

Resumen

Este proyecto tiene el objetivo de desarrollar servicios de movilidad cooperativa, a través de redes híbridas, para mejorar la seguridad a los Vulnerable Road User (VRU) proporcionando la información necesaria a los vehículos que compartan el mismo medio. Para recoger la información relevante, se debe combinar información centralizada en la nube y tecnologías distribuidas. Para la comunicación entre usuarios se ha planeado emplear redes de corto alcance (IEEE 802.11p) y largo alcance (Long Term Evolution (LTE)), ya que son las que permiten una rápida adopción de las aplicaciones de movilidad.

Para la realización del proyecto, se requiere diseñar y desarrollar varias aplicaciones: una plataforma en la nube que permita la comunicación entre Smartphone y unidades vehiculares, así como algoritmos para la detección de situaciones peligrosas (aquellas en las que los vehículos puedan colisionar) y avisar a los usuarios involucrados. Así mismo, se requiere de aplicaciones en todos los actores involucrados en el proyecto (vehículos, ciclistas y unidades desplegadas en carretera).

Para verificar el sistema se deben realizar pruebas en un entorno controlado, en el cual debe pasar un plan de pruebas definido: comunicación eficiente entre los usuarios, facilidad de uso, rendimiento de la plataforma...

Descriptores

redes vehiculares, agentes vulnerables, ciudades inteligentes.

Índice general

1	Introducción	1
1.1	Antecedentes	3
2	Alcance	5
3	Definición de objetivos	7
4	Producto final	9
5	Metodología	11
5.1	Tareas	11
5.1.1	T1 - Requisitos	11
5.1.2	T2 - Nube de conductores	12
5.1.3	T3 - App de ciclistas	12
5.1.4	T4 - Aplicaciones vehiculares	12
5.1.5	T5 - Validación	12
5.1.6	T6 - Estudios adjuntos	12
5.1.7	T7 - Grabación de DEMO	13
6	Planificación: fases y calendario	15
7	Presupuestos	19
8	Desarrollo	23
8.1	Arquitectura del sistema	24
8.1.1	Comunicación entre plataformas	26
8.1.2	Formato de los mensajes	27
8.2	Nube de Conductores	29
8.2.1	Procesos	29
8.3	Aplicación de ciclistas	30

8.3.1	Modos de funcionamiento	30
8.3.2	Comunicación con la nube	33
8.3.3	Comunicación del grupo	34
8.3.4	Casco BLE	35
8.4	Comunicación vehicular	37
8.4.1	Unidad de a bordo	37
8.4.2	Unidad de carretera	38
9	Resultados y pruebas	41
10	Conclusiones	43
A	Apéndice	45
A.1	Código fuente del funcionamiento de mensajes GCM	45
A.2	Código fuente Casco BLE	45
A.3	Posición vehicular relativa	47
A.4	Posición vehicular relativa	48
A.4.1	GATT	48
A.4.2	UUID	48
A.5	Open-XC	48

Índice de figuras

Capítulo 2

2.1 Diagrama de desglose de trabajo	6
---	---

Capítulo 6

6.1 Estrategia del desarrollo	16
6.2 Diagrama de Gantt	18

Capítulo 8

8.1 Casos de uso	24
8.2 Arquitectura del sistema	25
8.3 <i>Hub</i> del líder	31
8.4 Ingreso al grupo del seguidor	31
8.5 UI de la salida	32
8.6 Controlador de la salida	33
8.7 Estructura de la comunicación GCM	34
8.8 Comunicación líder-seguidor	35
8.9 UI de la aplicación instalada en el HMI	38

Capítulo A

A.1 Rumbo: el ángulo 0° está desplazado 90° con respecto al eje cartesiano.	47
A.2 Posición relativa vehicular.	47

Índice de tablas

Capítulo 7

7.1	Costes de material y equipo	20
7.2	Costes laborales	20
7.3	Costes de subcontrataciones	20
7.4	Resumen de costes	21

Capítulo 8

8.1	Formato de mensaje Vehículo a Motor	28
8.2	Formato de mensaje Ciclista	28
8.3	Formato de mensaje Ciclista	29
8.4	Tipo de mensajes en grupo	34
8.5	Tabla de la verdad de señales LED	37

Índice de algoritmos

Capítulo 8

8.1 Envío de mensajes mediante GCM	27
8.2 Formato de mensajes	27
8.3 Cálculo de la proximidad de los vehículos	30
8.4 Envío de peticiones desde la aplicación de ciclistas a la Nube de Ciclistas	33
8.5 Envío de mensajes LED desde la aplicación de ciclistas	36

Capítulo A

A.1 Envío de mensajes mediante GCM	45
A.2 Declaración del servicio LED	46
A.3 Implementación del callback para el servicio LED	46
A.4 Cálculo de la posición relativa vehicular.	47

1. INTRODUCCIÓN

Según la Directiva General de la Comisión Europea para el Transporte y la Movilidad, en 2014, algo más de 25.700 accidentes de tráfico fueron informados en la Unión Europea. Aunque el número de accidentes está decreciendo sustancialmente, el informe sobre Transporte ha anunciado un objetivo estratégico para la seguridad en las carreteras europeas para el período de 2011 a 2020: reducir el número de muertes en carretera a la mitad. Si las estadísticas son analizadas, en el período de 2010 a 2012, el número de ciclistas muertos en siniestros ha aumentado un 6 %; siendo el único agente de la carretera cuyos resultados vayan a peor. Esto se explica, al menos parcialmente, por un aumento de la presencia ciclista en la carretera. Se podría decir que el ciclismo es un medio de transporte donde los VRU tienen un mayor contacto con el tráfico de mayor afluencia y velocidad. Cuando están involucrados en un accidente, son los que sufren las consecuencias más graves derivadas de una colisión con otro agente de la carretera; ya que están completamente expuestos a otros vehículos.

