

Universidad Central de Venezuela  
Facultad de Ciencias  
Escuela de Computación  
Laboratorio de Comunicación y Redes

**Diseño e Implementación de un  
Dispositivo que Soporta la  
Comunicación entre Vehículos  
Conectados vía una VANET**

Trabajo Especial de Grado  
presentado ante la Ilustre  
Universidad Central de Venezuela  
por los Bachilleres:

Jackson Moreno  
C.I.: 17.076.008  
e-mail: jackson.morenop@gmail.com

José Herrera  
C.I.: 19.123.650  
e-mail: herrera.c.jose.m@gmail.com

para optar al título de Licenciado en Computación

Tutora: Profesora María Elena Villapol

Caracas, Octubre 2013

# Universidad Central de Venezuela

Facultad de Ciencias

Escuela de Computación

Laboratorio de Comunicación y Redes



## ACTA DEL VEREDICTO

Quienes suscriben, Miembros del Jurado designado por el Consejo de la Escuela de Computación para examinar el Trabajo Especial de Grado, presentado por los Bachilleres Jackson Moreno C.I.: 17.076.008 y José Herrera C.I.: 19.123.650, con el título **“Diseño e Implementación de un Dispositivo que Soporta la Comunicación entre Vehículos conectados vía una VANET”**, a los fines de cumplir con el requisito legal para optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído el trabajo por cada uno de los Miembros del Jurado, se fijó el día 23 de Octubre de 2013, a las 2:30 pm, para que sus autores lo defendieran en forma pública, en la Sala PAIII de la escuela de Computación UCV, lo cual estos realizaron mediante una exposición oral de su contenido, y luego respondieron satisfactoriamente a las preguntas que les fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de lo cual se levanta la presente acta, en Caracas el 23 de Octubre de 2013 dejándose también constancia de que actuó como Coordinador del Jurado el Profesor Tutor María Elena Villapol.

---

Prof. David Pérez  
(Jurado Principal)

---

Prof. María Elena Villapol  
(Tutora)

---

Prof. Carlos Acosta  
(Jurado Principal)

## Tabla de Contenido

Índice de Figuras .....	6
Índice de Tablas.....	8
1. Introducción .....	10
1.1. Planteamiento del Problema .....	10
1.2. Objetivo General .....	11
1.3. Objetivos Específicos .....	11
1.4. Justificación.....	12
1.5. Distribución del documento .....	12
2. Redes Ad Hoc Vehiculares .....	13
2.1. Definición de VANET.....	13
2.1.1. Características de las Redes VANETs.....	14
2.1.2. Servicios y Aplicaciones de Redes VANETs.....	14
2.1.3. Enrutamiento en Redes VANETs.....	16
2.1.4. Protocolos de Enrutamiento.....	17
2.2. Estándar IEEE 802.11p.....	22
2.3. Pila de Protocolos para VANET.....	23
2.3.1. WAVE.....	24
2.3.2. CALM .....	24
2.3.3. C2C-CC.....	24
3. Metodología y Herramientas de Desarrollo .....	26
3.1. Fases de la Metodología .....	26
3.1.1. Análisis .....	26
3.1.2. Diseño .....	26
3.1.3. Implementación .....	26
3.1.4. Pruebas .....	27
3.2. Herramientas de Desarrollo.....	27

3.2.1.	GCDC (Grand Cooperative Driving Challenge) .....	27
3.2.2.	Compat Wireless .....	28
3.2.3.	Debian Live 6.0.....	28
3.2.4.	Cargeo6.....	28
3.2.5.	Java.....	29
3.2.6.	Unetbootin .....	30
3.2.7.	Downloadosmtiles.pl.....	30
4.	Análisis y Diseño del Dispositivo .....	31
4.1.	OBU .....	31
4.2.	Características de la OBU .....	31
4.3.	Requerimientos funcionales del prototipo.....	32
4.4.	Requerimientos no funcionales del prototipo .....	32
4.5.	Diseño del equipo .....	33
4.5.1.	Diseño de hardware.....	33
4.5.2.	Diseño de software .....	34
4.5.3.	Aplicación Icaro-vanet .....	34
5.	Implementación del Dispositivo .....	39
5.1.	Selección de los componentes de hardware .....	39
5.1.1.	Módulo principal .....	39
5.1.2.	Módulo de alimentación .....	40
5.1.3.	Interfaz Inalámbrica .....	40
5.1.4.	GPS.....	41
5.1.5.	Pantalla Táctil .....	41
5.1.6.	Chasis del dispositivo .....	42
5.2.	Ensamblaje del equipo .....	42
5.3.	Instalación del sistema operativo Debian Live 6.0 .....	43

5.4.	Instalación y configuración de la pantalla táctil .....	44
5.5.	Instalación del software GCDC.....	44
5.6.	Instalación de Cargeo6 .....	45
5.7.	Aplicación Icaro-vanet.....	46
6.	Pruebas y Análisis de Resultados.....	48
6.1.	Pruebas fijas .....	48
6.1.1.	Escenarios de prueba.....	48
6.1.2.	Resultados de las pruebas fijas .....	51
6.1.3.	Análisis de resultados de las pruebas fijas .....	55
6.2.	Pruebas de movilidad.....	56
6.2.1.	Resultados de las pruebas de movilidad.....	57
6.2.2.	Análisis de resultados de las pruebas de movilidad.....	58
6.3.	Prueba de cobertura.....	60
6.3.1.	Resultados .....	60
7.	Conclusiones .....	61
7.1.	Contribuciones .....	61
7.2.	Limitaciones .....	62
7.3.	Trabajos futuros .....	62
	Referencias.....	64

## Índice de Figuras

Figura 2. 1: Arquitectura VANET.....	15
Figura 2. 2: Taxonomía de los protocolos de enrutamiento en redes VANETs.....	18
Figura 2. 3: Arquitecturas VANET.....	23
Figura 3. 1: Arquitectura de Cargeo6.....	29
Figura 4. 1: Módulos de hardware.....	33
Figura 4. 2: Módulos de software.....	34
Figura 4. 3: Diagrama de casos de uso, nivel 0.....	35
Figura 4. 4: Diagrama de casos de uso, nivel 1.....	35
Figura 4. 5: Diagrama de caso de uso "Ver Información del vehículo", nivel 2.....	36
Figura 4. 6: Diagrama de caso de uso "Enviar mensaje", nivel 2.....	37
Figura 4. 7: Diagrama de caso de uso "Recibir mensaje", nivel 2.....	38
Figura 5. 1: Módulo principal.....	39
Figura 5. 2: Módulo de alimentación.....	40
Figura 5. 3: Interfaz Inalámbrica.....	40
Figura 5. 4: GPS.....	41
Figura 5. 5: Pantalla táctil.....	41
Figura 5. 6: Chasis del dispositivo.....	42
Figura 5. 7: Módulo principal ensamblado.....	43
Figura 5. 8: Dispositivo construido.....	43
Figura 5. 9: Formato de mensajes en la aplicación Icaro-vanet.....	46
Figura 5. 10: Interfaz de la aplicación Icaro-vanet.....	47
Figura 6. 1: Escenarios de pruebas fijas.....	48

Figura 6. 2: Escenario de comunicación en la topología 1. ....	50
Figura 6. 3: Escenario de comunicación en la topología 2 .....	50
Figura 6. 4: Escenario de comunicación en la topología 3. ....	51
Figura 6. 5: Captura en la interfaz tun0 de OBU3 correspondiente a la topología 1. ....	52
Figura 6. 6: Captura en la interfaz wlan0 de OBU2 correspondiente a la topología 1.....	52
Figura 6. 7: Listas de vecinos de OBU1, OBU2 y OBU3 en la topología 1. ....	52
Figura 6. 8: Captura en la interfaz tun0 de OBU1 correspondiente a la topología 2. ....	53
Figura 6. 9: Captura en la interfaz wlan0 de OBU2. ....	53
Figura 6. 10: Listas de vecinos de OBU1, OBU2, OBU3 y OBU4 en la topología 2.....	54
Figura 6. 11: Captura en la interfaz tun0 de OBU1 correspondiente a la topología 3. ....	54
Figura 6. 12: Captura en la interfaz wlan0 de OBU2. ....	55
Figura 6. 13: Listas de vecinos de OBU1, OBU2, OBU3 y OBU4 en la topología 3.....	55
Figura 6. 14: Escenario para las pruebas de movilidad. ....	56
Figura 6. 15: Comparación del tiempo promedio que tardan los mensajes en llegar al destino.	58
Figura 6. 16: Relación entre mensajes recibidos y perdidos por la aplicación. ....	59
Figura 6. 17: Mensajes de accidente y congestión en la trayectoria del vehículo. ....	59

## Índice de Tablas

Tabla 4. 1: Especificación de caso de uso "Ver información del vehículo".....	36
Tabla 4. 2: Especificación de caso de uso "Enviar mensaje".....	36
Tabla 4. 3: Especificación de caso de uso "Recibir mensaje".....	36
Tabla 4. 4: Especificación de caso de uso "Ver ID".....	37
Tabla 4. 5: Especificación de caso de uso "Ver Latitud".....	37
Tabla 4. 6: Especificación de caso de uso "Ver Longitud".....	37
Tabla 4. 7: Especificación de caso de uso "Ver Velocidad".....	37
Tabla 4. 8: Especificación de caso de uso "Ver ubicación en el mapa".....	37
Tabla 4. 9: Especificación de caso de uso "Enviar mensaje de congestión".....	38
Tabla 4. 10: Especificación de caso de uso "Enviar mensaje de accidente".....	38
Tabla 4. 11: Especificación de caso de uso "Recibir mensaje de congestión".....	38
Tabla 4. 12: Especificación de caso de uso "Recibir mensaje de accidente".....	38
Tabla 6. 1: Configuraciones de red en la topología 1.....	49
Tabla 6. 2: Configuraciones de red en la topología 2.....	50
Tabla 6. 3: Mensajes de accidente y congestión a 20Km/H, 40Km/H y 55 Km/H.....	57
Tabla 6. 4: Cobertura de la señal.....	60

## Resumen

**Título:** Diseño e Implementación de un Dispositivo que Soporta la Comunicación entre Vehículos Conectados vía una VANET.

**Autores:** Jackson A. Moreno P. José M. Herrera C.

**Tutora:** Profa. María Elena Villapol.

El presente Trabajo Especial de Grado consiste en el desarrollo de un dispositivo que cumple con los requerimientos necesarios para ser ubicado a bordo de un vehículo, con soporte para la comunicación entre vehículos conectados vía una VANET. El dispositivo es una computadora de dimensiones reducidas compuesta de una tarjeta madre mini ITX Intel D525MW, una fuente de poder picoPSU-160-XT, una tarjeta de red inalámbrica Belkin F6D3000-HP-NA con chipset Atheros AR5414, un receptor GPS Globalsat BU-353, una pantalla táctil LCD VM70 7", una memoria flash USB y un chasis elaborado en el proyecto tomando en cuenta las dimensiones de los componentes de hardware. Entre los componentes de software destaca el sistema operativo Debian Live 6.0 con las configuraciones necesarias para el correcto funcionamiento de los componentes de hardware y la modificación del controlador de la tarjeta de red inalámbrica *ath5k* para habilitar las frecuencias del estándar IEEE 802.11p. En la capa de software también se incluye un protocolo de enrutamiento y la implementación de la aplicación Icaro-vanet, para el envío y recepción de mensajes de alerta por parte del conductor mediante el uso de la pantalla táctil.

El trabajo se inicia con un estudio teórico de las características más relevantes de las redes VANETs. Los trabajos en redes VANETs se enfocan en alcanzar mejoras en el rendimiento de los sistemas de transporte, con la finalidad de lograr seguridad en las carreteras y reducir la congestión de tráfico vehicular. Muchos de estos trabajos en redes VANETs se basan en simulaciones y estudios teóricos sobre arquitecturas y protocolos para este tipo de redes, patrones de movilidad de los nodos, propagación de las señales de radio, etc. Sin embargo, como producto de este trabajo se realizaron pruebas en ambientes reales para evaluar en la práctica algunas de estas características.

Palabras claves: VANET, 802.11p, protocolo de enrutamiento, tráfico vehicular.

## 1. Introducción

Los recientes avances en redes inalámbricas han llevado al surgimiento de un nuevo tipo de red conocida como *Red Ad Hoc Vehicular (Vehicular Ad hoc Network, VANET)*. Las redes VANETs son una especialización de las *redes móviles Ad Hoc (Mobile Ad Hoc Networks, MANET)*, donde los nodos móviles son vehículos equipados con dispositivos de comunicación inalámbrica, con el fin de implementar nuevos sistemas para mejorar la seguridad vial.

Estas redes se han desarrollado para realizar mejoras en el rendimiento de los sistemas de transporte convencionales, dando origen a lo que se conoce como *Sistemas de Transporte Inteligentes (Intelligent Transportation Systems, ITS)*, que tienen como objetivo, mejorar la seguridad en las carreteras y reducir la congestión de tráfico. Es por esto, que la integración de computadoras embebidas, sensores, *Sistemas de Posicionamiento Global (Global Positioning System, GPS)*, mapas digitales, y dispositivos de comunicación inalámbrica, junto con algoritmos inteligentes ayudará a desarrollar muchos tipos de aplicaciones para los ITS y así mejorar la seguridad en las carreteras. Razón por la cual, las redes VANETs conllevan a la investigación sobre protocolos de enrutamiento e implementación de dispositivos que permitan el desarrollo de las aplicaciones enfocadas a este fin.

### 1.1. Planteamiento del Problema

Muy ligado al crecimiento de las grandes ciudades, por ejemplo Caracas, está el crecimiento del transporte urbano y las necesidades de transporte público. El transporte público superficial del Distrito Metropolitano de Caracas está constituido por autobuses, minibuses y rústicos operados en su mayoría por organizaciones privadas tales como asociaciones civiles, cooperativas y compañías, operando en decenas de rutas. Actualmente, el sistema de transporte superficial trabaja de forma desarticulada, carece de control y fiscalización, consume tiempos de recorrido muy largos debido al gran volumen de tráfico vehicular. Las unidades de transporte no cuentan ni con dispositivos de comunicación y ni con una red de comunicación articulada que permitan intercambiar información, tal como ubicación del vehículo, condiciones de tráfico, condiciones viales, condiciones del tiempo, accidentes, entre otros, entre ellos y con sus supervisores. Actualmente, algunas unidades de transporte como las de taxis, agrupadas en líneas, han adquirido unidades de radio que permiten el intercambio de información de voz entre miembros de dicha línea. En su mayoría los choferes de unidades de autobuses,

minibuses y taxis cuentan con teléfono celulares de uso personal, que le permiten reportar algún incidente con el vehículo. Algunos dueños de unidades han integrado en sus vehículos sistemas de localización basados en GPS. Sin embargo, estas soluciones son aisladas y no contribuyen a aliviar el problema del sistema de transporte de forma global. Además, estas soluciones implican un gasto en equipos y en servicios que en muchos casos es trasladado a los usuarios del servicio.

La creación de una red de comunicación inalámbrica que permita comunicar las unidades de transporte público del Distrito Capital sería un paso inicial para lograr una mejor articulación entre las unidades de este sistema a la vez que permitiría atacar ciertos problemas relacionados al monitoreo de estas unidades, rastreos en caso de incidentes de seguridad, recepción de alertas, entre otros.

## **1.2. Objetivo General**

Desarrollar un prototipo de dispositivo móvil que disponga de los componentes de hardware y software necesarios para el uso de aplicaciones en redes VANETs.

## **1.3. Objetivos Específicos**

Con este trabajo se pretenden alcanzar los siguientes objetivos específicos:

- Identificar las funciones y características que deben estar presentes en un dispositivo para redes VANETs.
- Realizar el diseño del dispositivo a nivel de hardware y software.
- Seleccionar las tecnologías y herramientas.
- Ensamblar el prototipo de dispositivo mediante la integración de componentes de hardware y software.
- Realizar pruebas para verificar el correcto funcionamiento del dispositivo.
- Validar el funcionamiento de un protocolo de enrutamiento para redes VANETs.

## **1.4. Justificación**

Los trabajos en redes VANETs se enfocan en alcanzar mejoras en el rendimiento de los sistemas de transporte, con la finalidad de lograr seguridad en las carreteras y reducir la congestión de tráfico vehicular. Los estudios que se hacen cada vez más populares contemplan reportes de tráfico, advertencia de obstáculos, notificación de accidentes, entre otros. Para ofrecer a los conductores información en tiempo real que les permita reaccionar según sea el caso.

La integración de distintas tecnologías de comunicación inalámbrica, sistemas de posicionamiento global y otro tipo de sensores, mediante la elaboración de un dispositivo que pueda ser ubicado a bordo de los vehículos, ayudará en la investigación y soporte de aplicaciones para la seguridad vial, que puedan ser evaluadas en condiciones y escenarios reales.

## **1.5. Distribución del documento**

En el presente documento se describen los aspectos más importantes del Trabajo Especial de Grado donde se describe el desarrollo de un dispositivo que cumpla con las características y tecnologías necesarias para la comunicación entre vehículos vía una VANET.

El documento se encuentra estructurado en siete capítulos. En el Capítulo 2 se introducen conceptos, características, protocolos de enrutamiento, pilas de protocolos y otros detalles de las redes VANETs, como el estándar IEEE 802.11p. En el Capítulo 3 se describe la metodología, tecnologías y herramientas que se utilizaron para el desarrollo del dispositivo. El capítulo 4 presenta el estudio de los requerimientos del dispositivo, así como la definición de los componentes de hardware y software necesarios para el correcto funcionamiento de los módulos del equipo. Seguidamente, el capítulo 5 describe los detalles del proceso llevado a cabo durante la construcción y configuración del dispositivo. En el siguiente capítulo se presentan los tipos de pruebas que se llevaron a cabo para validar el funcionamiento del dispositivo, y finalmente, en el capítulo 7 se presentan las conclusiones del trabajo.

## 2. Redes Ad Hoc Vehiculares

Una de las tecnologías más prometedoras y discutidas en esta década es poder comunicar dispositivos de forma inalámbrica, lo que permite una alta movilidad. Entre las áreas de investigación más activas de los últimos años se encuentran la de las redes VANETs, con el objetivo de desarrollar plataformas de comunicación entre vehículos en movimiento y entre éstos con una infraestructura vial, aprovechando las capacidades de cómputo y comunicaciones inalámbricas.

