



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE CUENCA

CARRERA DE INGENIERÍA AUTOMOTRIZ

**DESARROLLO DE UNA APLICACIÓN MÓVIL PARA LA ASISTENCIA A LA
CONDUCCIÓN**

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

AUTORES: JONATHAN FERNANDO APOLO GUALÁN

CHRISTIAN FRANCISCO YANANGÓMEZ CALERO

TUTOR: ING. NÉSTOR DIEGO RIVERA CAMPOVERDE, MSc.

Cuenca - Ecuador

2023

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Jonathan Fernando Apolo Gualán con documento de identificación N° 1900574888 y Christian Francisco Yanangómez Calero con documento de identificación N° 1150167441; 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, 20 de julio del 2023

Atentamente,



Jonathan Fernando Apolo Gualán

1900574888



Christian Francisco Yanangómez Calero

1150167441

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

Nosotros, Jonathan Fernando Apolo Gualán con documento de identificación N° 1900574888 y Christian Francisco Yanangómez Calero con documento de identificación N° 1150167441, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto técnico: “Desarrollo de una aplicación móvil para la asistencia a la conducción”, el cual ha sido desarrollado para optar por el título de: Ingeniero Automotriz, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente 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.

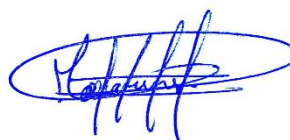
Cuenca, 20 de julio del 2023

Atentamente,



Jonathan Fernando Apolo Gualán

1900574888



Christian Francisco Yanangómez Calero

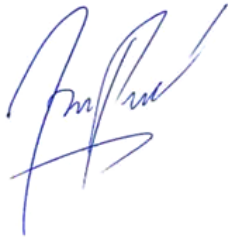
1150167441

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Néstor Diego Rivera Campoverde con documento de identificación N° 0103898995, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DESARROLLO DE UNA APLICACIÓN MOVIL PARA LA ASISTENCIA ALA CONDUCCION, realizado por Jonathan Fernando Apolo Gualán con documento de identificación N° 1900574888 y por Christian Francisco Yanangómez Calero con documento de identificación N° 1150167441, 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, 20 de julio del 2023

Atentamente,



Ing. Néstor Diego Rivera Campoverde, MSc.

0103898995

DEDICATORIA

El ser humano es de naturaleza sociable, de todas sus relaciones puede sacar una reflexión y aprender para ser mejor persona, sin esto somos seres insípidos, sin alegría, sin emociones, es un ser sin vida, lo dijo Humboldt “En el fondo son las relaciones con las personas lo que da sentido a la vida.” Y son estas mismas personas quienes con su forma de ser y enseñanzas cambian nuestra percepción y la forma de ver el mundo, siendo nosotros una construcción de cada uno de ellos, cada acción que los vimos realizar, cada forma de pensamiento sobre la vida.

Es egoísta pensar que nuestros logros son sólo nuestros, porque le debemos a las personas que han convivido y nos han ayudado desde niños, y lo más triste es que las palabras de agradecimiento no son suficiente para expresar lo que ellos han realizado. Pero creo que mencionarlos es una forma de plasmar sobre el papel su trabajo y desempeño.

Jonathan Fernando Apolo Gualán

DEDICATORIA

El presente proyecto lo dedico a toda mi familia, en especial a mis padres Francisco Yanangómez y Magdalena Calero quienes han sido el pilar de esta inmensa estructura de logros que he conseguido, además de brindarme siempre su inmenso apoyo en los momentos más difíciles durante esta trayectoria académica.

De la misma manera dedico a mis hermanas Yadira Neira y Dayra Yanangómez por estar junto a mi apoyándome y siempre brindándome un buen consejo, mis hermanas me han apoyado moralmente como económicamente, así mismo a mis sobrinos Lizbeth, Cindy, Julián y Vicente que son alegría e inspiración para seguir en adelante. Y a muchos más familiares y amigos que, aunque no los nombre me dieron fuerzas y me motivaron a seguir adelante.

Christian Francisco Yanangómez Calero

AGRADECIMIENTO

Doy mi agradecimiento a las personas que me ayudaron en mi construcción personal, donde cada uno ha aportado de manera inmensurable. A mis padres, Nancy Gualán y Fernando Apolo, a mis hermanos Lizbeth Abarca y Fabricio Apolo, y a mis sobrinos Martina y Miguel Cisneros quienes constituyeron mi familia primordial en todo este camino, por su dedicación y apoyo para obtener este logro. Un agradecimiento especial para todos mis amigos y compañeros universitarios que fueron figuras claves para que la estancia universitaria fuera más amena, donde brindaron consejos, compañía y sobre todo carisma. Gracias Andrés Celi, Christian Yanangomez, Adrián Lavanda, John Morocho, Fernando Peralta, Fernanda Juma, Alexis Medina y última instancia al Ing. Néstor Rivera. MSc. tutor de este trabajo, más que un docente se convirtió en un gran amigo con quién se podía contar con su sabiduría y su conocimiento.

Jonathan Fernando Apolo Gualán

AGRADECIMIENTO

En primer lugar, al Padre Celestial quien me dio la oportunidad de llegar hasta este momento tan importante de mi vida, por ayudarme a superar los momentos más difíciles y siempre estar junto a mí.

A mis padres Francisco Yanangómez y Magdalena Calero por siempre darme fuerzas y esperanzas cuando más lo necesitaba, por todo ese sacrificio, todo el amor que me brindan y nunca dejarme solo, su apoyo fue vital para conseguir este logro a pesar de los momentos duros que se pasó.

De manera muy especial a mis hermanas, que vieron en mi la fuerza para poder seguir adelante, motivándome y siempre viendo el lado bueno de las cosas.

A mí amigo, Jonathan Apolo por brindarme su apoyo, siempre estando juntos en los buenos y malos momentos de este proceso académico.

A todos los docentes de la carrera de Ingeniería Automotriz de la Universidad Politécnica Salesiana Sede Cuenca.

A mi tutor Ing. Néstor Rivera. MSc. agradezco su sabiduría, su conocimiento y su tolerancia, encaminándome por la culminación de este proceso académico, conservo un gran aprecio y respeto hacia su persona.

Christian Francisco Yanangómez Calero

RESUMEN

En la actualidad el avance tecnológico ha evolucionado considerablemente, llegando a tener en nuestras manos celulares inteligentes que cuentan la misma capacidad computacional que una de escritorio, en el cual se han aprovechado para generar aplicaciones que ayuden a los usuarios en muchos ámbitos de su vida cotidiana, uno no tan explorado es los medios de transporte, que a pesar de encontrar muchas aplicaciones que nos brindan acceso a mapas no se encuentran tan enfocadas para los conductores, siendo el enfoque del proyecto técnico, el diseño de una aplicación móvil para la asistencia vehicular prestando los servicios de ubicación, velocidad y las curvas próximas antes de la llegada del vehículo a dicha curva en Ecuador. El fin del proyecto es notificar al usuario la velocidad máxima permitida en la vía que circula, el nombre de la vía y notificar la velocidad que debe entrar el vehículo antes de la curva, así como la distancia.

La aplicación móvil fue diseñada con la plataforma para desarrolladores llamada Android Studio, una plataforma predilecta para diseñar aplicaciones que funcionen con sistema operativo Android, el lenguaje que se puede utilizar en esta plataforma es Java y Kotlin, donde para el proyecto el lenguaje más utilizado es el Kotlin por la mayor facilidad que representa programar en dicho lenguaje, debido a eso la disponibilidad de la aplicación es únicamente para los dispositivos con el mismo sistema operativo. Dentro de los resultados de las mediciones obtenidas son satisfactorias, ya que tanto la dispersión de datos como el coeficiente de correlación son altos, lo que nos proporciona una alta exactitud en los datos obtenidos

Palabra Claves: *Aplicación móvil, Confiabilidad, Kotlin, Java, Seguridad Actividad, Seguridad Vial, Movilidad Segura*

ABSTRACT

Nowadays, technological progress has evolved considerably, reaching to have in our hands smart phones that have the same computational capacity as a desktop, which have been used to generate applications that help users in many areas of their daily lives, one not so explored is in the means of transport, Despite finding many applications that provide access to maps, they are not so focused for drivers, being the focus of the technical project, the design of a mobile application for vehicular assistance providing location services, speed and the next curves before the arrival of the vehicle to the curve in the country of Ecuador. The purpose of the project is to notify the user of the maximum speed allowed on the road, the name of the road and notify the speed that the vehicle must enter before the curve, as well as the distance.

The mobile application was designed with the platform for developers called Android Studio, a favorite platform for designing applications that work with Android operating system, the language that can be used on this platform is Java and Kotlin, where for the project the most used language is Kotlin for the greater ease of programming in that language, because of that the availability of the application is only for devices with the same operating system. Within the results of the measurements obtained, they are satisfactory, since both the dispersion of data and the correlation coefficient are high, which provides us with high accuracy in the data obtained.

Keywords: *Mobile Application, Reliability, Kotlin, Java, Security Activity, Road Safety, Safe Mobility.*

ÍNDICE GENERAL

DEDICATORIA	V
AGRADECIMIENTO.....	VII
RESUMEN	X
ABSTRACT	12
1 INTRODUCCION	20
2 PROBLEMA	20
2.1 Antecedentes.....	21
2.2 Importancia y Alcances	21
2.3 Delimitación	22
3 OBJETIVOS	23
3.1 Objetivo General.....	23
3.2 Objetivos Específicos	23
4 ESTADO DEL ARTE	24
4.1 Accidentes de Transito	24
4.2 Movilidad segura	25
4.3 Distancia de frenado	27
4.4 Peralte	28
4.5 Aplicaciones Móviles	30
4.6 Android Studio	30
4.7 Lenguaje de programación	32
4.8 API.....	33
4.9 Servicios de mapas	33
4.10 Datos GeoJson	35

4.11	GPS.....	35
4.12	Aplicaciones móviles parecidas existente	36
5	DESARROLLO DE LA APLICACIÓN MOVIL	37
5.1	DISEÑO DE LAYOUTS.....	40
5.1.1	Diseño de layout de la pantalla principal.....	42
5.1.2	Diseño de layout de la pantalla para escoger el vehículo	44
5.1.3	Diseño de layout de la pantalla de Datos Almacenados	45
5.1.4	Diseño de layout de la pantalla de configuración.....	46
5.2	PROGRAMACION DE LAS ACTIVIDADES	47
5.2.1	Programación del mainactivity	48
5.2.2	VehicleActivity.....	50
5.2.3	Folder activity	51
5.2.4	SettingsActivity	52
5.2.5	Variables globales.....	53
5.2.6	Código de tratamiento de datos	54
5.2.7	Main screen activity.....	59
5.2.8	El segmento no está dentro de la lista.....	62
5.2.9	El segmento está dentro de la lista.....	64
6	COMPROBACION Y RESULTADOS	70
6.1	Comprobación de la búsqueda de curvas	71
6.2	Comprobación de resultados en la medición de la ubicación y la velocidad .	72
6.3	Comprobaciones de la exactitud de los datos de las coordenadas.....	74

6.4	Comprobación de los datos de velocidad	75
7	CONCLUSIONES	78
8	RECOMENDACIONES	79
9	REFERENCIAS BIBLIOGRAFICAS	80
10	ANEXOS.....	82

INDICE DE FIGURAS

Figura 1: Delimitación del Proyecto Técnico.....	22
Figura 2: Estructura general de GeoJSON	35
Figura 3: Constelación de Satélites para GPS	36
Figura 4: Logo de Android Studio.....	38
Figura 5: Estructura del proyecto	39
Figura 6: Diseño de Layout	40
Figura 7: Interfaz	41
Figura 8: Diseño de layout de la interfaz principal	42
Figura 9: Diseño de la interfaz en pantalla horizontal.....	43
Figura 10: Diseño de Layout para escoger el vehículo	44
Figura 11: Diseño de Layout de almacenamiento de datos	45
Figura 12: Diseño de Layout de la configuración	46
Figura 13: Actividades principales	47
Figura 14: Términos y condiciones	48
Figura 15: Mensaje de para aceptar términos y condiciones.....	49
Figura 16: Pantallas para permitir los accesos.....	50
Figura 17: Codificación de VehicleActivity.....	51
Figura 18: Codificación de Folder activity.....	52
Figura 19: Codificación de SettingsActivity	53
Figura 20: Variables globales y configuraciones	54
Figura 21: Codificación para consulta de datos.....	55
Figura 22: Datos de la consulta obtenidos.....	56
Figura 23: Codificación de tratamiento de datos.....	57

Figura 24: Estructura de datos de GeoJSON.	58
Figura 25: Diagrama de flujo de la parte inicial del programa.....	59
Figura 26: Parte del diagrama de flujo para el condicionante de velocidad.....	60
Figura 27: Parte del diagrama de flujo para el suavizado de datos	61
Figura 28: Posible casos para los cálculos de curvas	62
Figura 29: Diagrama de flujo cuando el segmento no está dentro de la lista.....	63
Figura 30: Diagrama cuando el segmento está dentro de la lista	64
Figura 31: Diagrama cuando el tamaño de curva es cero.	65
Figura 32: Diagrama de tamaño de curva mayor a 1.....	66
Figura 33: Distancia no calculada e igual a cero.	67
Figura 34: Diagrama de distancia diferente de cero (Primera Parte)	68
Figura 35: Diagrama de distancia diferente a cero (Segunda parte).....	69
Figura 36: Diferente rango de velocidad.	70
Figura 37: Distancia para la curva.....	71
Figura 38: Comprobación de la geometría de curva obtenida.....	72
Figura 39: Prueba de ruta con celular y freematics.	73
Figura 40: Diagrama de dispersión de los datos GPS	74
Figura 41: Representación grafica obtenida por el Freematics y Celular.....	76
Figura 42: Diagrama de dispersión obtenido de freematics y celular.	77

TABLA DE LUSTRACIONES

Tabla 1: Información para el peralte.....	29
Tabla 2: Aplicación móviles similares del país.	37
Tabla 3: Funcionalidades de los Botones de la APP.	41
Tabla 4: Significado de la estructura de GeoJSON.	58
Tabla 5: Acciones a realizar según la velocidad y la zona	71
Tabla 6: Coeficientes obtenidos.	75
Tabla 7: Coeficientes para los datos de velocidad.....	77

INDICE DE ANEXOS

Figura A- 1 Primeras Pruebas de Ruta.	82
Figura A- 2: Correcciones en la APP en la vía.	82
Figura A- 3 Colocación del Freematics.	83
Figura A- 4 Correcto funcionamiento de la App.	83
Figura A- 5: Diagrama de flujo de toda la Aplicación Móvil.	84

1 INTRODUCCION

El proyecto técnico propuesto sobre “Desarrollo de una aplicación móvil para la asistencia a la conducción”, tiene como fin que la conducción sea lo más seguro posible, para que esto suceda se requiere que el conductor tenga una atención constante y habilidades específicas al momento de conducir.

Hoy en día la mayoría de las personas cuenta con un dispositivo inteligente o smartphone ya que el mundo va en constante desarrollo tecnológico, por ello las aplicaciones móviles son un gran potencial para brindar información bastante precisa y en tiempo real y otros parámetros que ayudarán a disminuir los riesgos como los accidentes de tránsito.

Este proyecto encontrara la manera de aprovechar el potencial de las aplicaciones móviles y con ello se pretende realizar una aplicación móvil que brinde a los conductores instrumentos útiles y eficaces de tal manera que la experiencia en la conducción sea la mejor.

2 PROBLEMA

El tema de velocidad en las vías de Ecuador es una de las aristas principales en accidentes a nivel nacional, según las estadísticas obtenidas por la Agencia Nacional de Tránsito (ANT) sobre los accidentes de tránsito suscitados en el país, de enero a diciembre del año 2022 están registrados 21.739 accidentes, de estos incidentes el 16.3% corresponde al exceso de velocidad como causa probable. Debido a la circulación de los vehículos superando el límite de velocidad máximo en este periodo de tiempo, 2.725 personas resultaron heridas y 873 perdieron la vida, en el caso de las

personas lesionadas el exceso de velocidad es la tercera causal, mientras que para las personas que perdieron la vida representa la primera.

Los accidentes que se han suscitado en el periodo de 2022 producto de la velocidad, en su mayoría se dieron en zonas urbanas, dichas zonas son las más vulnerables debido que según el tipo de siniestro los peatones llegan a involucrarse dentro del incidente. Los accidentes producidos en zonas rurales tienen una tasa de fallecimiento muy alta debido a la pérdida de pista y volcamiento que tiene el vehículo como resultado de circular a velocidades que sobrepasen el límite máximo permitido, suscitándose este tipo de eventos en zonas curvas de la carretera.

2.1 Antecedentes

En la Ciudad de Cuenca la mayoría de los accidentes de tránsito son debido a las altas velocidades con las que circulan las personas en sus vehículos por las vías y por el tipo de curvas que existen, se ha propuesto radares, cámaras en los semáforos, y otras tecnologías se ha incorporado y aun así no se ha logrado un gran cambio o disminución importante.

2.2 Importancia y Alcances

Se espera que el presente proyecto técnico brinde una aplicación móvil muy eficiente e innovadora que ayude a los conductores a mejorar la experiencia en el manejo y sobre todo evitar accidentes de tránsito, con ello ayudara a que aumente la seguridad vial y así fomentar una cultura de conducción responsable.

Esta aplicación podrá ser utilizada por las personas que necesiten y que cuenten con dispositivo de software Android, se espera que en el futuro pueda ser ejecutable para cualquier sistema operativo.

2.3 Delimitación

Este proyecto técnico se realizó en la ciudad de Cuenca, provincia del Azuay, Ecuador. Sin embargo, se propone que para proyectos futuros se extienda para todo el país.

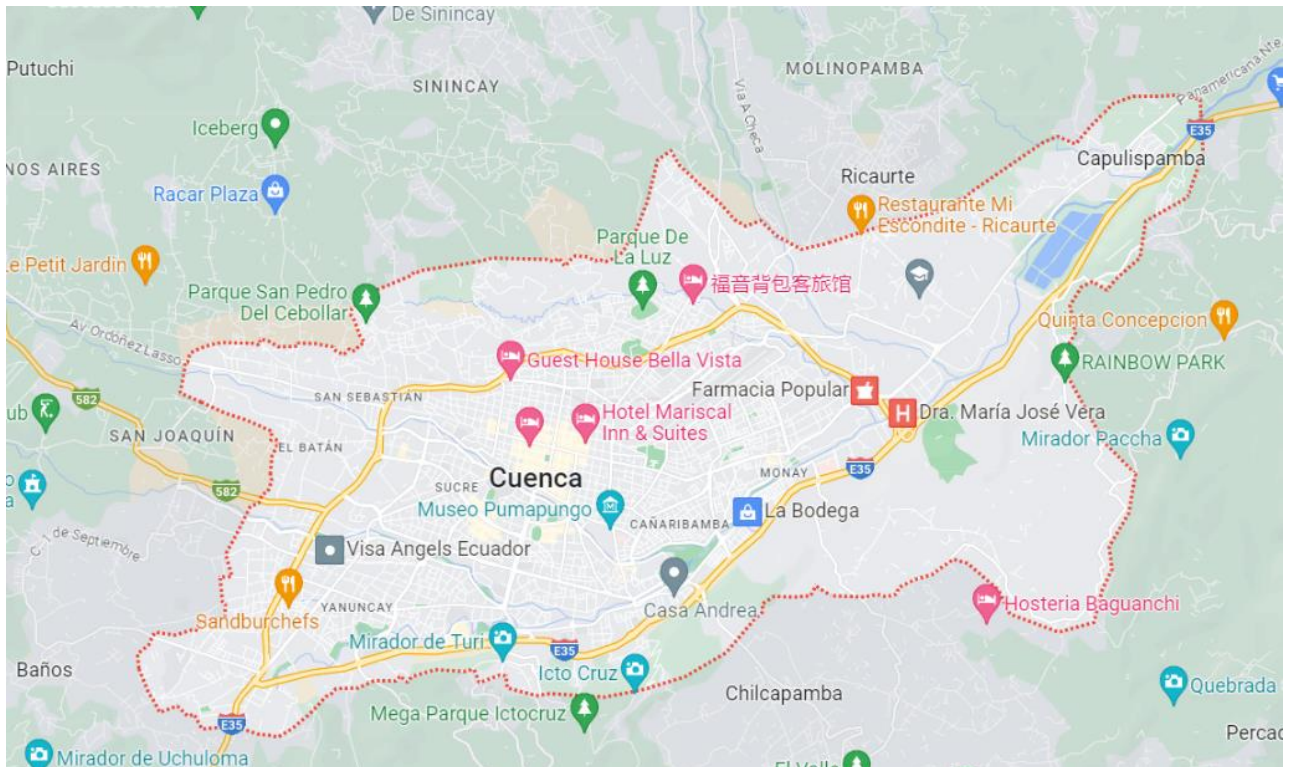


Figura 1: Delimitación del Proyecto Técnico

3 OBJETIVOS

3.1 Objetivo General

Desarrollar una aplicación móvil para la asistencia a la conducción.