Basados en estos estudios, los accidentes en la que están involucrados VRUs ocurren frecuentemente en vías diseñadas para viandantes y ciclistas; por ejemplo en pasos de peatones y caminos para ciclistas cercanos a infraestructuras comunes de tráfico, las carreteras. Por lo tanto, la pregunta es: ¿Cómo se pueden reducir los accidentes de VRUs, y cómo minimizar la gravedad de un siniestro y sus consecuencias? Se pueden tomar varias soluciones: mejorar el diseño y trazado de las vías de comunicación, mejorar la iluminación, instalar más infraestructuras de protección, promocionar equipamiento de seguridad y enseñar cómo utilizarlo...

Sin embargo, hay otras soluciones viables aparte del re-diseño de las infraestructuras existentes, o soluciones pasivas como el uso del casco de seguridad. Una opción que está ganando fuerza en los países desarrollados es el desarrollo de soluciones para la movilidad en el entorno de ciudades inteligentes. Aunque el término de ciudad inteligente pueda parecer confuso, se podría decir que se considera *inteligente* cuando se ha aplicado tecnologías de la información y comunicación para mejorar la calidad de vida en áreas como la seguridad, gasto energético, reducción de costes, y gobierno y transporte, permitiendo una participación efectiva y activa por parte de los ciudadanos.

En el dominio de las Ciudades Inteligentes, las soluciones para transporte diseñadas tratan de hacer un uso más seguro, sostenible y eficiente de la carretera a través de un mejor entendimiento del estado de tráfico, la posición de los vehículos y usuarios, y el registro de eventos que suceden durante el transporte. Estas soluciones combinan la capacidad y beneficios de los sensores, dispositivos, infraestructura física y arquitecturas de comunicación combinada con sistemas de información en la nube, y la capacidad de analizar grandes volúmenes de datos.

En este contexto, los Sistemas Inteligentes de Información (ITS) emergen como una respuesta tecnológica para una mejor monitorización y caracterización del tráfico. Estos sistemas permiten al mismo tiempo mejorar el uso y eficiencia de la carretera, así como la seguridad de los usuarios, particularmente aquellos definidos como vulnerables; ciclistas, peatones o motoristas. Los ITS

1. INTRODUCCIÓN

actuales requieren el uso de cámaras de tráfico, paneles informativos, o sensores de inducción que obtengan datos para ser posteriormente mandados y procesados en la central de gestión de tráfico. A diferencia de estas soluciones que requieren el uso de sensores, actualmente surgen sistemas conocidos como Información Vehicular Flotante (FCD), que se encargan de reunir información de los sistemas de posicionamiento global (GPS) obtenidos de terminales móviles y el uso de páginas web colaborativas como Waze, que permite a los conductores obtener y proveer información sobre la carretera sin necesidad de ningún sensor en la carretera. Este tipo de soluciones basadas en FCD tienen a favor la monitorización del estado de los usuarios de manera ubicua, pero su fiabilidad depende del número de vehículos y usuarios informando sobre los eventos y aportando datos.

En el dominio de los ITS, los ITS Cooperativos (C-ITS) son sistemas que permiten la conexión directa entre vehículos (comunicaciones V2V) ó entre vehículos e infraestructuras (comunicaciones V2I) para intercambiar de información con el objetivo de mejorar la seguridad vial y la gestión del tráfico. Estos enlaces son posibles gracias a las unidades de abordo (OBU), dispositivos C-ITS dedicados que habilitan interfaces de comunicación, y dispositivos localizados en infraestructuras llamados RSU.

En un escenario C-ITS, hay generalmente cuatro agentes a considerar: dos entidades móviles (OBUs y peatones), y dos entidades estacionarias (la RSU y el sistema central). Estas entidades son capaces de ejecutar cuatro tipos diferentes de aplicaciones: seguridad activa en la carretera, tráfico eficaz cooperativo, servicios locales cooperativos, y servicios globales en Internet. Sobre cada tipo, hay diferentes definiciones de casos de uso y aplicaciones, donde cada agente puede ser considerado un sensor que genera información. Dependiendo de la aplicación y las restricciones de tiempo, el intercambio de información entre las entidades se puede clasificar como:

Mensajes de alerta: se definen como notificaciones descentralizadas y pueden ser enviados desde cada vehículo o RSU.

Mensajes de latido o *"beacons"*: son usados por los OBU para notificar su posición, velocidad e identidad a las RSU que forman parte del FCD. Además, estos mensajes también son usados para conocer la situación actual del tráfico. Por ello, el Acceso Inalámbrico en Entornos Vehiculares (WAVE) define los Mensajes de Aviso Cooperativo (CAMs), que son transmitidos periódicamente a todos los vehículos en área de alcance.

Mensajes sobre infotainment: notificaciones no relacionadas con la seguridad, sino que son usados para aportar mayor información y confort al conductor; datos turísticos, acceso a Internet, asistencia en navegación, etc.

En el campo de los servicios C-ITS, una gran variedad de aplicaciones y casos de uso se centran en incrementar la seguridad del usuario. Teniendo en cuenta requisitos estratégicos, económicos y de organización, características del sistema así como requisitos legales y de estandarización, el Comité del Instituto Técnico Europeo de Estándares en la Telecomunicación ha definido un conjunto básico de aplicaciones para usar como referencia en ITS para desarrolladores. Entre ellos, los Avisos a Usuarios Vulnerables en Carretera tratan de proveer notificaciones a los vehículos sobre la presencia de usuarios vulnerables, por ejemplo ciclistas, y en caso de existir situaciones de peligro también se avisa a los VRU sobre la presencia de un vehículo cercano.

