

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO ADAS CON COMUNICACIÓN IOT
CON RADAR LIDAR PARA DETECCIÓN DE PSEUDOACCIDENTES.

SANTIAGO ANDRÉS MONTOYA FLÓREZ
JEFERSSON NIÑO SILVA

UNIVERSIDAD SANTO TOMÁS

INGENIERÍA ELECTRÓNICA

TUNJA

2020

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO ADAS CON COMUNICACIÓN
IOT CON RADAR LIDAR PARA DETECCIÓN DE PSEUDOACCIDENTES.

SANTIAGO ANDRÉS MONTOYA FLÓREZ
JEFERSSON NIÑO SILVA

TRABAJO PARA OPTAR POR EL TITULO DE INGENIERO ELECTRÓNICO

DIRECTOR
CESAR MAURICIO GALARZA BOGOTÁ

CODIRECTOR
LUIS FERNANDO CASTELLANOS GUARIN

UNIVERSIDAD SANTO TOMÁS

INGENIERÍA ELECTRÓNICA

TUNJA

2020

Las ideas y afirmaciones presentes en este documento son responsabilidad de los autores. La universidad Santo Tomas no se hace responsable de las consecuencias que pueda generar este trabajo.

NOTA DE ACEPTACIÓN

La tesis de grado “Diseño e implementación de un prototipo ADAS con comunicación IOT con radar LIDAR para detección de pseudoaccidentes.” desarrollada por Santiago Andrés Montoya Flórez y Jefersson Niño Silva, estudiantes de la Universidad Santo Tomás seccional Tunja, ha sido aprobada para obtener el título de ingeniero electrónico.

CESAR MAURICIO GALARZA BOGOTÁ

Director del proyecto

LUIS FERNANDO CASTELLANOS GUARIN

Codirector del proyecto

WILLIAM FABIÁN CHAPARRO BECERRA

Lector

ANGÉLICA MARÍA SALAZAR

Lector

TABLA DE CONTENIDO

LISTA DE FIGURAS	6
LISTA DE TABLAS	7
DEDICATORIA	8
RESUMEN.....	9
INTRODUCCIÓN.....	10
1. PLANTEAMIENTO DEL PROBLEMA.....	12
1.1. Formulación de Preguntas.....	12
1.2. Definición del problema.....	12
1.3. Delimitación del problema.....	14
2. JUSTIFICACIÓN.....	15
3. OBJETIVOS.....	17
3.1. Objetivo General.....	17
3.2. Objetivos Específicos.....	17
4. MARCO TEÓRICO.....	18
4.1. Teórico.....	18
4.2. Conceptual.....	19
5. DISEÑO METODOLÓGICO	24
6. CONSTRUCCION PROTOTIPO ADAS.....	27
6.1. SELECCIÓN DE DISPOSITIVOS.....	27
6.1.1. Sensores.....	27
6.1.2. Sistema embebido.....	29
6.2. SELECCIÓN DE TECNOLOGÍA IoT.....	32
6.2.1. Cobertura.....	32
6.3. DISEÑO DEL PROTOTIPO.....	35
6.3.1. Programación.....	36
6.3.2. Interfaz Gráfica.....	49
6.3.3. Módulo 4G.....	52
6.4. IMPLEMENTACIÓN DEL PROTOTIPO.....	52
7. RESULTADOS.....	58
8. CONCLUSIONES.....	72
BIBLIOGRAFÍA.....	74
ANEXOS	77

LISTA DE FIGURAS

Figura 1. Decesos en siniestros viales en Colombia en el año 2019 [3].	12
Figura 2. Vehículos asegurados contra todo riesgo en Colombia para el año 2020 [7].	15
Figura 3. Esquema de fases del proyecto.	25
Figura 4. Mapa cobertura Sigfox [30].	32
Figura 5. Mapa cobertura 2G, 3G y 4G para Claro Colombia (Izquierda) y Movistar (Derecha). [32].	34
Figura 6. Porcentaje de usuarios por empresa de internet móvil [33].	34
Figura 7. Diagrama general del sistema.	35
Figura 8. Diagrama de flujo programa del MPU parte 1.	36
Figura 9. Diagrama de flujo programa del MPU parte 2.	37
Figura 10. Diagrama de conexión mpu a Raspberry.	37
Figura 11. Diagrama de flujo programa GPS parte 1.	38
Figura 12. Diagrama de flujo programa GPS parte 2.	39
Figura 13. Diagrama de conexión GPS a Raspberry.	39
Figura 14. Código de números generados por la configuración del RADAR LIDAR.	43
Figura 15. Diagrama de flujo programa Radar AWR1642 parte 1.	44
Figura 16. Diagrama de flujo programa Radar AWR1642 parte 2.	44
Figura 17. Diagrama de conexión Radar con Raspberry.	45
Figura 18. Diagrama de flujo programa TeraRanger Evo60m parte 1.	46
Figura 19. Diagrama de flujo programa TeraRanger Evo60m parte 2.	47
Figura 20. Rango para los sensores TeraRanger Evo60m.	47
Figura 21. Diagrama de conexión para los sensores TeraRanger Evo60m.	48
Figura 22. Tiles para los datos adquiridos de Raspberry.	49
Figura 23. Interfaz gráfica en Initial State.	50
Figura 24. Fuente del archivo de initial state.	51
Figura 25. Diagrama de flujo subida de datos a la plataforma initial state.	51
Figura 26. Diagrama de conexión modem a Raspberry.	52
Figura 27. Diseño de la caja en acrílico.	53
Figura 28. Distribución del dispositivo en la caja.	53
Figura 29. Prototipo ADAS.	54
Figura 30. Powerbank.	55
Figura 31. Inversor.	56
Figura 32. Inversor conectado al radar LIDAR.	57
Figura 33. Mapa recorrido diurno en la plataforma (Initial State).	59
Figura 34. Toma de datos del radar prueba diurna primer punto de interés.	60
Figura 35. Toma de datos del radar prueba diurna segundo punto de interés.	60
Figura 36. Toma de datos del radar prueba diurna tercer punto de interés.	61
Figura 37. Toma de datos del radar prueba diurna cuarto punto de interés.	61
Figura 38. Ruta y velocidad en un punto específico del recorrido (Initial State).	62
Figura 39. Mapa recorrido nocturno en la plataforma (Initial State).	63
Figura 40. Toma de datos del radar prueba nocturna primer punto de interés.	63
Figura 41. Toma de datos del radar prueba nocturna segundo punto de interés.	64
Figura 42. Toma de datos del radar prueba nocturna tercer punto de interés.	64
Figura 43. Toma de datos del radar prueba nocturna cuarto punto de interés.	64
Figura 44. Ruta y velocidad en un punto específico del recorrido (Initial State).	65
Figura 45. Mapa recorrido neblina en la plataforma (Initial State).	66
Figura 46. Toma de datos del radar prueba neblina primer punto de interés.	66
Figura 47. Toma de datos del radar prueba neblina segundo punto de interés.	67
Figura 48. Toma de datos del radar prueba neblina tercer punto de interés.	67
Figura 49. Ningún objeto cercano detectado. (los datos en coordenada (6.5, -2) pertenecen a un árbol ubicado en el rango de medición del radar)	68
Figura 50. Objeto cercano detectado.	68
Figura 51. Prueba a través de la ventana.	69
Figura 52. Prototipo ubicado en la parte inferior del carro.	70
Figura 53. Prueba Auto.	70

LISTA DE TABLAS

Tabla 1. Autos más comercializados en Colombia [5].	13
Tabla 2. Análisis de precio y tiempo de entrega del MPU6050 en diferentes mercados nacionales e internacionales [24] [25] [26].	28
Tabla 3. Análisis de precio y tiempo de entrega del Módulo GPS NEO-6M V2 en diferentes mercados nacionales e internacionales [24] [25] [26].	28
Tabla 4. Configuraciones de overclock para Raspberry [29].	30
Tabla 5. Análisis de precio y tiempo de entrega de la Raspberry pi4 en diferentes mercados nacionales e internacionales [24] [25] [26].	31
Tabla 6. Análisis de precio y tiempo de entrega del kit de sobremesa de Raspberry pi4 en diferentes mercados nacionales e internacionales [24] [25] [26].	31
Tabla 7. Descripción del formato GPRMC [35].	40
Tabla 8. Equivalente grados letra a grados.	41
Tabla 9. Análisis comparativo de pruebas	71

DEDICATORIA

Este proyecto se lo dedicamos con amor a las familias Montoya Flórez y Silva, quienes se esforzaron para formarnos con sabiduría y ayudarnos a llegar a este importante punto de la vida en la que nos encontramos hoy, por trabajar duro y a pesar del cansancio siempre ofrecernos una sonrisa de corazón.

La ayuda que nos brindaron, es el carácter con el que asumiremos nuestra vida de aquí en adelante. Muchas gracias.

Santiago Andrés Montoya Flórez.

Jefersson Niño Silva.

RESUMEN

En el presente documento, se describen las diferentes etapas abordadas en el proyecto, diseño e implementación de un prototipo ADAS con comunicación IoT con radar LIDAR para detección de Pseudoaccidentes; basado en 6 fases que fundamentan la propuesta para el desarrollo del prototipo de asistencia al conductor a fin de reducir la probabilidad de sufrir un accidente.

Este prototipo consta de un sistema GPS para obtener la ubicación en tiempo real, un sensor MPU6050 del cual se obtiene la aceleración y posición de los ejes X, Y y Z, un radar tipo LIDAR de referencia AWR1642 y tres sensores de distancia Evo60m, esta información es adquirida por una Raspberry pi 4, la cual se encarga de procesar los datos, para finalmente realizar una comunicación con una app IoT mediante un módulo 4G.

Según los datos obtenidos por el radar se podría deducir que la funcionalidad de este trae consigo ventajas a pesar de la corta distancia con la que este trabaja; distancia variable de 10 a 15m. Así mismo se precisa que las rutas y demás datos enviados a la plataforma IoT concuerdan con los adquiridos en el protocolo de pruebas implementadas en dos automóviles marca Nissan y Kia.

INTRODUCCIÓN

Teniendo en cuenta la evolución de los diferentes campos tecnológicos, industriales, robóticos, domótica, se evidencia la necesidad fortalecer e innovar con mecanismos en pro de la integridad humana y la seguridad informática en el campo automovilístico, la gran cantidad de accidentes que ocurren a diario en las diferentes ciudades del país, han generado la necesidad de crear programas basados en sistemas ADAS, que permitan al conductor anticipar situaciones probables de accidentalidad [1].

Sin embargo; la problemática puede radicar en la distribución de estos sistemas ADAS, puesto que se encuentran diseñados para automóviles de gama alta, como lo es BMW, Porsche, Aston Martin, entre otras, mientras que automóviles de gama baja, o con varios años de trayectoria, es posible que no cuenten con estos dispositivos [2].

Desde esta perspectiva se propuso diseñar e implementar un prototipo ADAS con comunicación IoT con radar LIDAR para detección de pseudoaccidentes, de cualquier tipo de vehículo y así brindar a los conductores seguridad y reducción de los índices de accidentalidad en la ciudad de Tunja.

El diseño y construcción del prototipo se explica en ocho capítulos desde el planteamiento del problema hasta terminar con las conclusiones del proyecto, en estos se abordan todos los procesos que dan a conocer los aspectos importantes para comprender el trasfondo y la ejecución del proyecto.

En el primer capítulo, se encuentra el planteamiento del problema donde se definen los factores de la posible situación a abordar, así mismo se realiza la formulación de preguntas problémicas para estructurar el proyecto, delimitando el problema a resolver. El segundo capítulo presenta la necesidad del porque fundamentar el sistema en las diferentes marcas y modelos de automóviles existentes en el mercado.

El capítulo tercero, presenta el paso a paso para formular la finalidad del proyecto, generando las bases para la construcción del cuarto capítulo donde se sustentan teorías y conceptos de proyectos similares que permiten la viabilidad del mismo.

El diseño metodológico, corresponde al quinto capítulo donde sustenta las fases del sistema a implementar, dando apertura al capítulo sexto, que estructura el proyecto, explicando cada una de las fases mencionadas en el capítulo anterior.

Los resultados obtenidos en las pruebas realizadas se evidencian en el séptimo capítulo para finalmente obtener conclusiones.

1. PLANTEAMIENTO DEL PROBLEMA.

1.1. Formulación de Preguntas.

- ¿Qué ventajas y desventajas tiene la utilización de un radar LIDAR para la detección de objetivos y obstáculos a diferencia de una cámara en un sistema ADAS?
- ¿Qué ventajas tiene usar tecnología de comunicación 4G/3G/2G, en lugar de otras tecnologías de IoT como Sigfox, lora o xbee?

1.2. Definición del problema.

Según cifras arrojadas por el último boletín estadístico sobre muertes, lesiones y accidentes de tránsito en Colombia publicado por la agencia nacional de seguridad vial (ANSV) y el observatorio nacional de seguridad vial (ONSV), entre enero del año 2019 y diciembre del mismo año, se presentaron 6826 muertes; de los cuales 905 fueron usuarios de vehículos, esto equivale al 13.25% del total de decesos. [3]

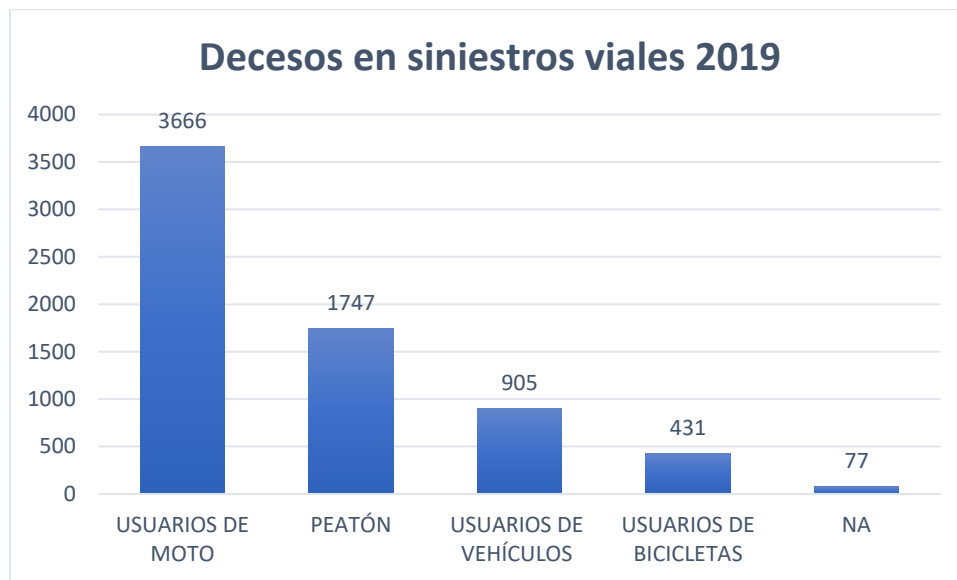


Figura 1. Decesos en siniestros viales en Colombia en el año 2019 [3].

Aunque estas cifras son mayores en un 2.44% a las presentadas en el año 2018, siguen siendo unas cifras muy altas, según el Registro único nacional de tránsito (RUNT), 46% de los vehículos registrados no tienen Seguro Obligatorio de Accidentes de Tránsito (SOAT) [4],

por otro lado, el 56% de los vehículos que deberían poseer una revisión técnico-mecánica al parecer no cuentan con el certificado vigente, por lo cual es muy factible que suceda una falla mecánica en medio de la vía.

En Colombia la compra de automóviles ha tenido un alza hasta junio de 2019 del 2.4% con respecto al mismo periodo del año anterior, entre estas cifras se encuentran los autos más comerciales del país, estos automóviles presentan puntuaciones bajas en el NCAP para Latinoamérica. [5]

TOP 10 DE AUTOS VENDIDOS POR LINEA		VENTA POR MES		ACUMULADO DE VENTAS		CALIFICACION NCAP (0 A 5)	
RANKING 2019	AÑO	jun-18	jun-19	2018	2019	Conductor	Pasajero
	LINEA						
1	RENAULT SANDERO	889	975	5416	5219	1	2
2	KIA PICANTO	829	854	5171	5143	1	0
3	RENAULT LOGAN	1028	846	5254	4544	1	3
4	CHEVROLET SPARK	916	517	7255	4384	0	2
5	RENAULT STEPWAY	689	753	4393	4204	NA	NA
6	RENAULT DUSTER	555	762	3820	4199	4	2
7	MAZDA 2	363	677	2513	3946	2	3
8	CHEVROLET BEAT	410	637	2002	3501	NA	NA
9	CHEVROLET ONIX	440	507	2077	3446	0/3	3/2
10	RENAULT KWID	0	747	1	2893	3	3

Tabla 1. Autos más comercializados en Colombia [5].

Según la información proporcionada anteriormente; se puede observar que en caso de presentarse un accidente grave, la probabilidad que el conductor y los pasajeros sobrevivan son mínimas; teniendo en cuenta la calificación obtenida en seguridad por el conductor y los pasajeros la cual se encuentra en la última columna de la tabla anterior, de esta manera se propone el diseño e implementación de un prototipo ADAS con comunicación IoT con Radar LIDAR, el cual analizará los comportamientos de los conductores en las vías mediante radar y al detectar un comportamiento inusual en los diferentes vehículos, altas velocidades, violación de distancia mínima y movimientos bruscos, generará un mensaje en la interfaz gráfica para que el usuario pueda minimizar el riesgo de colisión.

1.3. Delimitación del problema.

- Delimitación Geográfica: La implementación del sistema ADAS se realizará en la ciudad de Tunja por facilidad de desplazamiento y adquisición de materiales.
- Delimitación Temporal: La propuesta tendrá un periodo de durabilidad de 9 meses aproximadamente, proyectándose a inicios del mes de febrero y culminando en el mes de octubre del presente año.
- Delimitación Temática: La finalidad que tiene la realización de este proyecto, es la implementación de un prototipo ADAS, considerando factores a tener en cuenta para su correcto funcionamiento como lo son: La detección temprana de objetos, la aceleración y ubicación del vehículo automotor.

2. JUSTIFICACIÓN.

La accidentalidad en Colombia ha sido fuente de debates por las diferentes autoridades en materia de automovilismo y viabilidad, donde el incremento de los decesos por siniestros en automóviles ha causado que las mismas exijan programas y herramientas de seguridad para los conductores y pasajeros [6]. Desde esta perspectiva las aseguradoras a nivel nacional fomentan la necesidad de asegurar los diferentes vehículos automotores y así contar con un respaldo económico y legal al momento de sufrir cualquier siniestro, en la (figura 2), se establece la cantidad de autos asegurados en el último año en Colombia. Por lo anterior se observa la necesidad de implementar un prototipo para mitigar la tasa de accidentalidad y de esta manera reducir las pérdidas económicas para los diferentes sectores involucrados; puesto que un accidente de tránsito puede conllevar a una pérdida total o parcial del automotor, e implicaciones legales y monetarias que involucran el deceso o las posibles lesiones físicas a los involucrados.

porcentaje de vehiculos asegurados en colombia
para el año 2020

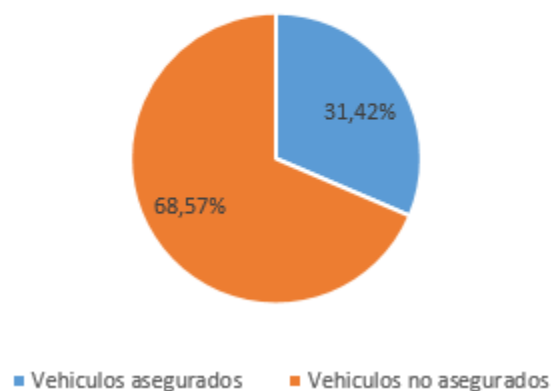


Figura 2. Vehículos asegurados contra todo riesgo en Colombia para el año 2020 [7].

El porcentaje de vehículos asegurados en Colombia, pertenece al 31,42%, de los cuales al presentarse un accidente se consideran pérdida parcial cuando el costo de reparación del auto siniestrado es menor a 75% de su valor asegurado, mientras que la pérdida total excede el

75% del valor asegurado. En Colombia el SOAT es el seguro obligatorio para todos los vehículos automotrices; sin embargo, 31,42% de los propietarios de automotores adquieren seguros complementarios al SOAT con otro tipo de póliza parcial o todo riesgo que protege el bien material o daños que se puedan generar a terceros [7].

Por ende, ante un posible siniestro automovilístico, la pérdida económica de una familia de estrato medio o bajo, que no cuente con un seguro para su vehículo, puede generar daños económicos irreparables, así como la posible pérdida de los ingresos generados para su sustento en caso que el vehículo sea de servicio público o de transporte de carga [8].

