



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA (SG)

Grado en Ingeniería Informática de Servicios y Aplicaciones

SegoviaEnBus APP: aplicación Android del servicio de
autobuses urbanos de Segovia



Alumno: Julen Barrio García

Tutor: Miguel Ángel Martínez Prieto

Agradecimientos

Muchas gracias a mis profesorxs por su esfuerzo y dedicación para transmitirme sus conocimientos y experiencias, y en especial a mi tutor Miguel Ángel por estar siempre ahí para guiarme y ayudarme, consiguiendo con mucho esfuerzo sacar tiempo de donde no lo había.

Gracias a mi familia por apoyarme en todos estos años, en particular a mi hermana por ofrecerse a leer este documento para ayudar a buscar fallos en la redacción.

Gracias a mis compañerxs por acompañarme en esta etapa y por prestar su ayuda con cualquier problema que pudiese surgir. Y sin olvidar la mención especial a mis compañerxs de cañas los ratos de descanso.

MUCHAS GRACIAS A TODXS!!

Resumen

El objetivo de este proyecto es la creación de una aplicación móvil para los autobuses urbanos de Segovia, con la finalidad de mejorar la experiencia de los usuarios y de este modo incentivar el uso del transporte público.

Además se tratará de ofrecer una serie de funcionalidades enfocadas a aquellos usuarios que no vivan en Segovia, tratando de facilitar la búsqueda de las líneas y paradas de su interés, ya sea a través de una búsqueda de paradas de carácter turístico o a través de un mapa.

En definitiva, con el desarrollo de este proyecto se pretende facilitar la experiencia de los usuarios con el servicio de autobuses urbanos e incentivarla.

Palabras clave: autobús, transporte público, Segovia, aplicación móvil, Android

Abstract

The objective of this project is the creation of a mobile application for urban buses in Segovia, in order to improve the experience of users and thus encourage the use of public transport.

In addition, we will try to offer a series of functionalities focused on users who do not live in Segovia, trying to facilitate the search of the routes and bus stops of their interest, either through a search for service stops through a map.

In short, with the development of this project is intended to facilitate the experience of users with the urban bus service and encourage it.

Keywords: bus, public transport, Segovia, mobiles application, Android

Índice de contenidos

1. INTRODUCCIÓN	15
1.1. Motivación	17
1.2. Objetivos y Alcance	20
1.3. Estructura de la memoria	23
1.4. Contenido del DVD.....	24
2. ESTADO DEL ARTE.....	25
2.1. Antecedentes	27
2.1.1. Bus Segovia – Autobuses urbanos de Segovia	27
2.1.2. BMS – Bus metropolitano de Salamanca	27
2.1.3. Auvasa Bus – Autobuses urbanos Valladolid	28
2.1.4. Bus Madrid – Autobuses urbanos de Madrid	30
2.1.5. Avanza Zaragoza – Autobuses urbanos de Zaragoza	30
2.1.6. GranadaBus – Autobuses urbanos de Granada	31
2.2. Comparación del análisis con la propuesta	32
3. PLANIFICACIÓN Y PRESUPUESTO.....	35
3.1. Metodología de Trabajo	37
3.2. Planificación Temporal.....	38
3.3. Coste Temporal Real	41
3.3. Presupuesto	43
3.3.1. Componentes Hardware	44
3.3.2. Componentes Software.....	44
3.3.3. Costes de personal	44
3.3.4. Método de puntos de función	45
3.3.5. COCOMO II	47
3.3.6. Comparativa de presupuestos	49
3.4. Coste Real	49
4. ANÁLISIS.....	51
4.1. Descripción de los actores	53
4.2. Requisitos de usuario.....	53
4.2.1. Diagramas de casos de uso	54
4.2.1. Especificación de casos de uso.....	57
4.4. Requisitos de Información	60

4.4.1. Modelo conceptual de datos	60
4.4.2. Diccionario de datos.....	61
5. DISEÑO.....	65
5.1. Arquitectura lógica.....	67
5.2. Arquitectura física.....	69
5.2.1. Arquitectura física utilizada	69
5.2.2. Arquitectura física ideal	70
5.3. Diagramas de clases.....	71
5.4. Modelo lógico de la base de datos	72
5.5. Diseño de la interfaz	73
6. IMPLEMENTACIÓN	81
6.1. Herramientas utilizadas	83
6.1.1. Herramientas de análisis	83
6.1.2. Herramientas para el desarrollo web	86
6.1.3. Herramientas para el desarrollo de la aplicación móvil	89
6.2. Implementación	91
6.2.1. Datos utilizados	91
6.2.2. Desarrollo de la aplicación web	92
6.2.3. Desarrollo de la aplicación móvil	99
7. PRUEBAS	105
7.1. Pruebas de caja blanca	107
7.2. Pruebas de caja negra.....	108
8. MANUALES.....	113
8.1. Manual de administrador	115
8.2. Manual de gestor	121
8.3. Manual de usuario	123
9. CONCLUSIONES.....	131
BIBLIOGRAFÍA	135
Apéndice A – Desarrollo detallado de la especificación	139

Índice de tablas

Tabla 1: Cuota de mercado de los diferentes sistemas operativos en smartphones	18
Tabla 2: Comparación del análisis con la APP propuesta	32
Tabla 3: Planificación horas dedicadas.....	40
Tabla 4: Tiempos de desarrollo del proyecto (1).....	41
Tabla 5: Tiempos de desarrollo del proyecto (2).....	42
Tabla 6: Tiempo real invertido	42
Tabla 7: Costes componentes hardware	44
Tabla 8: Costes componentes software	44
Tabla 9: Costes de personal estimados	45
Tabla 10: Complejidades de las componentes	45
Tabla 11: Complejidad de los componentes	46
Tabla 12: Cálculo de puntos de función sin ajustar	46
Tabla 13: Cálculo de coeficientes para el factor de ajuste	46
Tabla 14: Presupuesto Puntos de Función	47
Tabla 15: Factores de ajuste presupuesto COCOMO II	48
Tabla 16: Modificadores para los modelos de proyecto	48
Tabla 17: Presupuesto COCOMO II.....	49
Tabla 18: Coste real de personal	49
Tabla 19: Requisitos de usuario de la aplicación web.....	54
Tabla 20: Requisitos de usuarios de la aplicación móvil	54
Tabla 21: CU-01 Añadir Gestores	57
Tabla 22: CU-02 Activar Mantenimiento.....	57
Tabla 23: CU-03 Desactivar Mantenimiento	58
Tabla 24: CU-04 Crear nueva parada.....	58
Tabla 25: CU-28 Visualizar líneas.....	59
Tabla 26: CU-29 Visualizar recorrido de una línea	59
Tabla 27: CU-30 Visualizar línea en mapa	59
Tabla 28: CU-31 Ver información “Bus Turístico”	60
Tabla 29: Entidad línea	61
Tabla 30: Entidad parada.....	62
Tabla 31: Entidad recorrido	62
Tabla 32: Entidad punto recorrido	63
Tabla 33: Entidad horario	63
Tabla 34: Entidad notificación	64
Tabla 35: Diseño Bienvenida	74
Tabla 36: Diseño Líneas	75
Tabla 37: Diseño Paradas cercanas	76
Tabla 38: Diseño Bus turístico	77
Tabla 39: Diseño Tarifas	78
Tabla 40: Diseño Selección de idioma	79
Tabla 41: Requisitos Mínimos Android Studio	89
Tabla 42: Entidad festivos.....	91