Siguiendo los requisitos presentados por el ETSI, este proyecto presenta un sistema que emplea a los vehículos y ciclistas como sensores móviles que aportan información sobre su posición, velocidad y rumbo con el objetivo de detectar la proximidad entre estas dos entidades y avisarles en el caso de detectar peligro. Esta solución tiene un sistema centralizado que despliega comunicación inalámbrica vehicular, conectividad móvil y computación en la nube, y gestiona la información obtenida por los usuarios (vehículos y ciclistas). El sistema ha sido desplegado y verificado en un dominio real, y se han realizado pruebas de rendimiento en diferentes escenarios para comprobar el correcto funcionamiento de las comunicaciones en los peores escenarios.

Actualmente, existe un proyecto llamado Detección de agentes Inteligentes Cooperativos para la mejora de la eficiencia en el tráfico (ICSI), con similares aplicaciones que se encuentra bajo desarrollo usando una proximación descentralizada con la finalidad de mejorar el rendimiento y la seguridad.

1.1. ANTECEDENTES

La importancia de las tecnologías C-ITS para la administración pública y la Comisión Europea está reflejada en la directiva 2010/40/EU, donde la UE reconoce la capacidad de los C-ITS para mejorar los sistemas de gestión de tráfico actuales y de dirigir los procesos de implantación y despliegue de estos sistemas en las carreteras europeas. Tras decenas de investigaciones y proyectos de desarrollo tales como Sistemas Cooperativos Vehículo-Infraestructura (CVIS), Sistemas Cooperativos para la seguridad en carretera *vehículos inteligentes en ciudades inteligentes* (SAFESPOT), El transporte flexible y adaptable del mañana (TEAM), el despliegue masivo de sistemas C-ITS se está acercando. Un ejemplo es el Memorandum del Entendimiento (MOU) firmado por la industria automovilística y organizaciones constructoras con el objetivo de comenzar a desplegar soluciones basadas en C-ITS en 2015. Las administraciones públicas también están trabajando en la misma dirección, resaltando el tratado que han pactado Alemania, Austria y Holanda para desplegar sistemas C-ITS en las vías que comunican estos tres países.

Se puede destacar de la misma manera el anuncio realizado en Febrero del 2014 por la Administración Nacional de Seguridad del Tráfico en Autopistas (NHTSA), que pertenece al Departamento de Transporte de los Estados Unidos (USDOT), sus intenciones para dar los pasos necesarios para el despliegue de sistemas V2V cooperativos en los próximos años, exactamente a partir de 2017, para vehículos comerciales.

2. ALCANCE

La mejora en la miniaturización de la tecnología, y la calidad de los servicios de comunicación el concepto de ciudad inteligente es más posible de lograr que nunca. Este proyecto propone el diseño, desarrollo y validación de un escenario de movilidad cooperativa, segura y eficiente en la cual los usuarios (conductores, pasajeros y ciclistas) dispongan de una serie de aplicaciones, que proporcionen información y reciban información; la cual es procesada en tiempo real. Son tres los retos que afronta este proyecto:

1. Diseñar una infraestructura de comunicaciones dinámica, y que integre diferentes tecnologías de comunicaciones de corto (IEEE 802.11p) y largo alcance (LTE).
2. Procesar la información generada por todos los usuarios y agentes en la infraestructura.
3. Promover la participación de todos los usuarios.

Para lograrlo, este proyecto combina información centralizada en la nube y tecnologías de computación distribuida con tecnologías de comunicaciones, donde las comunicaciones Vehicle to X (V2X) y las comunicaciones móviles son utilizadas indistintamente para proporcionar información relevante a todos los usuarios. Gracias a ello, este proyecto favorecerá un entorno donde los vehículos, Smartphone y/o Tablet podrán operar entre ellas para proveer servicios de movilidad a los usuarios.

Este proyecto, basándose en la arquitectura Intelligent Transport System (ITS) definida por ITS EN 302 665, integrará información procedente de elementos de infraestructura, vehículos, terminales móviles con servicios desplegados en la nube y ofrecidos a los usuarios, en una red colaborativa capaz de diferenciar necesidades individuales y globales. Para ello, los datos recogidos serán procesados por algoritmos preparados para procesar grandes volúmenes de información, y que tendrán en cuenta los efectos globales de sus acciones, para lo cual se dispondrá de enlace de control para regular los sistemas de un modo descentralizado. Por tanto, el sistema deberá responder dinámicamente las necesidades de los usuarios.

Se ha seleccionado la tecnología LTE ya que se ha establecido como la siguiente generación de comunicaciones móviles, disponible masivamente y favorecerá el despliegue de ITS. De este modo, la integración de tecnologías de corto alcance como IEEE 802.11p con LTE es obligatorio para la rápida adopción de aplicaciones para la movilidad.

Para validar este proyecto, se implementarán las siguientes aplicaciones:

- Intersecciones seguras: los vehículos próximos a una intersección intercambiarán su destino en la misma, avisando de su presencia para evitar accidentes.
- Navegación segura: los usuarios vulnerables de la carretera podrán reportar su posición a los vehículos que les rodeen y al mismo tiempo ser alertados de vehículos que se aproximen.
- Conducción colaborativa: se desplegarán una serie de aplicaciones destinadas a mejorar la eficiencia de las infraestructuras y su seguridad gracias a los datos intercambiados entre

2. ALCANCE

todos los usuarios y elementos fijos: frenada de emergencia, cambio de carril, velocidad adaptativa, etc.

Se ha estimado la realización del proyecto en un plazo inicial de 2 años, comenzando en octubre, 2014 y finalizando en octubre de 2016. En la figura 2.1 se muestra la división inicial de las actividades del proyecto.