### 2.1. Definición de VANET

Las *Redes Ad Hoc Vehiculares (Vehicular Ad hoc NETWORK, VANET)* son una especialización de las *Redes Ad Hoc Móviles (Mobile Ad-Hoc Network, MANET)*, es decir, una red *Ad Hoc* móvil donde sus nodos se corresponden con vehículos y éstos forman dicha red en pleno movimiento. El enrutamiento de los paquetes es responsabilidad de los nodos que forman parte de la red y éstas no tienen una infraestructura fija.

Existe muchas aplicaciones para este tipo de redes, una de las más importantes es la transmisión de mensajes de alerta referente a accidentes o situaciones peligrosas en general. Otra aplicación interesante es la información de servicios públicos (restaurantes cercanos, hoteles, entre otros servicios) a los conductores. Además, pueden proporcionar acceso a Internet para descargas de contenido multimedia o chat entre vehículos.

Este tipo de redes se ha desarrollado como parte de los *Sistemas de Transporte Inteligentes (Intelligent Transport Systems, ITS)* para realizar mejoras en el rendimiento de los sistemas de transporte convencionales. Uno de los objetivos principales de los ITS es mejorar la seguridad en las carreteras, y reducir congestión de tráfico. La integración de computadoras embebidas, dispositivos sensores, GPS, mapas digitales, y los dispositivos de comunicación inalámbrica junto con algoritmos inteligentes ayudará a desarrollar muchos tipos de aplicaciones para los ITS y así mejorar la seguridad en las carreteras.

La integración de estos sistemas ayuda a los conductores a adquirir información en tiempo real sobre las condiciones de tráfico, permitiéndoles reaccionar a tiempo. Por ejemplo, mensajes de alerta enviados por los vehículos involucrados en un accidente de tráfico,

aumenta la seguridad al ayudar a los conductores que se acercan a tomar las decisiones adecuadas antes de entrar a la zona del accidente.

### **2.1.1. Características de las Redes VANETs**

Las redes VANETs tienen características únicas que las distinguen de las MANETs. Estas características pueden resumirse de la siguiente manera:

- **Alta movilidad:** Los nodos de las redes VANETs se caracterizan por su alta velocidad relativa, que hace a un entorno VANET de alta dinámica.
- **Patrones de movilidad restringida y predecible:** A diferencia de la movilidad al azar en las redes MANETs, los movimientos de los nodos en las redes VANETs se rigen por reglas restringidas (reglas de teoría de flujo de tráfico), lo que los hace predecible, al menos en el corto plazo.
- **Rápido cambio en la topología:** Los nodos de una VANET se caracterizan por su alta velocidad. Esto lleva a frecuentes cambios en la topología de la red, que introduce alta sobrecarga de comunicación para el intercambio de la información de la topología.
- **Localización:** Los vehículos pueden usar GPS para identificar su ubicación con gran precisión.

### **2.1.2. Servicios y Aplicaciones de Redes VANETs**

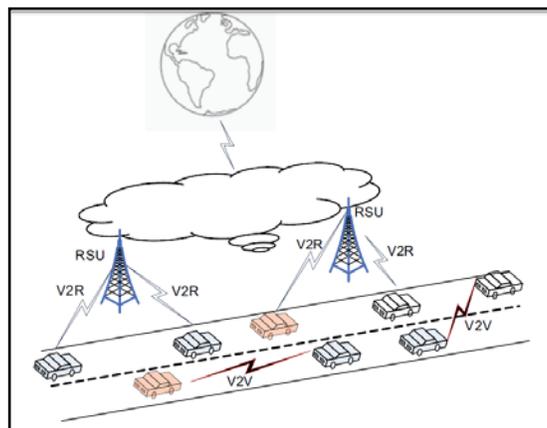
Las redes vehiculares están compuestas de nodos móviles, vehículos equipados con una *unidad a bordo (On-Board Unit, OBU)* y nodos fijos llamados *unidad en carretera (Road-Side Unit, RSU)* unidos a la infraestructura que se despliega a lo largo de la carretera. Las OBUs se comunican entre sí y con las RSUs de una manera *Ad Hoc*. Principalmente hay dos tipos de escenarios de comunicación en una VANET: *vehículo a vehículo (Vehicle-to-Vehicle, V2V)* y *vehículo a unidad en carretera (Vehicle-to-RSU, V2R)*. Las RSUs también pueden comunicarse entre sí y a través de otras redes como Internet.

En las redes vehiculares se espera que se empleen distintas tecnologías inalámbricas avanzadas, por ejemplo, las *comunicaciones dedicadas de corto alcance (Dedicated Short Range Communications, DSRC)*, que es una versión mejorada de la tecnología WiFi adecuada

para entornos VANET. Las DSRC se han desarrollado para soportar la transferencia de datos en ambientes de comunicación que cambian rápidamente, como es el caso de una VANET, donde las respuestas en tiempo crítico y altas velocidades se requieren. La tecnología DSRC se utilizará en el dominio ITS para proporcionar enlaces de comunicación segura y fiable entre los vehículos y entre los vehículos e infraestructura. En la Figura 2. 1 tomada de [01] se muestra un ejemplo de la Arquitectura VANET.

Los servicios y aplicaciones de una VANET difieren significativamente de los clásicos conocidos en una MANET. Posiblemente el grupo más importante de servicios, lo que hace que los estudios de investigación en redes VANETs sean cada vez más populares, son los relacionados con la seguridad en la conducción. Las aplicaciones de seguridad incluyen entre otros: Reportes de tráfico actual, advertencia de obstáculos en caminos (accidentes, trabajos y otras situaciones inusuales), etc.

Otro grupo de aplicaciones que pueden ganar popularidad entre los usuarios de carretera son los servicios de información y entretenimiento. Los más básicos están relacionados con la publicidad, como por ejemplo, distribución de información de habitaciones libres en hoteles, ofertas en restaurantes, descuentos en tiendas, precio de la gasolina, información sobre plazas de estacionamiento libres, estación de servicio más cercana, etc. Algunos de los servicios de información y entretenimiento pueden ser útiles a los peatones, por ejemplo, los autobuses pueden estimar el momento de la llegada a la parada de autobús con el conocimiento de las condiciones del tráfico y distribuir esta forma, información a los pasajeros que esperan [01].



**Figura 2. 1: Arquitectura VANET.**

Numerosas aplicaciones habilitadas por la tecnología DSRC se han propuesto para las redes VANETs. Estas aplicaciones son clasificadas como aplicaciones de seguridad y de no seguridad, las cuales tienen distintos requerimientos. A los mensajes de seguridad se les da mayor prioridad que a los datos que no son de seguridad. Los mensajes de seguridad son sensibles al tiempo y deben ser difundidos a los vehículos en los alrededores del evento dentro de un tiempo limitado, son de naturaleza *broadcast*, por lo tanto, una estrategia de difusión inteligente debe ser empleada para garantizar la entrega rápida de estos mensajes. Estos mensajes pueden ser periódicos o basados en eventos. Los mensajes de seguridad basados en eventos son generados y enviados por los vehículos involucrados en un accidente para alertar a los vehículos que se acercan al área donde se encuentra el siniestro. Por otro lado, los mensajes de seguridad periódicos son considerados métodos preventivos de seguridad enviados a intervalos de tiempo específicos, que llevan información de estado actual de los nodos, como velocidad de los vehículos, aceleración, dirección, etc. Esta información es usada por los vehículos en la zona para actualizar el estado de sus vecinos. Los mensajes de seguridad periódicos también pueden ser enviados por las RSUs, por ejemplo, algunas instaladas en las intersecciones, envían periódicamente mensajes acerca de las condiciones en ese lugar.

Las aplicaciones de no seguridad tienen objetivos diferentes y pueden ser usadas para proporcionar una serie de servicios que van desde administración de transporte, cobro de peajes, información y entretenimiento, descarga de música y anuncios comerciales. Los datos de no seguridad tienen prioridad baja en comparación de los datos de seguridad.

Las DSRC también son compatibles con diferentes protocolos de red, lo que les da la capacidad de interactuar con distintos tipos de redes. Las DSRC soportan TCP/IP, por lo tanto, las aplicaciones de Internet también pueden estar disponibles en redes VANETs.

### **2.1.3. Enrutamiento en Redes VANETs**

A diferencia de los protocolos de enrutamiento implementados en una MANET, los protocolos de enrutamiento en el entorno VANET deben hacer frente a los siguientes desafíos:

- Topología muy dinámica: Las redes VANETs son formadas y sostenidas de una manera *Ad Hoc* con vehículos uniéndose y dejando la red todo el tiempo, a veces sólo están en el rango de unos pocos segundos.
- Particiones de red: En áreas rurales el tráfico puede llegar a ser tan disperso que crean particiones de red separadas.
- Transmisiones sensibles al tiempo: Las advertencias de seguridad deben ser transmitidas lo más rápido posible y deben ser de alta prioridad, respecto a los datos regulares.

Aplicar los protocolos de enrutamiento tradicionales de redes MANETs directamente en redes VANETs puede resultar ineficiente, ya que estos métodos no toman en cuenta las características de una VANET. Por lo tanto, la modificación de los protocolos de enrutamiento para redes MANETs o el desarrollo de nuevos protocolos de enrutamiento específicos para redes VANETs son los enfoques prácticos para el uso eficiente de métodos de enrutamiento. Un ejemplo de la modificación de protocolos MANET para trabajar en entornos VANET es la modificación del protocolo AODV (*Ad Hoc On Demand Distance Vector*) con PGB (*Preferred Group Broadcasting*). Por otro lado, nuevos protocolos de enrutamiento se han desarrollado específicamente para redes VANETs. Estos protocolos están basados en la posición de los nodos, y toman ventaja del conocimiento de mapas, carreteras, velocidad, y posición actual del vehículo.

#### **2.1.4. Protocolos de Enrutamiento**

En esta sección se presenta una breve descripción de los protocolos de enrutamiento propuestos o adaptados para las redes VANETs. De acuerdo al tipo de información usada para tomar las decisiones de enrutamiento, estos protocolos pueden ser clasificados en cinco categorías como se muestra en la Figura 2.2 tomada de [01].

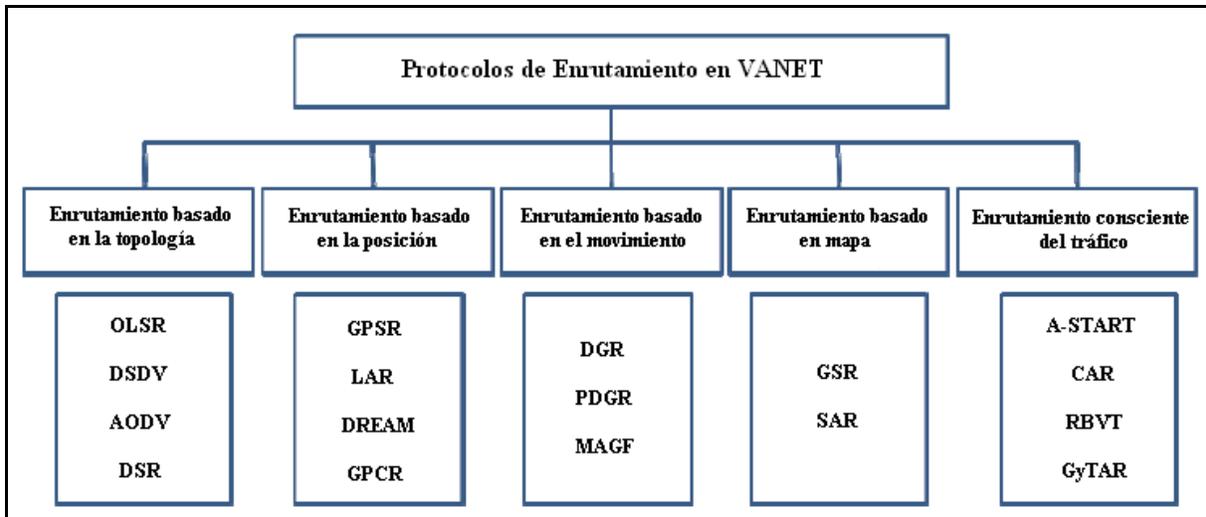


Figura 2. 2: Taxonomía de los protocolos de enrutamiento en redes VANETs.

### ***Enrutamiento Basado en la Topología***

Los protocolos basados en la topología usan información sobre la topología de la red y el estado de los enlaces de comunicación entre los nodos para llevar a cabo las decisiones de enrutamiento. Ellos pueden clasificarse en los enfoques proactivos y reactivos.

Los protocolos proactivos, tales como OLSR (*Optimized Link State Routing Protocol*) y DSDV (*Destination-Sequenced Distance Vector*), calculan y mantienen la información de enrutamiento de todas las rutas disponibles en las redes, incluso si no hay tráfico de datos, se intercambia esta información. Por ejemplo, en DSDV, cada nodo mantiene un vector de distancia para cada nodo conocido. Por lo tanto, frecuentes mensajes *broadcast* son emitidos por todos los nodos periódicamente para aprender acerca de sus vecinos de un salto o para anunciar cambios en la topología (por ejemplo enlaces rotos). Del mismo modo, OLSR inunda la red por los mensajes de control de topología con el fin de difundir la información del estado del enlace a lo largo de toda la red, mostrando cuales nodos están conectados a los demás nodos.

Este tráfico adicional usado en los enfoques proactivos para el mantenimiento de las rutas no utilizadas tiene varias desventajas. En primer lugar, que consume los recursos de la red y desecha una parte del ancho de banda para los mensajes de control que se incrementan con los cambios rápidos. Además, el uso de las inundaciones incrementa la congestión en la

red y conduce a la pérdida de los mensajes debido a la colisión. Claramente, las soluciones proactivas no se adaptan bien en redes muy grandes con un alto número de nodos que entran y salen de la red durante un corto periodo de tiempo, que es el caso de las redes VANETs.

Por el contrario, los protocolos reactivos tales como AODV y DSR (*Dynamic Source Routing*) determinan una ruta a un destino determinado sólo bajo demanda. Ellos reducen la sobrecarga por restringir el mantenimiento de la única ruta entre los nodos que necesitan comunicarse. En otras palabras, el descubrimiento de la ruta sólo se inicia cuando un nodo emisor tiene que establecer una ruta válida a un destino determinado. Obviamente, esto extiende el retardo antes de que los paquetes puedan ser enviados en realidad a través de la red. Además, la mayoría de los protocolos reactivos usan la técnica de inundación para establecer la comunicación entre el origen y el destino y por tanto consumen gran cantidad de ancho de banda.

Debido a la alta movilidad de los vehículos, los algoritmos basados en la topología fallan al manejar frecuentes rutas caídas usualmente construidas como una sucesión de vehículos entre la fuente y el destino. Por otro lado, la inestabilidad de rutas y los frecuentes cambios de topología aumentan la sobrecarga para reparaciones de ruta, notificaciones de cambio y por lo tanto, se degrada el rendimiento del enrutamiento.

Generalmente, en el enfoque de enrutamiento basado en topología, las rutas se construyen como sucesiones de los nodos móviles, y por lo tanto la posibilidad de perder la conectividad es más alta.

### ***Enrutamiento Basado en la Posición***

Los protocolos basados en la posición realizan las decisiones de enrutamiento basado en la información geográfica de los nodos. Esta clase ofrece un enfoque alternativo conocido por ser más robusta para enfrentar los problemas de movilidad. De hecho, ningún conocimiento global de la topología de red es necesario; una decisión puramente local es hecha por cada nodo para hacer un mejor progreso hacia el destino. Por lo tanto, requieren que todos los nodos sean conscientes de sus posiciones físicas, así como las posiciones de sus vecinos. También se supone que el nodo emisor conoce la posición del destino. Por lo general, un servicio de gestión de localización es responsable de consultar esta información.

Como ejemplo representativo de los algoritmos basados en la posición, GPSR (*Greedy Perimeter Stateless Routing*) parece ser el candidato más popular para redes dinámicas. Por lo general hay una serie de requerimientos en la disponibilidad de información de posición. GPSR requiere que cada nodo sea capaz de obtener su ubicación actual, por ejemplo, a través de un receptor GPS, que ya se está convirtiendo en equipo estándar en los vehículos. Por otra parte, se asume que cada nodo aprende de la existencia de sus vecinos directos mediante el intercambio de mensajes periódicos *hello*. Para tomar las decisiones de enrutamiento, un nodo origen necesita saber la posición del destino. El nodo fuente envía los paquetes a su vecino que está geográficamente más cercano a su destino. Este procedimiento se aplica de forma recursiva por los nodos intermedios hasta que el destino final se alcanza.

### ***Enrutamiento Basado en el Movimiento***

Numerosos protocolos mejoran el esquema básico basado en la posición para optimizar las decisiones de enrutamiento. Algunos enfoques como DGR (*Directional Greedy Forwarding*) y MAGF (*Movement Aware Greedy Forwarding*) sugieren hacer uso de la información adicional sobre el movimiento de vehículos, tales como dirección y velocidad. La idea básica es calcular una puntuación ponderada  $W_i$  como una función de diferentes factores (posición, dirección, velocidad) para asignar prioridad entre los nodos vecinos durante la selección del siguiente retransmisor. Esta mejora del esquema basado en posición reduce el número máximo local de encuentros evitando el envío de paquetes fuera de su destino durante la selección de una dirección equivocada.

Considerando que los vehículos siguen un patrón de movilidad predecible, los autores proponen PDGR (*Predictive Directional Greedy Routing*), para reenviar un paquete al siguiente salto más adaptable basado en ubicación actual y ubicación futura. El esquema de predicción de movilidad permite una retransmisión de paquetes para asegurar la validez de un vecino seleccionado.

### ***Enrutamiento Basado en Mapa***

Los protocolos de enrutamientos basados en mapas combinan la información de posición con el conocimiento topológico de la carretera y los alrededores. La idea es construir un modelo espacial que representa la topología de la carretera subyacente y selecciona una ruta de acceso que se superpone a las calles. Para este propósito, los mapas de carretera

están representados por las gráficas donde los vértices son los cruces y los bordes son segmentos de ruta. Comúnmente, los bordes del grafo son ponderados con los datos estáticos extraídos de los mapas de calles. Ejemplos de estos datos estáticos podrían ser la distancia o los límites de velocidad. En consecuencia, el camino de enrutamiento es seleccionado basado en el nuevo grafo construido y los paquetes de datos sólo son enviados respetando el patrón de movilidad restringida, en particular por la topología de la carretera.

