

Grado en ingeniería Informática

Diseño e Implementación del protocolo de notificación electrónica basado en redes VANET para E-SAVE

Trabajo Fin de Grado



Autor:

Eduardo Ruiz Alonso

Tutores:

José María de Fuentes García-Romero de Tejada

Lorena González Manzano

21/06/2013

Agradecimientos

Este capítulo, por extraño que parezca, ha sido el último en terminar y el más duro de escribir. Nunca terminaré de agradecer a todos aquellos que comparten mi vida y me hacen crecer y evolucionar como persona día a día. He de agradecerles su apoyo y comprensión, no sólo durante la realización del proyecto, sino durante toda la carrera e incluso durante gran parte de mi vida.

En estos momentos recuerdo el comienzo del proyecto, cuando me introducía en las redes VANET. Siendo sincero, no sabía dónde me estaba metiendo. No por la dificultad del proyecto sino como por el tiempo dedicado a la investigación sobre nuevas tecnologías de las que no tenía ningún conocimiento. No engañaré a nadie diciendo que estos últimos años han sido un camino de rosas. Todo lo contrario, ha sido una experiencia larga y dura, en especial los últimos años, pero no por ello menos enriquecedora. Ha sido un camino por el que ha merecido la pena luchar y por el que he aprendido muchísimo.

En primer lugar, me gustaría mencionar a toda mi familia, quienes siempre han estado conmigo y me han animado a continuar con los estudios. Como mención especial, he de destacar a mis padres Mayte e Ignacio así como mi hermano Carlos, que durante toda mi vida me han apoyado y me animaban a mejorar en todos los aspectos de mi vida. A mis abuelos, gracias por hacerme un sitio en vuestra vida y en vuestro hogar cuando el tiempo era primordial para continuar con mis estudios, así como por enseñarme el valor del trabajo y del conocimiento.

No podría olvidarme de mis amigos y compañeros de la Universidad, Carlos, Javier, David, David, Carlos, Alejandro, Andrés, Francisco y otros muchos compañeros que han compartido conmigo una buena parte de la carrera. A todos ellos, gracias por los buenos momentos que hemos vivido. Sin vosotros no habría sido lo mismo.

Gracias a todos mis amigos, que en tiempos difíciles han sabido aguantarme y permanecer a mi lado. Gracias a grandes amigos como Sergio y Norma que siempre saben hacerme sonreír. Y un muy especial agradecimiento a Manuel, que, a pesar de las dificultades, siempre ha estado conmigo, y a Daniel, quien consigue alegrar el día a cualquiera. Ambos sois como hermanos para mí.

Me gustaría mencionar de forma muy especial a Irene. Gracias por estar conmigo, por estar ahí cuando más lo he necesitado, tanto en los momentos buenos como en los malos. Gracias por animarme y apoyarme. Gracias por hacerme tan feliz durante estos años. En definitiva, gracias por todo. Siempre consigues sacar lo mejor de mí.

Quiero agradecer especialmente el apoyo y comprensión ofrecido por Moisés y Jorge, mi jefe y tutor respectivamente, en la empresa donde he comenzado mi vida laboral: OSRAM. No podría haber pedido mejores compañeros de trabajo y mentores. Tampoco he de olvidar a mi compañero Jorge, quien nos alegraba todas las mañanas con su buen humor y energía.

Para ir terminando, un sentido agradecimiento dedicado a Lorena y Chema que siempre han creído en mí y han supuesto un gran apoyo. Siempre me infundían ánimos cuando el proyecto no iba bien.

Por último, agradecer a todas aquellas personas que no puedo mencionar (de lo contrario no acabaría nunca) y que han compartido grandes y pequeños momentos de mi vida que han hecho de mí la persona que soy.

De nuevo, gracias a todos.

Abstract

Each day we are more worried about road safety through compliance with traffic rules in order to prevent or reduce accidents on the roads. The awareness of road safety regulations involves, at least, a drastic reduction in the number of incidents on roads.

The current enforcement method through traffic fines aims to educate drivers of vehicles who have committed an offense by paying them. However, the current enforcement system is not very efficient in terms of time to deliver the notification. The communication channel (regular mail) increases this time of notification. The use of electronic media (electronic availability or Address Electronic Road) neither allows an immediate knowledge of the sanction by the offender. In the case of not having updated driver details, offence notifications may not be received by the offender or be sent to a wrong recipient. This causes the driver doesn't know that he has committed an offence until long after committing it. When the driver receives the notification, the corrective offence factor has almost lost its meaning.

This Final Project contributes to the creation of a communication protocol between the enforcement entity and the vehicle driver who commits an offence in a much more efficient and faster way. For this, it is necessary to use a device that is in the vehicle, called On-Board Unit (OBU), which can use the mobile communication technologies and DSRC technologies. On the other hand, it is needed a device service provider located in different parts of the road, called Road-Side Unit (RSU).

The main objective of this Final Project corresponds to the analysis and implementation of the communication protocol for a communication network which we call Vehicular Ad-hoc Network (VANET). Using these devices connected via VANET networks is achieved immediate communication between them, which perform an exchange of vehicle information. In this way, the RSU can send notifications to the OBU of vehicles to inform drivers of the offense. Thanks to these notifications sent in real time, the driver would notice the irregularity in their driving and proceed to correct it.

However, it is necessary to use an interface that makes an intermediary between the notification system and the user. For this, there are two types of interfaces, oriented according to the purpose of the device to which they are intended. In the case of the OBU, the interface should be oriented to inform the driver about the received notification without causing any distraction while driving. On the other hand, the RSU should be oriented to control and manage the notifications, as well as the devices themselves, by those authorized to do this work.

Key Words: notification, Vehicular Ad-hoc Network (VANET), communication protocol, OBU, RSU.

Resumen

Cada día nos preocupamos más sobre la seguridad vial mediante el cumplimiento de las normas de circulación con el fin de evitar o disminuir la siniestralidad en las carreteras. La concienciación de las normas de seguridad vial conlleva, como mínimo, a una reducción drástica del número de incidentes ocurridos en las vías de circulación.

El actual método de sancionamiento mediante multas de tráfico pretende concienciar a los conductores de los vehículos que han cometido una infracción mediante el pago de las mismas. Sin embargo, el actual sistema sancionador no es muy eficiente en cuanto al tiempo de envío de la notificación. El medio de comunicación empleado (correo ordinario) incrementa este tiempo de notificación. El uso de medios electrónicos (puesta a disposición electrónica o Dirección Electrónica Vial) tampoco permite el inmediato conocimiento de la sanción por parte del infractor. En el caso de no tener los datos del conductor actualizados, las notificaciones de sancionamiento pueden no ser recibidas por el infractor o bien ser enviadas a un destinatario erróneo. Esto produce que el conductor del vehículo no sepa que ha cometido una infracción hasta mucho tiempo después de cometerla. Cuando el conductor reciba la notificación, el factor correctivo de la sanción ha perdido casi todo su significado.

Este Proyecto Fin de Grado contribuye a la creación de un protocolo de comunicación entre la entidad sancionadora y el conductor del vehículo que comete una sanción de una forma mucho más eficiente y rápida. Para ello, es necesaria la utilización de un dispositivo que se encuentre en el propio vehículo, llamado On-Board Unit (OBU), que puede utilizar las tecnologías de comunicación móvil y tecnologías DSRC. Por otro lado, es necesario un dispositivo proveedor de servicios situado en distintos puntos de la calzada, al que llamaremos Road-Side Unit (RSU).

El objetivo principal de este Proyecto Fin de Grado se corresponde con el análisis e implementación de dicho protocolo de comunicación para una red de comunicación que llamaremos Red Vehicular Ad-hoc (VANET). Mediante la utilización de estos dispositivos conectados a través de redes VANET se logra la comunicación inmediata entre ellos, que realizan un intercambio de la información del vehículo. De este modo, la RSU podría mandar notificaciones a las OBU de los vehículos para informar a los conductores de la infracción cometida. Gracias a estas notificaciones enviadas en tiempo real, el conductor se daría cuenta de la irregularidad en su conducción y procedería a corregirla.

No obstante, es necesaria la utilización de una interfaz que haga de intermediaria entre el sistema de notificaciones y el propio usuario. Para ello se distinguen dos tipos de interfaces, orientadas según el objetivo del dispositivo al que van destinados. En el caso de la OBU, la interfaz deberá orientarse de forma que se informe al conductor de la notificación recibida sin que se produzca ninguna distracción durante la conducción. Por otro lado, la RSU deberá orientarse a controlar y gestionar tanto las notificaciones, como los propios dispositivos por aquellas personas autorizadas para ello.

Palabras Clave: notificación, Vehicular Ad-hoc Network (VANET), protocolo de comunicación, OBU, RSU.

Índice

1.	Introducción	11
1.1.	Objetivo	11
1.2.	Contexto y motivación	15
1.3.	Organización del documento	16
1.4.	Acrónimos y Definiciones	18
1.4.1.	Acrónimos	18
1.4.2.	Definiciones	19
2.	Análisis del Sistema	22
2.1.	Definición del Contexto	22
2.2.	Determinación del Alcance del Sistema	24
2.3.	Identificación de los Usuarios Participantes y Finales	27
2.4.	Establecimiento de Requisitos	28
2.4.1.	Especificación de Casos de Uso	28
2.4.2.	Obtención de Requisitos	32
2.5.	Estudio tecnológico	43
2.5.1.	RSU-OBU	43
2.5.2.	Interfaz de usuario	45
2.5.3.	Almacenamiento de Datos	47
2.6.	Marco regulador	49
2.7.	Arquitectura del sistema	50
2.7.1.	Patrón de desarrollo	50
2.7.2.	Arquitectura preliminar	51
2.8.	Definición de Interfaces de usuario	53
2.8.1.	Especificación de principios generales de la Interfaz	53
2.8.2.	Especificación del comportamiento dinámico de la Interfaz	53
2.9.	Especificación del Plan de Pruebas	58
2.9.1.	Definición del Alcance de las pruebas	58
2.9.2.	Definición de las Pruebas de Aceptación del Sistema	58
3.	Diseño Detallado	64
3.1.	Diseño de componentes	64
3.1.1.	Componente 1: Vista	64
3.1.2.	Componente 2: Modelo	72
3.1.3.	Componente 3: Controlador	75

3.1.4.	Componente 4: Gestión de mensajes	78
3.1.5.	Componente 5: Respuesta mensajes	82
3.2.	Diagramas de secuencia	86
3.2.1.	Conectar RSU.....	87
3.2.2.	Enviar sanción	89
3.2.3.	Visualizar lista de sanciones	91
3.3.	Diagrama relacional de la Base de Datos	93
4.	Implementación e Implantación del Software	95
4.1.	Procesos de Codificación y Pruebas	95
4.1.1.	Implementación y Pruebas del Software	95
4.2.	Resultados de las Pruebas de Aceptación	97
5.	Gestión del Proyecto	99
5.1.	Medios técnicos empleados.....	99
5.2.	Planificación del Proyecto	100
5.2.1.	Planificación Inicial	100
5.2.2.	Desarrollo Real del Proyecto	104
5.2.3.	Planificación Inicial vs Desarrollo Real	108
5.3.	Análisis económico del proyecto.....	108
5.3.1.	Metodología de estimación de costes	108
5.3.2.	Presupuesto inicial	109
5.3.3.	Presupuesto para el cliente.....	112
5.3.4.	Coste final.....	113
6.	Conclusiones y Líneas Futuras.....	118
6.1.	Conclusiones sobre el Proyecto	118
6.1.1.	Resultado obtenido	118
6.1.2.	Dificultad del Proyecto	119
6.2.	Líneas Futuras	120
	Bibliografía y Referencias.....	122
	Anexos.....	125
	Anexo 1: Estructura sanciones	125
	Anexo 2: Estructura fichero acuses de recibo.....	126
	Anexo 3: Manual de Usuario.....	127

Índice de Tablas

Tabla 1. Modelo de tabla de caso de uso.....	29
Tabla 2. Caso de uso 1: Conectar RSU	30
Tabla 3. Caso de uso 2: Enviar sanción.....	31
Tabla 4: Caso de uso 3: Visualizar lista de sanciones	32
Tabla 5. Modelo tabla requisitos.....	32
Tabla 6. Requisitos funcionales	34
Tabla 7. Requisitos Funcionales de Información.....	37
Tabla 8. Requisitos de interfaz	41
Tabla 9. Requisitos de aceptación.....	41
Tabla 10. Requisitos de portabilidad.....	42
Tabla 11. Modelo tabla pruebas	58
Tabla 12. Pruebas de Aceptación	63
Tabla 13: Resultados Pruebas de Aceptación	98
Tabla 14. Tareas planificadas iniciales	101
Tabla 15. Planificación inicial	102
Tabla 16. Tareas reales.....	105
Tabla 17. Desarrollo real del proyecto.....	106
Tabla 18. Costes de personal	109
Tabla 19. Costes de software	110
Tabla 20. Costes de equipamiento.....	111
Tabla 21. Costes de oficina.....	112
Tabla 22. Presupuesto para el cliente	112
Tabla 23. Comparativa costes de personal	113
Tabla 24. Comparativa costes de software	114
Tabla 25. Comparativa costes de equipamiento.....	115
Tabla 26. Comparativa costes de oficina.....	116
Tabla 27. Costes estimados vs costes totales	117

Índice de Ilustraciones

Ilustración 1. Dispositivo LOCOMATE (OBU) [3]	12
Ilustración 2. Dispositivo LOCOMATE (RSU).....	13
Ilustración 3: Sistema básico	14
Ilustración 4. Diagrama de bloques LOCOMATE con conexión al ordenador host [3]	24
Ilustración 5: Sistema básico	26
Ilustración 6. Diagrama de casos de uso	29
Ilustración 7. Representación Modelo-Vista-Controlador [28].....	51
Ilustración 8. Arquitectura preliminar.....	52
Ilustración 9. Prototipo ventana de conexión.....	54
Ilustración 10. Prototipo ventana operaciones RSU	55
Ilustración 11. Prototipo mensaje de error.....	55
Ilustración 12. Prototipo ventana información proyecto.....	56
Ilustración 13. Prototipo ventana de envío de sanción	56
Ilustración 14. Prototipo ventana últimas incidencias.....	57
Ilustración 15. Componente 1: Vista	65
Ilustración 16. Diagrama de clases 1 - <i>Conectar</i>	66
Ilustración 17. Diagrama de clases 2 - Enviar sanción.....	69
Ilustración 18. Diagrama de clases 3 - Visualizar sanciones.....	71
Ilustración 19. Diagrama de clases 4 - Modelo	74
Ilustración 20. Componente 3: Controlador	75
Ilustración 21. Diagrama de clases 5 - Conexión.....	76
Ilustración 22: Diagrama de clases 6 - Operaciones	77
Ilustración 23. Componente 4: Gestión de mensajes	78
Ilustración 24. Diagrama de clases 6 - Enviar sanción.....	80
Ilustración 25. Diagrama de clases 7 - Recibir acuse de recibo.....	82
Ilustración 26. Componente 5: Respuesta mensajes	83
Ilustración 27. Diagrama de clases 8 - Recibir sanción	84
Ilustración 28. Diagrama de clases 9 - Enviar acuse de recibo.....	86
Ilustración 29. Diagrama de secuencia 1: Conectar RSU.....	88
Ilustración 30. Diagrama de secuencia 2: Enviar sanción	90
Ilustración 31. Diagrama de secuencia 3: Visualizar lista de sanciones.....	92
Ilustración 32. Diagrama relacional de la Base de Datos	94
Ilustración 33. Planificación inicial	103
Ilustración 34. Desarrollo real del proyecto.....	107
Ilustración 35. Error conexión BBDD.....	128
Ilustración 36. IndexUI	129
Ilustración 37. AboutUI	130
Ilustración 38. PropertiesUI	131
Ilustración 39. AdminFavoritesUI.....	132
Ilustración 40. PingUI	132
Ilustración 41. ESaveUI.....	134
Ilustración 42. SendNotificationUI	135
Ilustración 43. LastReportsUI	136
Ilustración 44. Mensaje de desconexión RSU	136

1. Introducción

En este capítulo se realiza una presentación del Proyecto Fin de Grado que se ha realizado. Se mostrará el objetivo perseguido, las motivaciones que están presentes para su realización. También se mostrará una panorámica del proyecto, donde se pueda ver de forma fácil y sencilla qué se pretende conseguir, así como la organización del presente documento. Por último, se listarán aquellos acrónimos y definiciones con sus respectivos significados que se han utilizado durante el desarrollo del proyecto.

Este proyecto se enmarca en el ámbito del proyecto de Arquitectura de Seguridad y Generación de Pruebas Electrónicas Forenses en Entornos Vehiculares (E-SAVE). Dicho proyecto ha sido desarrollado por el Grupo de Seguridad en las Tecnologías de la Información de la Universidad Carlos III de Madrid en el marco del Plan Nacional de I+D+i 2008-2011, Código: TIN2009-13461.

El protocolo de intercambio de datos que se propone y desarrolla en el presente proyecto se describe en la Tesis Doctoral de José María De Fuentes [1].

1.1. Objetivo

Hoy en día se pretende mejorar cada día la calidad de la seguridad vial ofrecida tanto a los conductores como los viandantes. Para lograr este propósito se han tomado una serie de medidas para aumentar dicha seguridad. Algunos ejemplos serían la mejora de vehículos inteligentes, capaces de detectar vehículos u obstáculos cerca cuando el propio vehículo se encuentra en movimiento, el uso de sistemas de frenado más seguros y eficientes, etc.

Este proyecto trata sobre la preocupación de la velocidad de los vehículos en las distintas calzadas, puesto que, con un mayor control sobre la velocidad de los vehículos se consigue una disminución notoria sobre los distintos accidentes de tráfico. Para ello, tanto el Gobierno, como la industria y el sector de la educación han realizado una gran inversión sobre los llamados Sistemas Inteligentes de Transporte (ITS), que se correspondería con el desarrollo de las tecnologías de gestión del tráfico y la seguridad en los vehículos y la infraestructura vial. Las comunicaciones inalámbricas son, actualmente, la clave para los futuros servicios ITS.

La Organización Internacional de Normalización (ISO) TC204 WG16 está desarrollando un conjunto de estándares y una arquitectura internacional sobre el Acceso a las Comunicaciones Móviles Terrestres (CALM). Se espera que el sistema CALM pueda hacer uso de una amplia gama de tecnologías, como satélites, redes móviles (3G y 4G entre otros), Redes de Área Local Inalámbrica Wi-Fi (WLAN) y evoluciones del Acceso Inalámbrico en Entornos Vehiculares (WAVE) (IEEE802.11P y IEEE P1609). Además, muchas tecnologías de detección actuales podrán ser integradas en el sistema CALM, como el procesamiento de imágenes y vídeos [2].

Las Comunicaciones de Corto Alcance Dedicadas (DSRC)/WAVE parece la tecnología inalámbrica más adecuada para el contexto de circulación de vehículo, donde los accidentes geográficos, túneles, etc. impiden eventualmente la conectividad habitual. Gracias a DSRC es posible alcanzar una latencia. Esta característica asegura el papel de las tecnologías DSRC en las futuras redes CALM, que harán uso de múltiples radios en múltiples bandas. Sin embargo, las tecnologías DSRC actuales no han sido totalmente probadas. Actualmente hay desafíos sociales y técnicos sobre la tecnología DSRC que deben ser estudiados y resueltos antes de la implementación a gran escala.

El objetivo principal de este Trabajo final de Grado es realizar un análisis y posterior desarrollo de un protocolo de notificación de sanciones entre dos dispositivos que utilizan la tecnología DSRC, diferenciados según su utilidad. El primer dispositivo será la Unidad de A Bordo (OBU, del inglés *On-Board Unit*), que será el dispositivo que tendría equipado el propio vehículo y utilizará un dispositivo GPS integrado para el posicionamiento del vehículo. De forma predeterminada, el dispositivo transmite su posición en un canal de servicio continuo (5.86GHz) [3], codificados según el formato *Basic Safety Message* [4] que contiene la información relacionada con la seguridad del vehículo. El segundo se corresponde con la Unidad de Carretera (RSU, del inglés *Road-Side Unit*), que equivaldría a un poste que recibe los datos de las demás OBU para su procesamiento.

Estos dispositivos son equivalentes a la Estación Móvil (MS) y la Estación Base (BS) en los sistemas móviles, respectivamente. Hay dos clases de comunicaciones habilitados por las OBU y RSU: vehículo a vehículo (V2V) y vehículo a infraestructura (V2I). Mientras que una MS en el entorno móvil normalmente se comunica con otra MS a través de la BS, la OBU en un vehículo normalmente se comunica directamente con otros OBU dentro del área de cobertura de radio. Esta comunicación directa V2V reduce la latencia de mensajes y la baja latencia es un requisito esencial para las aplicaciones de seguridad como prevención de colisiones [5].



Ilustración 1. Dispositivo Locomate (OBU) [3]

La Ilustración 1 muestra el dispositivo LOCOMATE (OBU). Este dispositivo estará dentro de los vehículos, se encargará de recoger los datos del propio vehículo: velocidad, posición, fecha y hora, etc. Este dispositivo se comunicará con la RSU, que se muestra en la Ilustración 2:



Ilustración 2. Dispositivo LOCOMATE (RSU)

Al mismo tiempo que se analiza y desarrolla el sistema de comunicación OBU-RSU, se realizará un estudio sobre cómo representar esos datos, o lo que es lo mismo, estudiar la Interfaz de Usuario. La Interfaz estará para aquellos usuarios interesados y definidos en el apartado [3.3 Identificación de los Usuarios Participantes y Finales](#). Para ello se desarrollará una Interfaz de usuario que se encargará de hacer más fáciles todas las funciones que este sistema permite hacer y que estén dentro del alcance de nuestro sistema definido en el punto [3.2 Determinación del Alcance del Sistema](#).

Por último, en la Ilustración 3 se muestra un diagrama básico representativo del sistema que se pretende desarrollar. Como se puede apreciar, ambas tecnologías tendrán su propia Interfaz de Usuario, orientada a distintos usuario (la interfaz de la RSU estará orientada a empleados mientras que la de la OBU estará orientada a conductores). La RSU por su parte

gestionará los mensajes de forma distinta a la de la OBU, dependiendo de la función de cada dispositivo.

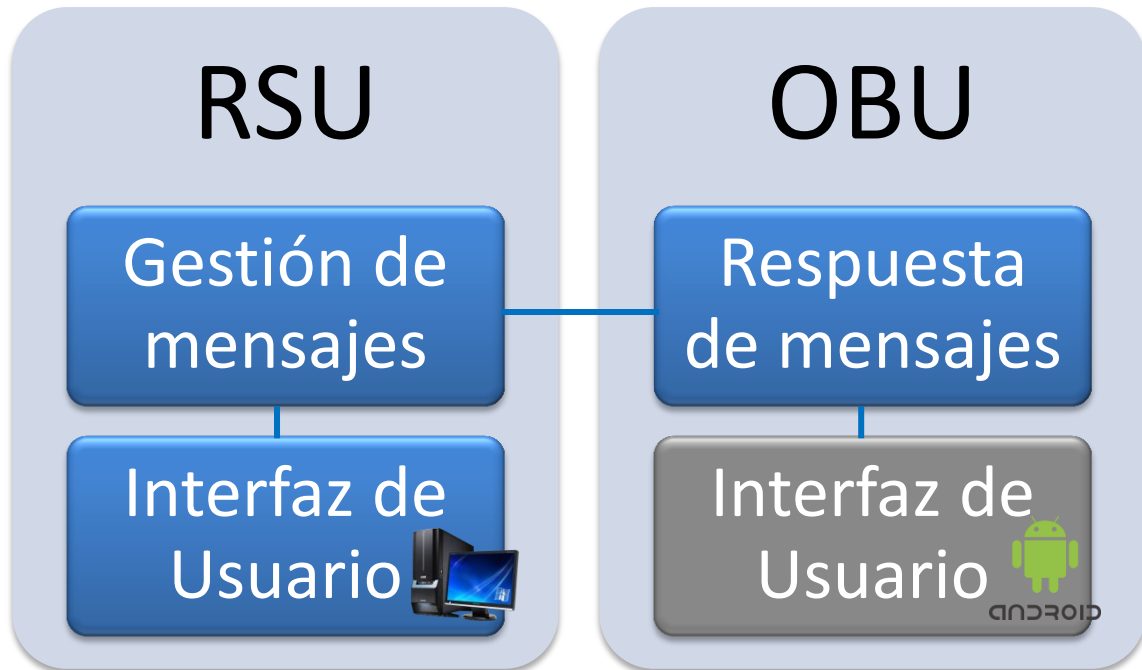


Ilustración 3: Sistema básico

Se ha de señalar que el sistema que se pretende implementar se correspondería con todo lo relacionado con la RSU y la respuesta de mensajes de la OBU. La Interfaz de Usuario de la OBU (marcada en gris) quedaría fuera del alcance del sistema.

1.2.Contexto y motivación

Cada día salen al mercado nuevos vehículos con sistemas más seguros y eficientes que han sido diseñados para mejorar la frenada de los vehículos, aumentar la visibilidad en las curvas, conseguir un mejor agarre de los neumáticos, etc. Pero la evolución de los vehículos no se para en este punto, también se están diseñando nuevos vehículos “inteligentes” capaces de realizar acciones de forma independiente al conductor. Gracias a esto se ha conseguido que los vehículos sean capaces de aparcar por sí solos, sin necesidad de la interacción del conductor. También se ha conseguido que el vehículo sea capaz de frenar de forma automática cuando detecta que existe un riesgo de colisión con el vehículo que circula delante. Por último, las últimas tendencias nos llevan a la conducción automática de los vehículos, que serían capaces de calcular y seguir una ruta planificada, detectar peligros durante el viaje, etc.

El actual sistema sancionador de España contiene una serie de procesos ineficientes que dificultan y retrasan la entrega de notificaciones de tráfico. Uno de lo más destacables se corresponde con el envío de la notificación mediante correo ordinario, de forma que la entrega de la misma se realizaba muy posterior a la realización de la infracción. Por otro lado, de no estar correctamente actualizados los datos del conductor del infractor, la notificación podría ser recibida con irregularidades, no llegar al conductor o enviar la notificación a una persona errónea.

Gracias al sistema de comunicación propuesto en este proyecto, se podrá mantener una comunicación directa entre OBU (integrado en el vehículo) y RSU, y, en el futuro, entre distintas OBU, de forma que exista una forma más eficiente y segura de controlar las medidas de seguridad impuestas en las carreteras o en los tramos de éstas que se deseen. Los conductores de los vehículos que incumplan alguna norma de seguridad vial serán avisados mediante un mensaje enviado por Bluetooth a su dispositivo Smartphone con sistema operativo Android [6] y éstos podrán rectificar su conducta, adaptándose a las normativas.

Mediante una interfaz intermediaria entre el usuario y la máquina, la notificación puede ser presentada al mismo de forma inmediata (se debe analizar la forma y momento adecuado para informar al conductor de forma que no le provoque una distracción). Los pasos anteriores se realizan utilizando la red vehicular basada en la tecnología DSRC. Cada uno de los envíos se repite un número N de veces para contrarrestar la eventual pérdida de información en la red vehicular.

Otro punto a favor del sistema de comunicación propuesto anteriormente es que no se trata de una detección de una infracción en un punto concreto y de forma estática, sino que se podrá cambiar la configuración de los dispositivos RSU de forma dinámica, de esta forma, se podrán adaptar a las necesidades del momento (si hay alguna obra en la calzada que requiera reducir la velocidad, algún tipo de condición meteorológica que requiera mayor seguridad, etc.). Con este sistema se podrá añadir mayor seguridad a todos los tipos de vía deseados: autovías, vías de poblado, túneles, etc. de una forma más fácil y económica.

Además de mejorar la seguridad, las redes WAVE pueden jugar un papel importante en el plan de viaje, gestión del tráfico, la navegación, control del medio ambiente, la logística, la

congestión y la reducción de emisiones, cobro de peajes, estacionamientos inteligentes, servicios de emergencia y una amplia gama de otros servicios basados en la localización.

Dadas las trágicas consecuencias de la mortalidad en las carreteras y su impacto económico, cualquier propuesta que contribuya a reducirlas tendrá un positivo impacto socio-económico. Este Trabajo Fin de Grado pretende contribuir en esta dirección, bajo la premisa de que una notificación más inmediata contribuirá a sensibilizar antes a los conductores infractores y, con ello, disminuir los letales efectos de su peligrosa actitud.

1.3.Organización del documento

Este apartado se encarga de describir, mediante un breve resumen, las diferentes secciones del presente documento, de forma que se pueda tener una idea clara y concisa de todo el documento dividido en capítulos.

Introducción

Este capítulo ofrece una vista general del proyecto, se explican los objetivos generales que se pretenden conseguir así como las motivaciones que han llevado a cabo la realización del proyecto.

Análisis del Sistema

En este capítulo se realizará un estudio del sistema que se pretende desarrollar, de forma que primeramente lo limitaremos a un sistema más pequeño y manejable, analizaremos cada una de las partes de ese sistema y obtendremos los requisitos necesarios para su desarrollo. Se realizará un análisis de las tecnologías que se pueden utilizar así como la tecnología usada finalmente justificando su uso. También se definirán las interfaces de usuario según las necesidades del proyecto para que sirvan como base a la interfaz final. Por último, se definirá un plan de pruebas con el fin de asegurar la calidad del software desarrollado.

Diseño detallado

En este capítulo se completará el análisis realizado anteriormente, detallando todos los componentes del sistema que se han encontrado y definido. Para terminar, se muestran y se detallan una serie de diagramas de secuencia que explican las distintas interacciones del usuario con el sistema para que pueda llevar a cabo las acciones estudiadas, en estos diagramas también se pueden apreciar los pasos que se han de seguir para llevar a cabo cada una de las acciones.

Implementación e Implantación del Software

En este capítulo se explican los detalles importantes de la implementación del software así como los detalles que se deben tener en cuenta para la implantación del nuevo sistema.

Gestión del Proyecto

En este capítulo se muestran los métodos utilizados para realizar el seguimiento del proyecto, se comparará la planificación definida inicialmente así como el desarrollo real. También se explicarán las herramientas utilizadas durante el desarrollo del proyecto, así como un análisis económico del proyecto que revelará los costes del proyecto estimados y reales, los beneficios obtenidos, riesgos del proyecto, etc.

Conclusiones y Líneas Futuras

En este capítulo se detallan las conclusiones finales del proyecto, explicando las dificultades encontradas durante su desarrollo, el resultado obtenido final, un resumen del desarrollo del proyecto, así como otras conclusiones relevantes sobre el mismo. También se expondrán las líneas futuras que se deberían seguir tras la realización del proyecto, mejoras, etc.

Bibliografía

En este capítulo se exponen todas las referencias bibliográficas utilizadas durante todo el proyecto así como durante la documentación del mismo.

Anexos

En este capítulo se expondrán todos los documentos anexos necesarios para mejorar la comprensión sobre el presente documento y el proyecto en general, como la estructura de los mensajes que se envían, un manual de usuario de la aplicación, etc.

1.4. Acrónimos y Definiciones

1.4.1. Acrónimos

API: Interfaz de Programación de Aplicaciones (*Application Programming Interface*).

BBDD: Base de Datos.

BS: Estación Base (*Base Station*).

CALM: Acceso a las Comunicaciones Móviles (*Communications Access for Land Mobiles*).

DGT: Dirección General de Tráfico.

DSRC: Comunicaciones de Corto Alcance Dedicadas (*Dedicated Short Range Communications*).

FTP: Protocolo de Transferencia de Archivos (*File Transfer Protocol*).

GHz: Giga Hertzio.

GPL: Licencia Pública General (*General Public License*).

GPS: Sistema de posicionamiento global (*Global Positioning System*).

HTML: Hyper Text Markup Language.

IP: Protocolo de Internet (*Internet Protocol*).

IRPF: Impuesto sobre la Renta de las Personas Físicas.

ISO: Organización Internacional de Normalización (*International Organization for Standardization*).

ITS: Sistemas Inteligentes de Transporte (*Intelligent Transport Systems*).

IVA: Impuesto sobre el valor añadido.

JSP: Java Server Pages.

JVM: Máquina virtual de Java (*Java Virtual Machine*).

MB: Mega Bytes.

MS: Estación móvil (*Mobile Station*).

MVC: Modelo-Vista-Controlador (*Model-View-Controller*).

OBU: Unidad de abordó (*On Board Unit*).

RFID: Identificación por Radiofrecuencia (*Radio Frequency Identification*).

RSU: Unidad de Carretera (*Road Side Unit*).

SS: Seguridad Social.

UC3M: Universidad Carlos III de Madrid.

UI: Interfaz de Usuario (*User Interface*).

UML: Lenguaje Unificado de Modelado (*Unified Modeling Language*). Se trata de un lenguaje gráfico para construir, documentar, visualizar y especificar un sistema de software.

USB: Bus Universal en Serie (*Universal Serial Bus*).

V2I: Vehículo A Infraestructura (*Vehicle To Infrastructure*).

V2V: Vehículo A Vehículo (*Vehicle To Vehicle*).

VANET: Red Vehicular Ad-hoc (*Vehicular Ad-hoc Network*).

WAVE: Acceso Inalámbrico en Entornos Vehiculares (*Wireless Access in Vehicular Environments*).

WLAN: Redes de Área Local Inalámbrica (*Wireless Local Area Network*).

WPAN: Red Inalámbrica de Área Personal (*Wireless Personal Area Network*).

XML: Lenguaje de Marcas Extensible (*Extensible Markup Language*).

1.4.2. Definiciones

Android: Sistema operativo basado en Linux orientado a dispositivos móviles.

Base de datos: Conjunto de datos informativos organizados en un mismo contexto para su uso y vinculación.

Bluetooth: Especificación tecnológica para redes inalámbricas que permite la transmisión de voz y datos entre distintos dispositivos mediante una radiofrecuencia segura (2,4 GHz).

Caso de uso: Secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

Ciente: Cualquier organismo nacional con capacidad para imponer sanciones a los conductores que incumplan las normativas de circulación vial

Diagrama de Gantt: Herramienta de planificación de trabajo que representa las diferentes actividades de un proyecto a lo largo del tiempo.

Dirección IP: Número identificativo asociado a un dispositivo de forma lógica.

Drag & Drop: Arrastrar y colocar.

Equipo cliente: Equipo que utilizará el sistema desarrollado en el presente proyecto. Sinónimo de equipo host.

Ethernet: Estándar de conexión para redes de área local.

GMail: Herramienta de correo electrónico de Google [7].

Hardware: Componentes físicos de los dispositivos.

Host: equipo cliente.

Implementador: Persona encargada de implementar el sistema del presente proyecto.

Interfaz de usuario: Medio que permite comunicarse al usuario con el sistema desarrollado en el presente proyecto de forma gráfica.

Interfaz: es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo.

Investigador: Persona perteneciente al grupo de proyecto que ha participado en la investigación del proyecto E-SAVE.

Java: lenguaje de programación orientado a objetos.

Linux: Sistema operativo de software libre distribuido bajo la licencia GNU GPL.

Login: Proceso mediante el cual se controla el acceso individual a un sistema informático mediante la identificación del usuario utilizando credenciales provistas por el usuario.

MySQL: sistema de gestión de bases de datos.

NetBeans: herramienta para programar en diferentes lenguajes de programación.

Objeto (programación): Correspondiente a la instanciación de una clase en la Programación Orientada a Objetos.

Ping: Programa empleado para verificar si un host o servidor está disponible.

Plug-in: Módulo hardware o software que añade una característica o servicio específico a un sistema o herramienta.

Responsable del proyecto: Persona encargada de la realización del presente proyecto.

Sanción: Aplicación de una pena económica a un conductor que ha cometido una infracción según las Normas de Circulación Vial.

Sistema (informático): es el conjunto de partes interrelacionadas, hardware, software y de recurso humano (humanware) que permite almacenar y procesar información (un servidor, un ordenador, etc.).

Sistema operativo: software encargado de ejercer el control y coordinar el uso del hardware entre diferentes programas de aplicación y los diferentes usuarios.

Smartphone: Teléfono móvil inteligente con funciones más avanzadas que un teléfono móvil convencional.

Software: Componentes no físicos de un sistema informático, como programas, sistemas operativos, etc.

Subsistema: conjunto de elemento interrelacionados que, en sí mismo, es un sistema, pero a la vez es parte de un sistema superior.

TelnetClient: Herramienta que instancia y mantiene una conexión mediante el protocolo *Telnet*.

Usuario: Persona física de la DGT que hace uso de un recurso del sistema.

Windows: Sistema operativo diseñado por Microsoft [8].

2. Análisis del Sistema

En este capítulo se procede a explicar la definición del sistema que se pretende desarrollar, explicando las limitaciones que se tendrán en cuenta, puesto que no se desarrollará el sistema por completo. También se detallarán todas las herramientas necesarias utilizadas para que el proyecto se finalice con éxito. Se identificarán los usuarios que participen de alguna forma en el sistema: el cliente, el usuario final, etc. Se llevará a cabo la especificación de casos de uso del sistema con el fin de abarcar toda su funcionalidad, así como su posterior desglose en requisitos que lo definan. Se realizará un estudio de las tecnologías que se pueden usar actualmente para el desarrollo del sistema, así como se argumentará qué herramientas se han utilizado y por qué se han elegido.

Por otro lado se detallará la arquitectura básica que debe tener el sistema para cumplir con los requisitos propuestos, identificando los distintos componentes en los que se divide la aplicación. También se explicará cómo debería ser la Interfaz de Usuario de una forma básica, definiendo las bases para que la interfaz final sea consistente y contenga todo lo necesario. Por último, se detalla el plan de pruebas refinado que se realizará al sistema con el fin de garantizar que el sistema cumple con las expectativas.

2.1. Definición del Contexto

En este apartado explica el sistema final que se tiene pensado para poner en producción tras distintos procesos de análisis e implementación que serán posteriores del desarrollo de este proyecto.

El sistema final pensado deberá ser capaz de imponer sanciones a los conductores viales que superen la velocidad establecida por la entidad supervisora de la seguridad vial. Para ello se necesitará de un dispositivo colocado en un punto de la carretera (RSU) que haga las funciones de “poste” receptor, también necesitaremos un dispositivo colocado en los vehículos que transmitan toda la información necesaria del mismo (OBU).

La OBU del vehículo enviará información relevante con respecto a la posición del vehículo, su velocidad, etc. La RSU, comprobará los datos recibidos y sancionará al conductor si se ha cometido alguna infracción (exceso de velocidad). En caso de cometerse alguna infracción, la RSU enviará un mensaje informativo a la OBU indicando la infracción cometida, fecha y hora del momento en que se cometió el delito, la sanción aplicada, etc. La OBU recibirá los datos solicitados, a partir de aquí pueden ocurrir dos cosas:

- La OBU compara los datos recibidos con la sanción cometida, si las otras OBU indican que no se ha cometido ninguna infracción, la OBU sancionada enviará un mensaje rectificativo a la RSU, indicando que no se ha cometido la infracción señalada anteriormente, indicando también las OBU que han certificado que no se ha cometido

dicha infracción. La RSU al recibir el mensaje rectificativo procederá a retirar la sanción impuesta.