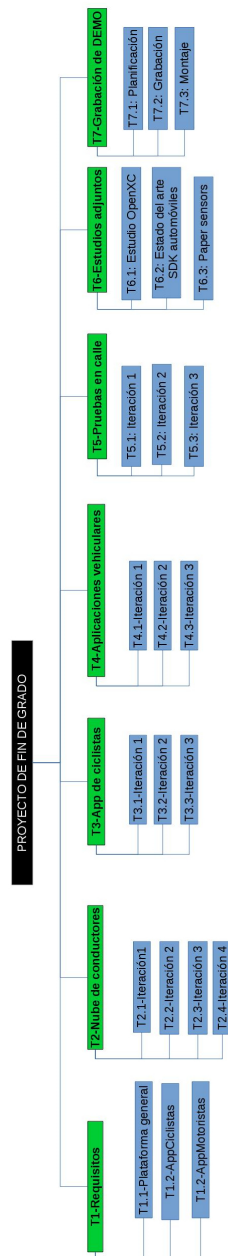


Figura 2.1: Diagrama de desglose de trabajo

3. DEFINICIÓN DE OBJETIVOS

El principal objetivo de este trabajo es el desarrollo de aplicaciones para hacer posible la comunicación entre vehículos y ciclistas, a través de redes vehiculares y móviles. Se desea que estos dos agentes de la carretera posean información para poder conocer la posición de los agentes a su alrededor y así poder evitar accidentes. Las aplicaciones desarrolladas serán desplegadas por un lado en los sistemas informáticos de los vehículos y la infraestructura en la carretera, en los móviles inteligentes de los ciclistas y en la nube, donde un servidor central permitirá la comunicación entre diferentes plataformas. Se busca la creación de una plataforma lo más abierta posible a diferentes tecnologías, así como el uso de software libre para su desarrollo.

Los resultados que permitan verificar el sistema se harán a través de las pruebas unitarias que se han creado para probar el sistema, las pruebas que se han realizado en la calle y a las conclusiones que se han llegado tras analizar las mediciones obtenidas. Se desea verificar la calidad de las comunicaciones entre los diferentes agentes de la carretera, así como la precisión de los algoritmos desarrollados para prever los accidentes y obtener información a través de los datos recogidos por los mensajes vehiculares. Para saber más sobre cómo se han verificado los resultados del proyecto ir al capítulo 9.

Las aplicaciones que serán desarrolladas deben poseer una serie de características:

- Universalidad: el sistema a desarrollar debe ser flexible a diferentes tecnologías. Actualmente existen diversos fabricantes que aún no cumplen los estándares que se están estableciendo, por lo que se debe permitir que la adaptación de los diferentes sistemas sea lo más sencillo posible.
- Interfaz gráfica: la interacción del usuario con las aplicaciones debe ser sencilla, que no requiera de ningún conocimiento técnico para realizar sus funciones.
- Localización de agentes en carretera: los vehículos y ciclistas intercambian información sobre sus posiciones a través de mensajes CAM.
- Predicción de accidentes: los usuarios deben poder percibir las situaciones de peligro. En el caso de los vehículos a motor se reproduce una alarma sonora, y en el de los ciclistas se enciende un LED en el casco que llevan puestos, además de mostrarse la alerta en el Smartphone; en caso de tenerlo a la vista.

4. PRODUCTO FINAL

El producto final se divide en tres grupos de aplicaciones que están presentes en diferentes agentes de la carretera. El primer grupo se trata del sistema en la nube que se encarga de comunicar el lado de vehículos a motor y ciclistas. Además, detecta situaciones de peligro y pone en aviso a cada agente implicado.

El segundo grupo se trata de todas las aplicaciones encargadas de obtener información en la red vehicular, y recibir la información generada en la nube: Unidad de carretera (RSU), Unidad de a bordo (OBU) y Human-Machine Interface (HMI).

Finalmente, el último grupo es la aplicación de ciclistas y servidor Google Cloud Messaging (GCM) que provee notificaciones a los ciclistas sobre los eventos en carretera, y al mismo tiempo permite a los ciclistas mandar información sobre su posición. Además del casco que permite notificar al ciclista de un peligro inminente a través de leds instalados en el casco de seguridad.

Así mismo, a través de las pruebas realizadas, los resultados obtenidos y las conclusiones generadas, se puede generar diferentes estudios que motiven nuevos desarrollos dentro del dominio de los VRU.

5. METODOLOGÍA

En el capítulo 2 se puede observar el EDT que ha sido seguido para la realización de este proyecto. Para el desarrollo de cada producto se sigue una metodología de desarrollo iterativa e incremental. Al final de cada iteración se genera un producto intermedio, con las características acordadas durante el que funcione correctamente con el resto de productos. A continuación se detallan las fases principales del proyecto:

- **T1 - Requisitos:** recolección y análisis de requisitos. Organización de requisitos funcionales y no funcionales.
- **T2 - Nube de conductores:** incluye todas las actividades para el desarrollo de la aplicación de conductores que será desplegada en la nube.
- **T3 - App de ciclistas:** actividades para el desarrollo de la aplicación de ciclistas y el casco Bluetooth Low Energy (BLE).
- **T4 - Aplicaciones vehiculares:** actividades para el desarrollo de la RSU, OBU y .
- **T5 - Pruebas en la calle:** pruebas que se han desarrollado en la calle para validar diferentes partes del proyecto.
- **T6 - Grabación de DEMO:** grabación de una demo del funcionamiento del proyecto completo.
- **T7 - Estudios adjuntos:** estudios realizados para el desarrollo de diferentes áreas del proyecto.

Las iteraciones se realizan dentro de las tareas T2, T3 y T4. En cada uno de las iteraciones se realiza cada una de las siguientes actividades:

- **Diseño:** selección de las especificaciones a implementar y realización del diseño de las funciones a implementar en la aplicación.
- **Desarrollo y depuración:** desarrollo y depuración de las funciones diseñadas.
- **Documentación:** generación de la documentación de toda la fase (comentarios en código, documentos explicativos, prototipos generados...).