3.2 Objetivos Específicos

- Realizar el estudio del arte en materia de aplicaciones que monitoreen la velocidad para obtener los parámetros de manejo del vehículo.
- Desarrollar la aplicación mediante programación con software especializado para obtener mediciones de la velocidad del vehículo.
- Comprobar el correcto funcionamiento de la aplicación previamente desarrollada y obtener datos de su funcionamiento.

4 ESTADO DEL ARTE

4.1 Accidentes de Tránsito

Los accidentes de tránsito son eventos no planificados que ocurren en la vía pública e involucran al menos un vehículo, ya sea automóvil, motocicleta, bicicleta o peatón. Estos accidentes pueden resultar en daños materiales, lesiones y, en algunos casos, la pérdida de vidas humanas.

La Organización Panamericana de la Salud califica como un problema social, cultural, tecnológico, económico, ambiental de alta complejidad a los accidentes de tránsito y de Salud Pública por el daño que produce a las personas, la comunidad, la familia y la sociedad en general. (MOPT, 2002).

Los accidentes de tránsito crean y reproducen diversas manifestaciones de violencia social; atacan la economía del país, la estabilidad ambiental, la esfera física y psíquica de las personas, el sistema de valores socioculturales, en especial la protección, la solidaridad y la percepción de la calidad de vida de las personas. (Fernández, 2009, p. 32).

Todo suceso eventual o acción involuntaria, que como efecto de una o más causas y con independencia del grado de estas, ocurre en vías o lugares destinados al uso público o privado, ocasionando personas muertas, individuos con lesiones de diversa gravedad o naturaleza y daños materiales en vehículos, vías o infraestructura, con la participación de los usuarios de la vía, vehículo, vía y/o entorno (RLOTTTSV, 2016)

Las consecuencias de los accidentes de tránsito pueden ser graves y variadas. Pueden resultar en lesiones leves o graves, discapacidades permanentes, trauma emocional y, en los casos más trágicos, la pérdida de vidas humanas. Además de los impactos personales, los accidentes de tránsito también tienen un impacto económico significativo debido a los gastos médicos, pérdida de productividad laboral y daños materiales.

La prevención de los accidentes de tránsito es un objetivo clave en la seguridad vial. Se implementan medidas para promover el cumplimiento de las normas de tránsito, mejorar la infraestructura vial, educar a los conductores y concienciar sobre los riesgos asociados con comportamientos peligrosos al volante como los excesos de velocidad.

Por ejemplo, el vehículo puede alcanzar la velocidad de derrape, que es la velocidad a la cuál un vehículo debido a las fuerzas centrífugas dada por la geometría de una curva genera un esfuerzo lateral en la cual puede perder la adherencia de los neumáticos con la vía (Aparicio Izquierdo, Vera Álvarez, & Díaz López, 1995).

En el límite de velocidad se tomará en cuenta lo que establece la Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial (LOTTTSV) para regular la velocidad en las vías estatales. La implementación de tecnologías de seguridad en los vehículos, como sistemas de frenado automático de emergencia y asistencia de mantenimiento de carril, también contribuye a la reducción de los accidentes de tránsito.

Los accidentes de tránsito son eventos no deseados que involucran vehículos y pueden causar daños materiales, lesiones y pérdida de vidas. La prevención de estos accidentes es fundamental para garantizar la seguridad vial y proteger la vida de las personas en las vías públicas.

4.2 Movilidad segura

El concepto de "movilidad segura" se refiere a la posibilidad de desplazarse de manera segura y sin riesgos en diferentes modos de transporte, ya sea caminando, en bicicleta, en automóvil o utilizando el transporte público. La movilidad segura implica la adopción de medidas y acciones para

garantizar la protección de los usuarios de las vías de tránsito y reducir los riesgos de accidentes y lesiones.

La movilidad segura abarca una serie de aspectos, entre los cuales se incluyen:

- **Infraestructura segura:** El diseño y mantenimiento de calles, carreteras, pasos de peatones, carriles para bicicletas y otros espacios destinados al tránsito, de manera que sean seguros para todos los usuarios. Esto implica la adecuada señalización, la separación física entre modos de transporte, la instalación de elementos de seguridad vial, como reductores de velocidad, y la planificación de infraestructuras accesibles para personas con movilidad reducida.
- **Comportamiento responsable:** La adopción de conductas seguras por parte de todos los usuarios de la vía. Esto incluye el respeto de las normas de tráfico, la conducción a una velocidad adecuada, el uso de dispositivos de seguridad como cinturones de seguridad y cascos, el respeto a los límites de alcohol o drogas al conducir, y la atención plena al entorno mientras se está en movimiento.
- **Educación y concienciación:** La promoción de la educación vial y la concienciación sobre los riesgos y las buenas prácticas en materia de seguridad vial. Esto implica programas de formación para conductores, campañas de sensibilización dirigidas a peatones y ciclistas, y la educación de los usuarios desde edades tempranas sobre comportamientos seguros en la vía.
- **Seguridad en el transporte público:** La implementación de medidas de seguridad en el transporte público, como la capacitación de conductores, la instalación de cámaras de seguridad en vehículos y estaciones, y la mejora de la accesibilidad para personas con discapacidad.

- Tecnología y vehículos seguros: La promoción de tecnologías de seguridad avanzadas en los vehículos, como sistemas de frenado automático, asistencia de mantenimiento de carril y sistemas de alerta de colisión. Además, se busca promover la seguridad en el diseño y fabricación de vehículos, garantizando estándares de seguridad elevados.

La movilidad segura busca asegurar que todas las personas puedan moverse de manera segura y protegida en su entorno, independientemente del modo de transporte utilizado. Esto implica la adopción de medidas en diversos frentes, desde el diseño de infraestructuras hasta el comportamiento responsable de los usuarios y el fomento de tecnologías y vehículos seguros (OMS, 2016).

4.3 Distancia de frenado

La distancia de frenado de un vehículo se refiere a la distancia que recorre desde el momento en que el conductor pisa el pedal del freno hasta que el vehículo se detiene por completo. Es importante entender que la distancia de frenado no solo depende de la velocidad del vehículo, sino también de otros factores como la eficiencia del sistema de frenos, las condiciones de la carretera y la adherencia de los neumáticos.

Cuando un conductor aplica los frenos, la fuerza de fricción entre los neumáticos y la superficie de la carretera ayuda a reducir la velocidad del vehículo. Sin embargo, durante el proceso de frenado, el vehículo sigue desplazándose debido a la inercia, por lo que se requiere una cierta distancia para que se detenga por completo.

La distancia de frenado puede variar en función de la velocidad inicial del vehículo y otros factores. En condiciones ideales, un vehículo bien mantenido con buenos neumáticos y un sistema de frenos en buen estado puede detenerse más rápidamente que un vehículo en condiciones desfavorables.

Velocidad de derrape: Velocidad a la cuál un vehículo debido a las fuerzas centrífugas dada por la geometría de una curva genera un esfuerzo lateral en la cual puede perder la adherencia de los neumáticos con la vía (Aparicio Izquierdo, Vera Álvarez, & Díaz López, 1995).

4.4 Peralte

También se lo puede llamar inclinación transversal, es la inclinación de una vía con relación al eje longitudinal. Lo que se busca es que los vehículos puedan tomar curvas con mayor comodidad y seguridad a velocidades muy altas.

El peralte se usa para compensar la fuerza centrífuga en vías y pistas de carreras que actúa sobre un vehículo al momento de tomar la curva. Cuando el vehículo circula en una trayectoria curva, resulta que va a sentir una fuerza hacia afuera que es la fuerza centrífuga. El peralte disminuye la fuerza centrífuga al inclinar la superficie de la vía hacia el interior de la curva.

La inclinación de la vía ayuda que el componente vertical de la fuerza centrífuga esté equilibrado por la fuerza de fricción entre la carreta y los neumáticos, ayuda a evitar el desplazamiento y el deslizamiento lateral del vehículo durante la curva. Si no existiera el peralte, sería muy difícil contrarrestar la fuerza centrífuga y los vehículos estarían obligados a disminuir la velocidad al tomar curvas, el resultado sería trayectorias menos eficientes y mal lentas.

Depende de muchos factores la cantidad de peralte que es necesario para una curva, como la velocidad recomendada, el radio de la curva y el coeficiente de fricción entre los neumáticos y la

carretera. Podrían provocarse situaciones peligrosas si no existe un peralte inadecuado, como deslizamiento lateral o vuelcos que son por la pérdida de control del vehículo.

En el caso de las pistas de carreras, comparado a las carreteras normales el peralte suele ser más pronunciado, debido a que pueden tomar curvas a velocidades mucho más altas los vehículos de competición. Los ingenieros de diseño de circuitos toman en cuenta los diferentes factores como la velocidad de los vehículos, la geometría de las curvas y la capacidad de agarre de los neumáticos para determinar el peralte óptimo en cada sección de la pista.

Tabla 1: Información para el peralte

Tipo de terreno	Dificultad de explanación del terreno	Pendiente transversal
Plano o Llano	Sin dificultad para obtener el terraplén	< 5 %
Ondulado	Dificultad moderada para obtener el terraplén siendo necesario un pequeño corte o relleno	6 – 12 %
Montañoso o accidentado	Requiere cortes importantes del terreno	13 – 40 %
Escarpado o Muy accidentado	Requiere movimientos máximos de terreno en corte fijado por divisorias de agua o construcción de túneles	>40%

4.5 Aplicaciones Móviles

Programas diseñados para ser ejecutados en teléfonos, tablets y otros dispositivos móviles, que permiten al usuario realizar actividades profesionales, acceder a servicios, mantenerse informado, entre otro universo de posibilidades. (Silva, F, 2019).

El uso de aplicaciones móviles ha experimentado un crecimiento exponencial en los últimos años, convirtiéndose en una herramienta fundamental en la vida cotidiana de las personas. El desarrollo de aplicaciones móviles puede desempeñar un papel crucial al proporcionar a los conductores información en tiempo real, alertas y funciones adicionales que mejoren la experiencia de conducción y fomenten prácticas seguras en las carreteras. Esta tesis se centra en el diseño y desarrollo de una aplicación móvil para la conducción, con el objetivo de mejorar la seguridad vial, la eficiencia y la comodidad del conductor.

4.6 Android Studio

Android Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) creado específicamente para el desarrollo de aplicaciones móviles en el sistema operativo Android. Proporciona una plataforma completa y robusta que incluye herramientas, bibliotecas y emuladores para facilitar el proceso de desarrollo de aplicaciones Android. A continuación, se presentan algunos conceptos importantes relacionados con Android Studio:

- Proyecto de Android: es el espacio de trabajo en Android Studio, donde se organiza y desarrolla una aplicación. Incluye todos los archivos fuente, recursos, bibliotecas y configuraciones necesarias para construir y ejecutar la aplicación.

- Gradle: es una herramienta de construcción y automatización utilizada en Android Studio. Se encarga de compilar el código fuente, gestionar las dependencias de las bibliotecas, crear el APK (archivo ejecutable de la aplicación) y realizar otras tareas relacionadas con la construcción del proyecto.
- Actividades (Activities): Una actividad es un componente fundamental de una aplicación Android. Representa una pantalla de interfaz de usuario con la que los usuarios pueden interactuar. En Android Studio, se pueden crear, diseñar y programar actividades utilizando el editor de diseño y el editor de código.
- Diseñador de interfaces gráficas (Layout Editor): Android Studio proporciona un diseñador de interfaces gráficas que permite crear y diseñar las interfaces de usuario de una aplicación de forma visual. Se utiliza para colocar elementos de interfaz, como botones, campos de texto y vistas, y definir su apariencia y comportamiento.
- Depuración (Debugging): Android Studio incluye herramientas de depuración que permiten detectar y solucionar errores en una aplicación. Permite establecer puntos de interrupción en el código, inspeccionar variables y seguir la ejecución paso a paso para identificar y corregir problemas.
- Emulador de Android (Android Emulator): Android Studio cuenta con un emulador de Android integrado que permite ejecutar y probar aplicaciones en un entorno virtual antes de desplegarlas en dispositivos reales. El emulador simula el comportamiento de un dispositivo Android, lo que facilita la prueba y depuración de aplicaciones en diferentes configuraciones.

Por lo tanto, será la aplicación designada para el desarrollo del código y la interfaz gráfica, debido a las librerías que se encuentran en la plataforma y la conexión que tiene esta con las aplicaciones de navegación de Google®, la cual permitirá tener una mejor comunicación y la transferencia de datos

para su utilización en los cálculos y valores que se requieran para determinar zonas pobladas y curvas peligrosas.

4.7 Lenguaje de programación

El lenguaje de programación es un conjunto de reglas y convenciones utilizadas para escribir instrucciones que una computadora puede entender y ejecutar. Es un medio de comunicación entre los programadores y las máquinas, permitiendo la creación de software y la automatización de tareas. Los lenguajes de programación proporcionados un conjunto de palabras clave, símbolos y estructuras de código que permiten expresar algoritmos y lógica de programación de manera precisa. Estos lenguajes se utilizan para escribir programas informáticos que realizan una variedad de tareas, desde cálculos matemáticos hasta operaciones complejas en aplicaciones de software. Existen muchos lenguajes de programación diferentes, cada uno con sus propias características, sintaxis y propósito.

- Java: Es un lenguaje de programación de alto nivel, orientado a objetos y multiplataforma. Fue desarrollado por Sun Microsystems (ahora propiedad de Oracle Corporation) en la década de 1990 y se ha convertido en uno de los lenguajes de programación más populares y ampliamente utilizados en la industria del software. Una de las características distintivas de Java es su capacidad para funcionar en diferentes plataformas, lo que significa que los programas escritos en Java pueden ejecutarse en cualquier dispositivo o sistema operativo que tenga una máquina virtual Java (JVM) instalada.
- Kotlin: Es un lenguaje de programación moderno y de alto nivel que fue desarrollado por JetBrains, una empresa de software con sede en Rusia. Fue presentado por primera vez en 2011 y se ha convertido en el lenguaje de programación preferido para el desarrollo de

aplicaciones en el ecosistema de Android. Kotlin está diseñado para ser interoperable con el lenguaje de programación Java, lo que significa que se puede utilizar junto con el código Java existente y aprovechar las bibliotecas y herramientas de Java. También se puede compilar en bytecode de Java y ejecutarse en la máquina virtual Java (JVM), lo que permite una fácil integración con el ecosistema de Java.

4.8 API

Las API (Interfaces de programación de aplicaciones) son conjuntos de reglas y protocolos que permiten la comunicación y la interacción entre diferentes componentes de software. En el contexto de las aplicaciones móviles, las API son esenciales para facilitar la integración de servicios y funcionalidades externas en una aplicación.

En otras palabras, una API define un conjunto de métodos y estructuras de datos que permiten a las aplicaciones interactuar con otros sistemas, servicios en línea o recursos externos. Las API permiten a las aplicaciones móviles acceder y utilizar funciones específicas, como servicios web, bases de datos, sensores del dispositivo, servicios de ubicación, servicios de pago, redes sociales y muchas otras.

4.9 Servicios de mapas

Los servicios de mapas son plataformas o aplicaciones que ofrecen funcionalidades relacionadas con la visualización, búsqueda y manejo de mapas digitales en dispositivos electrónicos, como computadoras, teléfonos inteligentes y tabletas. Estos servicios proporcionaron datos geográficos y

permitieron a los usuarios navegar por mapas, obtener indicaciones de ruta, realizar búsquedas de ubicación, visualizar información geoespacial y más.

- Google Maps es un servicio de mapas y navegación desarrollado por Google. Proporciona una amplia gama de funciones y características relacionadas con la ubicación y la cartografía, tanto en la versión web como en las aplicaciones móviles. Se utilizará los datos dados por programas como Google Maps u otras aplicaciones para la utilización de la información que tengan sobre las vías de Ecuador, y con ello parametrizar y obtener una base interna con la cual se pueda calcular los diferentes valores que se necesite en la aplicación, ante ello es muy probable que se deba cambiar el tipo de variable o utilizar los datos obtenidos a vectores y con ello obtener la curvatura de la vía.
- OpenStreetMap (OSM) es un proyecto colaborativo y de código abierto que tiene como objetivo crear y proporcionar datos geoespaciales libres y gratuitos para todo el mundo. Es una alternativa a los servicios de mapas comerciales, como Google Maps.

El mapa de OpenStreetMap se crea a través de la contribución de miles de voluntarios de todo el mundo. Estos utilizan dispositivos voluntarios GPS, imágenes de satélite y otra información disponible para recopilar datos sobre carreteras, edificios, lugares de interés y muchos otros elementos geográficos. Estos datos se almacenan en una base de datos central, que luego se utilizan para generar mapas interactivos y datos geoespaciales.

4.10 Datos GeoJson

Los datos GeoJson es un formato de codificación ligero y flexible utilizado para representar datos geoespaciales en formato JSON (JavaScript Object Notation). JSON es un formato de intercambio de datos ampliamente utilizado en aplicaciones web y móviles debido a su simplicidad y facilidad de lectura y escritura por parte de las máquinas.

Los datos GeoJSON están diseñados para almacenar y representar información geográfica, como puntos, líneas y polígonos, junto con atributos asociados a esas entidades geográficas. Este formato es ampliamente utilizado en sistemas de información geográfica (SIG) y aplicaciones que trabajan con datos geoespaciales.