Tabla 43: Entidad mantenimiento	91
Tabla 44: Entidad versiones.....	91
Tabla 45: CP-01: Añadir gestor	108
Tabla 46: CP-02: Autenticarse como gestor	109
Tabla 47: CP-03: Activar modo mantenimiento	109
Tabla 48: CP-04: Añadir parada	109
Tabla 49: CP-05: Eliminar una parada	110
Tabla 50: CP-06: Modificar descripción de una notificación.....	110
Tabla 51: CP-07: Habilitar línea	110
Tabla 52: CP-08: Visualizar listado de líneas (web)	110
Tabla 53: CP-09: Visualizar listado de líneas (app)	111
Tabla 54: CP-10: Visualizar mapa de paradas cercanas	111
Tabla 55: CP-11: Geolocalizar usuario en mapa	111
Tabla 56: CP-12: Ver información de parada.....	111
Tabla 57: CP-13: Añadir a favorita	112
Tabla 58: CP-14: Consultar información de tarifas.....	112
Tabla 59: CP-15: Modificar el idioma de la app.....	112
Tabla 60: CU-05 Visualizar listado de paradas	139
Tabla 61: CU-06 Eliminar una parada	139
Tabla 62: CU-07 Crear línea	140
Tabla 63: CU-08 Visualizar listado de líneas	140
Tabla 64: CU-09 Modificar nombre de línea	141
Tabla 65: CU-10 Habilitar línea	141
Tabla 66: CU-11 Deshabilitar línea	142
Tabla 67: CU-12 Añadir un recorrido a una línea	142
Tabla 68: CU-13 Visualizar Recorridos.....	143
Tabla 69: CU-14 Eliminar Recorrido	143
Tabla 70: CU-15 Añadir punto a recorrido	144
Tabla 71: CU-16 Visualizar puntos de recorrido.....	144
Tabla 72: CU-17 Eliminar punto de recorrido.....	145
Tabla 73: CU-18 Asignar horario.....	145
Tabla 74: CU-19 Visualizar horarios.....	146
Tabla 75: CU-20 Eliminar horario	146
Tabla 76: CU-21 Publicar notificación	147
Tabla 77: CU-22 Visualizar notificaciones.....	147
Tabla 78: CU-23 Eliminar notificación	148
Tabla 79: CU-24 Modificar descripción de notificación	148
Tabla 80: CU-25 Asignar notificación a parada	149
Tabla 81: CU-26 Asignar notificación a línea	149
Tabla 82: CU-27 Autenticar gestor	150
Tabla 83: CU-32 Buscar paradas por nombre.....	150
Tabla 84: CU-33 Visualizar paradas cercanas en mapa	150
Tabla 85: CU-34 Visualizar listado de paradas de interés turístico	151
Tabla 86: CU-35 Consultar información de parada	151

Tabla 87: CU-36 Consultar horarios de parada	152
Tabla 88: CU-37 Visualizar paradas favoritas	152
Tabla 89: CU-38 Modificar idioma de la aplicación	152
Tabla 90: CU-39 Visualizar notificaciones.....	153
Tabla 91: CU-40 Visualizar información tarifas	153
Tabla 92: CU-41 Visualizar información abonos.....	153
Tabla 93: CU-42 Visualizar información notificación.....	154
Tabla 94: CU-43 Añadir parada favorita	154
Tabla 95: CU-44 Eliminar parada favorita	154
Tabla 96: CU-45 Geolocalizar usuario en mapa.....	155
Tabla 97: CU-46 Visualizar parada en mapa.....	155

Índice de ilustraciones

Ilustración 1: Parada sin horarios debido a actos vandálicos	19
Ilustración 2: Ejemplo de dos horarios diferentes en una misma parada	20
Ilustración 3: Árbol de características APP	21
Ilustración 4: Árbol de características de la aplicación web	22
Ilustración 5: Capturas app autobuses urbanos de Segovia	27
Ilustración 6: Capturas Bus Metropolitano de Salamanca app	28
Ilustración 7: Capturas AUVASA Bus	29
Ilustración 8: Capturas Bus Madrid	30
Ilustración 9: Capturas Avanza Zaragoza	30
Ilustración 10: Capturas GranadaBus	31
Ilustración 11: Planificación temporal inicial.....	39
Ilustración 12: Diagrama Gantt Planificación Inicial.....	40
Ilustración 13: Diagrama de Gantt tiempo real invertido	43
Ilustración 14 Diagrama de casos de uso de la aplicación web:	55
Ilustración 15: Diagrama Entidad Relación	61
Ilustración 16: Arquitectura Lógica de la Aplicación Web.....	67
Ilustración 17: Arquitectura Lógica Aplicación Móvil.....	68
Ilustración 18: Arquitectura física utilizada	69
Ilustración 19: Arquitectura física ideal.....	70
Ilustración 20: Diagrama de clases de la aplicación web	71
Ilustración 21: Diagrama de clases de la aplicación móvil	72
Ilustración 22: Modelo relacional.....	72
Ilustración 23: Diseño Bienvenida	74
Ilustración 24: Diseño Líneas	75
Ilustración 25: Diseño Paradas cercanas	76
Ilustración 26: Diseño Bus Turístico	77
Ilustración 27: Diseño Tarifas	78
Ilustración 28: Diseño Selección de Idioma	79
Ilustración 29: Captura de la herramienta Draw.io	83
Ilustración 30: Captura de pantalla de StarUML	84
Ilustración 31: Captura de Photoshop	84
Ilustración 32: Captura Microsoft Word.....	85
Ilustración 33: Captura de la descripción de la plantilla	86
Ilustración 34: Captura edición de página de Inicio	87
Ilustración 35: Captura panel de administración de usuarios.....	87
Ilustración 36: Conexión mediante FileZilla	88
Ilustración 37: Captura Programa Sublime Text 2.....	89
Ilustración 38: Emulador de Android Studio	90
Ilustración 39: Ejemplo Mapa Google Maps en Android	90
Ilustración 40: Panel shortcodes aplicación	93
Ilustración 41: Login panel de administración de WordPress.....	117
Ilustración 42: Introducción de shortcode en la web.....	118