5.1. TAREAS

5.1.1. T1 - Requisitos

- **Plataforma general:** requisitos de la aplicación en la nube y las aplicaciones vehiculares básicas (OBU y RSU).
- **AppCiclistas:** requisitos de la aplicación para los ciclistas y el casco BLE.
- **AppMotoristas:** requisitos de la aplicación para el HMI de los vehículos.

5.1.2. T2 - Nube de conductores

- **Iteración 1:** implementación de la arquitectura básica, base de datos y acceso a través de sockets.
- **Iteración 2:** mejora del formato de mensajes utilizado en la comunicación, cambio del acceso por sockets a servlets, e inclusión de algoritmos de predicción de accidentes.
- **Iteración 3:** cambiada la comunicación con la aplicación de ciclistas a GCM.
- **Iteración 4:** optimización de la plataforma.

5.1.3. T3 - App de ciclistas

- **Iteración 1:** desarrollo de la base de la aplicación; incluye salidas individuales y en grupo.
- **Iteración 2:** mejora de la sensibilidad del GPS, prueba con mensajes UDP en salidas en grupo, cambio de comunicación a la nube a través de mensajes HTTP/1.1 y soporte a versiones antiguas de Android.
- **Iteración 3:** empleo de GCM para la recepción de mensajes y desarrollo del casco BLE.

5.1.4. T4 - Aplicaciones vehiculares

- **Iteración 1:** base de las aplicaciones OBU y RSU.
- **Iteración 2:** cambio de comunicación a servlets y mejora de los mensajes empleados en la comunicación.
- **Iteración 3:** optimización de la plataforma e implementación de la aplicación para el HMI.

5.1.5. T5 - Validación

Durante cada iteración de la validación se desarrollan tres actividades: una primera para planificar qué pruebas se desean hacer y qué datos se desean obtener, así como preparar la aplicación para pruebas. Durante la segunda actividad se desarrollan las pruebas. Y finalmente, la tercera actividad, se recogen y estudian los resultados y se proponen mejoras para futuros desarrollos.

- **Iteración 1:** pruebas de rendimiento en comunicaciones V2X.
- **Iteración 2:** comunicación Vehicle to vehicle (V2V) en entornos urbanos.
- **Iteración 3:** prueba del sistema completo.

5.1.6. T6 - Estudios adjuntos

- **Estudio de OpenXC:** se desarrolla un estudio sobre la plataforma OpenXC y la posibilidad de su uso en el proyecto.
- **Estado del arte de Unificación de SDK:** se estudia la creación de un subproyecto para desarrollar aplicaciones en el HMI de los vehículos en diferentes plataformas; AndroidAuto, CarPlay...
- **Paper para sensors:** desarrollo de un artículo para MDPI sobre comunicaciones vehiculares y su uso sobre VRU.

5.1.7. T7 - Grabación de DEMO

- **Planificación:** Requisitos generales para la grabación: selección de personas que ayuden durante el rodaje, selección de escenas y escenarios, materiales, fechas...
- **Grabación.**
- **Montaje:** montaje del vídeo y post-producción.

6. PLANIFICACIÓN: FASES Y CALENDARIO

En el siguiente apartado se describe cuál ha sido la planificación del proyecto, y cómo se ha organizado. Esta planificación se ha desarrollado para poder mantener un ritmo continuo de desarrollo de tipo ágil, y conseguir un aprovechamiento máximo del proyecto. Durante el desarrollo de este proyecto se ha realizado paralelamente pequeños proyectos, por lo que era necesario que se pudiesen observar los resultados del desarrollo de la forma más inmediata posible. Para ello, como se ha explicado en el capítulo 5, se realizan pequeños prototipos de una funcionalidad que más adelante son integradas en el proyecto.

La estrategia seguida durante el proyecto puede observarse en la imagen 6.1. Inicialmente se reúnen los requisitos necesarios y se realiza un diseño de las funcionalidades a incluir, se producen varias iteraciones hasta que todos los requisitos son satisfechos. Una vez finalizado el diseño se pasa al desarrollo, el cual incluye depuración y pruebas unitarias. Cuando el desarrollo ha finalizado correctamente, se pueden añadir nuevos requisitos o pasar a pruebas en calle si han sido programadas con anterioridad; algunas pruebas requieren de adaptar parte del programa al carácter de las pruebas. Una vez finalizadas las pruebas, si se han descubierto fallos en el software, es reparado y si se decide que es requerido ampliar la plataforma se vuelve a iniciar el ciclo.

6. PLANIFICACIÓN: FASES Y CALENDARIO

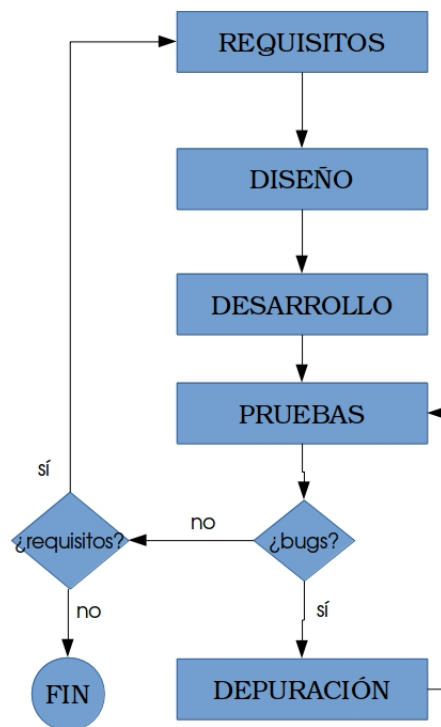


Figura 6.1: Estrategia del desarrollo

En la figura 6.2 se presenta un diagrama Gantt en el que se puede observar las diferentes tareas y fases por las que ha pasado el proyecto, presentando las correspondientes fechas de inicio y fin que se han estimado durante la ejecución del proyecto. Al tratarse de un proyecto incremental mediante generación de prototipos, se han realizado varias iteraciones durante el desarrollo.