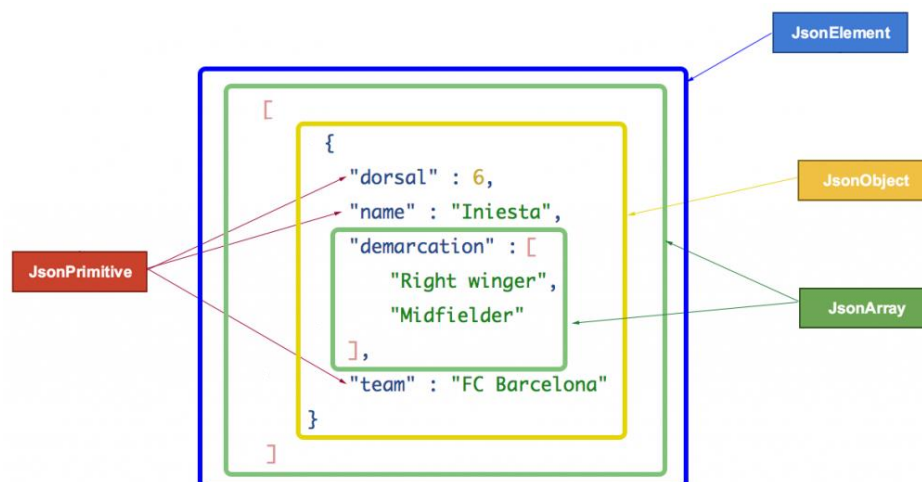


Figura 2: Estructura general de GeoJson

Fuente: Jarroba

4.11 GPS

El GPS (Global Positioning System) o Sistema de Posicionamiento Global es un sistema satelital capaz de localizar cualquier objeto, persona o vehículo de las siguientes maneras: Todo el tiempo; en cualquier lugar del planeta independientemente de la condición climática.

Según (Leica Geosystems, 2021) consiste en un grupo de 27 satélites (24 operativos y 3 de respaldo) que orbitan la tierra a una altura de aproximadamente 20.000 km en trayectorias sincronizadas repartidas en 6 planos orbitales de cuatro satélites cada uno y envían señales de radio a la superficie del globo terráqueo mostrado en la Figura 3.

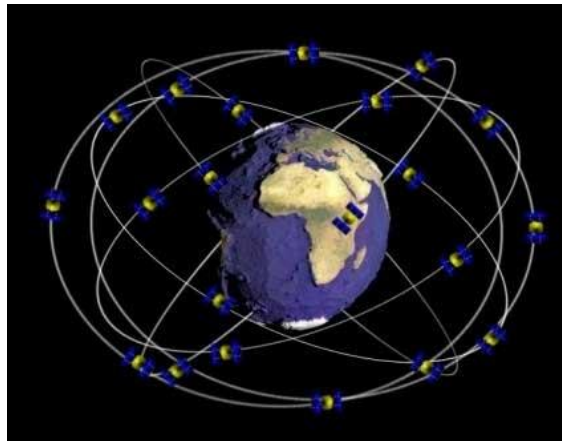


Figura 3: Constelación de Satélites para GPS

Fuente: Azimutmarine

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)

4.12 Aplicaciones móviles parecidas existente

En el país existen algunas aplicaciones móviles similares a la desarrollada en este proyecto y por ellos hemos tratado de investigarla y obtener información que nos sea de gran importancia.

Tabla 2: Aplicación móviles similares del país.

Fuente: (Gallegos, Urgirlés), 2022

Aplicación Móvil	Realiza	Servicio	Ubicación
ASISTENCIA VEHICULAR PREFECTURA DEL GUAYAS	Asiste en caso de emergencias en las vías concesionadas de la provincia del Guayas	Servicios de grúas, servicios de ambulancia	Guayas – Guayaquil
CHOFI	Asiste en la conducción	Servicio de choferes en caso de que no puedas manejar tu vehículo. Personas que no puedan manejar o no tengan licencia de conducir	Azuay – Cuenca
CODRIVER	Ayuda a encontrar servicios automotrices o talleres mecánicos que estén cercanos a lugar donde se encuentra	Auxilio para mecánica, vulcanizadoras, pintura, lavadoras, almacenes de repuestos y estaciones de servicio de combustible.	Pichincha - Quito

5 DESARROLLO DE LA APLICACIÓN MOVIL

El desarrollo de la aplicación se realiza en Android Studio, que es una herramienta de desarrollo de código 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.

Android Studio



Figura 4: Logo de Android Studio

Esta plataforma tiene una estructura diseccionado por carpetas donde cada una de ellas realiza un solo trabajo, en los cuales se pueden encontrar los archivos MainActivity que se encuentran en la carpeta example.dams ,que es la carpeta del proyecto, que realizan las funciones main similar al lenguaje java, los MainActivity son los encargados de declarar las funciones, variables, métodos y algunas cosas más, esto permitirá hacer los cambios como de Activity, o la obtención de datos de textView.

La estructura del proyecto se puede observar en la parte izquierda de la aplicación de desarrollo, en donde está estructurada por carpetas y estas carpetas representan los recursos que utiliza la aplicación debidamente ordenadas y separadas, por ejemplo, para los recursos de tipos de letras, los audios, valores y programación tienen cada una su propia carpeta donde no se pueden poner otros valores dentro de dicha carpeta que no le corresponda, este tipo de división mencionada se puede observar en la Figura 5. Para el proyecto se utilizó iconos, audios, los archivos del mapa en formato geojson, tipos de letras, los diseños gráficos y la programación de los diferentes servicios y acciones de la aplicación.

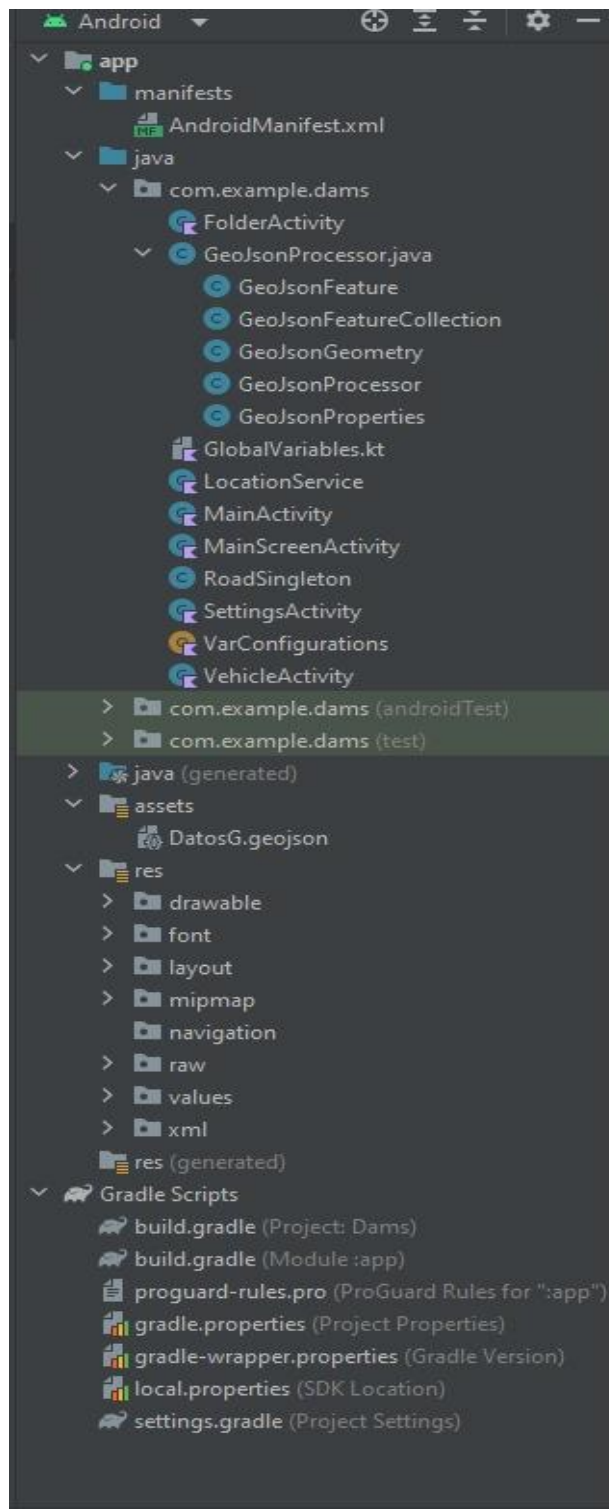


Figura 5: Estructura del proyecto

La división principal de la programación se puede comentar como el diseño del layout y la programación de las actividades, pueden existir actividades sin layout, pero no layout sin la programación de la actividad.

5.1 DISEÑO DE LAYOUTS

Los layout corresponde al diseño gráfico de la pantalla dónde únicamente se tiene es el aspecto físico y no la funcionalidad, para el proyecto se han desarrollado 6 layout diferentes que se encuentran en la carpeta de layout, eso se puede apreciar en la figura 6 en donde se puede mencionar que esos archivos tienen una nomenclatura definida conformado por la palabra activity más guion bajo y el nombre que le demos a la actividad.

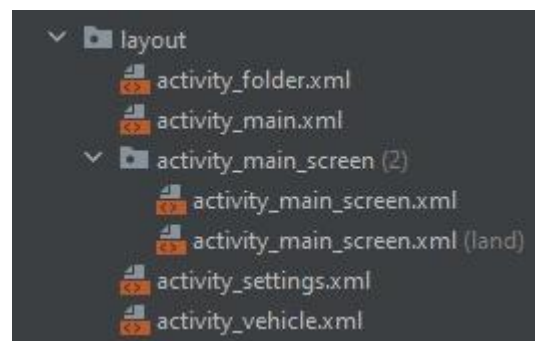


Figura 6: Diseño de Layout

5.1.1 Programación de la interfaz de la pantalla de inicio

En el diseño para la pantalla inicial se ubican el logo de la aplicación, cuatro botones con diferente funcionalidad representadas en la tabla de continuación:

Tabla 3: Funcionalidades de los Botones de la APP.

BOTON	FUNCIONALIDAD
Iniciar programa	Se mostrarán los resultados obtenidos por la Aplicación móvil
Identificar Vehículo	Se podrá elegir el vehículo que se está utilizando
Dados Obtenidos	Se guardará la información obtenida de los resultados
Configuración	Se mostrarán las configuraciones de la aplicación móvil

La utilización de cada uno de los botones es para abrir la actividad mencionada, la forma y color de cada uno de los botones están representados y designados en las carpetas “res”, además cada uno de íconos que son dinámicos se encuentran en esta carpeta dando un total de 8 íconos utilizados para los botones, además se encuentran como están organizados y estructurados en el archivo XML

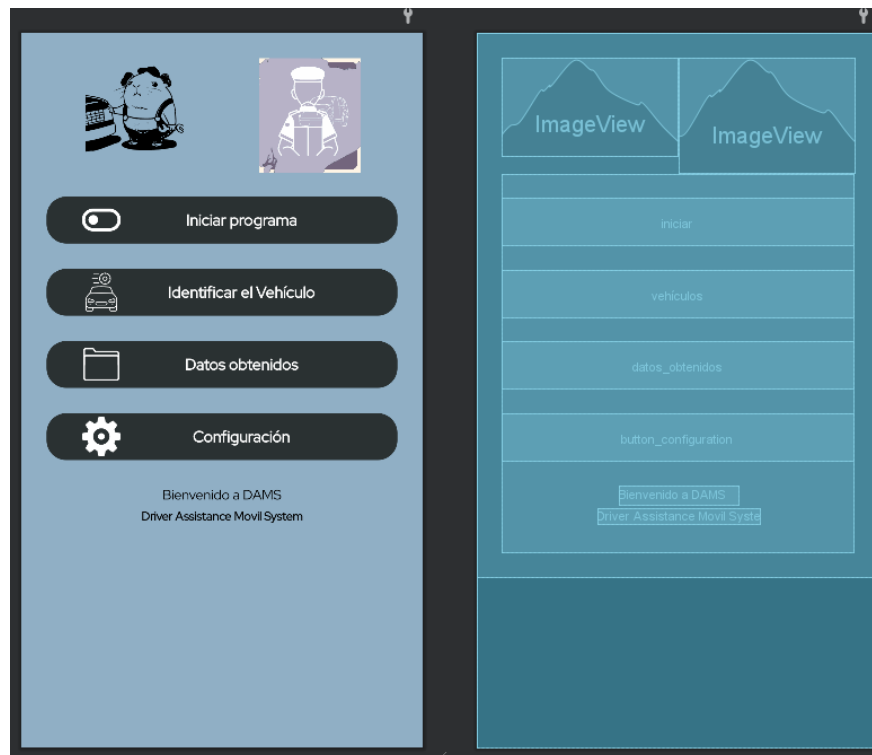


Figura 7: Interfaz

5.1.2 Diseño de layout de la pantalla principal para mostrar los valores de conducción

En esta pantalla es donde el usuario va a conocer las principales variables durante el trayecto de conducción para ello se presenta diferentes tipos de gráficos y textviews como se muestra en la figura 8, la primer gráfica es del tipo HalfGauge para la velocidad del vehículo donde tiene franjas de colores que representan la velocidad correcta que debe ir el conductor, un aviso antes de la velocidad máxima que corresponde a 5 Km/h menos y la velocidad que supera el límite dado por el gobierno.

Así mismo da el nombre de la vía, la velocidad crítica de la curva que corresponde a la velocidad en el punto donde el radio de curvatura es menor y la distancia que hay hasta el punto de que inicia la curva en la vía, de igual forma aparece un gráfico donde se puede observar la geometría de la curva.

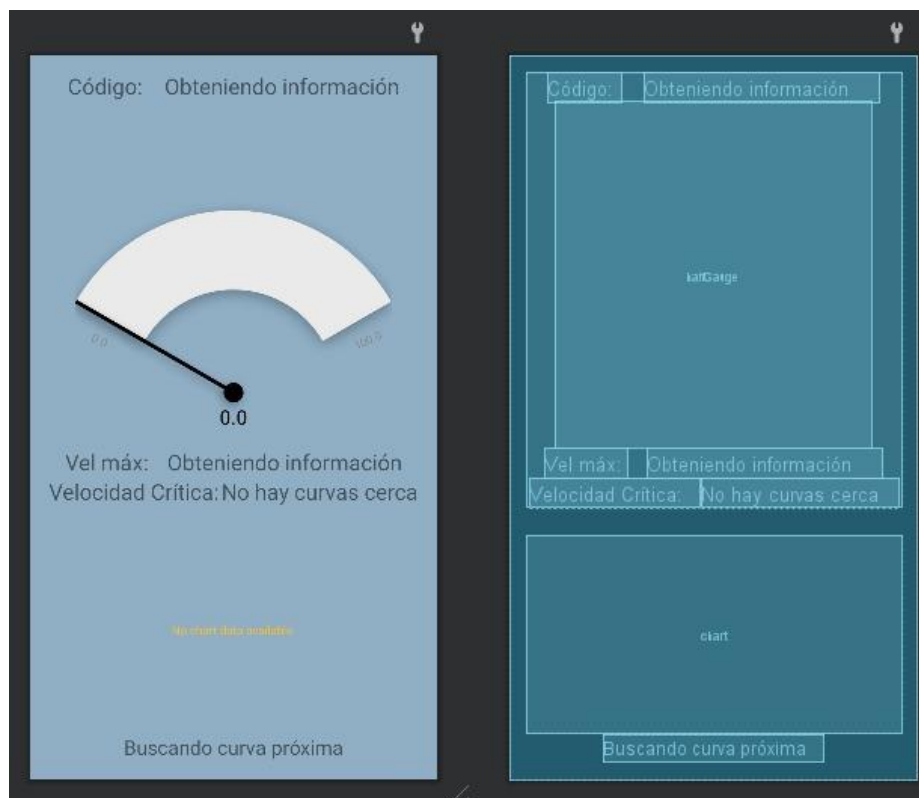


Figura 8: Diseño de layout de la interfaz principal

Debido al espacio que ocupan los gráficos, cuando se giran los celulares estos se desordenan por lo que no se puede apreciar las variables, para ello se crea otro con el fin de que tengan los mismos elementos pero que sólo se utilice cuando el teléfono sea girado la pantalla, ese mismo layout se puede observar en la figura 9.

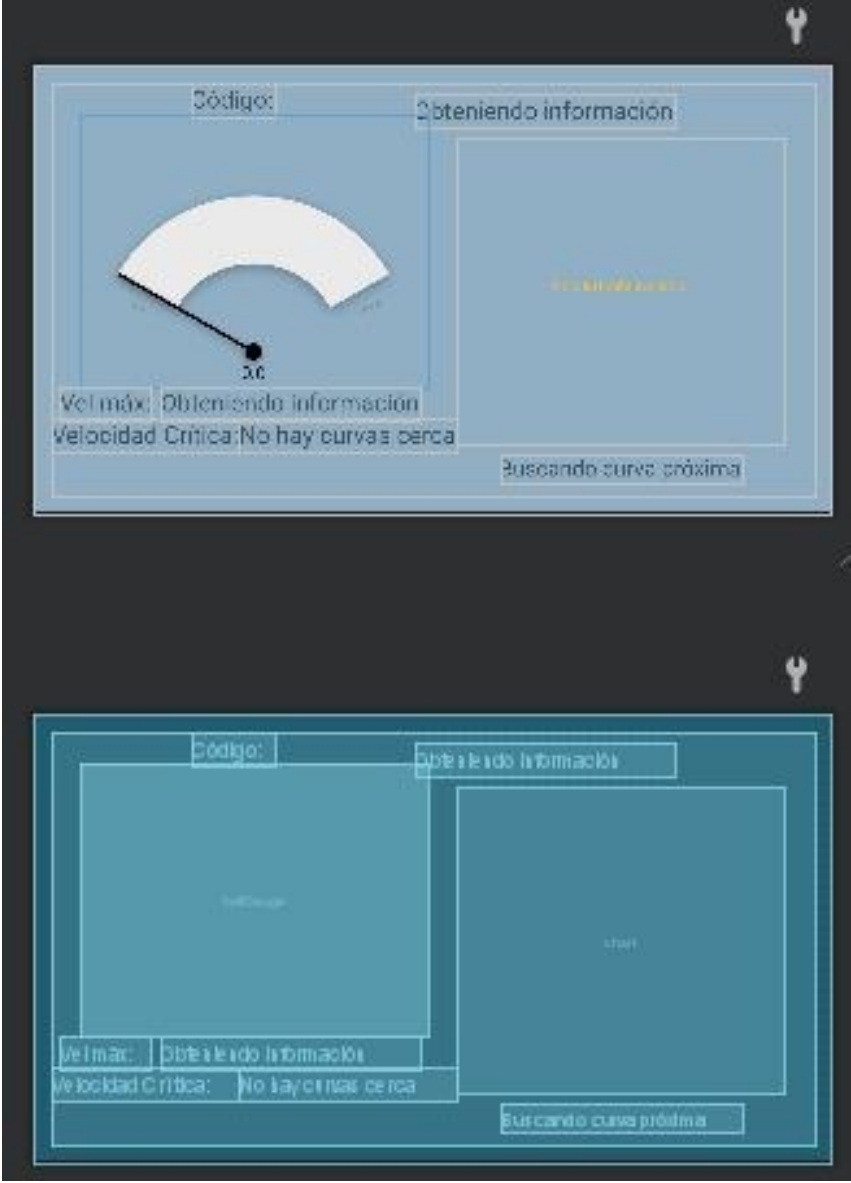


Figura 9: Diseño de la interfaz en pantalla horizontal

5.1.3 Diseño de layout de la pantalla para escoger el vehículo (activity_vehicle)

Dentro de este diseño de esta pantalla es sencillo debido que su función es escoger el tipo de vehículo que tenga el usuario, para el cálculo de distancia de frenado, por lo que únicamente presenta una lista, 2 t TextView y una lista, estos elementos se pueden observar en la figura 10.