Ilustración 43: Formulario Añadir Nueva Parada	118
Ilustración 44: Ejemplo de estructura del menú	119
Ilustración 45: Añadir nivel de membresía en plugin de sesión.....	119
Ilustración 46: Protección de acceso a una página de la aplicación web.....	120
Ilustración 47: Panel de inicio de sesión de la aplicación web.....	121
Ilustración 48: Estado de mantenimiento de la aplicación web	121
Ilustración 49: Ejemplo adición de parada desde la aplicación web.....	122
Ilustración 50: Ejemplo de adición de horario desde la aplicación web	122
Ilustración 51: Panel para habilitar o deshabilitar líneas desde la aplicación web.....	123
Ilustración 52: Captura líneas APP.....	124
Ilustración 53: Captura recorridos APP	124
Ilustración 54: Captura paradas de recorrido APP	125
Ilustración 55: Captura mapa línea APP	125
Ilustración 56: Captura información de parada	126
Ilustración 57: Captura buscar paradas APP	126
Ilustración 58: Captura paradas cercanas APP	127
Ilustración 59: Captura paradas favoritas APP	127
Ilustración 60: Captura información bus turístico APP	128
Ilustración 61: Captura tarifas APP.....	128
Ilustración 62: Captura cambiar idioma APP.....	129
Ilustración 63: Captura salir APP	129

CAPÍTULO 1

INTRODUCCIÓN

En este primer capítulo, se expondrán las razones que han motivado la elección del proyecto como Trabajo de Fin de Grado, junto a la importancia de la evolución de las nuevas tecnologías. Además se explicarán los objetivos que se desean cumplir y el alcance del proyecto.

1.1. Motivación

El uso del transporte público destaca como solución a los problemas de movilidad producidos en los núcleos urbanos. Mediante la promoción de estos servicios se podría aumentar su uso y por tanto, reducir las retenciones y la contaminación causada por la gran mayoría de los vehículos.

En la actualidad la tecnología está presente en nuestro día a día repartida en diferentes tipos de dispositivos. Entre éstos, los más comunes son los terminales móviles con los que podemos transmitir una gran cantidad de información. Hace unos años estos dispositivos tan solo permitían la realización de llamadas telefónicas pero, con el paso de los años, la tecnología ha evolucionado a pasos agigantados, permitiendo a los usuarios realizar todo tipo de tareas desde el terminal. El principal cambio surgió cuando, a través de los terminales, se pudo acceder a Internet, de este modo se podía acceder a todo tipo de contenido desde cualquier lugar con una conexión disponible. Con la mejora de la infraestructura de red, hoy en día, son pocas las zonas que no cuentan con acceso a la red, esto permite que cada vez haya más gente con la disponibilidad para utilizar estos servicios. La llegada de los smartphones ha hecho que los ordenadores de sobremesa se vean sustituidos progresivamente por estos dispositivos. Ésto se debe a la capacidad de ofrecer funcionalidades similares a un precio más barato y además pudiendo llevarlos a cualquier lugar gracias a su reducido tamaño. Como se puede ver en el artículo del periódico “El Mundo”¹, en el año 2015 en España el uso de dispositivos móviles frente al de los ordenadores estaba igualado, sin embargo, a principios de 2016 por primera vez en la historia los ordenadores pasaron a un segundo lugar, siendo los dispositivos móviles el principal modo de acceder a Internet con un 88.3% de los usuarios frente al 78.2% del ordenador.

Debido a la tendencia creciente del uso de los dispositivos móviles los contenidos web se han tenido que adaptar a ellos. Sin embargo no todo ha sido adaptación, sino que además se han tenido que desarrollar aplicaciones específicas para los terminales. De este modo los usuarios no necesitan buscar en Internet cada vez que desean obtener información sobre el servicio, tan solo tendrán que utilizar una aplicación que pone a su disposición una serie de funcionalidades concretas.

Respecto al sistema operativo para smartphones, el más utilizado en los últimos años ha sido Android. Además, como podemos observar en la *Tabla 1*, en el pasado año 2016 las

¹ «El móvil supera por primera vez al ordenador para acceder a Internet». ELMUNDO, 4 de abril de 2016. <http://www.elmundo.es/sociedad/2016/04/04/57026219e2704e90048b465e.html>.

ventas aumentaron un 4% respecto a 2015 ocupando una cuota de mercado del 86.2% del total de dispositivos móviles.

Este aumento de ventas puede ser debido en gran parte por el precio asequible de la gran mayoría de dispositivos frente al precio de aquellos que cuentan con IOs; su competidor más directo.

En todo caso, la cuota de mercado de Android es mayor y, por tanto, decidimos construir nuestra app para estos dispositivos con el fin de ofrecerla al mayor número de personas posibles.

Sistema Operativo	Unidades 2016	Cuota de mercado 2016 (%)	Unidades 2015	Cuota de mercado 2016 (%)
Android	296.913	86.2	271.647	82.2
iOS	44.395	12.9	48.085	14.6
Windows	1.971	0.6	8.198	2.5
BlackBerry	400	0.1	1.153	0.3
Otros	680	0.2	1.229	0.4
Total	344.359	100	330.312	100

Tabla 1: Cuota de mercado de los diferentes sistemas operativos en smartphones²

Son muchas las ciudades que cuentan con aplicaciones web o móviles para facilitar información sobre el transporte público disponible a los ciudadanos. Estas aplicaciones suelen actualizar su contenido periódicamente otorgando a los usuarios datos fiables. Además, en las aplicaciones más novedosas es frecuente la utilización de sistemas de geolocalización para ayudar a buscar las estaciones o paradas de los diferentes medios de transporte disponibles. Esta funcionalidad también puede ser utilizada para mostrar a los usuarios el tiempo restante para que el próximo vehículo efectúe su parada.

En el caso de Segovia, a nivel urbano el único medio de transporte público disponible es el autobús, que a diferencia de otras ciudades españolas, no cuenta con una aplicación oficial. Tan solo existe una app, explicada más detalladamente en el apartado Estado del arte, desarrollada por un particular con unas funcionalidades limitadas a la consulta de los horarios, insertados manualmente tras su obtención de la página web de autobuses urbanos de Segovia³, y a la visualización de las paradas que forman cada una de las líneas. En los últimos comentarios de la aplicación en Google Play se pueden observar las quejas de los usuarios sobre los horarios desactualizados, hecho que con el desarrollo de este proyecto se pretende solventar.