- La OBU compara los datos recibidos con la sanción cometida, si las otras OBU corroboran que se ha cometido la infracción señalada, la OBU procederá a informar al conductor de que ha sido sancionado, para ello enviará un mensaje vía Bluetooth a su Smartphone, que tendrá como sistema operativo Android.

Se ha de mencionar que se asume el autoarranque del sistema de la OBU cuando el vehículo se ponga en funcionamiento. Según esto último, no sería necesaria ninguna acción del conductor para que la OBU comenzara a escuchar los envíos de la RSU.

Todos los mensajes enviados entre la OBU y la RSU se enviarán con una serie de medidas de seguridad con el fin de garantizar tanto la confidencialidad de los vehículos equipados con la OBU como la autenticidad de sus transmisiones. Todos los mensajes que se envíen entre la OBU y la RSU deberán estar certificados. De esta forma se verifica que el mensaje ha sido enviado por un dispositivo concreto y que no se pueda rechazar el envío realizado (no-repudio).

Como se ha mencionado anteriormente, se ha de garantizar también la confidencialidad de los vehículos que utilicen este sistema, de lo contrario, se podría obtener el certificado de una OBU concreta y de esta forma realizar un rastreo de su recorrido según el certificado obtenido. Para evitar este fallo de seguridad, se ha decidido renovar el certificado de cada OBU tras su puesta en marcha, es decir, cada vez que el vehículo se encienda, se procederá a la creación de un certificado aleatorio.

2.2. Determinación del Alcance del Sistema

En este apartado se describirán qué partes del sistema se van a desarrollar en este proyecto teniendo en cuenta lo explicado en el anterior punto [3.1. Definición del Sistema](#) y las limitaciones que encontraremos o que se han impuesto durante su análisis y desarrollo.

Se ha de tener en cuenta como limitación del sistema que la conexión entre los distintos dispositivos LOCOMATE (RSU/OBU) y el equipo host se realizará mediante cables Ethernet. La Ilustración 4 muestra un diagrama de bloques que representa la conexión del equipo host con un dispositivo LOCOMATE:

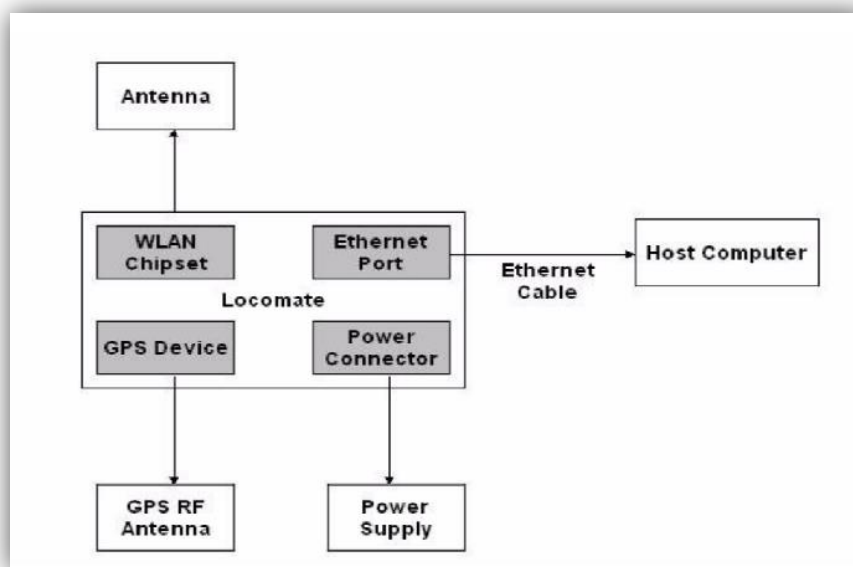


Ilustración 4. Diagrama de bloques LOCOMATE con conexión al ordenador host [3]

Gracias a la ilustración anterior también podemos ver los distintos componentes de los dispositivos que vamos a tratar, como por ejemplo la existencia de un dispositivo GPS.

Debido a las limitaciones de equipamiento y amplitud del sistema, la aplicación se desarrollará suponiendo que sólo hay una RSU, una OBU, un equipo cliente basado en el sistema de la RSU y un equipo cliente basado en el sistema de la OBU. Debido a esta limitación, la comunicación de distintas OBU quedará descartada del sistema que se pretende desarrollar. También quedará descartado el control de mensajes de la RSU, puesto que si existieran varias OBU enviando simultáneamente los datos a una RSU se tendría que controlar el envío y recepción de mensajes (no es un sistema full-dúplex, es decir, sólo puede enviar o recibir en un determinado momento, no puede realizar las dos acciones a la vez). Por último, también descartaremos que una OBU pudiera comunicarse con varias RSU a la vez (por ejemplo, la OBU podría estar al alcance de dos RSU).

La definición del sistema está altamente ligada al objetivo del sistema que queremos cumplir. El objetivo que se ha detallado anteriormente es demasiado amplio para que lo lleve a cabo una única persona en tan poco tiempo, por lo que se ha decidido separar ese objetivo en dos partes bien diferenciadas.

El proyecto actual se encargará de los envíos de los mensajes entre la RSU y la OBU, de forma que haya una forma de comunicación estable entre ambas. No se contemplará la interfaz de usuario del sistema OBU, así como tampoco se realizará el estudio y análisis de la seguridad de los mensajes enviados. A continuación, se explica con mayor detalle el alcance del proyecto llevado a cabo:

Se podrán realizar envíos de mensajes entre la OBU y la RSU. Esto supone el establecimiento de una conexión entre ambos dispositivos mediante el protocolo de comunicación que se desarrolle para este fin. La RSU será la encargada de enviar sanciones a la OBU, mientras que la OBU se encargará de enviar los acuses de recibo de las sanciones previamente enviadas. Cuando la RSU reciba algún acuse de recibo, se encargará de almacenar el estado de recepción de la sanción.

Con el fin de facilitar y agilizar el uso del sistema, se realizará un análisis y su posterior codificación de una Interfaz gráfica de Usuario, que se encargará de mostrar todos los datos almacenados, así como de realizar todas las operaciones necesarias para que el sistema funcione correctamente según lo propuesto y pueda llevar a cabo toda su funcionalidad, ganando, así mismo, una gran usabilidad. Recordemos que la Interfaz de Usuario que se ha desarrollado sólo influye al sistema de la RSU.

En la Interfaz de Usuario se implementarán una serie de opciones que no tendrán funcionalidad real, pero servirán para continuar el proyecto en vistas al futuro. Algunos ejemplos de estas opciones futuras fuera del alcance del proyecto actual serían la configuración de las RSU para control de velocidad dinámico o la recepción de denuncias ocultas.

Gracias a la implantación de una Interfaz de Usuario que haga de sistema intermedio entre un usuario y las funciones que nos permite la RSU, se consigue que un usuario no especializado en éste sistema pueda usarlo sin ningún tipo de complejidad.

Con este proyecto, por tanto, se pretende realizar el desarrollo de un protocolo de envío de sanciones. Los datos que se utilizan en este proyecto son ejemplos que muestran cómo es la estructura de las sanciones, cómo se realiza el envío etc. Muchos de los datos no variarán (aunque pueden ser modificados). Se realizarán diversas pruebas que verifiquen que el sistema puede realizar las distintas funciones propuestas anteriormente, así como diversas pruebas de campo que garanticen que el funcionamiento de los dispositivos integrados con el sistema implementado sea el esperado.

A continuación, en la Ilustración 5 se puede ver, como se ha mostrado anteriormente, una panorámica básica de la estructura del proyecto:

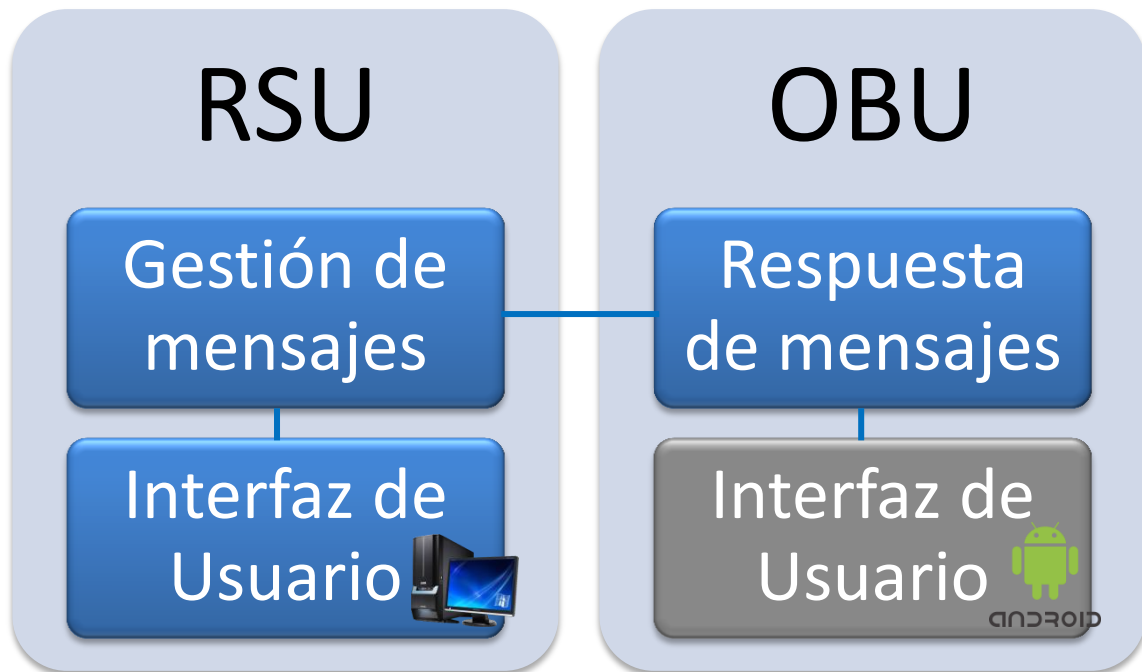


Ilustración 5: Sistema básico

Este proyecto se lleva a cabo con la colaboración de Alberto Aparicio, que llevará a cabo la parte de seguridad del sistema y la Interfaz de Usuario de la OBU. La seguridad del sistema consistirá en la creación y gestión de certificados que autentiquen tanto la RSU como la OBU. Además, se deberán tener en cuenta aspectos de seguridad, como la privacidad del conductor, entre otras. Por ejemplo, se deberá renovar el certificado de una OBU cada cierto tiempo estimado, con el fin de garantizar que no se pueda realizar un seguimiento de un vehículo sabiendo cuál es su certificado.

2.3. Identificación de los Usuarios Participantes y Finales

En este apartado se identifican los usuarios que participan en el análisis del sistema y los usuarios finales, que serán aquellos utilicen el sistema.

Interesados clave en el proyecto:

- **Jefes de proyecto:** *José María de Fuentes García-Romero de Tejada y Lorena González Manzano*. Es el encargado de supervisar la coordinación entre los distintos integrantes del equipo de proyecto, el trabajo realizado por cada uno de ellos para el correcto desarrollo del sistema con el fin de asegurar el desarrollo de software de calidad que se ajuste a las necesidades finales. Además, se encarga de proporcionar soporte al equipo de proyecto.
- **Equipo de proyecto:** Aquellos integrantes del equipo de proyecto encargados del desarrollo del proyecto:
 - *Eduardo Ruiz Alonso*: Responsable del apartado de RSU y la respuesta de sanciones de la OBU.
- **Fabricante:** Arada Systems [9].
- **Cliente:** El sistema que se pretende desarrollar es una herramienta que se desarrolla en el marco de un proyecto de investigación financiado en el plan nacional de I+D. El cliente podrá ser cualquier organismo nacional con capacidad para imponer sanciones a los conductores que incumplan las normativas de circulación vial.
- **Usuario final:** Cualquier empleado del cliente capacitado y encargado de la gestión de sanciones. Será el usuario que finalmente utilizará el sistema desarrollado.

2.4. Establecimiento de Requisitos

En este apartado se realizará un estudio de la funcionalidad del sistema que se pretende desarrollar, a continuación se compondrán una serie actividades generales llamadas casos de uso. Los casos de uso están formados por muchas actividades o tareas que se detallarán en el capítulo [3. Diseño Detallado](#). Una vez se tengan los casos de uso bien definidos, se procederá a la extracción de requisitos que proceden de estos casos de uso, que estarán categorizados según corresponda al requisito.

2.4.1. Especificación de Casos de Uso

En esta tarea se van a detallar los casos de uso que componen el sistema, mediante los cuales se procederá a la obtención de requisitos en el apartado [3.4.2. Obtención de Requisitos](#). Se representarán los casos de uso mediante un diagrama UML [10], así como unas tablas que acompañen y completen la información indicando el actor del caso de uso, el objetivo que perseguimos, precondiciones que se deben cumplir, post-condiciones que ocurrirán después, etc.

La notación UML utiliza una serie de símbolos para representar los distintos casos de uso de un sistema y sus relaciones. Nosotros utilizaremos esos símbolos: un rectángulo representará el propio sistema, dentro de él estarán los casos de uso representados mediante una elipse. El usuario será representado mediante un muñeco del que saldrán una serie de flechas o líneas que relacionarán el actor con los distintos casos de uso contenidos en el sistema.

Un caso de uso se identificará con un nombre y por el tipo de usuario de la aplicación, que se llama “**actor**”. Un actor no tiene por qué ser una persona que interactúa con el sistema, sino que puede tratarse de otro sistema, Consiste en la interacción entre los distintos actores y la aplicación y también la interacción entre los distintos sistemas de la aplicación.

La Ilustración 6 muestra el diagrama de casos de uso tal y como se ha explicado antes. Como se puede observar, el actor llamado usuario representa el usuario final que hará uso de la aplicación. Este usuario podrá realizar una serie de acciones definidas en cada caso de uso dentro del sistema de la RSU, a su vez, para que la aplicación sea completa, el sistema de la RSU hará uso de otras acciones pertenecientes al sistema de la OBU. Por ejemplo, para que el sistema esté completo, cuando la RSU envía un mensaje, necesita que la OBU lo reciba, así como que mande una contestación de dicho mensaje para que la RSU lo detecte.

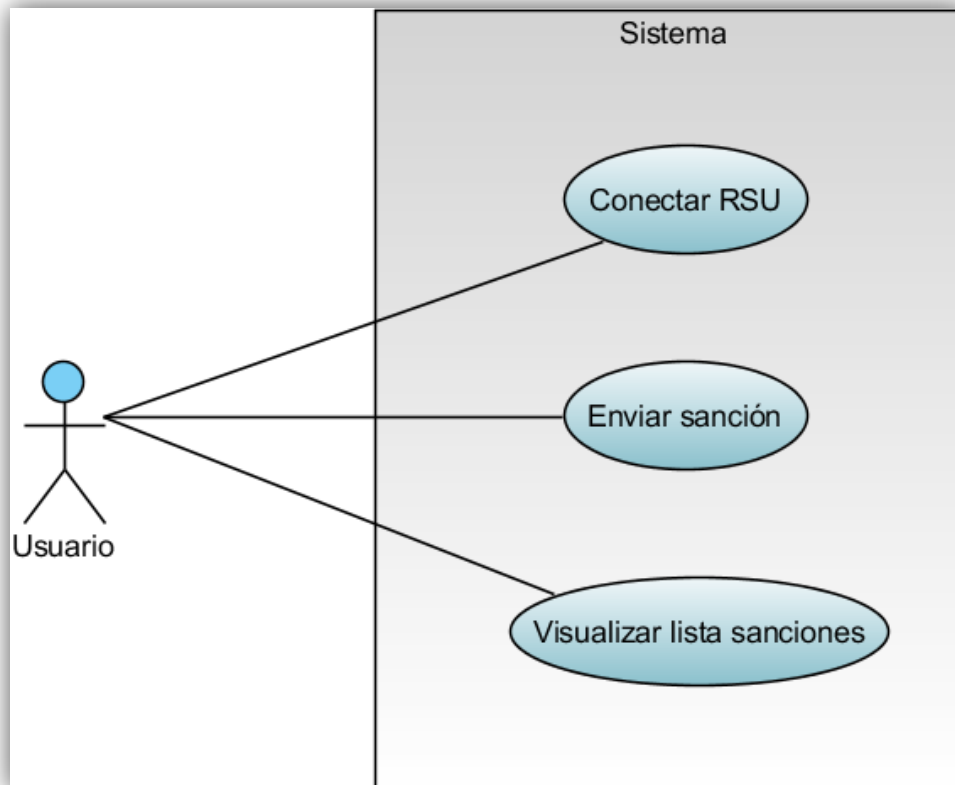


Ilustración 6. Diagrama de casos de uso

A continuación, completaremos todos los casos de uso mediante la utilización de una serie de tablas que seguirán el modelo definido por la Tabla 1. Se detallará toda la información que no se puede representar mediante el diagrama de casos de uso visto anteriormente en la Ilustración 6.

Nombre	<<Nombre del caso de uso>>
Actor/es	<<Persona o sistema externo que interactúa con el sistema>>
Descripción	<<Descripción detallada del caso de uso>>
Precondiciones	<<Estado del sistema para que se pueda dar el caso de uso>>
Post-condiciones	<<Estado posterior del sistema cuando se lleva a cabo el caso de uso>>
Flujo principal	<<Enumeración de los pasos a realizar>>
Flujo alternativo	<<Enumeración de los pasos alternativos a realizar para llevar a cabo el caso de uso>>

Tabla 1. Modelo de tabla de caso de uso

A continuación, en la Tabla 2, podemos ver el primer caso de uso, que se corresponde con la conexión de un equipo cliente con una RSU. El usuario, tras iniciar la aplicación, se podrá conectar a una RSU existente sabiendo su dirección IP (también se puede elegir mediante el

nombre si la IP ha sido guardada en “favoritos”) e introduciendo el usuario y contraseñas correspondientes a la RSU de conexión.

Nombre	Conectar RSU
Actor/es Descripción	Usuario El usuario se conectará a la RSU mediante la introducción de un usuario y una contraseña.
Precondiciones	-
Post-condiciones	Se realizará la conexión con la RSU deseada.
Flujo principal	<ol style="list-style-type: none"> 1. Introducir nombre de usuario. 2. Introducir contraseña. 3. Elegir RSU a conectar (nombre/IP). 4. Seleccionar la opción para realizar la conexión. 5. El sistema tratará de realizar la conexión con la RSU. 6. El sistema se conectará a la RSU.
Flujo alternativo	<p>Flujo alternativo 1:</p> <ol style="list-style-type: none"> 6a. El sistema no se podrá conectar a la RSU porque no está preparado el dispositivo. 7a. El sistema informará del error ocurrido. 8a. Se podrá realizar la conexión a otra RSU volviendo al paso 1. <p>Flujo alternativo 2:</p> <ol style="list-style-type: none"> 6b. El sistema no se podrá conectar a la RSU porque el dispositivo no está conectado o la dirección IP no es correcta. 7b. El sistema informará que el dispositivo no está disponible o es incorrecto. 8b. Se volverá al paso 3. <p>Flujo alternativo 3:</p> <ol style="list-style-type: none"> 6c. El sistema no se podrá conectar por error de nombre de usuario o contraseña. 7c. El sistema indicará que el nombre de usuario y/o contraseña son incorrectos. 8c. Se volverá al paso 1.

Tabla 2. Caso de uso 1: Conectar RSU

En la Tabla 3 podemos ver el Caso de Uso número 2, que se corresponde con el envío de una sanción desde la RSU a una OBU. El usuario, tras iniciar la aplicación y haberse conectado a una RSU, podrá enviar una sanción a la OBU con el fin de comprobar la comunicación RSU-OBU. La RSU quedará a la espera de recibir el acuse de recibo por parte de la OBU. La OBU, en su caso, recibirá la sanción enviada anteriormente por la RSU, procesará todos los datos y enviará un acuse de recibo a la RSU. La RSU recibirá el acuse de recibo y lo almacenará en un archivo XML [11]. Una vez se haya enviado la sanción, el usuario podrá ver una lista con las últimas sanciones realizadas (Caso de uso 3).

Nombre	Enviar sanción
Actor/es	Usuario
Descripción	El usuario podrá enviar una sanción a una OBU. Tras el envío de la sanción, la RSU permanecerá a la espera del acuse de recibo de la sanción enviada. La OBU recibirá la sanción, procesará los datos y enviará a la RSU el acuse de recibo. La RSU recibirá el acuse de recibo y lo almacenará en un fichero XML.
Precondiciones	El usuario debe estar conectado a la RSU.
Post-condiciones	Se registrará la sanción enviada.
Flujo principal	<ol style="list-style-type: none"> 1. Seleccionar la opción de enviar sanción. 2. El sistema mostrará una ventana para rellenar todos los datos de la sanción 3. Escribir todos los datos de la sanción. 4. Seleccionar la opción de continuar con la sanción. 5. El sistema enviará la instrucción necesaria para que la RSU envíe la sanción. 6. La RSU enviará la sanción. 7. La RSU esperará el acuse de recibo de la OBU. 8. El sistema registrará la sanción enviada. 9. La OBU recibirá la sanción. 10. La OBU enviará un acuse de recibo de la sanción enviada. 11. La RSU recibirá el acuse de recibo. 12. La RSU registrará el acuse de recibo.
Flujo alternativo	-

Tabla 3. Caso de uso 2: Enviar sanción

La Tabla 4 muestra el Caso de Uso número 3, que se corresponde con la visualización de sanciones impuestas por la RSU. El usuario, tras iniciar la aplicación y haberse conectado a una RSU, podrá visualizar una lista de las últimas sanciones que se han impuesto. Previamente, se recogerán de forma automática todos los acuses de recibo que haya registrado la RSU y se actualizará la Base de Datos según los datos recogidos, mientras que el registro XML que está en la RSU quedará vacío.

Nombre		Visualizar lista de sanciones
Actor/es		Usuario
Descripción		Se recogerán los acuses de recibo de la RSU. Se actualizará la Base de Datos según los acuses de recibo registrados. Se borrarán todos los registros de los acuses de recibo de la RSU hasta ese momento. El usuario podrá visualizar una lista de las últimas sanciones impuestas.
Precondiciones		El usuario debe estar conectado a la RSU.
Post-condiciones		La Base de Datos quedará actualizada según los datos recibidos. Se reseteará el fichero de registro de la RSU.
Flujo principal		<ol style="list-style-type: none"> 1. Seleccionar la opción de visualizar lista de sanciones. 2. El sistema mostrará la ventana con las últimas sanciones enviadas.
Flujo alternativo		Flujo alternativo 1: <ol style="list-style-type: none"> 1. Seleccionar la opción de enviar sanción. 2. Seleccionar la opción de continuar con la sanción. 3. Seleccionar la opción de visualizar la lista de sanciones. 4. El sistema mostrará la ventana con las últimas sanciones enviadas.

Tabla 4: Caso de uso 3: Visualizar lista de sanciones

2.4.2. Obtención de Requisitos

El propósito de la definición de requisitos software es analizar los requisitos de usuario y producir un conjunto de requisitos que debe cumplir el software, tan completo constante y correcto como sea posible.

Cada requisito seguirá el siguiente modelo propuesto en la Tabla 5, de forma que sea identificable, comprensible, y se pueda hacer un seguimiento del requisito que proviene:

Ident.	Prioridad	Fuente	Necesidad	Verifica- bilidad	Descripción	Proviene	Se desglosa
---------------	------------------	---------------	------------------	------------------------------	--------------------	-----------------	------------------------

Tabla 5. Modelo tabla requisitos

Identificación: Identificador único para localizar los requisitos, se distinguirá según el tipo de requisito y un número secuencial. El identificador seguirá el siguiente modelo:

- RS-<tipo_requisito>numero_identificativo

Prioridad: Será útil, para el desarrollador en cuanto a su gestión y planificación del proyecto. La prioridad puede ser:

- Alta
- Media
- Baja

Fuente: Durante el desarrollo del proyecto pueden surgir requisitos adicionales, y de estos nuevos requisitos no todos provendrán del Jefe de proyecto, por lo tanto se ha decidido diferenciar la proveniencia de los requisitos. Las fuentes, en este caso, pueden ser:

- Jefe de proyecto
- Responsable de proyecto.

Necesidad: Indica el nivel de relevancia del requisito sobre el proyecto. La necesidad puede ser:

- Esencial
- Deseable
- Opcional

Verificabilidad: Identificador de la prueba realizada para verificar el requisito.

Descripción: Descripción detallada del requisito.

Proviene: Requisito padre (si lo hubiera).

Se desglosa: Requisitos que nacen de éste (si los hubiera).

2.4.2.1. Requisitos Funcionales

En este apartado se detallan todos los requisitos que describen la funcionalidad del sistema.

Ident.	Prioridad	Fuente	Necesidad	Verificabilidad	Descripción	Proviene	Se desglosa
RS-F01	Alta	Jefe proyecto	Esencial	PRU-A01	Se podrá realizar la conexión entre un equipo host y una RSU.	-	RS-F02
RS-F02	Alta	Jefe proyecto	Esencial	PRU-A01	La conexión se realizará mediante el protocolo de conexión <i>Telnet</i> .	RS-F01	-
RS-F03	Alta	Jefe proyecto	Esencial	PRU-A02	La RSU enviará sanciones a la OBU.	-	-
RS-F04	Alta	Jefe proyecto	Esencial	PRU-A03	La OBU recibirá sanciones de la RSU.	-	-
RS-F05	Alta	Jefe proyecto	Esencial	PRU-A03	La OBU enviará un mensaje a la RSU indicando que ha recibido la sanción.	-	-
RS-F06	Alta	Jefe proyecto	Esencial	PRU-A02, PRU-A04	Se registrarán todas las sanciones impuestas.	-	RS-F05
RS-F07	Alta	Jefe proyecto	Deseable	PRU-A02, PRU-A04	El registro de todas las sanciones se guardará en la Base de Datos.	RS-F04	-
RS-F08	Alta	Jefe proyecto	Esencial	PRU-A03	La RSU recibirá mensajes de respuesta de llegada de la sanción.	-	-

Ident.	Prioridad	Fuente	Necesidad	Verificabilidad	Descripción	Proviene	Se desglosa
RS-F09	Alta	Jefe proyecto	Esencial	PRU-A03, PRU-A04	La RSU almacenará el estado de los mensajes de respuesta de llegada de la sanción.	-	RS-F08
RS-F010	Alta	Jefe proyecto	Deseable	PRU-A13	El registro de todos los acuses de recibo se guardarán en un fichero XML.	RS-F07	-

Tabla 6. Requisitos funcionales

2.4.2.2. Requisitos Funcionales de Información

Este apartado detalla todos los requisitos relacionados con la información de los mensajes intercambiados. Se ha decidido crear un requisito por cada etiqueta XML de forma que su comprensión sea más clara. Véase los apartados [8.1. Anexo 1: Estructura sanciones](#) y [8.2. Anexo 2: Estructura fichero acuses de recibo](#) para visualizar la estructura de las sanciones enviadas y para ver la estructura del fichero XML de registro de acuses de recibo respectivamente.

Ident.	Prioridad	Fuente	Necesidad	Verificabilidad	Descripción	Proviene	Se desglosa
RS-FI01	Media	Jefe proyecto	Deseable	PRU-A11	Los mensajes enviados tendrán estructura XML.	-	-
RS-FI02	Media	Jefe proyecto	Deseable	PRU-A11	El mensaje de sanción estará dentro de la etiqueta <i>fineNotification</i> .	-	-
RS-FI03	Media	Jefe proyecto	Deseable	PRU-A11	Se identificará el mensaje mediante la etiqueta <i>id</i> .	-	-
RS-FI04	Media	Jefe proyecto	Deseable	PRU-A11	Se identificará la fecha y hora de la sanción mediante la etiqueta <i>fineDate</i> .	-	-
RS-FI05	Media	Jefe proyecto	Deseable	PRU-A11	La descripción de la sanción se guardará con la etiqueta <i>offenceDesc</i> .	-	RS-FI06, RS-FI07, RS-FI08, RS-FI09
RS-FI06	Media	Jefe proyecto	Deseable	PRU-A11	Dentro de <i>offenceDesc</i> se identificará la fecha y hora de la notificación mediante la etiqueta <i>time</i> .	RS-FI05	-
RS-FI07	Media	Jefe proyecto	Deseable	PRU-A11	Dentro de <i>offenceDesc</i> se identificará el lugar de la sanción mediante la etiqueta <i>place</i> .	RS-FI05	-
RS-FI08	Media	Jefe proyecto	Deseable	PRU-A11	Dentro de <i>offenceDesc</i> se identificará al infractor de la sanción mediante la etiqueta <i>offenderId</i> .	RS-FI05	-
RS-FI09	Media	Jefe proyecto	Deseable	PRU-A11	Dentro de <i>offenceDesc</i> se identificarán los artículos violados mediante la etiqueta <i>rule</i> .	RS-FI05	-
RS-FI10	Media	Jefe proyecto	Deseable	PRU-A11	Los datos de la sanción se guardarán con la etiqueta <i>offenceData</i> .	-	RS-FI11, RS-FI12, RS-FI13
RS-FI11	Media	Jefe proyecto	Deseable	PRU-A11	Dentro de <i>offenceData</i> se identificará la velocidad del vehículo sancionado mediante la etiqueta	RS-FI10	-

Ident.	Prioridad	Fuente	Necesidad	Verificabilidad	Descripción	Proviene	Se desglosa
RS-FI12	Media	Jefe proyecto	Deseable	PRU-A11	<i>perceivedSpeed</i> . Dentro de <i>offenceData</i> se identificará el límite de velocidad establecido mediante la etiqueta <i>speedLimit</i> .	RS-FI10	-
RS-FI13	Media	Jefe proyecto	Deseable	PRU-A11	Dentro de <i>offenceData</i> se identificará la ruta de la imagen que verifique la sanción mediante la etiqueta <i>evidenceFile</i> .	RS-FI10	-
RS-FI14	Media	Jefe proyecto	Deseable	PRU-A11	Los datos de un testigo de la sanción se guardarán mediante la etiqueta <i>offenceWitness</i> .	-	RS-FI15, RS-FI16, RS-FI17
RS-FI15	Media	Jefe proyecto	Deseable	PRU-A11	Dentro de <i>offenceWitness</i> se identificará el id del testigo mediante la etiqueta <i>witnessID</i> .	RS-FI14	-
RS-FI16	Media	Jefe proyecto	Deseable	PRU-A11	Dentro de <i>offenceWitness</i> se identificará el tipo de testigo mediante la etiqueta <i>witnessType</i> .	RS-FI14	-
RS-FI17	Media	Jefe proyecto	Deseable	PRU-A11	Dentro de <i>offenceWitness</i> se identificará la afiliación del testigo con la etiqueta <i>witnessAff</i> .	RS-FI14	-
RS-FI18	Media	Jefe proyecto	Deseable	PRU-A11	Los datos de la autoridad sancionadora se guardarán mediante la etiqueta <i>authority</i> .	-	RS-FI19, RS-FI20
RS-FI19	Media	Jefe proyecto	Deseable	PRU-A11	Dentro de <i>authority</i> se identificará la autoridad mediante la etiqueta <i>ID</i> .	RS-FI18	-
RS-FI20	Media	Jefe proyecto	Deseable	PRU-A11	Dentro de <i>authority</i> se identificará la información de contacto mediante la etiqueta <i>contactInfo</i> .	RS-FI18	-
RS-FI21	Media	Jefe proyecto	Deseable	PRU-A11	La efectos de la sanción se guardarán mediante la etiqueta <i>fineEffects</i> .	-	RS-FI22, RS-FI23, RS-FI24
RS-FI22	Media	Jefe proyecto	Deseable	PRU-A11	Dentro de <i>fineEffects</i> se identificarán los puntos de sanción mediante la etiqueta <i>demeritPoints</i> .	RS-FI21	-
RS-FI23	Media	Jefe proyecto	Deseable	PRU-A11	Dentro de <i>fineEffects</i> se identificará la sanción económica mediante la etiqueta <i>economicPenalty</i> .	RS-FI21	-
RS-FI24	Media	Jefe	Deseable	PRU-A11	Dentro de <i>fineEffects</i> se	RS-FI21	-

Ident.	Prioridad	Fuente	Necesidad	Verificabilidad	Descripción	Proviene	Se desglosa
		proyecto			identificará la retirada del carnet de conducir (si procede) mediante la etiqueta <i>drivingLicenseWithdrawal</i> .		
RS-FI25	Media	Jefe proyecto	Deseable	PRU-A11	La información de apelación se guardará mediante la etiqueta <i>appeallingInfo</i> .	-	RS-FI26, RS-FI27, RS-FI28
RS-FI26	Media	Jefe proyecto	Deseable	PRU-A11	Dentro de <i>appeallingInfo</i> se identificará el periodo para la presentación de alegaciones o recursos mediante la etiqueta <i>timeInterval</i> .	RS-FI25	-
RS-FI27	Media	Jefe proyecto	Deseable	PRU-A11	Dentro de <i>appeallingInfo</i> se identificará el método para la presentación de alegaciones o recursos mediante la etiqueta <i>method</i> .	RS-FI25	-
RS-FI28	Media	Jefe proyecto	Deseable	PRU-A11	Dentro de <i>appeallingInfo</i> se identificará la identidad para la presentación de alegaciones o recursos mediante la etiqueta <i>entity</i> .	RS-FI25	-
RS-FI29	Media	Resp. proyecto	Deseable	PRU-A13	El registro de acuses de recibo estará dentro de la etiqueta <i>responses</i> .	-	-
RS-FI30	Media	Resp. proyecto	Deseable	PRU-A13	Se identificará cada acuse de recibo mediante la etiqueta <i>offenceResponse</i> .	-	RS-FI31, RS-FI32
RS-FI31	Media	Resp. proyecto	Deseable	PRU-A13	Dentro de <i>offenceResponse</i> se identificará la matrícula del vehículo mediante la etiqueta <i>carLicense</i> .	RS-FI30	-
RS-FI32	Media	Resp. proyecto	Deseable	PRU-A13	Dentro de <i>offenceResponse</i> se identificará la fecha y hora de la sanción mediante la etiqueta <i>date</i> .	RS-FI30	-

Tabla 7. Requisitos Funcionales de Información

2.4.2.3. Requisitos de Interfaz

En este apartado se detallan todos los requisitos que definen la Interfaz de Usuario.

Ident.	Prioridad	Fuente	Necesidad	Verificabilidad	Descripción	Proviene	Se desglosa
RS-I01	Alta	Jefe proyecto	Esencial	PRU-A01	La interfaz debe tener algún método para establecer una conexión con la RSU.	-	RS-I02, RS-I03
RS-I02	Alta	Jefe proyecto	Esencial	PRU-A01	La pantalla principal tendrá un sistema de inicio de sesión mediante la introducción de usuario y contraseña.	RS-I01	-
RS-I03	Alta	Jefe proyecto	Esencial	PRU-A01	La interfaz deberá mostrar varios lenguajes.	-	RS-I05, RS-I06, RS-I07, RS-I09
RS-I04	Alta	Jefe proyecto	Esencial	PRU-A01	Un lenguaje de la interfaz será el inglés.	RS-I04	-
RS-I05	Alta	Jefe proyecto	Esencial	PRU-A01	Un lenguaje de la interfaz será el español.	RS-I04	-
RS-I06	Alta	Resp. proyecto	Deseable	PRU-A01	La interfaz tendrá un apartado de propiedades de configuración.	-	-
RS-I17	Media	Resp. proyecto	Deseable	PRU-A01	La interfaz tendrá una lista de RSU favoritas.	-	RS-I11
RS-I18	Media	Resp. proyecto	Deseable	PRU-A05	Se podrán añadir RSU favoritas.	RS-I10	RS-I12, RS-I13
RS-I19	Media	Resp. proyecto	Deseable	PRU-A05	Se podrá escribir opcionalmente un nombre a la RSU.	RS-I11	-
RS-I10	Media	Resp. proyecto	Deseable	PRU-A05	Se deberá escribir obligatoriamente la dirección IP de la RSU.	RS-I11	-
RS-I11	Baja	Resp. proyecto	Deseable	PRU-A06	Se podrá editar una RSU favorita	RS-I10	RS-I15, RS-I16
RS-I12	Baja	Resp. proyecto	Deseable	PRU-A06	Se podrá editar el nombre de un favorito.	RS-I14	-
RS-I13	Baja	Resp. proyecto	Deseable	PRU-A06	Se podrá editar la dirección IP de un favorito.	RS-I14	-
RS-I14	Media	Resp. proyecto	Deseable	PRU-A07	Se podrá eliminar un favorito.	RS-I10	-
RS-I15	Alta	Resp. proyecto	Deseable	PRU-A01	Se podrá introducir una dirección IP temporal (se puede utilizar	-	-

Ident.	Prioridad	Fuente	Necesidad	Verificabilidad	Descripción	Proviene	Se desglosa
					temporalmente hasta que la interfaz se cierre) para la conexión a una RSU.		
RS-I16	Media	Resp. proyecto	Deseable	PRU-A08	Se podrá comprobar la conexión con una RSU.	-	RS-I20
RS-I17	Media	Resp. proyecto	Deseable	PRU-A08	Se realizará un <i>Ping</i> para comprobar la conexión.	RS-I19	-
RS-I18	Alta	Jefe proyecto	Esencial	PRU-A02	La interfaz podrá enviar la instrucción necesaria para enviar una sanción a la OBU.	-	-
RS-I19	Alta	Jefe proyecto	Esencial	PRU-A03	La interfaz tendrá una opción para recopilar las sanciones puestas por la RSU.	-	RS-I23
RS-I20	Alta	Jefe proyecto	Deseable	PRU-A02, PRU-A03, PRU-A04	La interfaz almacenará las sanciones recopiladas.	RS-I22	-
RS-I21	Media	Resp. proyecto	Deseable	PRU-A05	Al añadir un favorito, éste se guardará.	RS-I11	-
RS-I22	Media	Resp. proyecto	Deseable	PRU-A06	Al editar un favorito, éste se actualizará.	RS-I14	-
RS-I23	Media	Resp. proyecto	Deseable	PRU-A07	Al eliminar un favorito, éste se borrará.	RS-I17	-
RS-I24	Alta	Jefe proyecto	Esencial	PRU-A10	La interfaz podrá redimensionarse para adaptarse a la pantalla.	-	-
RS-I25	Alta	Jefe proyecto	Esencial	PRU-A12	Las interfaces principales tendrán el logo de la Universidad Carlos III de Madrid.	-	-
RS-I26	Alta	Jefe proyecto	Esencial	PRU-A12	Las interfaces principales mostrarán el nombre del proyecto.	-	-
RS-I27	Alta	Jefe proyecto	Esencial	PRU-A12	La interfaz tendrá la opción de mostrar una ventana <i>About</i> .	-	RS-I31, RS-I32, RS-I33
RS-I28	Alta	Jefe proyecto	Esencial	PRU-A12	Se mostrarán los nombres de los investigadores del proyecto.	RS-I30	-
RS-I29	Alta	Jefe proyecto	Esencial	PRU-A12	Se mostrarán los nombres de los implementadores del proyecto.	RS-I30	-
RS-I30	Media	Resp. proyecto	Deseable	PRU-A12	Se mostrará el nombre del implementador de la Interfaz de Usuario.	RS-I30	-
RS-I31	Alta	Jefe proyecto	Esencial	PRU-A04	La interfaz mostrará una lista de las últimas	-	RS-I35

Ident.	Prioridad	Fuente	Necesidad	Verificabilidad	Descripción	Proviene	Se desglosa
RS-I32	Alta	Jefe proyecto	Esencial	PRU-A04	sanciones. La interfaz mostrará el estado de cada sanción.	RS-I34	RS-I36
RS-I33	Alta	Jefe proyecto	Esencial	PRU-A04	Los estados de las sanciones pueden ser: <ul style="list-style-type: none"> • Recibido • No recibido 	RS-I35	-
RS-I34	Baja	Jefe proyecto	Opcional	PRU-A14	La interfaz mostrará una opción para recoger denuncias ocultas. Esta opción no tendrá funcionalidad real, sólo será un componente visual.	-	-
RS-I35	Baja	Jefe proyecto	Opcional	PRU-A15	La interfaz mostrará una opción para recibir las evidencias de defensa de una sanción. Esta opción no tendrá funcionalidad real, sólo será un componente visual.	-	-
RS-I36	Baja	Jefe proyecto	Opcional	PRU-A16	La interfaz mostrará una opción para poder configurar RSU por tramos. Esta opción no tendrá funcionalidad real, sólo será una componente visual.	-	RS-I40, RS-I41
RS-I37	Baja	Jefe proyecto	Opcional	PRU-A16	La interfaz mostrará una opción para establecer una RSU como poste de entrada en un tramo. Esta opción no tendrá funcionalidad real, sólo será una componente visual.	RS-I39	-
RS-I38	Baja	Jefe proyecto	Opcional	PRU-A16	La interfaz mostrará una opción para establecer una RSU como poste de salida en un tramo. Esta opción no tendrá funcionalidad real, sólo será una componente visual.	RS-I39	-
RS-I39	Baja	Jefe proyecto	Opcional	PRU-A17	La interfaz mostrará una opción para ver un resumen de controles de las distintas RSU. Esta opción no tendrá funcionalidad real, sólo será una componente	-	-

Ident.	Prioridad	Fuente	Necesidad	Verificabilidad	Descripción	Proviene	Se desglosa
RS-I40	Media	Resp. proyecto	Esencial	PRU-09	visual. La interfaz se podrá desconectar de la RSU conectada.	-	-

Tabla 8. Requisitos de interfaz

2.4.2.4. Requisitos de Aceptación

En esta sección se detallan los requisitos que debe cumplir el sistema para que se pueda dar como válido.