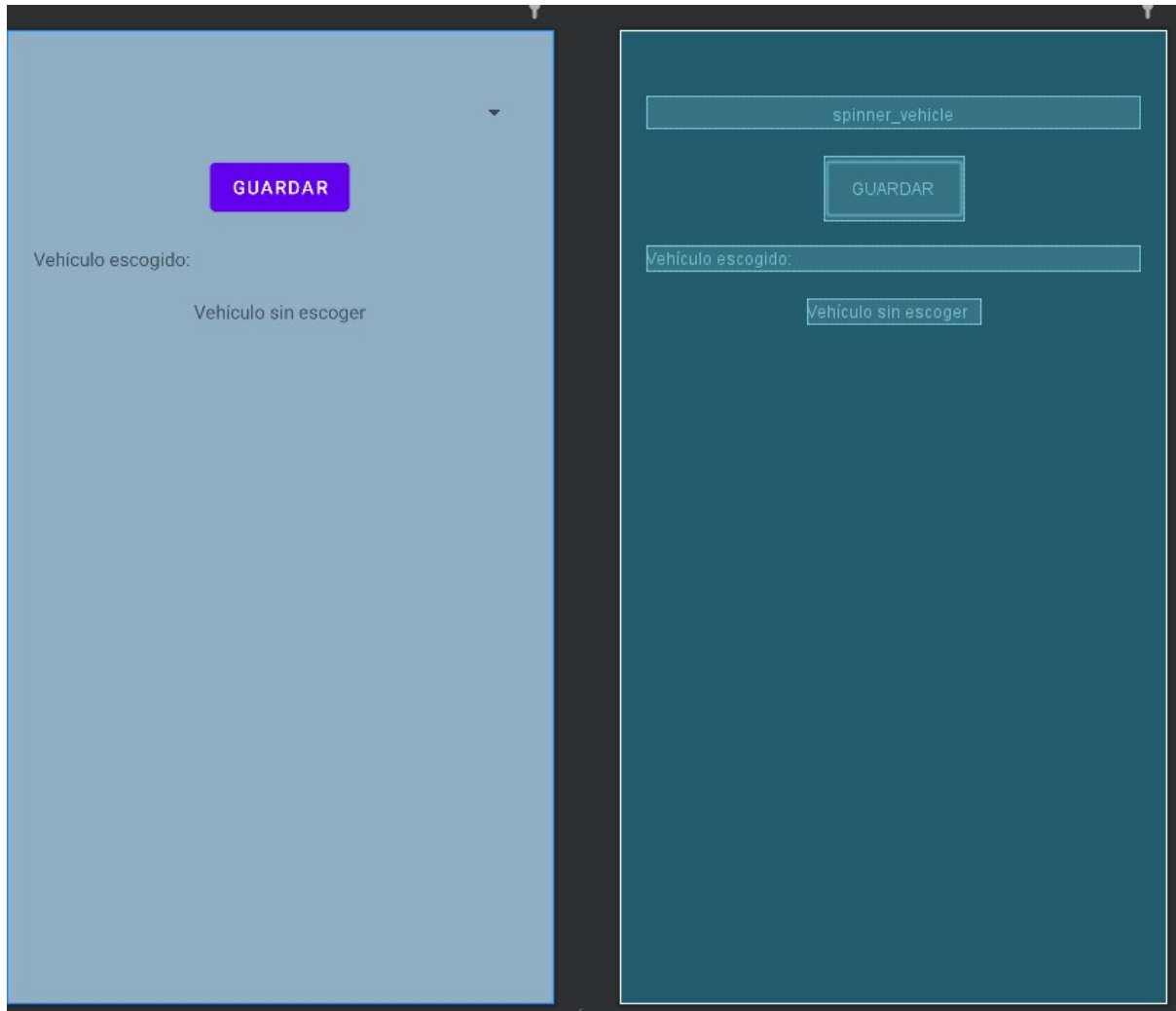


Figura 10: Diseño de Layout para escoger el vehículo

5.1.4 Diseño de layout de la pantalla de Datos Almacenados (activity_folder)

Dentro de esta actividad, como se puede apreciar en la figura 11 esta pantalla tiene únicamente un textView con el texto “Archivos Guardados” y una lista donde se muestran los archivos que se hayan guardado y un botón que sirve para compartir los archivos en otras plataformas.

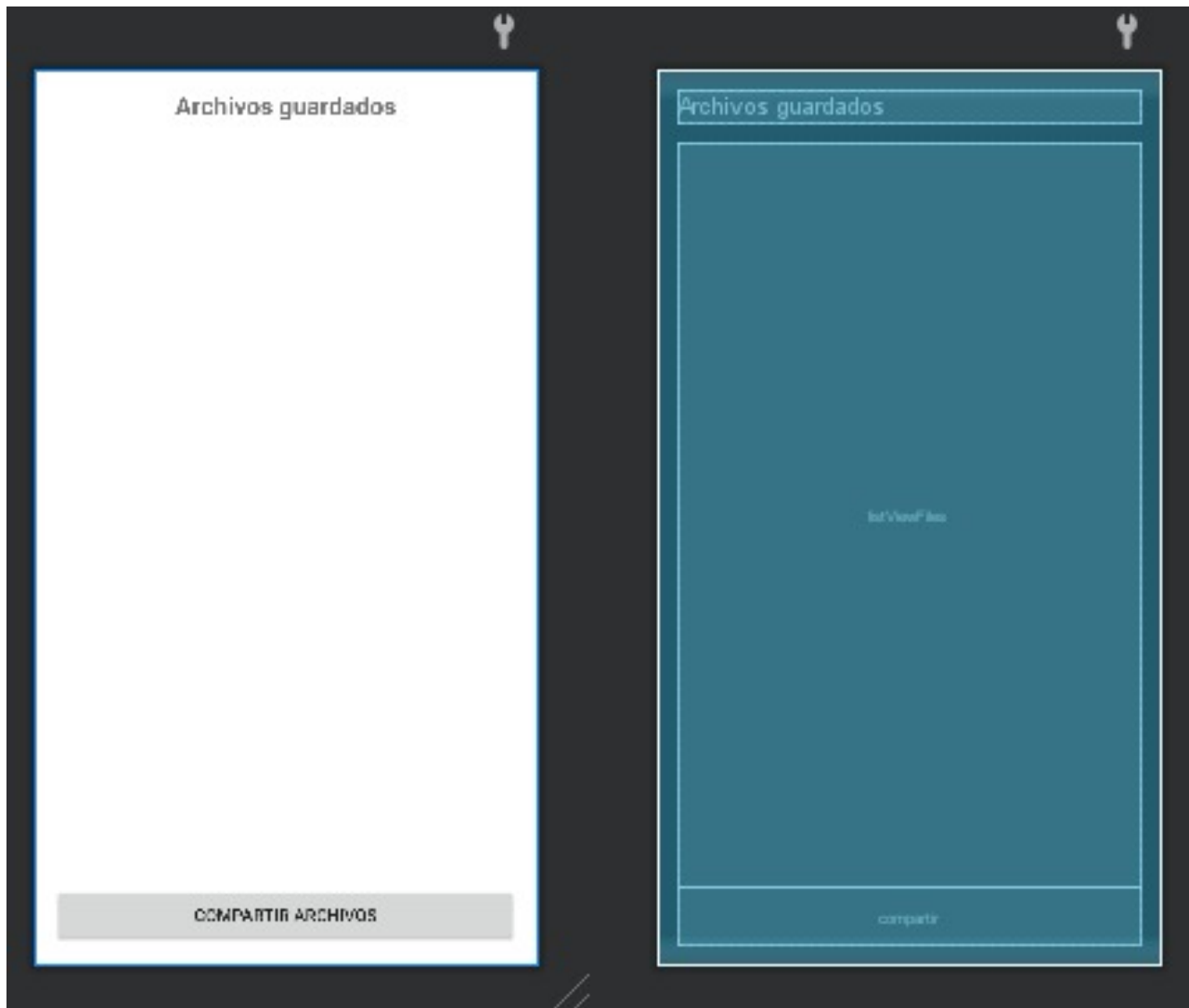


Figura 11: Diseño de Layout de almacenamiento de datos

5.1.5 Diseño de layout de la pantalla de configuración (activity_settings)

Dentro de este diseño se obtiene un botón donde el usuario puede directamente ingresar a la configuración de la aplicación y retirar el permiso de la aplicación para el uso del GPS, así como para que no se abra la aplicación bajo una acción determinada, que en el caso del proyecto es cuando se supere los 5km/h y que los datos no se guarden, para ello se utilizó un Switch que puede dar valores de falso o verdadero, que se creyó lo más conveniente para lo que se necesita.

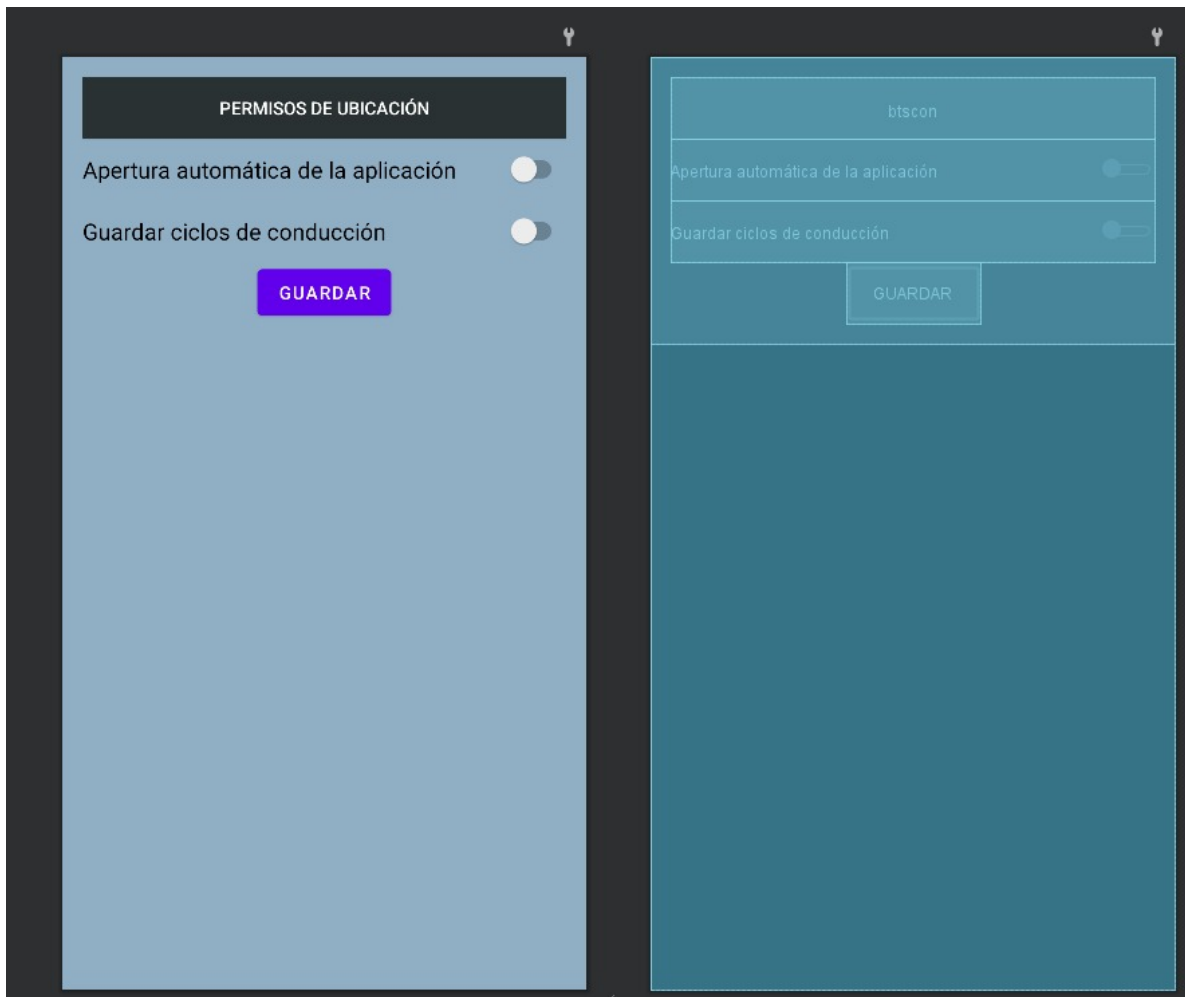


Figura 12: Diseño de Layout de la configuración

5.2 PROGRAMACION DE LAS ACTIVIDADES

Las actividades dentro de Android Studio son en su esencia todas las funciones que el programador quiera dar a su app, en la cual para mayor facilidad puede dividir las en muchos archivos en caso de que quiera dividir las acciones que realiza la aplicación para un manejo mucho más ameno. Aquí puede haber dos tipos de archivos, Kotlin y Java. Para el proyecto en su mayoría se utilizó Kotlin.

Para aquellos que tengan en su nombre la palabra “Activity” significa que esa actividad corresponde a una pantalla, mientras que, si no se encuentra en el nombre, el código en el archivo es una actividad que no necesariamente se necesite que sea visible para el usuario, esto se realiza con el objetivo de para cuando se realiza pruebas de funcionamiento, en caso de que algo no funcione correctamente se conozca que archivo puede estar generando el error y corregir la programación. Dentro del proyecto técnico se tiene las programaciones para cada una de las pantallas, además de otros archivos que realizan otras actividades.

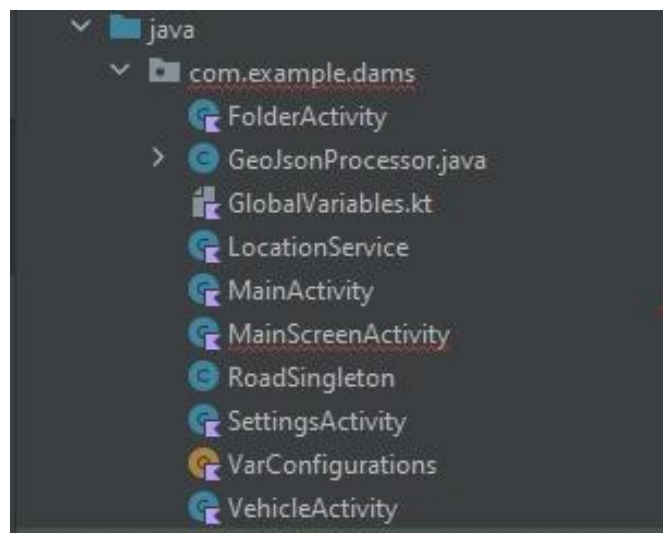


Figura 13: Actividades principales

5.2.1 Programación del mainactivity

Dentro del archivo llamado “MainActivity”, es el código que crea y carga los archivos del layout con el mismo nombre, además que aquí está el código para que los botones trabajen como uno desee.

Para ello dentro de la aplicación se genera un mensaje cuando se abra por primera vez la app, que son los términos de uso del usuario como se puede observar en la figura 14, en donde se muestra para que sirve la aplicación, a que permisos se necesita acceso para que funcione correctamente, y para que se va a utilizar el servicio de GPS.

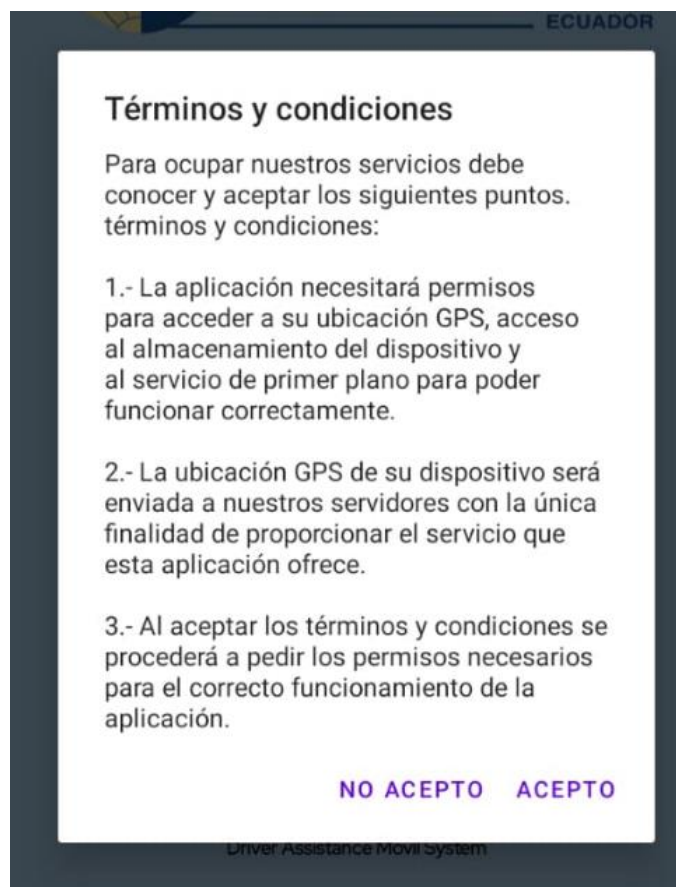


Figura 14: Términos y condiciones

En donde para ello debe dar aceptar al cuadro de diálogo caso contrario la aplicación se cierra si el usuario no acepta los términos y no podrá utilizar la aplicación, en caso de que no acepte el programa envía un mensaje con el siguiente contenido "Debes aceptar los términos y condiciones para usar la app."

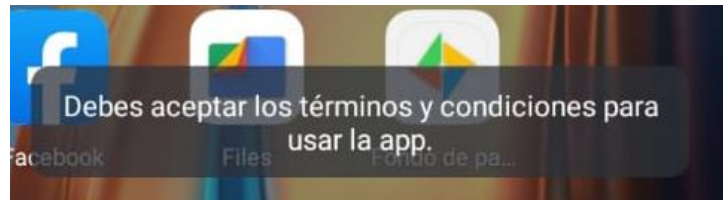


Figura 15: Mensaje de para aceptar términos y condiciones

Luego de eso para iniciar el programa, en caso de que no se escoge el vehículo saldrá otro mensaje en donde pide al usuario que escoja el vehículo, tiene el mismo formato que el anterior mensaje y el contenido del texto mostrado es el siguiente: "Por favor, elija un vehículo antes de continuar en el apartado de vehículos". Esto para conocer el vehículo donde se mueve el usuario y obtener datos como, área frontal, peso, entre otras variables que nos servirán más adelante para cálculos de frenado.

Una vez que se haya escogido el vehículo, cuando se de iniciar el botón revisa si los permisos de ubicación en primer plano y segundo plano, acceso a los archivos multimedia. Si el acceso no está permitido, pedirá a los usuarios cada uno de los permisos que necesita la aplicación como se puede apreciar en la figura 16.



Figura 16: Pantallas para permitir los accesos

De igual forma dentro del código se encuentra la asignación de cada acción para cada botones, así como el ocultar o hacer visible el botón de Datos guardados según el usuario quiera o no guardar la información del recorrido.