Las fases del proyecto se puede dividir en los siguientes bloques:

- Recolección de requisitos del sistema y estudio de la tecnología necesaria.
- Desarrollo de la Nube de conductores.
- Desarrollo de la aplicación para ciclistas.
- Desarrollo de la RSU, OBU y HMI.
- Verificación del sistema en la calle.
- Estudio de los resultados y generar documentación.

Los hitos del proyecto son las distintas pruebas que se han hecho de funcionalidades diferentes del sistema. Durante estas pruebas se analiza el rendimiento que se obtiene de la comunicación del sistema, se proponen y añaden mejoras que serán incluidas en siguientes pruebas.

6. PLANIFICACIÓN: FASES Y CALENDARIO

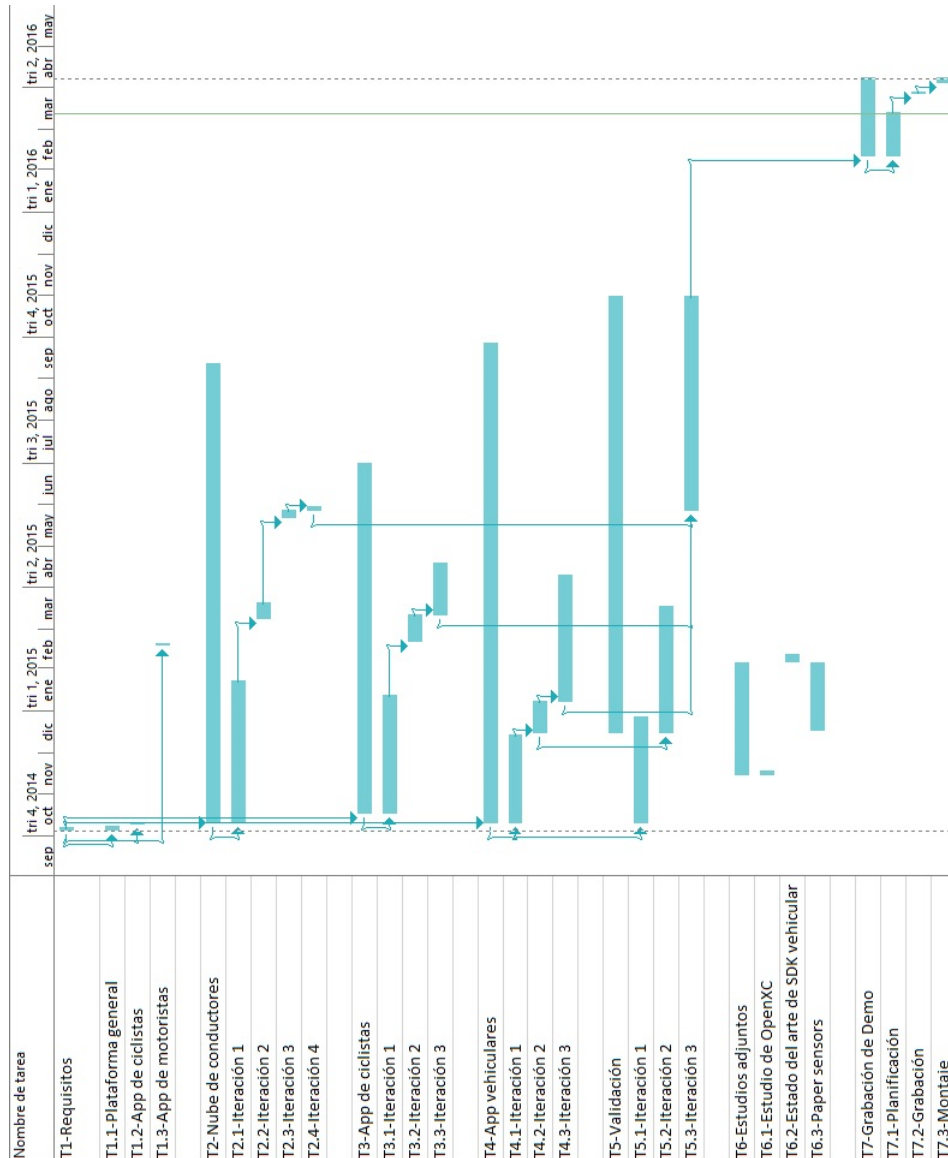


Figura 6.2: Diagrama de Gantt

7. PRESUPUESTOS

Se ha realizado una recopilación de todos los gastos que se han dado en el proyecto. Estos costes tienen que ver con la adquisición de equipo, contratación de diferentes servicios y software para ser posible el estudio, desarrollo, verificación y documentación del sistema.

Se requiere que los gastos sean lo más bajos posibles, ya que se trata de un proyecto para la investigación y no está pensado para ser comercializado. Debido a ello se han empleado plataformas Open Source para el desarrollo y despliegue de las aplicaciones, ya que no generan ningún tipo de gasto. Al desarrollarse el proyecto bajo amparo de la Universidad de Deusto, existen gastos de instalaciones, mantenimiento, servicios y plataformas que son nulos; como por ejemplo el acceso repositorios Git.

Los beneficios de esta investigación se traducen en el impacto que tengan los artículos que puedan ser generados y publicados en diferentes revistas, diarios...

Los costes se han dividido en tres categorías diferentes: los gastos de material y equipo necesario [7.1], costes laborales del personal [7.2] y el coste de otros proyectos subcontratados [7.3]. Finalmente, se presenta un resumen de todos los gastos [7.4].