Sin embargo, es necesaria la colaboración con el Ayuntamiento de Segovia para cumplir este objetivo, debido a que el único medio alternativo de obtención de horarios y recorridos de las diferentes líneas es recogerlos de las paradas distribuidas por la ciudad, y resulta imposible la obtención del conjunto completo y actualizado de los datos debido a los siguientes problemas observados:

² «Gartner Says Five of Top 10 Worldwide Mobile Phone Vendors Increased Sales in Second Quarter of 2016». <http://www.gartner.com/newsroom/id/3415117>.

³ Url de la web de autobuses urbanos de Segovia: <http://www.urbanosdesegovia.com/>

- Vandalismo: Algunas de las paradas de la ciudad han sufrido este tipo de actos y resulta completamente imposible poder acceder a la información de algunas de las paradas (*Ilustración 1*).
- Planos de líneas desactualizados: Son muchos los casos encontrados de líneas en las que sus planos no contienen todas las paradas existentes en los recorridos, tanto en la web, como en la información a la que se puede acceder en las paradas.
- Número de horarios incorrectos: Se ha observado que en algunas paradas hay un número determinado de horas para una línea y un día determinado y sin embargo, en algunas de las paradas pertenecientes al recorrido este número varía. Esto se debe a que algunas de las paradas no tienen los horarios actualizados.
- Horarios incorrectos: Por la misma razón que el problema anterior, algunos autobuses salen de un lugar a una hora determinada y unas paradas más adelante. supuestamente llega a esa misma hora o incluso antes de que haya salido.
- Desactualización de horarios: Ya se ha mencionado este problema en los dos puntos anteriores, pero se ha encontrado algún caso en el que en una parada hay dos horarios diferentes para una misma línea (*Ilustración 2*).
- Inexistencia de paradas físicas: al igual que se menciona que en los planos hay algunas paradas que no aparecen, también se da el suceso de que algunas de las que sí que lo hacen, no tienen una parada física.
- Horarios pegados sobre otros: Se ha detectado alguna parada en la que el horario de una línea estaba situado tapando el de otra diferente.



Ilustración 1: Parada sin horarios debido a actos vandálicos

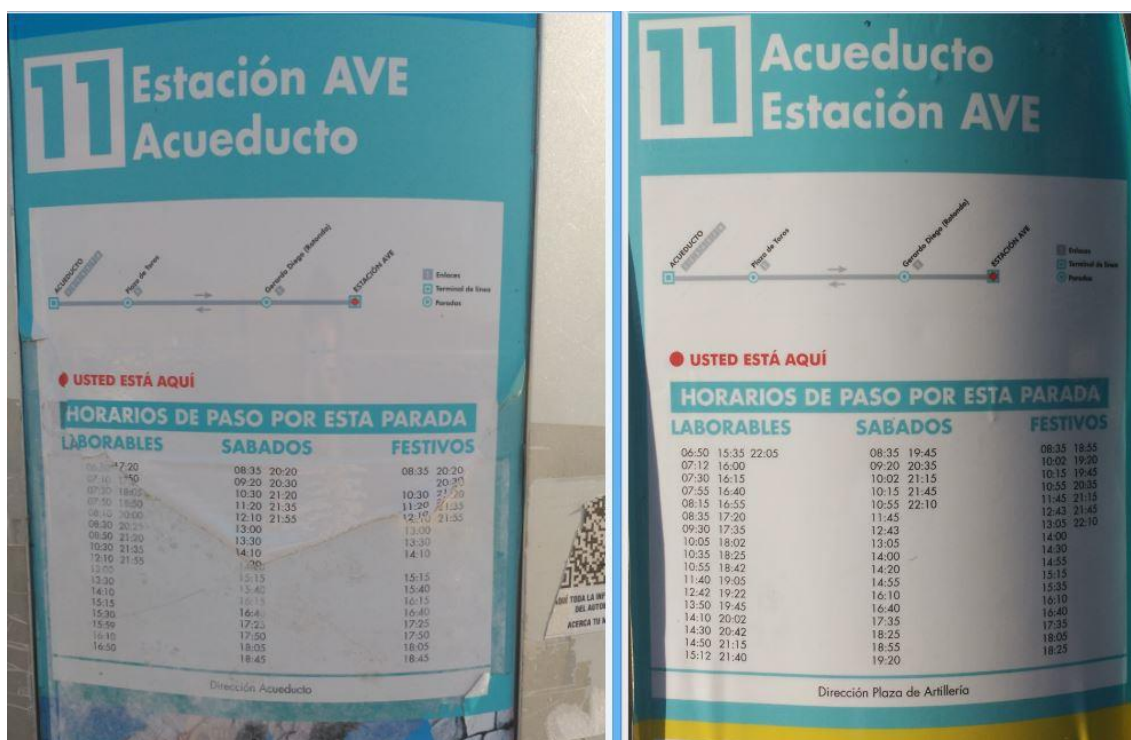


Ilustración 2: Ejemplo de dos horarios diferentes en una misma parada

Por lo tanto, se ve necesaria la colaboración con el Ayuntamiento de Segovia para poder ofrecer a los usuarios, a través de la aplicación que se pretende desarrollar, los horarios actualizados.

Otro de los problemas del servicio de autobuses urbanos de Segovia es la baja afluencia de usuarios, esto hace que, como se menciona en el artículo del periódico local “El Adelantado de Segovia”⁴, Segovia sea una de las capitales de la región donde el autobús urbano está más subvencionado. Por lo tanto, como se detalla en el siguiente apartado, se desarrollará una aplicación móvil para Android con el fin de solventar estos problemas.

1.2. Objetivos y Alcance

“Segovia En Bus” nace como una aplicación móvil (Android) destinada a los usuarios del transporte urbano de la ciudad Segovia. Esta aplicación permite acceder fácilmente a la información de las líneas de autobús, y conocer tanto sus paradas como los diferentes horarios en los que el bus correspondiente pasará a lo largo del día, informando también de posibles incidencias que se produzcan en la prestación del servicio. A diferencia de la operativa actual, los usuarios del transporte urbano de Segovia ya no tendrán que acercarse a mirar, in situ, los horarios publicados en las paradas ni tendrán que

⁴ «Segovia es una de las capitales de la región donde el autobús urbano está más subvencionado - Local – El Adelantado».

http://www.eladelantado.com/noticia/local/226114/segovia_es_una_de_las_capitales_de_la_region_donde_el_autobus_urbano_esta_mas_subvencionado.

descargarse los documentos (no siempre actualizados) disponibles en la web. Gracias a esta herramienta, tanto los ciudadanos como los turistas de las ciudad de Segovia podrán utilizar su *smartphone* para acceder a información actualizada sobre el servicio de autobuses urbanos, aprovechándose de las utilidades que prestan estos dispositivos electrónicos.