El nivel de contaminación del prototipo ADAS es mínimo puesto que el sistema está diseñado para no generar ninguna concentración de smog en el vehículo automotor, así mismo está programado para minimizar la contaminación al presentarse un vertimiento en un río o laguna donde pueda causar daños al ecosistema local, por esto el sistema no contribuye a aumentar el impacto ambiental.

3. OBJETIVOS.

3.1. Objetivo General.

- Diseñar e implementar un hardware para un prototipo de sistema de asistencia al conductor (ADAS), teniendo en cuenta variables de aceleración, detección de objetos y ubicación del automotor, asociado con un sistema de comunicación IoT, con el fin de mitigar la probabilidad de tener un accidente.

3.2. Objetivos Específicos

- Analizar entre los distintos dispositivos de sistemas embebidos y tecnologías IoT del mercado y elegir el sistema óptimo para suplir la necesidad de rango cubierto, señal y procesamiento de datos.
- Diseñar un prototipo ADAS con un sistema embebido, sensores (Inercial, GPS, LIDAR, y módulos IoT para comunicación) y Raspberry teniendo en cuenta que son sistemas multiplataforma.
- Implementar el prototipo ADAS en un vehículo, considerando aspectos como tamaño y características eléctricas a fin de conseguir un proceso efectivo en la toma de datos.
- Analizar e interpretar los resultados del prototipo del sistema ADAS a través de las diferentes pruebas propuestas en el proyecto. .

4. MARCO TEÓRICO.

4.1. Teórico.

La historia de los prototipos Advanced Driver Assistance Systems (ADAS), inicia en el año de 1986 [9] con el trabajo colaborativo de varias empresas e instituciones inmiscuidas en el campo automovilístico, a esta iniciativa se le conoce como proyecto prometheus. Los sistemas ADAS están diseñados con el fin de facilitar y proteger la integridad de los conductores y sus acompañantes, es por eso que, dependiendo su utilidad, cambiará su arquitectura y su comunicación con el usuario como lo son los sensores internos y externos, para que actúen como fuente de información, para prevenir o disminuir la cantidad de accidentes de tránsito provocados por errores humanos.

Algunos de los dispositivos ADAS más comunes son:

- Control de Crucero Adaptativo (ACC).
- Sistema de Advertencia de conducción incorrecta.
- Detector de cambio de carril.
- Sistemas de auto-aparcamiento.
- Control de descenso.
- Haz de luz adaptativo.
- Sistema de anti-colisión (FCW).
- Sistema supervisor del conductor.
- Adaptación de velocidad inteligente.
- Detección y reconocimiento de señal de tránsito.
- Detector de punto ciego.

A partir del 1 de enero de 2017 se hizo obligatorio la implementación de sistemas ABS, Airbag doble en vehículos nuevos y apoyacabezas, así lo determino el ministerio de transporte a través de la resolución 3752 [6], entre otras normativas.

- El Proyecto denominado “Obstacle detection using a 2D LIDAR system for an Autonomous Vehicle” [10], utiliza un sistema LIDAR 2D de bajo costo denominado LIDAR-Lite v1 con un campo de visión de hasta 360 grados donde se logró detectar obstáculos con anchos de 1 metro a distancias de 10 metros.

- Otro proyecto “DepthCN: Vehicle detection using 3D-LIDAR and ConvNet” [11], aborda el problema de la detección de vehículos mediante el uso de datos en redes neuronales de convolución profunda (ConvNet) y 3D-LIDAR con aplicación en sistemas avanzados de asistencia al conductor y conducción autónoma. Donde Se propone un sistema de detección de vehículos basado en los paradigmas de Generación de Hipótesis (HG) y Verificación (HV).

- Un proyecto con alta cercanía a los objetivos de esta investigación sobre ADAS, es “Diseño de un Sistema Avanzado de Asistencia al Conductor (ADAS) basado en Visión Artificial para predicción del Tiempo-de-Colisión (TTC)” [12] , en el que el autor diseño un sistema con filtrado de Gabor y clasificación SVM para caracterización y clasificación de vehículos y Posteriormente, seguimiento espacio-temporal de los resultados obtenidos, este es llevado a cabo mediante Filtrado de Kalman y así capaz de predecir el tiempo restante para producirse un posible impacto entre el vehículo propio y otros presentes en la carretera, a partir de imágenes adquiridas en tiempo real procedentes de una cámara instalada en el interior del vehículo mediante el empleo de técnicas de Visión Artificial.

4.2. Conceptual.

ANSV: Es una Unidad Administrativa Especial de carácter descentralizado, del orden nacional, que forma parte de la Rama Ejecutiva, con personería jurídica, autonomía administrativa, financiera y patrimonio propio, adscrita al Ministerio de Transporte, busca prevenir, reducir y controlar la siniestralidad vial, a través de las acciones administrativas,

educativas y operativas, concientizando a los diversos niveles de la población e integrándola dentro de una cultura vial [13].

RUNT: Es una plataforma tecnológica cuyo fin es registrar, actualizar y validar información del sector tránsito y transporte del país, al ser una plataforma tecnológica reduce al máximo cualquier tipo de alteraciones en los registros y con esto se garantiza un esquema transparente y acorde con la realidad del país lo cual beneficia directamente a todos los ciudadanos. [4]

SOAT: El SOAT es un seguro que se debe adquirir de forma obligatoria establecido por la Ley 100 de 1993, su función es asegurar la atención, de forma eficiente, inmediata e incondicional, de las víctimas de tránsito que sufren lesiones físicas de todo tipo y muerte. [14]

Pseudoaccidente: La palabra Pseudoaccidente está conformada etimológicamente por dos palabras diferentes, las cuales son Pseudo y accidente, la primera es un elemento compositivo, esto quiere decir que es el complemento de otra palabra que por lo general esta inmediatamente seguida, esta palabra viene del origen griego la cual significa falso [15]; La segunda palabra es accidente que viene del origen latín el cual significa que es un suceso no deseado y no planeado, por lo general provoca algún daño o lesión [16], después de haber entendido la etimología de la palabra completa, se puede dar un significado adecuado a el caso de estudio, el cual es un supuesto accidente lo que quiere decir que aún no ha sucedido pero tiene un porcentaje de posibilidad de suceder.

IoT: Las siglas hacen referencia a Internet of Things, lo cual al español se traduce internet de las cosas, este tipo de tecnología se basa en la interconexión digital de los objetos que se usan a diario, como electrodomésticos, u otras cosas de la vida cotidiana, la idea general de la tecnología es crear una red inteligente, esto con el fin de aumentar la pps, que es la protección, productividad y la seguridad; una de las características más importantes y por lo cual la tecnología sigue en auge, es que la información se saca directamente de los usuarios, y el análisis de esta nos permite innovar de acuerdo a las necesidades del usuario y así crecer mutuamente [17].

Para este proyecto de investigación se necesitan varios dispositivos electrónicos como lo son:

Modulo GPS NEO-6M: Este es el sensor GPS que se utilizara, este módulo se basa en u-blox NEO-6M, además cuenta con una memoria eeprom (ROM programable y borrrable eléctricamente), lo cual es ideal, porque esta eeprom es conocida por ser una memoria no volátil, lo que significa que si se llega a desconectar la energía los datos almacenados no serán eliminados y quedaran intactos, con esto se logra que los datos del posicionamiento del vehículo no se pierdan una vez se apague el carro, por causas no normales, tal como un apagón inesperado del carro, entre otros problemas no causados [18].

Otro aspecto importante que se debe ver de este módulo es la antena, las cuales pueden ser pasivas y activas, esto porque el módulo está diseñado para soportar ambos tipos de antenas, la diferencia entre este tipo de antenas recae principalmente en la presencia o ausencia, de dispositivos adicionales como por ejemplo, supresores de ruidos, entre otros dispositivos que ayuden al mejoramiento de la señal, es por eso que el costo de la antena activa es notablemente superior a la de la antena pasiva.

Una de las ventajas al usar este módulo es que tiene diversos protocolos e interfaces de conexión como UART, USB y SPI, esto nos deja con varias posibilidades de conexión, se tiene que saber con qué tipos de conexiones y protocolos cuenta el computador que en este caso es la Raspberry PI 4 [18].

Inercial: Este sensor hace referencia a algunas de las variables que se necesitan para romper el momento de la inercia, como se sabe existen dos tipos diferentes de inercia la mecánica y la térmica, en el caso que nos compete es la inercia mecánica, el módulo inercial escogido cuenta con dos sensores los cuales son acelerómetro y giroscopio, estos nos ayudaran a medir la aceleración y la velocidad angular respectivamente, además hay módulos que adicionalmente cuentan con magnetómetros que nos ayudan a centrar el polo magnético en

este caso hacia el norte, gracias a los sensores nombrados anteriormente es posible estudiar todo lo relacionado con el momento inercial del dispositivo, en este caso el automóvil en que se implementará [19].

Raspberry PI 4: Raspberry PI es un computador demasiado básico, esto porque su tamaño es pequeño comparado con un computador tradicional, pero en potencia no se queda atrás, la Raspberry que se usará en este caso es la PI 4, la cual cuenta como una versión de CPU cortex A-72, este permite la decodificación en 4k y a 60fps [20], en este caso se escogió este dispositivo como centro de todo el proyecto por su velocidad de procesamiento, a diferencia un PIC y un Arduino que son los microcontroladores más usados y populares para este tipo de prácticas, con Raspberry no se va a tener complicaciones de cuellos de botella, puesto que se tienen diversos datos que procesar, además de hacer varias tareas en paralelo como la adquisición, análisis y envío de datos, a la estación base y luego su posterior generación de reporte, además la Raspberry cuenta con diversos protocolos e interfaces como UART, I2C, USB, Bluetooth, SPI entre otros.

Otra gran ventaja es que al ser un computador no se tienen restricciones de lenguaje de programación, esto porque cuenta con una gran gama para escoger como Python 3, entre otros. [20]

Python 3: Este es un lenguaje multi-paradigma, puesto que cuenta con gran cantidad de funciones que no tiene un programa normal como la programación orientada a objetos, programación imperativa y algo de programación funcional; la ventaja que provee Python sobre otros compiladores es la facilidad y la potencia, ya que cuenta con un tipado dinámico, esto es una gran ventaja porque permite un desarrollo rápido de aplicaciones en diferentes áreas de la electrónica, además, su diversidad en las plataformas es bastante. Una característica importante que se debe tener en cuenta es que toda la programación es por línea de comandos, con esto se refiere a que no tiene entorno gráfico y su programación tiene que hacerse línea por línea. [21]

Radar LIDAR: Este es un término compuesto, por lo tanto, se debe saber el significado de ambas palabras para luego crear una representación acorde a las palabras, entonces la primera

palabra es radar, la cual es un acrónimo que se refiere a “Radio Detection And Ranging”, que en el español se traduce como “detección y medición de distancias por radio”; Pero ¿cómo funciona?, “Es un sistema que utiliza ondas electromagnéticas de ciertas frecuencias, para medir distancias, altitudes, direcciones y velocidades de objetos estáticos o móviles como aviones, barcos, vehículos, e incluso variaciones en ciertos terrenos.”[22]; La otra palabra es LIDAR, que también es otro acrónimo, el cual es “Light Detection and Ranging” este dispositivo nos permite identificar un objeto u obstáculo con un emisor láser y la tecnología funciona de acuerdo al tiempo de retraso entre que la luz o rayo fue emitido, hasta que retorna.

El radar LIDAR funciona como un capturador, procesador e interpretador de datos, y paralelamente con estos, crea una información contextual con el entorno; La característica principal es que todo esto funciona en tiempo real entonces es ideal para el funcionamiento en automóviles y para el proyecto de pseudoaccidentes.

5. DISEÑO METODOLÓGICO.

Este trabajo cuenta con los recursos asignados al proyecto Sistema de alertas tempranas para minimizar colisiones por violación de distancia prudencial mínima en accidentes. Dentro de la convocatoria de proyectos de investigación USTA Tunja 2018, este apoyo financiero fue de \$6'000.000 destinados para materiales y suministros, y salidas de campo.

5.1 TIPO DE INVESTIGACIÓN

El desarrollo de esta propuesta se fundamenta en las características de la investigación aplicada, cuya finalidad consiste en buscar la generación de conocimientos con aplicación directa a los problemas de la comunidad o sector productivo; esta investigación se basa en los hallazgos tecnológicos enlazando la teoría o el producto.

Teniendo en cuenta lo anterior, la metodología implementada se fundamentará en 6 fases, las cuales se correlacionan con los objetivos planteados en esta propuesta.

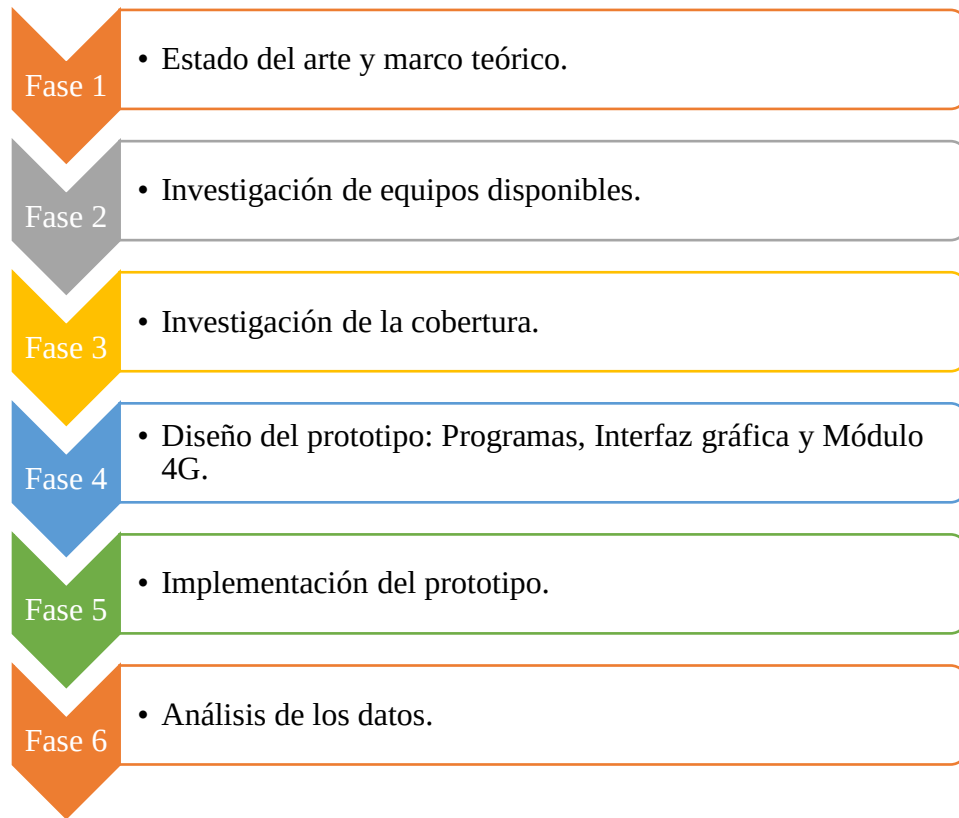


Figura 3. Esquema de fases del proyecto.

Fase 1

- Estado del arte y marco teórico: Se realizará un análisis detallado del funcionamiento de diferentes sistemas ADAS, además de buscar diferentes dispositivos para el desarrollo del trabajo, esto con el fin de encontrar módulos de fácil uso y adquisición.

Fase 2

- Investigación de equipos disponibles: Se analizará los equipos disponibles de sistemas embebidos y sensores con aplicabilidad al comercio latinoamericano, analizando aspectos económicos, accesibilidad y funcionalidad para el desarrollo del prototipo.

Fase 3

- Investigación de la cobertura: Se analizará las diferencias que hay entre la red 4G y los sistemas IoT convencionales como Xbee, Sigfox y Lora. Además de las regulaciones, disponibilidades y legalidad de las Redes IoT en Colombia.

Fase 4

- **Diseño del prototipo:** Esta fase se comprende por 3 subfases nombradas, programación, interfaz gráfica y módulo 4G.
 - **Programación:** Se realizará la programación necesaria en Raspberry pi esto por la capacidad de almacenamiento, la velocidad de procesamiento, entre otros factores, el lenguaje que se utilizará será Python 3, esto con el fin de buscar que se recolecten los datos de la mejor manera a partir de los sensores.
 - **Interfaz Gráfica:** La interfaz gráfica que se realizará será intuitiva y únicamente de consulta ya que los sensores envían los datos de manera automática a la web IoT.
 - **Módulo 4G:** Se realizará las pruebas con un módulo 4G propiedad de los autores para el diseño del prototipo.

Fase 5

- **Implementación del prototipo:** Con el diseño del sistema ADAS, se busca que este integre las unidades embebida, INS, GPS, LIDAR y el módulo IoT; en un vehículo en la ciudad de Tunja.

Fase 6

- **Análisis de los datos:** Con base en los datos obtenidos, se realiza un análisis del prototipo funcionando con los dos sistemas (radar AWR1642 y sensores Evo60m) y así evaluar su efectividad.

6. CONSTRUCCION PROTOTIPO ADAS.

Para iniciar el proyecto, se investiga acerca de los dispositivos ADAS, funcionamiento, los más comunes y los más útiles, correspondiente a la fase 1 del diseño metodológico, donde se observa el funcionamiento de acuerdo a la necesidad del sistema; existen varias clases de sistemas ADAS, destacándose entre ellos: Frenado autónomo de emergencia con detección de peatones y ciclistas, control de cruce adaptativo, mantenimiento de carril, advertencia de punto ciego y sensor de aparcamiento [23], cada uno de estos sistemas tiene una función diferente dependiendo de la parte del automóvil y de la acción a ejecutar.

6.1. SELECCIÓN DE DISPOSITIVOS.

6.1.1. Sensores.

Los sensores cumplen un rol destacado para cada actividad de la industria por sus capacidades de adquisición y transmisión de datos, existen gran variedad de sensores en el mercado, para uso inalámbrico de medición. Para el proyecto se implementaron 4 tipos de sensores, un GPS, un inercial, un sensor de distancia y un radar LIDAR.

A continuación, se explica detalladamente los sensores que se eligieron para cumplir los objetivos del proyecto:

- **Sensor inercial:** La referencia escogida fue MPU 6050, este sensor permite tomar 7 datos al mismo tiempo, ya que cuenta con 3 sensores de aceleración, giroscopio y temperatura; Los primeros 2 tienen componentes en el plano X, Y, Z. Este sensor primordial, ya que permite saber la orientación del carro, si ha tenido un accidente de tránsito como un volcamiento, ya que los datos del giroscopio permitirán analizar los patrones obtenidos por el sensor, finalmente, el tercer sensor para saber la temperatura del prototipo.

En el mercado colombiano e internacional se destacan los siguientes precios:

Tienda	Objeto	Precio	Tiempo de entrega
Ferretrónica	MPU-6050	\$9.950 cop	30min - Medio día
Sigma Electrónica	MPU-6050	\$49.980 cop	1 -2 días hábiles
Digikey	MPU-6050	\$18.855 cop	2-4 semanas

Tabla 2. Análisis de precio y tiempo de entrega del MPU6050 en diferentes mercados nacionales e internacionales [24] [25] [26].

Teniendo en cuenta la disponibilidad y costo del sensor MPU 6050 en el sector comercial local, se opta por comprarlo en *Ferretrónica*.

La adquisición del segundo sensor, GPS, de referencia GPS NEO-6M V2, ya que cuenta con una antena en cerámica que permite acceder a su configuración con facilidad sin requerir adicionales para su funcionamiento fusionándose de manera adecuada con el sensor anterior; pues al tener conocimiento de la trayectoria del automotor es posible comprobar los cambios en los datos en el giroscopio y el acelerómetro, al mismo tiempo, posibilitando la localización del vehículo.

En el mercado colombiano e internacional se encuentran los siguientes precios:

Tienda	Objeto	Precio	Tiempo de entrega
Ferretrónica	Módulo GPS NEO-6M V2	\$38.000 cop	30min - Medio día
Sigma Electrónica	Módulo GPS NEO-6M V2	No hay producto	-----
Digikey	Módulo GPS NEO-6M V2	\$45.216 cop	2-4 semanas

Tabla 3. Análisis de precio y tiempo de entrega del Módulo GPS NEO-6M V2 en diferentes mercados nacionales e internacionales [24] [25] [26].

Se escogió el sensor Módulo GPS NEO-6M V2 de la tienda Ferretrónica, ya que este producto es de fácil acceso en los mercados locales.