El primer protocolo en usar el conocimiento del mapa subyacente de las calles fue GSR (*Global State Routing*), el cual es propuesto principalmente para entornos urbanos. Suponiendo que la disponibilidad de dicha información se obtiene a través de un sistema de navegación, una fuente dada calcula la ruta más corta a un destino usando el algoritmo de *Dijkstra* basado en la métrica de distancia. El camino calculado consiste de una secuencia de identificadores unidos, conocidos como *puntos de anclaje*, a lo largo de los paquetes que deben enviarse para llegar al destino. Estos anclajes, que se obtienen en el mapa de calles, reflejan la topología de la carretera subyacente y suelen representar las intersecciones de carreteras donde se toman las decisiones.

SAR (*Spatial Aware Routing*) es introducido para mejorar el protocolo básico GSR. La evaluación de rendimiento ha demostrado que SAR es más robusto para la movilidad que los protocolos de enrutamiento basado en la topología (DSR), ya que la ruta es calculada independientemente de nodos móviles específicos.

### ***Enrutamiento Consciente del Tráfico***

Estos protocolos sugieren el uso de los datos disponibles sobre la densidad del tráfico vehicular y de los flujos, además de la información espacial. Por lo tanto, sólo las calles donde los vehículos se mueven serán utilizadas para el reenvío de paquetes.

Uno de los protocolos que explota la idea de conocimiento de tráfico en el diseño de un esquema de enrutamiento es el A-START (*Anchor-based Street and Traffic Aware Routing*). Generalmente, en ambientes de ciudad, los vehículos están más concentrados en unas áreas que en otras, por lo tanto, la conectividad es alta en estas carreteras.

El protocolo CAR (*Connectivity Aware Routing*) usa los datos estadísticos recogidos por diferentes vehículos para estimar la probabilidad de la conectividad de cada segmento de la

carretera. La información de conectividad se difunde en toda la red para proporcionar una visión global acerca de la conectividad de red.

El protocolo RBVT (*Road-based with vehicular traffic*) incorpora tráfico vehicular en tiempo real para calcular rutas. Tiene dos variantes: un protocolo reactivo RBVT-R (*RBVT-Reactive*), y un protocolo proactivo RBVT-P (*RBVT-Proactive*). En RBVT-R, los nodos fuentes descubren los segmentos de carretera conectado bajo demanda mediante el inicio de los paquetes de descubrimiento de ruta que atraviesan la red hacia el destino. Por otro lado, RBVT-P mantiene un grafo de todos los segmentos de carretera conectados. Para descubrir la topología de red, los *paquetes de conectividad* (*Connectivity Packets, CP*) se generan periódicamente por varios vehículos seleccionados al azar en la red.

## 2.2. Estándar IEEE 802.11p

IEEE 802.11p, actualiza y expande el estándar IEEE 802.11a con modificaciones a nivel físico y MAC para mejorar su comportamiento en el entorno que representa el ambiente de redes vehiculares, y de esta forma ser utilizado por los ITS. Al igual que IEEE 802.11a, 802.11p utiliza la *multiplexación por división en frecuencias ortogonales* (*Orthogonal Frequency Division Multiplexing, OFDM*) pero con tasas de transmisión de 3, 4, 5, 6, 9, 12, 18, 24 y 27 Mbps en canales de 10 MHz, aunque puede alcanzar tasas hasta de 54 Mbps con canales de 20 MHz. Utiliza 52 sub-portadoras moduladas utilizando BPSK, QPSK, 16-QAM o 64-QAM, así como codificaciones con tasas de 1/2, 2/3, o 3/4 [02].

El estándar IEEE 802.11p permite el uso de la banda de 5.9 GHz (5.850-5.925 GHz), frecuencia utilizada como base en las *comunicaciones dedicadas de corto alcance* (*Dedicated Short Range Communication, DSRC*), permitiendo así comunicaciones inalámbricas entre vehículos que se desplacen a altas velocidades, y entre los vehículos con la infraestructura vial.

Respecto al manejo de canales, el estándar define siete canales no solapados de 10 MHz en la banda de los 5.9 GHz; seis *canales de servicio* (*Service Channel, SCH*) y un *canal de control* (*Control Channel, CCH*). El canal CCH se emplea para la transmisión en modo *broadcast* de mensajes de seguridad vial. SCH permite la transmisión de datos relacionados con seguridad, entretenimiento, administración remota y también para la transmisión de voz y video sobre IP.

IEEE 802.11p emplea *acceso múltiple por detección de portadora con prevención de colisiones (Carrier Sense Multiple Acces Collision Avoidance, CSMA/CA)*, que a la vez es la base de la función de coordinación de capa MAC conocida como *acceso al canal distribuido mejorado (Enhanced Distributed Channel Access, EDCA)*, inicialmente prevista en el estándar IEEE 802.11e. Se utiliza tanto en comunicaciones V2R como en V2V.

### 2.3. Pila de Protocolos para VANET

Los ITS se enfocan en abordar el problema de seguridad vial. Organismos como el *Instituto de Ingenieros Eléctricos y Electrónicos (Institute of Electrical and Electronics Engineers, IEEE)*, la *Organización Internacional de Normalización (International Organization for Standardization, ISO)* y el *Consortio de Comunicaciones Vehículo a Vehículo (Car to Car Communication Consortium, C2C-CC)* están trabajando en propuestas de arquitecturas ITS [03].

Se han propuesto un conjunto de estándares para comunicaciones vehiculares en ITS. Algunas de las arquitecturas para redes VANETs se observan en la Figura 2. 3. *Acceso Inalámbrico en Entornos Vehiculares (Wireless Access in Vehicular Environment, WAVE)* introducido por el IEEE, *Acceso a la Comunicación para Móviles Terrestres (Communication Access for Land Mobiles, CALM)* por la ISO, y C2C-CC respaldado por industrias europeas de automóviles. Estas arquitecturas pueden funcionar en la banda de frecuencias de 5.9 GHz.

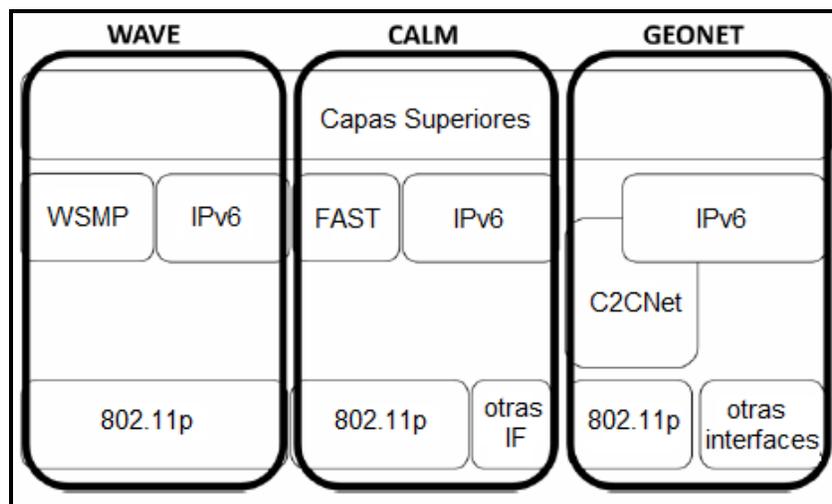


Figura 2. 3: Arquitecturas VANET.

### **2.3.1. WAVE**

IEEE introduce una pila de protocolos llamada WAVE basada en la familia de estándares IEEE 1609, donde se definen seis sub estándares, enumerados del uno al seis. Cada uno con diferentes propósitos en distintas capas [03].

El estándar 1609.1 detalla las actividades de gestión de recursos requeridas para las operaciones propias de las aplicaciones, definiendo el flujo de datos y recursos; así como también los formatos de los mensajes y almacenamiento de datos. 1609.2 describe las consideraciones tomadas en cuenta para la seguridad en las comunicaciones. 1609.3 define los servicios de la capa de red y transporte incluyendo direccionamiento y enrutamiento, a la vez que provee un protocolo llamado WSMP (*Wave Short Messages Protocol*), ofreciendo una eficiente alternativa para IP. 1609.4 define la coordinación entre los múltiples canales del espectro radioeléctrico. 1609.5 especifica los servicios de gestión en la comunicación inalámbrica entre los dispositivos OBUs y RSUs, para el acceso inalámbrico en entornos vehiculares. Por último, el estándar 1609.6 ofrece una capa intermedia entre la capa de transporte y de aplicación.

### **2.3.2. CALM**

CALM fue diseñada como una solución que permite comunicaciones continuas o casi continuas entre vehículos o con la infraestructura, y está siendo desarrollado por el grupo de trabajo ISO TC204. Proporciona un conjunto de protocolos y parámetros estandarizados para comunicaciones inalámbricas de medio y largo alcance de alta velocidad. También está basado en el concepto de DSCR 5.9 GHz [03].

La arquitectura cubre todas las capas del modelo de referencia OSI, como también todos los rangos de comunicación (corto, mediano y largo) y todos los tipos de aplicaciones como por ejemplo, las aplicaciones de seguridad en la vía, eficiencia de tráfico y entretenimiento. Los posibles protocolos de la capa de red son IPv6 y no-IP (FAST).

### **2.3.3. C2C-CC**

Es una organización sin fines de lucro iniciada por fabricantes europeos de automóviles con el objetivo de desarrollar un estándar industrial abierto para la comunicación entre vehículos. El consorcio tiene como objetivo incrementar la seguridad vial [03].

C2C-CC diseñó el protocolo C2CNet que difiere de IP, y puede ser usado por aplicaciones de seguridad y de no seguridad, como también para las comunicaciones basadas en infraestructura y sin infraestructura. Para el enrutamiento se usa un algoritmo basado en posicionamiento. En C2C-CC, la capa de red (C2CNet) provee comunicación multisalto basada en direccionamiento y enrutamiento geográfico. El envío periódico de *beacons*, servicio de ubicación y reenvío de datos son los principales componentes del enrutamiento geográfico. Las características de la arquitectura C2C-CC son las siguientes:

- Rápida transmisión para comunicaciones V2V y V2R.
- Soporte para la transmisión de diferentes tipos de mensajes incluyendo los mensajes de seguridad y entretenimiento.
- Utiliza diferentes estándares para redes LAN inalámbricas incluyendo IEEE 802.11p, las tradicionales IEEE 802.11 a/b/g/n, entre otras.

### **3. Metodología y Herramientas de Desarrollo**

En este capítulo se describe la metodología utilizada durante la realización de este trabajo, así como el conjunto de herramientas empleadas para la implementación del mismo.

#### **3.1. Fases de la Metodología**

La metodología propuesta para la elaboración del prototipo de dispositivo para redes VANETs contempla distintas actividades agrupadas en cuatro fases principales: análisis, diseño, implementación y pruebas.

##### **3.1.1. Análisis**

La etapa de análisis abarca la investigación y búsqueda de información necesaria para definir formalmente el problema, y consecuentemente fijar los límites del proyecto. Se establecen las bases para la elaboración de ideas y propuestas que busquen dar solución al problema definido. Básicamente se definen los requerimientos funcionales y no funcionales del equipo.

##### **3.1.2. Diseño**

A partir de los resultados obtenidos durante el análisis, se realiza el diseño conceptual del prototipo, donde se definen los módulos que deben formar parte de la solución. Para el diseño físico del prototipo, cada módulo corresponde a uno o varios componentes de hardware. En esta etapa de diseño se contemplan aspectos relativos a las dimensiones físicas, interfaces de comunicación, interacción con el usuario, costos, disponibilidad de compra, y otros aspectos de relevancia para que la integración de los componentes se pueda realizar de forma correcta. De esta forma se obtendrá el diseño conceptual y físico que sirva de solución al problema planteado. La etapa de diseño contempla también los componentes de software necesarios para el correcto funcionamiento de todos los módulos del dispositivo.

##### **3.1.3. Implementación**

En la etapa de implementación se lleva a cabo la construcción del dispositivo. Contempla la adquisición y ensamblaje de los componentes definidos en la fase de diseño, y la configuración de los componentes de software necesarios para su correcto funcionamiento.

### 3.1.4. Pruebas

En esta etapa se evalúa el funcionamiento del dispositivo. Se realizan pruebas de funcionalidad, para determinar que el prototipo cumpla con todos los requerimientos definidos.

## 3.2. Herramientas de Desarrollo

A continuación se describen las herramientas utilizadas para llevar a cabo la implementación del trabajo.

### 3.2.1. GCDC (Grand Cooperative Driving Challenge)

Los orígenes de GCDC se remontan en 2008, con la idea de una competencia técnica abierta para los interesados en sistemas cooperativos, tales como industrias de automóviles, proveedores de primer nivel, organizaciones académicas y de investigación y cualquier otra persona interesada en el campo [04].

GCDC es un proyecto en el cual equipos internacionales compiten para ofrecer un sistema de vehículo-infraestructura cooperativo más efectivo en escenarios de tráfico predeterminados. GCDC tiene como objetivo acelerar la implementación de estos sistemas y contribuir significativamente para aliviar los problemas de tráfico a nivel mundial [05].

La pila de comunicación de GCDC está basado en el trabajo de Eric Koenders, quien creó una serie de parches que se aplican sobre el módulo *ath5k* del *kernel* 2.6.32 de GNU/Linux, para manejar tarjetas Atheros, con el fin de usar las frecuencias del estándar IEEE 802.11p en las tarjetas en que utilicen el chipset Atheros AR5414. También tiene una implementación del protocolo CALM FAST en espacio de usuario.

En la página del proyecto se encuentra el software disponible y la documentación para su instalación [04]. Los principales componentes son:

- Parches para los módulos de *kernel* que dan soporte de IEEE 802.11p.
- Varias bibliotecas de soporte necesarias para CALM y CALM FAST.
- La herramienta *iw* para la configuración del módulo modificado.

### 3.2.2. Compat Wireless

Son un conjunto de controladores mejorados para el funcionamiento de tarjetas inalámbricas compatibles con IEEE 802.11a/b/g/n en GNU/Linux [05]. Se encuentra disponible para la versión del *kernel* 2.6.30 en adelante y soporta tarjetas con chipset Atheros, Broadcom, Ralink, entre otros.

Este paquete contiene el código fuente del módulo *ath5k*, sobre el cual se pueden aplicar los parches ofrecidos por el proyecto GCDC, para que las tarjetas con chipset Atheros AR5414 funcionen con el estándar IEEE 802.11p.

### 3.2.3. Debian Live 6.0

Debian Live es una distribución de GNU/Linux que no requiere de una instalación clásica para ser usada [06]. Se basa en 4 entornos de escritorio que son GNOME, KDE, LXDE y Xfce, al igual que el sistema estándar. Se conoce como un sistema en vivo (*live*), ya que se puede arrancar desde medios extraíbles como una memoria USB, sin necesidad de estar previamente instalado en un disco duro. Para guardar las modificaciones tras el reinicio del equipo, los sistemas en vivo incorporan una solución llamada *persistencia*, que consiste en la creación de una imagen binaria, que se instala en la memoria USB, donde se guardan los cambios realizados

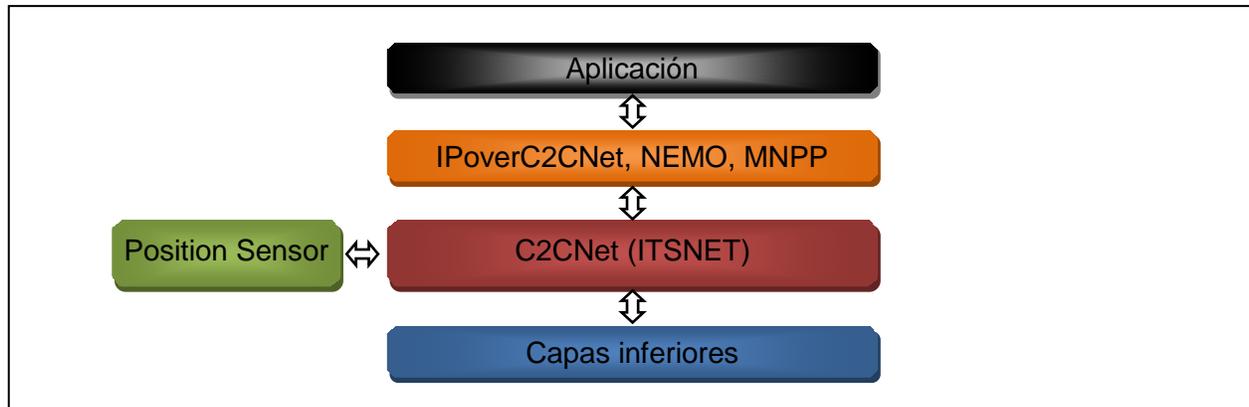
Debian está compuesto de paquetes de software de código abierto. Es una de las distribuciones de GNU/Linux más populares que basa su principio y fin en el software libre, además esta distribución combina el *kernel* Linux y utilidades GNU.

Entre las características más importantes de Debian se encuentra: la disponibilidad en varias plataformas de hardware (amd64, ia64, arm, mips, i386, entre otras) y un grupo de herramientas para facilitar el proceso de instalación y actualización del software (APT, Aptitude, Dpkg, Synaptic, etc).

### 3.2.4. Cargeo6

Cargeo6 es una implementación en código abierto de una arquitectura para comunicaciones vehiculares en ITS, incluye una pila de protocolos de red que combina IPv6 y Geonetworking (enrutamiento y direccionamiento geográfico) [09]. Sus capacidades extienden

el alcance de IPv6 y permiten la distribución de paquetes a todas las OBU y RSU ubicadas en una zona geográfica específica. Su última versión estable es la 0.9.8.1. En la Figura 3. 1 se observa una descripción simplificada de la arquitectura de Cargeo6.