Ident.	Prioridad	Fuente	Necesidad	Verificabilidad	Descripción	Proviene	Se desglosa
RS-A01	Alta	Jefe proyecto	Esencial	PRU-A03	La OBU es capaz de recibir mensajes de la RSU.	-	-
RS-A02	Alta	Jefe proyecto	Esencial	PRU-A03	La OBU envía un mensaje de confirmación de llegada de mensaje a la RSU.	-	-
RS-A03	Alta	Jefe proyecto	Esencial	PRU-A03	La RSU es capaz de recibir los mensajes de confirmación de llegada de la OBU.	-	-
RS-A04	Alta	Jefe proyecto	Esencial	PRU-A03	La RSU almacena los mensajes de confirmación de llegada de la OBU.	-	-
RS-A05	Alta	Jefe proyecto	Esencial	PRU-A01	El usuario, a través de la interfaz, se puede conectar con la RSU.	-	-
RS-A06	Alta	Jefe proyecto	Esencial	PRU-A02	El usuario, a través de la interfaz, puede enviar una sanción a la OBU.	-	-
RS-A07	Alta	Jefe proyecto	Esencial	PRU-A03, PRU-A04	El usuario, a través de la interfaz, puede almacenar los acuses de recibo.	-	-
RS-A08	Alta	Jefe proyecto	Esencial	PRU-A04	El usuario, a través de la interfaz, puede visualizar las sanciones enviadas.	-	-
RS-A09	Alta	Jefe proyecto	Esencial	PRU-A04	El usuario, a través de la interfaz, puede visualizar el estado de recepción de las sanciones enviadas.	-	-

Tabla 9. Requisitos de aceptación

2.4.2.5. Requisitos de Portabilidad

En este apartado se detallan los requisitos que definen las características que tendrá el sistema para que pueda utilizarse en otros equipos o países.

Ident.	Prioridad	Fuente	Necesidad	Verificabilidad	Descripción	Proviene	Se desglosa
RS-P01	Alta	Jefe proyecto	Esencial	PRU-A01	La interfaz será multilinguaje.	-	RS-P02, RS-P03
RS-P02	Alta	Jefe proyecto	Esencial	PRU-A01	La interfaz se mostrará en inglés.	RS-P01	-
RS-P03	Alta	Jefe proyecto	Esencial	PRU-A01	La interfaz se mostrará en español.	RS-P01	-
RS-P04	Alta	Jefe proyecto	Esencial	PRU-A10	La interfaz se podrá adaptar a las dimensiones de la pantalla del equipo en que esté funcionando.	-	-

Tabla 10. Requisitos de portabilidad

2.5. Estudio tecnológico

Uno de los pasos previos a la codificación y el diseño del proyecto es la elección de la tecnología a usar, puesto que según qué se elija se puede obtener una serie de ventajas y/o limitaciones en la facilidad de uso así como en el propio diseño del proyecto. Esta parte estará dividida en tres grandes secciones de análisis: tecnologías impuestas (RSU-OBU) la Interfaz de Usuario y el almacenamiento de datos.

2.5.1. RSU-OBU

Este apartado realiza un estudio de las tecnologías impuestas para la utilización de los dispositivos RSU y OBU.

Estos dispositivos vienen de fábrica con algunos accesorios y herramientas implementadas con el fin de asegurar su funcionamiento. El dispositivo trae implementada una API para los desarrolladores que quieran realizar un proyecto software sobre éste tema. Junto con esta API, el dispositivo trae muchos ejemplos como software base, de forma que se pueda ver el funcionamiento del mismo a través de ellos. Estos ejemplos y toda la API están basados en el lenguaje C. Se ha de señalar que la RSU tiene como sistema operativo una versión reducida y modificada de Linux cuya versión actual en el dispositivo sería la versión 2.6.

En la documentación de los dispositivos OBU y RSU se aconseja desarrollar las aplicaciones en un sistema operativo basado en Linux [12]. Aunque proporcionan una serie de programas que se pueden ejecutar y codificar en el sistema operativo de Windows [13], se recomienda que esta parte sea únicamente para realizar pruebas.

Para tener un sistema operativo basado en Linux existen múltiples opciones. La primera opción sería comprar un equipo cliente que tenga dicho sistema operativo de fábrica, sin embargo ya se tiene un equipo y éste tiene instalado el sistema operativo de Windows. Por otro lado, y teniendo en cuenta esto último, se podría crear una partición en el equipo cliente, de forma que dicho equipo tuviera ambos sistemas operativos. Esto supone que sólo se pueda tener un sistema operativo utilizándose al mismo tiempo, además podría suponer un riesgo al equipo si la instalación no se hace correctamente. Por último, se podría utilizar un programa que genere una máquina virtual dentro del equipo cliente. Esta última opción permitiría utilizar el sistema operativo de Windows con normalidad, y arrancar mediante la máquina virtual creada el sistema operativo deseado.

Debido a las ventajas que ofrece la tercera opción sobre las demás, se ha decidido utilizar una máquina virtual, de forma que el equipo cliente no se viera seriamente afectado por la instalación de dicho sistema operativo. Dentro de esta elección existen dos posibilidades bien claras debido a la familiaridad del alumno con ambas máquinas virtuales: VirtualBox [14] y VMWare [15]. Ambas máquinas virtuales son muy conocidas actualmente y tienen características muy similares.

Gracias a la utilización de máquinas virtuales podremos tener un nuevo sistema operativo en nuestro equipo cliente que sea capaz de facilitar el uso de la codificación de las aplicaciones que vamos a desarrollar. Se recuerda que el sistema operativo que se instalará en la máquina virtual será Linux o alguna variante que esté basada en ese sistema operativo. Los sistemas operativos más conocidos por el alumno para este propósito son Linux [12] (la versión utilizada en la universidad se corresponde con la 6.0.7) y Ubuntu [16] (se han utilizado varias versiones). Las diferencias entre ambos sistemas operativos son mínimas, aunque Ubuntu ofrece un mejor sistema de instalación de software y mejor soporte al ser un sistema operativo más actual.

Por último, el fabricante restringe el traspaso de archivos entre los dispositivos RSU/OBU y el equipo host, de forma que sólo se pueden realizar intercambios de archivos mediante un servidor FTP. Los servidores FTP más usados actualmente son FileZilla [17] y Cerberus [18]. FileZilla ofrece una interfaz más intuitiva y manejable, bastante recomendable para principiantes y con transmisión de datos más rápida. Por otro lado, Cerberus se caracteriza por un mejor control y manejo de la conexión que se establece entre cliente y servidor, capaz de conectarse a múltiples plataformas.

2.5.1.1. Selección de Tecnologías no impuestas

En este apartado se seleccionarán las distintas tecnologías no impuestas por el fabricante con respecto al apartado anterior.

Para la creación de la máquina virtual, se ha utilizado la herramienta VMware versión 5.0.0 [15] debido a la experiencia adquirida durante las prácticas realizadas en las asignaturas de la carrera. La familiaridad del alumno con esta herramienta así como su gran rapidez de procesamiento frente a VirtualBox ha sido decisiva para seleccionarla.

Con respecto a la elección del sistema operativo se ha decidido utilizar la última versión de Ubuntu, que en el momento de realizar la descarga era Ubuntu 12.10 [16]. Esta selección, al igual que antes, ha sido tanto por la familiaridad del sistema como por tener una Interfaz de Usuario más intuitiva para el alumno.

Se ha de señalar que para realizar una fácil comunicación entre ambos sistemas operativos (el del equipo cliente y el de la máquina virtual), se aprovechará el uso de la Herramienta Dropbox versión 2.0.8 [19]. Gracias a esta herramienta, podremos tener sincronizados todos los documentos y ejecutables del proyecto. Para ello se vinculará Dropbox con la máquina virtual descargando la aplicación y registrando el correo electrónico del responsable de proyecto. A continuación, configuraremos Dropbox de forma que sólo mantenga sincronizada la carpeta asignada al proyecto. Una vez hecho esto, todos los cambios que hagamos sobre dicha carpeta se verán reflejados en ambos sistemas operativos, esto quiere decir que si se añade en dicha carpeta, por ejemplo, una captura de pantalla de la máquina virtual, también se añadirá dicha imagen en el sistema operativo.

Por último, con el fin de realizar la comunicación entre la RSU o la OBU y el equipo cliente de forma que se puedan transferir los archivos de uno a otro, se ha elegido la herramienta

FileZilla Server versión 0.9.41 beta [17]. Con este programa, aprovecharemos las funciones *ftpget* y *ftpput* que vienen residentes en ambos dispositivos. Para la OBU también se podrá utilizar un USB facilitado por el fabricante. Debido a la falta de experiencia del alumno sobre servidores FTP, se ha elegido la herramienta más intuitiva y sencilla de utilizar, sin embargo, para futuras versiones del sistema se debería usar la herramienta Cerberus, puesto que ofrece un mejor control de la comunicación.

2.5.2. Interfaz de usuario

Este apartado tratará de exponer las principales tecnologías que se podrían utilizar para la codificación de la Interfaz de Usuario.

Antes de comenzar con la selección de posibles lenguajes de programación así como las herramientas para su codificación, se debe mencionar que la Interfaz debe ser lo más portable posible. Esto supondría que la Interfaz de Usuario pudiera utilizarse en múltiples sistemas operativos.

Antes esta imposición inherente al proyecto, surgen dos fuertes elecciones de lenguajes de programación para cubrir la Interfaz de Usuario. Por un lado tenemos el lenguaje Java, que se caracteriza por su gran portabilidad a múltiples sistemas operativos gracias a la Máquina Virtual de java (JVM). Por otro lado, se puede destacar la programación en formato web, por el cual, para acceder a la Interfaz de Usuario sólo sería necesario el uso de un navegador web cualquiera, independientemente del sistema operativo en el que se acceda.

Java

La programación de la Interfaz basada en Java fue una de las primeras opciones en barajar, debido a la gran familiaridad del alumno con este lenguaje, puesto que durante la carrera universitaria se han dado múltiples casos en los que se ha tenido que programar con este lenguaje.

Java ofrece la programación orientada a objetos, lo que ayuda al programador a abstraerse ligeramente del código programando objetos que posteriormente usará. El programador ya no está atado a la ejecución secuencial de métodos, sino que encapsula un conjunto de datos y/o métodos que constituyen un pequeño componente del proyecto en un objeto. A partir de entonces, el programador podrá seguir codificando el proyecto en un objeto principal, que usará las funciones del otro objeto anteriormente codificado cuando quiera. De esta forma, el programador consigue separar los distintos componentes de un proyecto en distintos objetos y la función principal del proyecto hará uso de todos esos objetos siempre que les sean necesarios. Esto también ayuda a la hora de evitar la repetición de código y a su reutilización, es decir, supongamos que tenemos un objeto llamado *Calculadora*, que contiene las funciones *sumar*, *restar*, *multiplicar* y *dividir*. La utilización de objetos nos ayuda a aislar dichas funciones en ese objeto, por lo que cada vez que queramos realizar alguna operación matemática llamaremos al objeto *Calculadora* y utilizaremos la función que más nos convenga en ese momento.

Otra de sus ventajas principales es la Máquina Virtual de Java (JVM) que nos permite que la interfaz sea multiplataforma y no dependa de la arquitectura del equipo. Por tanto, la interfaz de usuario podrá utilizarse en cualquier equipo que contenga la JVM. Actualmente, casi todos los equipos disponen de una JVM para la ejecución de múltiples programas basados en este lenguaje altamente extendido.

Formato Web

Sin duda, ésta es una de las mejores opciones que existen a día de hoy. La mayoría de los sistemas están basados en servicios Web, y han cogido mucha fuerza las empresas basadas en páginas Web, ya sea para facilitar una gestión o contacto, para agilizar la compra mediante la venta on-line, etc.

Una de las ventajas que supondría la utilización de páginas web como medio gráfico para conectar con la RSU sería la facilidad para hacer que la página web sea dinámica y muestre los datos correspondientes a cada RSU. También tendríamos en cuenta que no existiría ningún límite de portabilidad, siempre que el equipo host tuviera un navegador web podría conectarse a la interfaz.

Se utilizarían páginas JSP debido al conocimiento del alumno sobre este tipo de tecnología sobre los demás tipos de páginas web dinámicas. Este tipo de tecnología está basado en Java, por lo que utilizaría también la JVM. Esto supondría casi las mismas ventajas que en el apartado: el alumno posee unos conocimientos previos que facilitan el desarrollo de la interfaz. Durante la formación del alumno en la universidad se ha utilizado habitualmente tanto el lenguaje Java como el lenguaje HTML.

Otra de las ventajas sería la facilidad para hacer que la interfaz fuera flexible en cuanto a la resolución y el tamaño de la pantalla, así como el control sobre el sistema multilinguaje.

2.5.2.1. Conclusiones

La principal ventaja de la utilización de formato web también supone una gran desventaja para el cliente que adquiera este sistema si se desarrollara con esta tecnología: se tendría que controlar el tráfico de Internet.

Por tanto, finalmente se ha optado por la codificación en Java, puesto que actualmente casi todos los equipos contienen la JVM, por lo que no debería preocuparnos la portabilidad de un sistema implementado con este lenguaje.

La herramienta que usaremos durante la codificación será: **NetBeans 7.2.1** [20]. Es una herramienta bastante potente que simplifica la generación de interfaces de usuario. Otra herramienta que podríamos usar sería *Eclipse* [21] mediante la utilización de un *plug-in*, sin embargo se ha optado por la primera debido a su simplicidad y familiaridad con el desarrollador del proyecto. *NetBeans* ofrece la posibilidad de crear objetos de tipo formulario

(campos de texto, botones, etc.) arrastrando estos objetos desde el menú de esta herramienta al lugar de trabajo (Drag & Drop).

2.5.3. Almacenamiento de Datos

Este apartado explicará las distintas tecnologías que se han tenido en cuenta para el almacenamiento de los datos del sistema.

Se ha de destacar que existen algunos datos que deben ser almacenados, puesto que vienen inherentes al proyecto, como son los datos de las sanciones impuestas por la RSU.

Existen infinidad de formas de almacenamiento de información actualmente: ficheros de texto plano, ficheros con texto identificativo mediante etiquetas (por ejemplo XML), bases de datos, etc. Las tecnologías anteriormente mencionadas son las que se han tenido en cuenta para almacenar la información del sistema.

Texto plano

El almacenamiento de datos en ficheros de texto plano conlleva algunas ventajas, pero demasiados inconvenientes. Como ventaja se ha de destacar el fácil almacenamiento de los datos: se crea un fichero si no existe ya y se escribe al final los datos que queremos almacenar. Sin embargo, los inconvenientes de este tipo de tecnología de almacenamiento son demasiados: no existe seguridad en el fichero, la recuperación de los datos es muy lenta, la modificación de algún dato se convierte en un gran problema, etc.

XML

La utilización de ficheros basado en etiquetas, como un fichero XML mejora bastante el apartado anterior. La búsqueda de los datos se realiza de forma más rápida y efectiva y el almacenamiento de datos es sencillo y ordenado. Sin embargo, seguimos encontrando los mismos problemas de la seguridad de los ficheros de almacenamiento, así como la dificultad a la hora de la modificación de cualquier dato ya almacenado.

Aunque podría utilizarse para el almacenamiento de los datos de nuestro sistema, este tipo de tecnologías suelen ser más efectivas para ficheros estáticos, que no crecen ni varían los datos.

Base de Datos

La utilización de una base de datos para el almacenamiento de información es la elección óptima en cualquier sistema que requiera el almacenamiento masivo de información, como es el caso de este proyecto. Las bases de datos están a la orden del día debido a su gran seguridad de almacenamiento, gran facilidad de almacenamiento, búsqueda, modificación y eliminación de registros.

Como gran ventaja, las bases de datos permiten el uso de relaciones entre distintas tablas de almacenamiento, lo que supondría una gran ventaja para no repetir información cuando

existen muchísimos registros en un sistema. Las búsquedas por clave principal se realizarán en un tiempo de acceso muy corto, frente a las otras elecciones donde la búsqueda se realizaría secuencialmente.

Como posibles gestores de bases de datos a utilizar se han comparado MySQL [22], Microsoft Access [23] y SQL Server [24].

2.5.3.1. Conclusiones

Debido a las grandes ventajas ofrecidas por las bases de datos, se han elegido como tecnología a utilizar para el almacenamiento de datos. Gracias a la base de datos podremos almacenar otros datos adicionales que mejoren la calidad del sistema. Por ejemplo, se podrían guardar todos los textos de la Interfaz de Usuario, de forma que se consiguiera una interfaz mucho más portable respecto al país donde se implantaría, puesto que se podrían almacenar diferentes textos para los mismos botones.

Como se ha mencionado anteriormente, al usar una base de datos el desarrollador no tendría que abstraer y realizar las operaciones de la base de datos preocupándose únicamente por los datos que se deben controlar y almacenar.

En cuanto al sistema gestor de bases de datos, se ha decidido utilizar MySQL. Las grandes ventajas por las que se ha elegido este sistema gestor de bases de datos han sido por sus destacadas características: su estabilidad, su seguridad, por ser multiplataforma, etc. Actualmente MySQL es uno de los sistemas gestores de bases de datos más utilizados en el mundo. Además, el alumno se ha visto involucrado con este tipo de sistemas anteriormente en la carrera más a menudo que con los demás sistemas gestores de bases de datos. Además, MySQL es un sistema gratuito, mientras que SQL Server se trata de un sistema de pago, por lo que quedaría descartado dado el precio y las grandes ventas de MySQL.

A pesar de que Access resultaría más fácil a la hora de implementar la base de datos, así como a la hora de controlar los datos introducidos de forma visual (sin necesidad de software intermedio ni conocimientos previos sobre consultas a bases de datos), las ventajas que ofrece MySQL son muy superiores a las demás herramientas desde el punto de vista empresarial, por lo que se ha elegido como herramienta de desarrollo de la base de datos.

Para la implementación de la Base de Datos se ha decidido utilizar la herramienta MySQL Workbench 5.2 CE [25] para crear el diagrama relacional de la base de datos. Una vez creada la Base de Datos, se ha utilizado la herramienta XAMPP Control Panel versión 3.1.0 [26] para introducir todos los datos iniciales a la base de datos. Esta herramienta también permite que la Base de Datos se aloje en el equipo host, de forma que se simule un servidor local para poder utilizarla. De nuevo, la elección de estas dos últimas herramientas ha sido por familiaridad del alumno con las herramientas, además de ser herramientas gratuitas.

2.6. Marco regulador

El actual sistema sancionador de España contiene una serie de procesos ineficientes que dificultan y retrasan la entrega de notificaciones de tráfico. A pesar de que la reforma efectuada en la Ley 18-2009 busca mejorar esta cuestión, tanto el envío por correo ordinario como la puesta a disposición por medios electrónicos consiguen que la notificación se practique en un momento muy posterior a la realización de la infracción (Ley 18/2009¹). Uno de lo más destacables se corresponde con el envío de la notificación mediante correo ordinario, de forma que la entrega de la misma se realizaba muy posterior a la realización de la infracción. Por otro lado, de no estar correctamente actualizados los datos del conductor del infractor, la notificación podría ser recibida con irregularidades, no llegar al conductor o enviar la notificación a una persona errónea. A raíz de estos problemas, la Ley 18/2009 reformó dicho procedimiento para agilizar la tramitación de las sanciones. Dicha Ley ofrece la posibilidad de realizar el envío de estas notificaciones por medios telemáticos, de forma que se corrigieran algunos de los defectos anteriormente mencionados. Con este fin, la Ley 11/2007 y el Real Decreto 1671/2009² que la desarrolla parcialmente proponen, como uno de los métodos de notificación, cualquier sistema “siempre que quede constancia de la recepción por el interesado en el plazo y en las condiciones que se establezcan en su regulación específica”.

Este mecanismo, ofrece la posibilidad de notificar al conductor del vehículo en el momento en el que se está cometiendo la infracción, potenciando así el factor correctivo de la notificación. Dicho mecanismo puede utilizar tecnologías que no estén explícitamente contempladas en la legislación, pues tal y como expone la Ley 11/2007³, “la Ley no puede limitarse a regular el uso de los canales electrónicos disponibles hoy en día”. De acuerdo a la Ley 18/2009, las notificaciones deben contener los siguientes elementos:

- a) **Sobre el infractor:** la identificación del vehículo, la del denunciado (si fuera conocida). El domicilio que, en su caso, indique el interesado a efectos de notificaciones.
- b) **Sobre la infracción:** una descripción del hecho (especificando el lugar, fecha y hora) y la infracción presuntamente cometida.
- c) **Sobre el denunciante:** el nombre y domicilio del denunciante (o, en su caso, el número de identificación profesional del Agente).
- d) **Sobre la sanción:** El órgano competente para imponer la sanción y la norma que le atribuye tal competencia, la sanción que pudiera corresponder y el número de puntos cuya pérdida lleve aparejada la infracción.
- e) **Sobre las acciones posibles:** Deberá indicarse que dicha denuncia inicia el procedimiento sancionador y que dispone de un plazo de veinte días naturales para efectuar el pago, con la reducción y las consecuencias establecidas en el artículo 80, o

¹ ESPAÑA (2009). “Ley 18/2009, de 23 de noviembre, por la que se modifica el texto articulado de la Ley sobre Tráfico, Circulación de Vehículos a Motor y Seguridad Vial, aprobado por el Real Decreto Legislativo 339/1990, de 2 de marzo, en materia sancionadora”. Boletín Oficial del Estado (BOE).

² ESPAÑA (2009). “Real Decreto 1671/2009, de 6 de noviembre, por el que se desarrolla parcialmente la Ley 11/2007, de 22 de junio, de acceso electrónico de los ciudadanos a los servicios públicos”. Boletín Oficial del Estado (BOE).

³ ESPAÑA (2007). “Ley 11/2007, de 22 de junio, de acceso electrónico de los ciudadanos a los Servicios Públicos.”. Boletín Oficial del Estado (BOE).

para formular las alegaciones y proponer las pruebas que estime convenientes. En este caso, se indicarán los lugares, oficinas o dependencias donde puede presentarlas.

2.7.Arquitectura del sistema

En este apartado se detallarán todos los aspectos relacionados con la arquitectura preliminar del sistema. Primeramente se describirá el patrón de desarrollo elegido para la posterior fase de implementación. Elegir el patrón de desarrollo adecuado permite estructurar el sistema de forma distinta a fin de conseguir un objetivo, en este caso el objetivo del patrón será conseguir un sistema portable.

En segundo lugar se describirá la arquitectura del sistema en sí misma. Se tendrán en cuenta los requisitos necesarios para el sistema, así como el alcance que se haya determinado para el proyecto.

2.7.1. Patrón de desarrollo

Como se ha mencionado anteriormente, el sistema establecerá un sistema de comunicación básica entre los dispositivos OBU y RSU mediante el protocolo establecido para el proyecto.

Se ha decidido utilizar el patrón de Modelo-Vista-Controlador (MVC) [27]. Este patrón de diseño ayudará a la fragmentación del sistema en distintos componentes, dependiendo de si pertenecen a la Interfaz de Usuario, realizan operaciones o bien tratan de los datos del sistema.

Este patrón permite separar y definir el sistema de forma que la lógica de negocio sea independiente de la interfaz de usuario utilizada. Se ha decidido usar este patrón debido al objetivo propuesto para el proyecto: comprobar y realizar la comunicación OBU-RSU para la transmisión de sanciones. Gracias a este patrón de diseño, la arquitectura resultará en un sistema portable y modificable capaz de realizar todas las acciones requeridas en los requisitos propuestos en el punto [3.4.2. Obtención de Requisitos](#), así como una interfaz propuesta para utilizar dicho sistema. La interfaz de usuario no ha de ser la interfaz final que se utilizará cuando el sistema esté completo según lo definido en el apartado [7.2. Líneas Futuras](#), sino que es una propuesta para facilitar el uso del sistema desarrollado en este proyecto.

El patrón MVC divide el sistema en tres módulos diferenciados:

- **Modelo:** Es la representación de la información en el sistema. Trabaja junto a la vista para mostrar la información al usuario y es accedido por el controlador para añadir, eliminar, consultar o actualizar datos.
- **Vista:** Representa la información del sistema para que el usuario pueda interactuar con ella. Se trata de la Interfaz de Usuario.
- **Controlador:** Recibe, trata y responde los eventos enviados por el usuario o por la propia aplicación. Interactúa tanto con el modelo como con la vista.

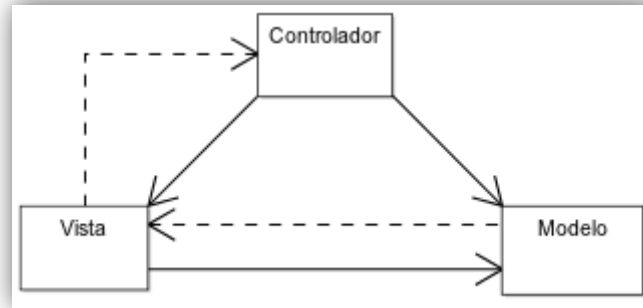


Ilustración 7. Representación Modelo-Vista-Controlador [28]

2.7.2. Arquitectura preliminar

En este apartado se muestra el sistema separado en distintos componentes de forma que cada uno quede bien diferenciado por su utilidad. Se parte del sistema básico explicado e ilustrado en el punto [2.2. Determinación del Alcance del Sistema](#), por el que se define de una forma más precisa cada parte del sistema.

El sistema general se puede dividir en dos sub-sistemas básicos, que pertenecerían a la OBU y a la RSU. Cada sub-sistema tiene su propia Interfaz de Usuario, que tratará los datos de una forma distinta, puesto que el objetivo de cada uno es diferente. La interfaz de usuario de la RSU estará orientada al envío de sanciones, así como la recepción y manejo de los acuses de recibo. También será posible realizar las configuraciones necesarias de la RSU. Por otro lado, la Interfaz de Usuario de la OBU estará orientada a la recepción de sanciones y su visualización por el usuario, así como el tratamiento de las sanciones.

Recordamos que la OBU se verá limitada únicamente a la respuesta de sanciones enviada previamente por la RSU, por lo que el tratamiento de las sanciones será exclusivamente el necesario para enviar una respuesta que identifique la sanción y corrobore que la sanción ha sido recibida y procesada.

Vemos un tercer componente equivalente a los elementos comunes de ambos sub-sistemas. En este caso, los elementos comunes que utilizarán ambos sistemas serán los relacionados con la seguridad y certificación de la transmisión de mensajes. Estos elementos comunes serán tratados de forma muy similar en ambos sub-sistemas.

En la Ilustración 8 se corresponde con la arquitectura preliminar del sistema, o lo que es lo mismo, el sistema dividido en diferentes componentes según su función. Se aprovecha esta ilustración para recordar que no se implementará todo el sistema, sólo el componente RSU y la respuesta de mensajes de la OBU.

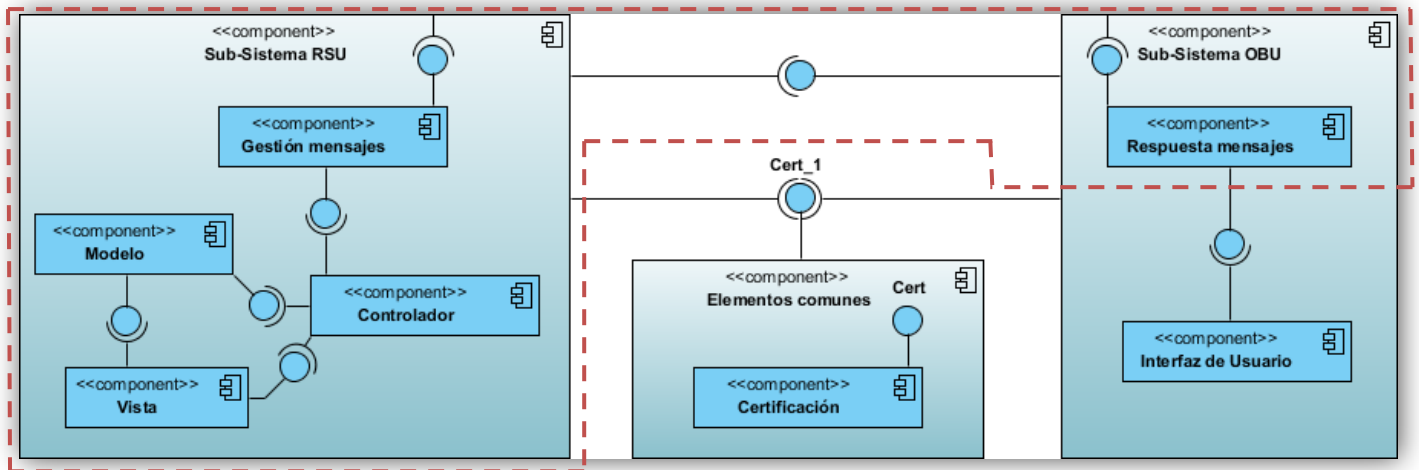


Ilustración 8. Arquitectura preliminar

Todos los elementos rodeados por la línea roja de rayas se corresponden con los distintos componentes o elementos que se abordan en este Trabajo Fin de Grado. Los componentes situados fuera de la línea se corresponden con elementos fuera del alcance del proyecto actual.

Gracias a la separación del sistema en los componentes mostrados en la Ilustración 8, se obtiene un sistema adaptable, capaz de ser modificado fácilmente si cambiamos algún componente del sistema. Por ejemplo, se podrá modificar la gestión de los mensajes de la RSU mientras que la Interfaz no se vería involucrada en gran medida (salvo el tratamiento de las sanciones enviadas y acuses de recibo). Por otro lado, se podría realizar una Interfaz de Usuario totalmente nueva (ya sea por una renovación o cambio de tecnología) sin que la gestión de mensajes estuviera afectada.

Se ha de señalar que las interfaces *Cert* y *Cert_1* son la misma, pero que por limitaciones del *Visual Paradigm* no ha sido posible utilizar la misma interfaz sin perder su significado en el diagrama. Se ha decidido utilizar este método para relacionar ambas interfaces y señalar que son la misma, superando así este problema.

2.8. Definición de Interfaces de usuario

En este apartado se especifican las interfaces de usuario que se mostrarán y harán de intermediarias entre el sistema RSU y el usuario final. Se analizarán los procesos del sistema en los que se requiere interacción por el usuario y se definirá una interfaz que satisfaga todos los requisitos establecidos en el punto [3.4.2.2. Requisitos de Interfaz](#) así como teniendo en cuenta al usuario al que va dirigido. El propósito es construir una interfaz de usuario acorde a sus necesidades, flexible, coherente, eficiente y sencillo de usar.

El primer sub-apartado de esta sección definirá los principios generales de la Interfaz de Usuario desarrollada, restricciones o limitaciones encontradas, etc. En el segundo apartado, se definirá cómo debe comportarse la interfaz, haciendo un primer “boceto” de la Interfaz de Usuario final.

2.8.1. Especificación de principios generales de la Interfaz

El objetivo de este apartado es la definición de la interfaz de usuario, especificando las normativas y estándares que se tendrán en cuenta durante su desarrollo. También se tendrá en cuenta las capacidades generales del usuario final, así como las necesidades y experiencia del mismo.

Supondremos que un usuario final será un empleado de la DGT, que trabaja en sus oficinas especializadas y tiene conocimientos básicos sobre el sistema gestor de multas correspondiente, de forma que tendrá los conocimientos básicos sobre los reglamentos de seguridad vial. Se supone también que los usuarios finales tienen conocimientos básicos sobre el uso del sistema operativo Windows.

Con el fin de acercar la interfaz a los usuarios finales, se desarrollará acorde a los términos y conceptos relacionados con el ámbito de la seguridad vial. La interfaz estará adaptada para que su uso sea sencillo e intuitivo. Los menús y submenús tendrán una apariencia similar, así como su disposición y representación en la interfaz, intentando reducir el esfuerzo de aprendizaje sobre la nueva aplicación.

La interfaz controlará los posibles errores que puedan surgir de su uso. Cuando se capturen errores inesperados, el sistema incorporará mecanismos que retornarán al sistema a un estado anterior y estable, o bien procederán a corregir el error si es posible.

2.8.2. Especificación del comportamiento dinámico de la Interfaz

El objetivo de este apartado es definir la navegación entre las distintas “pantallas” o ventanas de la interfaz de una forma textual y gráfica. Tras la lectura del siguiente apartado, la

utilización del sistema quedará mucho más definida, de forma que la interfaz final contenga al menos la funcionalidad mínima explicada a continuación.

La representación gráfica de las ventanas en este apartado será únicamente orientativa. La Interfaz que se utilizará finalmente en este sistema se explicará más adelante en este documento.

Existirán dos ventanas principales:

- **Sin conexión:** Esta parte de la interfaz servirá para conectar con las distintas RSU que estén disponibles, para ello se dispondrá de los campos necesarios para escribir el usuario y la contraseña correspondiente a cada RSU. También se proporcionará un componente para elegir la RSU a la que se desea conectar. Por último, desde esta ventana se tendrá acceso al componente que cambie el idioma de la interfaz. Esta ventana tendrá la capacidad de salir de la aplicación. La Ilustración 9 muestra el prototipo de interfaz explicado anteriormente:

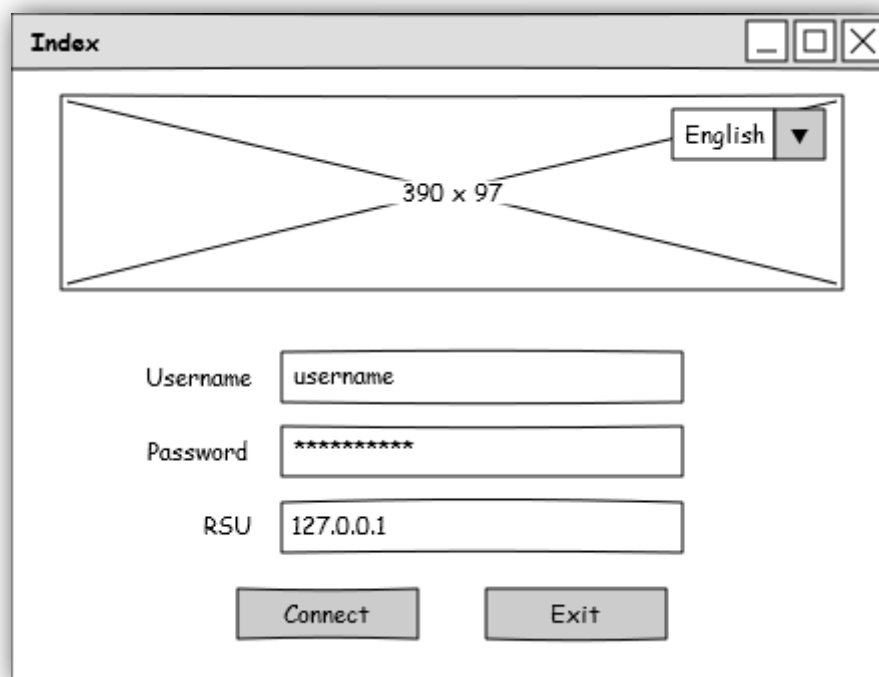


Ilustración 9. Prototipo ventana de conexión

- **Con conexión:** Esta parte de la interfaz estará conectada a una RSU concreta y servirá para recoger y visualizar las últimas sanciones impuestas así como para realizar la propia sanción a una OBU que esté al alcance. Esta ventana será capaz de salir de la aplicación o de volver a la ventana principal sin conexión para conectar con otra RSU. La Ilustración 10 muestra el prototipo de interfaz explicado anteriormente.