La tercera referencia elegida es el radar AWR1642 de Texas Instruments, el cual es propicio para saber la distancia del objeto fácilmente y la detección de objetos es más amplia ya que el radar cuenta con campo de visión de la antena $\pm 60^\circ$ con una resolución angular de aproximadamente 15° , mientras que la cámara con lente estándar es de 25° y 50° [27].

Como complemento al radar se eligen los sensores de distancia TeraRanger Evo60m, de la marca Terabee, este brinda un adecuado funcionamiento y adaptabilidad con los diferentes sensores.

6.1.2. Sistema embebido.

Un sistema embebido, puede catalogarse como un sistema de computación que está diseñado para realizar varias funciones dedicadas, la mayoría de sus componentes se encuentran sobre una placa base, es muy similar a un pc, pero difieren en una parte fundamental y es en sus usos, mientras que el sistema embebido se usa para cubrir una necesidad específica, el pc se usa para cubrir una amplia gama de necesidades, la arquitectura de un sistema embebido es similar a la de una pc, puesto que contiene los siguientes componentes [28]:

1. Microprocesador.
2. Memoria.
3. Caché.
4. Periféricos de entrada y salida.
5. ChipSet.
6. BIOS-ROM.
7. CMOS-RAM.

El sistema embebido que se ha elegido para la propuesta es Raspberry pi 4, por las siguientes razones:

1. Procesamiento: La velocidad de procesamiento de la Raspberry pi 4 es de 700MHz, esta velocidad es alta respecto a los programas que dan funcionamiento total del proyecto, permitiendo hacer overclock al procesador, término que hace relación “sobre el reloj”, el cual modifica la velocidad de procesamiento de componentes para disminuir el tiempo que tarda en realizar una tarea, con esto se puede lograr que cada

componente ejecute más operaciones por segundo aumentando así el rendimiento. Es preciso resaltar que con la utilización del overclock se pueden presentar algunas desventajas entre las que se encuentra el calor, ya que al trabajar los componentes de forma rápida se eleva la temperatura, haciendo que el refrigeramiento sea de forma inmediata y prolongada para mantener la vida útil del componente.

Tipo	Procesador	Núcleo	Memoria RAM	Sobre Tensión
No	700MHz	250MHz	400MHz	0
Modesta	800MHz	250MHz	400MHz	0
Medio	900MHz	250MHz	450MHz	2
Alta	950MHz	250MHz	450MHz	6
Turbo	1000MHz	200MHz	600MHz	6

Tabla 4. Configuraciones de overclock para Raspberry [29].

2. Memoria: Otra de las razones por las que se escogió Raspberry como sistema embebido, es por la capacidad de almacenamiento, puesto que puede ir desde las 4GB hasta las 128GB, esto porque el sistema usa una tarjeta de memoria SD, de las cuales 2.5 GB están destinadas al sistema operativo de la Raspberry (Raspbian que está basado en Linux), este pack de 2.5 GB incluye el escritorio y los softwares recomendados de Raspberry que son para la programación de los siguientes lenguajes:

- JavaScript.
- Java.
- Python.
- C++.

En el mercado colombiano existen diferentes precios y bundles, para la Raspberry pi 4, al realizar el análisis se tiene como referencia diferentes tiendas electrónicas, locales, nacionales e internacionales. El primero a evaluar será la Raspberry pi 4 en su edición sencilla, este únicamente trae la Raspberry, lo que quiere decir sin cargador, sin cable HDMI, entre otros implementos obligatorios para su uso:

Tienda	Objeto	Precio	Tiempo de entrega
Ferretrónica	Raspberry pi 4	\$450.000 cop	30- Medio día
Sigma Electrónica	Raspberry pi 4	\$297.500 cop	1 – 2 días hábiles
Digikey	Raspberry pi 4	\$242.215 cop	4 Semanas

Tabla 5. Análisis de precio y tiempo de entrega de la Raspberry pi4 en diferentes mercados nacionales e internacionales [24] [25] [26].

El kit de sobre mesa, este compuesto por:

- Raspberry Pi 4 Modelo B, 4GB.
- Teclado y mouse Raspberry Pi.
- 2 × micro HDMI a HDMI estándar (Cables de 1 m).
- Fuente de alimentación USB-C.
- Caja oficial de Raspberry Pi 4
- Guía oficial para principiantes de Raspberry Pi (idioma inglés)
- Micro SD de 16GB con Raspbian y NOOBS

Tienda	Objeto	Precio	Tiempo de entrega
Ferretrónica	Kit de sobremesa rps4	No hay producto	-----
Sigma Electrónica	Kit de sobremesa rps4	\$535.500 cop	1 – 2 días hábiles
Digikey	Kit de sobremesa rps4	\$527.961 cop	2- 4 Semanas.

Tabla 6. Análisis de precio y tiempo de entrega del kit de sobremesa de Raspberry pi4 en diferentes mercados nacionales e internacionales [24] [25] [26].

Se adquiere el kit de sobremesa de Raspberry pi 4 de 4GB de memoria RAM de Sigma Electrónica, ya que contaba con los componentes necesarios para el funcionamiento del dispositivo como cargador, HDMI y otros periféricos.

6.2. SELECCIÓN DE TECNOLOGÍA IoT.

En la actualidad, las tecnologías IoT son importantes para el sector industrial, ya que permiten la combinación de diferentes objetos como sensores de varios tamaños para ser utilizados en una misma red, permitiendo recibir y enviar datos para el correcto funcionamiento del objeto en cuestión, para esto se tiene la opción de trabajar con las tecnologías convencionales como Sigfox, Lora y Xbee, o trabajar con otro tipo de tecnologías no convencionales como lo sería redes de internet.

6.2.1. Cobertura.

Uno de los puntos más importantes es la cobertura, aunque el dispositivo está pensado para la ciudad, es importante que también pueda abarcar el territorio nacional para futuras actualizaciones. A continuación, se muestra la cobertura de algunas de las tecnologías para realizar una comparación con las redes 4G.

6.2.1.1. Sigfox

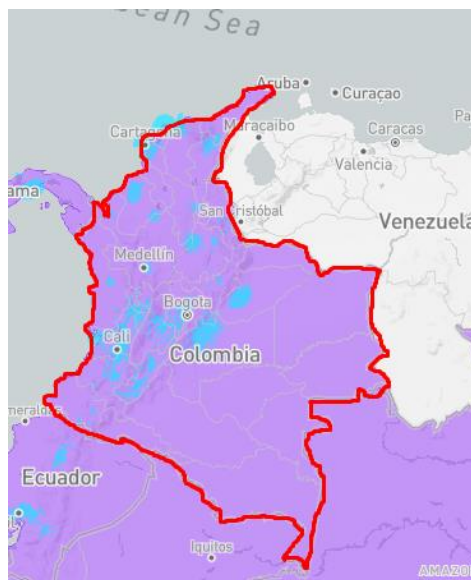


Figura 4. Mapa cobertura Sigfox [30].

En la figura 4, se observa el mapa de cobertura de Sigfox, donde se puede percibir 3 colores diferentes.

El primero es el color blanco que lo porta el vecino país Venezuela, significa que es un país donde no se está trabajando para la implementación de Sigfox.

El segundo color prominente es el violeta, este color representa los países donde se está implementando la tecnología Sigfox, como se visualiza; Colombia, Brasil, Panamá y Ecuador, son países donde se está trabajando en esta tecnología.

Finalmente, el color Cian, donde se tiene cobertura en vivo, o sea activa en algunas zonas del país como Bogotá, Medellín, Cali, Cartagena, Barranquilla, Tunja, Yopal, Valledupar y Santa Marta.

6.2.1.2. Lora:

Esta tecnología funciona de manera diferente a la anterior, ya que requiere de dos módulos, los cuales tienen una distancia máxima de cobertura de 5Km en ciudad y 20 kilómetros en campo abierto con un solo Gateway, también se puede utilizar en una conexión punto a punto sin línea de vista pero esto reduce su distancia máxima a 1 Km, además de necesitar internet en cualquiera de sus dos configuraciones para el correcto funcionamiento, una de las tarjetas más comunes de Lora es la Tarjeta Wireless LORA-02 433MHz V1.0, la cual tiene un costo de \$ 22.610 COP y tiene una distancia de comunicación máxima de 10km, por lo que no es suficiente para el proyecto. [31].

6.2.1.3. 4G:

Se realizó la comparación entre dos empresas que tienen la mayor cobertura en Colombia, *Claro Colombia* y *Movistar*, a continuación, se muestra los mapas correspondientes para ambas empresas

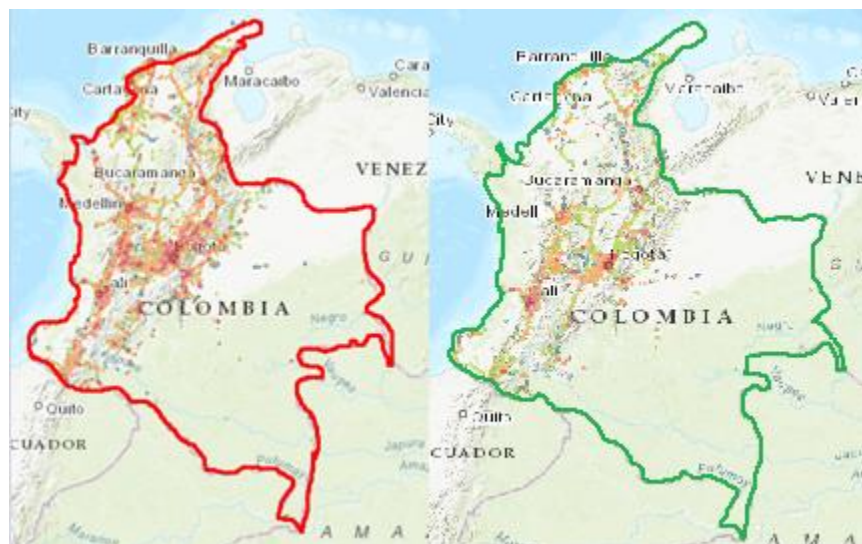


Figura 5. Mapa cobertura 2G, 3G y 4G para *Claro Colombia* (Izquierda) y *Movistar* (Derecha). [32].

En la figura 5 se observa la cobertura 2G, 3G y 4G de *Claro Colombia* y *Movistar*, respectivamente la cobertura 2G se visualiza de color azul, la 3G de color verde, mientras que la 4G se puede visualizar en el mapa de color naranja y rojo, la cual tiene una cobertura mayor en el territorio nacional frente a las tecnologías convencionales Sigfox y Lora. Se toma como referencia la red de *Claro Colombia* porque según datos emitidos por el Ministerio de las TIC, *Claro Colombia* posee más de la mitad de usuarios de internet móvil en Colombia con 57.56% [33] como se muestra en la siguiente gráfica:

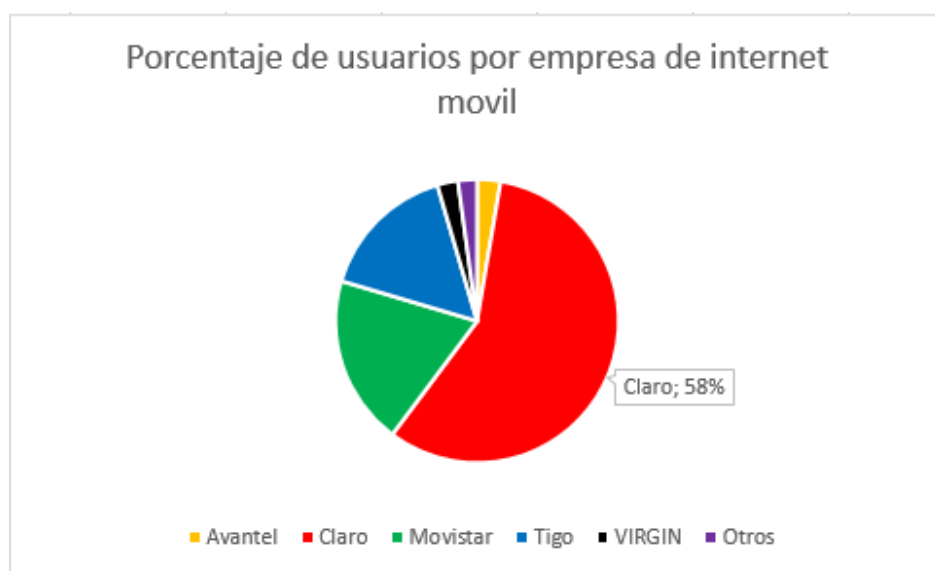


Figura 6. Porcentaje de usuarios por empresa de internet móvil [33].

6.3. DISEÑO DEL PROTOTIPO.

A continuación, se muestra un diagrama general del sistema donde se aclaran todos los componentes del proyecto y la unión de los mismos.

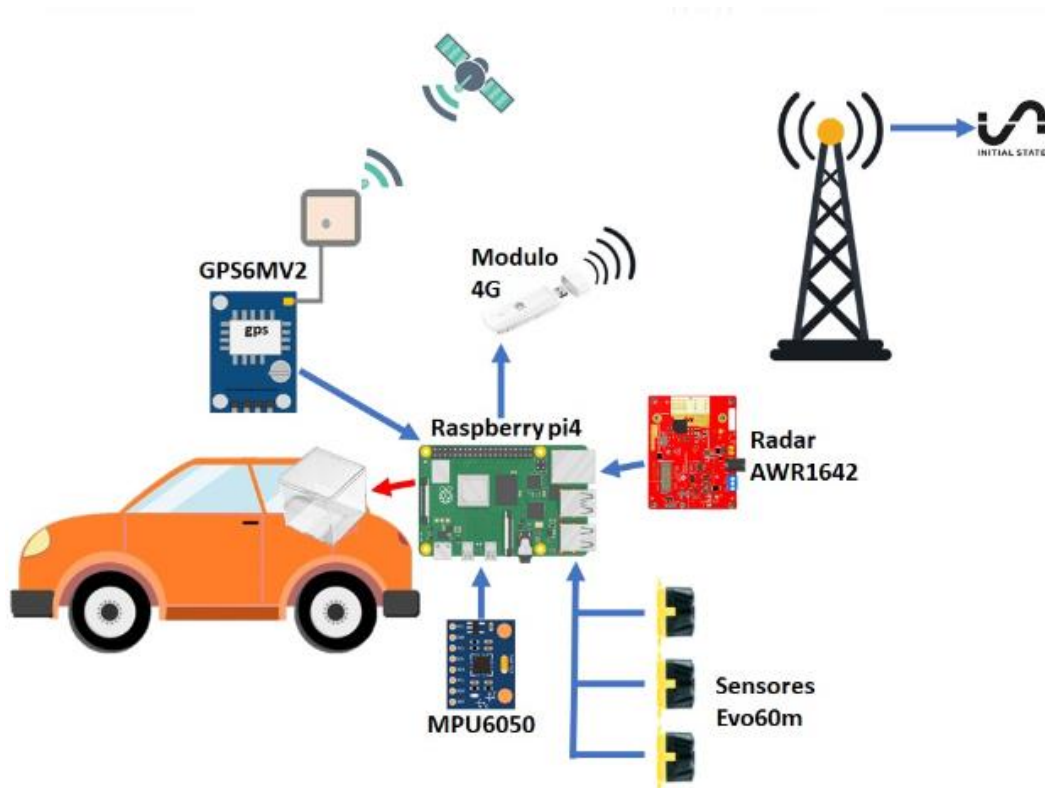


Figura 7. Diagrama general del sistema.

El prototipo funciona de la siguiente forma, el sensor MPU6050 capta las señales de aceleración y giro de los ejes X, Y y Z; el GPS comienza a establecer señal con el satélite y tan pronto lo logra comienza a enviar datos del que se obtienen posición, velocidad y rumbo; El radar AWR1642 entrega coordenadas de puntos; Los tres sensores Evo60m envían información de distancias de cada sensor, esta información es enviada a la Raspberry, la cual se encarga de procesar los datos para posteriormente enviarlos por medio del módulo 4G a la app IoT, la cual al recibir los datos graficará esta información en el Dashboard previamente creado.

6.3.1. Programación.

Inicialmente, se adaptó la Raspberry para el uso correcto de todos los sensores teniendo en cuenta las librerías necesarias para el correcto funcionamiento de estos, la principal función es permitir el envío y recepción de datos, esta librería base tiene el nombre “serial” y en el código es llamada a partir de la siguiente línea o función “import serial”.

Se inició a trabajar en orden con los 4 sensores explicados en la sección 6.1, el primer sensor a trabajar fue el MPU 6050, se dio la necesidad de instalar varias librerías para el uso correcto del dispositivo, la principal librería era la propia del dispositivo llamada con el mismo nombre “MPU6050”, aparte de esta, la librería “time” para controlar los “sleep” o tiempos de espera, librería “String”, esta con el fin de poder trabajar con concatenaciones, caracteres ASCII y principalmente con el formato de cadena personalizado con la que cuenta la librería.

Cabe destacar que el sensor cuenta con un conversor ADC 16 bits, por lo que sus salidas son digitales, permitiendo programar en Python, esto es importante ya que con utilización de librerías es posible la reducción en el costo computacional al momento de procesar información

Estructura del programa:

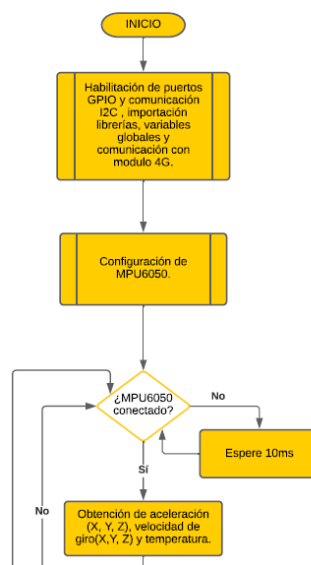


Figura 8. Diagrama de flujo programa del MPU parte 1.

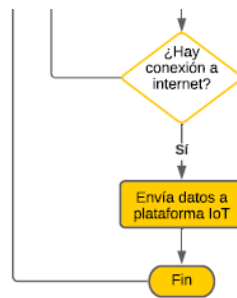


Figura 9. Diagrama de flujo programa del MPU parte 2.

Una vez definido el programa e instaladas las librerías, se procede a realizar la conexión del dispositivo a la Raspberry de la siguiente manera:

Raspberry	MPU 6050
Pin 2 (5V)	Pin 1 (VCC)
Pin 6 (Ground)	Pin 2 (GND)
Pin 5 (GPIO3-SCL)	Pin 3 (SLC)
Pin 3 (GPIO 2 -SDA)	Pin 4 (SDA)

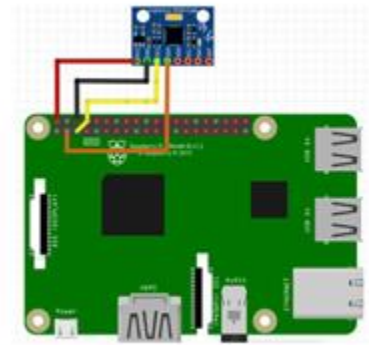


Figura 10. Diagrama de conexión mpu a Raspberry.

Finalizado el proceso de conexiones pertinentes se procedió a hacer la prueba del dispositivo. Los datos que se recibieron fueron graficados en el mismo programa mediante la librería “pyqtgraph” para su análisis e interpretación.

El siguiente sensor con el que se trabajo fue el módulo GPS NEO-6M V2, la librería base para el uso del sensor fue “pynmea2”, la cual hace posible leer e interpretar los protocolos de comunicación NMEA (Es un medio de comunicación donde los GPS receptores pueden comunicarse con los satélites, los datos que llegan están codificados en formato ASCII de 7 bits, y cada línea tiene un formato predefinido, este formato está compuesto de varios campos los cuales están separados de comas, para la identificación de los parámetros individuales

como posición, velocidad entre otros [34]). Cabe destacar que, al transmitir y recibir datos, se usa la librería “serial”.

El tiempo de inicio del sensor cada vez que se activa con la Raspberry es de 38s, mientras que en los intentos posteriores fue de 35s, estos tiempos se registran con la antena cerámica que viene incluida en la compra del GPS, para una mayor precisión y rapidez de captación de los datos, se recomienda comprar la antena activa, además del respectivo cable adaptador RF SMA; con estos dos complementos disminuye el tiempo de adquisición de la antena a 20s por primera vez en el día y 5s en intentos posteriores.