**Figura 3. 1: Arquitectura de Cargeo6.**

El módulo *C2CNet* implementado en Cargeo6 se llama *itsnet*. Está a cargo del direccionamiento geográfico e implementa las funcionalidades de enrutamiento definidas en la especificación GeoNet D2.2 [11]. Se comunica directamente con la capa de enlace de datos y hace uso del módulo *position sensor* para obtener información del dispositivo GPS. En ausencia de un dispositivo GPS real, el módulo opcional *static position sender* suministra esta información, a partir de un archivo de configuración.

El módulo *IPoverC2CNet* implementa una interfaz entre IPv6 y la capa *C2CNet* que permite enviar datagramas IPv6 a través de *itsnet*. NEMO (Network Mobility) provee mecanismos para el manejo de movilidad del segmento de red asociado a una OBU. MNPP (Mobile Network Prefix Provisioning) implementa un mecanismo que informa a las OBUs vecinas sobre el segmento de red asociado a una OBU en particular.

### 3.2.5. Java

Java es una tecnología desarrollada por Sun Microsystems que ofrece la plataforma necesaria para la creación de interfaces gráficas (mediante el paquete swing) y todo el entorno de la aplicación implementada en el proyecto. Entre sus características se encuentran:

- Es portable.

- Posee una serie de bibliotecas donde están implementados los algoritmos que permiten enviar y recibir información a través de la interfaz de red.
- Tiene soporte de red IPv4 e IPv6.
- Algunas bibliotecas de java utilizadas son: GPSd4Java para usar los datos desde el demonio gpsd en las aplicaciones de Java, y JXMapKit para el despliegue de mapas en interfaces gráficas.
- Es gratuito.

### **3.2.6. Unetbootin**

Es una herramienta multiplataforma para la instalación de distintas distribuciones de GNU/Linux en memorias flash USB, a partir de una imagen ISO [08]. El sistema instalado en la memoria USB permite arrancar una computadora, sin necesidad de utilizar un disco duro o una unidad de CD/DVD. La herramienta permite crear una imagen binaria para activar la opción de persistencia en las distribuciones que tienen esta característica.

### **3.2.7. Downloadosmtiles.pl**

Es una herramienta para descargar mapas de una zona específica del planeta, pertenecientes al proyecto OpenStreetMaps [09][10]. Los mapas son descargados como imágenes PNG y automáticamente almacenados en un árbol de directorios en función del nivel de acercamiento (*zoom*) seleccionado.

## 4. Análisis y Diseño del Dispositivo

Para poder alcanzar los objetivos planteados en este trabajo, es necesario definir correctamente que es una *Unidad a bordo (On-board Unit, OBU)* y sus características más importantes, con el fin de identificar correctamente los requerimientos para poder elaborar un diseño adecuado del dispositivo.

### 4.1. OBU

La OBU es el componente principal de un vehículo que pertenece a una VANET. Es básicamente una computadora que se encuentra instalada en el vehículo, con los componentes de hardware y software necesarios que permiten la comunicación inalámbrica en escenarios V2V y V2R [12].

### 4.2. Características de la OBU

Una OBU debe cumplir con ciertas características para garantizar su correcto funcionamiento, entre las que destacan: poseer al menos una interfaz de comunicación inalámbrica, que según las especificaciones de los organismos de estandarización para las comunicaciones en redes VANETs, la tecnología ideal es IEEE 802.11p. Debe tener sensores para monitorear el comportamiento de algunas variables de un vehículo en movimiento, como: velocidad, posición (latitud, longitud, altitud), temperatura del motor, funcionamiento de los frenos, funcionamiento de las luces, etc. Incorporar una interfaz que le permita al conductor interactuar con el equipo mientras conduce el vehículo, por lo general se utilizan pantallas táctiles.

Además de esto, una OBU debe tener soporte de red, bien sea la pila de protocolos TCP/IP, CALM FAST, etc., una capa de software que permita utilizar un protocolo de enrutamiento y correr alguna aplicación para mantener informado al conductor.

Existen otros aspectos relativos al equipo, que no influyen directamente en su funcionamiento, pero que igualmente son considerados. La OBU se alimenta con la energía eléctrica suministrada por el vehículo, que corresponde a 12 VDC, y su consumo energético no debe poner en riesgo el correcto funcionamiento del mismo. Debe tener las dimensiones adecuadas para poder ser instalada dentro del vehículo, sin entorpecer el espacio interno del

mismo. Los sensores y las antenas deben estar ubicados en lugares donde no se limite la transmisión y/o recepción de las señales de radio.

### **4.3. Requerimientos funcionales del prototipo**

Luego de investigar y observar las características funcionales de una OBU, desarrollada en otros proyectos, y tomando en cuenta las características antes expuestas, se definen los siguientes requerimientos funcionales:

- El equipo debe establecer comunicación inalámbrica con otro equipo, que se encuentre instalado en un vehículo (comunicación V2V) o ubicado de forma permanente a los lados de la vía (V2R).
- Monitorear la posición (Latitud, Longitud) y velocidad del vehículo de forma constante.
- Debe permitir el correcto funcionamiento de un protocolo enrutamiento para redes VANETs.
- Transmitir y recibir mensajes de alerta cuando el conductor del vehículo lo considere necesario, haciendo uso de una pantalla táctil.

### **4.4. Requerimientos no funcionales del prototipo**

Existen un conjunto de restricciones y aspectos que se deben tomar en cuenta para el diseño del dispositivo:

- La alimentación eléctrica disponible en un vehículo es de 12 VDC, por ende, el dispositivo debe funcionar con esa energía.
- Las dimensiones de un vehículo son limitadas. El dispositivo debe ser pequeño y de fácil instalación dentro del vehículo, sin alterar ni entorpecer la comodidad de los pasajeros.
- Las antenas y sensores que requieran la transmisión y recepción de señales de radio desde exterior del vehículo, deben poder ubicarse en lugares donde no se limite su funcionamiento.
- No debe poseer componentes que se puedan dañar debido al movimiento del vehículo, como el caso de un disco duro tradicional.

- El equipo debe ser de bajo costo, y sus componentes deben encontrarse con relativa facilidad.

## 4.5. Diseño del equipo

Una vez definidos los requerimientos y las limitaciones del prototipo, se procede a realizar el diseño del hardware y el software del prototipo. Para esto se toma en cuenta la investigación sobre los posibles kits de desarrollo, componentes y materiales disponibles en el mercado, con los cuales se puede construir una OBU, realizada en [13].

El diseño final corresponde a la integración de todos los módulos de hardware y software requeridos para el correcto funcionamiento del equipo, a continuación se desarrollan por separado los diseños de hardware y software.

### 4.5.1. Diseño de hardware

El diseño de hardware contempla la parte física del equipo y se emplea un diseño modular, con la finalidad de que cada módulo corresponda al menos a un componente físico o tarjeta, que luego se puedan integrar por medio de interfaces bien definidas, equivalente al ensamblaje de una computadora de escritorio. En la Figura 4. 1 se observan los módulos que conforman el equipo.

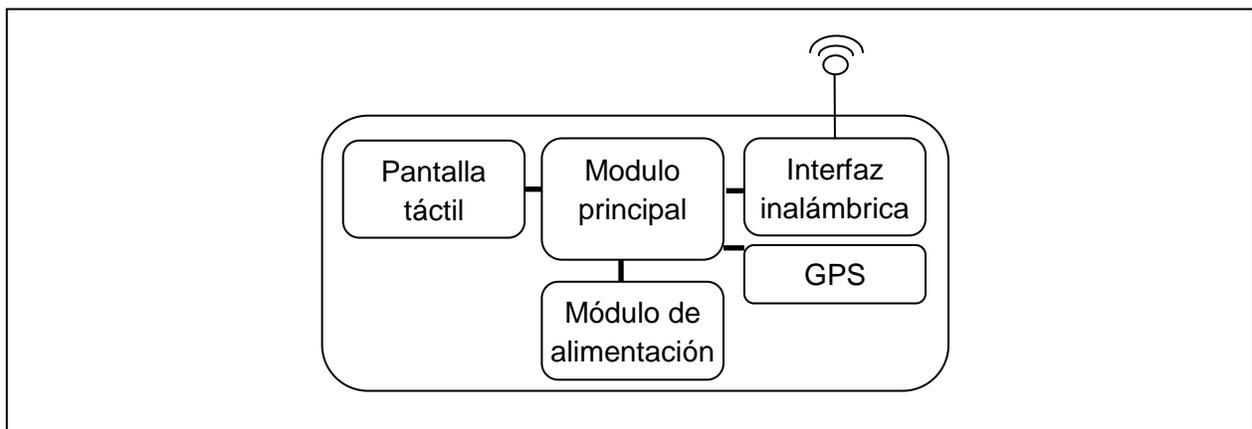


Figura 4. 1: Módulos de hardware.

El módulo principal es la base para integrar todos los módulos del equipo, está representado por una tarjeta madre que debe cumplir con los requerimientos definidos en el capítulo anterior. El módulo de alimentación se encarga de suministrar la energía eléctrica al equipo a partir de los 12 VDC de la batería del vehículo, la interfaz inalámbrica es el módulo para la comunicación inalámbrica del prototipo, el módulo GPS permite obtener las coordenadas de posición del vehículo (Latitud y longitud) y la velocidad, y la pantalla táctil permite al conductor del vehículo interactuar con el equipo.

#### 4.5.2. Diseño de software

El componente básico a nivel de software es el sistema operativo, el cual permite la instalación y configuración de los componentes de software que hacen funcionar correctamente cada módulo de hardware, como los son: el controlador de pantalla táctil, el controlador inalámbrico y el controlador GPS. Se incorpora un protocolo de enrutamiento, y el desarrollo de una aplicación para el envío y recepción de mensajes de alerta por parte del conductor (Aplicación Icaro-vanet). En la Figura 4. 2 se observa cuáles son los módulos correspondientes al diseño de software.

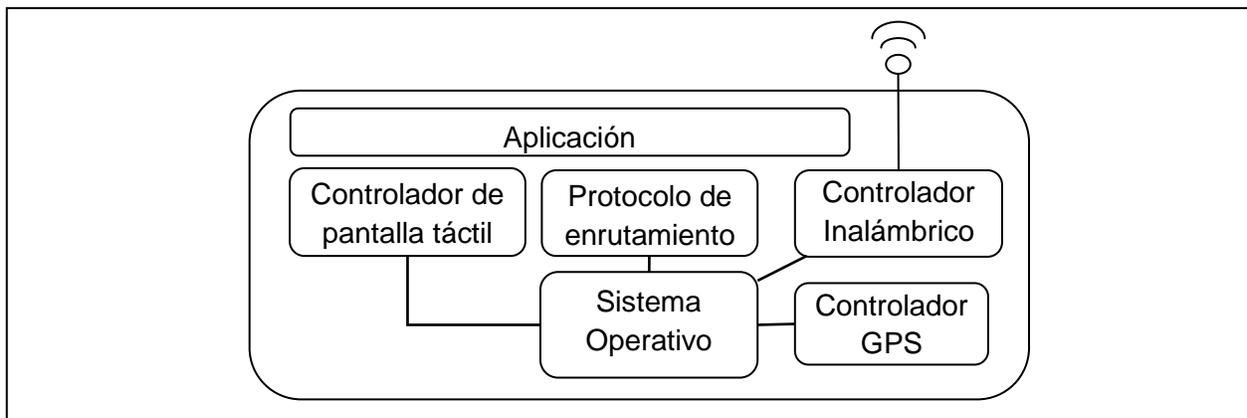


Figura 4. 2: Módulos de software.

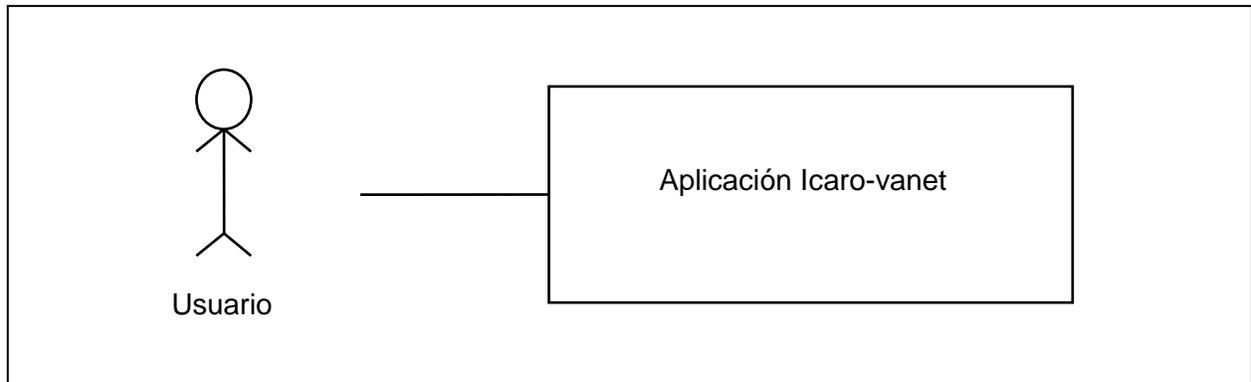
#### 4.5.3. Aplicación Icaro-vanet

Icaro-vanet es una aplicación para el envío y recepción de mensajes de alerta por parte del conductor, la cual debe cumplir con los siguientes requerimientos: mostrar información del vehículo (identificación, velocidad, latitud y longitud), mostrar la ubicación geográfica del vehículo en un mapa, enviar y recibir mensajes de congestión de tráfico vehicular y accidentes

de tránsito, interactuar con el conductor por medio de una interfaz gráfica sencilla, fácil de utilizar y que requiera el menor esfuerzo posible.

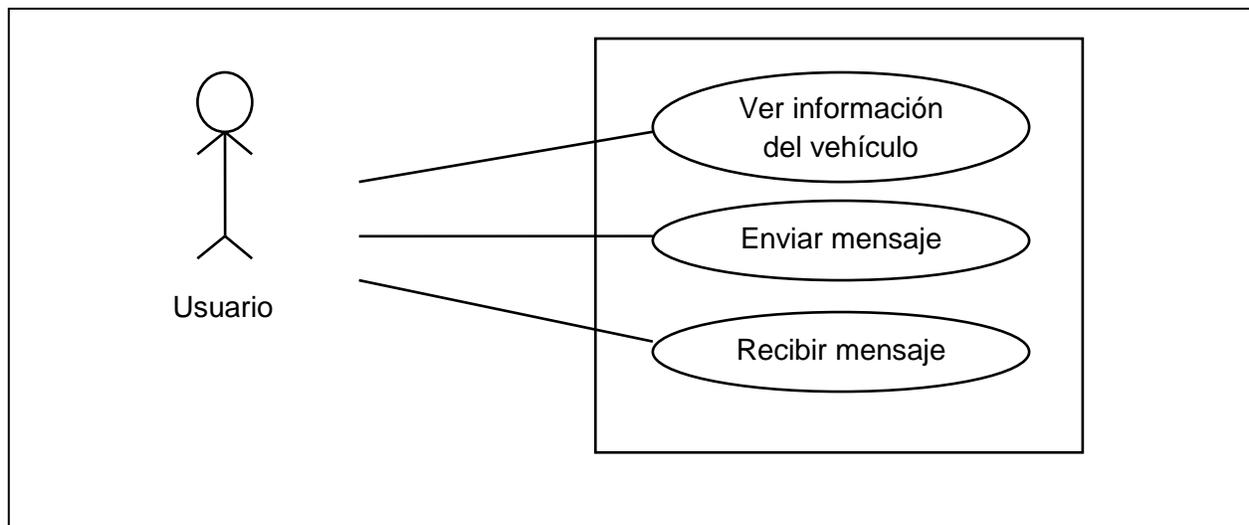
### **Diagrama de casos de uso**

A continuación se presentan los casos de uso. En la Figura 4. 3 se observa el diagrama de casos de uso nivel 0, donde se refleja la aplicación a crear y la interacción con el usuario.



**Figura 4. 3: Diagrama de casos de uso, nivel 0.**

En la Figura 4. 4 se observa el diagrama de casos de uso nivel 1, donde se definen las funcionalidades que debe cumplir la aplicación.



**Figura 4. 4: Diagrama de casos de uso, nivel 1.**

A continuación se encuentra la descripción de cada caso de uso de nivel 1 definido en la Figura 4. 4. Su especificación se encuentra en las siguientes tablas: Tabla 4. 1, Tabla 4. 2, y Tabla 4. 3.

Caso de uso	Ver información del vehículo.
Descripción	Vizualizar la información sobre la posición, velocidad, identificación y ubicación geográfica del vehículo.
Pre-condicion	Acceder a la aplicación.
Post-condicion	Información del vehículo.

**Tabla 4. 1: Especificación de caso de uso "Ver información del vehículo".**

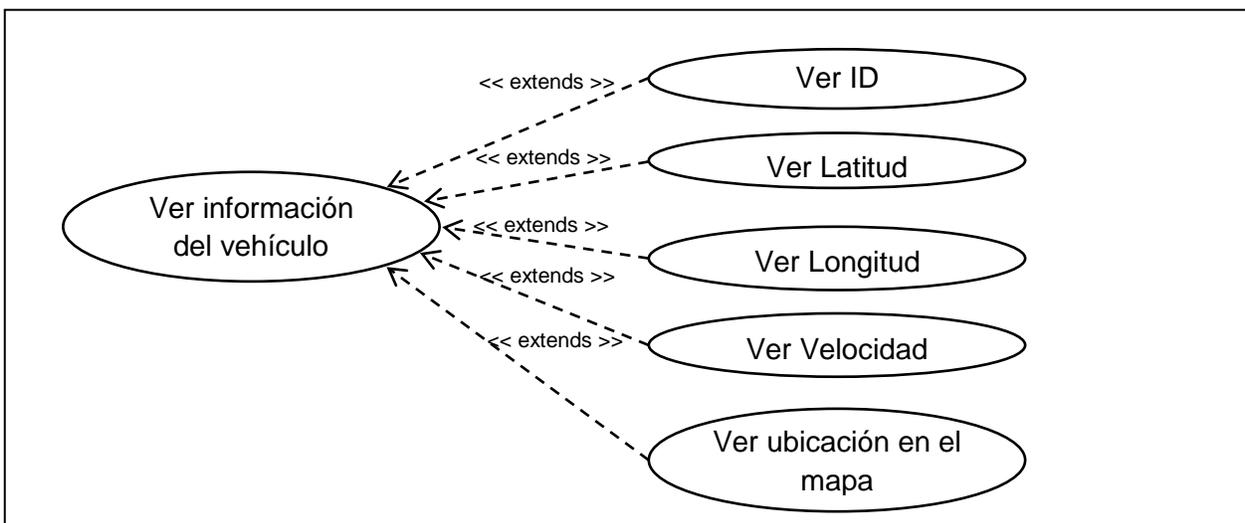
Caso de uso	Enviar mensaje.
Descripción	Enviar un mensaje de accidente de tránsito o congestión de tráfico a otro usuario.
Pre-condicion	Acceder a la aplicación.
Post-condicion	Mensaje de accidente o congestión enviado a otro usuario.