Su objetivo principal es utilizar la tecnología móvil para acercar a ciudadanos y turistas la información de interés sobre el servicio de autobuses urbanos de la ciudad de Segovia. Además, el proyecto analiza el estado actual de dicho servicio y reflexiona sobre las posibles mejoras que podrá incorporar gracias al uso de nuevas tecnologías.

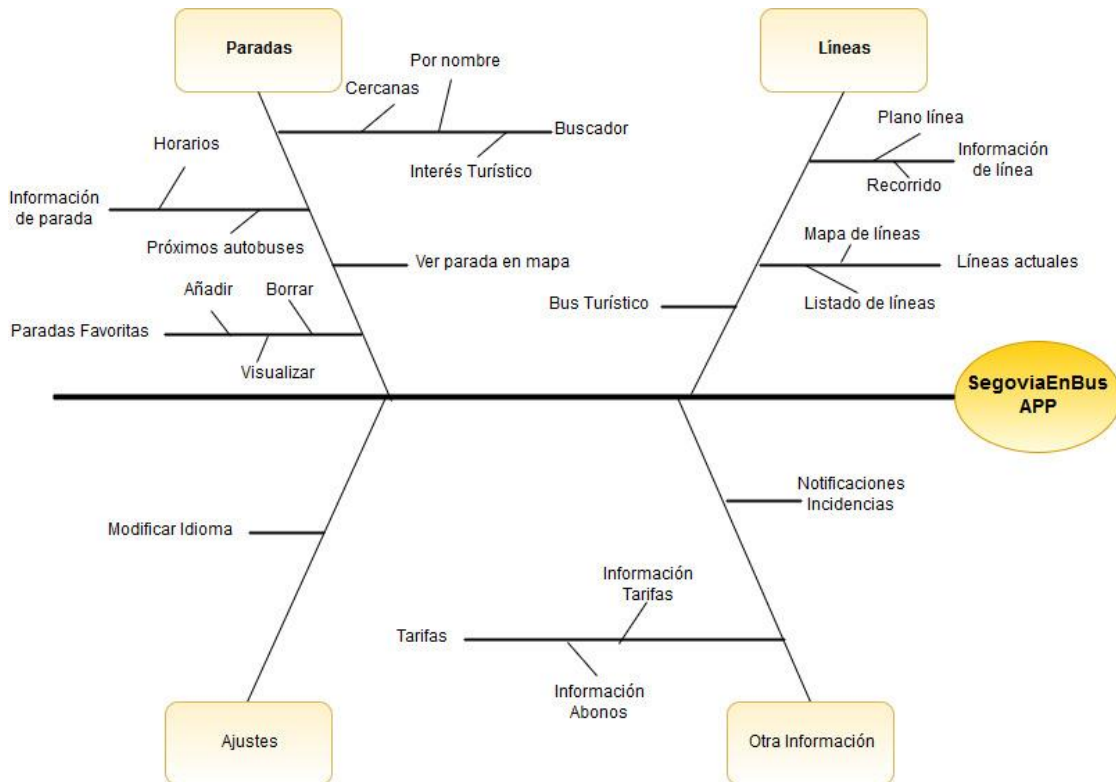


Ilustración 3: Árbol de características APP

Para ello la aplicación deberá proporcionar las siguientes características:

- Consulta de horarios actualizados de las diferentes líneas de autobuses urbanos de Segovia.
- Servicio de geolocalización de paradas.
- Búsqueda de paradas de interés turístico.
- Mostrar el tiempo restante para la llegada de los autobuses a la parada.
- Facilitar información sobre las tarifas y abonos disponibles.

Estas características se ilustran en el anterior árbol de características (*Ilustración 3*).

En el árbol de características de la APP se han establecido cuatro grupos: Líneas, Paradas, Ajustes y Otra Información. En el primer grupo se contemplan todas las funcionalidades referentes a las líneas, tales como la visualización del listado de todas ellas, su información específica (plano de la línea o recorrido), su representación en un

mapa o la búsqueda por nombre. Además, se ofrece una funcionalidad específica para el autobús turístico.

Respecto al apartado de paradas, tenemos la sección de búsqueda. Esta función puede ser realizada de tres maneras: búsqueda por nombre, paradas cercanas y paradas de interés turístico. También podremos visualizar la información específica de cada parada (horarios o información acerca de los próximos autobuses en llegar). Finalmente tenemos las opciones de visualizar la parada en el mapa y el listado de paradas favoritas.

En la rama de ajustes tan solo contemplamos la funcionalidad de modificar el idioma de la aplicación para adaptarse al público extranjero.

En el último apartado, otra información, tenemos la sección de tarifas con la información acerca de ellas y de los abonos y el apartado de notificaciones de incidencias.

Además de la aplicación móvil, se creará una aplicación web desde que se puedan introducir todos los datos utilizados por la app. De este modo, la web tendrá estas características:

- Registrar líneas y paradas.
- Crear los recorridos de cada línea con sus respectivas paradas.
- Asignar horarios a los puntos de cada recorrido.
- Publicar notificaciones y asignarlas a las paradas o líneas que afecten.

Estas características se ilustran en el árbol de características mostrado a continuación (*Ilustración 4*).