Estructura del programa:

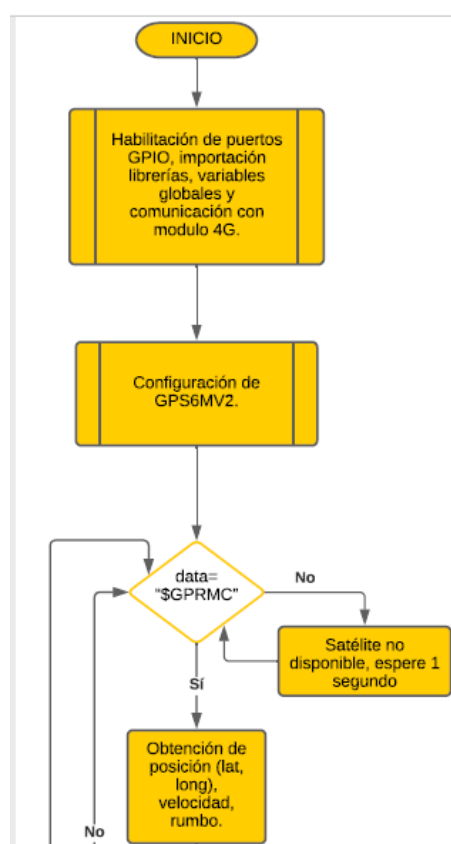


Figura 11. Diagrama de flujo programa GPS parte 1.

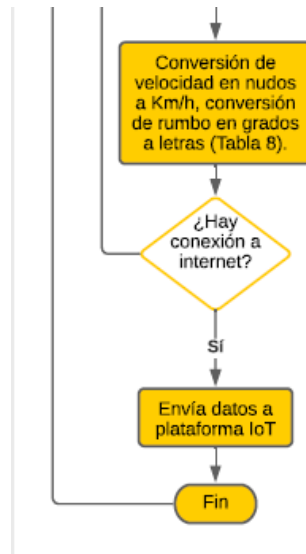


Figura 12. Diagrama de flujo programa GPS parte 2.

Después de la realización del programa con base en los protocolos NMEA, se procede a realizar el diagrama de conexiones del GPS a la Raspberry:

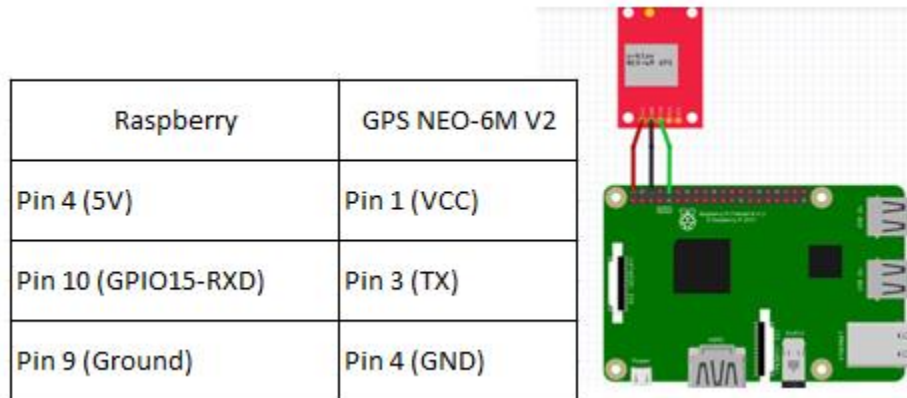


Figura 13. Diagrama de conexión GPS a Raspberry.

Como se observa en la figura 13, se tienen 3 pines conectados, ya que solo se necesita la recepción de los datos, por ende, no es necesario conectar el pin adicional.

Como se dijo anteriormente, el GPS trabaja con protocolos de comunicación NMEA, estos arrojaron en el código diferentes formatos de acuerdo a la referencia del sensor que se estaba usando, cabe destacar que dependiendo de la calidad del sensor más formatos se podrán

recibir, el que se tiene para el proyecto es de bajo costo, por lo que arrojaba 4 formatos diferentes:

- GPRMC: Este es el formato recomendado mínimo específico.
- GPGGA: Datos fijos del sistema de posicionamiento global.
- GPGSA: Sistema global de navegación por satélite y satélites activos.
- GPGSV: Sistema global de navegación por satélite y satélites a la vista.

Para el caso aplicado al proyecto, se necesita velocidad, posicionamiento y curso, estos datos los da el primero formato GPRMC, este formato contiene datos organizados de la siguiente manera:

\$GPRMC,124126.00,A,0533.287,N,07321.094 ,W,0.080,323.3,210307,0.0,E,A*20

A continuación, se muestra una tabla con el desglose del formato por campo:

# de campo	Nombre del campo	Descripción del campo	Ejemplo
1	GPRMC	Nombre del formato	GPRMC
2	UTC	Hora en formato UTC	124126.00
3	Estatus	Comprobación del estatus Válido (A) o Invalido(V)	A
4	Lat	Latitud	0533.287
5	Dirección de lat	Dirección de la latitud norte (N) o sur (S)	N
6	Lon	Longitud	07321.094
7	Dirección de lon	Dirección de la longitud Este (E) u Oeste (O)	W
8	Velocidad	Velocidad sobre el suelo en nudos	0.080
9	Curso verdadero	Curso en grados	323.3
10	Fecha	Fecha en formato dd/mm/aa	210307
11	Mag var	Variación magnética	0.0
12	Dir var	Dirección magnética	E
13	Mode ind	Indicador de modo del sistema de posicionamiento	A
14	xx	Validación de la suma	*20

Tabla 7. Descripción del formato GPRMC [35].

Después de comprobar los datos obtenidos por el formato GPRMC, se procedió a descomponer los datos solicitados para el proyecto los cuales como se dijo anteriormente son velocidad, posicionamiento y curso, para la descomposición de los datos se procedió a usar las funciones dadas por la librería “pynmea2”, las cuales comenzaban con el comando “msg.”, para la posición el comando fue “msg.lat” para la latitud, “msg.lat_dir” para la dirección de la latitud, “msg.lon” para la longitud y “msg.lon_dir” para la dirección de la longitud, teniendo así completa las coordenadas GPS para la ubicación del automotor.

Luego, se procedió a descomponer el curso, con el comando “msg.true_course”, obteniendo así el curso verdadero en grados (323°), en este caso para que la interpretación del controlador sea más cómoda y eficiente, se decidió realizar una tabla equivalente que consiste en convertir esos grados a letra teniendo en cuenta 0° como el Norte, la tabla quedó de la siguiente manera:

Letra	Grados
Norte (N)	0° a 22° y 338° a 360°
Noreste (NE)	23° a 75°
Este (E)	76° a 112°
Sureste (SE)	113° a 157°
Sur (S)	158° a 202°
Suroeste (SO)	203° a 247°
Oeste (O)	248° a 292°
Noroeste (NO)	293° a 337°

Tabla 8. Equivalente grados letra a grados.

Para que el curso sea preciso el automóvil debe cumplir con una velocidad mínima de 4 Km/h, su precisión depende de la velocidad, así que a mayor velocidad conlleva más precisión; Al no cumplir el requisito mínimo de velocidad, el rumbo se marcará con la letra C.

Finalmente, se procedió a descomponer la velocidad con el comando “msg.spd_over_gnd”, obteniendo la velocidad en tierra en nudos, para entender los datos mejor se procedió a convertir esos nudos a km/h donde se realizó una regla de tres donde se definió 1 km/h a 0.539957 nudos.

El tercer sensor fue el radar LIDAR, para examinar el funcionamiento del dispositivo se dispuso probar primero con el software otorgado por la empresa fabricante Texas Instruments, para ver el funcionamiento general del mismo, para hacer esta comprobación se procedió a descargar diferentes softwares para la calibración del sensor y la lectura del sensor por parte de Windows estos programas son mmwaveSdk y Ticloudagent.

Después de la calibración y habiendo entendido el funcionamiento del sensor en Windows, se procedió a realizar el programa en Python donde finalmente sería desarrollado el trabajo final, para esto se utilizaron librerías ya instaladas previamente para los sensores anteriores, “serial” para el envío y recepción de datos, “time” para los sleep o tiempos de espera, “pyqtgraph” para las gráficas y finalmente se añadió una nueva que se llamada “numpy”. Esta librería funciona como soporte para crear vectores y matrices multidimensionales.

Teniendo en cuenta las funciones de las librerías anteriores se propuso hacer el programa dando como resultado un error proveniente del radar LIDAR, ya que el radar necesitaba una configuración para su uso, esa configuración se hacía dependiendo su uso final, como mejor rango, mejor rango de resolución o mejor velocidad de resolución; Como se buscaba un rango amplio, se tomó la decisión de mejor rango, esta configuración se generaba automáticamente junto a otros factores como el número de antenas, las bandas de frecuencia, cuadros por segundo, rango de resolución, umbral de detección de rango, umbral de detección doppler, entre otros valores, esta configuración después de terminada se guarda en un archivo de configuración “.cfg” el cual traduce todas las peticiones antes mencionadas en un código de números, como se muestra en la siguiente imagen.

```

sensorStop
flushCfg
dfeDataOutputMode 1
channelCfg 15 3 0
adcCfg 2 1
adcbufCfg -1 0 0 1 0
profileCfg 0 77 7 7 58 0 0 68 1 256 5500 0 0 30
chirpCfg 0 0 0 0 0 0 0 1
chirpCfg 1 1 0 0 0 0 0 2
bpmCfg -1 0 0 1
frameCfg 0 1 32 0 100 1 0
lowPower 0 1
guiMonitor -1 1 1 1 0 0 1
cfarCfg -1 0 2 8 4 4 0 5120
cfarCfg -1 1 0 8 4 4 0 5120
peakGrouping -1 1 1 0 1 224
multiObjBeamForming -1 1 0.5
calibDcRangeSig -1 1 -5 8 256
extendedMaxVelocity -1 0
clutterRemoval -1 0
compRangeBiasAndRxChanPhase 0.0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
measureRangeBiasAndRxChanPhase 0 1.5 0.2
nearFieldCfg -1 0 0 20
CQRxSatMonitor 0 3 4 127 0
CQSigImgMonitor 0 63 8
analogMonitor 1 1
lvdsStreamCfg -1 0 0 0
sensorStart

```

Figura 14. Código de números generados por la configuración del RADAR LIDAR.

La figura 14 muestra el archivo “.cfg” generado por el software de Texas Instruments a base de las peticiones antes nombradas, este archivo de configuración funciona ejecutando el programa y seguido de eso con una línea dentro del programa “configFileName = 'NombreDelArchivo.cfg’” se lee el archivo de configuración de forma ascendente, primero detiene la toma de datos del sensor, aplica la configuración deseada y finalmente inicia nuevamente la toma de datos.

Estructura del programa:

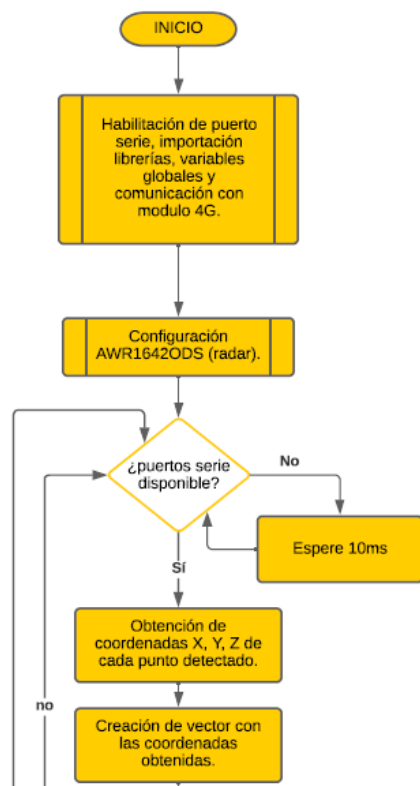


Figura 15. Diagrama de flujo programa Radar AWR1642 parte 1.

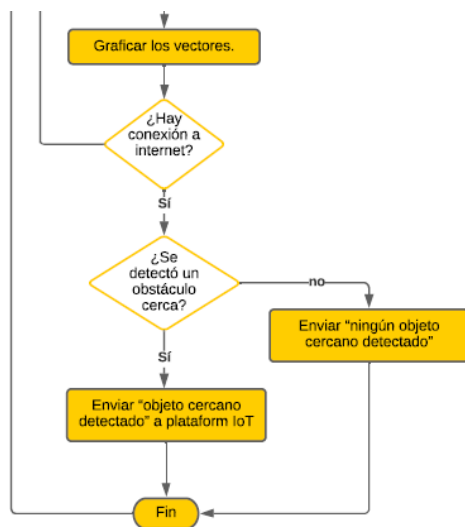


Figura 16. Diagrama de flujo programa Radar AWR1642 parte 2.

Después de aplicar la configuración, el sensor funciona de manera correcta salvo el rango, las configuraciones se plantearon de manera que el Radar pueda leer de 40 a 60 metros, con el fin tener una proyección mayor del automotor que se avecina por la parte trasera del vehículo, pero el rango máximo medía era 11m, a pesar del rango, las tomas de los datos fueron perfectas ya que los objetos grandes se marcaban como una asociación de puntos (se profundizará en la parte de resultados).

El radar LIDAR usa comunicación USB; Con respecto a la alimentación del dispositivo son 5V a 3A, demasiada carga para la Raspberry es por eso que se optó por una alimentación externa, en este caso un inversor (se explicará a detalle en la sección 6.4), por otra parte, el sensor tiene 120° de amplitud por lo que es óptimo para vías citadinas. A continuación, se muestra el diagrama de conexión del dispositivo a la Raspberry.

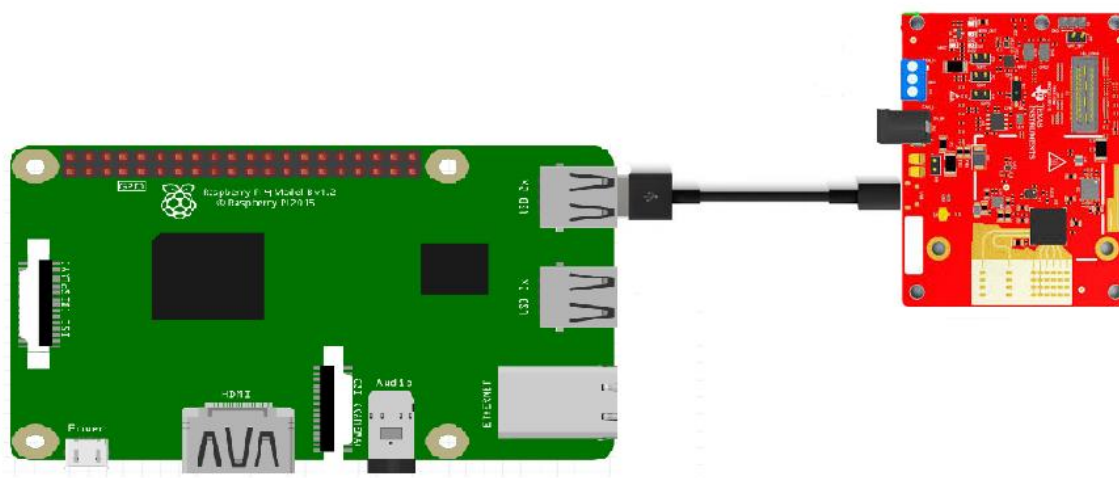


Figura 17. Diagrama de conexión Radar con Raspberry.

Finalmente, los sensores TeraRanger Evo60m se utilizaron como una ampliación del radar, para la identificación de objetos como motos y carros; Para el programa se usaron librerías que ya se habían trabajado para sensores anteriores como “numpy” que es la librería para crear vectores y matrices multidimensionales, además de esta librería se utilizaron 4 nuevas librerías, la primera es “pandas” la cual es una extensión de “numpy” pero no pertenece a está, “pandas” ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales, la segunda librería en cuestión es “matplotlib”, sirve para la creación de

gráficos a partir de listas creadas o arrays, otra de las librerías usadas en “Kmeans” sirve para la agrupación de datos sin etiquetar, finalmente la librería “collections” esta librería proporciona diferentes tipos de contenedores. “Un contenedor es un objeto que se utiliza para almacenar diferentes objetos y proporcionar una forma de acceder a los objetos contenidos e iterar sobre ellos [36]”. Algunos de los contenedores integrados son Tuple, List, Dictionary, etc.

Estructura del programa:

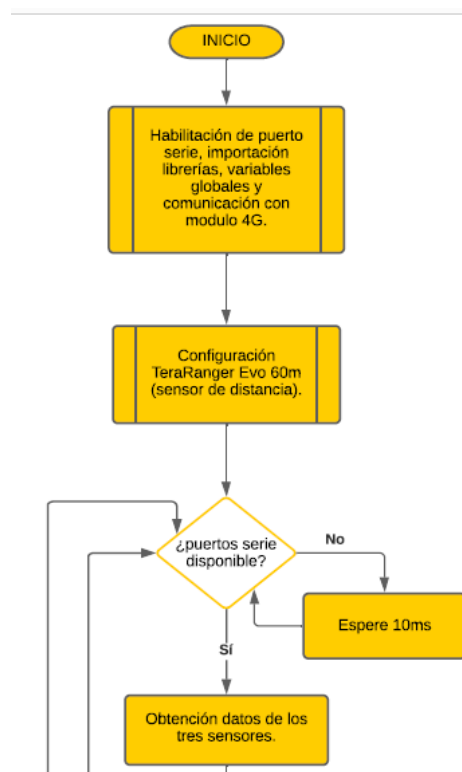


Figura 18. Diagrama de flujo programa TeraRanger Evo60m parte 1.

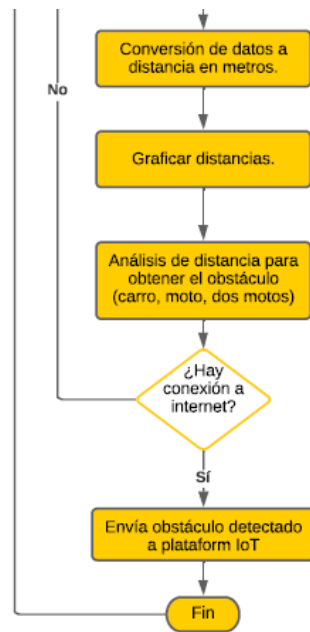


Figura 19. Diagrama de flujo programa TeraRanger Evo60m parte 2.

Los sensores tienen un campo de visión aproximado de 2° , a 1 metro este campo se traduce a 3 cm de rango en forma de cuadrado, teniendo estos datos se puede hacer una aproximación de cuánto va a hacer el rango en su máxima distancia que son 60M, para realizar esta aproximación se usa el teorema de Pitágoras.

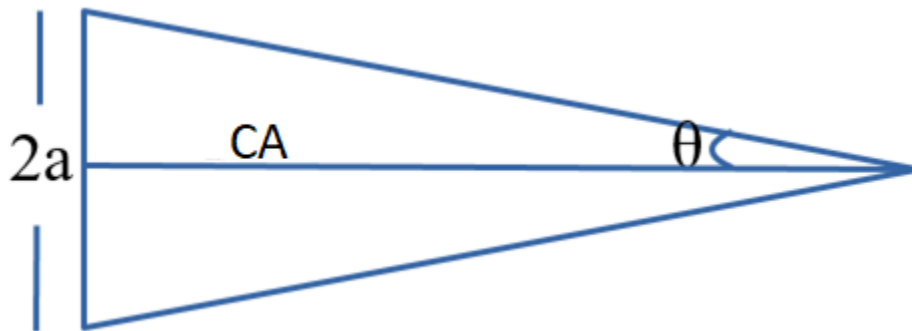


Figura 20. Rango para los sensores TeraRanger Evo60m.

Como se observa de la figura 20, el rango de visión del radar son dos triángulos rectángulos es por eso que se hace la ecuación para un solo triángulo y se multiplica el resultado por 2 para poder hallar el cateto opuesto (a), para ver el rango final a los 60 metros, por eso se aplica el teorema de Pitágoras de la siguiente manera:

Sabiendo que $\theta = 1^\circ$ y $CA = 60m$, se encuentra el $CO = a$:

$$\tan(\theta) = \frac{CO}{CA}$$

$$\tan(2^\circ) = \frac{a}{60m}$$

$$a = 60m * \tan(1^\circ)$$

$$a = 60m * 0.17$$

$$a = 1.047m$$

El Cateto Opuesto (a) de 1 triángulo rectángulo es de 1.047, al ser dos triángulos se multiplica el resultado por 2 y así se obtiene lo siguiente:

$$a = 1.047m * 2$$

$$a = 2.09m$$

Según las ecuaciones anteriores se puede determinar que a 60m se va a tener un rango de visión de 2m por cada sensor, al ser 3 sensores van a tener 6m de rango de visión; estos sensores se pueden conectar a la Raspberry de 2 formas la primera USB y la segunda tipo UART, al ser 3 sensores no alcanzan los puertos UART, por lo que se optó por conectarlos de manera USB, el diagrama de conexión es el siguiente:

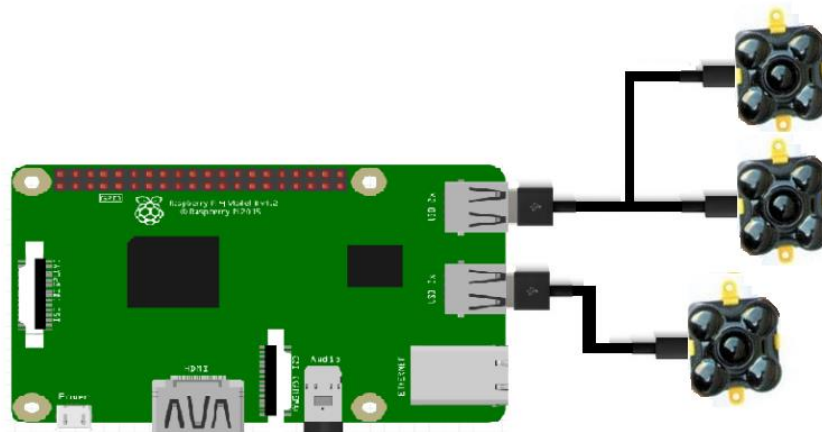


Figura 21. Diagrama de conexión para los sensores TeraRanger Evo60m.