**Tabla 4. 2: Especificación de caso de uso "Enviar mensaje".**

Caso de uso	Recibir mensaje.
Descripción	Recibir un mensaje de accidente de tránsito o congestión enviado por otro usuario.
Pre-condicion	Acceder a la aplicación.
Post-condicion	Mensaje de accidente o congestión recibido.

**Tabla 4. 3: Especificación de caso de uso "Recibir mensaje".**

En la Figura 4. 5 se observa en detalle el caso de uso Ver información del vehículo, correspondiente al nivel 2 de abstracción.



**Figura 4. 5: Diagrama de caso de uso "Ver Información del vehículo", nivel 2.**

En las tablas: Tabla 4. 4, Tabla 4. 5, Tabla 4. 6, Tabla 4. 7, y Tabla 4. 8 se describen en detalle cada caso de uso de nivel 2 definidos en la Figura 4. 5.

Caso de uso	Ver ID.
Descripción	Vizualizar la identificación del vehículo.
Pre-condición	Acceder a la aplicación.
Post-condición	Información del identificador (ID) del vehículo.

**Tabla 4. 4: Especificación de caso de uso "Ver ID".**

Caso de uso	Ver Latitud.
Descripción	Vizualizar la coordenada de Latitud del vehículo.
Pre-condición	Acceder a la aplicación.
Post-condición	Latitud del vehículo.

**Tabla 4. 5: Especificación de caso de uso "Ver Latitud".**

Caso de uso	Ver Longitud.
Descripción	Vizualizar la coordenada de Longitud del vehículo.
Pre-condición	Acceder a la aplicación.
Post-condición	Longitud del vehículo.

**Tabla 4. 6: Especificación de caso de uso "Ver Longitud".**

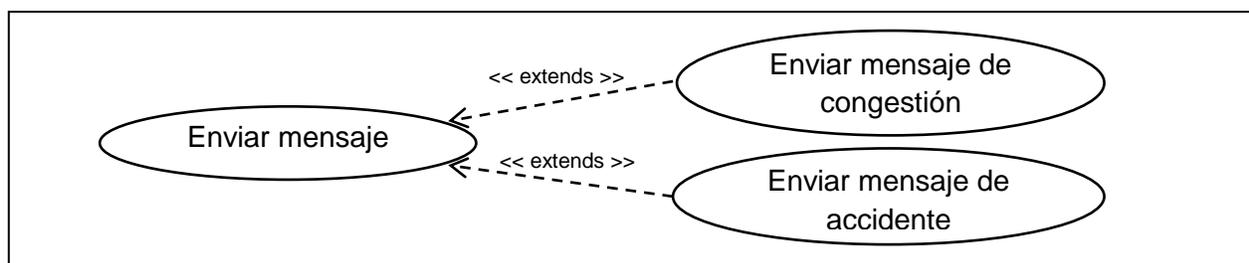
Caso de uso	Ver Velocidad.
Descripción	Vizualizar la velocidad actual del vehículo.
Pre-condición	Acceder a la aplicación.
Post-condición	Velocidad actual del vehículo.

**Tabla 4. 7: Especificación de caso de uso "Ver Velocidad".**

Caso de uso	Ver ubicación en el mapa.
Descripción	Vizualizar la ubicación geografica del vehiculo en un mapa.
Pre-condición	Acceder a la aplicación.
Post-condición	Ubicación geográfica del vehículo.

**Tabla 4. 8: Especificación de caso de uso "Ver ubicación en el mapa".**

En la Figura 4. 6 se observa en detalle el caso de uso Enviar mensaje, correspondiente al nivel 2 de abstracción.



**Figura 4. 6: Diagrama de caso de uso "Enviar mensaje", nivel 2.**

En la Tabla 4. 9, y Tabla 4. 10 se describen en detalle cada caso de uso de nivel 2 definidos en la Figura 4. 6.

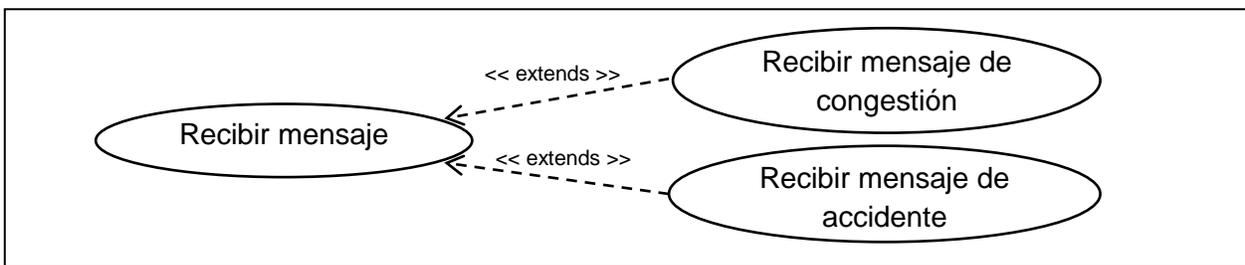
Caso de uso	Enviar mensaje de congestión.
Descripción	Enviar un mensaje de congestión de tráfico vehicular a todos los usuarios.
Pre-condición	Acceder a la aplicación.
Post-condición	Mensaje de congestión enviado a todos los usuarios.

**Tabla 4. 9: Especificación de caso de uso "Enviar mensaje de congestión".**

Caso de uso	Enviar mensaje de accidente.
Descripción	Enviar un mensaje de accidente de tránsito a un usuario.
Pre-condición	Acceder a la aplicación.
Post-condición	Mensaje de accidente enviado a un usuario.

**Tabla 4. 10: Especificación de caso de uso "Enviar mensaje de accidente".**

En la Figura 4. 7 se observa en detalle el caso de uso Recibir mensaje, correspondiente al nivel 2 de abstracción.



**Figura 4. 7: Diagrama de caso de uso "Recibir mensaje", nivel 2.**

En la Tabla 4. 11, y Tabla 4. 12 se describen en detalle cada caso de uso de nivel 2 definidos en la Figura 4. 7.

Caso de uso	Recibir mensaje de congestión.
Descripción	Recibir un mensaje de congestión de tráfico vehicular.
Pre-condición	Acceder a la aplicación.
Post-condición	Mensaje de congestión recibido.

**Tabla 4. 11: Especificación de caso de uso "Recibir mensaje de congestión".**

Caso de uso	Recibir mensaje de accidente.
Descripción	Recibir un mensaje de accidente de tránsito.
Pre-condición	Acceder a la aplicación.
Post-condición	Mensaje de accidente recibido.

**Tabla 4. 12: Especificación de caso de uso "Recibir mensaje de accidente".**

## 5. Implementación del Dispositivo

Luego de definir los requerimientos en la fase de análisis y la definición de los módulos de hardware y software en la fase de diseño, se inicia la fase de implementación. Esta fase contempla la selección, adquisición e integración de los componentes de hardware, y la instalación y configuración de los componentes de software para el correcto funcionamiento del equipo, para obtener como resultado un dispositivo que cumpla con todas las funcionalidades definidas en el capítulo 4.

### 5.1. Selección de los componentes de hardware

La base fundamental de desarrollo del dispositivo es la selección de las piezas de hardware que corresponden a los módulos definidos en la fase de diseño.

#### 5.1.1. Módulo principal

Para este módulo se selecciona la tarjeta Intel D525MW [11], cuyas dimensiones físicas se corresponden al estándar Mini-ITX (17 cm x 17 cm), viene integrada con un procesador Intel Atom, una ranura de expansión PCI, varios puertos USB 2.0, y una salida VGA, entre otras características. También forman parte de este módulo, una tarjeta de memoria RAM DDR3 Crucial de 2GB y una memoria flash USB de 16 GB. En la Figura 5. 1 se observan los tres componentes básicos de este módulo.

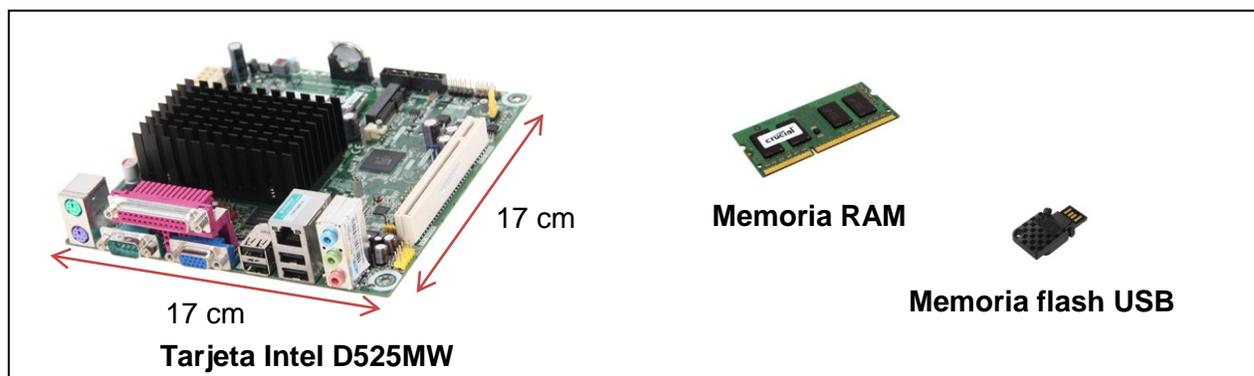


Figura 5. 1: Módulo principal.

### 5.1.2. Módulo de alimentación

Este módulo está integrado únicamente por la fuente de poder picoPSU-160-XT. Esta fuente es un conversor DC-DC que opera con 12VDC y genera los voltajes correspondientes a una fuente ATX estándar. Puede entregar una potencia de hasta 160 W y sus dimensiones son 5 cm x 3cm x2cm, equivalente al espacio ocupado por dos baterías AA. La Figura 5. 2 corresponde a la fuente picoPSU-160-XT.

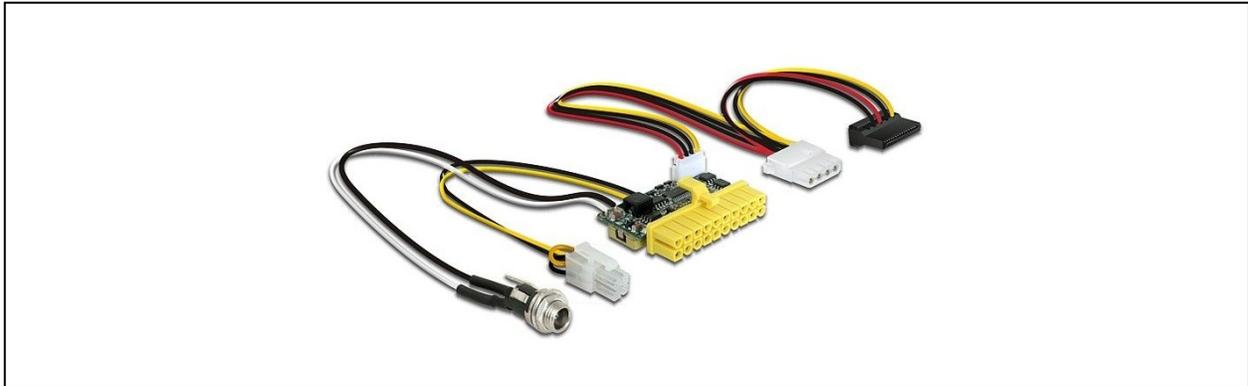


Figura 5. 2: Módulo de alimentación.

### 5.1.3. Interfaz Inalámbrica

Está formado por la tarjeta de red Belkin F6D3000-HP-NA, cuyo chipset es Atheros AR5414 y es compatible con el estándar IEEE 802.11a/b/g. Se integra al módulo principal a través de la ranura de expansión PCI. La Figura 5. 3 corresponde a la tarjeta red.



Figura 5. 3: Interfaz Inalámbrica.

#### 5.1.4. GPS

Está formado únicamente por el receptor GPS Globalsat BU-353, que se conecta al módulo principal por medio de un puerto USB. El receptor es de alta sensibilidad, tamaño reducido y puede ubicarse en cualquier parte del vehículo. En la Figura 5. 4 se observa el receptor GPS.



Figura 5. 4: GPS.

#### 5.1.5. Pantalla Táctil

Lo integra la pantalla LCD VM70, de 7 pulgadas. Se conecta al módulo principal por medio de un puerto VGA para obtener la señal de video, y utiliza una conexión por puerto USB para manejar las funciones táctiles. Esta pantalla operara con 12VDC y se puede observar en la Figura 5. 5.



Figura 5. 5: Pantalla táctil.

### 5.1.6. Chasis del dispositivo

El chasis del equipo se construyó con láminas de acero de 0.4 mm de espesor, tomando en cuenta las dimensiones de la tarjeta Intel D525MW, cuyo ancho y largo son 17 cm. Para la altura del chasis se tomó como referencia la distancia alcanzada desde la parte inferior de la tarjeta madre, hasta la parte superior de la tarjeta de red inalámbrica, una vez instalada, que corresponde a 5 cm.

El chasis está formado por cuatro partes: la base, sobre la cual va colocada la tarjeta madre, la cara posterior, la cara frontal, con un orificio rectangular sobre el cual sobresalen los conectores de la tarjeta, y la tapa, con agujeros para disipar el calor. En la Figura 5. 6 se observa con detalle cada una de estas partes, con sus respectivas dimensiones.

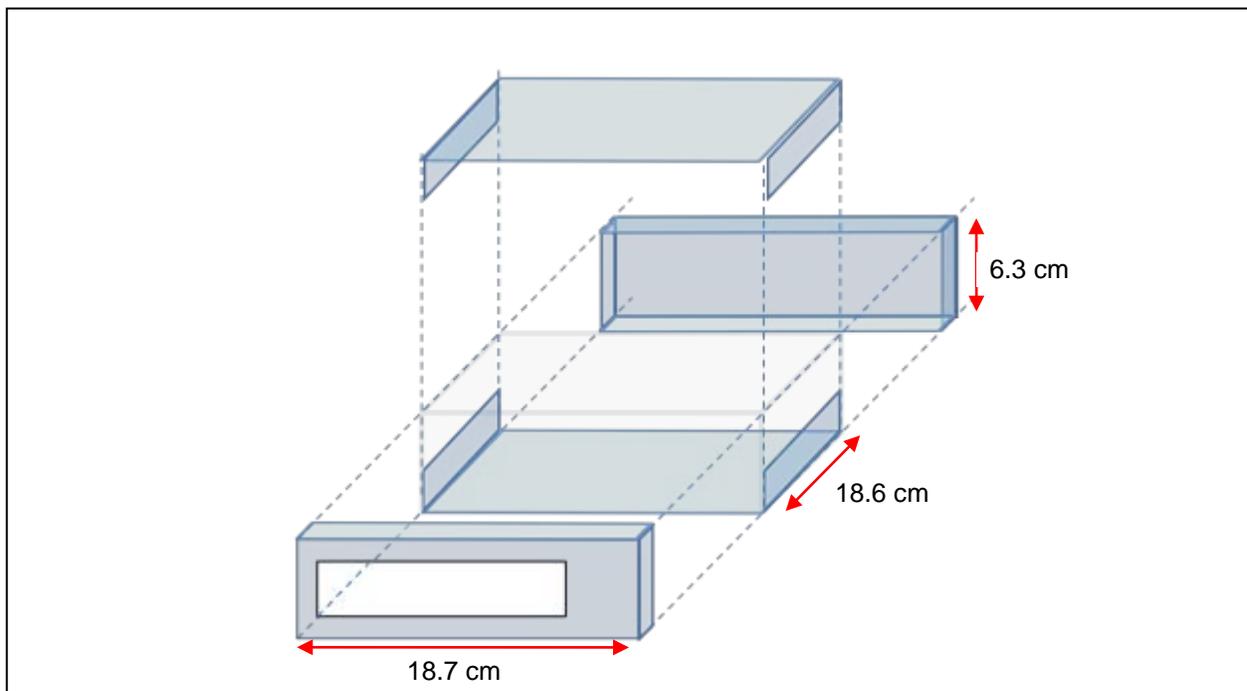


Figura 5. 6: Chasis del dispositivo.

### 5.2. Ensamblaje del equipo

La base fundamental del desarrollo del dispositivo es la construcción del mismo. Una vez obtenidas las piezas del hardware se procedió a ensamblar el equipo fijando la tarjeta madre Intel D525MW dentro del chasis, y sobre ésta se instalaron los demás componentes seleccionados, tal cual el proceso de ensamblaje de una computadora.

Es importante señalar que el dispositivo no utiliza disco duro, en su lugar se emplea una memoria USB de 16GB. La Figura 5. 7 y Figura 5. 8 corresponden a la implementación del dispositivo.



**Figura 5. 7: Módulo principal ensamblado.**



**Figura 5. 8: Dispositivo construido.**

### **5.3. Instalación del sistema operativo Debian Live 6.0**

Para el prototipo de unidad a bordo elaborado en este trabajo se ha elegido como sistema operativo la distribución de GNU/Linux Debian Live 6.0 porque ofrece la ventaja de arrancar desde medios extraíbles, como una memoria Flash USB, sin necesidad de estar

previamente instalado en un disco duro e incorpora una solución llamada *persistencia* para guardar las modificaciones realizadas tras el reinicio del equipo. Utiliza la versión de *kernel* 2.6.32-686, que permite la modificación del módulo *ath5k* con los parches provistos en el proyecto GCDC para hacer funcionar la tarjeta Belkin F6D3000-HP-NA (chipset Atheros AR5414) con una aproximación del estándar IEEE 802.11p. Adicionalmente incorpora el módulo de *kernel usbtouchscreen*, el cual es necesario para el funcionamiento de la pantalla táctil.