5.2.2 VehicleActivity

Dentro de esta actividad se encuentra la creación de la pantalla con los recursos que se ubicaron en el XML con el mismo nombre, además también se observa la lista con cada uno de los valores que corresponde a los vehículos, se declaran que valores se ubican en la lista y el cambio en los texview dependiendo del valor escogido, para que este valor tenga validez en el cambio, se debe

pulsar el botón que actualiza el valor del vehículo con el índice que ocupa y actualiza el valor en el textView, esto se puede observar en el código que se muestra en la figura 17.

```
private val opciones = listOf(
    OpcionSpinner( coeficiente: 0.3, distancia: 2.05, peso: 1150.0, nombre: "Auto(Sedán, Hatchback, CrossOver)" ),
    OpcionSpinner( coeficiente: 0.34, distancia: 2.4, peso: 1500.0, nombre: "SUV Mediano" ),
    OpcionSpinner( coeficiente: 0.37, distancia: 2.7, peso: 2000.0, nombre: "SUV Grande" ),
    OpcionSpinner( coeficiente: 0.38, distancia: 2.2, peso: 1100.0, nombre: "Camioneta Pequeña" ),
    OpcionSpinner( coeficiente: 0.4, distancia: 2.7, peso: 2000.0, nombre: "Camioneta Mediana" ),
    OpcionSpinner( coeficiente: 0.42, distancia: 3.1, peso: 2300.0, nombre: "Camioneta Grande" ),
    OpcionSpinner( coeficiente: 0.3, distancia: 2.5, peso: 1100.0, nombre: "Furgoneta Pequeña" )
)

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_vehicle)
    supportActionBar?.hide()

    vehicleSpinner = findViewById(R.id.spinner_vehicle)
    saveButton = findViewById(R.id.button_save)
    chosenVehicleTextView = findViewById(R.id.text_view_chosen_vehicle)
    notChosenVehicleTextView = findViewById(R.id.text_view_not_chosen_vehicle)

    val adapter = ArrayAdapter< context: this, android.R.layout.simple_spinner_item, opciones.map { it.nombre } >
    vehicleSpinner.adapter = adapter

    val sharedPreferences = getSharedPreferences( name: "MyPrefs", Context.MODE_PRIVATE )
    val savedVehicle = sharedPreferences.getString( key: "SavedVehicle", defaultValue: "Vehículo sin escoger" )
    notChosenVehicleTextView.text = savedVehicle

    saveButton.setOnClickListener { @View()
        val chosenVehicle = opciones[vehicleSpinner.selectedItemPosition]
        sharedPreferences.edit().putString( "SavedVehicle", chosenVehicle.nombre ).apply()
        notChosenVehicleTextView.text = chosenVehicle.nombre
        Log.d( tag: "MyApp", msg: "Vehículo guardado: $savedVehicle" )
        finish()
    }
}
```

Figura 17: Codificación de VehicleActivity

5.2.3 Folder activity

De igual forma con los anteriores archivos, esta programación corresponde a un layout, por lo que sus primeras líneas es inicializar los valores y los recursos que se crearon para mostrar en la pantalla, en esta programación se realiza una comprobación si la carpeta donde se guardan los archivos está creada, caso contrario la crea y guardar la ruta en una variable para luego acceder a ella.

Dentro de esta actividad con la ruta se actualiza la lista para que los datos guardados sean observados, además se crea la funcionalidad y la función del botón para compartir los archivos seleccionados y eliminar los archivos que no necesiten. Si los archivos seleccionados son mayores a 3 elementos estos se comprimen para no ocupar mucho espacio para el mensaje enviado.

```

listView.setOnItemClickListener = AdapterView.OnItemClickListener { _, _, position, _ ->
    if (isLongClickActive) {
        toggleSelection(position) // Alternar la selección del elemento en la posición dada
    } else {
        val selectedFile = files[position]
        openSelectedFile(selectedFile) // Abrir el archivo seleccionado solo si no estás en modo de selección múltiple
    }
}

listView.setOnItemLongClickListener { _, _, position, _ ->
    if (!isLongClickActive) {
        // En modo no selección, cambiar a modo selección múltiple
        listView.choiceMode = ListView.CHOICE_MODE_MULTIPLE
        listView.adapter = mListviewAdapterMultiple
        toggleSelection(position) // Seleccionar el elemento actual

        isLongClickActive = true // Activar el modo de selección prolongada
        true ^setOnItemLongClickListener // Indicar que se ha consumido el evento
    } else {
        // En modo selección múltiple, desactivar el modo de selección prolongada
        isLongClickActive = false
        false ^setOnItemLongClickListener // Permitir que se ejecute el evento "OnClick" del elemento
    }
}
}

private fun getDataFolder(): File {
    val parentDirectory = getExternalFilesDir(Environment.DIRECTORY_DOCUMENTS)
    val dataDirectory = File(parentDirectory, child: "Datos Obtenidos")
    if (!dataDirectory.exists()) {
        dataDirectory.mkdirs() // Crea el directorio si no existe
    }
    return dataDirectory
}
}

```

Figura 18: Codificación de Folder activity

5.2.4 SettingsActivity

Este archivo compuesto de varias líneas de código, lo que realiza es guardar el valor del Switch en valores globales para que se pueda acceder en todo el proyecto, además se muestra un botón por si el usuario decide eliminar los permisos otorgados a la aplicación, debido a que el usuario es el único capaz de quitar y eliminar los permisos concedidos anteriormente, además vale aclarar que los cambios sólo se guardan si se da al botón guardar.

```

supportActionBar?.hide()
setContentView(R.layout.activity_settings)
btscan = findViewById(R.id.btscan)
switchLocationService = findViewById(R.id.switchLocationService)
btnSave = findViewById(R.id.btnSave)
switchCycleData = findViewById(R.id.cycledata)
switchCycleData.isChecked = VarConfigurations.isCycleDataEnabled
switchLocationService.isChecked = VarConfigurations.Serviceacti
sharedPreferences = getSharedPreferences( name: "MyPrefs", MODE_PRIVATE)
switchCycleData.isChecked = sharedPreferences.getBoolean( key: "isCycleDataEnabled", defValue: false)
switchLocationService.isChecked = sharedPreferences.getBoolean( key: "Serviceacti", defValue: false)

switchCycleData.setOnCheckedChangeListener { _, isChecked ->
    VarConfigurations.isCycleDataEnabled = isChecked
    val sharedPreferences = getSharedPreferences( name: "MyPrefs", MODE_PRIVATE)
    sharedPreferences.edit().putBoolean("isCycleDataEnabled", isChecked).apply()
}
switchLocationService.setOnCheckedChangeListener { _, isChecked ->
    if (isChecked) {
        // Acciones a realizar cuando el Switch está encendido
        VarConfigurations.Serviceacti = true
        val sharedPreferences = getSharedPreferences( name: "MyPrefs", MODE_PRIVATE)
        sharedPreferences.edit().putBoolean("Serviceacti", true).apply()
        val serviceIntent = Intent( packageContext: this, LocationService::class.java)
        startService(serviceIntent)
        // Otras acciones adicionales si es necesario
    } else {
        // Acciones a realizar cuando el Switch está apagado
        VarConfigurations.Serviceacti = false
        val sharedPreferences = getSharedPreferences( name: "MyPrefs", MODE_PRIVATE)
        sharedPreferences.edit().putBoolean("Serviceacti", false).apply()

        val intent = Intent( packageContext: this, LocationService::class.java)
        stopService(intent)
        // Otras acciones adicionales si es necesario
    }
}
btscan.setOnClickListener { it: View!
    openAppSettings()
    finish()
}

```

Figura 19: Codificación de SettingsActivity

5.2.5 Variables globales

En este apartado tenemos variables que se utiliza de manera global y que se utiliza como variables para configuración, se lo realizo de esta manera para tener un mayor control sobre ellas y poder modificar el valor mucho más fácil, estás variables se actualizan en otras actividades.

```
MainScreenActivity.kt x GeoJsonProcessor.java x GlobalVariables.kt x
package com.example.dams

object VarConfigurations {
    var isCycleDataEnabled = true
    var Serviceacti = false
    var SavedVehicle = "Vehículo sin escoger"
}
```

```
package com.example.dams

import java.io.Serializable

object GlobalVariables {
    var altitude: Double = 0.0
    var latitude: Double = 0.0
    var longitude: Double = 0.0
    var speed: Float = 0f
    var fileDirectory: String = ""
    var date: String = ""
    var time: String = ""
    var diston: Double = 0.0
    var velomax: Double = 0.0

    val opciones = listOf(
        OpcionSpinner( coeficiente: 0.3, distancia: 2.05, peso: 1150.0, nombre: "Auto(Sedán, Hatchback, CrossOver)" ),
        OpcionSpinner( coeficiente: 0.34, distancia: 2.4, peso: 1500.0, nombre: "SUV Mediano" ),
        OpcionSpinner( coeficiente: 0.37, distancia: 2.7, peso: 2000.0, nombre: "SUV Grande" ),
        OpcionSpinner( coeficiente: 0.38, distancia: 2.2, peso: 1100.0, nombre: "Camioneta Pequeña" ),
        OpcionSpinner( coeficiente: 0.4, distancia: 2.7, peso: 2000.0, nombre: "Camioneta Mediana" ),
        OpcionSpinner( coeficiente: 0.42, distancia: 3.1, peso: 2300.0, nombre: "Camioneta Grande" ),
        OpcionSpinner( coeficiente: 0.3, distancia: 2.5, peso: 1100.0, nombre: "Furgoneta Pequeña" )
    )
}

data class OpcionSpinner(val coeficiente: Double, val distancia: Double, val peso: Double, val nombre: String) :
    Serializable
```

Figura 20: Variables globales y configuraciones

5.2.6 Código de tratamiento de datos

Los datos para las vías pueden ser consultadas de varias plataformas que ofrecen el servicio de mapas, sin embargo, muchas de estas cobran cierta cantidad de dinero por el uso de sus servicios y por derechos de autor, un ejemplo de ello es Google Maps, que es gratuita hasta cierto límite de llamados. En el proyecto se utilizó los mapas proporcionados por OpenStreetMap. Para descargar los datos que estos manejan, se lo puede realizar por medio de Overpass Api, que es una dirección web propia de ellos para realizar consultas. Para las consultas maneja su propio lenguaje donde

ubicamos las carreteras de interés para el proyecto y aquellas que unen a este tipo de carreteras, la estructura del llamado se puede apreciar en la imagen presentada a continuación.

```
/*
This has been generated by the overpass-turbo wizard.
The original search was:
"highway=primary"
*/
[out:json][timeout:25];
// gather results
(
  // query part for: "highway=primary"
  way["highway"="motorway"]({{bbox}});
  way["highway"="primary"]({{bbox}});
  way["highway"="trunk"]({{bbox}});
  way["highway"="secondary"]({{bbox}});
  way["highway"="motorway_link"]({{bbox}});
  way["highway"="trunk_link"]({{bbox}});
  way["highway"="primary_link"]({{bbox}});
  way["highway"="secondary_link"]({{bbox}});
);
// print results
out body;
>;
out skel qt;
```

Figura 21: Codificación para consulta de datos

Para obtener los datos de las carreteras se necesita delimitar la zona donde se obtenga la información, esto se lo realiza en la propia plataforma sin que nosotros coloquemos los valores, únicamente el recuadro que está en la parte derecha funciona como un delimitador, se realiza el llamado y se obtienen las vías y su información, para luego exportarla como formato geojson. Las carreteras que de las que se han tenido información se colorean de una tonalidad azul, y los puntos que representan a los nodos de dicha vía, esto se puede observar en el siguiente gráfico.

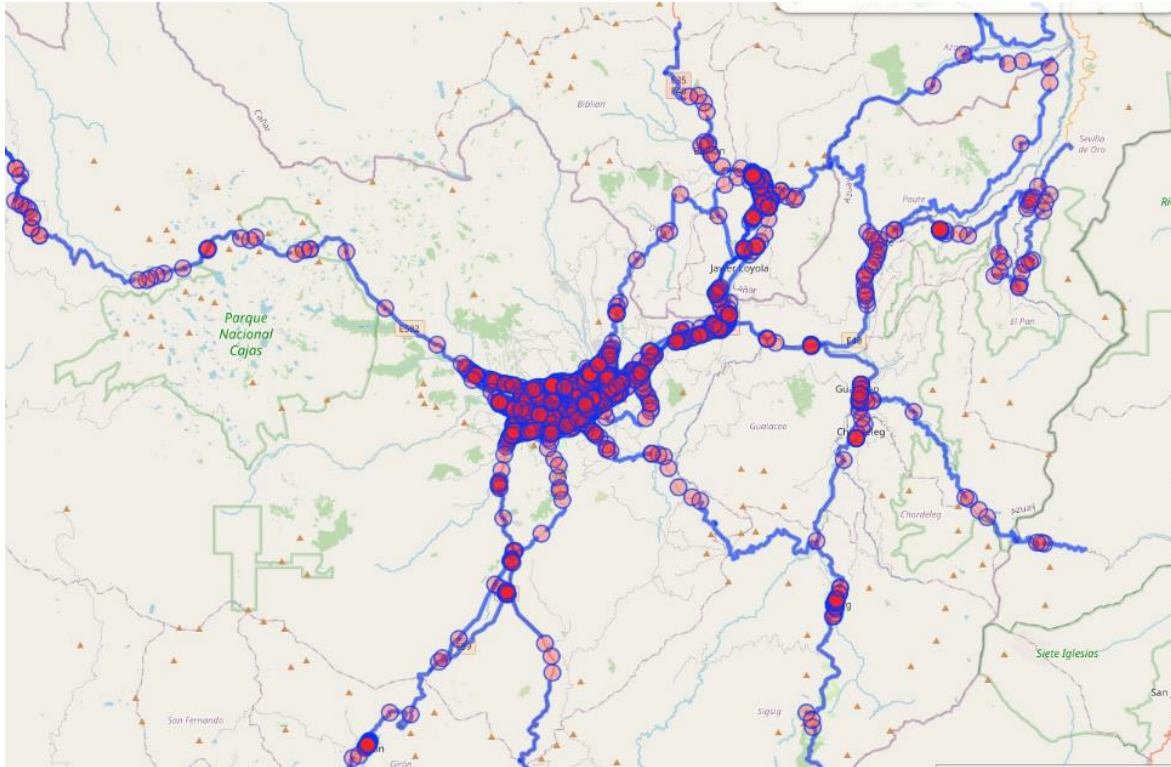


Figura 22: Datos de la consulta obtenidos.

Los datos que se obtienen de la Api Overpass, son del formato geojson, se necesitan de alguna forma extraer los datos y guardar en nuestra aplicación para la utilización los diferentes cálculos, para ello únicamente necesitamos el nombre de la vía, el código, la velocidad máxima y la geometría, para facilitar los cálculos debemos tener la información en la RAM, así que las cargamos como listas, hay que mencionar que las carreteras están construidas por segmentos por lo que una misma vía puede tener varios segmentos es por eso que tenerla la información en la RAM nos puede ayudar a buscar la vía más cercana.


```

GsonBuilder gsonBuilder = new GsonBuilder();
gsonBuilder.serializeSpecialFloatingPointValues(); // Permitir la serialización de valores especiales de punto
Gson gson = gsonBuilder.create();

try {
    // Comprobar si el archivo existe y se puede abrir correctamente
    String[] fileList = context.getAssets().list( path: "");
    if (!Arrays.asList(fileList).contains(fileName)) {
        Log.e( tag: "GeoJsonProcessor", msg: "Archivo no encontrado: " + fileName);
        return;
    }

    InputStream is = context.getAssets().open(fileName);
    Reader reader = new InputStreamReader(is);

    // Resto del código para leer y procesar el archivo GeoJSON
    // ...
} catch (IOException e) {
    // Manejo de errores
    Log.e( tag: "GeoJsonProcessor", msg: "Error al abrir el archivo: " + fileName);
}

try (InputStream is = context.getAssets().open(fileName);
    Reader reader = new InputStreamReader(is)) {
    GeoJsonFeatureCollection featureCollection = gson.fromJson(reader, featureCollectionType);

    List<Road> roads = new ArrayList<>();
    for (GeoJsonFeature feature : featureCollection.getFeatures()) {
        List<Coordinate> coordinates = new ArrayList<>();

        for (List<Double> point : feature.getGeometry().getCoordinates()) {
            coordinates.add(new Coordinate(point.get(0), point.get(1)));
        }

        LineString lineString = geometryFactory.createLineString(coordinates.toArray(new Coordinate[0]));
        String id = feature.getProperties().getId();
        String name = feature.getProperties().getName();
        Integer maxSpeed = feature.getProperties().getMaxspeed();
    }
}

```

Figura 23: Codificación de tratamiento de datos

Esto se realiza debido a que la estructura de los datos es compleja, y almacena muchas informaciones que actualmente no se necesita para el funcionamiento de la aplicación, por eso los únicos que se utilizan son el nombre de la vía, la velocidad máxima, el código de la vía y las coordenadas de la vía, para almacenarlas en una lista que se utiliza después.

```

{
  "type": "Feature",
  "properties": {
    "@id": "way/29223188",
    "highway": "primary",
    "lanes": "2",
    "maxspeed": "50",
    "name": "Avenida de las Am\u00e9ricas",
    "oneway": "yes",
    "sidewalk": "right",
    "surface": "asphalt"
  },
  "geometry": {
    "type": "LineString",
    "coordinates": [
      [
        -79.0334169,
        -2.9067077
      ],
      [
        -79.0334321,
        -2.9067327
      ],
      [
        -79.0335178,
        -2.9068151
      ]
    ]
  }
},
  "id": "way/29223188"
}

```

Figura 24: Estructura de datos de GeoJSON.

Tabla 4: Significado de la estructura de GeoJSON.

Numero	Objeto	Significado
1	"@id": "way/29223188"	Es un identificador num\u00e9rico \u00fanico para cada v\u00eda, as\u00ed tenga diferentes segmentos comparten un mismo id
2	"highway": "primary"	Clasificaci\u00f3n de v\u00edas seg\u00fan OpenStreetmap y se puede obtener v\u00edas, primarias, secundarias, terciarias (motorway, trunk, primary, secondary, tertiary, unclassified, residential)
3	"lanes": "2"	Cantidad de carriles que existen en la v\u00eda
4	"maxspeed": "50"	Es el l\u00edmite de velocidad que puede llegar el veh\u00edculo en la v\u00eda
5	"name": "Avenida de las Am\u00e9ricas"	Es el nombre de la v\u00eda
6	"oneway": "yes"	Es el sentido de la v\u00eda, puede ser de un solo sentido como tambi\u00e9n de doble sentido
7	"sidewalk": "right"	Es la acera que tiene la v\u00eda
8	"surface": "asphalt"	Material del que est\u00e1 compuesto la v\u00eda, por ejemplo, asfalto
9	"geometry"	Es la geometr\u00eda de la v\u00eda, es decir todas las coordenadas que corresponden al segmento

5.2.7 Main screen activity

Esta es la programación principal del proyecto donde su función es reproducir la alarma si la velocidad es mayor a la permitida en la vía, así como indicar al usuario si hay curvas en la vía próximas al vehículo en la carretera. Debido a que el código es muy extenso, se ha idealizado presentar mediante diagrama de flujos cada parte e indicar que se realiza en cada función.

En primer lugar, se crean las variables que se van a utilizar a lo largo de la vida de la actividad, así como el llamado del XML y los gráficos que se utiliza para mostrar la velocidad y las curvas que estén cerca al usuario. Además, dentro de esta parte del ciclo de vida se proporciona un locationlistener donde actualiza los datos del GPS cada segundo, en donde se obtienen latitud, longitud, velocidad y dirección del movimiento, además busca la carretera más cercana a mi posición donde asigna valores de velocidad máxima dependiendo del valor que hay, caso contrario ubica como 50.