6.3.2. Interfaz Gráfica.

Esta interfaz gráfica se conecta a través de internet 4G y se enlaza con una App online IoT, estas aplicaciones o página IoT funcionan con librerías especializadas para trabajar con diferentes lenguajes de programación entre las cuales se encuentra Python, la página seleccionada es “initial state”, esta tiene fácil manejo y es intuitiva para trabajar. Una de las ventajas principales de esta página es su capacidad, ya que permite trabajar con varios programas y Raspberry simultáneamente, al crear la cuenta en la plataforma se percató que contaba con una prueba gratuita de 14 días y después ingresaba un costo mensual de \$ 9.9 Dólares.

Al entrar en la página, lo primero que se debe hacer es crear un nuevo stream bucket o también llamado Dashboard, este sirve para identificar a que página se deben enviar los datos y así evitar que se mezclen los datos de diferentes Raspberry puesto que cada una tiene un stream bucket diferente, el comando para el código es el siguiente:

```
“streamer=Streamer (bucket_name="Nombre del stream bucket", bucket_key= "Llave del stream bucket", access_key="llave de la página “)”
```

Se establecen las llaves correspondientes a la configuración del stream bucket en los programas, lo que hizo que se actualizarán los datos en la página y se crearan diferentes Tiles, estos tiles o como su traducción literal lozas funcionan como tableros de visualización que se ven de la siguiente manera.



Figura 22. Tiles para los datos adquiridos de Raspberry.

Como se requería ver los datos de diferentes formas, se editaron cada uno de los Tiles para hacerlo de manera más intuitiva y fácil a la hora de los análisis finales quedando de la siguiente manera:

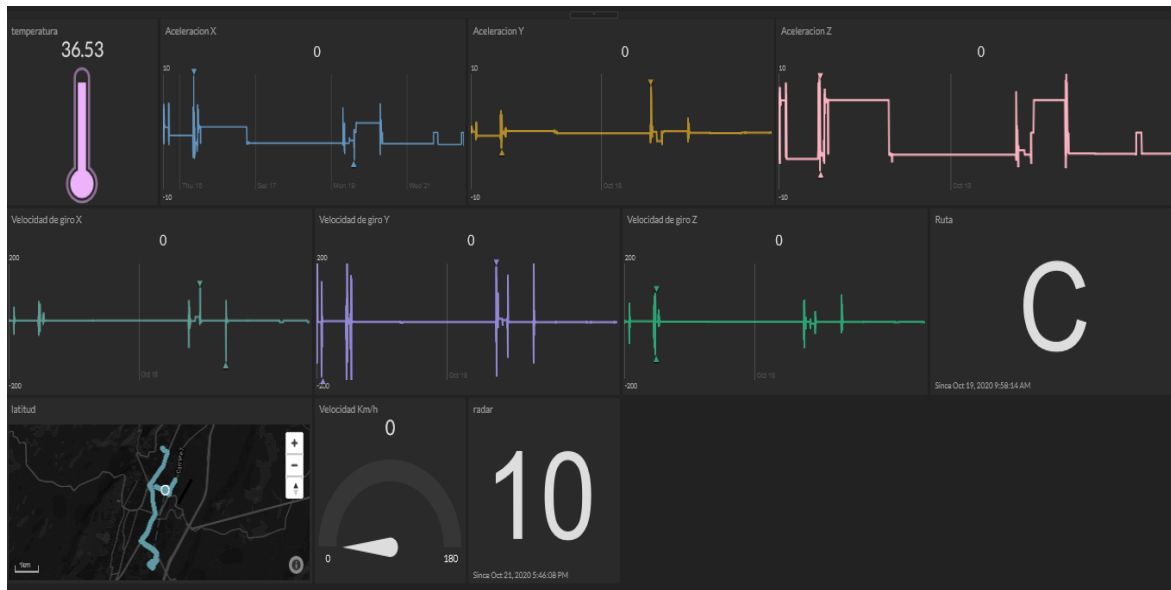


Figura 23. Interfaz gráfica en Initial State.

Como se observa en la figura 23, se tienen 11 diferentes Tiles, los cuales representan cada uno de los datos adquiridos, en la primera línea, la gráfica muestra la temperatura actual dentro del dispositivo, seguido de tres gráficas que representan la aceleración en X, Y, Z; En la segunda línea se tienen otras tres gráficas seguidas esta vez del giroscopio nuevamente en las componentes X, Y, Z, y finalmente la gráfica de la ruta en este caso representada por una C cuando no detecta movimiento. En la tercera línea, se tiene un mapa de ruta que ha tomado el carro, acompañada de un velocímetro y finalmente el número de datos que ha tomado el radar, este número es importante, gracias a él en el back up de la Raspberry se podrá ver que había detrás del carro en ese número de muestra y analizarlo junto a los datos obtenidos en la página “initial state”, cabe destacar que la pagina permite ver en tiempo real y buscar un momento según el día y la hora que se desee, como se muestra en la siguiente figura:

```

2020-10-22T17:05:48.019949000Z,Velocidad de giro Y,0.5343511450381679
2020-10-22T17:05:48.019910000Z,Velocidad de giro X,5.587786259541985
2020-10-22T17:05:48.019873000Z,Aceleracion Z,-3.845087866210937
2020-10-22T17:05:48.019832000Z,Aceleracion Y,-0.00718260498046875
2020-10-22T17:05:48.019785000Z,Aceleracion X,-2.0973206542968748
2020-10-22T17:05:48.019623000Z,temperatura,20.953529411764706
2020-10-22T17:05:47.589648000Z,Velocidad de giro Z,1.7709923664122138
2020-10-22T17:05:47.589612000Z,Velocidad de giro Y,0.061068702290076333
2020-10-22T17:05:47.589575000Z,Velocidad de giro X,5.511450381679389
2020-10-22T17:05:47.589537000Z,Aceleracion Z,-3.8091748413085935
2020-10-22T17:05:47.589492000Z,Aceleracion Y,-0.05746083984375
2020-10-22T17:05:47.589445000Z,Aceleracion X,-2.094926452636719
2020-10-22T17:05:47.589302000Z,temperatura,20.906470588235294
2020-10-22T17:05:47.115335000Z,Velocidad de giro Z,1.4427480916030535
2020-10-22T17:05:47.115298000Z,Velocidad de giro Y,0.29770992366412213
2020-10-22T17:05:47.115260000Z,Velocidad de giro X,5.648854961832061
2020-10-22T17:05:47.115222000Z,Aceleracion Z,-3.835511059570312
2020-10-22T17:05:47.115183000Z,Aceleracion Y,0.023942016601562498
2020-10-22T17:05:47.115138000Z,Aceleracion X,-2.104503259277344
2020-10-22T17:05:47.114980000Z,temperatura,21.000588235294117
2020-10-22T17:05:46.656598000Z,Velocidad de giro Z,1.4961832061068703
2020-10-22T17:05:46.656561000Z,Velocidad de giro Y,0.20610687022900764
2020-10-22T17:05:46.656523000Z,Velocidad de giro X,5.450381679389313
2020-10-22T17:05:46.656485000Z,Aceleracion Z,-3.718195178222656
2020-10-22T17:05:46.656444000Z,Aceleracion Y,-0.06943184814453125
2020-10-22T17:05:46.656399000Z,Aceleracion X,-2.0446482177734375
2020-10-22T17:05:46.656259000Z,temperatura,20.906470588235294

```

Figura 24. Fuente del archivo de initial state.

Como ya se dijo antes se modificaron los 4 programas principales e integraron las nuevas líneas de código, y la estructura del programa quedó de la siguiente manera.

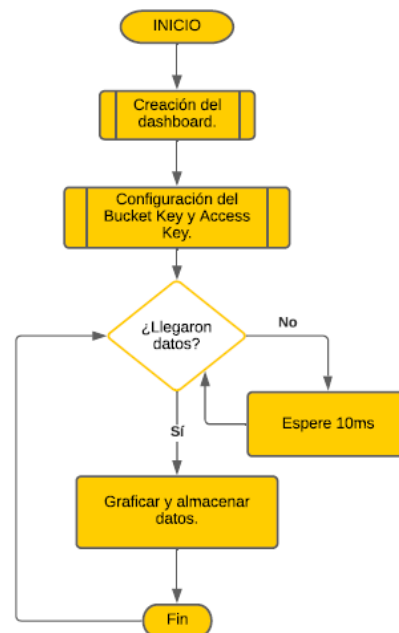


Figura 25. Diagrama de flujo subida de datos a la plataforma initial state.

6.3.3. Módulo 4G.

Para proveer internet a la Raspberry, era necesario que fuera móvil por esto se eligió un módem 4G.

El diagrama de conexión del modem es el siguiente:

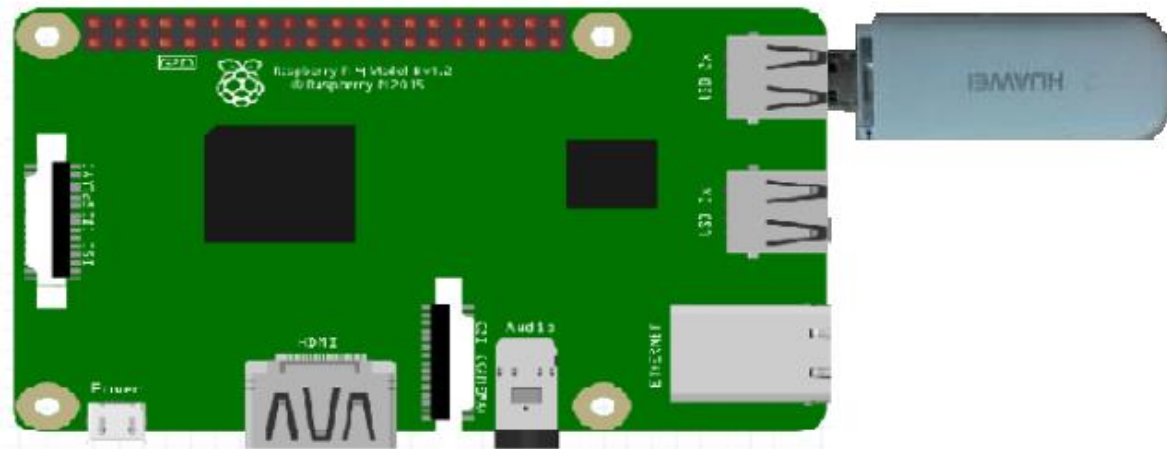


Figura 26. Diagrama de conexión modem a Raspberry.

Para el funcionamiento del modem debe haber una Sim Card, con navegación a internet, como se dijo en la sección 6.2, la empresa elegida fue *Claro Colombia*, esto por el hecho de tener la mayor cobertura, además es el más utilizado por los colombianos.

Para la Raspberry pi 4, el módem se lee como un dispositivo de red, mientras que para versiones posteriores funciona como una unidad de almacenamiento es por eso que se debe configurar de manera diferente entrando al menú de configuraciones con el comando “sudo raspi-config”.

6.4. IMPLEMENTACIÓN DEL PROTOTIPO.

Para la implementación del prototipo se decide hacer una caja protectora en acrílico, esto porque el material no interfiere con la toma de datos del radar y lo protege de ruidos electromagnéticos exteriores, además que el material no almacena calor en su interior por lo que no hay problema de sobrecalentamientos excesivos.

Esta caja tiene dimensiones de 21cm de largo por 15.5cm de ancho y con una altura de 10cm, en la que se encuentran los sensores MPU6050, GPS y el radar LIDAR, el sistema embebido y la distribución eléctrica para la alimentación de los diferentes componentes.

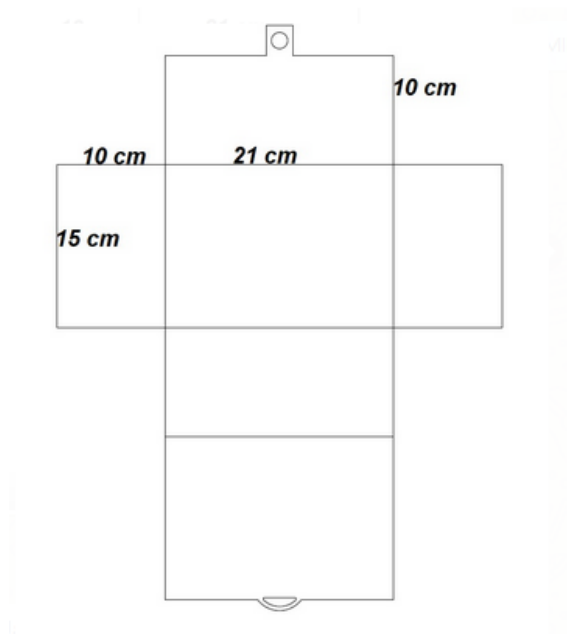


Figura 27. Diseño de la caja en acrílico.

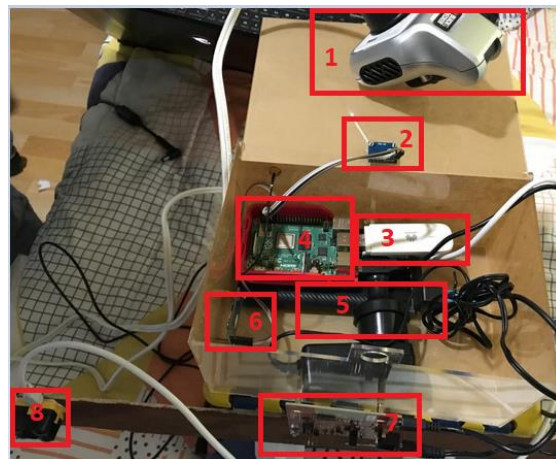


Figura 28. Distribución del dispositivo en la caja.

En la figura 28, es posible observar la distribución de los componentes, dicha distribución se hizo con el fin de hacer el prototipo más pequeño, para tener un mayor campo de acción en la parte trasera del automóvil. Los dispositivos se enumeran de la siguiente manera:

1. Inversor para auto de 100W.
2. Mpu 6050.
3. Modem 4G.
4. Raspberry.
5. Powerbank.
6. Gps Neo 60M.
7. Radar AWR1642.
8. Sensor TeraRanger Evo60m.

Aplicando los conocimientos aprendidos sobre el radar en las pruebas básicas se percató que los vidrios traseros no afectan la toma de datos del radar, por lo que se propuso dejar el prototipo en la parte superior del baúl, evitando así el peligro de caída del dispositivo en caso de que se hubiera dejado en el techo como se había planteado inicialmente, por lo que el prototipo queda de la siguiente manera:



Figura 29. Prototipo ADAS.

En la figura 29, se puede observar que el prototipo utilizó una plataforma con el fin de evitar las plumillas, debido a que estas interfieren en la toma de datos, por otra parte, se utilizó una cámara para hacer el análisis de datos más preciso, con el propósito de comparar los datos adquiridos por el radar y observar al mismo tiempo lo captado por la cámara.

Actualmente, el prototipo cuenta con dos tipos diferentes de alimentación, una para la Raspberry y la otra para el radar.

La alimentación de la Raspberry está dada por una powerbank tipo Quick Charge (QC), la cual es de carga rápida, ya que esta salida de powerbank otorga 5V a 3A, suficiente para abastecer a la Raspberry, también cuenta con capacidad de 10000mAh, por lo que la batería dura aproximadamente 5 horas y volverla a cargarla con un cargador carga rápida dura 1 hora hasta llegar al 100%.



Figura 30. Powerbank.

Por otra parte, está la alimentación del radar, se intentó probar el radar con la misma powerbank ya que la salida es 5V a 3A, pero como es QC, no funciona en el radar, así que se optó por otros tipos de alimentación como baterías lipo, el problema de las baterías lipo era el costo, por lo que para alcanzar los 2.5A que requería, se necesitaba un banco completo y esto elevaba el costo de implementación del proyecto, es por eso que se decidió probar con un inversor el cual transforma los 12V que ofrece las salidas del carro a 120V, afortunadamente se contaba con un inversor que arrojaba 100 W de potencia, por lo que se propuso a hacer los cálculos para ver si alcanzaba a soportar el inversor, la entrada del cargador, sabiendo que la potencia es:

$$P = V * I$$
$$P = 120V * 0.4A$$
$$P = 48W$$



Figura 31. Inversor.

Teniendo la potencia, se puede asegurar que es seguro usar el cargador de radar con el inversor de 100W, sabiendo eso, se procedió a conectar para ver si encendía el radar y funcionó perfectamente como lo muestra la figura 32.



Figura 32. Inversor conectado al radar LIDAR.

Finalmente se realizó un protocolo de pruebas el cual se encuentra en el anexo A, estas pruebas ayudarán a verificar el rendimiento y la eficiencia del prototipo, el resultado de la aplicación del protocolo de pruebas se expondrá en el ítem 7 resultados.

7. RESULTADOS.

Una vez culminado el prototipo, se realizan pruebas de funcionamiento con el fin de observar su efectividad, en situaciones reales que se puedan presentar. Anexo A: Protocolo de pruebas.

Esta primera sesión de pruebas fue realizada con el radar AWR1642, este dispositivo funciona con 4 antenas receptoras y 2 antenas transmisoras que envían señales a una frecuencia entre 76 a 81 GHz, su funcionamiento se basa en el movimiento y entrega información de los objetos detectados en coordenadas X, Y y Z que son las utilizadas para graficar. El radar se configuró para tener un mejor rango de resolución y una distancia de medición de 11 metros puesto que fue la configuración más óptima obtenida.

7.1. Prueba 1.

Esta prueba se realizó a las 11:00 de la mañana, para esta prueba se tomaron 4 puntos de interés en la que se captaron los autos y obstáculos con gran precisión en la medición de distancias al auto, además de esto se obtuvo la ruta realizada por el GPS, con el fin de compararla con la ruta trazada realizada en el protocolo de pruebas, y finalmente observar algunos datos como velocidad, rumbo del auto y aceleración.

Como se observa en las figuras 34, 35, 36 y 37, la figura central de la toma de datos es el objeto más grande captado por el radar, por lo que entre más grande sea el objeto más puntos a su alrededor van a haber, en las imágenes se ven los obstáculos detectados encerrados en rectángulos negros, estos encerramientos se hicieron manualmente para la demostración y comprensión más fácil del funcionamiento del radar, para el objeto principal en este caso el carro, y objetos secundarios de color rojo como se observa en la figura 37, los puntos que no están encerrados en ningún cuadro son ruido, por lo cual se realizó una modificación al programa en el que se desprecian esos datos a pesar de que se muestren.

A continuación, se muestra el mapa trazado por el GPS en la página IoT (initial State), se comparó con la ruta trazada en la figura 1 del protocolo de pruebas, además se agregaron los

4 puntos de interés mostrados con diferentes colores donde se percibe el funcionamiento del prototipo.

El primer punto de interés se toma de color rojo, el color amarillo es perteneciente al segundo punto de interés, el tercer punto de interés se encuentra de color verde y de color morado se observa el cuarto punto de interés, cabe aclarar que estos puntos fueron dibujados por los autores para una descripción fácil de entender.