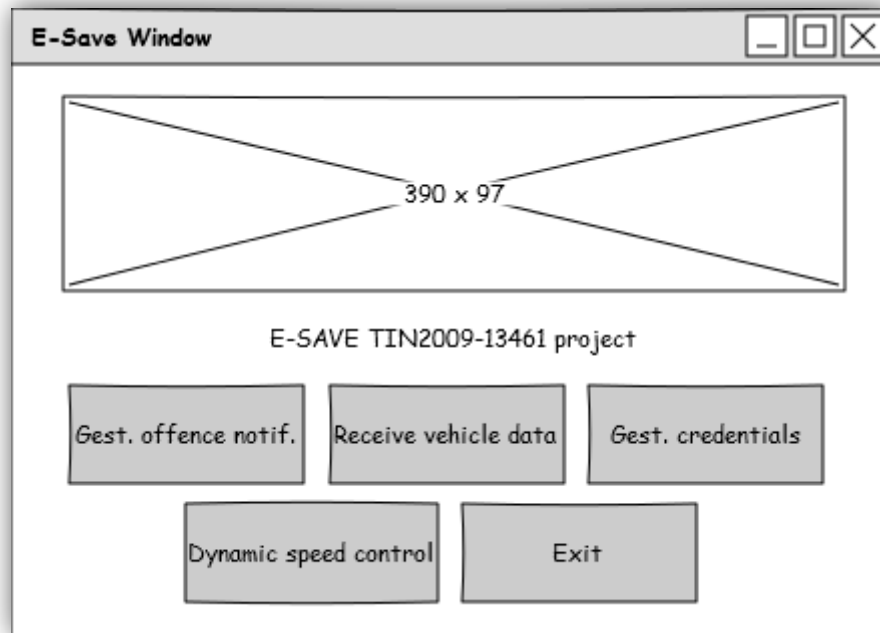


Ilustración 10. Prototipo ventana operaciones RSU

Cualquier aviso u error que se produzca durante la navegación por la interfaz se mostrará mediante una ventana que tendrá un símbolo (dependerá del tipo de aviso u error) así como del mensaje explicativo de lo que ha ocurrido durante la navegación. Se mostrará un botón para volver a la ventana principal (dos en el caso de que tenga que aceptar o rechazar algo). La Ilustración 11 muestra el prototipo de interfaz explicado anteriormente:

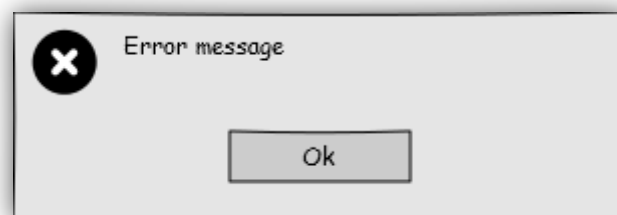


Ilustración 11. Prototipo mensaje de error

Con el fin de mostrar el equipo de desarrollo del proyecto que se ha realizado, se mostrará una ventana que contenga todo el equipo de desarrollo separando al equipo en diferentes categorías según la función que hayan desempeñado durante la realización del mismo. La Ilustración 12 muestra el prototipo de interfaz explicado anteriormente:



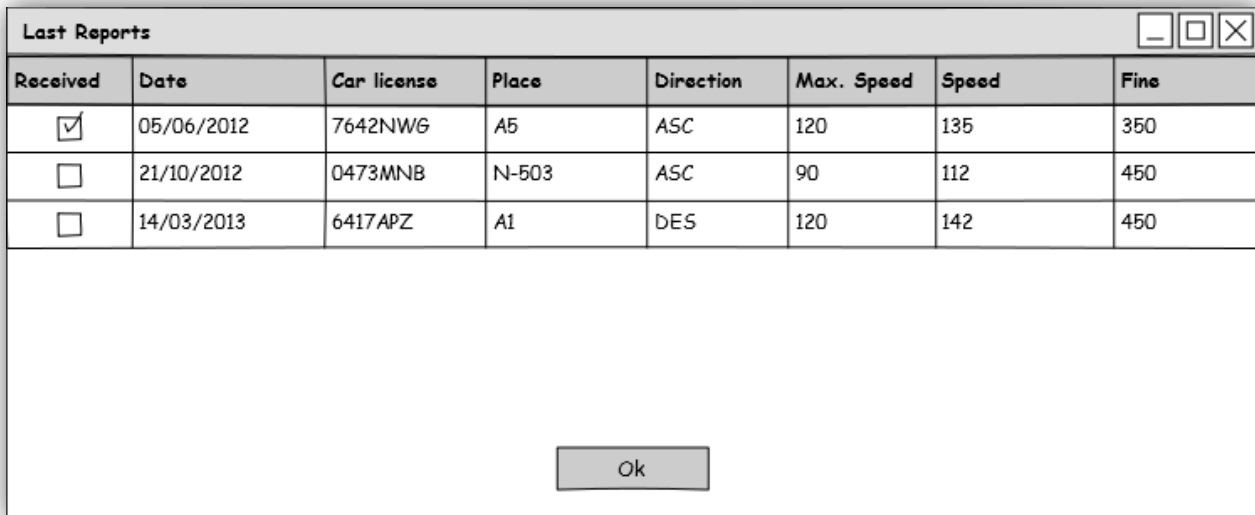
Ilustración 12. Prototipo ventana información proyecto

Para poder enviar una sanción a una OBU, primeramente se nos mostrarán los datos de la OBU sancionable; su velocidad, su ubicación, la cuantía de su sanción, etc. Si finalmente no queremos enviar la sanción procederemos a cancelar el envío mediante la opción proporcionada para ello, de lo contrario procederemos al envío de la sanción. La Ilustración 13 muestra el prototipo de interfaz explicado anteriormente:



Ilustración 13. Prototipo ventana de envío de sanción

Se podrán visualizar las últimas sanciones enviadas. Se mostrará el estado de recepción de la sanción enviada, la fecha de la sanción, la matrícula del vehículo, el lugar donde ocurrió, el sentido de la carretera, la velocidad máxima que se podía alcanzar, la velocidad a la que iba el vehículo y, por último, la cuantía de la sanción. La Ilustración 14 muestra el prototipo de la interfaz explicada anteriormente:



Received	Date	Car license	Place	Direction	Max. Speed	Speed	Fine
<input checked="" type="checkbox"/>	05/06/2012	7642NWG	A5	ASC	120	135	350
<input type="checkbox"/>	21/10/2012	0473MNB	N-503	ASC	90	112	450
<input type="checkbox"/>	14/03/2013	6417APZ	A1	DES	120	142	450

Ok

Ilustración 14. Prototipo ventana últimas incidencias

2.9. Especificación del Plan de Pruebas

En este apartado se definirá el Plan de Pruebas que tendrá como objetivo garantizar el sistema como software seguro, viable y verificable. Para ello se definirá el alcance de las pruebas que se van a realizar, o lo que es lo mismo, qué es lo que se pretende probar y con qué objetivo. Se establecerán las pruebas que debe superar el sistema con el fin de completar el objetivo propuesto en el apartado [1.1. Objetivo](#) teniendo en cuenta las limitaciones establecidas en el apartado [3.2. Determinación del Alcance del Sistema](#). Por último, se determinará los requisitos que se han comprobado en el Plan de pruebas mediante una matriz que relacione la prueba realizada con los requisitos de los que proviene.

2.9.1. Definición del Alcance de las pruebas

En este apartado se determinan los distintos niveles y tipos de pruebas que se realizarán con el fin de garantizar que el sistema cumple con los requisitos establecidos en el apartado [3.4. Establecimiento de Requisitos](#).

Se ha decidido establecer un único tipo de pruebas para que el documento sea más consistente así como menos extenso. Las pruebas que se van a realizar serán las pruebas de aceptación, que comprobarán el funcionamiento del sistema en general. Se comprobarán varios requisitos en la misma prueba, aunque los requisitos sean de distinto tipo.

2.9.2. Definición de las Pruebas de Aceptación del Sistema

En este apartado se especifican las pruebas que se han de cumplir para verificar la calidad de la aplicación con respecto a los objetivos y requisitos establecidos. La tabla 11 muestra el modelo que se va a seguir para la definición de las pruebas:

Identificador	Descripción	Resultado esperado	Proviene de
---------------	-------------	--------------------	-------------

Tabla 11. Modelo tabla pruebas

Identificador: Identificador único para localizar las pruebas, se distinguirá según el tipo de prueba y un número secuencial. PRU-<tipo_prueba>numero_identificativo

Descripción: Descripción detallada de la prueba. Se detallará de forma textual qué acciones se deben realizar, qué valores se deben introducir (si procede)

Resultado esperado: Sucesos que tienen que ocurrir y/o salida por pantalla que se tiene que mostrar.

Proviene de: Requisitos probados.

2.9.2.1. Pruebas de Aceptación

En este apartado se expone el plan de pruebas refinado que comprobará el funcionamiento del sistema. Se verificará que se cumplen todos los requisitos necesarios según la descripción de la prueba para que el sistema se pueda dar como válido y tenga la calidad deseada según el resultado obtenido.

Las pruebas se realizarán durante el último mes del proyecto. El estado de la implementación en este punto está lo suficientemente avanzado para realizar las primeras pruebas de aceptación que asegurarán el sistema. Si todavía se tuviera que codificar parte del sistema, todas las pruebas deberán realizarse tras la codificación por completo de la parte del sistema que se quiere verificar.

Las pruebas se superarán si se obtiene el resultado esperado definido en la propia prueba. En caso de no superarse dicha prueba, se realizarán las modificaciones necesarias para que el sistema funcione correctamente y poder realizar dicha prueba de nuevo.

Antes de la entrega del proyecto, todas las pruebas de aceptación definidas en este apartado deberán superarse de forma exitosa. Una semana antes de la entrega del proyecto, se realizarán todas las pruebas de aceptación para verificar el correcto funcionamiento del sistema en su conjunto.

La Tabla 13 muestra las pruebas de aceptación que se han llevado a cabo:

Identificador	Título	Descripción	Resultado esperado	Proviene de
PRU-A01	Conectar con la RSU	Se iniciará la aplicación. Se pulsará sobre el botón <i>Propiedades</i> o <i>Propiedades</i> . Se pulsará, en la esquina superior derecha sobre el desplegable de idiomas y se seleccionará el idioma Español. Se introducirá el texto "192.168.0.40" en el apartado de <i>Dirección IP</i> . Se pulsará sobre el botón <i>OK</i> . Se introducirá el texto "root" en el apartado de <i>Usuario</i> . Se introducirá el texto "password" en el apartado <i>Contraseña</i> . Se pulsará sobre la opción de <i>Conectar</i> .	En el cuadro de texto debajo de los botones de <i>Conectar</i> , <i>Propiedades</i> y <i>Salir</i> se mostrará un texto indicando que se está llevando a cabo la conexión. Tras unos instantes, se mostrará un texto indicando que la conexión ha finalizado con éxito. Se cerrará la ventana actual y se mostrará la ventana de <i>E-Save</i> .	RS-F01, RS-F02, RS-I01, RS-I02, RS-I03, RS-I04, RS-I05, RS-I06, RS-I08, RS-I09, RS-I10, RS-I18, RS-A05, RS-P01, RS-P02, RS-P03
PRU-A02	Enviar sanción	La prueba comenzará desde la ventana de <i>E-Save</i> en el idioma Español. Se pulsará sobre el botón de <i>Enviar sanción</i> . Se ocultará la ventana anterior y se mostrará una nueva ventana con los datos de la sanción (matrícula del coche, velocidad máxima, velocidad del vehículo, fecha y hora, etc.). Se pulsará sobre el botón de <i>Enviar</i> .	Se mostrará un mensaje indicando que la sanción ha sido enviada.	RS-F03, RS-F06, RS-F07, RS-I21, RS-I23, RS-A06
PRU-A03	Recibir acuses de recibo	La prueba comenzará desde la ventana de <i>E-Save</i> en el idioma Español. Se pulsará sobre el botón de <i>Recibir datos</i> .	Se mostrará una nueva ventana que mostrará el progreso y el estado de la recepción de datos. Al finalizar la recepción se cerrará y se mostrará un mensaje indicando que ha finalizado la recepción.	RS-F04, RS-F05, RS-F08, RS-F09, RS-I22, RS-I23, RS-A01, RS-A02, RS-A03, RS-A04, RS-A07
PRU-A04	Visualizar lista de sanciones	La prueba comenzará desde la ventana de <i>E-Save</i> en el idioma Español. Se pulsará sobre el botón de <i>Últimas incidencias</i> . La ventana actual se cerrará y se mostrará la ventana de <i>Últimas incidencias</i> .	Se mostrará la ventana de <i>Últimas incidencias</i> con las 10 últimas incidencias registradas.	RS-F06, RS-F07, RS-F09, RS-I23, RS-I34, RS-I35, RS-I36, RS-A07, RS-A08, RS-A09
PRU-A05	Añadir favorito	Se iniciará la aplicación. Se pulsará sobre el botón <i>Propiedades</i> o <i>Propiedades</i> . Se pulsará, en la esquina superior derecha sobre el desplegable de idiomas y	La ventana actual se cerrará y se mostrará la ventana principal de conexión. En la lista de favoritos se	RS-I11, RS-I12, RS-I13, RS-I24

Identificador	Título	Descripción	Resultado esperado	Proviene de
		se seleccionará el idioma Español. Se introducirá el texto "192.168.0.41" (esta dirección IP no debe estar en la Base de Datos) en el apartado de <i>Dirección IP</i> . Se pulsará sobre el botón <i>Admin</i> . La ventana actual se cerrará y se mostrará la ventana para administrar las RSU. Se introducirá el texto "RSU 2" en el apartado de <i>Nombre</i> . Se pulsará sobre el botón <i>Guardar</i> . Se cerrará la ventana para administrar las RSU y se mostrará la ventana de <i>Propiedades</i> . Se pulsará sobre el botón <i>Ok</i> .	mostrará la nueva RSU añadida.	
PRU-A06	Editar favorito	Se iniciará la aplicación. Se pulsará sobre el botón <i>Properties</i> o <i>Propiedades</i> . Se pulsará, en la esquina superior derecha sobre el desplegable de idiomas y se seleccionará el idioma Español. Se introducirá el texto "192.168.0.41" (esta dirección IP debe estar en la Base de Datos) en el apartado de <i>Dirección IP</i> . Se pulsará sobre el botón <i>Admin</i> . La ventana actual se cerrará y se mostrará la ventana para administrar las RSU. Se introducirá el texto "RSU 3" en el apartado de <i>Nombre</i> . Se pulsará sobre el botón <i>Guardar</i> . Se cerrará la ventana para administrar las RSU y se mostrará la ventana de <i>Propiedades</i> . Se pulsará sobre el botón <i>Ok</i> .	La ventana actual se cerrará y se mostrará la ventana principal de conexión. En la lista de favoritos se mostrará la nueva RSU editada.	RS-I14, RS-I15, RS-I16, RS-I25
PRU-A07	Eliminar favorito	Se iniciará la aplicación. Se pulsará sobre el botón <i>Properties</i> o <i>Propiedades</i> . Se pulsará, en la esquina superior derecha sobre el desplegable de idiomas y se seleccionará el idioma Español. Se introducirá el texto "192.168.0.41" (esta dirección IP debe estar en la Base de Datos) en el apartado de <i>Dirección IP</i> . Se pulsará sobre el botón <i>Admin</i> . La ventana actual se cerrará y se mostrará la ventana para administrar las	La ventana actual se cerrará y se mostrará la ventana principal de conexión. En la lista de favoritos ya no se mostrará la RSU eliminada.	RS-I17, RS-I26

Identificador	Título	Descripción	Resultado esperado	Proviene de
		RSU. Se pulsará sobre la opción de <i>Eliminar favorito</i> . Se pulsará sobre el botón <i>Guardar</i> . Se cerrará la ventana para administrar las RSU y se mostrará la ventana de <i>Propiedades</i> . Se pulsará sobre el botón <i>Ok</i> .		
PRU-A08	Ping	Se iniciará la aplicación. Se pulsará sobre el botón <i>Properties</i> o <i>Propiedades</i> . Se pulsará, en la esquina superior derecha sobre el desplegable de idiomas y se seleccionará el idioma Español. Se introducirá el texto "192.168.0.40" en el apartado de <i>Dirección IP</i> . Se pulsará sobre el botón <i>Test</i> . La ventana actual se cerrará y se mostrará la ventana de <i>Ping</i> .	Se mostrará un texto indicando que se está llevando a cabo el ping. Tras unos segundos el texto cambiará, indicando que la conexión es correcta. Se mostrará una imagen con un "tic" verde.	RS-I19, RS-I20
PRU-A09	Desconectar y cerrar	La prueba comenzará desde la ventana de <i>E-Save</i> en el idioma Español. Se pulsará sobre el botón de salir.	Se mostrará un mensaje indicando que se va a proceder a la desconexión de la RSU con el host y se cerrará la aplicación.	RS-I43, RS-I44
PRU-A10	Adaptabilidad Interfaz	Se iniciará la aplicación. Se pulsará sobre el botón para maximizar la ventana.	La ventana se adaptará al tamaño de la pantalla utilizada.	RS-I27, RS-P04
PRU-A11	Esquema XML envío sanciones	El registro de las sanciones en la Base de Datos sigue el modelo XML de envío de sanciones.	Se comprobará que el modelo de la Base de Datos correspondiente al apartado 4.4 Diagrama relacional de la Base de Datos contiene todos los campos que se aprecian en el apartado 9.1. Anexo 1: Estructura sanciones	RS-FI01, RS-FI02, RS-FI03, RS-FI04, RS-FI05, RS-FI06, RS-FI07, RS-FI08, RS-FI09, RS-FI10, RS-FI11, RS-FI12, RS-FI13, RS-FI14, RS-FI15, RS-FI16, RS-FI17, RS-FI18, RS-FI19, RS-FI20, RS-FI21, RS-FI22, RS-FI23, RS-FI24, RS-FI25, RS-FI26, RS-FI27, RS-FI28

Identificador	Título	Descripción	Resultado esperado	Proviene de
PRU-A12	About	Se iniciará la aplicación. Se pulsará sobre el logo de la UC3M.	Aparecerá una nueva ventana donde vendrán los nombres de aquellas personas colaboradoras en el proyecto, así como los desarrolladores. Al situar el ratón encima de cualquier logo de la UC3M aparecerá el identificador del proyecto.	RS-I28, RS-I29, RS-I30, RS-I31, RS-I32, RS-I33
PRU-A13	Esquema XML fichero acuses de recibo	Se abrirá una ventana de línea de comandos. Se introducirá la siguiente instrucción "telnet 192.168.0.40". Se introducirá el texto "root" cuando pida el usuario. Se introducirá "password" cuando pida la contraseña. Se introducirá la siguiente instrucción "vi /var/log/logFines.xml" (debe existir el fichero con al menos un registro).	Se mostrará el fichero XML dentro de la línea de comandos. Se comprobará que el fichero tiene la misma estructura que el mostrado en el apartado 9.2. Anexo 2: Estructura fichero acuses de recibo.	RS-FI29, RS-FI30, RS-FI31, RS-FI32
PRU-A14	Recoger denuncias ocultas	La prueba comenzará desde la ventana de <i>E-Save</i> en el idioma Español.	En la ventana actual se mostrará un botón con el texto "Recibir denuncias ocultas".	RS-I37
PRU-A15	Recibir evidencias de defensa de una sanción	La prueba comenzará desde la ventana de <i>E-Save</i> en el idioma Español.	En la ventana actual se mostrará un botón con el texto "Recibir evidencias def.".	RS-I38
PRU-A16	Configurar RSU por tramos	La prueba comenzará desde la ventana de <i>E-Save</i> en el idioma Español. Se pulsará sobre el botón "Dynamic speed control".	La ventana de <i>E-Save</i> se cerrará y aparecerá una nueva ventana que contendrá las opciones de configuración de la RSU para establecer un tramo de control.	RS-I39, RS-I40, RS-I41
PRU-A17	Resumen de controles RSU	La prueba comenzará desde la ventana de <i>E-Save</i> en el idioma Español.		RS-I42

Tabla 12. Pruebas de Aceptación

3. Diseño Detallado

En este capítulo se explica con más detalle la arquitectura del sistema explicada en el punto [3.6.2. Arquitectura preliminar](#) teniendo en cuenta las limitaciones del sistema que se detallan en el punto [3.2. Determinación del Alcance del Sistema](#).

En primer lugar, se explicarán con detalle cada uno de los componentes del sistema final, sus funciones así como las clases que se han programado para que se cumplan todos los requisitos propuestos. Por último, se detallarán los pasos que de realizar el usuario, así como los pasos que realizará el sistema para completar cada caso de uso mediante diagramas de secuencia.

3.1. Diseño de componentes

Aquellas clases que forman las distintas ventanas que componen la Interfaz de Usuario tienen UI al final de su nombre, por ejemplo, sabiendo que Index es una ventana de Interfaz, el nombre de la clase será *IndexUI*. Aquellas clases del sistema que no sean de la Interfaz, aunque la apoyen ofreciendo algún servicio, tendrán la nomenclatura normal de una clase: empezará por mayúscula y no tendrá UI al final del nombre.

3.1.1. Componente 1: Vista

El primer componente que se va a explicar se corresponde con la vista, o Interfaz de Usuario. La interfaz tiene tres grandes tareas, que se dividen en tres componentes bien diferenciados. La primera tarea se refiere a la conexión de la interfaz con la RSU mediante un protocolo de conexión. En este caso, el protocolo de conexión viene determinado por el fabricante y se corresponde con el protocolo *Telnet*. Esta primera tarea se ve reflejada en el sub-componente *Conexión*.

La segunda tarea se corresponde con el envío de una sanción a la OBU. La interfaz, tras realizar la conexión con la RSU, podrá enviar la instrucción necesaria a la RSU para que ésta envíe una sanción a la OBU. Por último, la interfaz registrará la sanción enviada. Esta tarea se corresponde con el sub-componente *Enviar sanción*.

La última tarea se corresponde con la visualización de las últimas sanciones registradas. La interfaz permitirá visualizar los datos de un determinado número de sanciones: su identificador, fecha y hora, estado de recepción del acuse de recibo, etc. El componente *Enviar sanción* también permite llegar a esta tarea, que se correspondería con el sub-componente *Visualizar sanciones*.

La Ilustración 15 muestra el diagrama de sub-componentes del componente *Vista*:

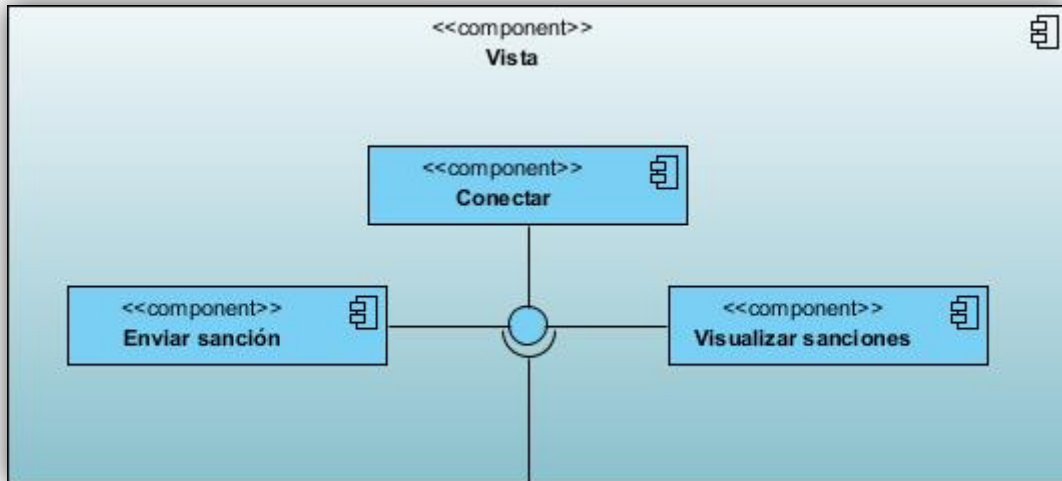


Ilustración 15. Componente 1: Vista

En los siguientes apartados se procede a desglosar cada uno de los sub-componentes pertenecientes al componente *Vista*. Se explicará el funcionamiento detallado de cada sub-componente y se mostrará un diagrama de clases asociado a cada uno de ellos.

3.1.1.1. *Conectar*

Este componente es la parte de la interfaz orientada a conectarse a la RSU, es decir, no está conectada a la RSU pero permite su conexión a distintas RSU, así como configurar la aplicación: cambiar el idioma, administrar la lista de RSU favoritas, probar la conexión de una dirección, etc.

La pantalla principal permite la conexión introduciendo a una RSU elegida de la lista de RSU favoritas, su usuario y su contraseña. Si la RSU a la que queremos conectarnos no está en la lista, accederemos al apartado de propiedades, donde se pueden añadir RSU de forma temporal introduciendo una dirección IP válida, o bien se pueden añadir a la lista de favoritos. Por otra parte se pueden editar o eliminar la lista de favoritos. Por último, se puede comprobar la conexión con la RSU introducida mediante la utilización de un *Ping*.

La Ilustración 16 muestra el diagrama de clases que representa este componente:



Ilustración 16. Diagrama de clases 1 - Conectar

Como se puede apreciar en la ilustración anterior, este componente se divide en 5 clases de interfaz de usuario. Algunas de las clases, como *IndexUI*, *PropertiesUI*, *AdminFavoritesUI* y *PingUI*, son aquellas clases de la interfaz orientadas a no utilizarlas con conexión a la RSU. Por otro lado, la clase *E-SaveUI* es la clase que se corresponde con la pantalla principal de la aplicación tras su inicio de sesión con la RSU.

- **IndexUI:** Clase de la interfaz que se corresponde con la ventana principal del sistema. Tiene como objetivo ofrecer al usuario la posibilidad de conectarse con cualquier RSU que esté en la red. La clase comprueba que se han introducido datos válidos para realizar la conexión con una RSU (que exista el usuario y la contraseña, etc.). Gracias a esta clase se puede también acceder al apartado de propiedades, que se correspondería con la clase *PropertiesUI*. Si el usuario realiza una conexión con una RSU de forma satisfactoria se instanciará la clase *E-SaveUI*. Esta clase tendrá un cuadro de texto destinado a mostrar todos los mensajes informativos relevantes del sistema.
- **PropertiesUI:** Clase de la interfaz correspondiente al apartado de propiedades de la aplicación. En esta ventana se permite escribir de forma manual una dirección IP de una RSU conocida. La clase comprueba el formato de la dirección IP y avisará al usuario si la dirección IP no tiene un formato correcto (el formato de la dirección IP será X.X.X.X donde X se corresponde con un número comprendido entre 0 y 255). Si se escribe una dirección IP que no esté dentro de la lista de RSU favoritas y se pulsa la opción de volver a la pantalla de conexión *IndexUI*, el sistema utilizará esa dirección IP de forma temporal hasta que se cierre la Interfaz de Usuario. Desde esta clase se puede acceder a la sección de gestión de RSU favoritas correspondiente a la clase *AdminFavoritesUI*. Por último, esta clase permite que se compruebe el estado de conexión de la dirección IP que se ha escrito anteriormente, que se correspondería con la clase *PingUI*.
- **AdminFavoritesUI:** Clase de la interfaz correspondiente a la administración de RSU favoritas del sistema. En esta ventana se puede añadir nuevas RSU favoritas o bien modificar o eliminar cada RSU favorita almacenada. La dirección IP utilizada se recoge de la ventana de propiedades, aunque puede ser modificada por la dirección IP que se desee. La clase detecta si la dirección IP escrita se corresponde con una dirección conocida (se permiten la modificación y eliminación, pero se bloquea la inserción) o desconocida (se permite la inserción, pero se bloquean la modificación y la eliminación). En el caso de añadir o modificar una RSU favorita se puede escribir un nombre identificativo, de estar vacío se utiliza la dirección IP como nombre.
- **PingUI:** Clase de la interfaz correspondiente a la comprobación de la conectividad con la RSU. Esta ventana comprueba inicialmente, sin acciones adicionales del usuario, si la RSU está disponible según la dirección anteriormente escrita en el apartado de propiedades. Se muestra tanto una imagen como un mensaje descriptivo que indica si la RSU está disponible o no. La comprobación de conectividad se puede realizar tantas veces como se quiera sin tener que volver a la ventana de propiedades.
- **E-SaveUI:** Única clase de este componente que se utiliza tras la conexión con la RSU. Esta ventana permite el uso de los componentes *Enviar sanción* y *Visualizar sanciones* que se explicarán a continuación. La ventana también muestra secciones no implementadas pensadas para las líneas futuras del proyecto, como la configuración

de RSU por tramos. Esta clase contendrá un cuadro de texto donde se expondrán los mensajes del sistema que puedan ser relevantes para el usuario. Por otro lado, también contendrá una barra de progreso que sólo será visible durante la recepción de los acuses de recibo y servirá para marcar el estado de progreso de la recepción de datos.

3.1.1.2. *Enviar sanción*

Este componente es la parte de la interfaz orientada al envío de sanciones de la RSU a la OBU. Para poder enviar una sanción, previamente el usuario debe haber iniciado sesión en la RSU. Tras el inicio de sesión, se mostrará al usuario la ventana principal de operaciones de la RSU, donde se pueden realizar todas las acciones de la RSU propuestas para este proyecto. Una de las acciones principales que se pueden hacer en esta ventana será el envío de una sanción a la OBU.

Cuando se quiera enviar una sanción a la OBU, primeramente se mostrará un resumen con los datos más relevantes de la sanción. En el caso de continuar con el envío de la sanción, la interfaz ejecutará el comando necesario y preguntará al usuario si desea ver las últimas sanciones impuestas o bien desea volver a la pantalla principal para realizar alguna otra operación. La visualización de las últimas sanciones impuestas se explica en el siguiente apartado, que se corresponde con el componente *Visualizar sanciones*.

La Ilustración 17 muestra el diagrama de clases correspondiente a este componente:



Ilustración 17. Diagrama de clases 2 - Enviar sanción

Como podemos ver en la ilustración, la interfaz que se corresponde con el envío de sanciones consta de dos clases principales, la pantalla principal (*E-SaveUI*) y la clase que realiza el envío de la sanción (*SendNotificationUI*).

Se ha de mencionar que el atributo que *idMsg* se correspondería con un número auto-incrementable para el envío de sanciones. Cada envío de una sanción requiere un identificador, pero si se envía una sanción con el mismo identificador será descartado. Para ello, esta variable se incrementará en 1 mientras se mantenga la conexión con la RSU.

La clase *E-SaveUI* permite la introducción de los datos de la sanción que se va a enviar por parte del usuario. Inicialmente se rellenan todos los campos de forma automática con datos estáticos de ejemplo, salvo los campos de fecha y hora que se establecen en la fecha y hora en que se instanció esta clase. Cuando el usuario haya rellenado todos los campos de forma correcta y proceda con el envío de la sanción, esta clase realizará las acciones oportunas para enviar el comando de envío de sanción desde la RSU.

3.1.1.3. Visualizar sanciones

Este componente es la parte de la interfaz orientada a la visualización de sanciones de la RSU. Para poder visualizar una lista de sanciones, previamente el usuario debe haber iniciado sesión en la RSU. Tras el inicio de sesión, se mostrará al usuario la ventana principal de operaciones de la RSU, donde se pueden realizar todas las acciones de la RSU propuestas para este proyecto. Una de las acciones principales que se pueden hacer en esta ventana será la visualización de las últimas sanciones impuestas.

Cuando el usuario quiera ver las últimas sanciones impuestas se recuperará un número determinado de sanciones registradas, junto con todos sus datos (id del mensaje, fecha y hora del envío, estado de recepción, etc.).

Este componente también tendría como acción la recogida de acuses de recibo. Se ha decidido que los acuses de recibo se recojan de forma independiente a la visualización de las últimas sanciones, puesto que se pueden querer recibir los acuses de recibo pero no querer ver las últimas sanciones y viceversa.

Se ha decidido que no es necesaria la creación de una nueva ventana para representar la recepción de datos, sino que se utiliza la propia clase *E-SaveUI* para mostrar el estado de la recepción. El estado de recepción se muestra con una barra de progreso que sólo es visible durante la recepción, mientras que el estado de finalización de la recepción se muestra en el cuadro de texto destinado a los mensajes relevantes del sistema.

La Ilustración 18 muestra el diagrama de clases correspondientes a este componente:

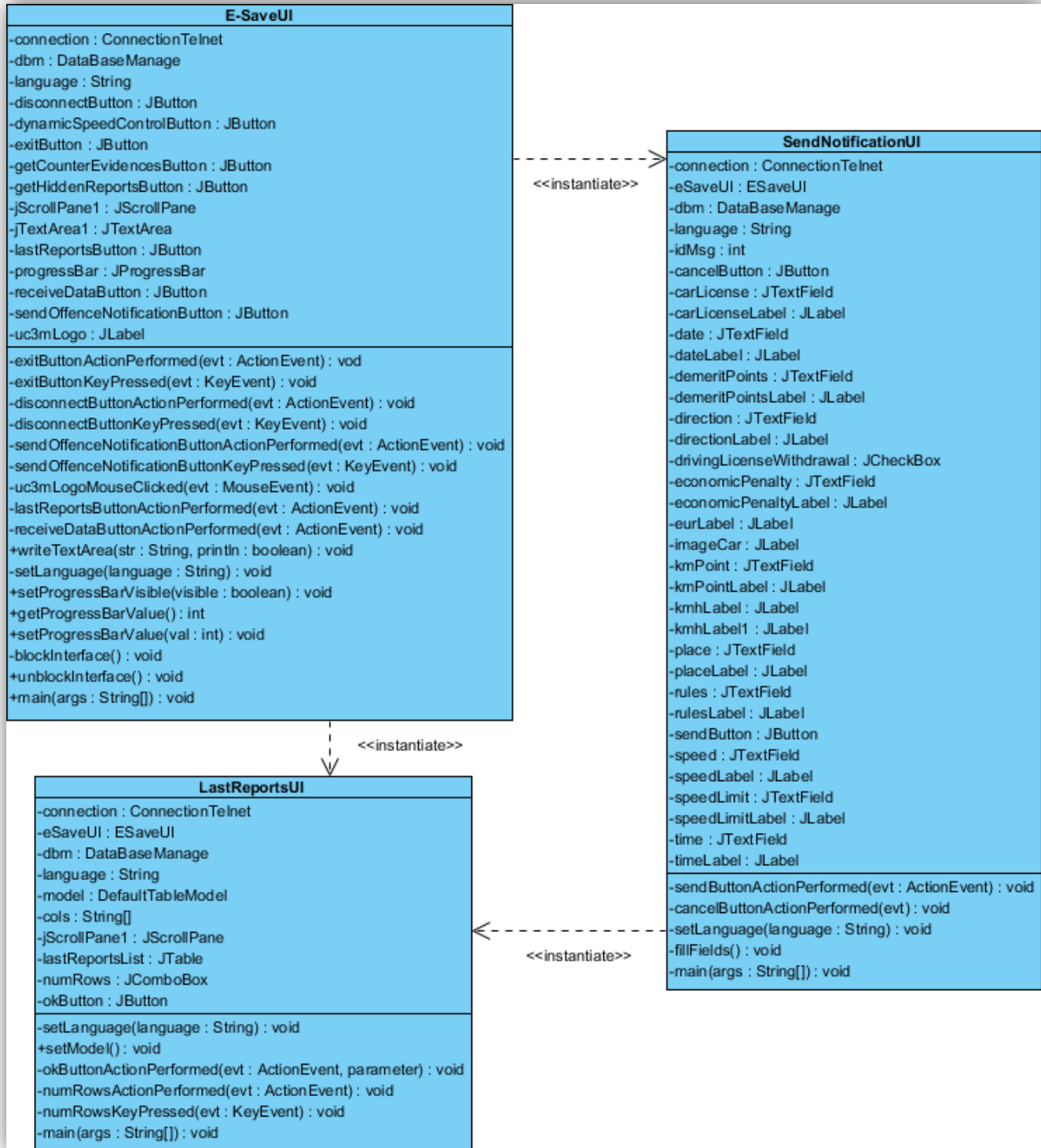


Ilustración 18. Diagrama de clases 3 - Visualizar sanciones

Como se puede ver en la ilustración anterior, el componente se divide en 3 clases, donde también se puede la clase *SendNotificationUI*, que se corresponde con el componente *Enviar sanción* previamente explicado. Esta clase pertenece también a este componente puesto que es posible visualizar las últimas sanciones impuestas tras el envío de una sanción.

La clase *LastReportsUI* recibirá la variable que mantiene la conexión con la base de datos, de forma que se pueda consultar las últimas sanciones impuestas sin tener que abrir una nueva conexión con la base de datos. De forma predeterminada, la clase realizará una consulta por las diez últimas sanciones, pero se podrá modificar esta consulta mediante un desplegable que permite buscar un mayor número de sanciones (permite realizar búsquedas de 10, 20, 50 y 100 sanciones).

Se ha de mencionar que al no tratarse de un sistema automatizado, el número de sanciones diarias se vería muy reducido, por lo que una búsqueda mayor actualmente no tendría mucha utilidad. Generalmente no se suelen realizar búsquedas de este tipo para listas mayores de un determinado número, por lo que aumentar dicha cantidad no tendría gran utilidad para el sistema. Para ello se utilizaría una búsqueda de sanciones mejorada que realizaría la búsqueda mediante la introducción de datos y selección con múltiples campos de búsqueda. Para más información, véase el apartado [6.2. Líneas Futuras](#).

3.1.2. Componente 2: Modelo

El siguiente componente que se va a presentar será el Modelo del sistema, que se corresponde con la utilización de una Base de Datos como método para el almacenamiento y recuperación de información. Este componente será el encargado de realizar todas las operaciones sobre la Base de Datos, tanto las inserciones y actualizaciones como las búsquedas.

Se ha de destacar que se ha utilizado este componente para conseguir que la interfaz sea multi-lenguaje. Se han almacenado en la base de datos todos los textos de las etiquetas, botones y mensajes de la interfaz, almacenando como claves la ventana en la que se encuentra el usuario y el lenguaje seleccionado para recuperar el texto. Dentro de una ventana y un lenguaje determinados sólo tendremos que identificar dónde poner cada texto, para lo que se ha utilizado el nombre de los distintos elementos de la interfaz. Por ejemplo, si estuviéramos en la clase *E-SaveUI* con idioma elegido inglés, la Base de Datos recuperaría todos los textos correspondientes a esa clase de Interfaz. A continuación se buscaría el botón de envío de sanciones (por ejemplo) que tendría la etiqueta univoca *sendOffenceNotificationButton* y le aplicaría el texto recuperado para ese identificador de elemento de interfaz.

En este caso, el componente *Modelo* no tiene ningún sub-componente, por lo que no será necesaria su representación mediante una ilustración como los demás apartados.

En el siguiente apartado se explicará el funcionamiento detallado de este componente y se mostrará un diagrama de clases asociado al mismo.

3.1.2.1. *Modelo*

Este componente, como se ha dicho, realizará todas las operaciones en las que intervenga la Base de Datos. Puesto que se van a recuperar todos los textos de todas las ventanas de la Interfaz de Usuario mediante esta Base de Datos, la conexión a la misma se mantendrá mientras la interfaz permanezca abierta. Tras el cierre de la interfaz se cerrará la conexión con la Base de Datos, de forma que se aprovechen mejor los recursos del equipo.

Se ha de mencionar que la Base de Datos está almacenada dentro del equipo host de forma local. Por otro lado, sólo se pretende usar una única base de datos, por lo que la dirección de conexión de la misma no será variable.

En esta clase no se tratará ningún tipo de dato, sólo se realizarán consultas, inserciones, actualizaciones y eliminaciones sobre la Base de Datos. Si es necesario el tratamiento de datos (por ejemplo durante las consultas), se devolverán los datos para que los trate la clase que ha llamado al *Modelo*.

Se ha seguido un criterio de nombramiento de métodos por el cual se diferencian de forma fácil en qué consiste cada método. Siguiendo este criterio los métodos comenzarán su nombre indicando qué tipo de operación realizan sobre la Base de Datos (consulta, inserción, etc.), a continuación irá un texto que identifique sobre qué se realiza la operación (textos de interfaz, sanciones, etc.). Un ejemplo sería *selectLastOptions()* que se corresponde con la consulta de las últimas opciones (idioma y dirección de RSU) elegidas en la interfaz antes del cierre de la misma.