Tabla 7.1: Costes de material y equipo

Concepto	Tipo	Unidades	Coste por unidad	Coste total
NEC Linkbird MX	Hardware	2	2.960,00 EUR	5.920,00 EUR
Dell Latitude D520	Hardware	1	134,75 EUR	134,75 EUR
Acer Aspire 4810TZG	Hardware	1	596,87 EUR	596,87 EUR
Estación de trabajo	Hardware	1	799,00 EUR	799,00 EUR
Cable UTP	Hardware	2	0,00 EUR	0,00 EUR
Adaptador RS-232 Hembra a Macho	Hardware	2	1,00 EUR	2,00 EUR
Adaptador RS-232 Null	Hardware	2	1,00 EUR	2,00 EUR
HI 204-III GPS USB	Hardware	2	59,00 EUR	59,00 EUR
Fuente de alimentación Belkin	Hardware	2	50,00 EUR	100,00 EUR
Casco BLE + Texas CC2540	Hardware	1	65,00 EUR	65,00 EUR
Sistemas operativos: Ubuntu y Debian	Software	3	0,00 EUR	0,00 EUR
Sistema operativo Windows 7	Software	1	0,00 EUR	0,00 EUR
Software de desarrollo Java y C	Software	1	0,00 EUR	0,00 EUR
LaTeX	Software	1	0,00 EUR	0,00 EUR
LibreOffice	Software	1	0,00 EUR	0,00 EUR
Kit de desarrollo NEC	Software	1	0,00 EUR	0,00 EUR
Máquina virtual Ubuntu	Servicio	1	0,00 EUR	0,00 EUR
Dominio web	Servicio	1	0,00 EUR	0,00 EUR
Repositorio GitLab	Servicio	1	0,00 EUR	0,00 EUR
Google Cloud Messaging	Servicio	1	0,00 EUR	0,00 EUR
Google Maps	Servicio	1	0,00 EUR	0,00 EUR
				7,732.62 EUR

Tabla 7.2: Costes laborales

Concepto	Coste por hora	Horas	Coste total
Estudios módulos NEC.	5	20	100,00 EUR
Desarrollo de prototipos para pruebas.	5	20	100,00 EUR
Recolección de requisitos del sistema.	30	23	690,00 EUR
Formación en Android SDK.	5	20	100,00 EUR
Desarrollo de la Nube de Conductores.	5	114	570,00 EUR
Desarrollo de la OBU/RSU.	5	74	370,00 EUR
Desarrollo del HMI.	5	17	85,00 EUR
Desarrollo de CiclistasApp.	5	170	850,00 EUR
Pruebas en la calle.	30	43	1.290,00 EUR
Generar documentación.	30	55	1.650,00 EUR
			5.805,00 EUR

Tabla 7.3: Costes de subcontrataciones

Concepto	Coste total
PFG de Idoia de la Iglesia.	32.310,00 EUR
32.310,00 EUR	

Tabla 7.4: Resumen de costes

Concepto	Coste total
Hardware	7.737,62 EUR
Software	0,00 EUR
Servicios	0,00 EUR
Personal	5.805,00 EUR
Subcontrataciones	32.310,00 EUR
	45.852,62 EUR

A. APÉNDICE

A.1. CÓDIGO FUENTE DEL FUNCIONAMIENTO DE MENSAJES GCM

```

1  HttpURLConnection httpRequest;
2  final String KEY = AIzaSyAu2LXHx7_rP00UinzizQg5r5mgln4Q-Y;
3
4  try {
5      // abrir conexión con el gestor GCM
6      URL url = new URL("https://android.googleapis.com/gcm/send");
7
8      httpRequest = (HttpURLConnection) url.openConnection();
9
10     // enviar datos mediante POST
11     httpRequest.setRequestMethod("POST");
12
13     // establecer el encabezado
14     httpRequest.setRequestProperty("Content-Type", "application/json");
15     httpRequest.setRequestProperty("Authorization", "key=" + KEY);
16     httpRequest.setDoOutput(true);
17
18     // enviar mensaje y leer respuesta
19     [...]
20 } catch (IOException e) {
21     logger.error(e.getMessage());
22 }

```

Algoritmo A.1: Envío de mensajes mediante GCM

A.2. CÓDIGO FUENTE CASCO BLE

A. APÉNDICE

```
1 // Declarar el UUID del perfil ATT (Attribute Protocol)
2 #define PROFILE_VEHICULAR_POS 5
3
4 // Declarar el UUID del servicio
5 #define SIMPLEPROFILE_SERV_UUID 0xFFFD
6
7 //Declarar el UUID de la característica, darle un tamaño y asignar
8 //permisos de escritura
9 #define PROFILE_VEHICULARPOS_UUID 0xFFFD
10 CONST uint8 simpleVehicularPositionProfileUUID[ATT_BT_UUID_SIZE] = {
11 ^ILO_UINT16(PROFILE_VEHICULARPOS_UUID), HI_UINT16(PROFILE_VEHICULARPOS_UUID)
12 };
13 static uint8 vehicularPositionProps = GATT_PROP_WRITE;
14 static uint8 vehicularPosition = 0;
```