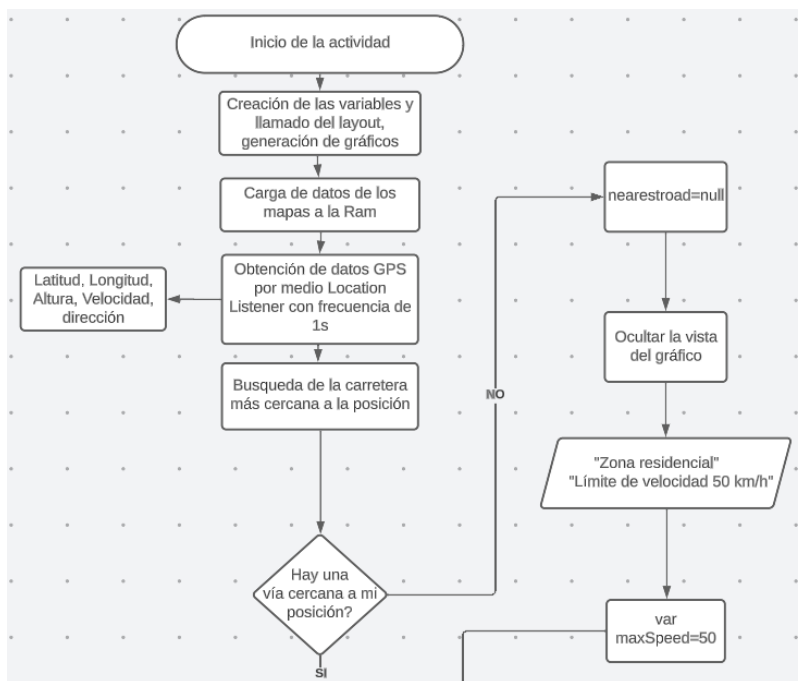


Figura 25: Diagrama de flujo de la parte inicial del programa

Una vez realizado ese paso, se obtiene la velocidad del vehículo con el fin de obtener el valor a la cual se encuentra circulando, debido a problemas con la velocidad baja y que el usuario puede estar caminando se coloca que la velocidad para que funcione sea mayor que 3, compara además el valor de la velocidad que circula con la velocidad máxima de la zona, si se supera la alarma comienza a sonar dependiendo de la velocidad que se mueve, también avisa cuando se esté acercando a la velocidad límite.

Si la velocidad supera se crea un vector de dirección que involucra la distancia de frenado también calculada en esta condición, caso que no supere la velocidad se elimina los datos de las listas de los segmentos usados y las curvas cercanas para liberar memoria, también la alarma deja de emitirse y el valor del vector de dirección se ubica en nulo.

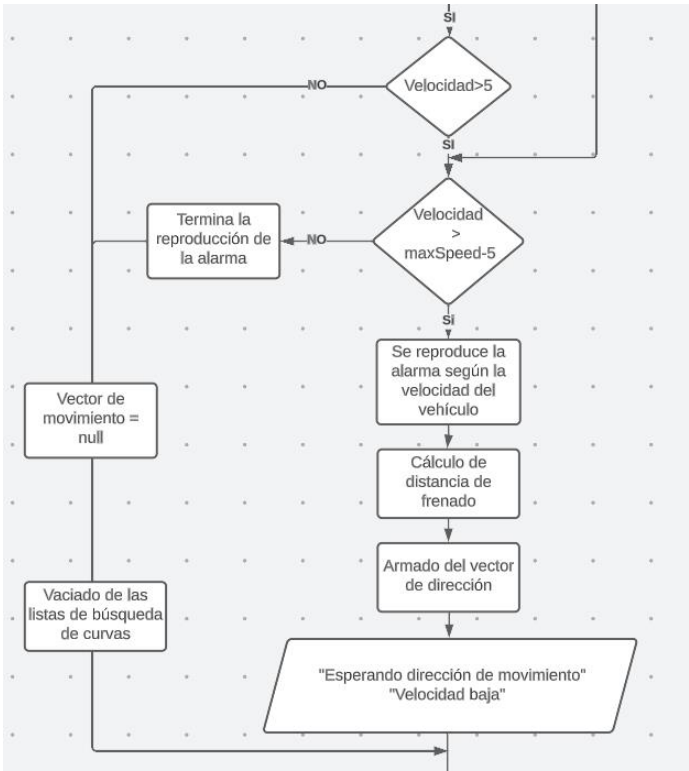


Figura 26: Parte del diagrama de flujo para el condicionante de velocidad

Antes de la llegada de la búsqueda de curvas cercanas, primeramente, se suaviza los valores de velocidad y el de pendiente con 5 instancias anteriores, debido a que son valores que cambian drásticamente por lo que, para el cálculo de distancia según Aparicio, puede dar a llegar valores muy variados.

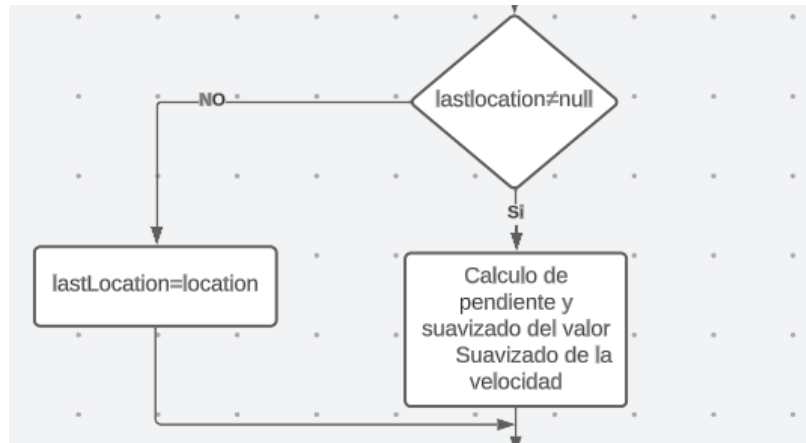


Figura 27: Parte del diagrama de flujo para el suavizado de datos

Después se hace un llamado de las variables del vector y la vía más cercana, en caso de que ninguno de los dos sea nulo se ejecuta en un hilo paralelo donde se busquen las curvas más cercanas, sin embargo, dentro de esto podemos tener más instancias las cuales son que si el segmento pertenece con anterioridad a los segmentos buscados con el fin de inicializar las curvas cuando el vehículo parte, o hay cambio de carril, esta parte a pesar de ser condicionante sólo se ejecuta una vez. Además, tenemos también si ya se buscó curvas en ese segmento, si encontró curvas y cuando no hay, esto será detallado más adelante. En el siguiente gráfico se colocará los posibles casos que encontremos.

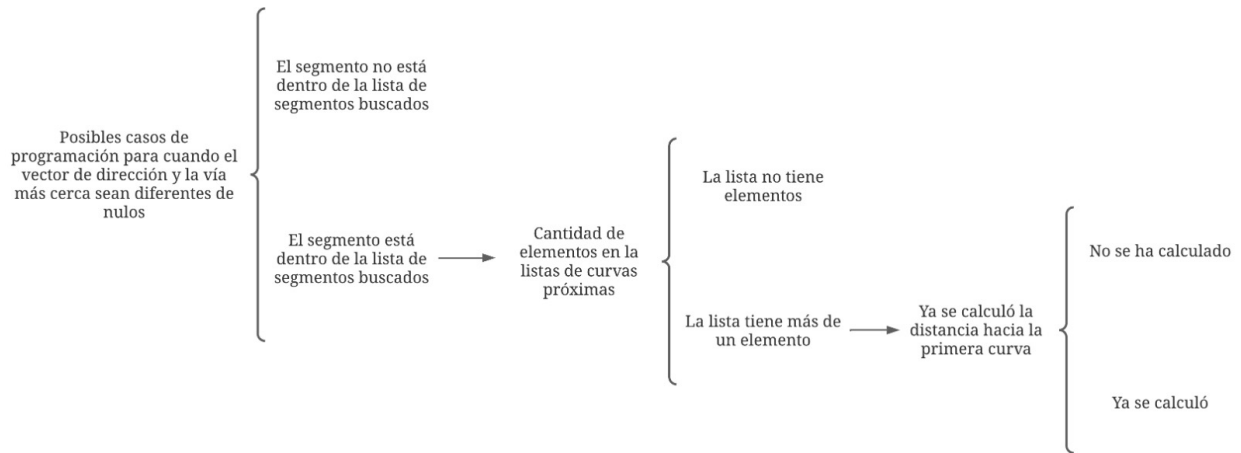


Figura 28: Posible casos para los cálculos de curvas

5.2.8 El segmento no está dentro de la lista

Cuando el segmento no está dentro de la lista es mucho más sencillo, puesto que significa que no se ha calculado ni las curvas próximas o el vehículo recién se empieza a mover, por lo que únicamente se busca la curvas en los próximos 750 metros, buscando desde el segmento actual y los segmentos consiguientes guardando en el proceso que segmentos se utilizó para la búsqueda.

Debido a que los segmentos de algunas vías son para ambos sentidos es vital tener cuidado en esos casos, eso se lo hace mediante los vectores donde calcula si estos tienen la misma dirección o si no cuál es el ángulo entre los dos, en caso de que estén en sentidos contrarios es decir 180 grados, los puntos se reordenan. Se considera una diferencia de error de 30 grados debido a que no siempre los segmentos se van a encontrar exactamente a esa diferencia de grados.

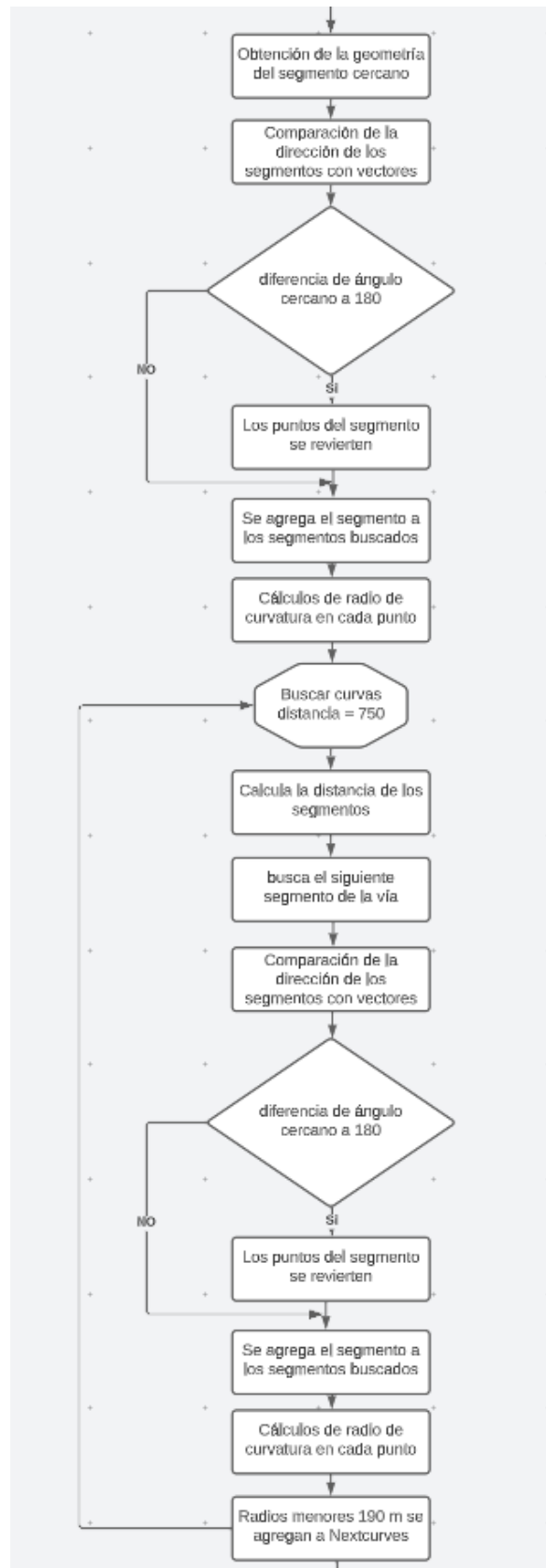


Figura 29: Diagrama de flujo cuando el segmento no está dentro de la lista

5.2.9 El segmento está dentro de la lista

En este caso se toma que el segmento de donde estamos ya se ha buscado con anterioridad, por lo que no vuelve a buscar, simplemente preguntamos cuantos elementos tiene la lista de curvas siguientes siendo el tamaño de la lista, según esto podremos saber si dentro los 750 metros hay o no hay curvas, si esto sucede tenemos diferentes manejos para cada caso.

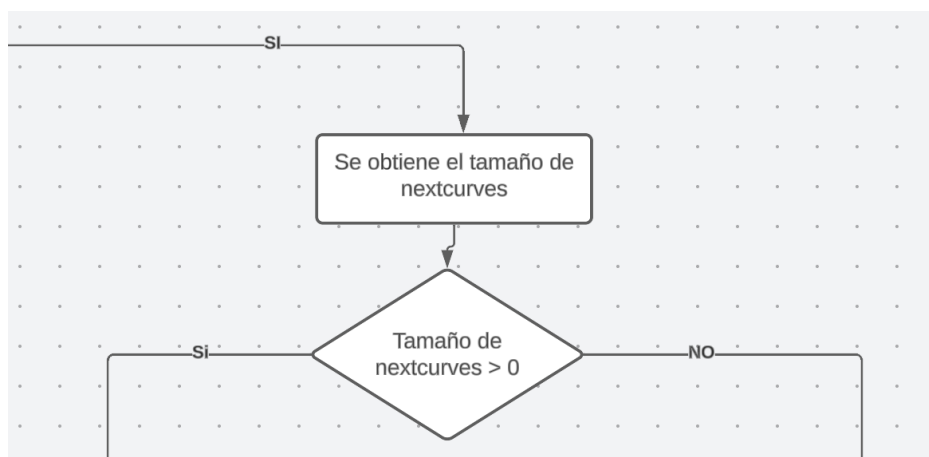


Figura 30: Diagrama cuando el segmento está dentro de la lista

5.2.9.1 Tamaño de curvas 0

Como se mencionó anteriormente, esto nos indica que en los próximos 750 metros no hay curvas cerca, por lo que no se le avisa al conductor, pero si comenzamos a obtener la distancia que recorre con el fin de cuando el vehículo recorra 650, comience de nuevo la búsqueda de las curvas siguiendo el mismo código anterior. Donde se considera el ángulo de los vectores, y la distancia que se recorre buscando la siguiente curva, para ese proceso cabe mencionar que la distancia se calcula punto por punto del segmento.

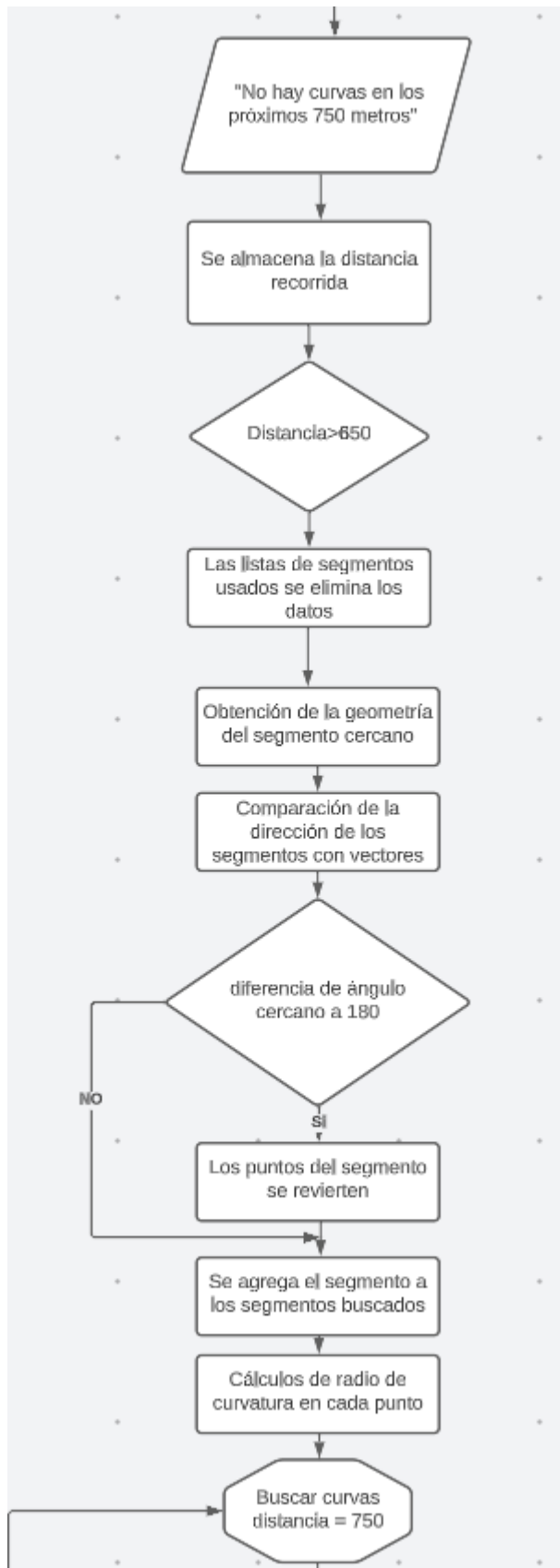


Figura 31: Diagrama cuando el tamaño de curva es cero.

5.2.9.2 Tamaño de curvas mayores que 1

Significa que existe curvas detectadas en la vía que se circule, aunque aún no se calcula la distancia que hay desde la posición del vehículo hacia la curva, por eso al igual que los anteriores casos se maneja dos cada uno con su respectivo código. Para ello compara el valor de la distancia que se tiene en memoria.

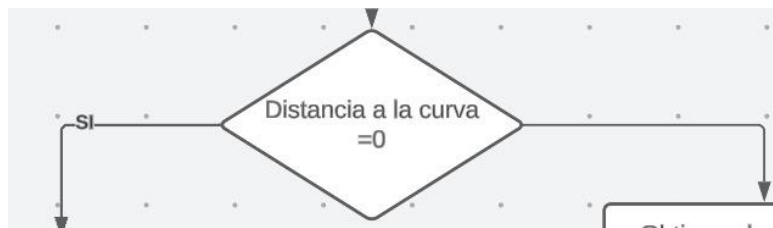


Figura 32: Diagrama de tamaño de curva mayor a 1.

5.2.9.2.1 Distancia no calculada o igual a cero

Este caso se ejecuta cuando recién se ha buscado la curva y existen elementos dentro de esta, para ello se calcula la distancia que se tiene desde el punto actual hasta el inicio de la curva, y se la guarda en la variable que se utilizó en el inicio para discernir los casos. Para ello busca el punto más cercano de nuestra ubicación e itera sobre todos los puntos de los segmentos utilizados y guardados en la variable, hasta encontrar el primer punto de la curva. Además, se obtienen los datos de la curva, es decir todas las coordenadas que son parte y son únicamente de la curva, para presentarla en el gráfico utilizando las coordenadas como puntos para graficar y unirlos mediante una línea. Este apartado sólo calcula la distancia que hay desde el usuario hasta la vía.

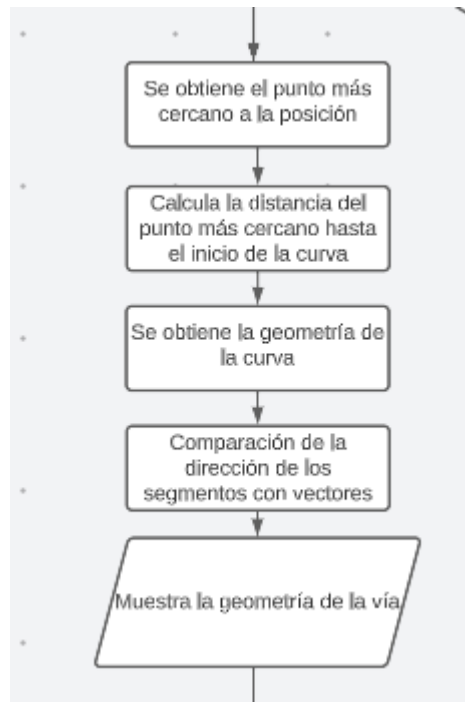


Figura 33: Distancia no calculada e igual a cero.

5.2.9.3 Distancia diferente de cero

Este caso la distancia ya se tiene desde el usuario hasta la curva, por lo que en este apartado se comienza a almacenar la información de la distancia recorrida del vehículo y se le resta dos distancias, la distancia inicial de la curva y la distancia según el frenado del vehículo y se almacena en variables diferentes, para tener la distancia crítica donde debe bajar la velocidad y mostrar cuanta falta para la curva, además obtiene el radio mínimo de la curva y obtiene la velocidad de derrape. Con estos datos se compara la velocidad del vehículo con la de la velocidad de derrape y la distancia crítica, en caso de que ambas se cumplan suena la alarma para que reduzca la velocidad del vehículo hasta la velocidad adecuada para tomar la curva.