MySQL proporciona conectividad para las aplicaciones cliente desarrolladas en el lenguaje de programación Java a través de un controlador JDBC. Para poder realizar la conexión con la base de datos MySQL se ha utilizado el conector *mysql-connector-java5.1.18-bin* [29].

El manejo de las posibles excepciones que puedan surgir durante el tratamiento con la Base de Datos no está supervisado en esta clase, sino que todas las excepciones son “lanzadas”. El método de cualquier clase que haga uso de la Base de Datos tratará los errores de forma distinta, dependiendo de para qué se utilice. Generalmente, las excepciones que surjan con el uso de la Base de Datos mostrará un mensaje indicando qué error ha ocurrido para que el usuario sea consciente y pueda informar del error, si la excepción impide la ejecución correcta de la aplicación, ésta se cerrará tras el mensaje de error.

En la Ilustración 19 se muestra el diagrama de clases correspondiente a este componente:

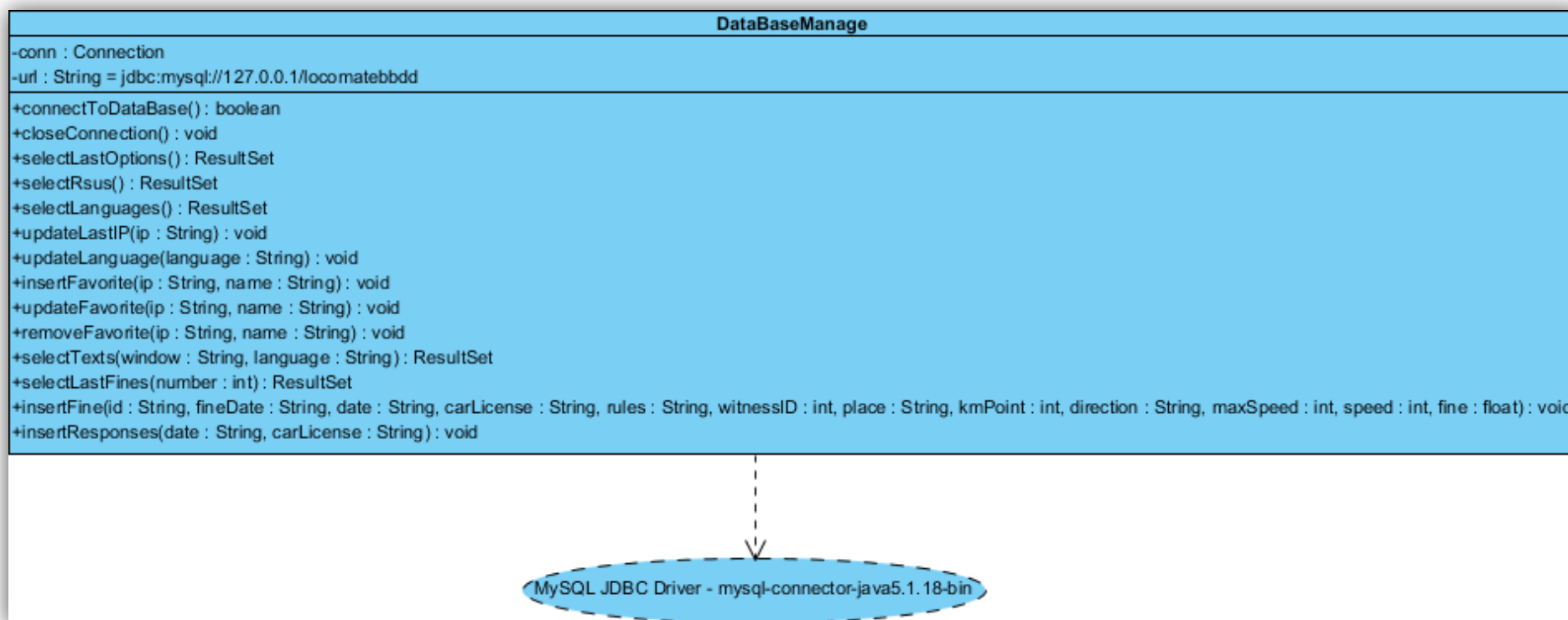


Ilustración 19. Diagrama de clases 4 - Modelo

3.1.3. Componente 3: Controlador

Este componente es el encargado de realizar todas aquellas operaciones necesarias para la aplicación que no están contempladas en los componentes *Vista* y *Modelo*. En este caso, este componente recibe y trata todos los eventos solicitados por la *Vista* que no requieren de su intervención, como por ejemplo comprobar la conexión con una RSU.

Este componente se puede diferenciar en dos sub-componentes: el sub-componente *Conexión* estará dirigido al establecimiento de conexión o inicio de sesión con la RSU, mientras que el sub-componente *Operaciones* está destinado a realizar todas aquellas operaciones auxiliares necesarias o útiles para el sistema.

La Ilustración 20 muestra el diagrama de sub-componentes del componente *Controlador*:

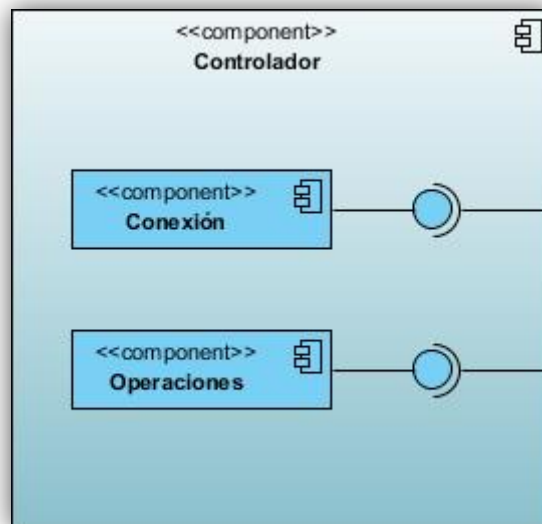


Ilustración 20. Componente 3: Controlador

En los siguientes apartados se procede a desglosar cada uno de los sub-componentes pertenecientes al componente *Controlador*. Se explicará el funcionamiento detallado de cada sub-componente y se mostrará un diagrama de clases asociado a cada uno de ellos.

3.1.3.1. Conexión

Como se ha mencionado anteriormente, este componente está orientado al inicio de sesión con la RSU y al uso de la conexión establecida para mandar instrucciones a la RSU. El fabricante establece que la conexión debe ser establecida mediante el protocolo *Telnet*, por lo que se ha decidido utilizar la librería *commons-net-3.2* [30] para crear un cliente. El uso de esta librería

permite establecer una conexión con la RSU con un lenguaje de programación de alto nivel (menos técnico). También ofrece una serie de operaciones interesantes para el desarrollo del sistema, como la comprobación de conexión dada una dirección IP.

Esta clase, por tanto, está orientada al uso de esta librería, por lo que la mayoría de los métodos se utilizarán para realizar las llamadas necesarias para aprovechar las funcionalidades de la librería ajustándolo al proyecto desarrollado.

En la Ilustración 21 vemos el diagrama de clases que se corresponde con este componente:

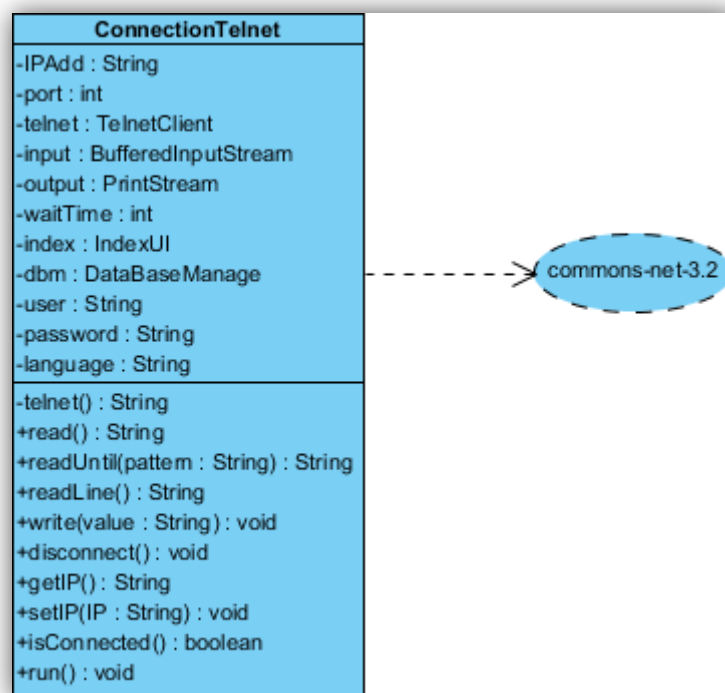


Ilustración 21. Diagrama de clases 5 - Conexión

Como vemos en el diagrama de clases anterior, el cliente *Telnet* mencionado anteriormente se correspondería con la variable *telnet* de tipo *TelnetClient*. Es interesante mencionar que el método *telnet()* sería el método que realizaría la conexión tras la preparación de los datos. Los métodos que contienen “*read*” se utilizan para la lectura de las respuestas de la RSU.

El método *isConnected()* se utiliza para comprobar la conectividad de la RSU. Al comprobar la conectividad de la RSU se recibirá el estado de la conectividad de la RSU, la cual se devolvería a la clase que utilice este método para comprobar la conexión y así tratar ese estado (dispositivo disponible, dispositivo no disponible, error de red, etc.)

Gracias al método *write()* se podrán enviar instrucciones a la RSU dentro de la conexión *Telnet* establecida sin bloquear otros procesos del sistema. Las instrucciones más relevantes que se desean enviar son la de envío de sanción a la OBU y la del establecimiento de conexión.

3.1.3.2. Operaciones

Este componente está destinado a realizar todas aquellas operaciones de apoyo con el *Modelo* y la *Vista*. Gracias a este componente podremos comprobar si un dispositivo está disponible antes de realizar la conexión, ejecutar instrucciones relevantes para el sistema o bien tratar el fichero de registro de acuses de recibo. Estas tareas se corresponderían con las clases *Ping*, *Execute* y *readXML* respectivamente.

En la Ilustración 22 podemos ver el diagrama de clases correspondiente a este componente:

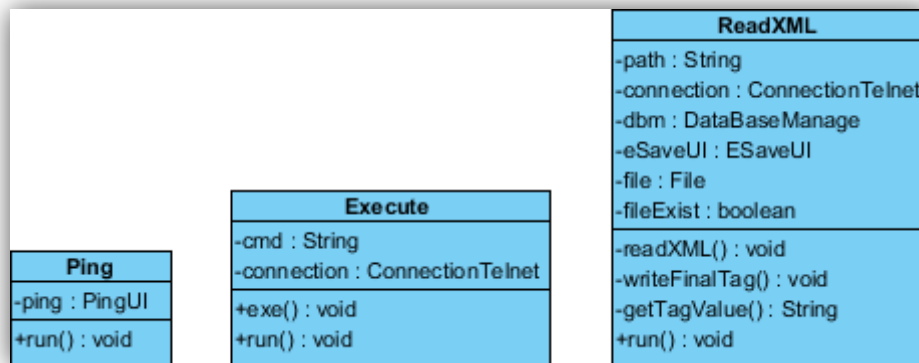


Ilustración 22: Diagrama de clases 6 - Operaciones

Como podemos ver en la ilustración anterior, para este caso concreto, ninguna de las clases están relacionadas entre sí. Esto es debido a que no hacen uso unas de otras, sino que son utilizadas por otros componentes del sistema de forma independiente a las demás clases de este componente.

La clase *Execute* permite ser bloqueante con el sistema (la interfaz se bloquea) o no bloqueante. Esto puede ser útil para algunos procesos, como la importación de acuses de recibo, puesto que no nos interesaría salir de la aplicación antes de terminar la importación.

Se ha de señalar que la escritura de las etiquetas por parte de la RSU estará incompleta. Cuando se reciba el fichero de registro de acuses de recibo faltará la etiqueta de cierre de fichero XML: `</responses>`. Se ha decidido esto para ahorrar tiempo de tratamiento de ficheros en la RSU, de lo contrario, se tendría que estar moviendo dicha etiqueta continuamente tras la escritura de un acuse de recibo. Este problema se solventaría con el método `writeFinalTag()` perteneciente a la clase *ReadXML*. Aunque lo recomendable es que exista una etiqueta de comienzo de fichero XML y otra de finalización, sería posible la lectura del fichero XML a pesar de no tener la etiqueta final.

3.1.4. Componente 4: Gestión de mensajes

Este componente comprendería toda la gestión del protocolo de envío de las sanciones en el lado de la RSU.

El primer sub-componente, llamado *Enviar sanción*, se corresponde con el envío de sanciones desde la RSU a la OBU. La RSU envía una sanción a la OBU con una serie de parámetros preestablecidos y un identificador de mensaje que provendría de la Interfaz de Usuario. Tras el envío de la sanción, la RSU finalizaría el envío y comenzaría con la recepción de acuses de recibo. Esta parte se corresponde con el siguiente sub-componente explicado a continuación.

El segundo sub-componente, llamado *Recibir acuses de recibo*, se corresponde con la recepción y gestión de los acuses de recibo enviados por la OBU. La RSU comenzaría la escucha de acuses de recibo. En cuanto se recibiera un acuse de recibo por parte de la OBU, la RSU procesaría el mensaje enviado y registraría el acuse de recibo si todo estuviera correcto.

La Ilustración 23 muestra el diagrama de sub-componentes del componente *Gestión de mensajes*:

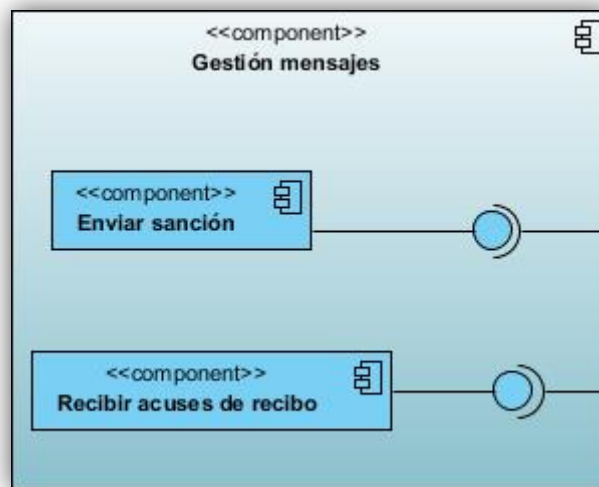


Ilustración 23. Componente 4: Gestión de mensajes

En los siguientes apartados se procede a desglosar cada uno de los sub-componentes pertenecientes al componente *Gestión de mensajes*. Se explicará el funcionamiento detallado de cada sub-componente y se mostrará un diagrama de clases asociado a cada uno de ellos.

3.1.4.1. *Enviar sanción*

Este componente es el encargado de realizar el envío de una sanción desde la RSU. Para poder realizar el envío tendremos que establecer unos parámetros iniciales relevantes, como por ejemplo el canal de envío, modo de utilización del canal, el proveedor de servicios que se utilizará, etc. Inicialmente, esta clase realizará una serie de funciones de inicialización que habilitarán el driver WAVE para el envío de mensajes DSRC.

La clase *snd* hace uso de una librería llamada *wave.h* que ha sido implementada por el fabricante para describir la API de los dispositivos WAVE. Esta librería contiene todas las funciones necesarias para completar el objetivo propuesto en el apartado [1.1. Objetivo](#).

Este componente, realiza los siguientes pasos con el fin de completar el envío de una sanción:

1. Registrar las funciones de envío mediante la librería *wave.h*.
2. Invoca el controlador.
3. Formular la petición de envío de mensaje.
 - 3.1. La petición de envío de mensajes se realizará múltiples veces para asegurar el envío del mensaje y la recepción por parte de la OBU.
4. Eliminar el registro cuando haya terminado el envío.
5. Llamar a la función de recepción de acuses de recibo (componente *Recibir acuses de recibo* explicado a continuación).

La Ilustración 24 muestra el diagrama de clases que representa este componente:

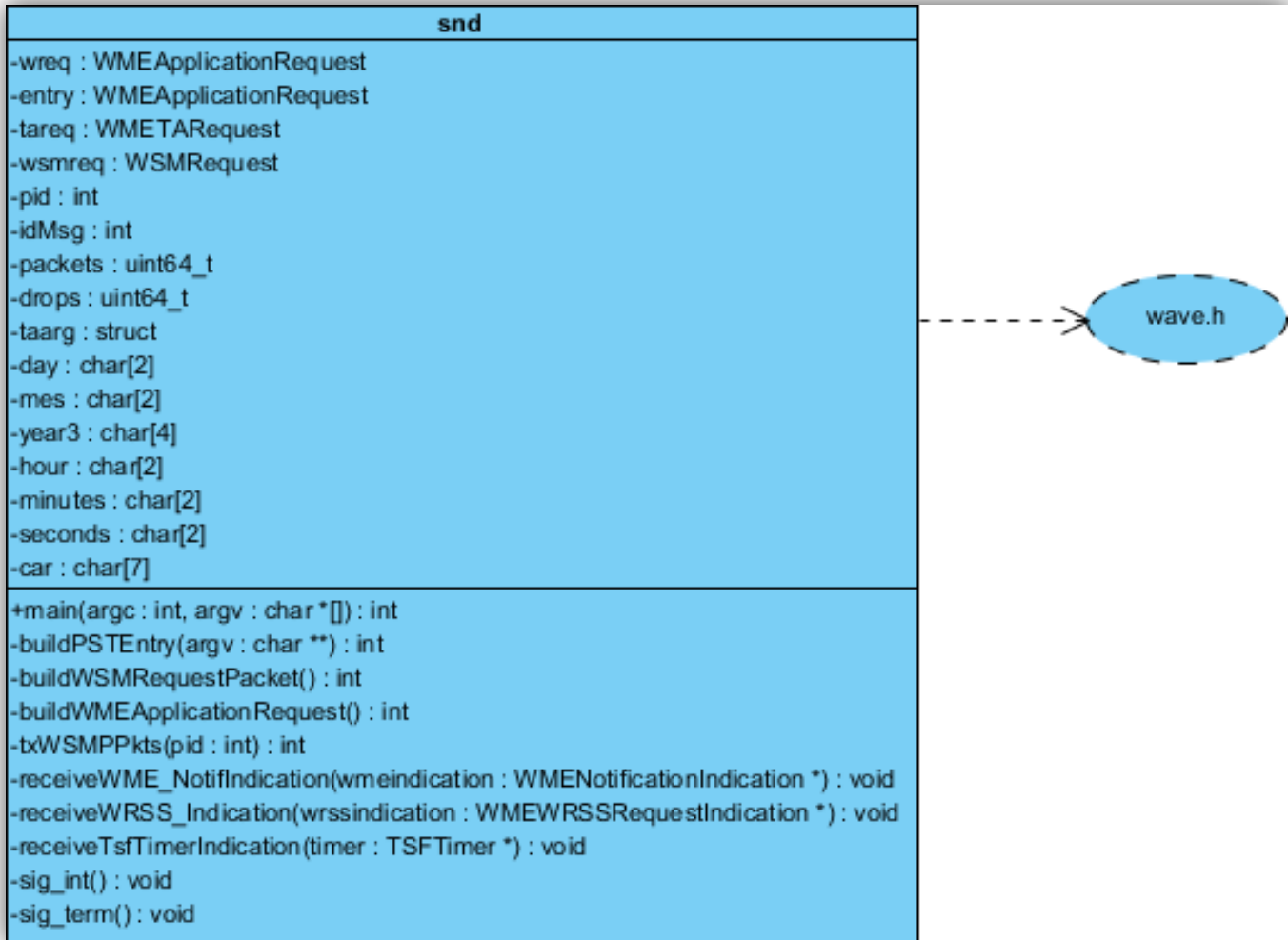


Ilustración 24. Diagrama de clases 6 - Enviar sanción

Como se puede observar en la ilustración anterior, existen una serie de métodos encargados para la configuración de la RSU, como son *buildPSTEntry()* o *buildWSMRequestPacket()*. Por otro lado, nos interesa el método *txWSMPPkts()*, que es el encargado de realizar los envíos de mensajes DSRC. La estructura del mensaje que enviará este método está definida en el apartado [2.4.2.2 Requisitos funcionales de Información](#), aunque también se puede ver la estructura del mensaje enviado en el apartado [8.1. Anexo 1: Estructura sanciones](#). Como se ha explicado anteriormente, para garantizar tanto el envío como la recepción de los mensajes enviados (recordamos que la tecnología todavía puede fallar), se realizarán varios envíos del mismo mensaje.

Cuando finalice el envío de una sanción, se borrará el proveedor anteriormente registrado para habilitar el envío y se llamará al componente *Recibir acuse de recibo*, que permitirá la recepción de mensajes por parte de la OBU.

3.1.4.2. *Recibir acuse de recibo*

Este componente es el encargado de realizar la recepción de los acuses de recibo de las sanciones enviadas. Para poder realizar la recepción tendremos que establecer unos parámetros iniciales relevantes, como por ejemplo el canal de recepción, modo de utilización del canal, el identificador usuario, etc. Inicialmente, esta clase realizará una serie de funciones de inicialización que habilitarán el driver WAVE para la recepción de mensajes DSRC.

La clase *rcv* también hace uso de la librería *wave.h*. Utilizaremos esta librería igual que en el componente anterior para aprovechar las llamadas que se han creído oportunas para la recepción de mensajes. Este componente, realiza los siguientes pasos con el fin de completar la recepción de un acuse de recibo:

1. Registrar las funciones de recepción mediante la librería *wave.h*.
2. Invocar el controlador.
3. Formular las peticiones de escucha de mensajes DSRC.
 - 3.1. Comprobar si se ha recibido algún mensaje.
 - 3.2. Permanecer en un bucle hasta que se reciba algún mensaje.
 - 3.3. Si se recibe un mensaje DSRC, se tratarán los datos recibidos.
 - 3.3.1. Se comprobará el identificador de mensaje, si es el mismo que un mensaje anterior se descartará.
 - 3.3.2. Se capturarán los datos recibidos y se registrarán en un fichero de registro.
4. Eliminar el registro de las funciones de inicio cuando haya terminado (pero no el fichero de registro de acuses de recibo).

La Ilustración 25 muestra el diagrama de clases que representa este componente:

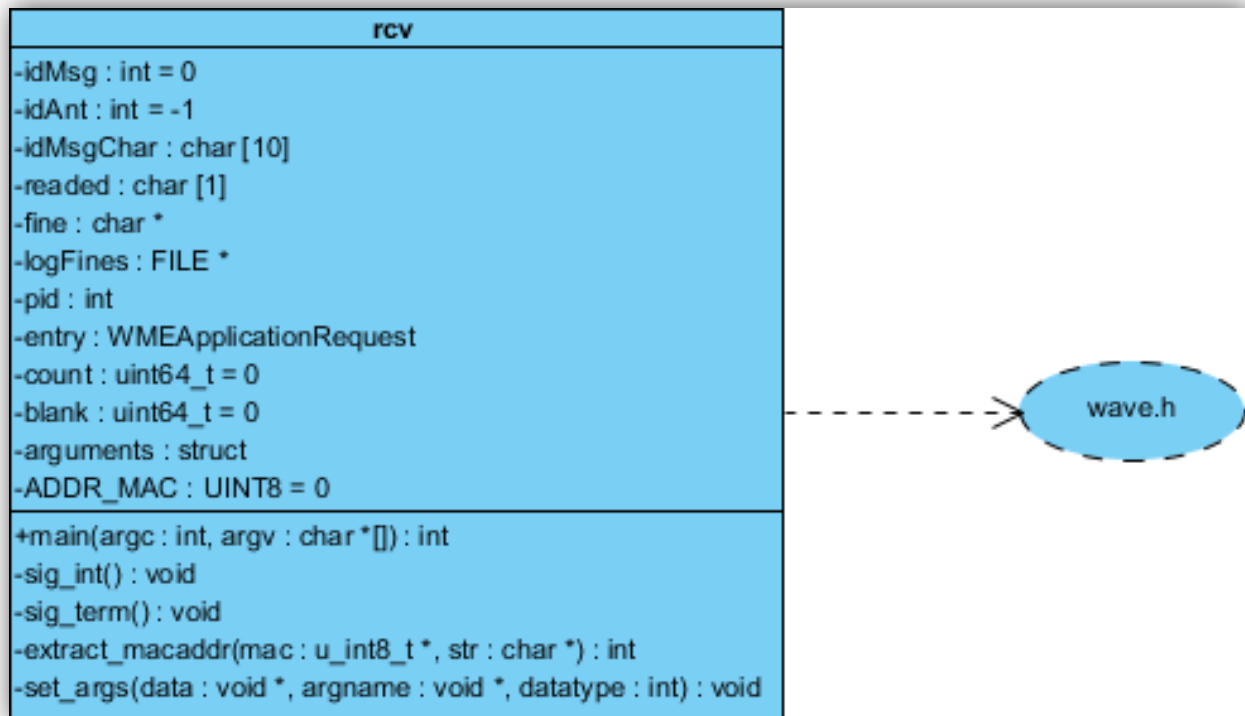


Ilustración 25. Diagrama de clases 7 - Recibir acuse de recibo

En este caso, la configuración de la RSU con el fin de poder escuchar los mensajes enviados por la OBU se realizará en el método principal *main* de la clase. Se ha de destacar que en este caso permaneceremos en un bucle hasta que se reciba algún mensaje DSRC. Cuando se reciba un mensaje de respuesta de la OBU se comprobará el identificador de mensaje para ver si el mensaje recibido ya ha sido tratado anteriormente. Si el mensaje no ha sido tratado, se recogerán los datos del acuse de recibo y se registrarán en el fichero de registro de acuses de recibo.

La estructura del fichero de registro tras la recepción de los mensajes DSRC que contengan un identificador de mensaje válido se define en el apartado [2.4.2.2 Requisitos funcionales de Información](#), aunque también se puede ver la estructura del fichero de registro en el apartado [8.2. Anexo 2: Estructura fichero acuses de recibo](#).

3.1.5. Componente 5: Respuesta mensajes

Este componente comprendería toda la gestión del protocolo de recepción y gestión de las sanciones en el lado de la OBU. Este componente es muy similar al cuarto componente anteriormente explicado: *Gestión de mensajes*. El protocolo de envío y recepción serán los mismos, salvo en los respectivo al tratamiento de los datos.

El primer sub-componente, llamado *Recibir sanción*, se corresponde con la recepción de sanciones enviadas desde la RSU a la OBU. La OBU permanece en espera comprobando la

recepción de nuevos mensajes provenientes de la RSU. Cuando recibe un mensaje de la RSU se comprueba el identificador del mensaje para descartar mensajes tratados anteriormente. Si el mensaje recibido no ha sido tratado, la OBU preparará el envío del acuse de recibo que se corresponde con el siguiente sub-componente explicado a continuación.

El segundo sub-componente, llamado *Enviar acuse de recibo*, se corresponde con el envío de mensajes que confirmen la llegada de la sanción a la OBU. La OBU se configuraría para poder realizar envíos de mensajes DSRC. Se recuerda que durante el envío de mensajes no se podrá realizar la recepción de nuevos mensajes, por lo que no se podrá producir solapamientos.

La Ilustración 26 muestra el diagrama de sub-componentes del componente *Respuesta mensajes*:

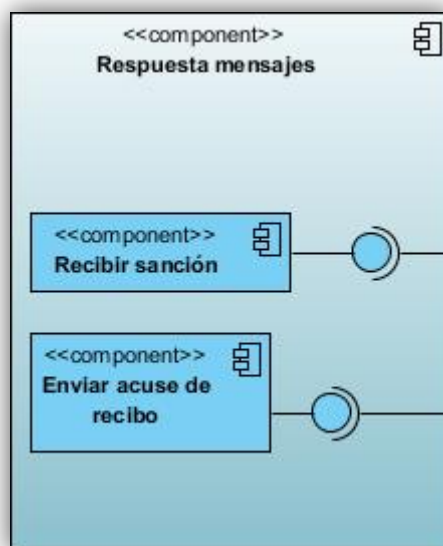


Ilustración 26. Componente 5: Respuesta mensajes

En los siguientes apartados se procede a desglosar cada uno de los sub-componentes pertenecientes al componente *Respuesta mensajes*. Se explicará el funcionamiento detallado de cada sub-componente y se mostrará un diagrama de clases asociado a cada uno de ellos.

3.1.5.1. *Recibir sanción*

Este componente es el encargado de realizar la recepción de los acuses de recibo de las sanciones enviadas. Para poder realizar la recepción tendremos que establecer unos parámetros iniciales relevantes, como por ejemplo el canal de recepción, modo de utilización del canal, el identificador usuario, etc. Inicialmente, esta clase realizará una serie de funciones de inicialización que habilitarán el driver WAVE para la recepción de mensajes DSRC.

La clase *obu* también hace uso de la librería *wave.h*. Utilizaremos esta librería igual que en el componente anterior para aprovechar las llamadas que se han creído oportunas para la recepción de mensajes. Este componente, realiza los siguientes pasos con el fin de completar la recepción de una sanción:

1. Registrar las funciones de recepción mediante la librería *wave.h*.
2. Invocar el controlador.
3. Formular las peticiones de escucha de mensajes DSRC.
 - 3.1. Comprobar si se ha recibido algún mensaje.
 - 3.2. Permanecer en un bucle hasta que se reciba algún mensaje.
 - 3.3. Si se recibe un mensaje DSRC, se tratarán los datos recibidos.
 - 3.3.1. Se comprobará el identificador de mensaje, si es el mismo que un mensaje anterior se descartará.
 - 3.3.2. Se capturarán los datos recibidos.
 - 3.3.3. Eliminar el registro de las funciones de inicio cuando haya terminado (pero no el fichero de registro de acuses de recibo).
 - 3.3.4. Se procederá al envío del acuse de recibo (siguiente componente a explicar: *Enviar acuse de recibo*).

La Ilustración 27 muestra el diagrama de clases que representa este componente:

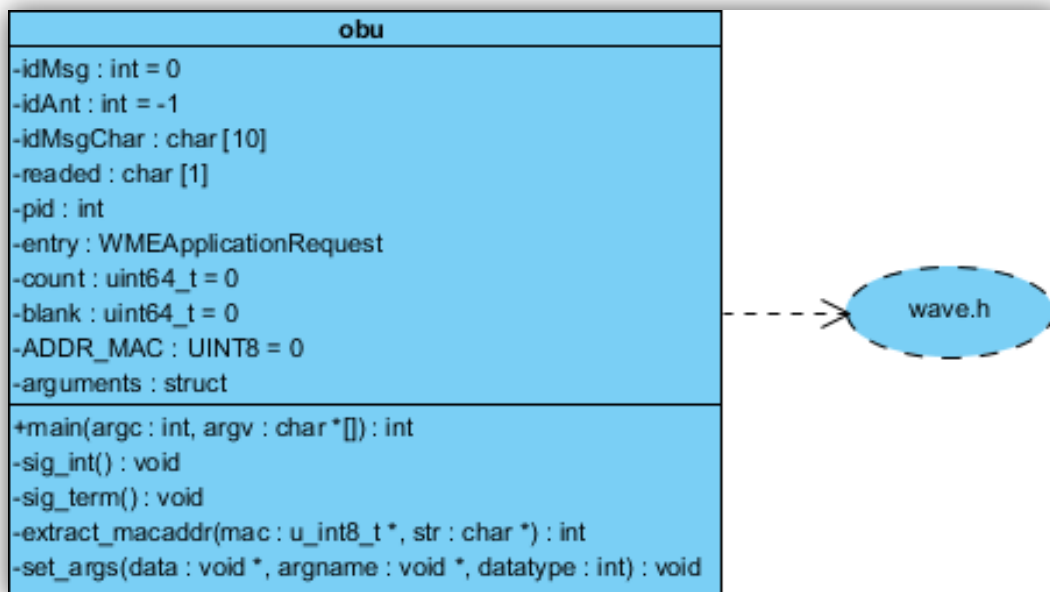


Ilustración 27. Diagrama de clases 8 - Recibir sanción

La estructura de esta clase es muy similar a la clase correspondiente al componente *Recibir acuse de recibo*. Como se ha mencionado anteriormente, el protocolo de recepción será el mismo, pero se tratarán los datos recibidos de forma distinta. En este caso, si el mensaje no ha sido tratado anteriormente, se recogerán los datos relevantes para el envío del acuse de

recibo. Estos datos serán aquellos datos identificativos de la sanción anteriormente enviada, como la matrícula del vehículo y la fecha y hora de la sanción.

En este caso, tras el tratamiento de los datos recibidos, la clase reconfigurará la OBU, de forma que se deje de recibir mensajes y se pueda realizar el envío del acuse de recibo de la sanción enviada. Esta parte se corresponde con el siguiente componente: *Enviar acuse de recibo*.

3.1.5.2. *Enviar acuse de recibo*

Este componente es el encargado de realizar el envío de los acuses de recibo de la OBU de aquellas sanciones enviadas por la RSU. Para poder realizar el envío tendremos que establecer unos parámetros iniciales relevantes de la misma forma que con el componente *Enviar sanción*, como por ejemplo el canal de envío, modo de utilización del canal, el proveedor de servicios que se utilizará, etc. Inicialmente, esta clase realizará una serie de funciones de inicialización que habilitarán el driver WAVE para el envío de mensajes DSRC.

Este componente, realiza los siguientes pasos con el fin de completar el envío de un acuse de recibo:

1. Registrar las funciones de envío mediante la librería *wave.h*.
2. Invoca el controlador.
3. Formular la petición de envío de mensaje.
 - 3.1. La petición de envío de mensajes se realizará múltiples veces para asegurar el envío del mensaje y la recepción por parte de la RSU.
4. Eliminar el registro cuando haya terminado el envío.

La Ilustración 28 muestra el diagrama de clases que representa este componente:

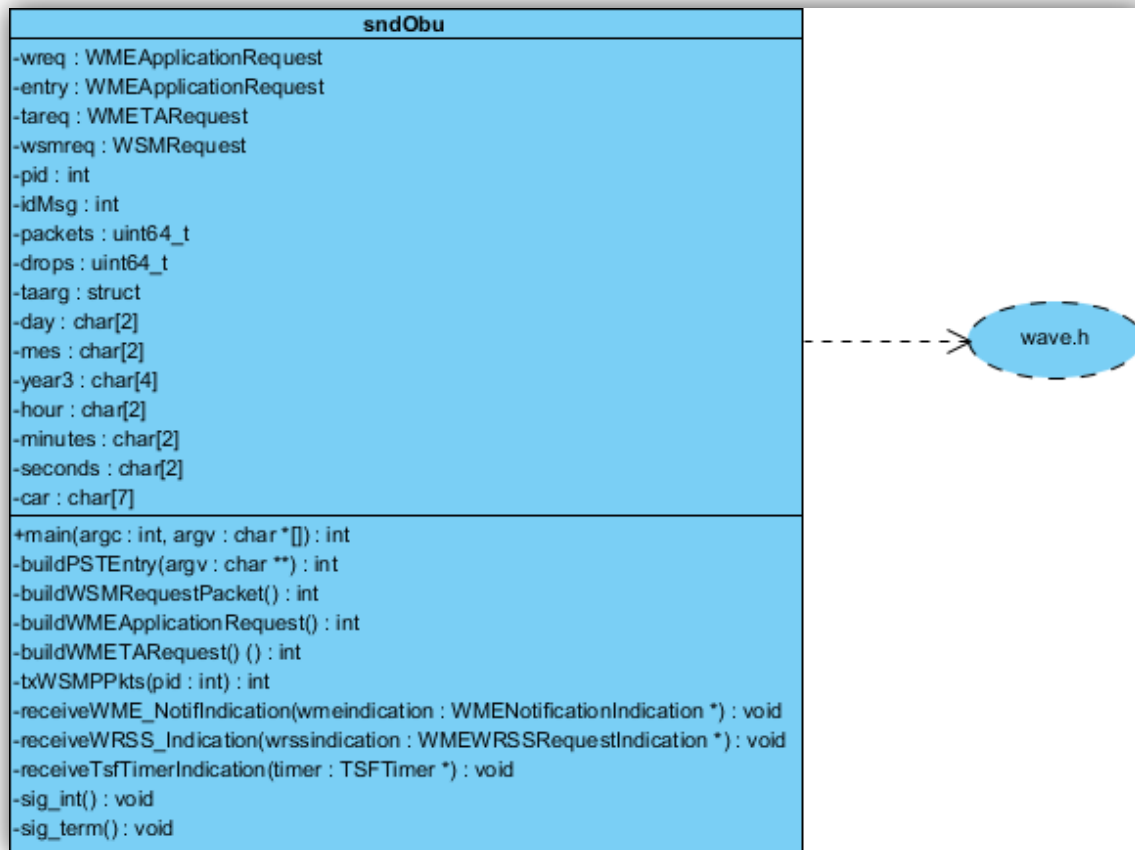


Ilustración 28. Diagrama de clases 9 - Enviar acuse de recibo

Como se ha mencionado anteriormente, la estructura de esta clase es muy similar a la clase correspondiente al componente *Enviar sanción*. El protocolo de envío será el mismo, pero, en este caso, se enviarán únicamente los datos relevantes para identificar tanto el mensaje enviado como la sanción impuesta.

La estructura de los mensajes que representan el acuse de recibo se define en el apartado [2.4.2.2 Requisitos funcionales de Información](#). También se puede ver la estructura del mensaje en el apartado [8.2. Anexo 2: Estructura fichero acuses de recibo](#) si únicamente nos fijamos en los datos correspondientes a la etiqueta `<offenceResponse>` y le añadimos los datos del identificador de mensaje.

3.2. Diagramas de secuencia

Este apartado está destinado a mostrar las distintas interacciones del usuario con el sistema, de forma que se dividirá en tantas secciones como acciones puede realizar el usuario, éstas acciones han sido definidas en el apartado [3.4 Establecimiento de Requisitos](#). Gracias a los diagramas de secuencia explicados a continuación también podremos apreciar los pasos que

deberá dar el usuario para llevar a cabo cada acción, así como qué ocurre dentro del sistema para que el caso de uso se complete con éxito.

Con el fin de aclarar y simplificar el siguiente apartado, se tendrán las siguientes consideraciones en cuenta:

- En los diagramas de secuencia, los métodos o constructores se representarán con un paréntesis a su derecha, mientras que si se desea pasar un atributo o parámetro no tendrá ningún paréntesis. Una línea discontinua representará un mensaje de retorno. En el caso de no tener ningún nombre la relación, que sea una línea discontinua y que vaya dirigida al usuario, se refiere a la devolución del control del sistema al usuario.
- Sólo se representarán los flujos principales de cada caso de uso. Los flujos alternativos no se tendrán en cuenta en los diagramas de secuencia.

3.2.1. Conectar RSU

Este apartado trata sobre el Caso de Uso 1: *Conectar RSU*. En la Ilustración 29 se muestra las acciones que el usuario debe realizar para completar el caso de uso, así como los pasos que se llevan a cabo desde las distintas clases del sistema (por ejemplo, la comunicación entre la RSU y la UI).

