-key>

Once you have your API key (it starts with "Alza"), define a new property in your project's local.properties file (e.g. MAPS_API_KEY=Aiza...), and replace the "YOUR_API_KEY" string in this file with "\${MAPS_API_KEY}".

```

-->
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="@string/google_maps_key" />

<activity

```

```
    android:name=".MapsActivity"
    android:exported="false"
    android:label="@string/title_activity_maps" >
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value=".MenuInicio"
    />
</activity>
<activity
    android:name=".MenuCamaras"
    android:exported="false" >
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value=".MenuInicio"
    />
</activity>
<activity
    android:name=".MenuInicio"
    android:exported="false" />
<activity
    android:name=".MainActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
</application>

</manifest>
```

Anexo 13: Programación del Activity_main

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/white"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="5"
        android:gravity="center"
        android:orientation="vertical">

        <FrameLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <ImageView
                android:id="@+id/imageView"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:adjustViewBounds="true"
                android:padding="10dp"
                android:scaleType="fitCenter"
                app:srcCompat="@drawable/autocontigosinfondo" />
            </FrameLayout>

        </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="3"
        android:orientation="vertical"
        android:padding="15dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_margin="5dp"
            android:orientation="vertical">
```

```
<EditText
    android:id="@+id/editTextTextEmailAddress"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="correo"
    android:inputType="textEmailAddress"
    android:padding="15dp" />
```

```
<EditText
    android:id="@+id/editTextTextPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:ems="10"
    android:hint="contraseña"
    android:inputType="textPassword"
    android:padding="15dp" />
```

```
<Button
    android:id="@+id/btnlogin"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    android:hint="Login"
    android:text="Login" />
```

```
</LinearLayout>
```

```
</LinearLayout>
```

```
</LinearLayout>
```

Anexo 14: Programación del Activity_maps

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:map="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:id="@+id/map"
  android:name="com.google.android.gms.maps.SupportMapFragment"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  map:uiZoomControls="true"
  map:uiRotateGestures="true"
  map:cameraTilt="30"
  tools:context=".MapsActivity" />
```

Anexo 15: Programación de activity_menu_camaras

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="15dp"
    android:layout_marginRight="15dp"
    android:orientation="vertical"
    android:background="@color/white"
    tools:context=".MenuCamaras">

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="15dp"
        app:srcCompat="@android:drawable/ic_menu_camera"
        android:contentDescription="TODO" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:layout_margin="16dp">

        <Button
            android:id="@+id/btncam1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:text="Camara 1" />

        <Button
            android:id="@+id/btncam2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:text="Camara 2" />

        <Button
            android:id="@+id/btncam3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:text="Camara 3" />

    </LinearLayout>

</LinearLayout>
```

```
</LinearLayout>  
</LinearLayout>
```


Anexo 16: Programación del Activity_menu_inicio

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/white"
    tools:context=".MenuInicio">

    <FrameLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_gravity="center">

        <ImageView
            android:id="@+id/imageView4"
            android:layout_width="163dp"
            android:layout_height="117dp"
            android:layout_gravity="center"
            app:srcCompat="@drawable/autocontigo"
            tools:ignore="ImageContrastCheck"
            android:contentDescription="TODO" />
    </FrameLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_weight="3"
        android:orientation="vertical">

        <Button
            android:id="@+id/btncamara"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="15dp"
            android:padding="15dp"
            android:text="Camaras" />

        <Button
            android:id="@+id/btngps"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="15dp"
```

```
android:hint="gps"  
android:padding="15dp"  
android:text="GPS" />
```

```
<Button
```

```
android:id="@+id/btnsalir"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_marginTop="15dp"  
android:padding="15dp"  
android:text="Salir" />
```

```
</LinearLayout>
```

```
</LinearLayout>
```

Anexo 17: Activity_video_view

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".VideoView">

    <WebView
        android:id="@+id/webview1"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</LinearLayout>
```