Una vez que haya tomado la curva, se inicializa las variables de distancia en cero y se elimina el primer elemento de la lista de nextcurve, que corresponde a la primera curva. Luego pasado los 600

metros calcula primero el radio del punto y pasado los 650 metros verifica si el punto es menor al radio de 190, si se cumplen las dos condiciones calcula las siguientes curvas y reinicia los valores de la lista y de las distancias.

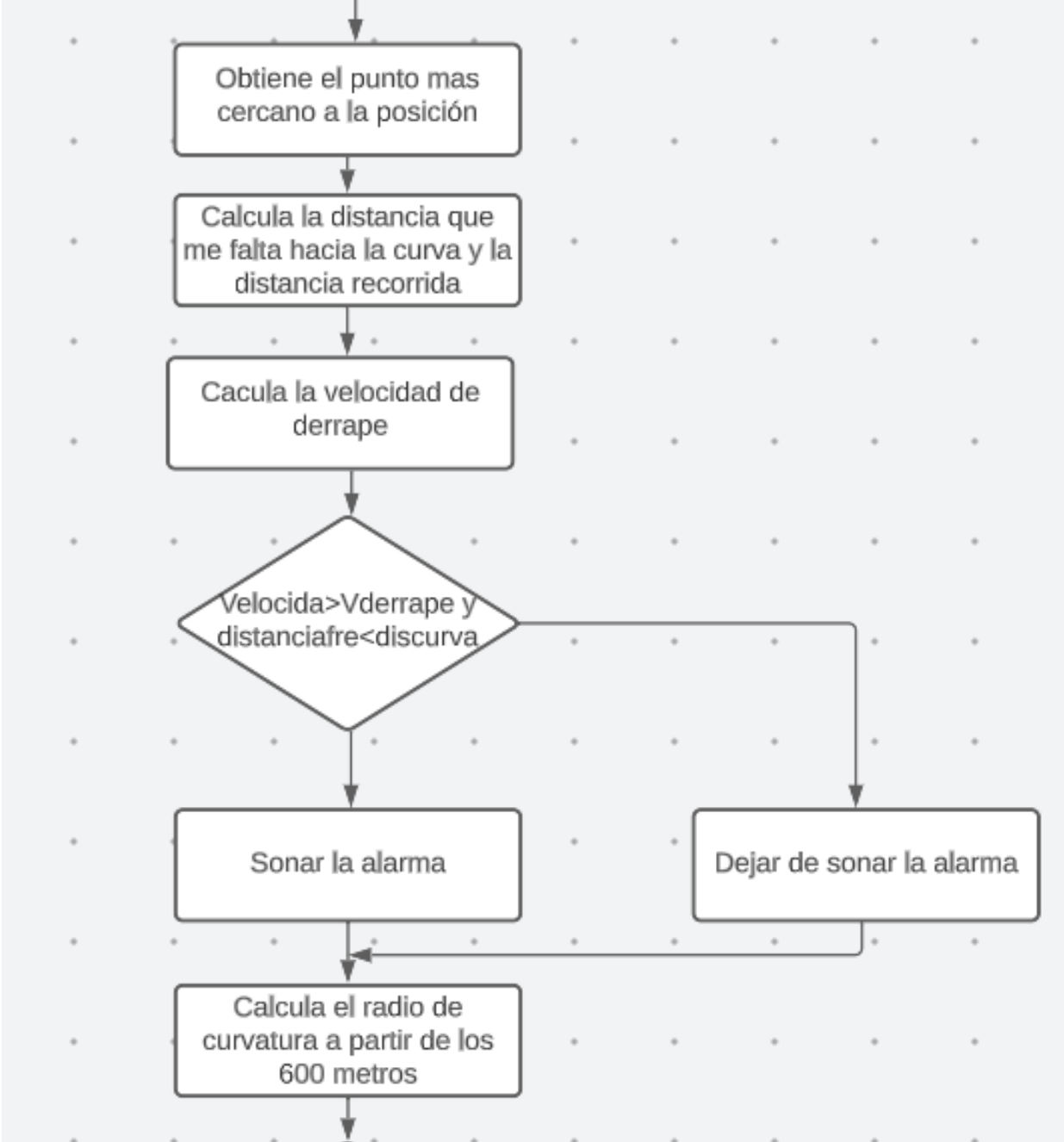


Figura 34: Diagrama de distancia diferente de cero (Primera Parte)

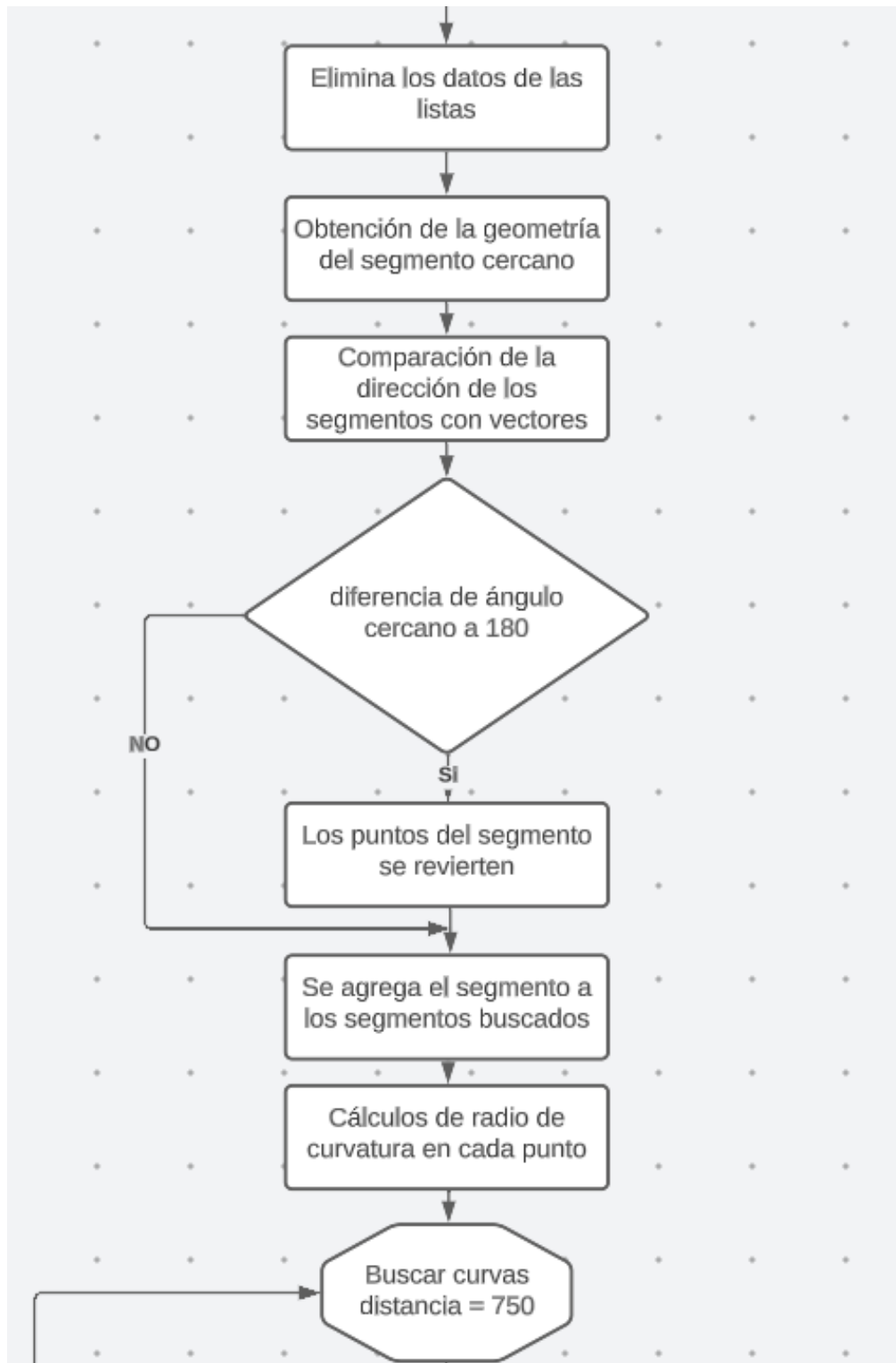


Figura 35: Diagrama de distancia diferente a cero (Segunda parte)

6 COMPROBACION Y RESULTADOS

Para la comprobación del funcionamiento se lo va a realizar por partes, siendo esto una forma de estar seguros de su funcionamiento sin importar las otras partes del código, en una primera instancia se comprueba el funcionamiento del gráfico que muestra la velocidad y las velocidades máximas de cada una de las vías.

En este caso los cambios de las velocidades máximas dependen de la vía en la que circulan, teniendo varios tipos de velocidades máximas, siendo estas máximas velocidades de 30, 50 y 90 Km/h, de igual forma el rango Amarillo son 5 km/h menos de las velocidades máximas que avisa al conductor cuando este se encuentra cerca de superar el límite de velocidad dada por la vía. Estos cambios en la velocidad y la actualización del gráfico se pueden observar en la figura 36.



Figura 36: Diferente rango de velocidad.

Se puede apreciar de igual forma en la figura anterior, que se detecta la vía en la que se circula y ubica a tiempo real el nombre de la vía, también se observa la velocidad del vehículo y dependiendo de la zona que se encuentre se realizan las siguientes acciones:

Tabla 5: Acciones a realizar según la velocidad y la zona que se encuentre el usuario

Acciones por realizar según la velocidad y la zona que se encuentre el usuario	
Velocidad baja o zona verde	No hace ninguna acción
Velocidad cercana al límite o zona amarilla	Aviso sonoro como precaución para no superar el límite de velocidad
Velocidad superior o zona Roja	Aviso sonoro con una velocidad de reproducción equivalente a la velocidad, entre mayor sea la velocidad mayor la velocidad de reproducción

6.1 Comprobación de la búsqueda de curvas

Para esta comprobación se la realiza en la Av. España en donde se recorren para que la búsqueda de curvas sea efectiva, una vez pasado el redondel la aplicación nos da el siguiente aviso:



Figura 37: Distancia para la curva

En donde la distancia a la siguiente curva y nos muestra la geometría de la curva, esta distancia se calcula en base a la geometría de la vía. Para comparar si la geometría corresponde a la curva vista, se sobrepondrá en un mapa. En el cual se observa la similitud que tiene estas, esto se puede verificar en el gráfico presentado a continuación.

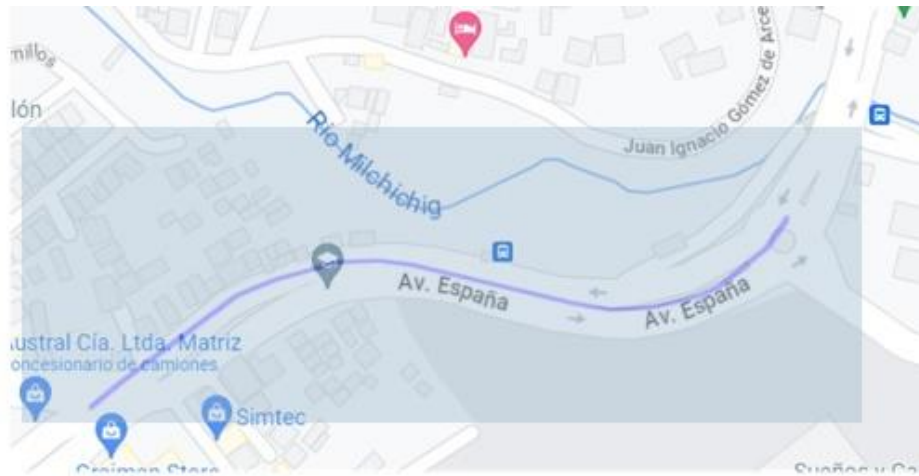


Figura 38: Comprobación de la geometría de curva obtenida de la app en un mapa.

6.2 Comprobación de resultados en la medición de la ubicación y la velocidad

Para esta comprobación se utilizó la aplicación y los datos proporcionados de un freemantics, estos generan el mismo tipo de información pero en frecuencias distintas debido a ciertos factores. En el celular la frecuencia de actualización es de 1 segundo, mientras que en el freemantic es similar a 1 cada milisegundo, el celular realiza estos cambios cada segundo debido a que el consumo de batería si este se incrementa es demasiado alto, entonces sacrifica frecuencia por mantener el dispositivo prendido por más tiempo.

De igual forma los archivos no son similares, el freemantics utiliza archivo .csv mientras que para el proyecto se decidió utilizar archivos xlxs con el objetivo de manejar de mejor manera y más rápida los datos.

La ruta utilizada se muestra en la siguiente ilustración, donde se puede observar los valores obtenidos tanto de la aplicación como de freematics, en esta ilustración se diferencia por color y tipo de gráfica a cada uno de los datos obtenidos, para los datos obtenidos por el celular es de color azul y está representados mediante puntos, mientras para que los datos dados por el Freematics se representan por líneas y de color rojo.

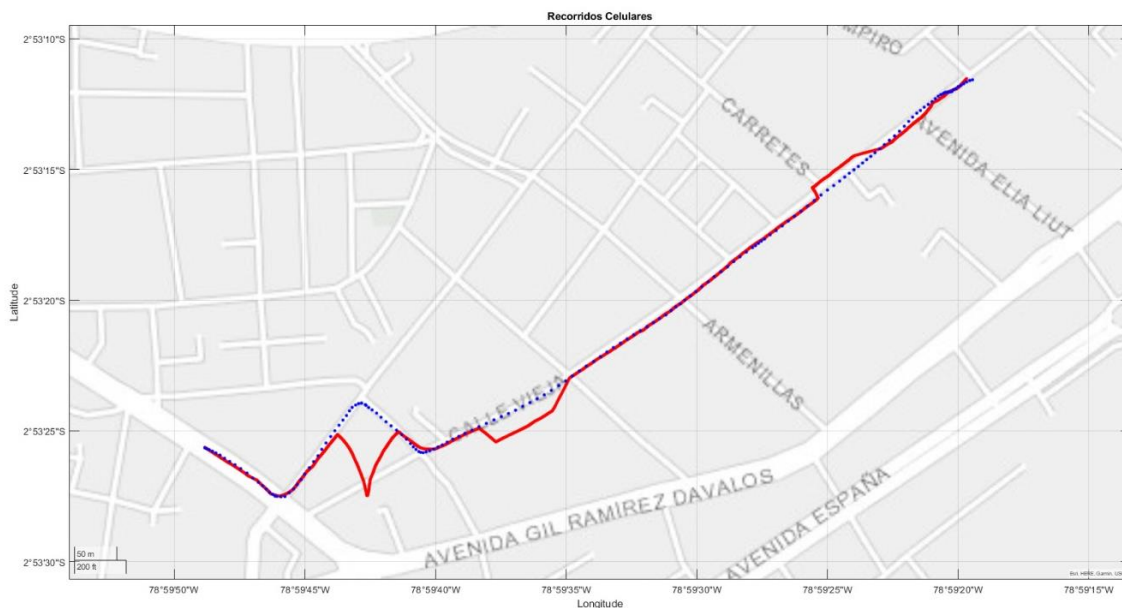


Figura 39: Prueba de ruta con celular y freematics.

Una vez obtenidos los valores, se puede observar que es la misma ruta, pero que hay ciertas partes donde los valores no son tan reales, estos valores pueden ser por las edificaciones que bloquean la señal y hace que se produzcan esos errores.

Queremos comprobar que tan exactos son los datos que se obtienen del celular para ello se quiere realizar de dos maneras, una marea gráfica y una manera de cálculos. Para la manera gráfica se realiza una gráfica de dispersión con la ayuda de Matlab, se obtuvo ciertos problemas por la frecuencia de actualización de los datos y se tuvo que modificar la matriz mediante iteraciones.

6.3 Comprobaciones de la exactitud de los datos de las coordenadas

La gráfica obtenida se puede observar que los datos no tienen muchas variaciones por lo que las mediciones son consideradas correctas. A pesar de eso se ha considerado realizar más cálculos para cerciorarnos que los datos tomados sean los mismos.

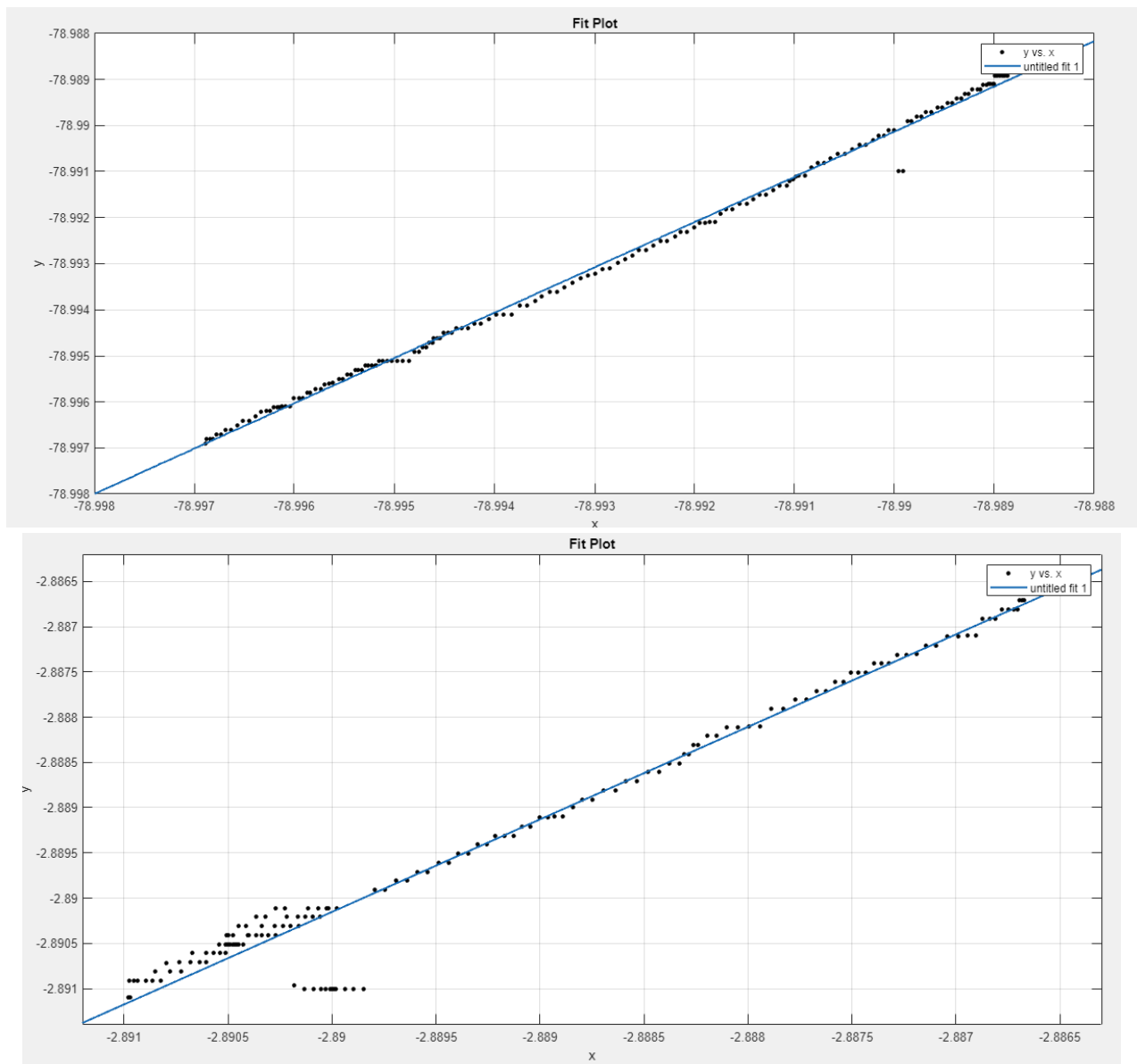


Figura 40: Diagrama de dispersión de los datos de longitud y latitud

Para completar de verificar se decidió realizar dos comprobaciones más, obtener el coeficiente de correlación y el r cuadrado para estar seguros de que los valores sean exactos. Estos valores que se calculó con la ayuda del programa Matlab en donde se obtuvieron estos resultados:

Tabla 6: Coeficientes obtenidos.