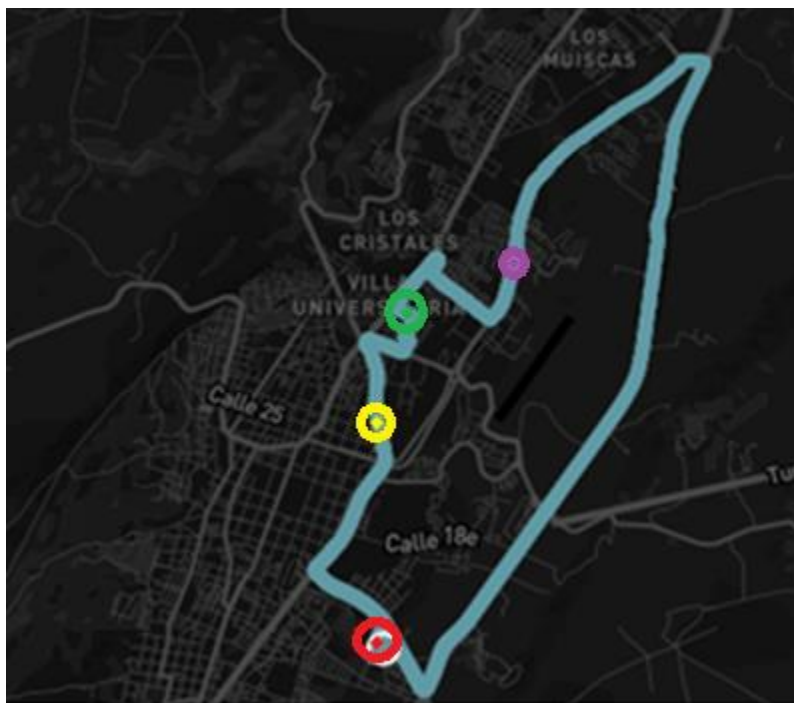


Figura 33. Mapa recorrido diurno en la plataforma (Initial State).

7.1.1. Primer punto de interés.

Este punto de interés se tomó en cuenta debido a que el vehículo transitó por una zona con baches en la vía lo que obliga a los conductores a bajar la velocidad, esto con el fin de comprobar el funcionamiento a bajas velocidades y vibraciones.



Figura 34. Toma de datos del radar prueba diurna primer punto de interés.

7.1.2. Segundo punto de interés.

En este segundo punto se toma en cuenta la avenida oriental, puesto que la velocidad aumenta en comparación del primer punto de interés, además al ser una vía principal de la ciudad, su tráfico vehicular es más alto que en el punto anterior; por lo que se quiso comprobar como influía la presencia de otros vehículos en la toma de datos al radar.

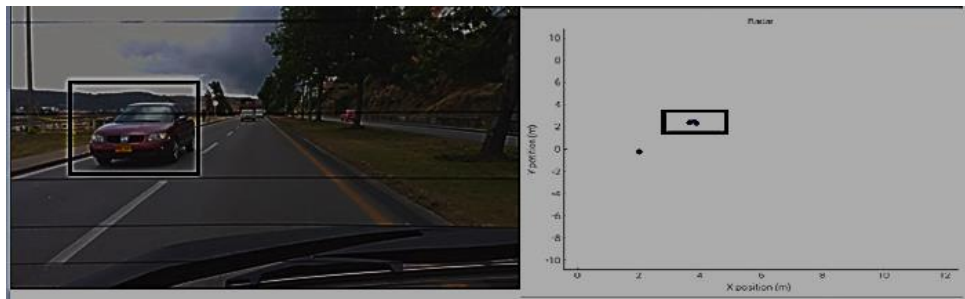


Figura 35. Toma de datos del radar prueba diurna segundo punto de interés.

7.1.3. Tercer punto de interés.

El tercer punto se tomó cerca a la vía del ferrocarril, se alcanza a observar que al haber movimientos bruscos de arriba abajo, el radar alcanza a captar en la coordenada 0,0 ruido generado por la plumilla del parabrisas.

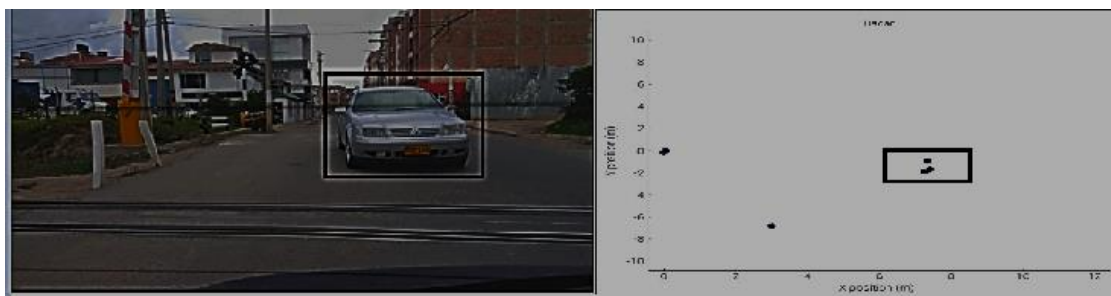


Figura 36. Toma de datos del radar prueba diurna tercer punto de interés.

7.1.4. Cuarto punto de interés.

Finalmente, el cuarto punto de interés se tomó porque fue uno de los puntos donde se captó un objeto secundario en este caso un reductor de velocidad.



Figura 37. Toma de datos del radar prueba diurna cuarto punto de interés.

Finalmente, para esta prueba se tomó un punto del mapa donde se puede comprobar la velocidad y el rumbo del automóvil, este punto se tomó en la avenida norte.



Figura 38. Ruta y velocidad en un punto específico del recorrido (Initial State).

Como se observa en la esquina superior derecha, la loza indica que el carro se dirige al Noreste, y unas lozas después se puede ver la velocidad que es de 45 km/h.

7.2. Prueba 2:

Esta prueba se realizó a las 07:00 de la noche, para esta se tomaron 4 puntos de interés en la que se captaron los autos y obstáculos con gran precisión, además de esto se obtuvo la ruta realizada por el GPS, con el fin de compararla con la ruta trazada en el protocolo de pruebas, y finalmente observar algunos datos como velocidad, rumbo del auto y aceleración.

A continuación, se muestra el mapa trazado por el GPS en la página IoT (initial State), se comparó con la ruta trazada en la figura 2 del protocolo de pruebas (Ver anexo A), además se agregaron los 4 puntos de interés mostrados con diferentes colores.

Donde el primer punto de interés se toma de color rojo, el color amarillo es perteneciente al segundo punto de interés, el tercer punto de interés se encuentra de color verde y de color morado se observa el cuarto punto de interés, cabe aclarar que estos puntos fueron dibujados por los autores para una descripción fácil de entender.

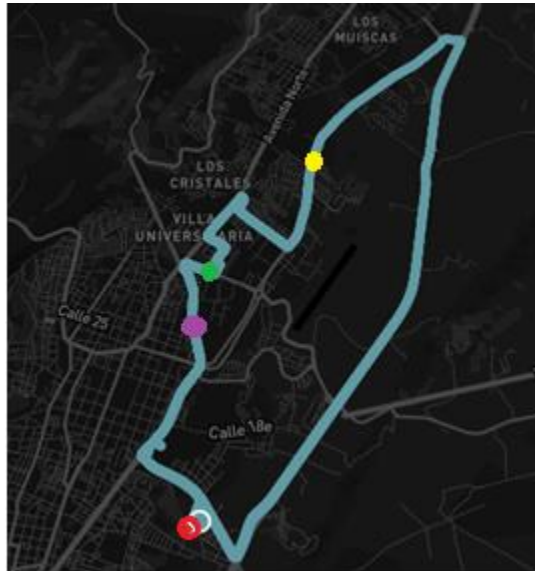


Figura 39. Mapa recorrido nocturno en la plataforma (Initial State).

7.2.1. Primer punto de interés.

El primer punto de interés se tomó en la misma calle del primer punto de interés de la primera prueba figura 34, en la cual se obtienen los mismos resultados de la mañana, únicamente se marca el carro en cuestión al no haber nada relevante que marcar como otro carro o moto.



Figura 40. Toma de datos del radar prueba nocturna primer punto de interés.

7.2.2. Segundo punto de interés.

El segundo se seleccionó para ver la capacidad de captación del sensor, como se aprecia en la figura 41, vienen dos taxis y en la gráfica se logra detallar la diferencia entre ambos.



Figura 41. Toma de datos del radar prueba nocturna segundo punto de interés.

7.2.3. Tercer punto de interés.

La tercera muestra se seleccionó por la escasa luz del ambiente, pues en este punto la luz proviene de las farolas del carro y como se observa en la figura 42 el cúmulo de puntos es más tupido.

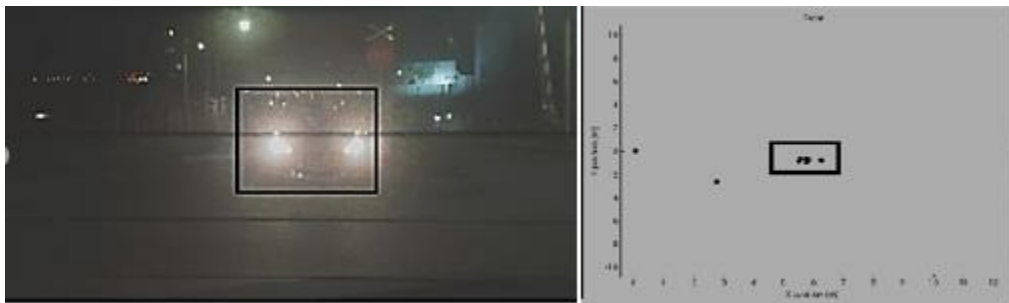


Figura 42. Toma de datos del radar prueba nocturna tercer punto de interés.

7.2.4. Cuarto punto de interés.

Finalmente, el 4 punto se toma porque el objeto, en este caso el bus de transporte público tiene gran tamaño y en la figura 43 se observa el punto inicial y final del vehículo.

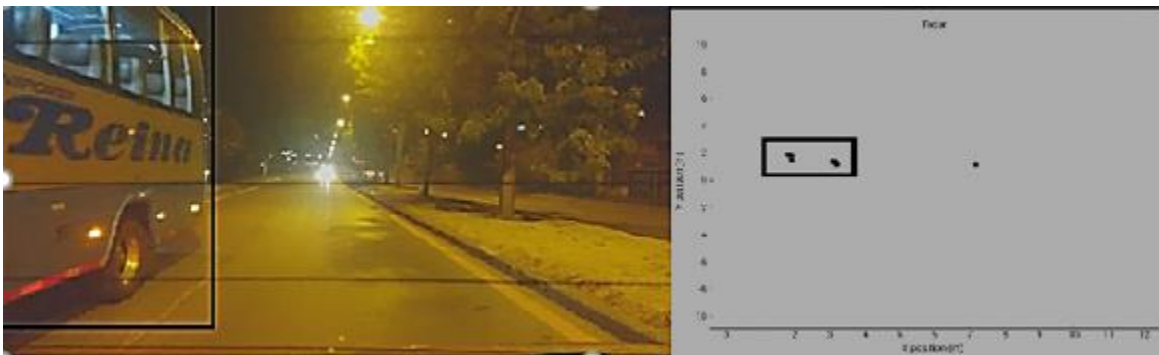


Figura 43. Toma de datos del radar prueba nocturna cuarto punto de interés.

Finalmente, para esta prueba se tomó un punto del mapa donde se puede comprobar la velocidad y el rumbo del automóvil, este punto se tomó en la circunvalar vía Sogamoso-Bogotá.

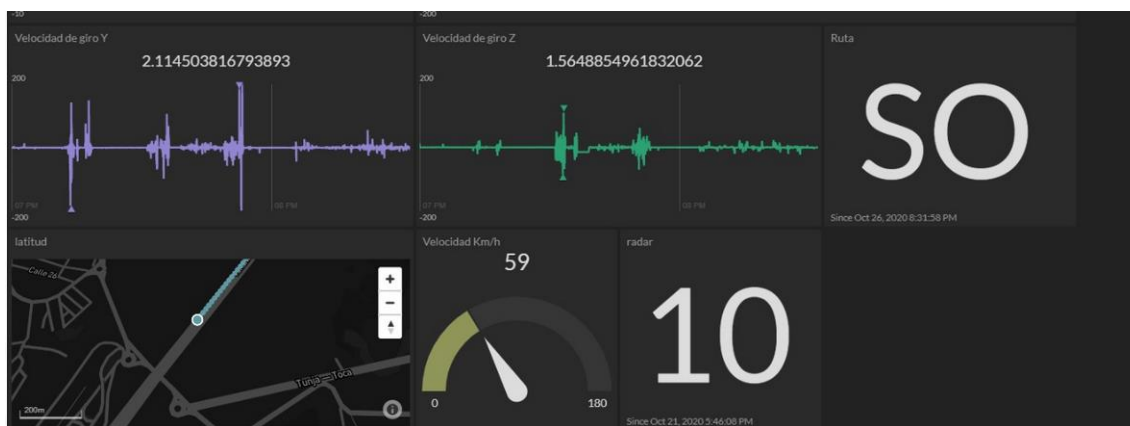


Figura 44. Ruta y velocidad en un punto específico del recorrido (Initial State).

Como se observa en la esquina superior derecha la loza indica que el carro se dirige al Suroeste, y unas lozas después se puede ver la velocidad que es de 59 km/h.

7.3. Prueba 3:

Para esta prueba se hizo una ruta distinta, ya que la neblina no se podía encontrar dentro de la ciudad de Tunja, es por eso que se decidió recorrer desde los hongos (barrio Surinama Tunja), hasta el retorno anterior al peaje de Albarracín, este tramo comprende Reten sur, la Germania y puente de Boyacá como sitios donde se tiene presente de neblina, seguido de esto se propuso tomar la misma ruta devuelta pero esta vez pasar la entrada a Tunja y seguir por la avenida circunvalar vía Sogamoso, para encontrar la neblina en esta zona hasta llegar al retorno siguiente del centro comercial Green Hills, para así terminar en el coliseo San Antonio de la ciudad de Tunja.

El mapa que se trazó en initial state fue:

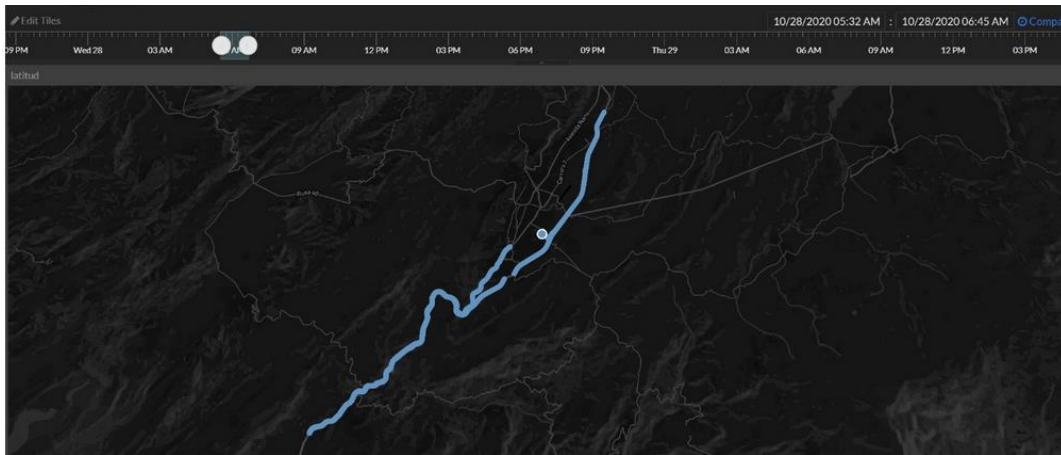


Figura 45. Mapa recorrido neblina en la plataforma (Initial State).

Para esta prueba se seleccionaron 3 puntos de interés donde normalmente existe presencia de neblina para corroborar el funcionamiento del dispositivo en este tipo de ambiente.

7.3.1. Primer punto de interés.

Para el primer punto de interés se tomó el camión de carga, por el tamaño del vehículo ya que en ninguna de las pruebas anterior se encontró un camión de gran tamaño, además que la muestra es bastante clara de la efectividad y funcionalidad del prototipo.

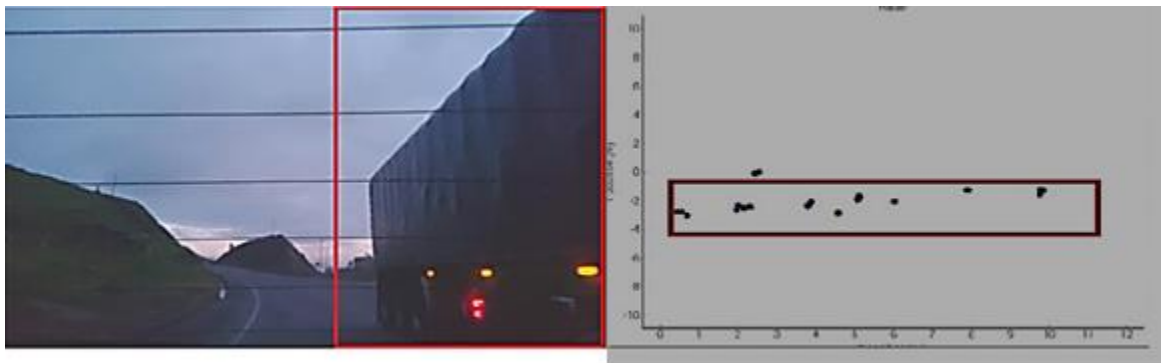


Figura 46. Toma de datos del radar prueba neblina primer punto de interés.

7.3.2. Segundo punto de interés.

Para la segunda toma se eligió este punto porque no había ningún obstáculo en la vía únicamente el carro trasero que se marcó con gran nitidez y precisión.



Figura 47. Toma de datos del radar prueba neblina segundo punto de interés.

7.3.3. Tercer punto de interés.

Finalmente, la tercera toma de datos se escogió para verificar el funcionamiento y la distancia de cobertura, como se puede observar en la figura 48, un automóvil es detectado a 3 metros mientras que el automóvil que va atrás de este, sale del rango de medición.

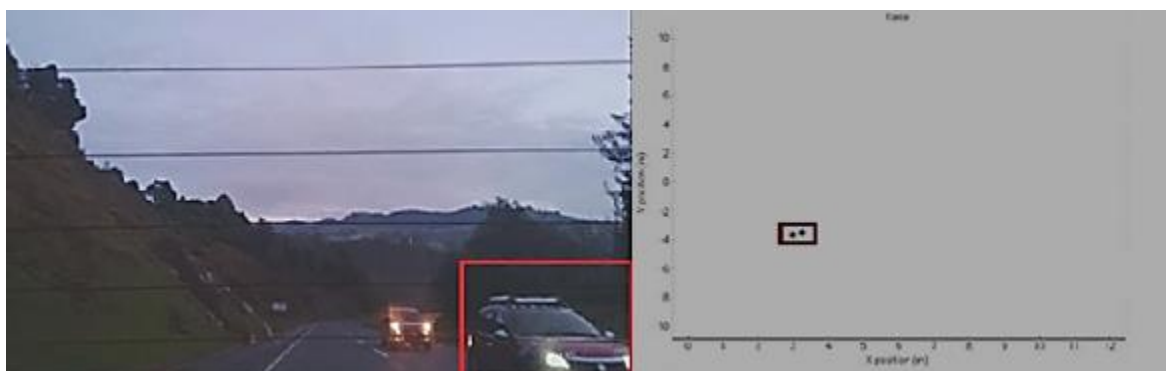


Figura 48. Toma de datos del radar prueba neblina tercer punto de interés.

7.4. Otras pruebas.

Adicionalmente al protocolo de pruebas se hicieron otras 2.

7.4.1. Prueba detección de objetos

Esta prueba consistió en la detección de objetos en el cual se creó una nueva loza en “initial state”, donde cuando el objeto está a menos de 6 metros enviará “objeto cercano detectado” y según el código nacional de tránsito la distancia prudencial a una velocidad de 30Km/h es de 10 metros [37], en caso contrario, si no, enviaba ningún objeto cercano detectado, como se muestra en las figuras 49 y 50.