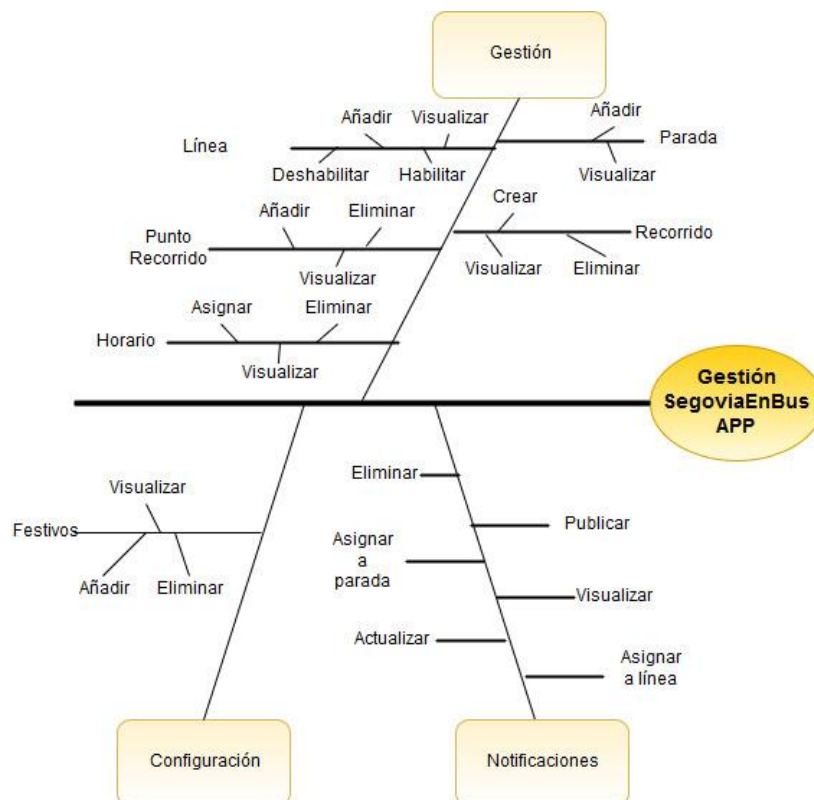


Ilustración 4: Árbol de características de la aplicación web

En este segundo árbol de características contamos con tres grupos: Gestión, Notificaciones y Configuración. En el de gestión, contamos con aquellas funcionalidades necesarias utilizadas introducir todos los datos necesarios para el funcionamiento básico de la aplicación móvil. Éste se centra en la creación de las líneas con sus respectivos recorridos, los cuales están formados por paradas con un horario determinado. Además se podrán visualizar todos los datos introducidos para mayor comodidad del usuario, de este modo, no necesitará acceder al panel de administración de la base de datos para comprobar el éxito de las operaciones. Finalmente en esta sección, contaremos con las operaciones para poder eliminar aquellos recorridos, puntos de recorrido u horarios, que ya no sean de utilidad para el usuario final.

En cuanto a la rama de notificaciones, podemos observar cómo además de poder publicarlas, podremos asignarlas a las líneas o paradas a las que afecten, visualizarlas, modificarlas o eliminarlas del sistema.

Por último, se nos permitirá añadir festivos como configuración de la aplicación y así, poder consultar los horarios de forma correcta dependiendo del día en el que estemos. Al igual que ocurre con el resto de las entidades con las que trabajamos en el sistema, podremos visualizar todos los festivos almacenados y eliminar aquellos que deseemos.

1.3. Estructura de la memoria

En este apartado se describen los diferentes capítulos de este documento. La memoria contará con nueve capítulos con sus respectivas secciones, un apartado de bibliografía y un apéndice final. A continuación se realizará una breve descripción del contenido de cada uno de los apartados de la memoria.

- **Capítulo 1 Introducción:** En este primer capítulo, se expondrán las razones que han motivado la elección del proyecto como Trabajo de Fin de Grado, junto a la importancia de la evolución de las nuevas tecnologías. Además se explicarán los objetivos que se desean cumplir y el alcance del proyecto.
- **Capítulo 2 Estado del arte:** En este capítulo se pretende detallar el estudio realizado sobre aplicaciones similares a la propuesta. Para ello, en primer lugar se describirán las funciones de las aplicaciones seleccionadas para el estudio y para finalizar, se realizará una comparación de las funcionalidades que ofrecen estas aplicaciones frente a la desarrollada en el proyecto.
- **Capítulo 3 Planificación y presupuesto:** En este tercer capítulo se explicará con detalle la metodología utilizada para la organización y el desarrollo del proyecto, detallando posteriormente la planificación temporal llevada a cabo. Para finalizar esta sección, se detallará el presupuesto de la aplicación y en último lugar, los costes temporales y reales del proyecto, junto a la comparativa con la estimación realizada previamente.
- **Capítulo 4 Análisis:** En este capítulo, se muestra el análisis previo a la implementación de los sistemas (aplicación web y aplicación móvil), separando

las especificaciones de los ambos, siempre que sea necesario, debido a la existencia de diferencias entre ellos.

- **Capítulo 5 Diseño:** En este capítulo se pretende exponer el trabajo de diseño realizado previo a la creación de las aplicaciones. Se tratarán tanto las arquitecturas lógicas y físicas de las aplicaciones, como los diseños de algunas pantallas de la interfaz de usuario de la app desarrollada.
- **Capítulo 6 Implementación:** En este capítulo se procede a explicar el desarrollo llevado a cabo a lo largo del proyecto. En primer lugar se definirán todas las herramientas utilizadas en la creación de la memoria y la implementación del plugin para WordPress y de la aplicación móvil.
- **Capítulo 7 Pruebas:** En este capítulo se expondrán las pruebas realizadas sobre el resultado de la implementación de los dos sistemas. En primer lugar se detallarán las pruebas de caja blanca, y para finalizar se analizarán las de caja negra.
- **Capítulo 8 Manuales:** En este capítulo se redactan tres guías orientativas para poder hacer uso de los sistemas desarrollados. En primer lugar se muestra un manual de administrador que facilitará la instalación y configuración de la aplicación web, posteriormente se detallará un manual para los usuarios de tipo gestor para que puedan aprender a manejar con soltura la aplicación web. Finalmente se ha añadido un manual de usuario para la aplicación móvil desarrollada.
- **Capítulo 9 Conclusiones:** En este capítulo se realiza una revisión del trabajo realizado, analizando los resultados y el aprendizaje llevado a cabo. Finalmente se explica lo que se pretende alcanzar en el futuro tras la finalización de este proyecto académico.
- **Bibliografía:** En este apartado se recogen todas las referencias del contenido que se ha consultado para poder llevar a cabo el proyecto.
- **Apéndice A – Desarrollo detallado del análisis:** Contiene el resto de la especificación de casos de uso que no han sido tratados en el capítulo de análisis, junto a un listado de requisitos funcionales.

1.4. Contenido del DVD

El DVD adjunto a esta memoria contiene los siguientes ficheros:

- **Datos Fotográficos:** Carpeta que contiene todas las fotografías de las paradas de autobuses y las de los horarios de los que se ha extraído la información.
- **SegoviaEnBus:** Carpeta del proyecto Android con la aplicación móvil.
- **SegoviaEnBus.apk:** Instalador de la aplicación para dispositivos móviles.
- **WordPress:** Carpeta con la plantilla utilizada en la aplicación web desarrollada.
- **GestionAutobuses-JulenBarrio.zip:** Plugin desarrollado para la creación de la aplicación web.
- **Memoria_JulenBarrio.pdf:** Documento pdf con la memoria del proyecto.