Coeficientes obtenidos		
	X	Y
Coeficiente de correlación	0.98544	0.99847
Coeficiente de determinación r^2	0.9711	0.99695

Los valores de coeficiente de correlación nos indican una fuerte correlación entre los valores obtenidos de X y Y del celular con los valores obtenidos del freematics, lo que nos indican que los valores obtenidos del celular están muy cercanos a los que se puede obtener del freematics, a pesar de su tasa de actualización baja.

A su vez los coeficientes de r^2 altos nos indican que la exactitud de la medición de los datos del GPS del celular es muy alta, por lo que no hay mucha diferencia entre utilizar los datos proporcionados por el celular o el freematics.

6.4 Comprobación de los datos de velocidad

De igual forma que el caso anterior para comparar la exactitud de los datos se seguirá la misma metodología, en donde se utilizará el grafico de dispersión para conocer qué tan exactos son las mediciones que se obtiene.

En la figura 41 de los valores de la misma prueba de ruta, con respecto a la velocidad se puede observar que la velocidad obtenida del celular, gráfico de color rojo, tiene cierto retraso debido a que generalmente la velocidad se actualiza cada segundo, por temas de bajo consumo de batería, así que, en cambios muy bruscos de velocidad, le cuesta al GPS del celular de tomar la velocidad de forma instantánea mientras que, para cambios no tan bruscos, este cambio no es perceptible.

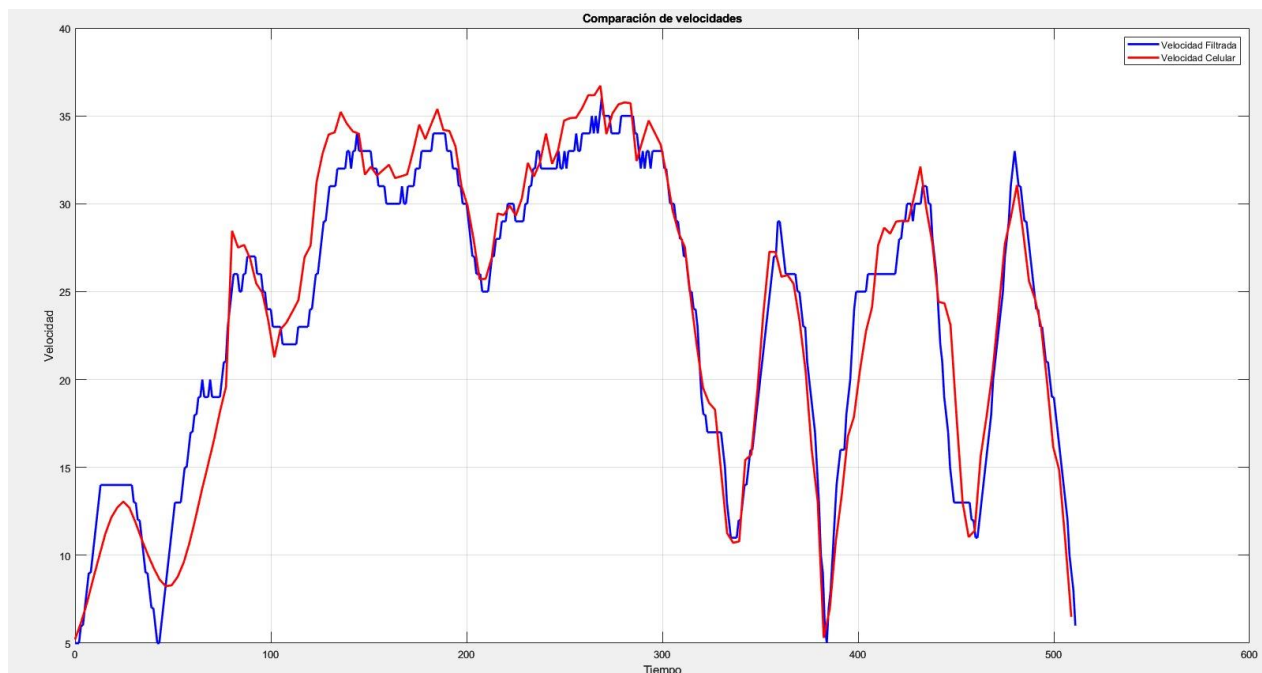


Figura 41: Representación gráfica obtenida por el Freematics y Celular.

Sin embargo, con este gráfico no se puede emitir un juicio muy acertado, para ello se utilizará el diagrama de dispersión para observar que tan dispersos son los datos que se obtuvieron, la ilustración siguiente es el gráfico mencionado en donde se observan cómo se comportan las dos mediciones, aquí se ve mucho más el retraso de la mediciones de la velocidad, sin embargo, en muchos casos la velocidad se cruzan por lo que crea un efecto de que los datos no están tan dispersos, aunque esto no es suficiente para que la comprobación sea correcta.

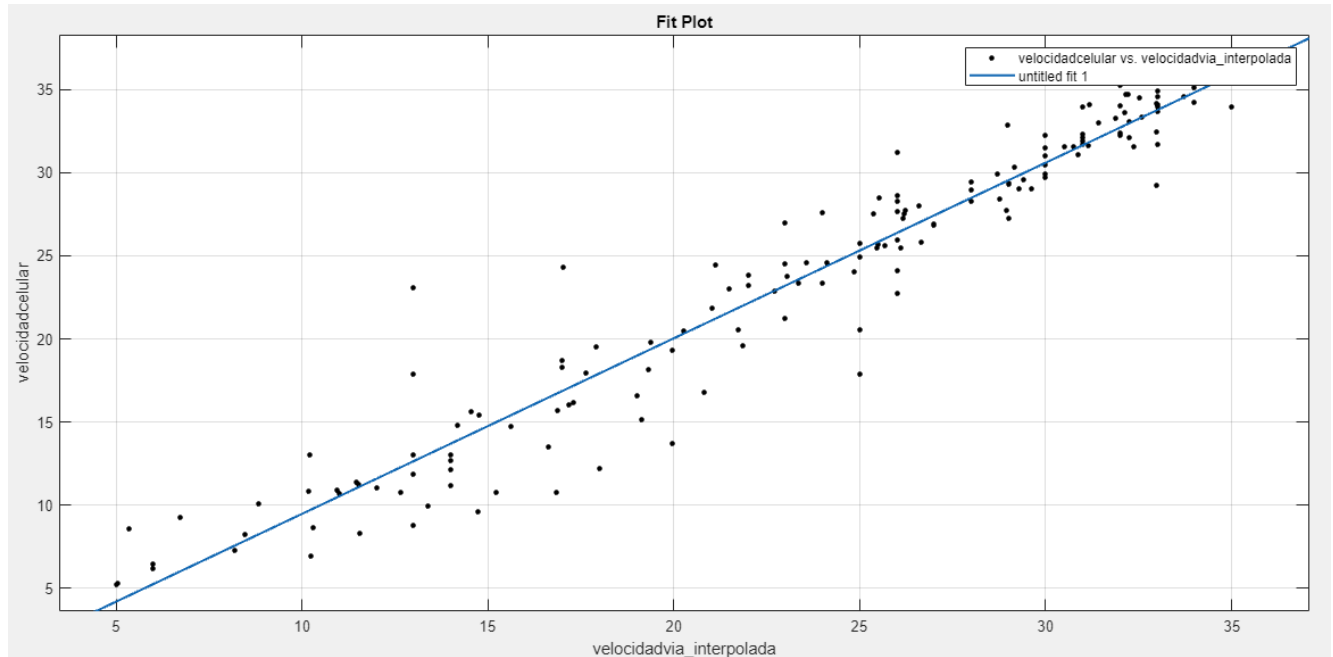


Figura 42: Diagrama de dispersión obtenido de freematics y celular.

Es por eso que se decidió de igual forma calcular los coeficientes de determinación y correlación para observar y concluir que sucede con los datos obtenidos de la velocidad, si estos son certeros para su utilización.

Tabla 7: Coeficientes para los datos de velocidad.

Coeficientes para los datos de velocidad	
Coeficiente de correlación	0.96878
Coeficiente de determinación R²	0.9385

Para el caso de la velocidad se observan valores cercanos al 1 lo que indican que los datos del celular se pueden trabajar sin problemas, cabe mencionar que los datos de velocidad en estos coeficientes son bajos en comparación al anterior debido al retraso que se obtiene, por lo que a pesar de ser bastantes cercanos quizás para casos puntuales no puede llegar a ser la mejor opción.

7 CONCLUSIONES

Para el desarrollo del proyecto se tuvo que tener en cuenta muchos aspectos que podían afectar al funcionamiento del mismo, los datos que se utilizaron de las vías significó un gran desafío debido a que su adquisición por parte de empresas creadoras de mapas generalmente son de pago, a pesar de eso existen alternativas que para desarrollar aplicación con un fin educativo es de mucha ayuda, así mismo el manejar estos datos fue otro grande problemas, debido a que las vías no son una reconstrucción total, si no que se encuentran en forma de segmentos, donde cada segmento tiene diferente longitud de datos y la cantidad de información obtenida puede ser tanta que descargar el sistema de mapas únicamente del ecuador y manejar en un teléfono celular demanda una gran cantidad de espacio en la memoria RAM , donde en equipos de baja gama el celular se vuelve demasiado lento.

Así mismo trabajar con segmentos significó verificar la dirección de estos para verificar si siguen el movimiento del vehículo para realizar una reconstrucción de la vía en busca de curvas.

La utilización del GPS del celular fue crucial para la obtención de datos no solo de la ubicación en general, sino también de otra información mediante métodos propios de la aplicación de desarrollo para minimizar los gastos en líneas de código para obtener la misma información como lo es para obtener la velocidad, y la dirección del movimiento. A partir de ello la utilización de estos datos para buscar la vía más cercana mediante algoritmos, que identifique cada caso que pueda suceder en la búsqueda de curvas, con los datos que se descarga se obtiene la velocidad máxima que constantemente está en monitoreo con el fin de avisar a los conductores cuando este pase de la velocidad o esté cerca de esta.

Se comprobó el funcionamiento de la aplicación tanto como la emisión de la alarma y el cambio de los límites según la carretera, cabe mencionar que debido a que la vía está por segmentos, muchos de los segmentos tienen límites de velocidades diferentes dependiendo de la zona, la más común de reducción de la velocidad máxima es en las zonas de paso peatonal, por lo que también detecta la aplicación.

Para la verificación de los datos que proporciona la aplicación se realizó un análisis de los datos obtenidos por la aplicación y otro dispositivo, obteniendo resultados muy positivos con respecto a la fiabilidad de los datos, estando por encima del 90% en exactitud de los datos que se analizaron obtenidos en una prueba de ruta.

8 RECOMENDACIONES

Para mejorar la aplicación se considera como una forma la automatización del descargo de datos de la zona que se encuentre el usuario y guardarlos por cuadros delimitados según la necesidad, entre más pequeños sean estos cuadros más rápidos serán los cálculos y el celular no estará cargado de tanta información.

De igual forma una de las recomendaciones es la optimización de ciertos llamados a variables a utilizar, así como la búsqueda de otros dispositivos que sean especializados para el propósito de utilización como un navegador, con esto se logra la reducción del consumo de batería del celular y la dependencia de esta, el aumento de la tasa de refresco de los datos y por consiguiente elaborar un entorno mucho más amigable con el usuario.

A pesar de que los teléfonos han evolucionado y tiene la misma capacidad computacional que una computadora, se debe pensar en los dispositivos con menor capacidad, ya que estos dependen de una manera muy fuerte la velocidad de los cálculos y en muchos casos estos cierran la aplicación por la utilización de la RAM.

9 REFERENCIAS BIBLIOGRAFICAS

Real Academia Española. (s.f.). Smartphone. En Diccionario de la lengua española. Recuperado en 26 de diciembre de 2022, de <https://dpej.rae.es/lema/smartphone>

De Ingeniería, C., Automotriz, M., Gustavo, J., Tacuri, C., Alejandro, M., Ordoñez, T., Néstor, I., Rivera, D., & Campoverde, MS (s/f). *UNIVERSIDAD POLITÉCNICA SALESIANA SEDE CUENCA*. Edu.ec. Recuperado el 18 de enero de 2023, de <https://dspace.ups.edu.ec/bitstream/123456789/20425/1/UPS-CT009168.pdf>

León Calle, JD, & Yunga Pañi, HI (pendiente T. (2022). Diseño de una APP para sistema operativo Android, para el monitoreo en tiempo real y rastreo de un automóvil.

Rojas, G., & Darío, O. (2021). *Desarrollo de una aplicación móvil para el rastreo de ubicación y comando remoto vehicular*.

A. John y PR Nishanth, "Sistema integrado en tiempo real para la prevención de accidentes", Conferencia internacional de tecnología electrónica, comunicación y aeroespacial (ICECA) de 2017, Coimbatore, India, 2017, págs. 645-648, doi: 10.1109/ICECA.2017.8212745.

CHOFI. (2019). *Te conduce seguro*. Chofi. <https://chofi.net/>

COdriver. (2020). *COdriver*. <https://www.codriver.city/>

CRUCE. (2016). Rescate vial rápido y fiable «Kenny». CRUCE. <https://cruce.iteso.mx/rescate-vial-rapido-fiable-kenny/>

Adarsh Fernando. (2022). *Android Studio Bumblebee*. <https://android-developers.googleblog.com/2022/01/android-studio-bumblebee-202111-stable.html>

Asmal, M., & Suárez, J. (2013). *Estudio del uso de dispositivos móviles, diseño e implementación de una aplicación que permita interactuar con los ambientes virtuales de aprendizaje de la Universidad Politécnica Salesiana*. <http://dspace.ups.edu.ec/handle/123456789/6305>

L. Google, «Google Maps - Navegación y transporte público - Aplicaciones en Google Play», feb.12,2021/<https://play.google.com/store/apps/details?id=com.google.android.apps.maps&hl=es&gl=US>

10 ANEXOS

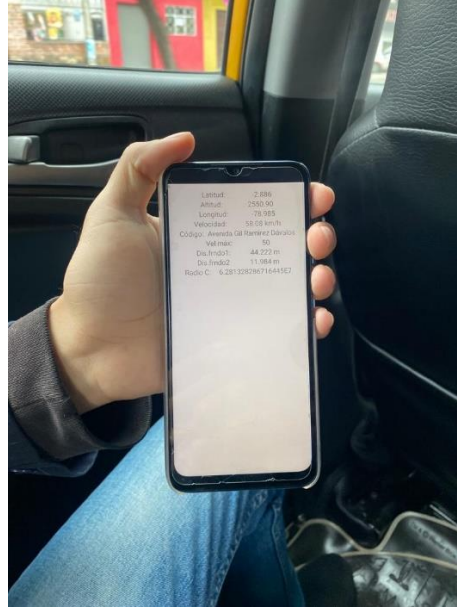


Figura A- 1 Primeras Pruebas de Ruta.



Figura A- 2: Correcciones en la APP en la vía.



Figura A- 3 Colocación del Freematics.



Figura A- 4 Correcto funcionamiento de la App

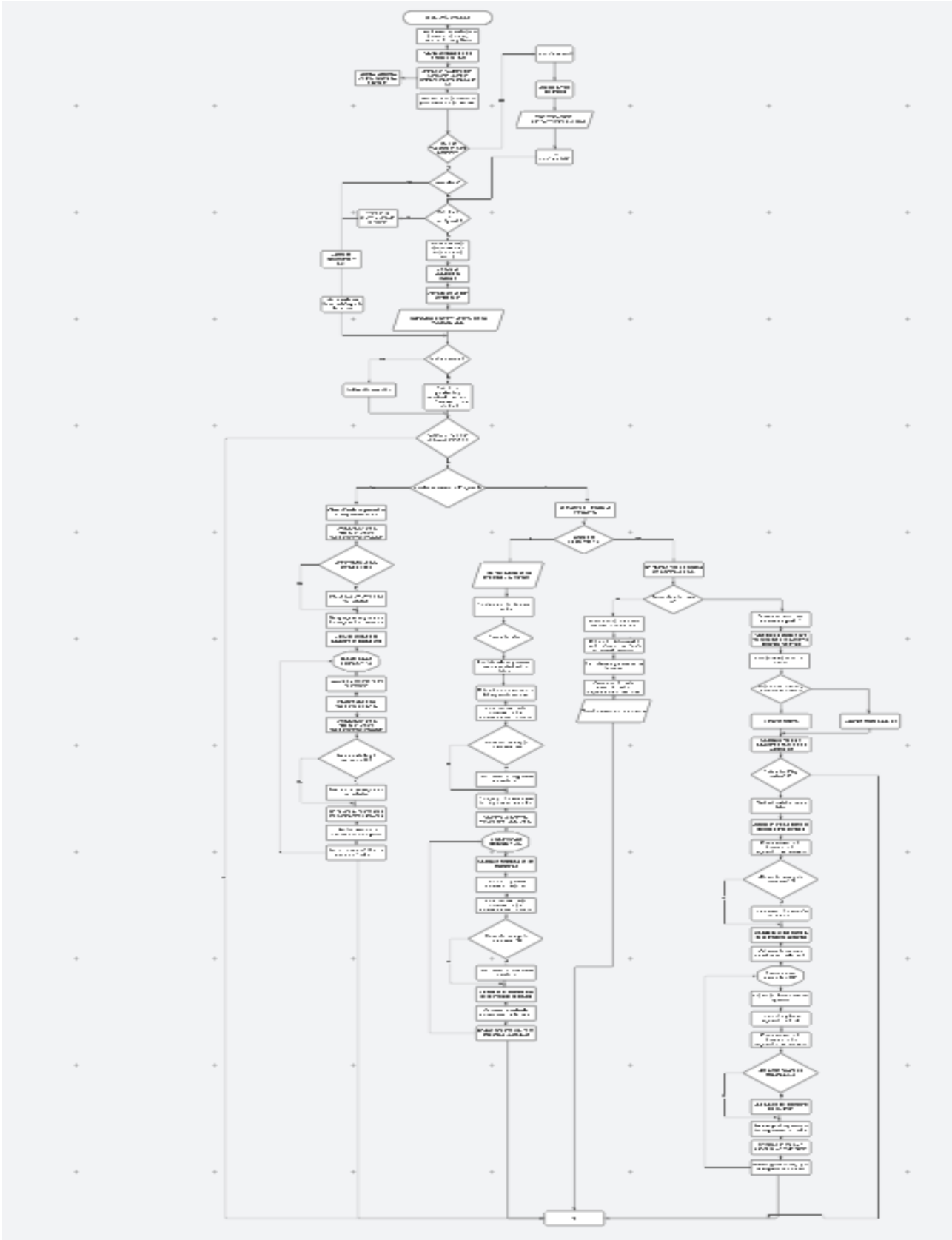


Figura A- 5: Diagrama de flujo de toda la Aplicación Móvil.