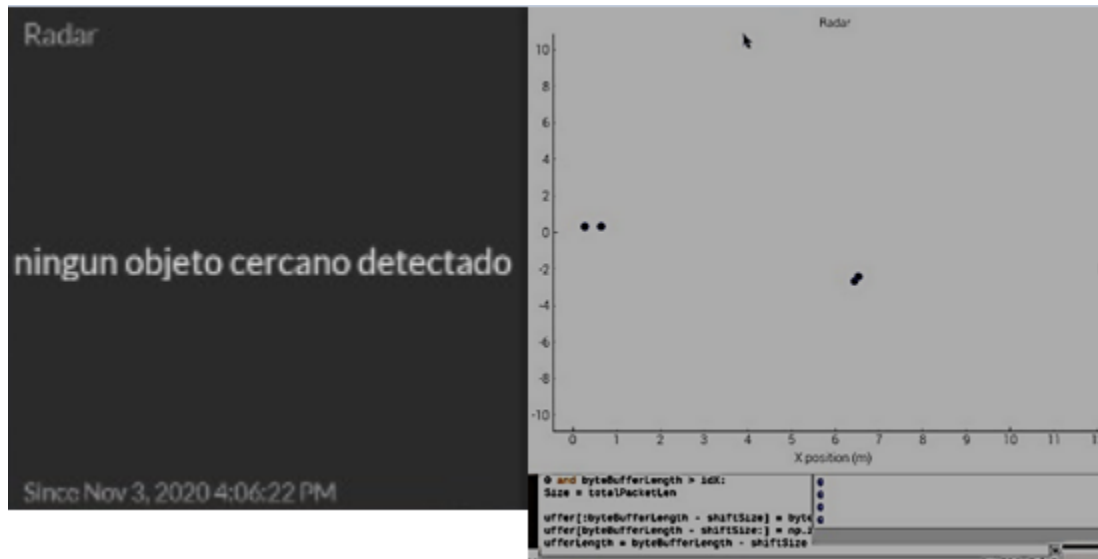


Figura 49. Ningún objeto cercano detectado. (los datos en coordenada (6.5, -2) pertenecen a un árbol ubicado en el rango de medición del radar)

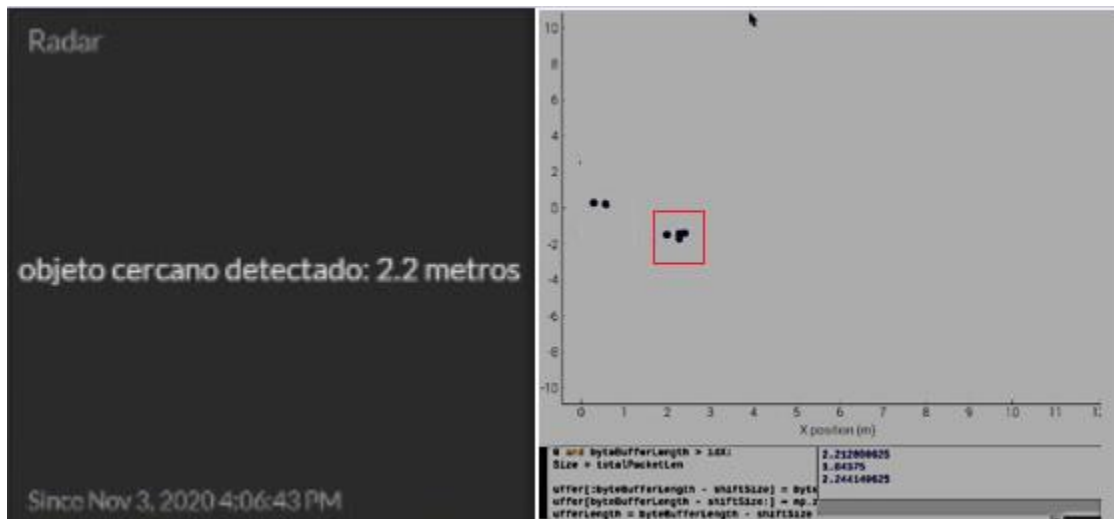


Figura 50. Objeto cercano detectado.

7.4.2. Prueba sensores TeraRanger.

La última prueba a realizar fue la de los sensores TeraRanger Evo60m, el programa se hizo bajo una lógica binaria asociada a las distancias que mide cada uno de los sensores, en el primer caso, si los tres sensores miden distancias similares con una diferencia de 1m o menos entre sus medidas, el resultado será 'automóvil' (ejemplo binario 111), esto quiere decir que el objeto que se está detectando posee una superficie plana y de más de 2 metros de ancho; Si un sensor mide una distancia y tiene una diferencia de más de 1m que los demás sensores,

el resultado será ‘moto’ (ejemplo binario 100, 010, 001); Si los sensores laterales miden una distancia mucho más pequeña que el sensor central y su diferencia es de 1m o más, el resultado será ‘dos motos’ (ejemplo binario 101). Existen dos casos especiales, cuando los sensores no pueden medir el resultado es ‘nan’ y cuando están fuera de rango el sistema arroja ‘inf’.

En esta prueba se percató que estos no funcionaban de manera correcta detrás del vidrio panorámico, esto porque es un cristal y afecta la toma de datos, se tuvo en cuenta dejarlos en la parte del baúl del auto, pero se descartó por el hecho de que ellos no cuentan con protección IP, la posición óptima es en el bómper trasero protegidos por lentes especiales que no afecten su medición, pero para esto es necesario hacer huecos en esta parte del carro, por lo que se descartó la posibilidad del uso de estos sensores.

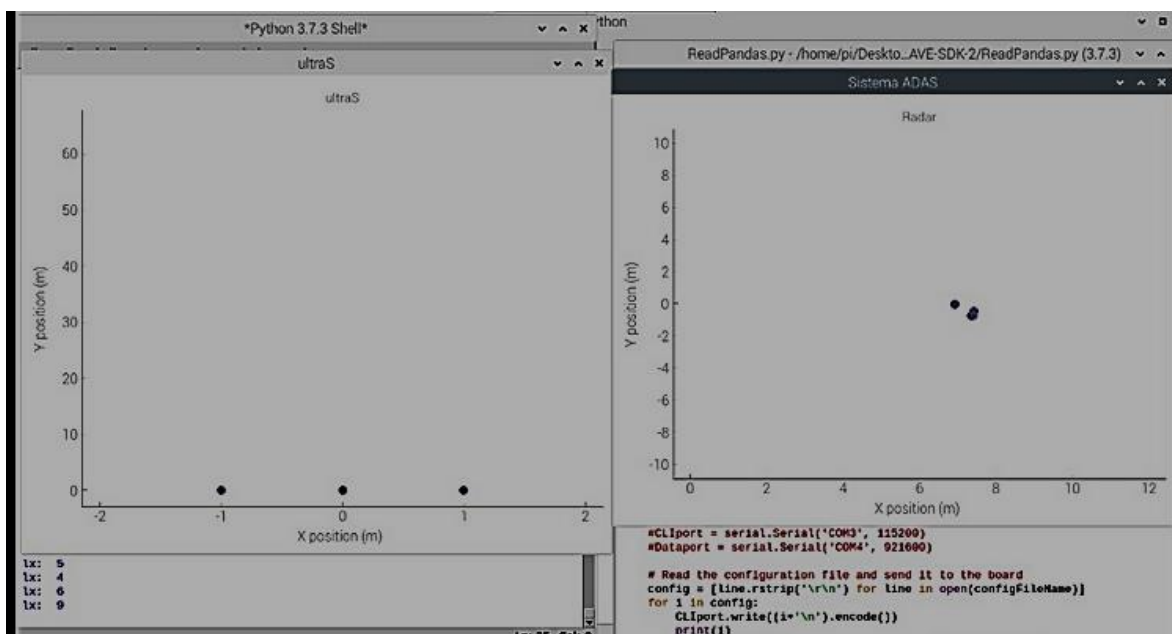


Figura 51. Prueba a través de la ventana.

Como se puede ver en la figura anterior, hay un objeto en la parte trasera del auto que es detectado por el radar (gráfica derecha), pero no es captado por los sensores TeraRanger Evo60m (gráfica izquierda) debido a la interferencia que produce el medio ambiente sobre este.

Para verificar el funcionamiento correcto, se hace una prueba sencilla donde se deja el baúl abierto y se procede ir hacia adelante y atrás con el carro, para captar los datos, a continuación, se muestra el posicionamiento del prototipo para esta funcionalidad.

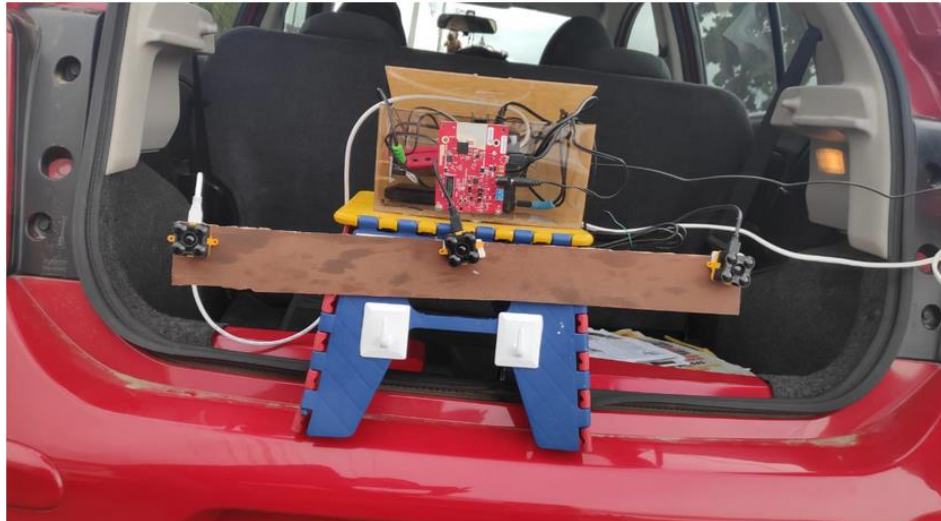


Figura 52. Prototipo ubicado en la parte inferior del carro.

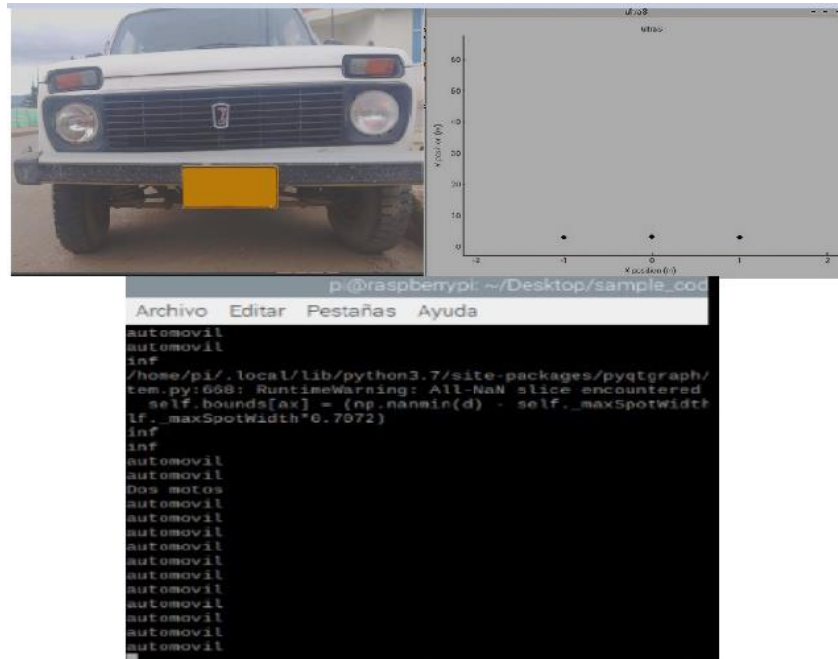


Figura 53. Prueba Auto.

7.4.3. Análisis comparativo de pruebas

PRUEBA 1	PRUEBA 2
RUTA: CIRCUITO 1 (Ver Anexo A)	RUTA: CIRCUITO 2 (Ver Anexo A)
HORA: 11 AM	HORA: 7:00 PM
Durante el desarrollo de esta prueba se captaron los datos GPS realizando un mapa en la página web Initial state, el cual arrojo como resultado el mismo mapa trazado por los autores a nivel manual, por lo anterior el sistema funciono durante el desarrollo de la prueba propuesta.	Al hacer cambio de horario y de condiciones climatológicas, el desarrollo de esta prueba se captaron los datos GPS realizando un mapa en la página web Initial state, el cual arrojo como resultado el mismo mapa trazado por los autores a nivel manual, por lo anterior el sistema funciono durante el desarrollo de la prueba propuesta.
En cada prueba realizada durante el desarrollo del circuito el radar obtuvo de forma óptima los diferentes datos arrojados por los puntos de interés.	En el cambio de horario los datos se obtuvieron en las condiciones establecidas por los autores del proyecto.
Los datos captados por la programación de los sensores mpu y gps se comprobaron frente a la velocidad del automotor y rumbo programado, en lo cual se evidencia que el proyecto es funcional.	Las condiciones climatológicas y el cambio de horario no interfieren en la toma de datos de los sensores MPU y GPS, ya que la velocidad y el rumbo coincidieron con la dirección del carro.

Tabla 9. Análisis comparativo de pruebas.

8. CONCLUSIONES.

- Se diseñó un sistema que puede adquirir los datos en tiempo real del sistema ADAS, dentro de la investigación se tuvieron en cuenta diferentes tecnologías IoT para el envío de datos captados en tiempo real, resultando escogida la no convencional 4G *Claro Colombia* por ser el operador con mayor cobertura de red móvil del país, además de poseer más de la mitad de usuarios de internet móvil en Colombia con 57.56%.
- Dentro del análisis, se evidenció que el funcionamiento de los sensores Evo60m no fue el esperado debido a que sus mediciones se ven alteradas por materiales como el vidrio, también se ven afectados por el medio ambiente, es por esto que lo ideal es abrir orificios en el bómper trasero y acoplarlos con lentes especiales para protegerlos puesto que estos no cuentan con alguna protección IP.
- Se puede concluir que la implementación del prototipo mediante el radar AWR1642 es mejor a comparación de los sensores Evo60m, no solo en su funcionamiento sino también en costos considerando que los tres sensores Evo60m cuestan \$1'644.543 COP mientras que el radar cuesta alrededor de \$1'160.000 COP. Para el correcto funcionamiento de los sensores Evo60m habría que invertir más dinero en los lentes especiales para protegerlos contra agua y polvo.
- De acuerdo con los resultados obtenidos y mostrados en el capítulo siete podemos concluir que la eficacia del radar en las pruebas realizadas fue buena, debido a que los datos obtenidos estuvieron dentro del rango de distancia máxima programado y los obstáculos detectados en las gráficas coincidían en las posiciones con las fotos del mundo real, pero cabe destacar que el rango no fue el esperado; Se tenía previsto que abarcara entre 30 y 60 metros de distancia, pero lo máximo fueron 10, ya que para tomar datos a mayor distancia la antena receptora se configura con una sensibilidad mayor lo que la hace propensa al ruido del medio ambiente y por consiguiente afecta la toma de datos, se puede diseñar un filtro para intentar corregir este problema pero esto aumentaría el tiempo de procesamiento de los datos, otra solución es cambiar las antenas por unas con mayor potencia de transmisión para aumentar su alcance.

- Con base en los resultados obtenidos en el proyecto por medio de la puesta en marcha del RADAR durante las actividades que se ejecutaron en el protocolo de pruebas, testeado en condiciones climatológicas diferentes con el fin de constatar que el dispositivo no se ve afectado por el cambio de horario ni ambiente, así verificar el funcionamiento del RADAR de acuerdo a los objetivos propuestos en el proyecto y evaluados en los resultados del capítulo siete.

BIBLIOGRAFÍA

- [1] Amate. C (2014), La importancia de la seguridad informática en los automóviles, Obtenido el 4 de noviembre de 2020 de <https://blogthinkbig.com/seguridad-informatica-en-los-automoviles>
- [2] Ibáñez. P (2011), Sistemas de detección en los coches para evitar accidentes Obtenido el 4 de noviembre de 2020 de <https://www.xataka.com/automovil/sistemas-de-deteccion-en-los-coches-para-evitar-accidentes>
- [3] ANSV, Agencia nacional de seguridad vial, BOLETÍN ESTADÍSTICO FALLECIDOS, LESIONADOS Y ACCIDENTES EN HECHOS DE TRÁNSITO - COLOMBIA (2018-2019). Obtenido de: <https://ansv.gov.co/observatorio>
- [4] RUNT, Registro único nacional de tránsito. Obtenido de: <https://www.runt.com.co/> (2019 y 2020).
- [5] ANDEMOS, Asociación Nacional de Movilidad Sostenible, Informe Vehículos junio 2019-06. Obtenido de: <http://www.andemos.org/index.php/2019/07/08/junio-7/> (2018).
- [6] El tiempo (2015), Con bolsas de aire, el país se pone a tono con el mundo Obtenido el 4 de noviembre de 2020 de <https://www.eltiempo.com/archivo/documento/CMS-16397843>
- [7] Portafolio (2018), En el país circulan 6,2 millones de carros sin Soat obtenido el 15 de septiembre de 2020 de <https://www.portafolio.co/economia/en-el-pais-circulan-6-2-millones-de-carros-sin-soat-524003>
- [8] Coy. E (2013), MORTALIDAD PREMATURA POR ACCIDENTES DE TRÁNSITO EN COLOMBIA EN EL PERIODO 2005-2013. Obtenido el 4 de noviembre de 2020 de https://bdigital.uexternado.edu.co/bitstream/001/437/1/DKA-spa-2017-Mortalidad_prematura_por_accidentes_de_tr%C3%A1nsito_en_Colombia_en_el_periodo_2005-2013.pdf
- [9] Ecomotor (2018), Sistemas ADAS: qué son, cómo funcionan y por qué la DGT los considera tan importantes para la seguridad obtenido el 15 de septiembre de 2020 de <https://www.eleconomista.es/ecomotor/motor/noticias/9391326/09/18/Sistemas-ADAS-que-son-como-funcionan-y-por-que-la-DGT-los-considera-tan-importantes-para-la-seguridad-de-los-vehiculos.html>
- [10] Lashkov, I., Smirnov, A., Kashevnik, A., & Parfenov, V. (s.f.). Ontology-Based Approach and Implementation of ADAS System for Mobile Device Use While Driving. En Communications in Computer and Information Science (págs. 117-131). International Conference on Knowledge Engineering and the Semantic Web.
- [11] Programa de Evaluación de Vehículos Nuevos para América Latina y el Caribe, década de acción para la seguridad vial. (2011-2020). Obtenido de: <https://www.latinncap.com/es/>.
- [12] Jung, S., Lee, U., Jung, J., & Shim, D. H. (2016). Real-time Traffic Sign Recognition system with deep convolutional neural network. 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI). Xi'an, China.