Se recuerda que la conexión entre la RSU y el host se lleva a cabo mediante un *TelnetClient*, por lo que la escritura del usuario y la contraseña de la RSU a la que realizamos la conexión se realizarán mediante esta librería, gracias al comando *write()*.

Una vez se haya realizado la conexión, se transferirá el *TelnetClient* entre las distintas ventanas de la interfaz para poder mantener la misma conexión en todas ellas.

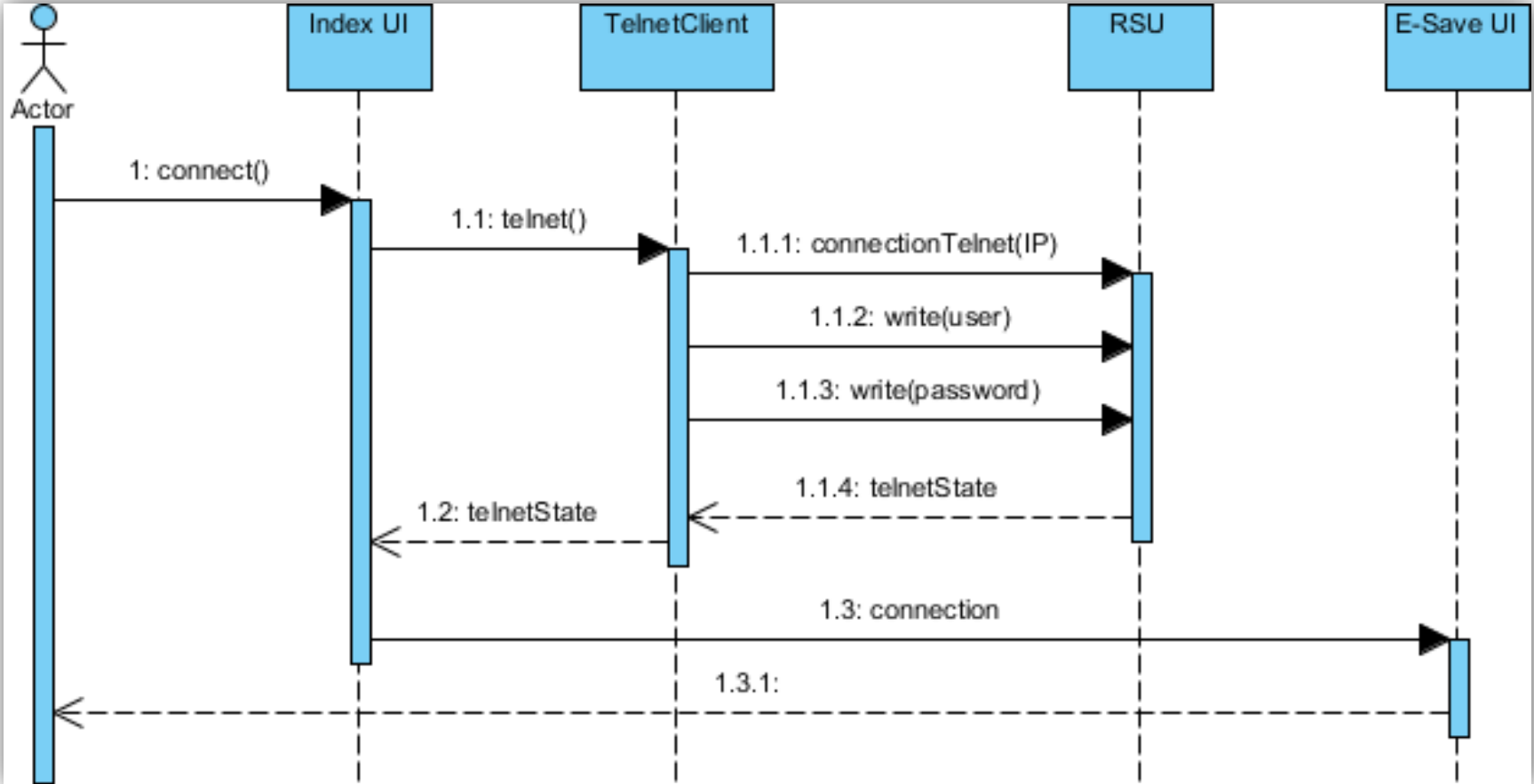


Ilustración 29. Diagrama de secuencia 1: Conectar RSU

3.2.2. Enviar sanción

Este apartado trata sobre el Caso de Uso 2: *Enviar sanción*. En la Ilustración 30 se muestra las acciones que el usuario debe realizar para completar el caso de uso, así como el funcionamiento interno de la aplicación.

Se supone que la conexión *TelnetClient* se mantiene con respecto al Caso de Uso 1, el usuario pulsará sobre la opción de enviar sanción. Aparecerá una ventana “resumen” donde el usuario tendrá que rellenar los datos de la sanción. A continuación se confirmará el envío pulsando sobre la opción de enviar. El sistema enviará el comando necesario para realizar el envío de la sanción.

La RSU se configurará para realizar el envío de la sanción según los parámetros establecidos en el comando enviado anteriormente. Tras la configuración de la RSU se procederá al envío de la sanción. El envío de la sanción se repetirá varias veces con el fin de garantizar tanto el envío como la recepción de la sanción por parte de la OBU.

Una vez enviada la sanción, la RSU borrará los anteriores registros que la habilitaban para el envío de mensajes DSRC. La RSU se reconfigurará de nuevo para habilitar la recepción de mensajes por parte de la OBU y permanecerá a la espera del acuse de recibo. Cuando la RSU reciba un acuse de recibo, éste tratará los datos recibidos y los registrará. Por otro lado, la Interfaz de Usuario registrará en la Base de Datos la sanción enviada.

Si nos ponemos en el lado de la OBU, se configurará al inicio de la aplicación para habilitar la recepción de mensajes DSRC. Se recuerda la suposición mencionada anteriormente en este documento, que supondría el arranque automático del programa que realiza la configuración de la OBU y la recepción de mensajes.

La OBU permanecería en espera de una sanción enviada por la RSU. Cuando reciba una sanción, se comprobará si se ha tratado dicho mensaje anteriormente, de no ser así se procederá a tratarlos. Cuando finalice el tratamiento de los datos recibidos, la OBU borrará los registros que la han configurado como receptora de mensajes y se reconfigurará para poder enviar mensajes a la RSU. Cuando la RSU sea capaz de enviar mensajes a la RSU, se procederá con el envío del acuse de recibo de la sanción anteriormente recibida. Con el fin de asegurar el envío y la recepción del acuse de recibo por parte de la RSU, el mensaje se enviará varias veces.

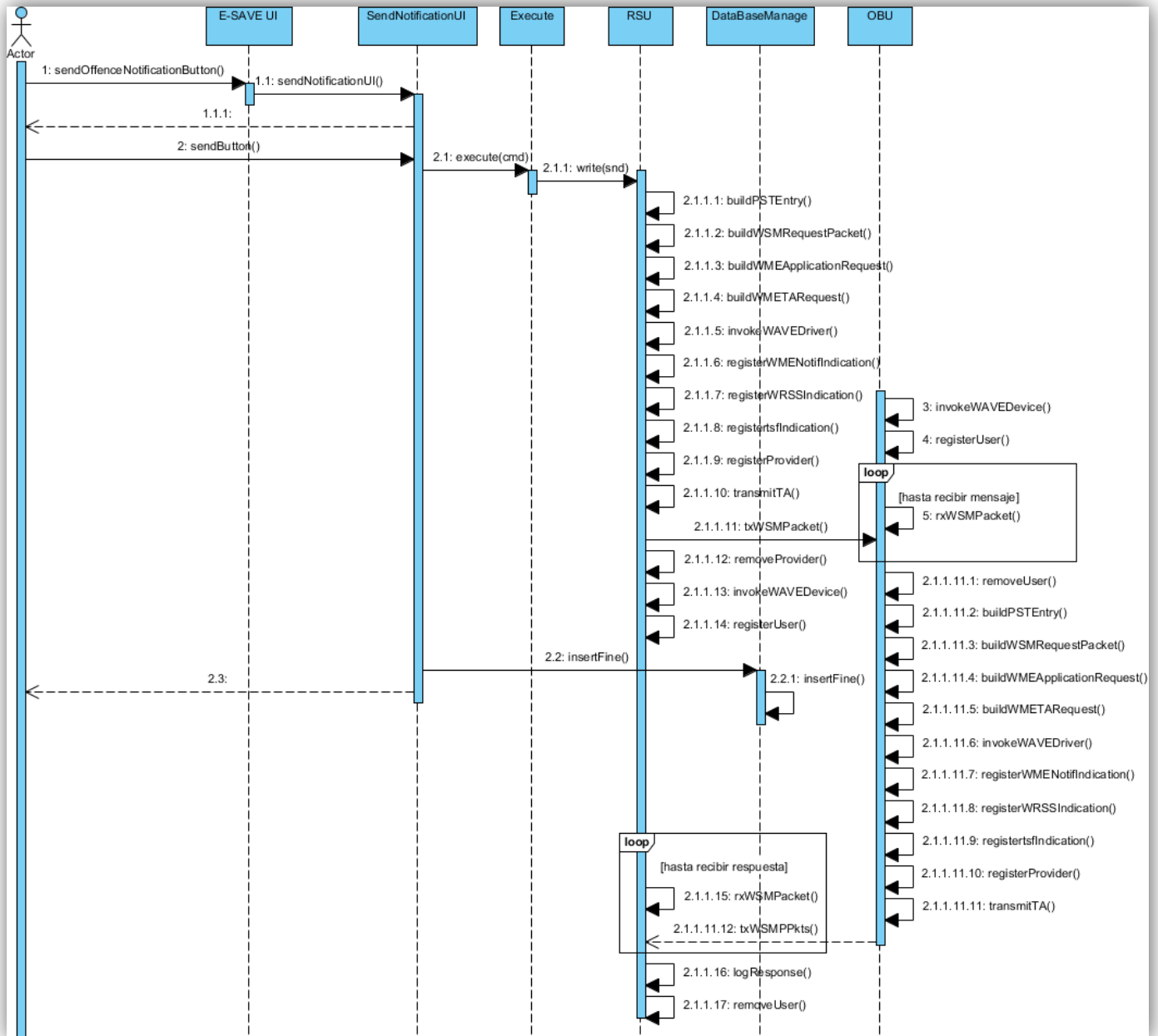


Ilustración 30. Diagrama de secuencia 2: Enviar sanción

3.2.3. Visualizar lista de sanciones

Este apartado trata sobre el Caso de Uso 3: *Visualizar sanciones*. En la Ilustración 31 se muestra las acciones que el usuario debe realizar para completar el caso de uso, así como el funcionamiento interno de la aplicación.

Se supone que la conexión *TelnetClient* se mantiene con respecto al Caso de Uso 1. Con el fin de mostrar toda la funcionalidad del sistema, este diagrama de secuencia englobará tanto la recogida de acuses de recibo registrados en la RSU, como la visualización de la información de las sanciones registradas.

La primera funcionalidad será la recogida de acuses de recibo registrados en la RSU. Para ello, el usuario pulsará sobre la opción para recoger los datos de la RSU. A continuación se mostrará al usuario una interfaz que muestre el progreso del tratamiento de los datos recibidos. El archivo de registro se importará desde la RSU al equipo host y se empezará con el tratamiento de los datos contenidos. Según se vayan procesando los distintos acuses de recibo se actualizará la Base de Datos para reflejar los distintos estados de recepción de las sanciones enviadas.

La segunda funcionalidad será la visualización de las últimas sanciones impuestas. Para ello el usuario pulsará sobre la opción para visualizar las últimas sanciones impuestas. A continuación, el sistema realizará una consulta a la Base de Datos de las últimas 10 sanciones registradas. Los datos recibidos de la Base de Datos se mostrarán mediante una tabla al usuario.

La Ilustración 31 estará dividida en dos secciones para diferenciar mejor cada funcionalidad explicada. Cada funcionalidad estará marcada con un recuadro azul y con una etiqueta que identifique a qué funcionalidad se refiere:

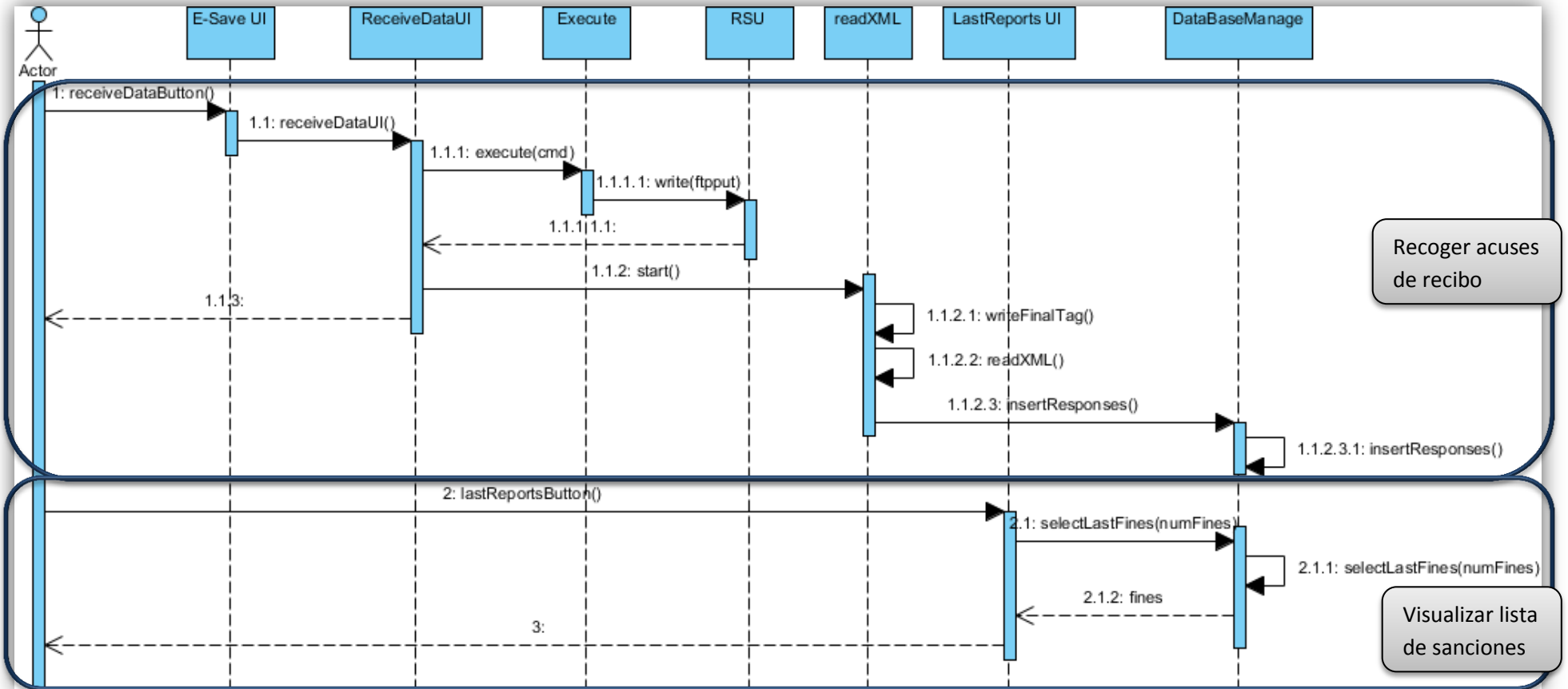


Ilustración 31. Diagrama de secuencia 3: Visualizar lista de sanciones

3.3. Diagrama relacional de la Base de Datos

En este apartado se explicará el diagrama relacional de la Base de Datos final utilizada para el desarrollo del sistema. Como se ha mencionado anteriormente en el apartado [2.5.3. Almacenamiento de Datos](#), la base de datos almacenará todos los datos enviados por la sanción. Se recuerda que estos datos a almacenar vienen impuestos por las necesidades de información requeridas en los envíos sanciones. La Base de Datos almacena toda la información de las sanciones enviadas en las tablas *locomatebbdd.fines* y *locomatebbdd.authority*.

Sin embargo, la Base de Datos no sólo almacenará los datos de las sanciones impuestas, sino que la aprovecharemos para aumentar la portabilidad del sistema. Para ello, como también se explicó anteriormente, se guardarán todos los textos de la interfaz de usuario: botones, mensajes, etc. La tabla que almacena todos los textos de la interfaz es *locomatebbdd.texts*.

También se aprovechará esta Base de Datos para almacenar RSU favoritas para el usuario, teniendo como supuesto que el sistema podría utilizarse en el futuro con varias RSU distintas. Por otro lado, se almacenará el último idioma seleccionado, así la dirección IP de la última RSU con la que se ha intentado realizar una conexión. Generalmente el idioma de una aplicación no suele cambiarse de forma habitual, por lo que es muy interesante el almacenamiento del idioma en el que se mostró por última vez. La tabla que almacena las RSU favoritas es *locomatebbdd.rsusfavorites*, mientras que la tabla que almacena las últimas opciones seleccionadas es *locomatebbdd.lastroptions*.

La Ilustración 32 muestra el diagrama relacional de la Base de Datos utilizada:

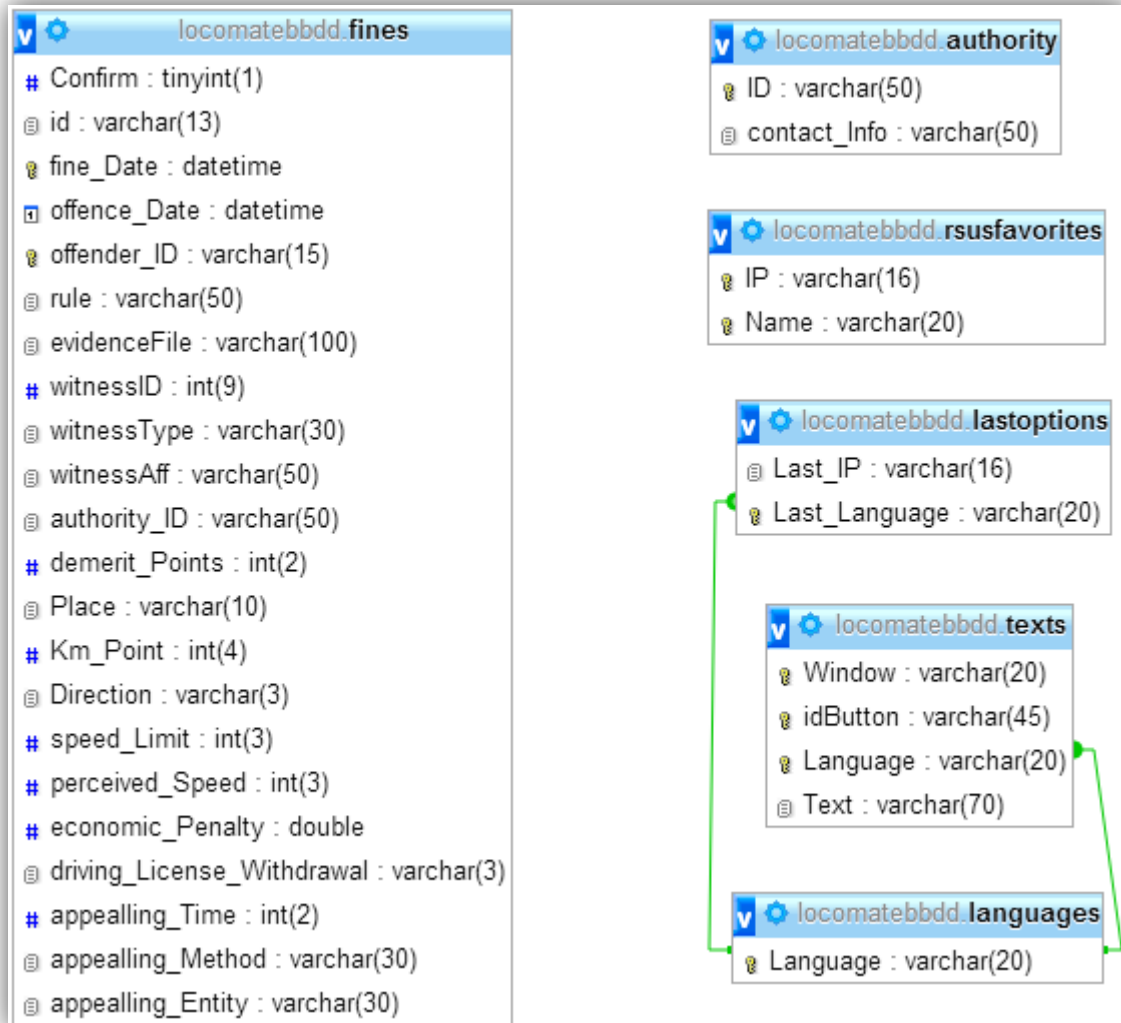


Ilustración 32. Diagrama relacional de la Base de Datos

En el diagrama relacional anterior no se representa una relación debido a algunas limitaciones del software utilizado para crear el diagrama. Sin embargo, existe una relación entre *authorityID* perteneciente a la tabla *locomatebbdd.fines* e *ID* perteneciente a la tabla *locomatebbdd.authority*. Esta relación no se ha podido añadir al modelo, pero se ha tenido en cuenta a la hora de la implementación de la interfaz, por el cual, un ID de una autoridad se refiere unívocamente con una información de contacto. Gracias a esta relación, la información de contacto de la autoridad permanecerá siempre actualizada en todas las sanciones a pesar de posibles cambios que puedan sufrir estos datos.

4. Implementación e Implantación del Software

En este capítulo se detallan las decisiones de implementación que se han llevado a cabo durante el desarrollo del sistema. También se mostrarán los resultados de las Pruebas de Aceptación propuestas anteriormente.

4.1. Procesos de Codificación y Pruebas

En este apartado se explican las dificultades que se han producido durante la implementación del sistema. También se mencionarán todas las decisiones que se han tomado durante la implementación del sistema desarrollado en este proyecto. Por último, se mostrarán los resultados de las pruebas de aceptación definidas en el punto [2.9.2. Definición del Plan de Pruebas](#).

4.1.1. Implementación y Pruebas del Software

En este apartado se detallan las decisiones de implementación tomadas durante el desarrollo del proyecto. Se explicarán las dificultades tecnológicas que se han encontrado con las nuevas tecnologías así como las soluciones adoptadas para abordar dichos problemas.

Como se ha mencionado anteriormente, el fabricante permite la conexión a los dispositivos LOCOMATE únicamente mediante la utilización del protocolo Telnet. Para establecer una conexión entre la Interfaz de Usuario y los dispositivos LOCOMATE, primeramente se pensó en la ejecución de comandos independientes, como haría un usuario en una ventana de comandos. Sin embargo, se observaron numerosas dificultades para establecer y mantener la conexión con los dispositivos mediante este proceso.

Para solucionar este problema se buscó una librería especializada en la conexión de este protocolo. La solución encontrada fue la librería *commons-net.jar* versión 3.3, que facilitaba una serie de características propias de este protocolo de conexión, así como de una serie de funciones útiles. Esta librería permitía el establecimiento de una conexión permanente con un dispositivo LOCOMATE, comprobación de conexión, cierre seguro de la conexión, etc.

Otra dificultad encontrada a la hora de la implementación fue la búsqueda de una carpeta que mantuviera los datos cuando el dispositivo se reiniciase. Como se ha mencionado anteriormente, los dispositivos LOCOMATE utilizan una versión limitada de Linux, de forma que un desarrollador no puede administrar el dispositivo en su totalidad. La mayoría de las carpetas del dispositivo eran únicamente de lectura, y no permitían la escritura ni la modificación de archivos. Aquellas carpetas que permitían la escritura eran borradas tras el reinicio del dispositivo. Inicialmente se usó la carpeta temporal `\tmp` para transferir

ejecutables de prueba y ficheros de registro. Sin embargo, esta ubicación borra y restablece todos los ficheros tras el apagado del dispositivo.

Para poder eludir este problema se tuvo que realizar la consulta al propio fabricante para encontrar una zona de lectura-escritura que mantuviera todos los ficheros tras el reinicio del dispositivo. Finalmente la carpeta habilitada para nuestro propósito fue `\var`, donde el desarrollador tenía libertad para asignar permisos, crear archivos, etc. así como de mantener todos los cambios realizados tras el apagado del dispositivo.

A pesar de la gran documentación enviada por el fabricante, los códigos de ejemplo que se han proporcionado no contienen una explicación demasiado detallada de los parámetros que se pueden utilizar, así como la finalidad que tiene cada uno de ellos. En algunos casos, la funcionalidad de los parámetros es fácil e intuitiva, como el canal de envío/recepción. Sin embargo, existen otros parámetros que no se detalla mucho el uso que se les da, como puede ser el identificador de proveedor/usuario.

Este problema nos lleva también a la gran complejidad que tiene el manejo del envío y la recepción de datos por parte de los dispositivos. Se recuerda que los dispositivos no pueden realizar ambas funciones simultáneamente, por lo que se debe controlar y sincronizar los envíos y recepciones para que se realicen de forma inversa en cada dispositivo. Es decir, cuando la RSU esté realizando un envío, la OBU deberá permanecer escuchando y viceversa.

Debido a las limitaciones del proyecto de equipamiento, y por tanto, del proyecto, sólo se realiza el envío de una sanción. Debido a esta limitación se ha decidido realizar una "limpieza" de las opciones que definen el dispositivo como emisor o receptor de mensajes tras la realización de una de estas acciones. Es decir, tras el envío de una sanción, la RSU limpiará las opciones que la definen como emisora, de forma que se pueda configurar como receptora de mensajes. En el caso de la OBU se registraría desde un primer momento como receptora, pero al recibir una sanción se borrarían las opciones necesarias y se registraría como emisora para poder enviar el acuse de recibo.

El manejo de estas opciones de configuración es muy delicado, de no borrar alguna de las opciones, el dispositivo quedaría en un estado "intermedio" y no se podrían realizar envíos/recepciones hasta su reinicio. Esto supondría un grave problema, puesto que el reinicio de un dispositivo de este tipo supondría un tiempo muy valioso en el que no se podrían transmitir ni recibir datos de ningún tipo.

Volviendo al problema del manejo de los envíos de sanciones y la recepción de los mensajes, se ha de mencionar la dificultad que se encontró a la hora de asegurar la comunicación entre los dispositivos. Esto quiere decir que en un principio, las comunicaciones establecidas entre la OBU y la RSU perdían más del 50% de los mensajes debido al mal control de los parámetros que realizaban la configuración de los dispositivos.

En un principio se utilizó como base para el aprendizaje un ejecutable de ejemplo que proporcionaba el fabricante donde se podían probar todas las funcionalidades de los dispositivos. La elección de este programa para su modificación era clara, si se conseguía

realizar la comunicación mediante este ejecutable, la certificación de mensajes, así como posteriores mejoras, se realizaría de forma más fácil utilizando este mismo código.

Sin embargo, este ejecutable de ejemplo contenía muchísimas funciones que no se podían modificar sin afectar a las opciones de los dispositivos, por lo que se producía el problema anteriormente mencionado con el envío y la recepción de los mensajes.

Para solucionar este problema, se partió de nuevo con otro de los programas de ejemplo, aunque esta vez estaba mucho más limitado, de forma que sólo demostraba el envío simple de mensajes. A partir de la experiencia obtenida en el ejecutable anterior, se pudo realizar una simulación de los envíos de sanciones configurando los dispositivos de forma correcta. La utilización de este nuevo programa no sólo facilitó la codificación, sino que, al no ser un sistema tan complejo ni funcional, la configuración de los dispositivos sólo cambiaba cuando lo deseaba el desarrollador. Esto garantizó el envío y recepción de los mensajes en un porcentaje superior al 95%.

Por último, se ha de mencionar que el desarrollador del sistema realizaba pruebas funcionales semanales sobre los distintos módulos analizados e implementados durante ese período. El desarrollador comprobaba el funcionamiento de cada uno de los módulos detectando posibles anomalías en el comportamiento del sistema así como comprobando el correcto funcionamiento del mismo.

4.2.Resultados de las Pruebas de Aceptación

En este apartado se muestran los distintos resultados tras la realización de las pruebas definidas en el punto [2.9.2. Definición del Plan de Pruebas](#) con el fin de garantizar que el sistema cumple con todos los requisitos propuestos y se llega a una aplicación de calidad.

Las pruebas podrán tener dos tipos de resultados:

- **Superada:** Se han realizado todos los pasos descritos en la prueba y el resultado obtenido es el que se indica en el apartado de resultados.
- **No superada:** No se han podido realizar todos los pasos descritos en la prueba o el resultado obtenido tras completar la prueba no ha sido el esperado.

Identificador	Resultado
PRU-A01	<input checked="" type="checkbox"/> Superada <input type="checkbox"/> No superada
PRU-A02	<input checked="" type="checkbox"/> Superada <input type="checkbox"/> No superada
PRU-A03	<input checked="" type="checkbox"/> Superada <input type="checkbox"/> No superada
PRU-A04	<input checked="" type="checkbox"/> Superada <input type="checkbox"/> No superada
PRU-A05	<input checked="" type="checkbox"/> Superada <input type="checkbox"/> No superada
PRU-A06	<input checked="" type="checkbox"/> Superada <input type="checkbox"/> No superada
PRU-A07	<input checked="" type="checkbox"/> Superada <input type="checkbox"/> No superada
PRU-A08	<input checked="" type="checkbox"/> Superada <input type="checkbox"/> No superada
PRU-A09	<input checked="" type="checkbox"/> Superada <input type="checkbox"/> No superada

Identificador	Resultado	
PRU-A10	<input checked="" type="checkbox"/> Superada	<input type="checkbox"/> No superada
PRU-A11	<input checked="" type="checkbox"/> Superada	<input type="checkbox"/> No superada
PRU-A12	<input checked="" type="checkbox"/> Superada	<input type="checkbox"/> No superada
PRU-A13	<input checked="" type="checkbox"/> Superada	<input type="checkbox"/> No superada
PRU-A14	<input checked="" type="checkbox"/> Superada	<input type="checkbox"/> No superada
PRU-A15	<input checked="" type="checkbox"/> Superada	<input type="checkbox"/> No superada
PRU-A16	<input checked="" type="checkbox"/> Superada	<input type="checkbox"/> No superada
PRU-A17	<input checked="" type="checkbox"/> Superada	<input type="checkbox"/> No superada

Tabla 13: Resultados Pruebas de Aceptación

Como hemos podido ver en la Tabla 13, todas las pruebas han resultado satisfactorias.

5. Gestión del Proyecto

En este capítulo se explican todos los detalles de la planificación y gestión del proyecto durante su realización, comparando la estimación del esfuerzo del proyecto con el esfuerzo final y real realizado. También se exponen y explican las herramientas utilizadas durante el desarrollo del proyecto, así como la elección de cada una. Por otro lado, se detallará y compararán los costes estimados para la realización del proyecto con el coste real que ha supuesto su desarrollo.

5.1. Medios técnicos empleados

Toda la documentación realizada durante el proceso de desarrollo del sistema ha sido realizado con la herramienta de ofimática *Microsoft Office 2010* [23], de la cual se hará uso del procesador de textos *Word 2010* para la realización de documentación escrita, la hoja de cálculo *Excel 2010* para la realización del documento de cálculo de costes, así como la herramienta de presentaciones *PowerPoint 2010*.

Se han utilizado otra serie de herramientas que servirán como apoyo a la hora del desarrollo del proyecto, como *Adobe Reader 10.1.6* [31] para leer artículos o proyectos anteriores, *Chrome* versión 26.0.1410.64 [32] para la búsqueda de información en Internet. Para poder utilizar todas las aplicaciones que requieran Java, se ha descargado la tecnología Java versión 1.6.0_22-b04 [33].

También se han utilizado herramientas específicas para la realización de algunas tareas concretas, como *Pencil* versión 2.0.3 [34] para la realización de interfaces prototipo, la herramienta de planificación *Microsoft Project 2010* [35] para los distintos diagramas de Gantt realizados y *Visual Paradigm for UML Enterprise Edition* versión 9.0 [36] para la realización de los diagramas de casos de uso y los diagramas de secuencia.

Con el fin de mantener una copia de seguridad de todo el desarrollo del proyecto se ha utilizado la herramienta *Dropbox* versión 2.1.18 [19]. Esta herramienta servirá también para la comunicación entre la máquina virtual creada y el equipo host, gracias a la sincronización de carpetas característico de esta aplicación. Por último, aprovecharemos el control de versiones que se puede encontrar en su sitio web.

Por otra parte para la comunicación entre las personas que forman el equipo de proyecto se ha utilizado *Dropbox* que facilita la compartición de distintos archivos y el contacto por correo electrónico a través de *GMail* [37].

5.2. Planificación del Proyecto

Este apartado está destinado a mostrar la gestión y planificación del proyecto. Para ello, se realizará una comparativa del esfuerzo estimado del proyecto con el esfuerzo real del proyecto, donde se estudiará las discrepancias que puedan haber surgido durante el desarrollo del proyecto.

5.2.1. Planificación Inicial

En este apartado se mostrarán y se comentará la planificación que se realizó al comienzo del proyecto, de forma que se vea de forma clara y concisa la organización estimada del proyecto, desde que comenzó hasta la fecha en la que se pretende terminar. Se podrán apreciar las distintas partes en las que se ha dividido el proyecto, así como el tiempo asignado según la dificultad estimada de cada una de ellas.

La fecha de comienzo del proyecto fue el 4 de Febrero de 2013, mientras que la fecha estimada de entrega es el 4 de Junio de 2013, es decir, una duración de 4 meses. La estimación del esfuerzo medio necesario para la realización del proyecto es de 15 horas semanales de lunes a viernes, excepto la última semana que serán 9 horas.

Se ha decidido este esfuerzo debido a la compaginación de este proyecto junto con la realización de prácticas en empresa, así como la finalización de los estudios restantes del alumno. En caso de que el alumno haya trabajado en el presente proyecto durante el fin de semana (sábado o domingo), las horas realizadas se pasarán al día laboral más próximo, es decir, si se han trabajado 2 horas durante el sábado, se sumarán con las horas realizadas el viernes, del mismo modo que las horas trabajadas el domingo se pasarán al lunes.

Durante la estimación de la planificación inicial se llevó a cabo una estimación sobre las distintas tareas que englobaría el proyecto. Estas tareas se explican a continuación en la Tabla 14:

Resumen	Tareas
Análisis	<p><u>Análisis de tecnologías</u>: Esta tarea se corresponde con el apartado 2.5. Estudio tecnológico. Este apartado trata sobre las diferentes tecnologías que se usarán durante el desarrollo del proyecto.</p> <p><u>Codificar envío y respuesta</u>: Esta tarea se corresponde con el análisis de las comunicaciones DSRC. Tras esto, se realizarán distintas pruebas con los códigos de ejemplos proporcionados por el fabricante. Por último, se comenzará la codificación de pequeños programas que realicen envíos de mensajes simples de la RSU a la OBU.</p> <p><u>Interfaz básica</u>: Esta tarea se corresponde con el estudio de las necesidades del usuario sobre la interfaz para realizar todas las acciones propuestas para el proyecto. Resultará en una representación muy básica de la interfaz que se desarrollará a partir de esta tarea.</p> <p><u>Envío de mensajes compuestos</u>: Esta tarea se realizará a continuación del estudio de mensajes simples. Una vez establecida la comunicación básica entre la RSU y la OBU se procederá al envío de mensajes con mayor complejidad en los datos enviados.</p>
Desarrollo	<p><u>Seguridad y firmas</u>: Se realizará un estudio de la seguridad que se puede implantar en los envíos de mensajes entre RSU y OBU. Se realizará la codificación de la seguridad y certificación necesaria para garantizar la identidad de una OBU sin comprometer su privacidad.</p> <p><u>Interfaz final</u>: Esta tarea se realizará tras la interfaz básica. Se corresponde con la implementación de las funciones que permitan al usuario utilizar la interfaz para establecer una conexión con la RSU y enviar una sanción a la OBU.</p>
Documentación	<p><u>Pruebas</u>: Esta tarea se corresponde con la ejecución de las pruebas que verificaran el correcto funcionamiento del sistema.</p> <p><u>Documentación</u>: Esta tarea se corresponde con la realización de la documentación del proyecto.</p>

Tabla 14. Tareas planificadas iniciales

A continuación, en la Tabla 15 podemos ver la duración total del proyecto, así como la de cada tarea según la primera estimación del proyecto:

Nombre de tarea	Duración	Comienzo	Fin
locomate_project	87 días	lun 04/02/13	mar 04/06/13
1. Análisis	27 días	lun 04/02/13	mié 13/03/13
1.1. Análisis de tecnologías	7 días	lun 04/02/13	mar 12/02/13
1.2. Revisión	0 días	mié 13/02/13	mié 13/02/13
1.3. Codificar envío y respuesta	11 días	mié 13/02/13	mié 27/02/13
1.4. Revisión	0 días	jue 28/02/13	jue 28/02/13
1.5. Interfaz básica	9 días	jue 28/02/13	mar 12/03/13
1.6. Revisión	0 días	mié 13/03/13	mié 13/03/13
2. Diseño	66 días	vie 01/03/13	vie 31/05/13
2.1. Envío mensajes compuestos	30 días	vie 01/03/13	jue 11/04/13
2.2. Pruebas	60 días	lun 11/03/13	vie 31/05/13
2.3. Seguridad y firmas	15 días	jue 14/03/13	mié 03/04/13
2.4. Revisión	0 días	mié 03/04/13	mié 03/04/13
2.5. Interfaz final	57 días	jue 14/03/13	vie 31/05/13
2.6. Revisión	0 días	vie 12/04/13	vie 12/04/13
3. Documentación	38 días	vie 12/04/13	mar 04/06/13
3.1. Revisión	0 días	vie 10/05/13	vie 10/05/13
3.2. Documentación	38 días	vie 12/04/13	mar 04/06/13
3.3. Revisión	0 días	vie 31/05/13	vie 31/05/13

Tabla 15. Planificación inicial

Como podemos ver, el proyecto se ha dividido en tres grandes tareas: Análisis, Diseño y Documentación. Se ha decidido que la codificación se realice al mismo tiempo que las demás tareas en vez de separarlo en otra distinta, puesto que al tratarse de nueva tecnología se necesita un esfuerzo extra en este apartado, por tanto estará presente en casi todo el proyecto.

Durante el análisis, se estudiará la tecnología que se va a utilizar, tanto la RSU y la OBU, como las herramientas de apoyo necesarias para el proyecto. Al mismo tiempo que se lleve el estudio de la tecnología del fabricante (sus características y programas de ejemplo), se realizarán pruebas de dicha tecnología para comprobar y verificar su funcionamiento.

A continuación, tras saber cómo funciona la tecnología y qué limitaciones tiene, pasaremos al desarrollo del proyecto, donde se procederá a la codificación propiamente dicha de la aplicación que se pretende desarrollar.

Una vez se ha terminado el desarrollo de las funcionalidades requeridas, se procederá a su comprobación mediante las pruebas, así como la documentación de todo el desarrollo del proyecto. Además, se terminará y se refinará la Interfaz de Usuario según las necesidades del usuario final.

En la Ilustración 33, se puede ver de forma gráfica la planificación inicial explicada anteriormente.