Se instaló Debian Live 6.0 en la memoria Flash USB de 16 GB, utilizando la herramienta Unetbootin. Adicionalmente se creó una imagen binaria de 3GB para el almacenamiento de la persistencia. Finalmente se configuraron los parámetros de arranque en modo seguro (*failsafe*) con la finalidad de evitar incompatibilidades con la tarjeta Intel D525MW, seleccionando el *kernel* 2.6.32-686 y la opción *persistent* para activar la persistencia.

#### **5.4. Instalación y configuración de la pantalla táctil**

Para el funcionamiento de la pantalla táctil VM70 de 7 pulgadas, se usan los módulos *usbtouchscreen* y *usbhid*. Estos módulos deben cargarse en el orden correcto para que el sistema pueda reconocer la pantalla y activar la funcionalidad táctil. También se necesitó la instalación de los paquetes *xinput* y *xinput-calibrator* para la calibración de la pantalla. Para configurar automáticamente la pantalla tras reiniciar el sistema, se elaboró un pequeño script que carga las configuraciones requeridas.

Adicionalmente, se instaló el paquete *matchbox-keyboard*, para tener a disposición un teclado virtual.

#### **5.5. Instalación del software GCDC**

Para poder utilizar una aproximación del estándar IEEE 802.11p, ideal para el entorno en redes VANETs, y usar la banda de frecuencias de 5.9 GHz (5.850-5.925 GHz), es necesario hacer algunas modificaciones sobre módulo que maneja la interfaz inalámbrica Atheros, llamado *ath5k*. En la página del proyecto GCDC se puede descargar el software y las herramientas necesarias para esto, y así hacer que una tarjeta Atheros compatible con IEEE 802.11a sea compatible con IEEE 802.11p [06]. Es importante señalar que estas

modificaciones se pueden realizar únicamente para las tarjetas Atheros con chipset AR5414, como es el caso de la tarjeta red Belkin F6D3000-HP-NA, incluida en el equipo.

Para esto, se descargó el paquete *compat-wireless* para la versión del *kernel 2.6.32*, y así se procedió con la instalación de los módulos *ath5k*, *mac80211* y *wireless*, sobre los cuales se realizan las modificaciones indicadas en el manual de instalación GCDC, para habilitar las frecuencias de IEEE 802.11p y las demás características de esta tecnología. También se instalaron los paquetes *pkg-config*, *libnl*, *libnl-dev*, *python-m2crypto*, *libgcrypt11-dev* para instalar las herramientas *iw*, *wireless-crda* y *wireless-regdb*, las cuales permiten las configuraciones para el uso de las frecuencias habilitadas.

## 5.6. Instalación de Cargeo6

Como protocolo de enrutamiento para redes VANETs se eligió Cargeo6 por ser una implementación en código abierto de una arquitectura para ITS definida por el *Instituto Europeo de Normas de Telecomunicaciones (European Telecommunications Standards Institute, ETSI)* bajo el estándar ETSI TS 302 636-4, que utiliza una combinación de IPv6 con un protocolo de enrutamiento basado en posición, el cual requiere para su funcionamiento, recibir datos de posición (Latitud y Longitud) de un receptor GPS.

La implementación del software Cargeo6 incluye cinco módulos, de los cuales, sólo se utilizaron tres. *C2CNet* también llamado *itsnet*, que implementa todas las funcionalidades de enrutamiento y direccionamiento geográfico. *IPoverC2CNet*, permite utilizar sobre *itsnet* la pila de protocolos TCP/IPv6 y *position sensor*, encargado de la adquisición de datos provenientes del GPS. Los otros módulos fueron descartados porque su funcionamiento está orientado a la existencia de una red móvil asociada a la OBU, que para el caso de este trabajo no existe.

Para la instalación del protocolo de enrutamiento se siguen las instrucciones de la guía de usuario del proyecto Cargeo6 [14], disponible en su portal web. Se requiere la instalación adicional de los paquetes *tunctlv1.5*, *gpsd*, y las bibliotecas *libpthread*, *libconfuse0*, *libgps19* y *libconfuse-dev*.

Haciendo uso del paquete *tunctlv1.5*, el módulo *IPoverC2CNet* crea y monta una interfaz de red virtual llamada *tun0* sobre la cual se escriben los datagramas provenientes de la pila de protocolos IPv6. De esta forma, los paquetes en la interfaz *tun0* pueden ser manipulados y

procesados fácilmente por cualquier programa en espacio de usuario, en particular el módulo *itsnet*.

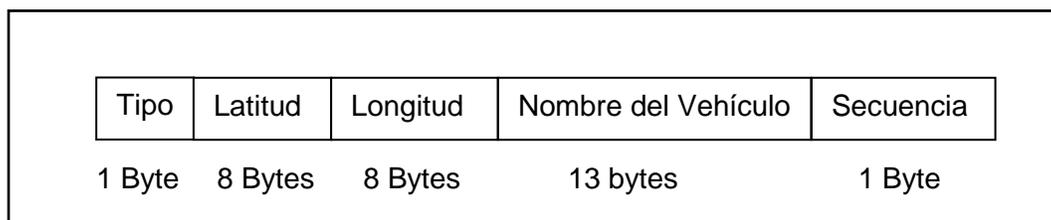
Para ejecutar Cargeo6 de forma correcta, el software incluye un archivo de llamado *itsnet.conf*, en el cual se configuran los parámetros básicos de *itsnet*. Adicionalmente se utilizan dos scripts: *create\_wifi.sh*, para configurar la red Ad Hoc y *start.sh*, para configurar los parámetros de IPv6 y levantar el demonio Cargeo6.

## 5.7. Aplicación Icaro-vanet

Hasta este punto de la implementación se tiene el equipo con los módulos de hardware y software instalados, el módulo *ath5k* modificado para el uso del estándar IEEE 802.11p y un protocolo de enrutamiento para redes VANETs. Teniendo este soporte de comunicación, se desarrolló una aplicación en Java para el envío y recepción de mensajes de alerta entre los nodos.

La aplicación se realizó con el *entorno de desarrollo integrado (integrated development environment, IDE)* Netbeans, que es un producto de código abierto y gratuito, sin restricciones de uso. Netbeans es un IDE hecho principalmente para el lenguaje de programación Java pero puede servir igualmente para otros lenguajes de programación.

Icaro-vanet tiene dos tipos de reportes básicos: accidente y congestión. Estos se envían en modo *unicast* y *multicast* respectivamente. Para el envío y recepción de reportes se utiliza el *protocolo de datagramas de usuario (User Datagram Protocol, UDP)*. Los datos que se envían en una secuencia de 31 Bytes con el formato que se describe en la Figura 5. 9.



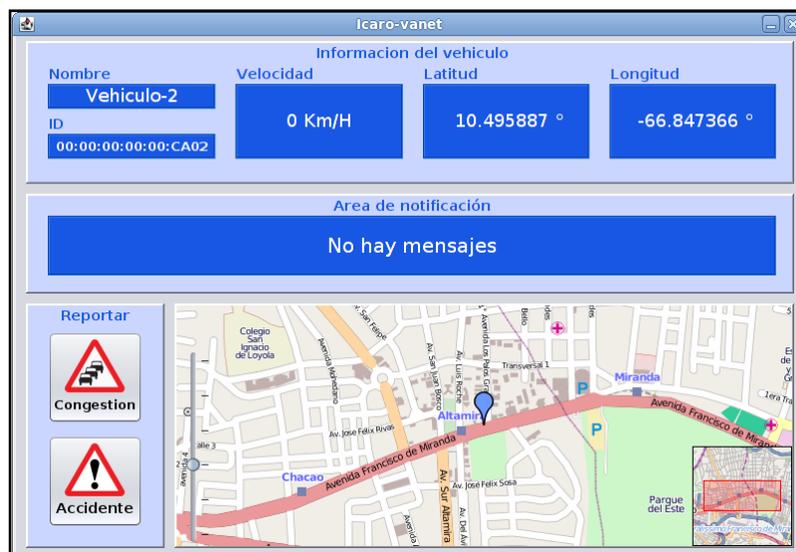
**Figura 5. 9: Formato de mensajes en la aplicación Icaro-vanet.**

- **Tipo:** indica si el reporte es congestión o accidente.
- **Latitud:** es la coordenada latitud tomada del receptor GPS.

- **Longitud:** es la coordenada longitud tomada del receptor GPS.
- **Nombre de vehículo:** es la cadena de caracteres del nombre del vehículo.
- **Secuencia:** es el número de secuencia de los reportes enviados.

También usa la información de las coordenadas recibidas desde el demonio *gpsd* a través de las funcionalidades de la biblioteca *GPSd4Java*. Tiene el despliegue de un mapa donde se refleja la posición actual del vehículo, de acuerdo a las coordenadas recibidas del GPS. Para el despliegue del mapa se usó la biblioteca *JXMapKit*, y se utilizan las imágenes del proyecto *OpenStreetMaps*, descargadas con la herramienta *downloadosmtiles.pl*.

La interfaz gráfica muestra información sobre vehículo: Latitud, longitud, velocidad, nombre e *identificador (identifier, ID)* del vehículo. Las coordenadas y la velocidad son tomadas a través del GPS, el nombre y el ID del vehículo se leen de un archivo de configuración. También se muestran en pantalla los mensajes recibido por un tiempo de 4 segundos, un mapa donde se indica la ubicación actual del vehículo y los botones para el envío de mensajes. En la Figura 5. 9 se observa la interfaz de la aplicación *Icaro-vanet*.



**Figura 5. 10: Interfaz de la aplicación Icaro-vanet.**

## 6. Pruebas y Análisis de Resultados

A continuación se describe el conjunto de pruebas realizadas para verificar el funcionamiento del dispositivo. En esta fase no se contemplan pruebas de rendimiento, sólo abarca un conjunto de pruebas fijas y pruebas de movilidad para verificar que el dispositivo funcione correctamente.

Con las pruebas fijas se busca validar el funcionamiento del protocolo de enrutamiento y con las pruebas de movilidad se desea evaluar el funcionamiento del dispositivo al ser instalado en un vehículo, donde la velocidad, distancia y otros factores afecten la conformación de la red Ad Hoc y la transmisión de los mensajes enviados con la aplicación Icaro-vanet. Las pruebas fijas fueron realizadas en las instalaciones del laboratorio ICARO, y las pruebas de movilidad en el estacionamiento de la Facultad de Ciencias.

### 6.1. Pruebas fijas

El objetivo de esta prueba es validar la funcionalidad del protocolo de enrutamiento, observando la ruta que toma un mensaje desde el nodo origen al nodo destino, para determinar si el camino recorrido por el mensaje es el esperado.

#### 6.1.1. Escenarios de prueba

Se definieron tres topologías, como se observa en la Figura 6. 1.

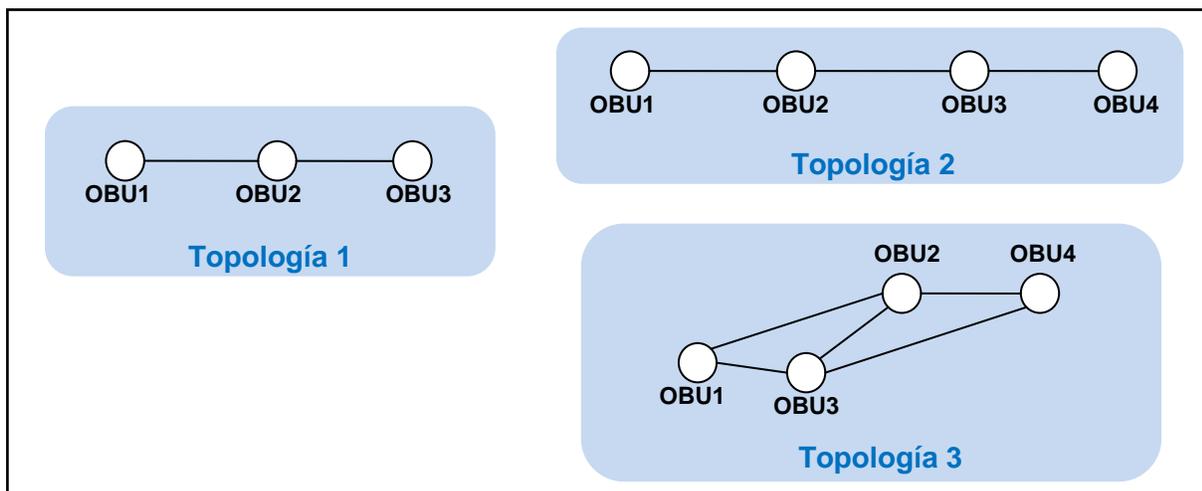


Figura 6. 1: Escenarios de pruebas fijas.

Se utilizaron las herramientas *wireshark*, *ping6* y el archivo con el registro de eventos de Cargeo6. *Wireshark* permite capturar los paquetes enviados y recibidos en cada interfaz de red. *Ping6* es utilizado para el envío de mensajes ICMPv6. El archivo del registro de eventos que genera Cargeo6 guarda entre otras cosas, la lista de vecinos en cada nodo.

Para realizar estas pruebas se utilizó el dispositivo construido y tres computadoras, equipadas con la tarjeta de red inalámbrica Belkin F6D3000-HP-NA (chipset Atheros AR5414). Para no modificar el sistema operativo de las mismas, se utilizaron tres memorias flash USB donde se instaló Debian 6.0 de forma persistente, con todos los componentes de software necesarios (controladores modificados, módulos, scripts, etc). No se utilizaron receptores GPS reales debido a mala recepción de señal que estos presentan dentro del laboratorio, en su lugar, se utilizó el módulo *static position sender*, que sustituye el funcionamiento de un GPS real. Este módulo lee un archivo de configuración donde se fijan las coordenadas de latitud y longitud deseadas para cada nodo. Las coordenadas se tomaron de Google Maps.

Adicionalmente se configuró la potencia de transmisión en las tarjetas de red inalámbricas, para garantizar que se establezcan solamente los enlaces indicados en cada topología. El canal de comunicación utilizado fue el 178, correspondiente a la frecuencia 5.890 MHz de la banda IEEE 802.11p, a una tasa de 6 Mbps

## **Topología 1**

En este escenario se utilizó el dispositivo construido (OBU1) y dos computadoras del laboratorio (OBU2 y OBU3) con las configuraciones de red reflejadas en la Tabla 6. 1.

Equipo	C2C-ID	Dirección IPv6 (tun0)	Dirección MAC (wlan0)
OBU1	0000:0000:0000:CA01	2001:660:3013:f001::ca01	00:17:3f:79:94:d8
OBU2	0000:0000:0000:CA02	2001:660:3013:f001::ca02	00:17:3f:79:9a:f7
OBU3	0000:0000:0000:CA03	2001:660:3013:f001::ca03	00:17:3f:79:9a:68

**Tabla 6. 1: Configuraciones de red en la topología 1.**

Luego de realizar las configuraciones de red necesarias, se obtuvo una red Ad Hoc con los enlaces de comunicación reflejados en la Figura 6. 2. Esto para asegurar que los mensajes que se envíen de un extremo a otro tengan que pasar por el nodo intermedio. Los mensajes ICMPv6 enviados a través del *ping6* se realizaron desde OBU3 como nodo origen y OBU1 como nodo destino.

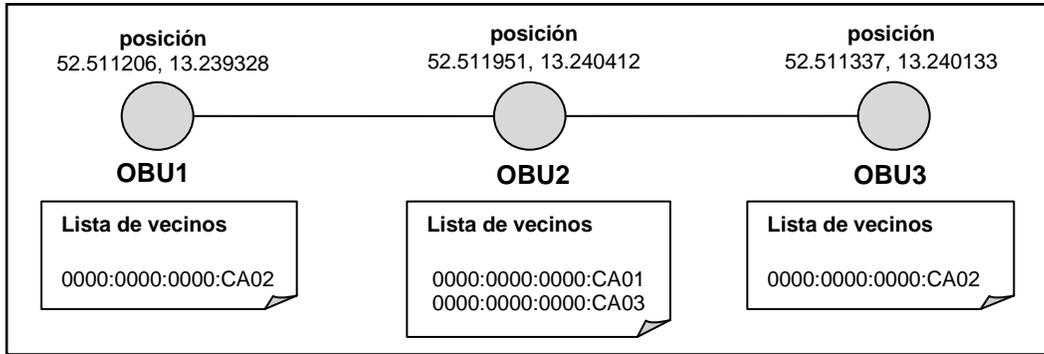


Figura 6. 2: Escenario de comunicación en la topología 1.

## Topología 2

Para esta topología se utilizó el dispositivo construido (OBU1) junto a tres computadoras del laboratorio (OBU2, OBU3 y OBU4) con las configuraciones de red que se muestran en la Tabla 6. 2.

Equipo	C2C-ID	Dirección Ipv6 (tun0)	Dirección MAC (wlan0)
OBU1	0000:0000:0000:CA01	2001:660:3013:f001::ca01	00:17:3f:79:94:d8
OBU2	0000:0000:0000:CA02	2001:660:3013:f001::ca02	00:17:3f:79:9a:f7
OBU3	0000:0000:0000:CA03	2001:660:3013:f001::ca03	00:17:3f:79:9a:68
OBU4	0000:0000:0000:CA04	2001:660:3013:f001::ca04	00:17:3f:79:99:03

Tabla 6. 2: Configuraciones de red en la topología 2.

Una vez que se realizaron las configuraciones de red necesarias se obtuvo la red Ad hoc deseada con el escenario de comunicación mostrado en la Figura 6. 3. En este escenario los mensajes son enviados desde OBU1 a OBU4, viéndose obligados a pasar por dos nodos intermedios.