- [13] ANSV, ¿quiénes somos? Obtenido el 15 de septiembre de 2020 de <https://ansv.gov.co/index.html>
- [14] BBVA, ¿Qué es el SOAT? Obtenido el 15 de septiembre de 2020 de <https://www.bbva.com.co/personas/blog/educacion-financiera/seguros/que-es-soat.html>
- [15] Wikipedia (2020), Pseudo, Obtenido el 4 de noviembre de 2020 de <https://es.wikipedia.org/wiki/Pseudo>
- [16] Wikipedia (2020), Accidente, Obtenido el 4 de noviembre de 2020 de <https://es.wikipedia.org/wiki/Accidente>
- [17] Rose. K, Eldridge. S, Chapin, L (2015), LA INTERNET DE LAS COSAS— UNA BREVE RESEÑA, Obtenido el 4 de noviembre de 2020 de <https://www.internetsociety.org/wp-content/uploads/2017/09/report-InternetOfThings-20160817-es-1.pdf>.
- [18] U-BLOX, Neo -6 GPS modules Data sheet, obtenido de: https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf?utm_source=en%2Fimages%2Fdownloads%2FProduct_Docs%2FNEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf
- [19] UniMinuto, Las Leyes de Newton, Obtenido el 4 de noviembre de 2020 de <https://sites.google.com/site/uniminutofisica/g-leyes-de-newton>
- [20] Raspberry pi, Raspberry pi model B data sheet obtenido de: https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf
- [21] Descripción Python 3 obtenido de: <http://docs.python.org.ar/tutorial/3/real-index.html>
- [22] López. M (2015), ¿Cómo funciona el radar?, Obtenido el 15 de septiembre de 2020 de <https://www.unocero.com/noticias/como-funciona-el-radar/>
- [23] Castillo. G (2019), Sistemas de seguridad ADAS: Cuáles son y cómo utilizarlos antes de que sean obligatorios, obtenido el 16 de septiembre de 2020 de https://www.autopista.es/noticias-motor/sistemas-de-seguridad-adas-cuales-son-y-como-utilizarlos-antes-de-que-sean-obligatorios_155205_102.html
- [24] <https://ferretronica.com/>
- [25] <https://www.sigmaelectronica.net/>
- [26] <https://www.digikey.com/es>
- [27] Wikipedia (2019), Angulo de visión obtenido el 25 de octubre de 2020 de https://es.wikipedia.org/wiki/%C3%81ngulo_de_visi%C3%B3n
- [28] Wikipedia (2020), Sistema embebido obtenido el 25 de octubre de 2020 de https://es.wikipedia.org/wiki/Sistema_embebido#Aplicaciones_de_un_sistema_embebido
- [29] Manuti (2014), Velocidad del procesador de la Raspberry Pi y overclocking obtenido el 26 de octubre de 2020 de <https://rasberryparatorpes.net/hardware/velocidad-del-procesador-de-la-raspberry-pi-y-overclocking/>
- [30] Sigfox, Coverage obtenido el 26 de octubre de 2020 de <https://www.sigfox.com/en/coverage>
- [31] Lora, Coverage obtenido el 26 de octubre de 2020 de <https://iotics.com.co/>
- [32] Nperf (2020), Mapa de cobertura 3G / 4G Móvil Colombia Obtenido el 4 de noviembre de 2020 de <https://www.nperf.com/es/map/CO/-/198197.Claro->

- Movil/signal/?ll=1.4500404973608074&lg=-70.31250000000001&zoom=5
- [33] MicTic obtenido el 26 de octubre de 2020 de <https://www.mintic.gov.co/portal/inicio/>
- [34] Riba. A, Tejada. J, Acosta. N, Tolosa. J, Procesamiento de sentencias NMEA-0183 para el análisis de la geometría satelital utilizando receptores GPS de bajo costo obtenido el 27 de octubre de 2020 de http://sedici.unlp.edu.ar/bitstream/handle/10915/56419/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y
- [35] Baddeley. G (2001), GPS - NMEA sentence information obtenido el 4 de noviembre de 2020 de <http://aprs.gids.nl/nmea/>
- [36] Data Carpentry (2016), Análisis y visualización de datos usando Python obtenido el 27 de octubre de 2020 de <https://datacarpentry.org/python-ecology-lesson-es/aio.html>
- [37] Código nacional de tránsito (2002), obtenido el 6 de noviembre de 2020 de: https://www.movilidadbogota.gov.co/web/sites/default/files/ley-769-de-2002-codigo-nacional-de-transito_3704_0.pdf

ANEXOS

A. PROTOCOLO DE PRUEBAS.

En el siguiente anexo se darán a conocer las pruebas que se realizarán del prototipo con el fin de verificar el correcto funcionamiento en circunstancias de la vida cotidiana, las siguientes serán las pruebas a realizar:

1. Prueba en ambiente diurno: La prueba está pensada para observar la efectividad del sensor GPS, el MPU, el sistema 4G y el radar; Para realizar esta prueba se trazó una ruta donde se encontrarán dos diferentes ambientes, uno congestionado donde se encontrarán carros edificios altos, entre otros obstáculos, y otro ambiente más despejado donde habrá gran cantidad de espacio libre e intentar evitar la mayoría de carros posibles, la ruta es la siguiente:

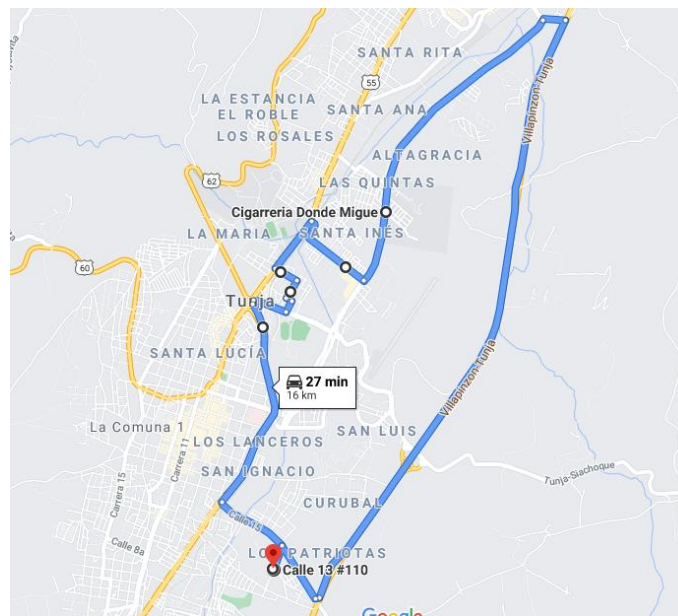


Figura 1. Mapa trazado para la ruta diurna.

La ruta consiste en iniciar desde el conjunto cerrado Mirador De Oriente, No, Cl. 13 #110, Tunja, Boyacá, ubicado al oriente de la ciudad, bajar por la avenida patriotas

vía este a oeste hasta la avenida norte antiguo terminal, luego, avanzar hacia norte hasta llegar a la glorieta norte y entrar al barrio Mesopotamia, este fragmento de la ruta se hace con las siguientes premisas la avenida norte tiene un gran flujo de autos, motos y camiones, por lo que se pondrá a prueba el dispositivo alrededor de ruido eléctrico; después de entrar al barrio Mesopotamia se circulará por el barrio anteriormente nombrado y se volverá entrar a la avenida norte para tomar la próxima salida por el barrio santa Inés, esto con el fin de llegar al centro comercial Unicentro, para finalmente tomar la avenida universitaria y llegar al centro comercial Viva Tunja y concluir este tramo del viaje, se tuvo a consideración estos barrios mencionados anteriormente (Mesopotamia y Santa Inés), por el hecho de que estos están rodeados por edificios altos, se quiere observar que no se pierdan datos por estas razones además de que estos barrios tienen una gran congestión de casas, después se toma la avenida universitaria pasando por la Universidad Santo Tomás campus universitario, hasta llegar al centro comercial Green Hills, para finalmente tomar la circunvalar y retornar al punto de inicio el conjunto cerrado Mirador de oriente; este último tramo se seleccionó porque es de ambiente abierto lo cual disminuye notablemente el flujo de autos a comparación de la avenida norte.

Esta prueba se realizó el lunes 26 de octubre a las 11:00 Am.

2. Prueba en ambiente nocturno: Esta prueba está pensada para observar la efectividad del sensor GPS, el MPU, el sistema 4G y el radar, se trazó una ruta donde se encontraran dos diferentes ambientes, uno congestionado donde se encontrarán carros edificios altos, entre otros obstáculos, y otro ambiente más despejado donde habrá gran cantidad de espacio libre e intentar evitar la mayoría de carros posibles, se tomó la misma ruta del ambiente diurno para realizar la comparación entre los diferentes ambientes, la ruta es la siguiente:

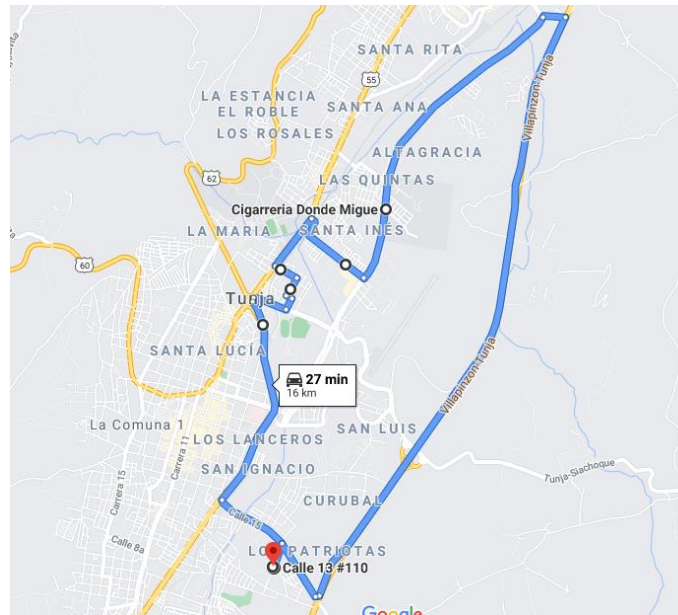


Figura 2. Mapa trazado para la ruta nocturna.

La ruta consiste en iniciar desde el conjunto cerrado Mirador De Oriente, No, Cl. 13 #110, Tunja, Boyacá, ubicado al oriente de la ciudad, bajar por la avenida patriotas vía este a oeste, hasta la avenida norte antiguo terminal, luego avanzar hacia norte hasta llegar a la glorieta norte y entrar al barrio Mesopotamia, este fragmento de la ruta se hace con las siguientes premisas la avenida norte tiene un gran flujo de autos, motos y camiones, por lo que se pondrá a prueba el dispositivo alrededor de ruido eléctrico; después de entrar al barrio Mesopotamia se circulará por el barrio anteriormente nombrado y se volverá entrar a la avenida norte para tomar la próxima salida por el barrio santa Inés, esto con el fin de llegar al centro comercial Unicentro, para finalmente tomar la avenida universitaria y llegar al centro comercial Viva Tunja y concluir este tramo del viaje, se tuvo a consideración estos barrios mencionados anteriormente (Mesopotamia y Santa Inés), por el hecho de que estos barrios están rodeados de edificios altos, y se quiere ver que no se pierdan datos por estas razones además de que en estos barrios existe una gran congestión de casas, después de que se toma la avenida universitaria pasando por la universidad Santo Tomás campus universitario, hasta llegar al centro comerciar Green hills, para finalmente tomar la circunvalar y retornar al punto de inicio el conjunto cerrado Mirador de oriente; este

último tramo se seleccionó porque es de ambiente abierto por lo que disminuye notablemente el flujo de autos a comparación de la avenida norte.

Esta prueba se realizará el lunes 26 de octubre a las 07:00 Pm.

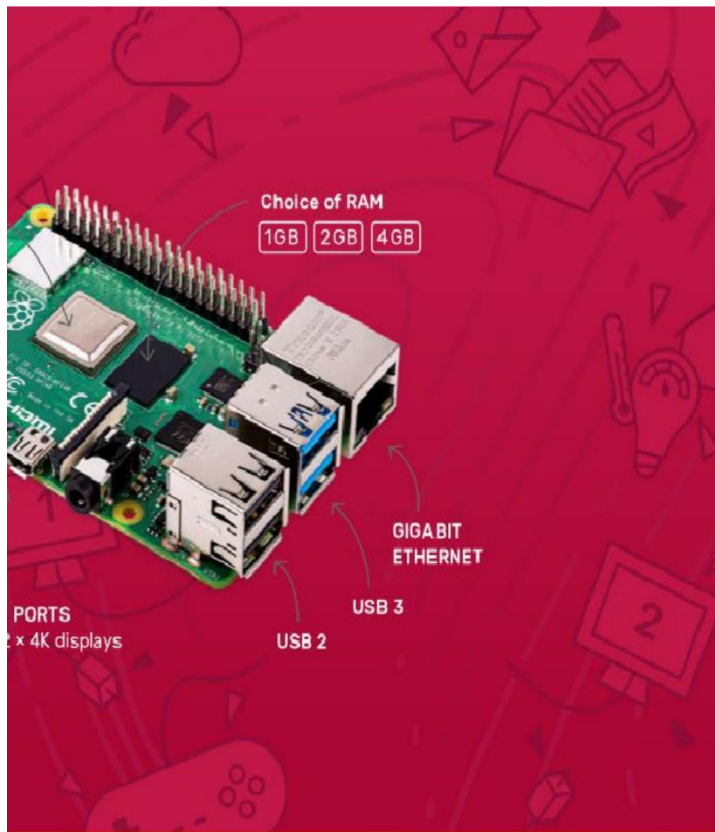
3. Prueba en ambiente neblina: Esta prueba está pensada para observar la efectividad del sensor GPS, el MPU, el sistema de red y el radar, para esta prueba no se trazó una ruta clara, esto porque no se sabe hacia donde va a haber neblina el día de la prueba, es por eso que se recorrerá por gran parte de la periferia de Tunja en busca de la neblina, este recorrido se llevará a cabo el día 27 de octubre a las 05:30 Am, puesto que a esa hora la neblina se está levantando.

B. PRESUPUESTO

ELEMENTO	CANTIDAD	PRECIO UNITARIO (COP)	TOTAL (COP)
Sensor MPU 6050	1	\$ 9.950	\$ 9.950
GPS NEO-6M V2	1	\$ 38.900	\$ 38.900
Antena GPS Activa	1	\$ 22.000	\$ 22.000
Cable Adaptador RF SMA.	1	\$ 6.000	\$ 6.000
Radar AWR 1642	1	\$ 1'160.000	\$ 1'160.000
Cargador 5V 3A para el Radar	1	\$ 18.000	\$ 18.000
Cables Hembra- Hembra	1	\$ 8.000	\$ 8.000
PowerBank 10000mAh	1	\$ 65.000	\$ 65.000
Inversor para carro de 100W Black & Decker	1	\$ 89.300	\$ 89.300
Kit de sobre mesa Raspberry pi 4	1	\$ 535.500	\$ 535.500
Caja de acrílico	1	\$ 40.000	\$ 40.000
Extensión de 2m	1	\$ 20.000	\$ 20.000
TeraRanger EVO 60M	3	\$ 548.181	\$ 1'644.543

Mensualidad página web	1	\$ 37.230	\$ 37.230
TOTAL			\$ 3'694.423

C. MANUAL DE USUARIO Y TÉCNICO.



Manual de usuario Prototipo ADAS con tecnología IoT y radar Lidar.

**Santiago Andrés
Montoya Flórez**

Jefersson Niño Silva

**Cesar Mauricio Galarza
Bogotá**

**Luis Fernando
Castellanos Guarín**

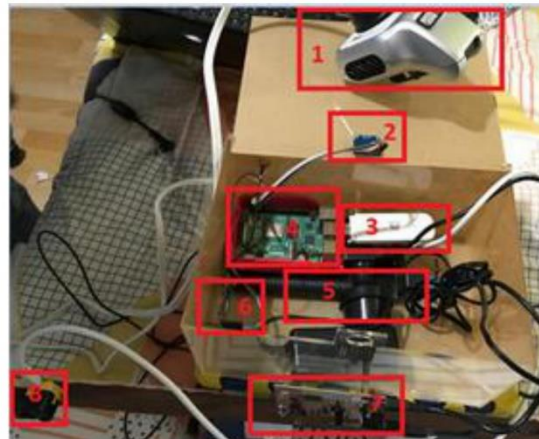
¿Qué es un sistema ADAS?, es un Sistema Avanzado de Asistencia al Conductor, son pequeños dispositivos ligados al carro con el fin de ayudar al conductor en las diferentes vías.

Los sistemas ADAS están diseñados con el fin de facilitar y proteger la integridad de los conductores y sus acompañantes, es por eso que dependiendo su utilidad cambiará su arquitectura y su comunicación con el usuario, como sensores internos o externos, que actúen como fuente de información, para prevenir o disminuir la cantidad de accidentes de tránsito provocado por errores humanos.

Se diseñó e implementó un prototipo ADAS con comunicación IoT con radar LIDAR para detección de pseudoaccidentes, brindando a los conductores más facilidades para una conducción más óptima y segura, reduciendo así los índices de accidentalidad en la ciudad de Tunja.

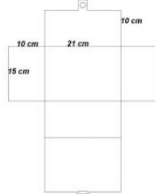
Descripción del sistema

El prototipo ADAS se encuentra almacenado en una caja de acrílico de 2mm de grosor, cuenta con un módulo GPS NEO6MV2 + antena GPS con frecuencia 1575.42MHz, un sensor MPU6050, un radar tipo Lidar AWR1642BoostODS, tres sensores de distancia Evo60m de la familia TeraRange, un módulo 4G para conexión a Internet y todo esto siendo controlado por un sistema embebido Raspberry Pi 4 modelo b, esto se alimenta de un inversor para automóvil de 150W.



Componentes:

- Caja de acrílico: 2mm de grosor.



- GPS NEO6MV2, 5V de alimentación máxima y 45mA de corriente de funcionamiento.
- MPU6050, 5V de alimentación y 42mA de corriente de funcionamiento.
- Radar AWR1642 5V-2.5A alimentación y cable micro USB.
- Evo60m campo de visión de 2° y cable micro USB.
- Módulo 4G con conexión USB tipo A.
- Raspberry Pi 4 modelo B.
- Inversor para automóvil de 150W.

1 y 5. alimentación (inversor de 100W y PowerBank Q.C3.0 o inversor de 150W).

2. MPU6050.

3. Modulo 4G.

4. Raspberry Pi 4 modelo B.

6. GPS-NEO6MV2.

7. AWR1642BoostODS.

8. Evo60M.

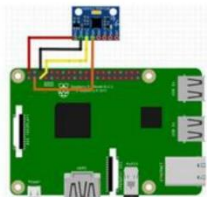
Especificaciones técnicas

Nombre	Imagen	Corriente	Voltaje	Temperatura de funcionamiento(°C)	Precio (COP)
Modulo GPSNeo6MV2		45mA	3.3V-5V DC	-40 a +85	\$38.000,00
MPU6050	 [1]	500µA-3.9mA	2.375V-3.46V DC	-40 a +85	\$9,950.00
AWR1642	 [2]	2.5A	5V DC	-40 a +125	\$1,157,264.65
Evo60M	 [3]	90mA-330mA	5V DC	-10 a +45	\$559,783.00
Modulo 4G	 [4]	400mA	5V DC	N/A	50,816.23
Raspberry Pi 4 modelo B	 [5]	3A	5V DC	0 a 50	\$390,000.00

Diagramas de conexión

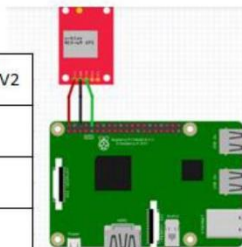
- Raspberry Pi 4- MPU6050

Raspberry	MPU 6050
Pin 2 (5V)	Pin 1 (VCC)
Pin 6 (Ground)	Pin 2 (GND)
Pin 5 (GPIO3-SCL)	Pin 3 (SCL)
Pin 3 (GPIO 2-SDA)	Pin 4 (SDA)



- Raspberry Pi 4- Modulo GPSNeo6MV2

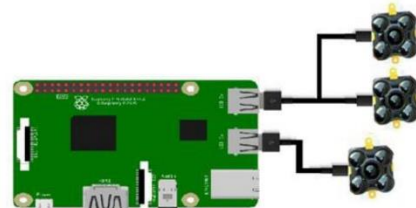
Raspberry	GPS NEO-6M V2
Pin 4 (5V)	Pin 1 (VCC)
Pin 10 (GPIO15-RXD)	Pin 3 (TX)
Pin 9 (Ground)	Pin 4 (GND)



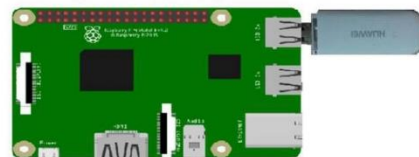
- Raspberry Pi 4- AWR1642



- Raspberry Pi 4- Evo60M

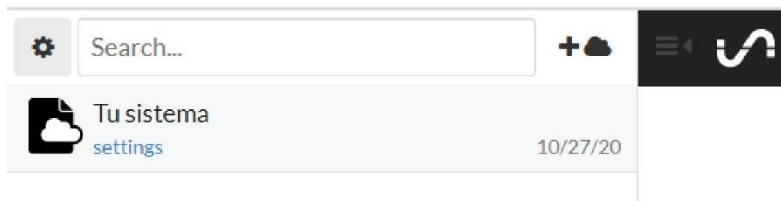


- Raspberry Pi 4- Modulo 4G



Manual de usuario

1. Seleccione el área en el que desea colocar el dispositivo.
Nota: Asegúrese que sea una superficie plana además de estar apuntando al parabrisas trasero y sin ningún obstáculo para los sensores o el radar.
2. Conecte el inversor de corriente al automóvil, y posteriormente conecte el dispositivo a este, mediante el cable de alimentación.
Nota: Si deja el dispositivo conectado mientras el vehículo esta apagado este consumirá la energía de la batería. En caso de tener la versión con PowerBank asegúrese de tenerla bien cargada y conectada al dispositivo mediante la entrada USB tipo C.
3. Seleccione el área donde desea colocar la antena, recuerde que esta tiene un imán y se puede adherir a superficies metálicas.
4. Desde un smartphone, Tablet o PC entre al link: <https://www.initialstate.com/> e inicie sesión con el usuario y contraseña suministrado.
5. Una vez allí encontraras tu dispositivo y podrás monitorizar tu automóvil.



Referencias

- [1] Mactronica, «Mactronica,» [En línea]. Available: <https://www.mactronica.com.co/>. [Último acceso: 31 10 2020].
- [2] T. Instruments, «Texas Instruments,» [En línea]. Available: <https://www.ti.com/>. [Último acceso: 31 10 2020].
- [3] TERABEE, «TERABEE,» [En línea]. Available: <https://www.terabee.com/>. [Último acceso: 31 10 2020].
- [4] Sotel, «Sotel,» [En línea]. Available: <https://www.sotel.de/es/initio/>. [Último acceso: 31 10 2020].
- [5] Ferretrónica, «Ferretrónica,» [En línea]. Available: <https://ferretronica.com/>. [Último acceso: 31 10 2020].