Algoritmo A.2: Declaración del servicio LED

```
1 // gestionar que se ha hecho una escritura en la característica
2 static bStatus_t simpleProfile_writeAttrCB(uint16 connHandle, gattAttribute_t *PAttr,
3     uint8 *pValue, uint8 len, uint16 offset) {
4     [...]
5     notifyApp = PROFILE_VEHICULARPOS;
6     [...]
7 }
8
9 // gestionar la petición
10 static void simpleProfileChangeCB(uint8 paramID) {
11     [...]
12     case PROFILE_VEHICULARPOS:
13         SimpleProfile_GetParameter(PROFILE_VEHICULARPOS, &newValue);
14         cambiarLED(paramID);
15     [...]
16 }
17
18 // escritura de la característica
19 bStatus_t SimpleProfile_SetParameter(uint8 param, uint8 len, void* value) {
20     [...]
21     case PROFILE_VEHICULARPOS:
22         if (len == SIMPLEPROFILE_CHAR5_LEN) {
23             vehicularPosition = *((uint8 *) value);
24         } else {
25             ret = bleInvalidRange;
26         }
27         break;
28     [...]
29 }
```

Algoritmo A.3: Implementación del callback para el servicio LED

A.3. POSICIÓN VEHICULAR RELATIVA

La posición vehicular relativa se refiere a en qué lado de un vehículo A se encuentra un vehículo B. Es decir, si se toma como referencia el primer vehículo, en qué lado se encuentra el segundo; derecha, izquierda, delante o detrás.

El "heading" o rumbo, es la dirección hacia la que se está dirigiendo un vehículo con respecto al Norte (0 grados). En la figura A.1, en el eje de coordenadas cartesiano. Cada vez que se realice una operación que involucre los ángulos, se emplea una función para mantenerlos en un rango de entre 0 a 360 grados.

El algoritmo A.4 mostrado se puede condensar en:

1. Calcular el ángulo existente entre los dos vehículos, sin tener en cuenta la dirección del primer vehículo.
2. Se resta el heading del vehículo de referencia al ángulo que hay entre los dos vehículos.
3. Se contrasta con una serie de casos ya conocidos, y se devuelve el ángulo relativo [Imagen A.2].

```

1 function calcularPosicionRelativa(heading, oLatitude, oLongitude, pLatitude, pLongitude) {
2   var degrees = calculateAngleBetweenTwoPoints(oLatitude, oLongitude, pLatitude, pLongitude);
3   var relativeAngle = correctDegrees(degrees - heading);
4
5   var tmp = "";
6   if (relativeAngle <= 15 || relativeAngle >= 345) {
7     return "FRONT";
8   } else if (relativeAngle >= 120 && relativeAngle <= 230) {
9     return "BACK";
10  } else if (relativeAngle < 120 && relativeAngle > 15) {
11    return "LEFT";
12  } else {
13    return "RIGHT";
14  }
15 }
16
17 function calcularAnguloEntreDosPuntos(ox, oy, x, y) {
18   return toDegrees(Math.atan2(y - oy, x - ox));
19 }

```

Algoritmo A.4: Cálculo de la posición relativa vehicular.

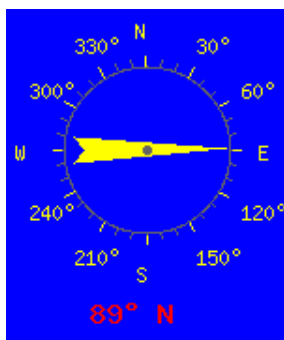


Figura A.1: Rumbo: el ángulo 0° está desplazado 90° con respecto al eje cartesiano.

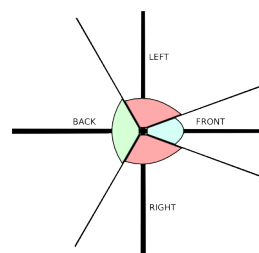


Figura A.2: Posición relativa vehicular.

A.4. POSICIÓN VEHICULAR RELATIVA

Bluetooth Low Energy. A diferencia del clásico Bluetooth, está diseñado para consumir una cantidad significativamente más pequeña de energía. Algunas diferencias durante el desarrollo respecto Bluetooth estándar a tener en cuenta en BLE son:

1. El dispositivo está continuamente durmiendo y despertándose para ahorrar batería.
2. La cantidad de información transmitida es pequeña, como máximo 216 bytes.
3. La transmisión de información se hace de manera rápida para poder poner el dispositivo a dormir tan pronto como se haya terminado de transmitir la información; latencias de hasta 2 ms por ráfaga.

A.4.1. GATT

Generic Attribute Profile. Establece cómo se va a transmitir la información sobre los perfiles y datos en la conexión BLE. GATT emplea el ATT (Attribute Protocol) como protocolo de transporte para intercambiar datos entre los dispositivos. Los datos están organizados jerárquicamente en secciones llamadas "servicios", los cuales tienen piezas relacionadas con ellos denominadas "características".

A.4.2. UUID

Universal Unique Identifier. Es un identificador de 128 bits que está garantizado que es único. Los UUID nos permiten identificar los servicios y características, además de poder operar con ellos.

A.5. OPEN-XC

Glosario de términos

3G Third Generation. 12

BLE Bluetooth Low Energy. 9

HTTP/1.1 Hypertext Transfer protocol. Protocolo de transferencia usado actualmente en la web (RFC2774).. 15

LTE Long Term Evolution. 12

OBD-II Se trata del sistema de diagnóstico del vehículo. Provee información sobre el estado de los diferentes subsistemas del vehículo.. 23

OBU Unidad de a bordo. 10, 11, 23, 24

RSU Unidad de carretera. 9–11, 23, 24

VRU Vulnerable Road User. 9

Glosario

3G Third Generation. 12

BLE Bluetooth Low Energy. 9

HTTP/1.1 Hypertext Transfer protocol. Protocolo de transferencia usado actualmente en la web (RFC2774).. 15

LTE Long Term Evolution. 12

OBD-II Se trata del sistema de diagnóstico del vehículo. Provee información sobre el estado de los diferentes subsistemas del vehículo.. 23

OBU Unidad de a bordo. 10, 11, 23, 24

RSU Unidad de carretera. 9–11, 23, 24

VRU Vulnerable Road User. 9