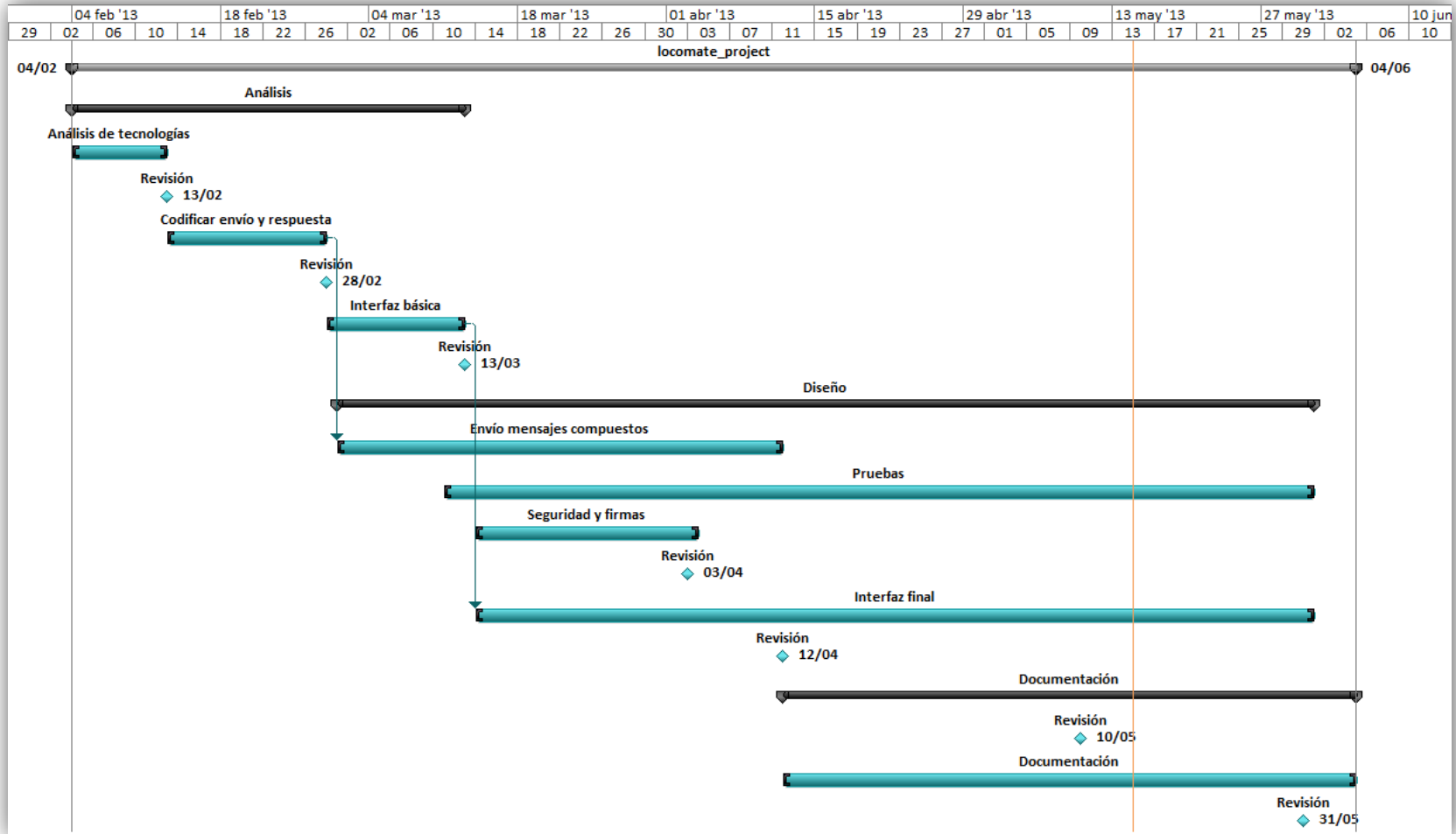


Ilustración 33. Planificación inicial

5.2.2. Desarrollo Real del Proyecto

En este apartado se mostrarán y se comentará el desarrollo real del proyecto. Se mostrará de forma textual y gráfica la organización real del proyecto, desde la fecha en que comenzó hasta la fecha de entrega. Se podrán apreciar las distintas partes en las que se ha dividido el proyecto, así como el tiempo asignado según la dificultad estimada de cada una de ellas.

Se podrán apreciar las distintas partes en las que se ha dividido el proyecto, así como el tiempo asignado según la dificultad estimada de cada una de ellas.

La fecha de comienzo del proyecto fue el 4 de Febrero de 2013, mientras que la fecha final de entrega es el 19 de Junio de 2013, es decir, una duración de 4 meses y medio. El esfuerzo medio realizado finalmente es de 16,1 horas semanales.

Durante el desarrollo del proyecto surgieron problemas no contemplados que requirieron un cambio en la planificación y reorganización de las tareas. Estas tareas se explican a continuación en la Tabla 16:

Resumen	Tareas
Análisis	<p><u>Análisis de tecnologías</u>: Esta tarea se corresponde con el apartado 2.5. Estudio tecnológico. Este apartado trata sobre las diferentes tecnologías que se usarán durante el desarrollo del proyecto.</p> <p><u>Codificar envío y respuesta</u>: Esta tarea se corresponde con el análisis de las comunicaciones DSRC. Tras esto, se realizarán distintas pruebas con los códigos de ejemplos proporcionados por el fabricante. Por último, se comenzará la codificación de pequeños programas que realicen envíos de mensajes simples de la RSU a la OBU y de la OBU a la RSU.</p> <p><u>Interfaz básica</u>: Esta tarea se corresponde con el estudio de las necesidades del usuario sobre la interfaz para realizar todas las acciones propuestas para el proyecto. Resultará en una representación muy básica de la interfaz que se desarrollará a partir de esta tarea.</p>
Desarrollo	<p><u>Envío de mensajes compuestos</u>: Esta tarea se realizará a continuación del estudio de mensajes simples. Una vez establecida la comunicación básica entre la RSU y la OBU se procederá al envío de mensajes con mayor complejidad en los datos enviados. Se tendrá en cuenta que la OBU debe responder a las sanciones enviadas desde la RSU.</p> <p><u>Interfaz final</u>: Esta tarea se realizará tras la interfaz básica. Se corresponde con la implementación de las funciones que permitan al usuario utilizar la interfaz para establecer una conexión con la RSU y enviar una sanción a la OBU.</p> <p><u>Registro de acuses de recibo</u>: Esta tarea se encarga de registrar todos</p>

Resumen	Tareas
Documentación	<p>los acuses de recibo que ha enviado la OBU.</p> <p><u>Pruebas</u>: Esta tarea se corresponde con la ejecución de las pruebas que verificarán el correcto funcionamiento del sistema.</p> <p><u>Documentación</u>: Esta tarea se corresponde con la realización de la documentación del proyecto.</p>

Tabla 16. Tareas reales

Como se puede ver en la tabla anterior, algunas tareas han quedado fuera del alcance del proyecto, han aparecido otras nuevas, y antiguas tareas abarcan mayor funcionalidad. Esto ha ocurrido debido a las dificultades encontradas para poder trabajar en grupo como se tenía previsto por el equipo de proyecto. Estas complicaciones surgieron en parte debido a la diferencia de horario de los dos alumnos con las asignaturas pendientes.

Para solucionar las dificultades que surgieron por las complicaciones de horario, el tutor realizó una reorganización de la estructura del proyecto, de forma que todo el protocolo de envío de sanciones pertenecería a este proyecto, mientras que la parte del aseguramiento de los mensajes intercambiados y de las comunicaciones subyacentes será para el proyecto que complementa al actual.

En resumen, el proyecto inicial se encargaría de realizar el protocolo de envío de sanciones a la OBU, certificar los mensajes enviados y recibir los acuses de recibo enviados desde la OBU. A partir de esta nueva organización, el proyecto final ha surgido con el objetivo de realizar el protocolo de envío de sanciones desde la RSU y respuesta mediante acuses de recibo desde la OBU.

A continuación, en la Tabla 17 podemos ver la duración final del proyecto, así como la de cada tarea:

Nombre de tarea	Duración	Comienzo	Fin
locomate_real	96 días	mié 06/02/13	mié 19/06/13
1. Análisis	43 días	mié 06/02/13	vie 05/04/13
1.1. Análisis de tecnologías	8 días	mié 06/02/13	vie 15/02/13
1.2. Revisión	0 días	mié 13/02/13	mié 13/02/13
1.3. Codificar envío y respuesta	35 días	lun 18/02/13	vie 05/04/13
1.4. Revisión	0 días	jue 28/02/13	jue 28/02/13
1.5. Interfaz básica	19 días	jue 07/03/13	mar 02/04/13
1.6. Revisión	0 días	mié 03/04/13	mié 03/04/13
2. Diseño	51 días	mié 03/04/13	mié 12/06/13
2.1. Envío de mensajes compuestos	35 días	lun 08/04/13	vie 24/05/13
2.2. Revisión	0 días	vie 12/04/13	vie 12/04/13
2.3. Interfaz final	40 días	mié 03/04/13	mar 28/05/13
2.4. Revisión	0 días	mié 24/04/13	mié 24/04/13
2.5. Revisión	0 días	vie 10/05/13	vie 10/05/13
2.6. Registro de acuses de recibo	4 días	jue 23/05/13	mar 28/05/13
2.7. Revisión	0 días	vie 24/05/13	vie 24/05/13
2.8. Pruebas	28 días	lun 06/05/13	mié 12/06/13
2.9. Revisión	0 días	vie 07/06/13	vie 07/06/13
3. Documentación	36 días	mié 01/05/13	mié 19/06/13
3.1. Documentación	36 días	mié 01/05/13	mié 19/06/13
3.2. Revisión	0 días	vie 14/06/13	vie 14/06/13
3.3. Revisión	0 días	mié 19/06/13	mié 19/06/13

Tabla 17. Desarrollo real del proyecto

En la Ilustración 34, se puede ver de forma gráfica el desarrollo real del proyecto explicado anteriormente.

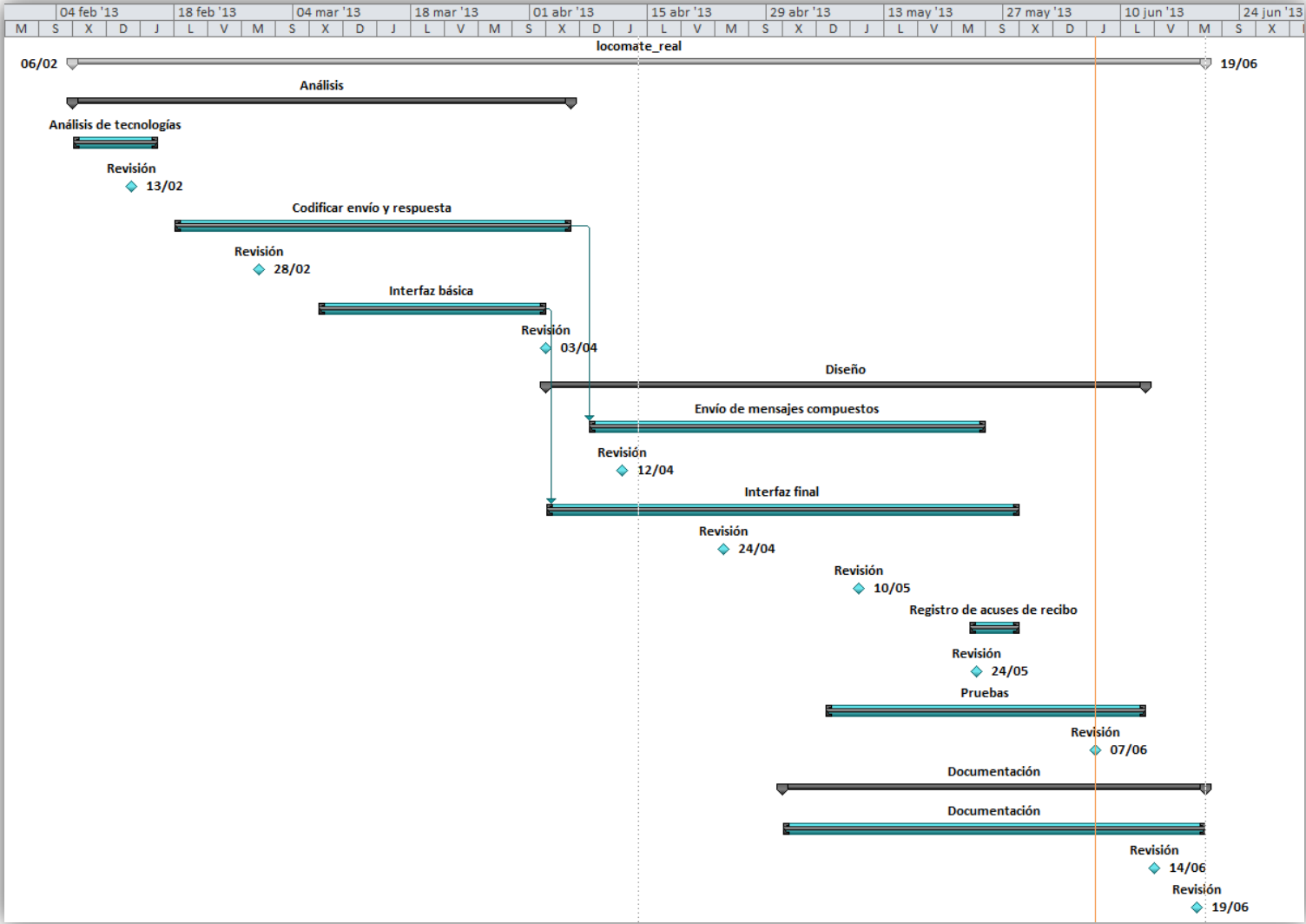


Ilustración 34. Desarrollo real del proyecto

5.2.3. Planificación Inicial vs Desarrollo Real

En este apartado se pretende comparar la planificación inicial estimada del proyecto con el desarrollo real que se ha llevado a cabo. Con esto se realizará un estudio de las diferencias encontradas entre ambos. Gracias a este estudio se podrá concluir las razones por las que han ocurrido discrepancias entre ambos, de forma que se pueda evitar en los futuros proyectos que se lleven a cabo.

Si nos fijamos en las gráficas y tablas de los apartados anteriores, veremos que la duración del proyecto ha sido superior a la estimada. La desviación producida entre la planificación inicial y el desarrollo del proyecto se debe al enfrentamiento del alumno con nuevas tecnologías que todavía permanecen en fase de pruebas.

También se ha de mencionar la dificultad de reorganización del proyecto, como se ha explicado anteriormente. Por otro lado, algunas tareas que se estimaban fáciles, como el envío de mensajes simples, requirieron un esfuerzo adicional que no se contemplaba en el proyecto.

Algunos retrasos se deben a la falta de disponibilidad del alumno para continuar con el desarrollo del proyecto durante algunos días. Por último, se presentó una gran dificultad a la hora de realizar la codificación de envío de mensajes, puesto que se debía estar en el mismo lugar en el que se encontraran los dispositivos LOCOMATE.

Todos estos retrasos llevaron a la desviación de un 16,3% de la duración del proyecto, que equivale a 15 días más de lo esperado.

5.3. Análisis económico del proyecto

En este apartado se explicará el análisis económico del proyecto: el presupuesto inicial del proyecto planificado, el presupuesto que se entregaría al cliente, así como la comparación entre los costes que han sido estimados en el presupuesto inicial con los costes reales que se han llevado a cabo durante la realización del proyecto.

5.3.1. Metodología de estimación de costes

Se calcularán los gastos del personal como si tratara de una empresa privada que realiza un estudio y posterior desarrollo de una aplicación solicitada por un cliente.

Los costes se dividirán en los siguientes apartados:

- **Costes de personal:** Todos aquellos costes relacionados con los salarios de los distintos ingenieros (o roles del alumno) que han intervenido para el desarrollo del sistema.

- **Costes de software:** Costes relacionados con la adquisición de programas que son necesarios para el desarrollo del proyecto. Se tendrán en cuenta las distintas vidas útiles de los programas comprados.
- **Costes de equipamiento:** Costes relacionados con los distintos dispositivos hardware que han sido necesarios para el desarrollo del proyecto.
- **Costes indirectos:** Costes relacionados consumibles, gastos de oficina y gestión, etc.

5.3.2. Presupuesto inicial

En este apartado se muestra el presupuesto inicial estimado al principio del proyecto. Este presupuesto inicial derivará en el presupuesto que se le presentará al cliente.

No se incluirá el IVA en ninguno de los costes presentados a continuación. Se ha de mencionar que debido a la antigüedad de algunos equipos, el IVA aplicable sobre ellos es inferior, pasando a ser el 18% en vez del 21% que se encuentra actualmente. Por último, se ha de señalar que la duración estimada del proyecto será de cuatro meses, por lo que el coste se referirá a los cuatro meses de trabajo esperado.

5.3.2.1. Costes de personal

Este apartado detalla los gastos derivados del personal que haya trabajado en el desarrollo del proyecto. Para realizar el cálculo de los costes de esta sección, se han tenido en cuenta el trabajo de un Graduado en Ingeniería de Informática como ingeniero desarrollador, así como el trabajo de un Jefe de Proyecto, que supervisa el correcto desarrollo del proyecto y ofrece ayuda y soporte en caso de que el desarrollador la necesite.

La tabla 18 muestra detalladamente los costes del personal que ha intervenido durante el proyecto:

Concepto	Coste/hora	Coste
Jefe de proyecto	50 €	1.980,00 €
Analista	30 €	3.168,00 €
Desarrollador	30 €	1.980,00 €
Especialista de pruebas	30 €	1.584,00 €
SUMA	-	8.712,00 €
IRPF+SS (10%)	-	871,20 €
TOTAL	-	9.583,20 €

Tabla 18. Costes de personal

Esto hace un coste total bruto de 8.712,00 € (Ocho mil setecientos doce euros). A este coste habría que sumarle el importe correspondiente a IRPF y Seguridad Social, que se correspondería con la cantidad de 871,20 € (Ochocientos setenta y un euros con veinte céntimos). Sumando ambos importes tendríamos un coste total neto de **9.583,20 € (Nueve mil quinientos ochenta y tres euros con veinte céntimos)**.

5.3.2.2. Costes de software

Este apartado detalla los gastos derivados al software necesario que se ha tenido que comprar o utilizar para el correcto desarrollo del proyecto.

La Tabla 19 muestra detalladamente los costes de cada uno de los programas utilizados durante el desarrollo del proyecto así como la vida útil de cada uno de ellos. La vida útil servirá para determinar cuándo un programa se vuelve obsoleto:

Concepto	Coste total	Vida útil (meses)	Coste imputable
Máquina Virtual (VMWARE)	457,44 €	36	19,06 €
NetBeans	0,00 €	-	0,00 €
Microsoft Project	592,5 €	36	24,69 €
Microsoft Office 2010 Professional	595,00 €	36	24,79 €
Visual Paradigm	552,21 €	36	23,01 €
Dropbox	138,00 €	12	13,63 €
Donación software libre	70,00 €	-	70,00 €
TOTAL	-	-	175,18 €

Tabla 19. Costes de software

Se ha decidido realizar una donación al software libre, entre los que se encuentra el programa NetBeans, con el fin de contribuir al desarrollo y mejora de este tipo de software.

Todos los costes representados en la tabla anterior no incluyen el IVA aplicable al momento de la compra. La suma de todos estos costes hace un total de **175,18 € (Ciento setenta y cinco euros con dieciocho céntimos)**.

5.3.2.3. Costes de equipamiento

Este apartado detalla los gastos equivalentes al equipamiento requerido para la realización del proyecto, es decir, todo el hardware necesario (ordenador portátil, dispositivo externo, etc.).

La Tabla 20 muestra con detalle los costes de cada dispositivo o equipo adquirido y/o utilizado durante el desarrollo del proyecto así como la vida útil de cada uno de ellos:

Concepto	Coste total	Vida útil (meses)	Coste imputable
Ratón inalámbrico	49,2 €	18	4,10 €
Dos ordenador portátiles	1394,00 €	18	116,17 €
Dispositivos Locomate	2.000,00 €	-	2.000,00 €
Total	-	-	2.120,27 €

Tabla 20. Costes de equipamiento

La vida útil de un equipo portátil suele ser de tres años, que equivale a treinta y seis meses. Los equipos que se han utilizado para este proyecto fueron adquiridos con anterioridad, por lo que su vida útil se ha visto reducida. Del mismo modo, el IVA aplicado en la fecha de compra de los equipos portátiles y del ratón fue del 18%, y no el 21% como es actualmente.

Todos los costes representados en la tabla anterior no incluyen el IVA aplicable al momento de la compra, incluidos los dos equipos portátiles y el ratón inalámbrico. Esto hace un total de **2.120,27 € (Dos mil ciento veinte euros con veintisiete céntimos)**.

5.3.2.4. Costes de oficina

Este apartado detalla los gastos equivalentes a los consumibles del material de oficina, los gastos derivados de la estancia donde se realiza el trabajo requerido (alquiler, luz, agua, etc.), gastos de gestión y servicios generales.

La Tabla 21 muestra con detalle los costes de cada consumible o de cada servicio utilizado durante el desarrollo del proyecto:

Concepto	Coste total	Vida útil (meses)	Coste imputable
Luz/agua/gas	60,00 €	-	47,40 €
Limpieza	75,00 €	-	75,00 €
Seguridad edificio	150,00 €	-	150,00 €
Material de Oficina	20,00 €	-	15,80 €
Internet	48,75 €	-	38,51 €
Router	35,81 €	36	1,49 €
Mesa	110,52 €	36	4,61 €
Silla oficina	138,17 €	24	8,64 €
Papelera	7,82 €	36	0,33 €

Concepto	Coste total	Vida útil (meses)	Coste imputable
Cajonera	48,59 €	24	3,04 €
Alquiler	450,00 €	-	450,00 €
Regleta	7,10 €	24	0,44 €
Alfombrilla para el ratón	7,10 €	24	0,44 €
Funda/Mochila portátil	31,52 €	18	2,63 €
Desplazamientos	59,40 €	-	59,40 €
TOTAL	-	-	857,72 €

Tabla 21. Costes de oficina

Todos los costes representados en la tabla anterior no incluyen el IVA aplicable al momento de la compra o del servicio. Esto hace un total de **857,72 € (Ochocientos cincuenta y siete euros con setenta y dos céntimos)**.

5.3.3. Presupuesto para el cliente

Este apartado muestra el presupuesto que se enseña al cliente, que trata de un resumen de los gastos realizados para la realización del proyecto. Se tendrán en cuenta los costes indirectos de gestión, así como el riesgo de realización del proyecto y un porcentaje de beneficios.

Los costes indirectos del proyecto se estiman en un 10%, elegido así debido a anteriores experiencias del alumno con proyectos desarrollados durante la carrera. El riesgo establecido para el proyecto actual ha sido del 15%, un porcentaje superior al normal, debido a la utilización de nuevas tecnologías en pruebas. El beneficio se corresponde con el 10% del coste acumulado, de nuevo, por anteriores experiencias del alumno. Por último, el IVA aplicable a la fecha de entrega del proyecto se corresponde con el 21% de los costes totales.

La Tabla 22 muestra el presupuesto final que se entregaría al cliente:

Concepto	Coste	Acumulado
Costes de personal	9.583,20 €	9.583,20 €
Costes de software	175,18 €	9.758,38 €
Costes de equipamiento	2.120,27 €	11.878,65 €
Costes de oficina	857,72 €	12.736,37 €
Riesgo (15%)	1.910,46 €	14.646,83 €
Beneficio (10%)	1.464,68 €	16.111,51 €
TOTAL+IVA (21%)	3.383,42 €	19.494,92 €

Tabla 22. Presupuesto para el cliente

Se ha decidido asignar al proyecto un 10% de beneficios, en función de las ganancias que obtendrá el cliente debido a la falta de competitividad por parte de otras empresas. Actualmente, el número de competidores es con respecto a este proyecto es muy reducido.

Como se puede observar en la tabla anterior, el presupuesto final del proyecto es de **19.494,91 € (Diecinueve mil cuatrocientos noventa y cuatro euros con noventa y dos céntimos)**.

5.3.4. Coste final

Este apartado está destinado a mostrar el coste final real del proyecto y a realizar una comparativa con el presupuesto inicial estimado.

Se ha de señalar que la duración total del proyecto, como se ha mencionado anteriormente, ha supuesto un esfuerzo mayor del esperado (medido en horas de trabajo/persona). El esfuerzo realizado final ha supuesto un 16,3% superior al esperado, resultando en un total de cuatro meses y medio de trabajo. Todos los costes reflejados a continuación se referirán a esta duración del proyecto.

5.3.4.1. Costes de personal

Este apartado compara los gastos estimados derivados del personal que haya trabajado en el desarrollo del proyecto con los gastos reales.

La tabla 23 muestra detalladamente los costes del personal que ha intervenido durante el proyecto:

Concepto	Coste/hora	Coste estimado	Coste real	Desviación
Jefe de proyecto	50 €	1.980,00 €	2.316,75 €	336,75 €
Analista	30 €	3.168,00 €	3.706,80 €	538,80 €
Desarrollador	30 €	1.980,00 €	2.316,75 €	336,75 €
Especialista de pruebas	30 €	1.584,00 €	1.853,40 €	269,40 €
SUMA	-	8.712,00 €	10.193,70 €	1.481,70 €
IRPF+SS (15%)	-	871,20 €	1.019,37 €	148,17 €
TOTAL	-	9.583,20 €	11.213,07 €	1.629,87 €

Tabla 23. Comparativa costes de personal

El coste bruto real se correspondería en este caso a un total de 10.193,70 € (Diez mil ciento noventa y tres euros con setenta céntimos). La diferencia con el coste bruto estimado sería de 1.481,70 € (Mil cuatrocientos ochenta y un euros con setenta céntimos).

A este coste habría que sumarle el importe correspondiente a IRPF y Seguridad Social, que se correspondería con la cantidad de 1.019,37 € (Mil diecinueve euros con treinta y siete céntimos) para el coste real, lo que supone una desviación de 148,17 € (Ciento cuarenta y ocho euros con diecisiete céntimos).

Sumando el coste total real con el IRPF+SS correspondiente a esta cifra, obtenemos un coste total neto real de **11.213,07€ (Once mil doscientos trece euros con siete céntimos)**, por lo que la desviación ha sido de un total 1.629,87 € (Mil seiscientos veintinueve euros con ochenta y siete céntimos).

En este caso, la diferencia entre ambos costes se corresponde con la ampliación del tiempo de desarrollo del proyecto explicado en el apartado [5.2. Planificación del Proyecto](#).

5.3.4.2. Costes de software

Este apartado compara los gastos estimados derivados al software necesario que se ha tenido que comprar o utilizar para el correcto desarrollo del proyecto con los gastos de software reales.

La Tabla 24 muestra detalladamente los costes de cada uno de los programas utilizados durante el desarrollo del proyecto así como la vida útil de cada uno de ellos:

Concepto	Coste estimado	Coste real	Desviación
Máquina Virtual (VMWARE)	19,06 €	21,44 €	2,38 €
NetBeans	0,00 €	0,00 €	0,00 €
Microsoft Project	24,69 €	30,28 €	5,59 €
Microsoft Office 2010 Professional	24,79 €	30,41 €	5,62 €
Visual Paradigm	23,01 €	28,22 €	5,21 €
Dropbox	13,63 €	16,72 €	3,09 €
Donación software libre	70,00 €	70,00 €	0,00 €
TOTAL	175,18 €	197,07 €	21,89 €

Tabla 24. Comparativa costes de software

Todos los costes representados en la tabla anterior no incluyen el IVA aplicable al momento de la compra. La suma de todos los costes reales hace un total de **197,07 € (Ciento noventa y siete euros con siete céntimos)**, esto supone una desviación de 21,89 € (Veintiún euros con ochenta y nueve céntimos).

Al igual que antes, la desviación de los costes estimados contra los reales se debe a un mayor uso de los equipos debido a la ampliación de la duración del proyecto.

5.3.4.3. Costes de equipamiento

Este apartado compara los gastos estimados equivalentes al equipamiento requerido para la realización del proyecto frente a los gastos reales de los mismos.

La Tabla 25 muestra con detalle los costes de cada dispositivo o equipo adquirido y/o utilizado durante el desarrollo del proyecto así como la vida útil de cada uno de ellos:

Concepto	Coste estimado	Coste real	Desviación
Ratón inalámbrico	4,10 €	5,03 €	0,93 €
Dos ordenador portátiles	116,17 €	142,50 €	26,33 €
Dispositivos Locomate	2.000,00 €	2.000,00 €	0,00 €
TOTAL	2.120,27 €	2.147,53 €	27,26 €

Tabla 25. Comparativa costes de equipamiento

Todos los costes representados en la tabla anterior no incluyen el IVA aplicable al momento de la compra. Esto hace un total de **2.147,53 € (Dos mil ciento cuarenta y siete euros con cincuenta y tres céntimos)** para los costes reales, por lo que la desviación supondría un total de 27,76 € (Veintisiete euros con veintiséis céntimos).

La desviación de los costes estimados contra los reales se debe a un mayor uso de los equipos debido a la ampliación de la duración del proyecto.

5.3.4.4. Costes de oficina

Este apartado compara los gastos estimados equivalentes a los consumibles del material de oficina, gastos de gestión, etc. con los gastos reales.

La Tabla 26 muestra con detalle los costes de cada consumible o de cada servicio utilizado durante el desarrollo del proyecto:

Concepto	Coste estimado	Coste real	Desviación
Luz/agua/gas	47,40 €	58,14 €	10,74 €
Limpieza	75,00 €	92,00 €	17,00 €
Seguridad edificio	150,00 €	184,00 €	34,00 €
Material de Oficina	15,80 €	15,80 €	0,00 €
Internet	38,51 €	47,24 €	8,73 €
Router	1,49 €	1,83 €	0,34 €
Mesa	4,61 €	8,47 €	3,87 €
Silla oficina	8,64 €	10,59 €	1,96 €
Papelera	0,33 €	0,60 €	0,27 €
Cajonera	3,04 €	3,72 €	0,69 €
Alquiler	450,00 €	552,00 €	102,00 €
Regleta	0,44 €	0,54 €	0,10 €
Alfombrilla para el ratón	0,44 €	0,54 €	0,10 €
Funda/Mochila portátil	2,63 €	3,22 €	0,60 €
Desplazamientos	59,40 €	72,86 €	13,46 €
TOTAL	857,72 €	1.051,58 €	193,86 €

Tabla 26. Comparativa costes de oficina

Todos los costes representados en la tabla anterior no incluyen el IVA aplicable al momento de la compra o del servicio. Esto hace un total de **1.051,58 € (Mil cincuenta y un euros con cincuenta y ocho céntimos)** para los costes reales, por lo que la desviación sumaría un total de 193,86 € (Ciento noventa y tres euros con ochenta y seis céntimos).

La desviación de los costes estimados contra los reales se debe a un mayor uso de los equipos debido a la ampliación de la duración del proyecto.

5.3.4.5. Costes estimados vs costes reales

En este apartado se realiza una comparación de los costes totales estimados con los costes totales reales, de forma que se pueda ver la diferencia que ha ocurrido entre los presupuestos. Gracias a esta comparativa se mostrará si se ha perdido o ganado dinero durante todo el proyecto.

La Tabla 27 muestra la comparación entre los distintos costes estimados del proyecto con los costes reales:

Concepto	Coste	Real	Desviación
Costes de personal	9.583,20 €	11.213,07 €	1.629,87 €
Costes de software	175,18 €	197,08 €	21,90 €
Costes de equipamiento	2.120,27 €	2.147,53 €	27,26 €
Costes de oficina	857,72 €	1.051,58 €	193,86 €
TOTAL	12.736,37 €	14.609,26 €	1.872,89 €

Tabla 27. Costes estimados vs costes totales

Como se puede observar en la tabla anterior, existe una desviación de los costes de **1.872,89 € (Mil ochocientos setenta y dos euros con ochenta y nueve céntimos)**.

Como se ha mencionado anteriormente, esto se ha debido a la necesidad de ampliar el periodo de trabajo del proyecto, pasando a ser de 4 meses a cuatro meses y medio. Esto ha supuesto un mayor coste de personal (donde más se nota la diferencia) así como una pérdida en cuanto a materiales utilizados que no se han amortizado como deberían al no contemplarse este aumento de tiempo.

6. Conclusiones y Líneas Futuras

En este capítulo se exponen las conclusiones a las que el alumno ha llegado tras la finalización del proyecto. También se expondrán las líneas futuras, que detallan las mejoras que se pueden realizar al sistema desarrollado en este proyecto fin de grado, así como consideraciones o ideas que podrían ayudar a mejorar el sistema en futuras versiones.

6.1. Conclusiones sobre el Proyecto

En este apartado se expondrán las distintas conclusiones del alumno sobre el proyecto ya finalizado. Entre ellas se encontrarán las dificultades que han surgido durante la realización del proyecto, los resultados obtenidos y otras conclusiones de interés.

6.1.1. Resultado obtenido

El planteamiento original de este proyecto fin de grado englobaba tanto el protocolo de envío de sanciones desde la RSU a la OBU, como la certificación de los mensajes enviados con el fin de garantizar el servicio de no-repudio. También se tenían en cuenta el desarrollo de interfaces de usuario que hicieran más sencillo el uso del sistema. Debido a las dificultades encontradas durante el desarrollo del proyecto, se decidió reorganizar el proyecto, de forma que se encargara únicamente de establecer un protocolo de comunicación entre la RSU y la OBU, así como de crear una Interfaz de Usuario perteneciente a la RSU. Tras la realización de dichas reorganización, se puede observar que el proyecto ha conseguido satisfacer el objetivo propuesto.

Por tanto, la funcionalidad actual del sistema consiste en el establecimiento de una conexión desde un equipo host con una RSU dada mediante la interfaz de usuario creada. A través de la interfaz se podrán enviar mensajes equivalentes a las sanciones que se enviarían de ser un sistema sancionador real. La información intercambiada entre los dispositivos LOCOMATE sirve como ejemplo, no es un sistema sancionador.

Este protocolo permite el envío y notificación inmediatos del conductor sobre la infracción que se está cometiendo en ese momento. Debe destacarse que este mecanismo gozaría de cobertura legal de acuerdo al marco del proceso sancionador administrativo en el ámbito del tráfico [1]. El momento y forma de presentación de la notificación que recibe el conductor se estudia en otro Proyecto Fin de Grado paralelo al actual. El protocolo propuesto en el presente documento, ayuda a crear un protocolo de notificación más rápido y eficiente. Esto contribuye también al poder educativo que tienen las sanciones sobre los conductores, que al recibir la notificación, corregirán su conducta para adaptarse a las normativas de circulación vial.

Para poder usar este sistema, el usuario iniciaría la aplicación de la Interfaz de Usuario. Por el cual introduciría los datos correspondientes a la RSU a la que deseemos conectar (nombre de usuario, contraseña y dirección IP). Una vez se establezca la conexión con la RSU se podrán enviar mensajes de tipo sancionador a la OBU, que permanecerá en espera escuchando hasta que la RSU envíe un mensaje. El usuario rellenará todos los datos editables de la sanción, cuando ésta se envíe se registrará en una base de datos para que quede constancia de la sanción.

La OBU recibirá la sanción y procesará todos los datos relevantes para el proyecto que la RSU ha enviado (en este caso nos interesarán únicamente los datos que identifiquen la sanción de forma unívoca: matrícula del vehículo y la fecha y hora de la sanción). La OBU enviará un mensaje indicando que ha recibido correctamente la sanción. La RSU, que permanecerá en espera hasta recibir el acuse de recibo, procesará los datos del mensaje enviado por la OBU. La RSU guardará en un registro todos los acuses de recibo que le envíe la OBU para su posterior recogida por la interfaz. Cada uno de los envíos se repite un número N de veces para contrarrestar la eventual pérdida de información en la red vehicular.

El usuario, por último, podrá visualizar el estado de recepción de las sanciones así como todos los datos de la propia sanción: matrícula, fecha y hora de la sanción, fecha y hora del envío, etc.

Teniendo en cuenta esto último, se puede concluir que el resultado obtenido del proyecto es muy satisfactorio con respecto a la reorganización establecida, si bien todavía es posible el desarrollo de múltiples mejoras.

Para reforzar esta idea se ha de añadir que se han llegado a completar tareas adicionales al proyecto, como la preparación de la Interfaz de Usuario para funciones futuras, como la configuración de la RSU para establecer un tramo de control. También se han añadido mejoras personales, como la comprobación de la conexión con la RSU desde la Interfaz.

6.1.2. Dificultad del Proyecto

En este apartado se expondrán los distintos contratiempos y dificultades encontradas durante el desarrollo del proyecto.

En primer lugar, se ha de mencionar la necesidad de tener físicamente los dispositivos LOCOMATE como dificultad principal. A pesar de la documentación proporcionada por el fabricante, la cantidad de funcionalidad que contienen estos dispositivos así como el uso de tecnología muy nueva que todavía permanece en pruebas, hacía muy difícil el análisis de dichas funciones sin la ejecución de los programas de ejemplo que venían con los dispositivos. Para ejecutar los programas de ejemplo era necesario tener acceso al menos a un dispositivo, por lo que el horario de trabajo se veía muy reducido.

Esta dificultad nos conduce a otra de las principales dificultades que se han encontrado para desarrollar el proyecto de forma más rápida y dinámica: el alumno realizaba prácticas de

empresa al mismo tiempo que terminaba los estudios y realizaba el proyecto. Este hecho apoya la anterior dificultad, puesto que el tiempo dedicado al proyecto se veía muy reducido, tanto por el horario de las prácticas como la necesidad de dedicar tiempo a los demás estudios.

Otra de las dificultades principales del proyecto ha sido la reorganización de la planificación del proyecto, debido a que este TFG estaba concebido para ser realizado en paralelo a otro, que desarrollaría otro alumno. Por dificultades sobrevenidas, dicha cooperación no fue posible, lo que conllevó a cambiar la estructura del proyecto que se estaba desarrollando con el fin de entregar un proyecto completo y probado.

6.2. Líneas Futuras

En este apartado se explican posibles mejoras que se podrían tener en cuenta como directrices para aumentar la funcionalidad de los siguientes sistemas basados en el envío de mensajes DSRC.

En este proyecto la comunicación entre OBU y RSU se realiza de forma estática, manteniendo siempre el mismo canal, proveedor y usuario de recepción. Con el manejo adecuado de estos parámetros, podríamos tratar con un sistema multi-canal que recibiera los datos de múltiples dispositivos. Gracias a un sistema similar se podría realizar también la comunicación entre distintas OBU, y no sólo realizar el envío de mensajes entre RSU y OBU.

Como se ha mencionado anteriormente, este sistema debería tener un sistema de seguridad, por el cual se certifiquen todos los mensajes que se envíen por estos dispositivos. Los certificados de los dispositivos OBU deberán renovarse cada cierto tiempo establecido, o bien cuando ocurriera algún tipo de evento. De esta forma no se podría rastrear los itinerarios de los vehículos que tengan esta tecnología.

En un futuro, con un sistema mucho más desarrollado, la RSU podría enviar mensajes de aviso a los conductores que sobrepasaran la velocidad límite establecida para una calzada. Mediante el aviso, el conductor reduciría la velocidad sin llegar a imponer una sanción.