CAPÍTULO 2

ESTADO DEL ARTE

En este capítulo se pretende detallar el estudio realizado sobre aplicaciones similares a la propuesta. Para ello, en primer lugar se describirán las funciones de las aplicaciones seleccionadas para el estudio y para finalizar, se realizará una comparación de las funcionalidades que ofrecen estas aplicaciones frente a la desarrollada en el proyecto.

2.1. Antecedentes

En la actualidad existen multitud de aplicaciones móviles para complementar los servicios de autobuses urbanos en las ciudades. Muchas de estas apps son creadas por particulares, aunque la gran mayoría, son ofrecidas de forma oficial por parte de los ayuntamientos o de las empresas concesionarias del servicio.

Antes de comenzar a desarrollar la aplicación se han examinado varias de las ya existentes utilizadas en algunas ciudades de España. A continuación, se mostrará un análisis detallado de ellas junto a la ya existente en Segovia.

2.1.1. Bus Segovia – Autobuses urbanos de Segovia

Aplicación no oficial de los autobuses urbanos de Segovia. Como podemos observar en la primera captura de la *Ilustración 5*, al iniciar la aplicación se muestran unos botones con las diferentes líneas. Al seleccionar uno de ellos se accede a los horarios pudiendo elegir el tipo de día; laboral, sábado o festivo (Captura 2). Finalmente se ofrece la opción de visualizar un listado con las paradas en función de la de origen (Captura 3).

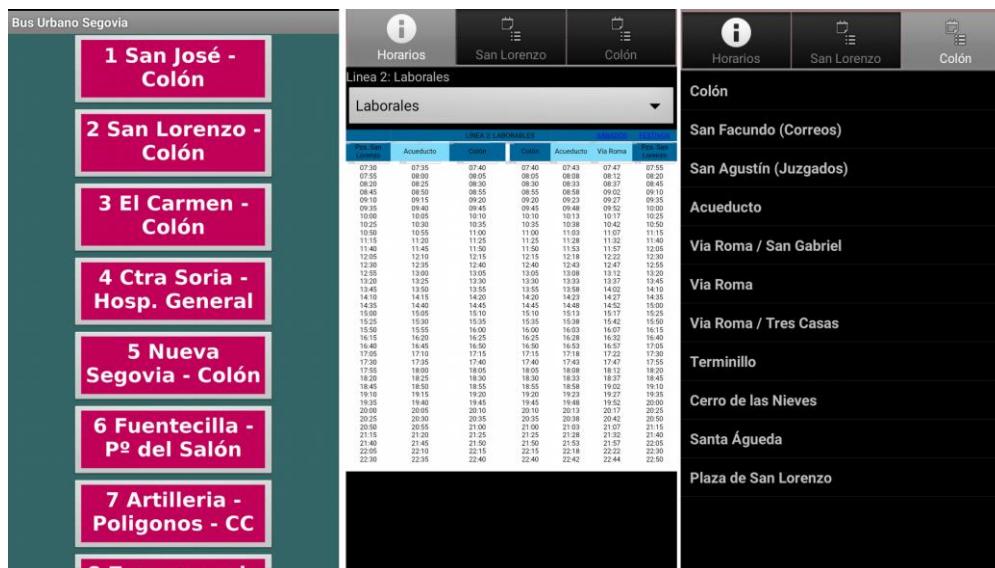


Ilustración 5: Capturas app autobuses urbanos de Segovia

2.1.2. BMS – Bus metropolitano de Salamanca

Aplicación utilizada para obtener información sobre los autobuses interurbanos de Salamanca. En su pantalla principal (ver Captura 1 *Ilustración 6*) podemos observar las opciones de “Líneas”, “Mis Paradas” y “Avisos” junto a la fecha y hora actual. Además, contamos con dos iconos en la parte superior, uno para acceder a las tarifas y otro para visualizar la información acerca de la aplicación. En el caso de seleccionar el icono de

tarifas observaremos los diferentes tipos de billete y abonos existentes. Además, cuentan con un enlace a los puntos de venta y recarga.

Como podemos ver en la segunda captura, con la opción de “Líneas” accedemos a un listado de las diferentes líneas del servicio. Al seleccionar una de ellas observamos un cuadro de texto con todas las paradas y bajo este tres botones, “Horarios”, “Paradas” y “Mapa”. En “Horarios” vemos las tablas con las horas de los servicios separadas por días (Captura 3). En el apartado de “Paradas” vemos un listado con todas ellas junto a una estrella para poder registrar las paradas deseadas como favoritas. Al seleccionar una parada obtendremos la correspondencia con otras líneas y un botón para enviar sugerencias de mejora de los alrededores. Finalmente, en el apartado de “Mapa” podemos observar un plano con la ruta que recorre la línea.

En la cuarta captura podemos ver la sección “Mis paradas”, en ella se muestra un listado seleccionable con las paradas establecidas previamente como favoritas. De este modo el usuario puede acceder de una forma más sencilla a las paradas que frecuente.

Para finalizar con los botones de la pantalla de inicio tenemos la opción de “Avisos”, en ella nos encontraremos con archivos pdf, con su respectiva fecha de subida, sobre notificaciones de modificaciones de servicio.



Ilustración 6: Capturas Bus Metropolitano de Salamanca app

2.1.3. Auvasa Bus – Autobuses urbanos Valladolid

Aplicación de los autobuses urbanos de Valladolid. Cuando accedemos a ella podemos obtener los tiempos de espera de una parada determinada de tres modos (Captura 1 Ilustración 7); mediante la búsqueda de paradas cercanas. En la cabecera de la aplicación junto al logotipo tenemos a la derecha el icono de una casa, éste será utilizado para redirigirnos a la pantalla de inicio cuando lo deseemos, y a la izquierda un icono para desplegar el menú de la aplicación.

En la segunda captura podemos observar el menú desplegable de la app. En él disponemos de las opciones de “Inicio”, “Noticias”, “Líneas”, “Medios de pago”, “Bus Turístico”, “Información”, “Vallabici” y en la parte inferior tenemos información de contacto.

Con la opción de “Noticias” observamos las notificaciones sobre incidencias en el servicio de autobuses junto a la fecha de las mismas.

En “Líneas” tenemos varias subsecciones en las que están repartidos todos los tipos de líneas con las que cuenta el servicio: ordinarias, búho, matinales, polígonos, fútbol y otros servicios. Dentro de cada línea se pueden observar las diferentes paradas ordenadas según el trayecto (Captura 3). En los detalles de las paradas se ve la información de cuándo llegarán los próximos autobuses, indicando el número de línea, el sentido y el tiempo restante para la llegada. Además, se proporcionan las opciones de actualizar los datos, añadir la parada a favoritos o visualizar el mapa (Captura 4).