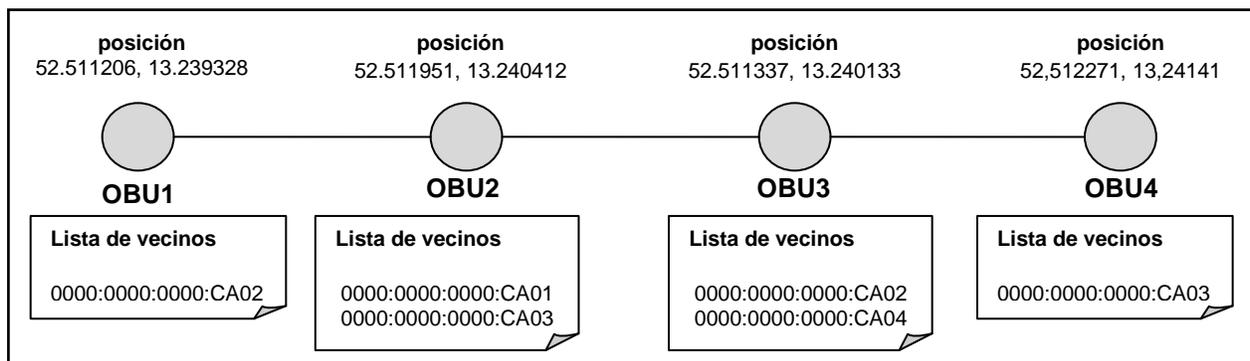


Figura 6. 3: Escenario de comunicación en la topología 2.

Al igual que en la topología 1 se enviaron mensajes ICMPv6 desde OBU1 a OBU4, para que estos pasen por los nodos intermedios OBU2 y OBU3.

### Topología 3

En este último escenario de pruebas se utilizó la misma configuración de red de la topología 2, mostrada en la Tabla 6. 2, con la topología para la red Ad hoc planteada en la Figura 6. 4.

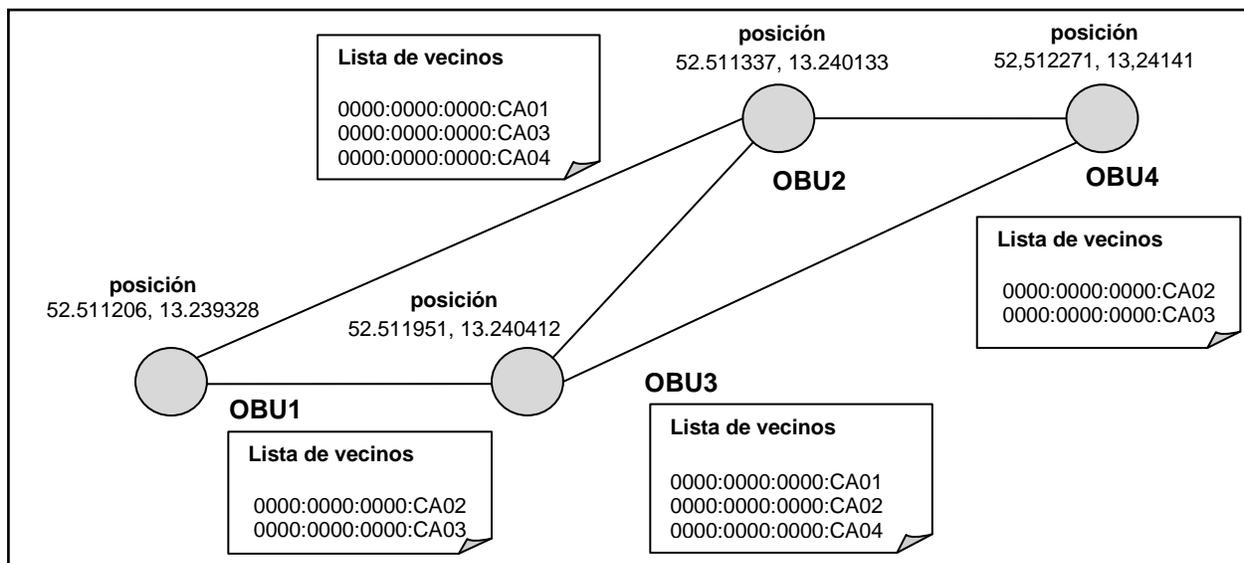


Figura 6. 4: Escenario de comunicación en la topología 3.

En este escenario se envían mensajes ICMPv6 desde OBU1 a OBU4, los cuales deben pasar por el nodo intermedio más cercano al nodo destino, que en este caso corresponde a OBU2. No debe haber réplica de mensajes por parte de OBU3.

#### 6.1.2. Resultados de las pruebas fijas

Los resultados obtenidos de las pruebas fijas se presentan en tres secciones distintas, correspondientes a cada topología planteada por cada escenario.

#### Resultados de la Topología 1

Luego de enviar los mensajes ICMPv6 por la interfaz tun0 desde OBU3 a OBU1, se puede observar en la Figura 6. 5 que se recibe repuesta del nodo OBU1. En la Tabla 6. 1 se especifican las direcciones IPv6 de la interfaz tun0 en cada uno de los nodos.

No. .	Time	Source	Destination	Protocol	Info
6	5.040785	2001:660:3013:f001::ca03	2001:660:3013:f001::ca01	ICMPV6	Echo request
7	5.062315	2001:660:3013:f001::ca01	2001:660:3013:f001::ca03	ICMPV6	Echo reply
8	6.041521	2001:660:3013:f001::ca03	2001:660:3013:f001::ca01	ICMPV6	Echo request
9	7.048635	2001:660:3013:f001::ca03	2001:660:3013:f001::ca01	ICMPV6	Echo request
10	7.070075	2001:660:3013:f001::ca01	2001:660:3013:f001::ca03	ICMPV6	Echo reply

**Figura 6. 5: Captura en la interfaz tun0 de OBU3 correspondiente a la topología 1.**

Además, en la Figura 6. 6 se puede observar en las dos capturas en negrita, que los mensajes enviados de OBU3 a OBU1 efectivamente primero pasan por el nodo intermedio OBU2 y este nodo entrega los mensajes al nodo OBU1, tal como se planteó en el escenario de la topología 1. Esto se observa por las direcciones MAC reflejadas en la Tabla 6. 1. Primero el mensaje es enviado desde *Belkin\_79:9a:68* (OBU3) como origen a *Belkin\_79:9a:f7* (OBU2) como destino, luego en el siguiente salto el mensaje es enviado desde *Belkin\_79:9a:f7* (OBU2) como origen a *Belkin\_79:94:d8* (OBU1), siendo este último el destino final.

No. .	Time	Source	Destination	Protocol	Info
603	122.181340	Belkin_79:9a:68	Broadcast	0x0707	Ethernet II
604	122.181378	Belkin_79:9a:f7	Broadcast	0x0707	Ethernet II
605	122.186410	Belkin_79:9a:68	Belkin_79:9a:f7	0x0707	Ethernet II
606	122.186454	Belkin_79:9a:f7	Belkin_79:94:d8	0x0707	Ethernet II
607	122.548418	Belkin_79:9a:f7	Broadcast	0x0707	Ethernet II

**Figura 6. 6: Captura en la interfaz wlan0 de OBU2 correspondiente a la topología 1.**

Adicionalmente, en la Figura 6. 7 se observan las listas de vecinos de cada uno de los nodos presentes. OBU1 y OBU3 sólo conocen a OBU2, OBU2 conoce a los nodos OBU1 y OBU3. Como consecuencia, no existe comunicación directa entre los nodos OBU1 y OBU3.

OBU1	
===== neighbor list =====	
mac_id=00:17:3f:79:9a:f7	node_id=0000:0000:0000:ca02   time=22   lat=420095608   long=105923296
=====	
OBU2	
===== neighbor list =====	
mac_id=00:17:3f:79:9a:68	node_id=0000:0000:0000:ca03   time=85   lat=420090696   long=105921064
mac_id=00:17:3f:79:94:d8	node_id=0000:0000:0000:ca01   time=51   lat=420089648   long=105914624
=====	
OBU3	
===== neighbor list =====	
mac_id=00:17:3f:79:9a:f7	node_id=0000:0000:0000:ca02   time=80   lat=420095608   long=105923296
=====	

**Figura 6. 7: Listas de vecinos de OBU1, OBU2 y OBU3 en la topología 1.**

## Resultados de la Topología 2

Tal como se describió anteriormente en la topología 2, los mensajes ICMPv6 fueron enviados desde el nodo OBU1 (nodo origen) al nodo OBU4 (nodo destino) por la interfaz tun0, las direcciones IPv6 son *2001:660:3013:f001::ca01* y *2001:660:3013:f001::ca04* en OBU1 y OBU4 respectivamente. En la Figura 6. 8 se observan los mensajes ICMPv6 enviados y recibidos en la interfaz tun0 del nodo OBU1.

No. .	Time	Source	Destination	Protocol	Info
127	980.941432	2001:660:3013:f001::ca01	2001:660:3013:f001::ca04	ICMPv6	Echo request
128	981.941460	2001:660:3013:f001::ca01	2001:660:3013:f001::ca04	ICMPv6	Echo request
129	981.963076	2001:660:3013:f001::ca04	2001:660:3013:f001::ca01	ICMPv6	Echo reply
130	982.943163	2001:660:3013:f001::ca01	2001:660:3013:f001::ca04	ICMPv6	Echo request

**Figura 6. 8: Captura en la interfaz tun0 de OBU1 correspondiente a la topología 2.**

Para que los mensajes lleguen de un extremo a otro deben pasar por los nodos intermedios OBU2 y OBU3 de acuerdo al escenario planteado en la topología 2. En la Figura 6. 9 se observa la ruta que sigue un mensaje desde el nodo OBU1 al nodo OBU4 en las capturas resaltadas en negrita. Como se puede apreciar de acuerdo a las direcciones MAC de cada nodo especificada en la Tabla 6. 2, el mensaje sale desde el origen *Belkin\_79:94:d8* (OBU1) a *Belkin\_79:9a:f7* (OBU2), luego va de *Belkin\_79:9a:f7* (OBU2) a *Belkin\_79:9a:68* (OBU3), y como último salto el mensaje se envía de *Belkin\_79:9a:68* (OBU3) a *Belkin\_79:99:03* (OBU4), siendo este último nodo el destino final del mensaje enviado por el nodo OBU1.

No. .	Time	Source	Destination	Protocol	Info
8784	1376.37246	<b>Belkin_79:9a:68</b>	Broadcast	0x0707	Ethernet II
8785	1376.37323	<b>Belkin_79:94:d8</b>	<b>Belkin_79:9a:f7</b>	0x0707	Ethernet II
8786	1376.37329	<b>Belkin_79:9a:f7</b>	<b>Belkin_79:9a:68</b>	0x0707	Ethernet II
8787	1376.37495	<b>Belkin_79:9a:68</b>	<b>Belkin_79:99:03</b>	0x0707	Ethernet II
8788	1376.38368	<b>Belkin_79:9a:68</b>	Broadcast	0x0707	Ethernet II

**Figura 6. 9: Captura en la interfaz wlan0 de OBU2.**

En la Figura 6. 10 se observan las listas de vecinos de cada uno de los nodos presentes. OBU1 sólo conoce a OBU2, OBU2 conoce a OBU1 y OBU3, OBU3 conoce a OBU2 y OBU4, OBU4 sólo conoce a OBU3. Por lo tanto no existe comunicación directa entre OBU1 y OBU4.

OBU1					
===== neighbor list =====					
mac_id=00:17:3f:79:9a:f7	node_id=0000:0000:0000:ca02	time=68	lat=420095608	long=105923296	
=====					
OBU2					
===== neighbor list =====					
mac_id=00:17:3f:79:9a:68	node_id=0000:0000:0000:ca03	time=12	lat=420090696	long=105921064	
mac_id=00:17:3f:79:94:d8	node_id=0000:0000:0000:ca01	time=84	lat=420089648	long=105914624	
=====					
OBU3					
===== neighbor list =====					
mac_id=00:17:3f:79:9a:f7	node_id=0000:0000:0000:ca02	time=42	lat=420095608	long=105923296	
mac_id=00:17:3f:79:99:03	node_id=0000:0000:0000:ca04	time=34	lat=420098168	long=105931280	
=====					
OBU4					
===== neighbor list =====					
mac_id=00:17:3f:79:9a:68	node_id=0000:0000:0000:ca03	time=95	lat=420090696	long=105921064	
=====					

Figura 6. 10: Listas de vecinos de OBU1, OBU2, OBU3 y OBU4 en la topología 2.

### Resultados de la topología 3

En este escenario como se describió anteriormente los mensajes se enviaron desde OBU1 a OBU4. Los mensajes ICMPv6 fueron enviados por la interfaz tun0, y las direcciones IPv6 para cada nodo se especifican en la Tabla 6. 2. En la Figura 6. 11 se observa los mensajes enviados y recibidos por OBU1 (nodo origen).

No. .	Time	Source	Destination	Protocol	Info
1	0.000000	2001:660:3013:f001::ca01	2001:660:3013:f001::ca04	ICMPv6	Echo request
2	0.024727	2001:660:3013:f001::ca04	2001:660:3013:f001::ca01	ICMPv6	Echo reply
3	1.001793	2001:660:3013:f001::ca01	2001:660:3013:f001::ca04	ICMPv6	Echo request
4	1.025342	2001:660:3013:f001::ca04	2001:660:3013:f001::ca01	ICMPv6	Echo reply

Figura 6. 11: Captura en la interfaz tun0 de OBU1 correspondiente a la topología 3.

Los mensajes enviados desde OBU1 (nodo origen) a OBU4 (nodo destino) pasan por OBU2 (por ser el más cercano al destino) y luego son reenviados a OBU4. Esto se observa en la Figura 6. 12 con las 2 capturas que están en negrita. El mensaje sale de *Belkin\_79:94:d8* (OBU1) a *Belkin\_79:9a:f7* (OBU2), y luego es entregado al destino *Belkin\_79:99:03* (OBU4) desde *Belkin\_79:9a:f7* (OBU2).

No. .	Time	Source	Destination	Protocol	Info
9	0.774929	<b>Belkin_79:9a:f7</b>	Broadcast	0x0707	Ethernet II
10	0.779124	<b>Belkin_79:9a:68</b>	Broadcast	0x0707	Ethernet II
11	0.784066	<b>Belkin_79:94:d8</b>	<b>Belkin_79:9a:f7</b>	0x0707	Ethernet II
12	0.784102	<b>Belkin_79:9a:f7</b>	<b>Belkin_79:99:03</b>	0x0707	Ethernet II
13	0.789719	<b>Belkin_79:99:03</b>	Broadcast	0x0707	Ethernet II

**Figura 6. 12: Captura en la interfaz wlan0 de OBU2.**

En la Figura 6. 13 se observan las listas de vecinos de cada uno de los nodos presentes. OBU1 conoce a OBU2 y OBU3. OBU2 conoce a OBU1, OBU3 y OBU4. OBU3 conoce a OBU1, OBU2 y OBU4. OBU4 conoce a OBU2 y OBU3.

OBU1					
===== neighbor list =====					
mac_id=00:17:3f:79:9a:f7	node_id=0000:0000:0000:ca02	time=59	lat=420095608	long=105923296	
mac_id=00:17:3f:79:9a:68	node_id=0000:0000:0000:ca03	time=63	lat=420090696	long=105921064	
=====					
OBU2					
===== neighbor list =====					
mac_id=00:17:3f:79:9a:68	node_id=0000:0000:0000:ca03	time=52	lat=420090696	long=105921064	
mac_id=00:17:3f:79:94:d8	node_id=0000:0000:0000:ca01	time=23	lat=420089648	long=105914624	
mac_id=00:17:3f:79:99:03	node_id=0000:0000:0000:ca04	time=36	lat=420098168	long=105931280	
=====					
OBU4					
===== neighbor list =====					
mac_id=00:17:3f:79:9a:f7	node_id=0000:0000:0000:ca02	time=66	lat=420095608	long=105923296	
mac_id=00:17:3f:79:9a:68	node_id=0000:0000:0000:ca03	time=70	lat=420090696	long=105921064	
=====					
OBU3					
===== neighbor list =====					
mac_id=00:17:3f:79:9a:f7	node_id=0000:0000:0000:ca02	time=99	lat=420095608	long=105923296	
mac_id=00:17:3f:79:99:03	node_id=0000:0000:0000:ca04	time=86	lat=420098168	long=105931280	
mac_id=00:17:3f:79:94:d8	node_id=0000:0000:0000:ca01	time=73	lat=420089648	long=105914624	
=====					

**Figura 6. 13: Listas de vecinos de OBU1, OBU2, OBU3 y OBU4 en la topología 3.**

### 6.1.3. Análisis de resultados de las pruebas fijas

De acuerdo a los resultados obtenidos en las pruebas que fueron realizadas en el laboratorio ICARO se pudo validar que el protocolo de enrutamiento funcionó correctamente usando las rutas deseadas en las tres topologías planteadas anteriormente. Sin embargo, aunque no se realizaron medidas de rendimiento exhaustiva, se pudo apreciar que en la topología 2 donde los mensajes emplean tres saltos hasta su destino, se perdieron varios mensajes.

Se puede decir que en la implementación del protocolo hay algunas limitaciones en cuanto al número de saltos que debe recorrer un mensaje para llegar a su destino. Esto es sustentado con los resultados de las pruebas realizadas a Cargeo6 por parte de los desarrolladores del mismo en el documento GeoNet D7.1 [15], donde se afirma que en topologías donde un mensaje debe recorrer más de dos saltos para llegar a su destino, el

protocolo presenta algunos inconvenientes, que según el documento, son objeto de estudio para realizar mejoras.

## 6.2. Pruebas de movilidad

El objetivo de estas pruebas es evaluar el funcionamiento del dispositivo al ser instalado en un vehículo, donde la velocidad, distancia y otros factores afecten la transmisión de los mensajes enviados con la aplicación Icaro-vanet.

Las pruebas fueron realizadas en el estacionamiento de la Facultad de Ciencias frente a la Escuela de Matemáticas. El escenario planteado se observa en la Figura 6. 14. La prueba consiste en enviar mensajes desde una OBU en un vehículo en movimiento a una RSU ubicada en un lugar fijo. Se definen tres puntos de referencia (A, B, C) en el trayecto de movilidad del vehículo para enviar los mensajes. La prueba se realizó tres veces, a velocidades de 20 Km/h, 40 km/h y 50 km/h.