Además, este tipo de dispositivos podría aprovecharse para otros sistemas basados en la geo-localización. Por ejemplo, podría utilizarse como GPS, de forma que se pudiera calcular los itinerarios que desee el conductor del vehículo como hacen otros dispositivos actualmente.

Hay que destacar algunas mejoras que deberían hacerse de forma inmediata tras la finalización del proyecto, como puede ser la implementación de consultas de sanciones mucho más refinadas, capaces de buscar por uno o varios campos pertenecientes a la sanción. Esto permitiría la localización de una sanción concreta, independientemente de la fecha, o bien se podrían recuperar las sanciones impuestas por una autoridad en concreto, etc.

Cuando el sistema esté mucho más avanzado, debería ser capaz de recoger todos los datos de un vehículo de forma automática, sin necesidad de la ventana de envío de sanciones. La RSU

detectaría la infracción e impondría una multa de forma automática. La recogida de datos se realizaría de forma muy similar a la recepción de acuses de recibo que se ha implementado en este proyecto.

También sería posible el desarrollo de un cliente móvil para realizar la conexión con la OBU. Gracias a este cliente móvil, el conductor sería capaz de ser informado de aquellas infracciones que ha cometido de forma instantánea. También sería posible informar al conductor antes de ser sancionado, o bien informar de condiciones adversas en la conducción en una vía próxima a la situación del vehículo en ese momento.

Otra mejora a tener en cuenta trataría sobre el almacenamiento de todos los datos en la nube. Con este tipo de sistema se podrían realizar copias de seguridad de la Base de Datos de forma fácil, periódica y segura. La sincronización de datos en la nube supondría la posibilidad de utilizar dispositivos móviles para la visualización de la información o bien la utilización de un dispositivo que no tuviera que tener acceso físico a los datos (siempre y cuando se tengan los credenciales y permisos necesarios para llevar a cabo dichas acciones).

Bibliografía y Referencias

Este capítulo está dedicado a la mención de todas las referencias que se han utilizado a lo largo del proyecto.

- [1] J. M. de Fuentes, «Improvements on the enforcement process based on Intelligent Transportation techniques. Model and mechanisms for electronic reporting, offence notification and evidence generation,» Julio 2012. [En línea]. Available: http://e-archivo.uc3m.es/bitstream/10016/15535/2/tesis_jose_maria-de-fuentes_garcia-romero.pdf. [Último acceso: Junio 2013].
- [2] Y. (. Li, «Nicta,» [En línea]. Available: <http://www.nicta.com.au/pub?doc=4390>.
- [3] «Arada Systems Support - User Guide,» [En línea]. Available: <http://support.aradasystems.com/file.php?tab=files&file=3401>. [Último acceso: Febrero 2013].
- [4] «National Institute of Standards and Technology,» [En línea]. Available: <http://gsi.nist.gov/global/docs/sit/2010/its/GConoverFriday.pdf>. [Último acceso: Mayo 2013].
- [5] S. E. Shladover y S. K. Tan, «Journal of ITS,» de *Analysis of Vehicle Positioning Accuracy Requirements for Communication-Based Cooperative Collision Warning*, 2006, pp. 131-140.
- [6] «Android,» Android, [En línea]. Available: <http://www.android.com/>. [Último acceso: Junio 2013].
- [7] «Google,» Google, [En línea]. Available: <https://www.google.es/intl/es/about/>. [Último acceso: Abril 2013].
- [8] «Microsoft,» Microsoft, [En línea]. Available: <http://www.microsoft.com/es-es/default.aspx>. [Último acceso: Abril 2013].
- [9] «Arada Systems,» Arada Systems, [En línea]. Available: <http://www.aradasystems.com/>. [Último acceso: Febrero 2013].
- [10] «UML,» [En línea]. Available: <http://www.uml.org/>. [Último acceso: Marzo 2013].
- [11] «W3C,» [En línea]. Available: <http://www.w3.org/XML/>. [Último acceso: Mayo 2013].
- [12] «Linux,» [En línea]. Available: <http://www.linux.org/>. [Último acceso: Mayo 2013].
- [13] «Microsoft Windows,» [En línea]. Available: [Página 122 de 136](http://windows.microsoft.com/es-</div><div data-bbox=)

- es/windows/home. [Último acceso: Junio 2013].
- [14] «VirtualBox,» [En línea]. Available: <https://www.virtualbox.org/>. [Último acceso: Junio 2013].
- [15] «VMware,» [En línea]. Available: <http://www.vmware.com/es/>. [Último acceso: Junio 2013].
- [16] «Ubuntu,» [En línea]. Available: <http://www.ubuntu.com/>. [Último acceso: Mayo 2013].
- [17] «FileZilla Server,» [En línea]. Available: <https://filezilla-project.org/>. [Último acceso: Mayo 2013].
- [18] «Cerberus FTP Server,» [En línea]. Available: <http://www.cerberusftp.com/>. [Último acceso: Junio 2013].
- [19] «Dropbox,» [En línea]. Available: <https://www.dropbox.com/>. [Último acceso: Febrero 2013].
- [20] «NetBeans,» [En línea]. Available: <https://netbeans.org/>.
- [21] «Eclipse,» [En línea]. Available: <http://www.eclipse.org/>.
- [22] «MySQL,» [En línea]. Available: <http://www.mysql.com/>. [Último acceso: Mayo 2013].
- [23] «Microsoft Office 2010,» [En línea]. Available: <http://office.microsoft.com/es-es/support/introduccion-de-microsoft-office-2010-FX100996114.aspx>. [Último acceso: Mayo 2013].
- [24] «SQL Server,» [En línea]. Available: <http://www.microsoft.com/es-es/sqlserver/default.aspx>. [Último acceso: Mayo 2013].
- [25] «MySQL Workbench,» [En línea]. Available: <http://www.mysql.com/products/workbench/>. [Último acceso: Abril 2013].
- [26] «XAMPP,» [En línea]. Available: <http://www.apachefriends.org/en/xampp-windows.html>. [Último acceso: Abril 2013].
- [27] «wordpress.com,» [En línea]. Available: <http://debuenamano.wordpress.com/2012/03/05/modelo-vista-controlador/>. [Último acceso: Mayo 2013].
- [28] [En línea]. Available: http://upload.wikimedia.org/wikipedia/commons/thumb/a/a9/ModelViewControllerDiagram_es.svg/300px-ModelViewControllerDiagram_es.svg.png. [Último acceso: Junio 2013].

- [29] «MySQL JDBC Connector,» [En línea].
13.171,97 € | 15.118,95 € | 1.946,98 € |
Available: <http://dev.mysql.com/downloads/connector/j/>. [Último acceso: Marzo 2013].
- [30] «Apache Commons,» [En línea]. Available:
<http://commons.apache.org/proper/commons-net/>. [Último acceso: Marzo 2013].
- [31] «Adobe Reader,» [En línea]. Available: <http://www.adobe.com/es/>. [Último acceso: Mayo 2013].
- [32] «Google Chrome,» [En línea]. Available:
<https://www.google.com/intl/es/chrome/browser/?hl=es>. [Último acceso: Junio 2013].
- [33] «Java,» [En línea]. Available: <http://www.java.com/es/>. [Último acceso: Junio 2013].
- [34] «Pencil,» [En línea]. Available: <http://pencil.evolus.vn/>. [Último acceso: Mayo 2013].
- [35] «Microsoft Project 2010,» [En línea]. Available:
<http://www.microsoft.com/project/es/es/project-professional-2010.aspx>. [Último acceso: Mayo 2013].
- [36] «Visual Paradigm,» [En línea]. Available: <http://www.visual-paradigm.com/>. [Último acceso: Mayo 2013].
- [37] «GMail,» [En línea]. Available: <https://mail.google.com/mail/?tab=wm>. [Último acceso: Febrero 2013].

Anexos

En este capítulo se muestran todos los anexos relevantes para el presente proyecto.

Anexo 1: Estructura sanciones

El primer anexo se corresponde con la estructura de los mensajes que envía la RSU a la OBU para imponer una sanción. El mensaje tiene formato XML para poder identificar cada campo de forma fácil y rápida. Dentro de las diferentes etiquetas hay cinco asteriscos que representarían los datos que se deseen enviar.

```
<fineNotification>
  <id>*****</id>
  <fineDate>*****</fineDate>
  <offenceDesc>
    <time>*****</time>
    <place>*****</place>
    <offenderID>*****</offenderID>
    <rule>*****</rule>
  </offenceDesc>
  <offenceData>
    <perceivedSpeed>*****</perceivedSpeed>
    <speedLimit>*****</speedLimit>
    <evidenceFile>*****</evidenceFile>
  </offenceData>
  <offenceWitness>
    <witnessID>*****</witnessID>
    <witnessType>*****</witnessType>
    <witnessAff>*****</witnessAff>
  </offenceWitness>
  <authority>
    <ID>*****</ID>
    <contactInfo>*****contactInfo>
  </authority>
  <fineEffects>
    <demeritPoints>*****</demeritPoints>
    <economicPenalty>*****</economicPenalty>
  <drivingLicenseWithdrawal>*****</drivingLicenseWithdrawal>
</fineEffects>
  <appealingInfo>
    <timeInterval>*****</timeInterval>
    <method>*****</method>
    <entity>*****</entity>
  </appealingInfo>
</fineNotification>
```

Anexo 2: Estructura fichero acuses de recibo

El segundo anexo se corresponde con la estructura del fichero de registro de acuses de recibo que almacena la RSU cuando la OBU le envía un acuse de recibo. Al igual que en el apartado anterior, los cinco asteriscos escritos dentro de cada etiqueta representarían los datos enviados y registrados.

También se puede utilizar el presente anexo para visualizar la estructura de los acuses de recibo enviados por parte de la OBU. Para ello deberemos fijarnos únicamente en la etiqueta `<offenceResponse>` y en todos los datos que contenga.

```
<responses>
  <offenceResponse>
    <license>*****</license>
    <date>*****</date>
  </offenceResponse>
  <offenceResponse>
    <license>*****</license>
    <date>*****</date>
  </offenceResponse>
  <offenceResponse>
    <license>*****</license>
    <date>*****</date>
  </offenceResponse>
</responses>
```

Anexo 3: Manual de Usuario

Este apartado tiene gran utilidad con el fin de saber el funcionamiento real de la aplicación final. Se explicará las condiciones previas que se deben cumplir para que el sistema funcione correctamente. Por otro lado se explicará detalladamente para qué sirve cada ventana de la Interfaz de Usuario, qué acciones se pueden realizar en dicha ventana, la navegación entre distintas ventanas así como posibles líneas futuras que se han tenido en cuenta.

Este manual explica el uso del sistema mediante la utilización de dos portátiles conectados cada uno a un dispositivo LOCOMATE, o bien conectando estos dispositivos a un switch que a su vez se conectaría a un portátil que haría de equipo host.

En primer lugar, tanto la RSU como la OBU deben estar enchufadas a la red eléctrica. Primeramente configuraremos la IP de la interfaz de red Ethernet para establecer como IP fija la dirección "192.168.0.100", a continuación conectaremos la OBU al equipo host mediante un cable Ethernet. Nos conectaremos a la OBU mediante el protocolo *Telnet* utilizando el comando siguiente: *telnet 192.168.0.40*. Escribiremos como usuario "*root*" y como contraseña "*password*". Una vez conectados, tendremos que matar el proceso *getwbssencdec*, para ello, usaremos el comando *ps* para visualizar todos los procesos activos y miraremos el identificador de dicho proceso. Mataremos el proceso mediante el comando *kill XXX*, donde XXX se corresponde con el número identificativo de dicho proceso. Para terminar con la OBU, ejecutaremos el siguiente comando: */var./obu 2 1 1 172*. Para salir de la OBU usaremos el comando *exit*.

Ahora que hemos acabado con la OBU, desconectamos el cable Ethernet de la OBU y lo conectamos a la RSU. Al igual que con la OBU, nos conectaremos utilizando un *Telnet* con el mismo usuario y contraseña. Mataremos todos los procesos *getwbssencdec*, siguiendo el mismo procedimiento anterior. Una vez hecho esto, podremos desconectar de la RSU utilizaremos de nuevo el comando *exit*, pero esta vez mantendremos el cable Ethernet conectado al equipo host.

La Interfaz de Usuario ha sido desarrollada para el sistema operativo Windows 7, aunque sería posible ejecutar la aplicación en cualquier sistema operativo que tenga Java versión 1.6.0_22-b04, puesto que se ha desarrollado en el lenguaje Java con el fin de que pueda ser exportable a distintos sistemas operativos.

Para que la Interfaz de Usuario funcione correctamente, la Base de Datos ha de estar conectada desde el primer momento antes de que se comience la ejecución de la Interfaz, esto es debido a que, al ser un sistema multi-lenguaje, se ha utilizado la Base de Datos para almacenar todos los textos de la aplicación. Si no estuviera la Base de Datos disponible al iniciar la aplicación, se mostrará un mensaje de error, indicando que no se ha podido conectar con la Base de Datos, así como el error que ha ocurrido para que fallara la conexión.

La Ilustración 35 muestra el mensaje de error explicado anteriormente, indicando que no se ha llevado a cabo la conexión con la Base de Datos debido a que no se ha iniciado el servidor de MySQL:

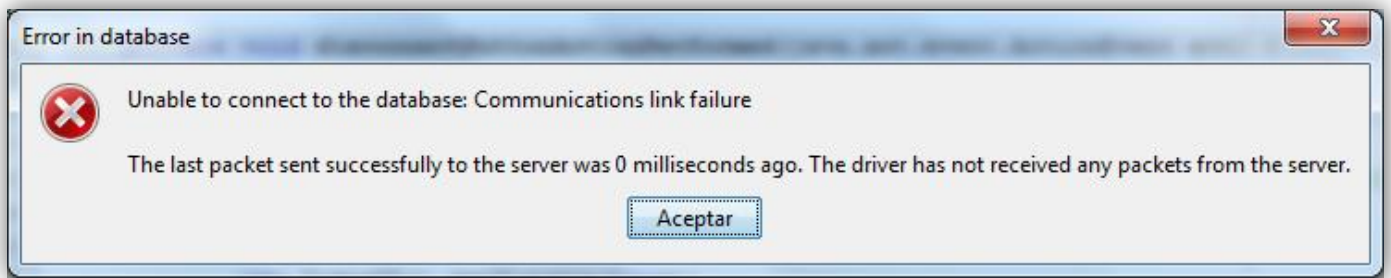


Ilustración 35. Error conexión BBDD

Si no hay ningún problema durante la conexión con la Base de Datos, la aplicación se iniciará en la ventana principal, que permitirá, entre otras cosas, la conexión con una RSU. Inicialmente no hay ninguna RSU favorita, por lo que tendremos la dirección IP por defecto de la RSU que el fabricante Arada Systems ha enviado para la realización del proyecto en la lista de RSU (la dirección IP por defecto es 192.168.0.40).

Desde esta ventana podremos acceder a la ventana de *“Propiedades”* pulsando el botón de *“Properties”*, donde se configurará la aplicación y se podrán gestionar las RSU favoritas. Por otro lado, podremos ver todas aquellas personas involucradas en el proyecto pulsando sobre el logo de la UC3M. Por último, se podrá realizar una de las acciones principales de la aplicación: realizar la conexión con una RSU. Para realizar la conexión con una RSU se necesitará introducir el usuario de acceso, la contraseña para ese usuario, así como la dirección IP correcta de la RSU. En esta ventana no se podrán añadir nuevas RSU, sino que tendremos que ir a la ventana de *“Propiedades”* para gestionar la lista de favoritos. En la ventana principal de la Interfaz, se mostrarán inicialmente los nombres de las RSU favoritas (o la dirección IP si no tuviera un nombre asignado). Si pulsáramos sobre la opción *“Show IP”* mostrarían las distintas IP de las RSU favoritas.

Se ha de señalar que en el caso de realizar una conexión con una RSU, aunque ésta sea fallida, se guardará la última dirección IP utilizada durante la conexión para que la próxima vez que se inicie la Interfaz se muestre como primera opción. Esto también se realizará con direcciones IP que no estén dentro de la lista de favoritos.

La Ilustración 36 muestra la ventana principal de la Interfaz:



Ilustración 36. IndexUI

Si pulsáramos sobre el logo de la UC3M se nos abriría una nueva ventana como la que se puede ver en la Ilustración 37, que muestra la ventana "About", donde aparecen todos los implicados en el proyecto así como el nombre del proyecto:



Ilustración 37. AboutUI

Si en la ventana principal pulsáramos sobre el botón *“Properties”*, la ventana principal desaparecería y aparecería a cambio la ventana de *“Propiedades”*. Aunque inicialmente la aplicación está en inglés, en esta nueva ventana se podrá elegir el idioma de la aplicación (inglés o español). Cuando se elija un idioma, la aplicación guardará el idioma seleccionado en la Base de Datos, de forma que al iniciar de nuevo la aplicación, se inicie con el último idioma seleccionado.

En esta ventana también se podrán añadir nuevas direcciones IP de RSU temporales, estas direcciones no se guardarán en la Base de Datos, sino que se podrán utilizar para la conexión, pero tras el cierre de la Interfaz se borrarán todas (excepto que una dirección IP temporal sea la última usada, como se ha explicado anteriormente). En el caso de introducir una dirección IP no válida (que no tenga el formato correcto), se mostrará un mensaje de error indicando el fallo cometido.

Pulsando sobre el botón *“Test”* Se podrá realizar una prueba de conexión de la dirección IP elegida con el fin de saber si la RSU está disponible o no. También se podrán gestionar la lista de RSU favoritas pulsando sobre el botón *“Admin”*.

En el caso de querer guardar la dirección IP introducida se pulsará sobre el botón *“Save”*, de lo contrario, pulsando sobre el botón *“Cancel”*, se mantendrá la IP anterior. En ambos casos se volverá a mostrar la ventana principal de conexión.

La Ilustración 38 muestra la ventana de *“Propiedades”*:

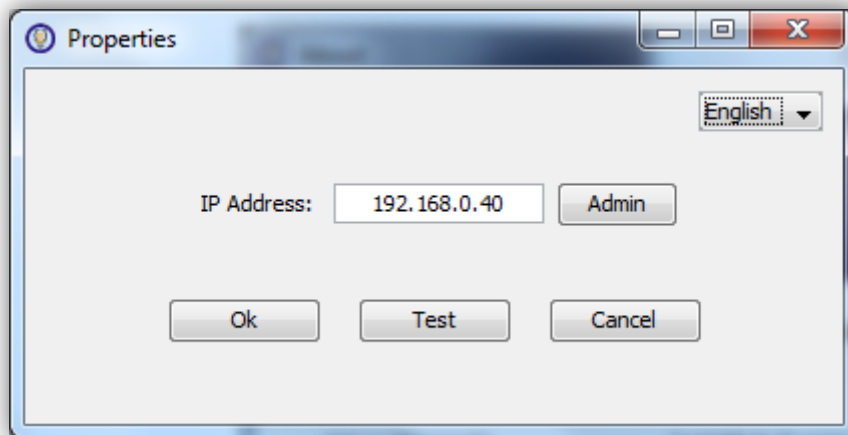


Ilustración 38. PropertiesUI

En el caso de que pulsemos sobre el botón “Admin”, se cerrará la ventana de “Propiedades” y se nos mostrará la ventana de “Administración de Favoritos”. En esta ventana se podrán añadir, editar o eliminar las RSU favoritas. Todas las RSU podrán tener un nombre textual. En caso de no introducir ningún nombre, el nombre será la propia dirección IP.

Esta ventana comprobará automáticamente las direcciones IP que se introduzcan en el campo de texto “IP”. Esto quiere decir que si introducimos una dirección IP que no sea favorita se seleccionará automáticamente la opción de añadir favorito, por el contrario, si introducimos una dirección IP que ya está en la lista de favoritos se bloqueará la opción de añadir (puesto que ya está en la lista) y se desbloquearán las opciones de editar y eliminar favorito.

Al igual que en las demás ventanas, si queremos guardar los cambios producidos, pulsaremos sobre el botón “Save”, por lo que se añadirá, modificará o eliminará el favorito introducido. De no querer aplicar los cambios, pulsaremos sobre el botón “Cancel”.

Si la dirección IP introducida no tiene el formato correcto, el campo de IP se pondrá en rojo, indicando que algún carácter no es correcto. Además, si a pesar de estar el formato incorrecto pulsamos sobre la opción de guardar, se nos mostrará un mensaje, al igual que antes, indicando que el formato de la dirección IP no es correcto.

La Ilustración 39 muestra la ventana de “Administración de Favoritos”:

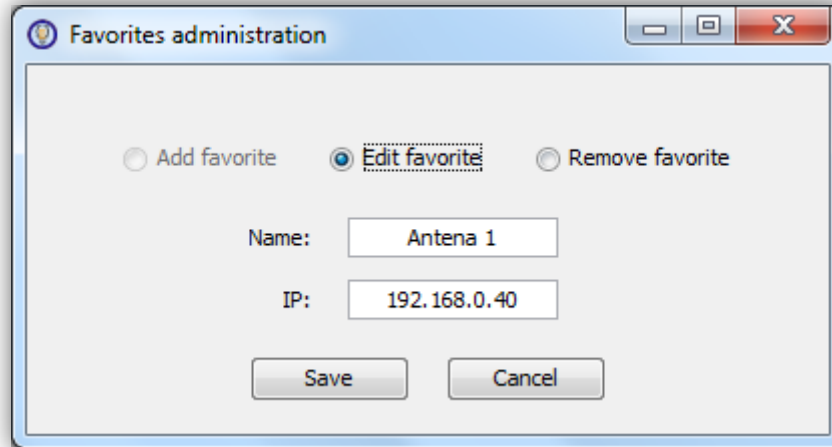


Ilustración 39. AdminFavoritesUI

Si pulsamos sobre el botón “Test” en la ventana de “Propiedades”, se cerrará dicha ventana y se mostrará la ventana de “Test”, que comprobará la conexión con la dirección IP que estuviera en la ventana anterior mediante un ping. En esta ventana se mostrará un texto indicando el proceso testeo, por lo que primero se indicará que se está realizando el ping a la dirección introducida. Una vez se haya realizado el ping, se mostrará mediante una imagen y de forma textual, el estado del ping, indicando si la conexión con la RSU ha sido satisfactoria o bien no se ha podido realizar la conexión.

Esta ventana permitirá ir a la ventana principal de conexión pulsando sobre el botón “Index”, ir a la ventana de “Propiedades” pulsando sobre el botón “Go back”, o bien repetir el ping pulsando sobre el botón “Test”.

La Ilustración 40 muestra la ventana “Test” tras la realización del ping, que ha resultado ser satisfactorio:

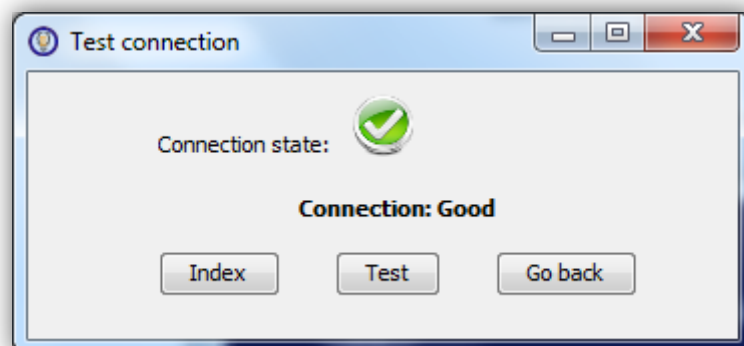


Ilustración 40. PingUI

Si se ha realizado la conexión con éxito a una RSU, la ventana de conexión principal desaparecerá y se nos mostrará la ventana de “E-Save”, desde donde podremos realizar las acciones principales de la aplicación, como enviar una sanción pulsando sobre el botón “Send offence notification”, ver las últimas sanciones pulsando sobre el botón “Last reports” o recibir los últimos acuses de recibos que tenga la RSU registrados pulsando sobre el botón “Receive data”.

En esta ventana también tendremos el logo de la UC3M, que realizará lo mismo que las veces anteriores: permitirá mostrar la ventana de “About” y ver la información del proyecto.

En el caso de pulsar la opción de recibir los últimos acuses de recibo, se nos mostrará una barra de progreso debajo de los botones (entre los botones y el cuadro de texto), que indicará el nivel de progreso de la importación de datos. Al terminar la importación la barra desaparecerá y se mostrará un texto en el cuadro de texto, indicando que se han recibido todos los datos.

Los botones “Dynamics speed control” “Get hidden reports” y “Get counter-evidences” están implementados para las líneas futuras del proyecto. El primer botón se correspondería con el control de la RSU para establecer postes de entrada y salida de un tramo, de forma que no se evaluara la velocidad instantánea de los vehículos, sino que se estableciera la media de la velocidad de los vehículos en un tramo. Este botón, por tanto, sería para configurar la RSU de forma que se pueda realizar esta acción. El segundo botón serviría para recibir todas aquellas denuncias ocultas, es decir, si hay alguna OBU que detecte una infracción fuera del alcance de una RSU, registraría la infracción, y cuando estuviera dentro del alcance de una RSU enviaría esta infracción oculta. El tercer botón serviría para recibir las evidencias de defensa de aquellas sanciones que se hayan impuesto incorrectamente. En este caso, se enviaría la sanción a una OBU y se registraría, la OBU por su parte “preguntaría” a las demás OBU dentro de su alcance para comprobar la infracción, de ser incorrecta la sanción, las OBU responderían a la OBU sancionada indicando que no se ha cometido tal infracción, por lo que enviaría a la RSU la evidencia de defensa de dicha infracción.

Si queremos desconectar de la RSU y volver a la pantalla de conexión, pulsaremos sobre el botón “Disconnect”, pero si por el contrario queremos desconectar de la RSU y salir de la aplicación pulsaremos sobre el botón “Exit”.

La Ilustración 41 muestra la ventana “E-Save”:

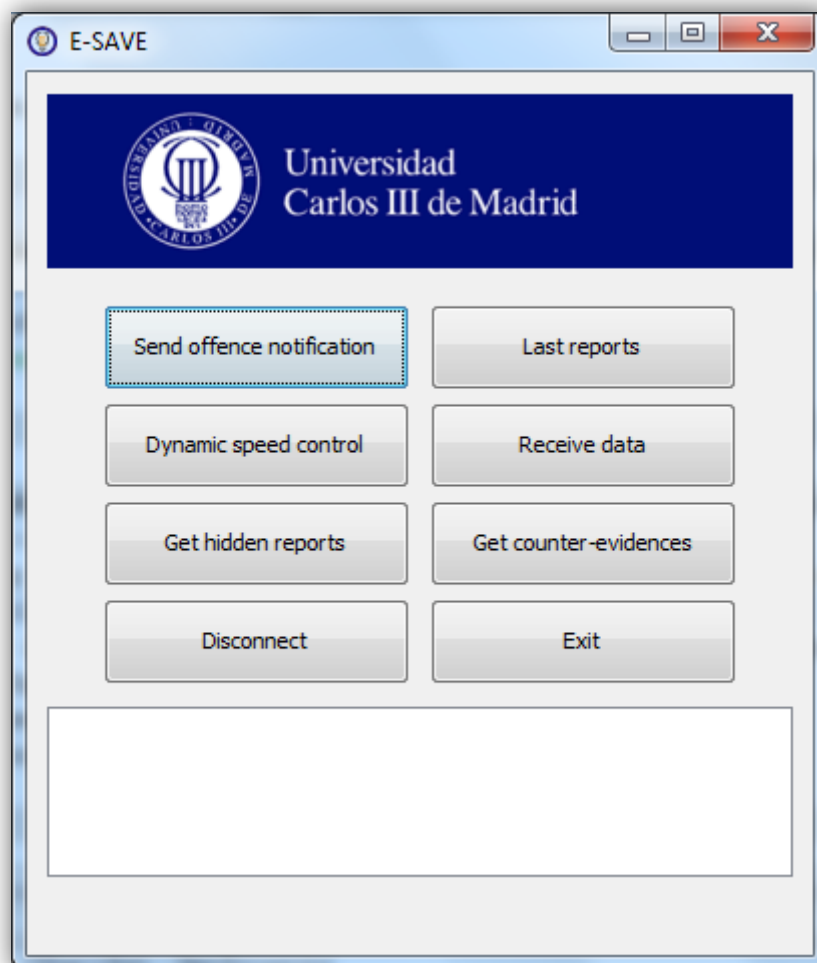


Ilustración 41. ESaveUI

Si pulsamos el botón *"Send offence notificacion"* la ventana *"E-Save"* se cerrará y se nos mostrará la ventana de *"Enviar sanción"*. Esta ventana será la encargada de mostrar un resumen de los datos por los que se sanciona el vehículo.

Se recuerda que este proyecto no está orientado a la imposición de sanciones. El sistema realizará los envíos de las sanciones impuestas mediante la utilización del protocolo desarrollado. Puesto que el sistema no está orientado a la automatización de sanciones, en esta ventana se mostrarán una serie de datos fijos prefijados para ayudar al usuario al envío de sanciones, pero no se tratará de datos reales en ningún momento. La fecha y hora serán los únicos campos dinámicos de este resumen, que se utilizará la fecha y hora del momento en que se llegó a esta ventana.

Se debe señalar también que todos los campos que se muestran en la siguiente ventana deberían estar bloqueados de forma que no se pueda cambiar nada. Sin embargo, con el fin de poder realizar mejores pruebas durante el envío de sanciones, todos los campos están habilitados para que se puedan cambiar y que se pueda realizar el envío de sanciones con datos más variables. A pesar de estar los campos habilitados, no se realiza la comprobación

para que los datos de dicho campo sean correctos, por lo que se deberán utilizar siempre datos válidos.

Al pulsar sobre el botón “Send” se procederá al envío de la sanción a la OBU, se registrará la sanción en la Base de Datos con los datos introducidos en todos los campos de la ventana, así como otros datos que no varían pero que son relevantes para la sanción (como información de contacto, tiempo de reclamación, etc.) y se mostrará un mensaje preguntando si se desean ver las últimas sanciones o bien se desea volver a la ventana “E-Save”. En el caso de pulsar sobre el botón “Cancel”, volveremos a la ventana “E-Save” sin enviar ni registrar la sanción.

La Ilustración 42 muestra la ventana “Enviar sanción”:

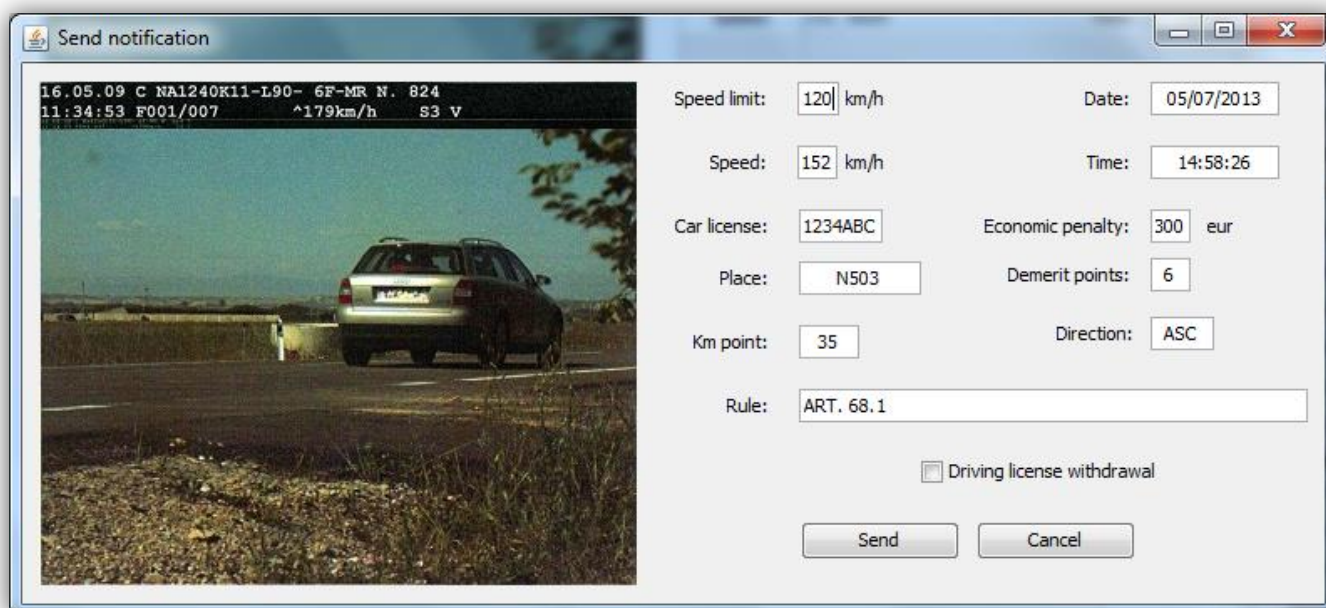


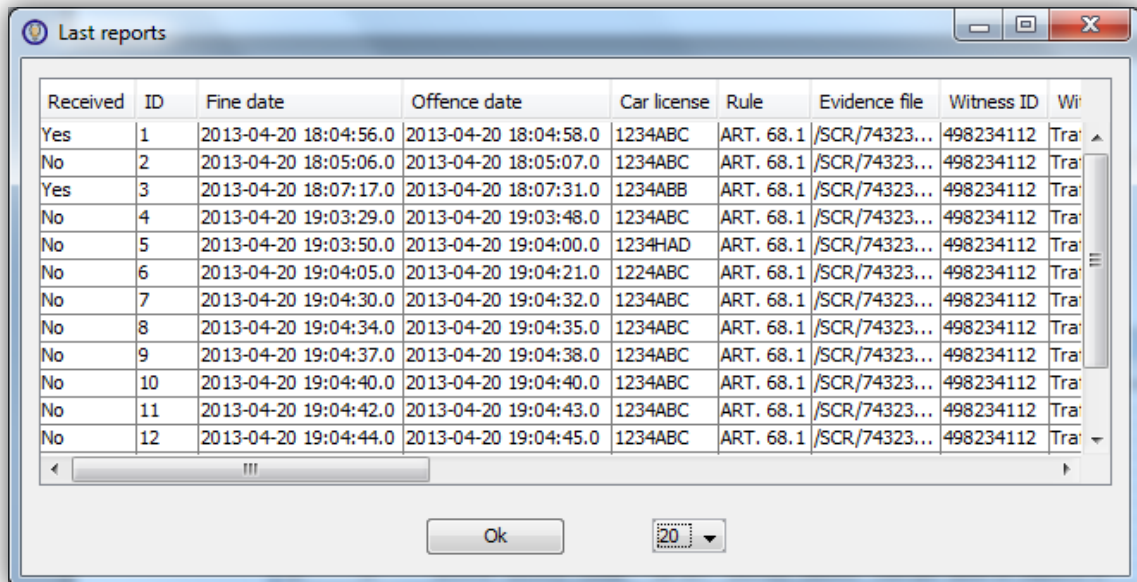
Ilustración 42. SendNotificationUI

Si en la ventana “E-Save” pulsamos sobre el botón “Last reports” o bien pulsamos sobre la opción de ver las últimas incidencias registradas tras el envío de una sanción, se nos mostrará la ventana de “Últimas incidencias”, donde se nos mostrarán las 10 últimas incidencias registradas con todos sus datos (como el estado de recepción de la sanción, la matrícula del vehículo, la fecha de la sanción, la fecha de envío, etc.).

La tabla donde se muestran todos los datos de las sanciones únicamente mostrará los datos de las sanciones, por lo que no se podrán modificar ninguna de las celdas.

Tendremos la opción de ver más incidencias pulsando sobre el botón desplegable que está cerca del botón “Ok”. Gracias a este desplegable se podrán visualizar tantas sanciones como diga el número que seleccionemos de esa lista. Si pulsamos sobre el botón “Ok” volveremos a la ventana de “E-Save”.

La Ilustración 43 muestra la ventana de “Últimas incidencias” con una serie de sanciones enviadas simultáneamente modificando algunos datos y seleccionando que se muestren 20 sanciones:



Received	ID	Fine date	Offence date	Car license	Rule	Evidence file	Witness ID	Wi
Yes	1	2013-04-20 18:04:56.0	2013-04-20 18:04:58.0	1234ABC	ART. 68.1	/SCR/74323...	498234112	Tra
No	2	2013-04-20 18:05:06.0	2013-04-20 18:05:07.0	1234ABC	ART. 68.1	/SCR/74323...	498234112	Tra
Yes	3	2013-04-20 18:07:17.0	2013-04-20 18:07:31.0	1234ABB	ART. 68.1	/SCR/74323...	498234112	Tra
No	4	2013-04-20 19:03:29.0	2013-04-20 19:03:48.0	1234ABC	ART. 68.1	/SCR/74323...	498234112	Tra
No	5	2013-04-20 19:03:50.0	2013-04-20 19:04:00.0	1234HAD	ART. 68.1	/SCR/74323...	498234112	Tra
No	6	2013-04-20 19:04:05.0	2013-04-20 19:04:21.0	1224ABC	ART. 68.1	/SCR/74323...	498234112	Tra
No	7	2013-04-20 19:04:30.0	2013-04-20 19:04:32.0	1234ABC	ART. 68.1	/SCR/74323...	498234112	Tra
No	8	2013-04-20 19:04:34.0	2013-04-20 19:04:35.0	1234ABC	ART. 68.1	/SCR/74323...	498234112	Tra
No	9	2013-04-20 19:04:37.0	2013-04-20 19:04:38.0	1234ABC	ART. 68.1	/SCR/74323...	498234112	Tra
No	10	2013-04-20 19:04:40.0	2013-04-20 19:04:40.0	1234ABC	ART. 68.1	/SCR/74323...	498234112	Tra
No	11	2013-04-20 19:04:42.0	2013-04-20 19:04:43.0	1234ABC	ART. 68.1	/SCR/74323...	498234112	Tra
No	12	2013-04-20 19:04:44.0	2013-04-20 19:04:45.0	1234ABC	ART. 68.1	/SCR/74323...	498234112	Tra

Ilustración 43. LastReportsUI

Como se ha mencionado anteriormente, desde la ventana de “E-Save” se podrá salir de la aplicación o desconectar de la RSU. En ambos casos se mostrará un mensaje para confirmar que el usuario desea salir y/o desconectar de la RSU actual.

La Ilustración 44 muestra un ejemplo del mensaje que se mostraría, en este caso se trata del mensaje de desconexión de la RSU:

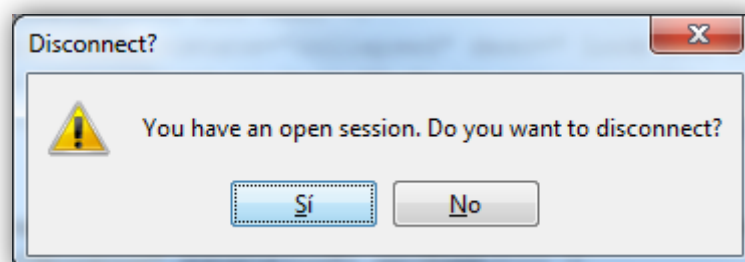


Ilustración 44. Mensaje de desconexión RSU