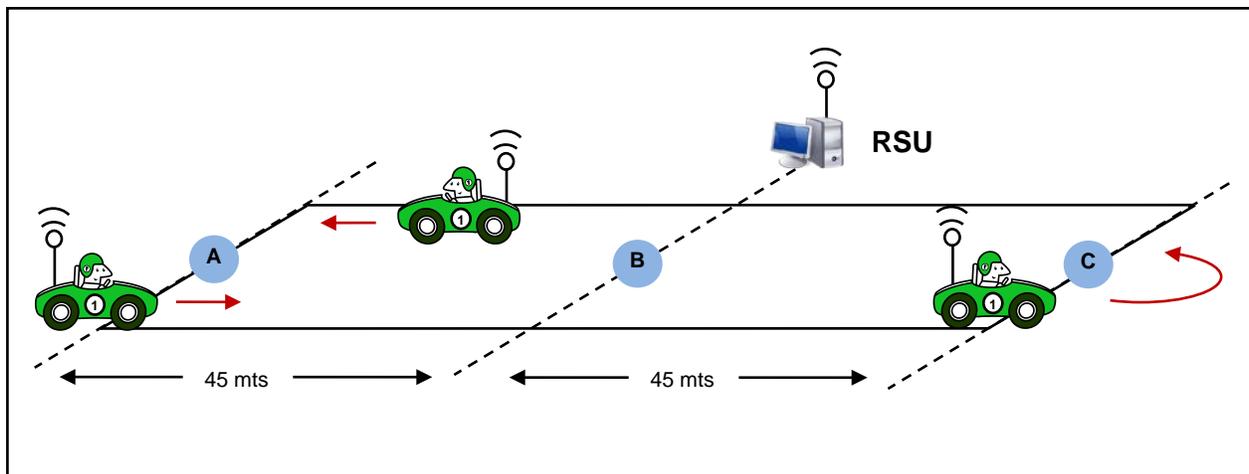


Figura 6. 14: Escenario para las pruebas de movilidad.

Como RSU se utilizó una computadora ubicada en la entrada del centro IOMMA (Centro de Investigación de Operaciones y Modelos Matemáticos). Como OBU se utilizó el prototipo construido, el cual fue instalado en un vehículo particular. La antena de la OBU se ubicó dentro del vehículo.

Tanto la RSU como la OBU utilizaron receptores GPS reales. Ambos equipos tenían instalados todos los componentes de software necesarios (controladores modificados, módulos,

scripts, aplicaciones, etc). Se utilizó el canal de comunicación 178, correspondiente a la frecuencia 5.890 MHz de la banda IEEE 802.11p, a una tasa de 6 Mbps. La potencia de transmisión de los equipos fue fijada en 30 dBm.

Para esta prueba se utilizó la aplicación Icaro-vanet y un servidor NTP (*Network Time Protocol*). Con la aplicación Icaro-vanet se envían mensajes de reporte de accidente y de congestión haciendo uso de la pantalla táctil del equipo, y por medio del registro de eventos de Icaro-vanet, es posible conocer cuántos mensajes son recibidos exitosamente y cuánto tiempo tardan en llegar a su destino. El servidor NTP permite sincronizar los equipos a una misma hora, para medir el tiempo que tarda en llegar un mensaje desde el origen al destino.

La prueba consistió en dar un recorrido de dos vueltas enviando mensajes de accidentes con la aplicación Icaro-vanet y posteriormente un recorrido de dos vueltas enviando reportes de congestión, en cada punto señalado en la Figura 6. 14, para un total de veinticuatro mensajes entre accidente y congestión. La prueba se realizó tres veces, a velocidades de 20Km/H, 40 Km/H y 55Km/H.

### 6.2.1. Resultados de las pruebas de movilidad

Al culminar las pruebas de movilidad tomaron los siguientes resultados reflejados en la Tabla 6. 3. Estos resultados corresponden a los mensajes enviados y recibidos por la aplicación Icaro-vanet. El tiempo promedio se refiere al tiempo que tarda en llegar un mensaje enviado desde la OBU al RSU.

Velocidad	Mensajes enviados	Mensajes recibidos	Mensajes perdidos	Tiempo promedio (ms)
20 Km/h	24	22	2	347.09
40 Km/h	24	22	2	342.04
55 Km/h	24	19	5	338.21

**Tabla 6. 3: Mensajes de accidente y congestión a 20Km/H, 40Km/H y 55 Km/H.**

Adicionalmente, en la Figura 6. 15 se observa una gráfica comparativa del tiempo promedio que requiere cada mensaje para llegar al destino, a velocidades de 20 Km/H, 40 Km/H y 55 Km/H.

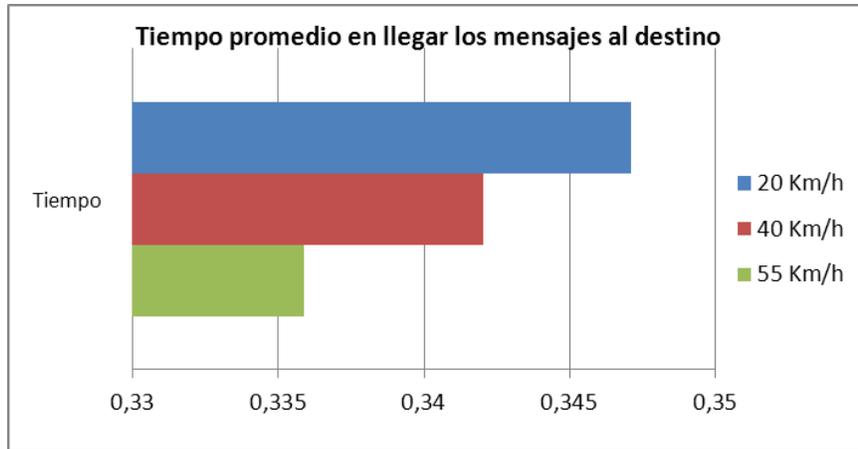
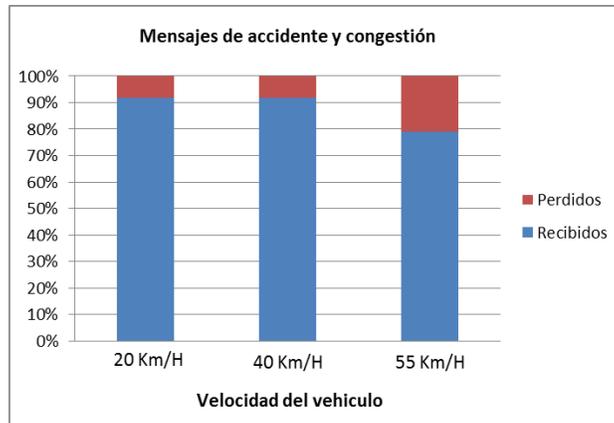


Figura 6. 15: Comparación del tiempo promedio que tardan los mensajes en llegar al destino.

### 6.2.2. Análisis de resultados de las pruebas de movilidad

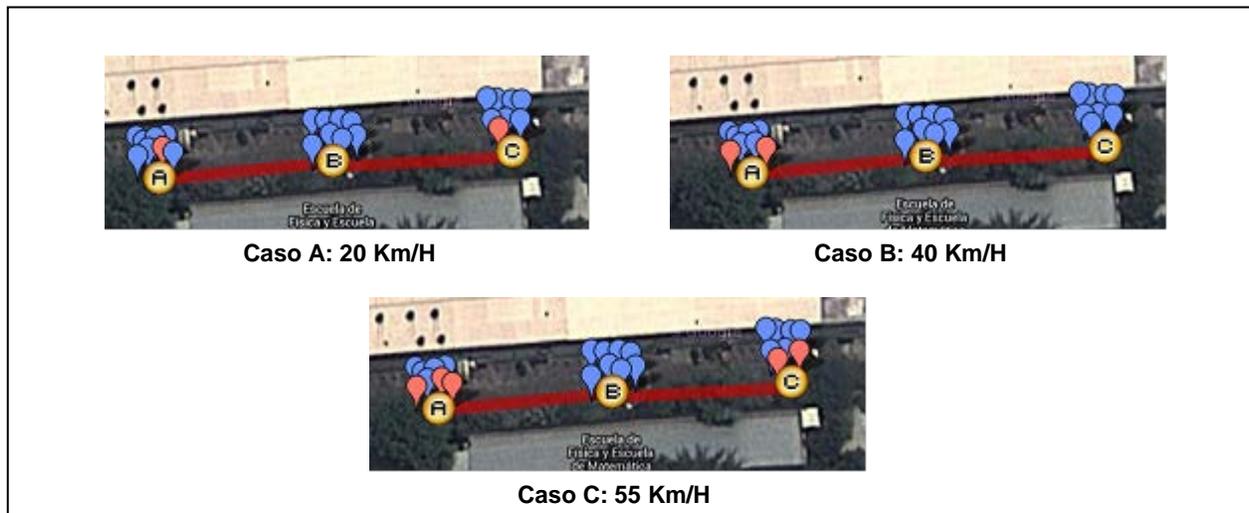
Con los resultados obtenidos en las pruebas de movilidad, se evidencia que a mayor velocidad, aumenta la pérdida de mensajes de accidente y/o congestión, tal como se refleja en la Tabla 6.3 y Figura 6. 16.

En el registro de eventos de la aplicación Icaro-vanet se observa que a 20 Km/H se pierde el cuarto y el séptimo mensaje, que corresponden a los puntos C y A de la trayectoria del vehículo, como se indica con los puntos de color rojo en la Figura 6. 17 (Caso A). Siguiendo la secuencia de los mensajes se observa que ambos son enviados al inicio de la trayectoria. A 40 Km/H se pierden igualmente dos mensajes, en este caso en el punto A (Caso B de la Figura 6. 17). El primero al final de la trayectoria y el segundo al inicio. A 55 Km/H se pierden cinco mensajes, tres en el punto A y dos en el punto C (Caso C de la Figura 6. 17). Cuatro de estos iniciando la trayectoria del vehículo y uno finalizando.



**Figura 6. 16: Relación entre mensajes recibidos y perdidos por la aplicación.**

De los nueve mensajes perdidos, siete son enviados al inicio de la trayectoria del vehículo, lo que puede corresponder al momento en el que se establece la red Ad Hoc entre la OBU y el RSU. Y los dos restantes se pierden al final de la trayectoria. Con esto puede considerarse que en el momento de enviar estos mensajes la OBU no tenía conexión a la red Ad Hoc. Adicionalmente se puede afirmar que en la zona de cobertura de la señal, representada por el punto B, la pérdida de los mensajes es de 0%, independientemente de que el vehículo se desplace a 20Km/H, 40 Km/H o 55 Km/H.



**Figura 6. 17: Mensajes de accidente y congestión en la trayectoria del vehículo.**

### 6.3. Prueba de cobertura

Además de realizar las pruebas fijas y de movilidad se realizó una prueba para medir la cobertura de la señal de la red Ad Hoc. Para esto, se realizó una prueba en el estacionamiento de la Facultad de Ciencias usando la herramienta *iwlist* de Linux [19], la cual es usada para desplegar información de las redes inalámbricas disponibles. La prueba consistió en fijar una computadora en un punto establecido y definir seis puntos de referencia para medir la señal en estos puntos con el prototipo construido.

#### 6.3.1. Resultados

La Tabla 6. 4 refleja los resultados obtenidos en los puntos de referencia, donde la distancia se mide desde la computadora fija al prototipo construido en cada uno de los puntos de referencia.

Distancia (mts)	Calidad de señal	Nivel de señal (dBm)
1	61/70	-49
15	24/70	-86
30	16/70	-94
45	14/70	-96
60	13/70	-97

**Tabla 6. 4: Cobertura de la señal.**

A una distancia aproximada de 45 metros, el nivel de la señal permite enviar y recibir mensajes ICMPv6 y de la aplicación Icaro-vanet de forma satisfactoria.

## 7. Conclusiones

En este trabajo se ha diseñado y construido un dispositivo de comunicación inalámbrica que cumple con los requerimientos básicos para ser instalado a bordo de un vehículo, con soporte de comunicación vía una VANET haciendo uso la especificación IEEE 802.11p. La solución incluye un protocolo de enrutamiento basado en posición y la implementación de la aplicación Icaro-vanet, para el envío y recepción de mensajes de accidente y congestión por parte del conductor, mediante el uso de una pantalla táctil.

### 7.1. Contribuciones

Los resultados obtenidos en la realización de este trabajo indican que es posible implementar un dispositivo de dimensiones físicas adecuadas para ser instalado dentro de un vehículo, con los componentes de hardware y software que permiten la comunicación inalámbrica en escenarios de redes VANETs con alta confiabilidad.

El dispositivo desarrollado cumple con los requerimientos funcionales y no funcionales planteados en la fase de Análisis. Los componentes de hardware y software utilizados ofrecen una amplia flexibilidad al poseer las bondades de una computadora de escritorio con soporte para distintos sistemas operativos, en el que se pueden realizar configuraciones e instalaciones con pocas limitaciones a diferencia de otros dispositivos como teléfonos, tabletas, entre otros. Lo que permite la ejecución y prueba de diferentes algoritmos de enrutamiento y aplicaciones para redes VANETs (gran cantidad de ellas desarrolladas para funcionar en computadoras de escritorios y no dispositivos móviles), sin necesidad de realizar cambios importantes en el dispositivo.

IEEE 802.11p es el estándar definido para trabajar en el entorno de redes VANETs, pero en la actualidad no se cuenta con una implementación comercial de esta tecnología. Sin embargo, aun sin poder contar con el estándar IEEE 802.11p es posible utilizar una aproximación de la misma haciendo modificaciones al controlador *open source* para interfaces inalámbricas *ath5k* de ciertas versiones del kernel de GNU/LINUX. Con esta aproximación se puede hacer uso de varias características de IEEE 802.11p como por ejemplo utilizar la banda de frecuencias de 5.9 GHz para la comunicación en redes VANETs.

Se puede decir que los objetivos planteados en la solución propuesta se cumplieron en su totalidad, teniendo como resultado un prototipo de dispositivo móvil que dispone de los componentes de hardware y software para permitir la ejecución de protocolos de enrutamiento y el soporte de aplicaciones para la seguridad vial en redes VANETs.

## 7.2. Limitaciones

Para la elaboración de este trabajo se presentaron varias limitaciones. En primer lugar, se cuenta con una reducida cantidad de componentes de hardware debido a las dificultades para adquirir productos provenientes del exterior, por lo cual, el diseño de la solución debió ajustarse a las componentes de hardware que se pueden conseguir en el país y en las pocas tiendas en el exterior en las cuales se pueden comprar productos desde Venezuela. En segundo lugar, el presupuesto para la adquisición de componentes fue reducido.

Otra dificultad se presentó durante la realización de las pruebas fijas por la falta de herramientas para seguir la ruta de los mensajes enviados en redes Ad hoc, como la herramienta *traceroute* utilizada en redes LAN para seguir la pista de un mensaje enviado desde un punto de la red a otro. También se presentaron inconvenientes para configurar las topologías de red planteadas en las pruebas fijas, debido a la cercanía de los nodos dentro del laboratorio.

En la prueba de cobertura de la señal no pudieron utilizarse las herramientas empleadas comúnmente para ese propósito como *kismet*, *wavemon*, entre otras, porque no funcionan con el estándar IEEE 802.11p.

## 7.3. Trabajos futuros

Algunas propuestas a considerar para posibles trabajos futuros son las siguientes:

1. Hacer un estudio comparativo de las tecnologías IEEE 802.11a y la implementación aproximada de IEEE 802.11p con el controlador modificado *ath5k*, para evaluar el rendimiento de ellas en ambientes reales y observar si realmente existe una ventaja favorable de una tecnología sobre la otra para el entorno de redes VANETs.

2. Realizar pruebas en ambientes reales con varios vehículos en movimiento para evaluar el comportamiento de una red VANET en presencia de varias OBU.

3. Probar distintos protocolos de enrutamiento para redes VANETs, cuyas implementaciones se pueden instalar con facilidad en el dispositivo construido.

4. Probar otras topologías de red más complejas.

## Referencias

- [01] X. Wang. Mobile Ad Hoc Networks: Applications. Intech. 2011
- [02] A. Caicedo, J Martínez, G Agredo. Evaluación del Desempeño de Redes 802.11p/WAVE en la Transmisión de Datos, Voz y Video IP. Universidad del Cauca. Popayán. 2011.
- [03] A. Sajjad, R. Asim, Q. Amir. VANET Architectures and Protocol Stacks: A Survey. Universidad Muhammad Ali Jinnah. Islamabad. 2011.
- [04] De Jongh. Jan. GCDC Communication Stack - Building and installing the CALM FAST router. 2011.
- [05] GCDC, Grand Cooperative Driving Challenge – FAQ. <http://www.gcdc.net/servicemenu/faq>. Agosto, 2013.
- [06] GCDC, Grand Cooperative Driving Challenge – Communication Stack. [http://www.gcdc.net/mainmenu/Home/technology/Communication\\_Stack](http://www.gcdc.net/mainmenu/Home/technology/Communication_Stack). Agosto, 2013.
- [07] Compat-wireless. Stable – Linux Wireless. <http://wireless.kernel.org/en/users/Download/stable>. Agosto, 2013.
- [08] Debian Live. Live System Project. <http://live.debian.net/>
- [09] TOUKABRI. T. CarGeo6 Project “Development results and user guide”. SPIRIT-INRIA. Francia. Junio. 2011.
- [10] Unetbootin. UNetbootin – Home Page and Downloads. <http://unetbootin.sourceforge.net/>
- [11] Downloadosmtiles.pl. downloadosmtiles.pl – search.cpan.org. <http://search.cpan.org/~rotkraut/Geo-OSM-Tiles-0.02/downloadosmtiles.pl>
- [12] OpenStreetMap. <http://www.openstreetmap.org/>
- [13] Intel D525MW. <http://www.intel.com/content/www/xl/es/motherboards/desktop-motherboards/desktop-board-d525mw.html>
- [14] D2.2 Final GeoNet Specification.

- [15] Prasad. A, Buford. J, Vijay. K. Future Internet Services and Service Architectures. Dinamarca. Junio 2011.
- [16] Moreno. J, Herrera. J. Propuesta de Desarrollo de un Dispositivo que Soporte Comunicación entre Vehículos conectados vía una VANET. Caracas. Junio. 2011.
- [17] TOUKABRI. T. CarGeo6 Project “Development results and user guide”. SPIRIT-INRIA. Francia. Junio. 2011.
- [18] D7.1 GeoNet Experimentation Results.
- [19] Iwlist. [http://linuxcommand.org/man\\_pages/iwlist8.html](http://linuxcommand.org/man_pages/iwlist8.html)