En la opción de “Medios de pago”, hay cuatro opciones: “tarjetas de transporte”, “tarifas vigentes”, “puntos de recarga” y “consulta de saldo”. En las tarjetas de transporte se informa sobre las diferentes formas de pagar (billete ordinario o abonos). En la sección de tarifas se muestran los precios para los diferentes medios de pago o tipo de servicios. Con la opción de “puntos de recarga” observamos un mapa con los establecimientos en los que recargar el abono de transporte junto con un listado de los mismos. Finalmente, en “consulta de saldo” podemos ver cuánto nos queda en la tarjeta de transporte tras introducir su número de identificación.

En el apartado de “Bus Turístico” se muestra información sobre el vehículo utilizado, horarios de salida y tarifas.

En la sección de “Información” se muestran consejos de interés, datos de contacto con AUVASA (empresa concesionaria del servicio de autobuses urbanos de Valladolid) y preguntas frecuentes de los usuarios.

En la última opción, “Vallabici”, se informa sobre el servicio de préstamo de bicicletas del Ayuntamiento de Valladolid y un enlace para conocer más acerca de este servicio.

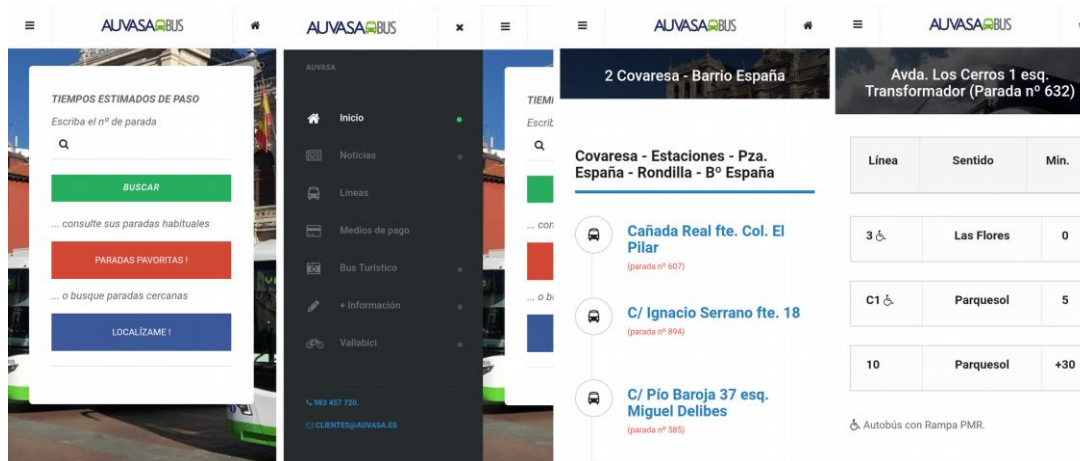


Ilustración 7: Capturas AUVASA Bus

2.1.4. Bus Madrid – Autobuses urbanos de Madrid

Aplicación usada obtener información sobre los autobuses urbanos de Madrid. Al iniciarla por primera vez (Captura 1 *Ilustración 8*) observamos un listado de todas las líneas del servicio (urbanas, nocturnas y especiales). La página de inicio se puede modificar en las opciones de la aplicación para que muestra el contenido que deseamos cada vez que iniciemos la app.

Si seleccionamos una de las líneas debemos elegir el sentido de ésta. Una vez completada la acción se mostrará un listado con las paradas y las opciones de hacerlas favoritas, ver el mapa o esquema de la línea, o ver una información detallada de los próximos autobuses en llegar a la parada (una vez seleccionada). En ella podremos ver el número de línea de los autobuses, el destino, los minutos restantes y los metros hasta la parada (Captura 3).

Como podemos observar en la segunda captura, la aplicación cuenta con un menú lateral desplegable. En él encontramos las opciones de “Paradas favoritas”, para poder ver las paradas guardadas con anterioridad; “Mapa”, para detectar la ubicación del usuario y buscar paradas cercanas, e “Incidencias”, para mostrar los avisos sobre alteraciones en alguna línea del servicio (Captura 4).



Ilustración 8: Capturas Bus Madrid

2.1.5. Avanza Zaragoza – Autobuses urbanos de Zaragoza

Aplicación de los autobuses urbanos de Zaragoza. En la pantalla principal (Captura 1 *Ilustración 9*) se muestran dos opciones, visualizar tiempo de llegada (ver información de parada) y ver cómo llegar a un destino desde un punto de origen (Capturas 2 y 3).

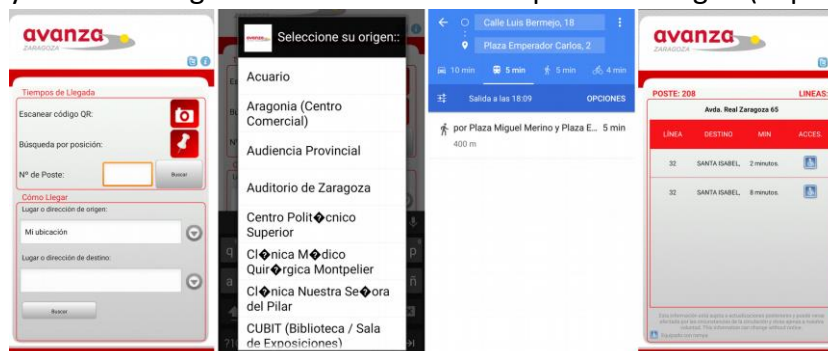


Ilustración 9: Capturas Avanza Zaragoza

Como podemos ver en la página de inicio, para buscar paradas se puede introducir el número de parada, escanear un código QR situado en la parada o buscar la más cercana a partir de la ubicación del usuario. Al visualizar la parada veremos el número de línea, el destino y el tiempo restante (Captura 4).

2.1.6. GranadaBus – Autobuses urbanos de Granada

Aplicación del servicio de autobuses urbanos de Granada. Cuando iniciamos la aplicación (Captura 1 *Ilustración 10*) observamos dos apartados; las paradas recientemente consultadas y las noticias con las notificaciones de cambios en el servicio; éstas son cargadas desde una cuenta de Twitter.

En la segunda captura tenemos el menú desplegable de la aplicación con las opciones de “Portada”, “Paradas”, “Líneas”, “Favoritos” y Noticias”.

En el apartado de “Paradas” (Captura 3) se muestra un mapa de Granada con la localización de las diferentes paradas de autobús y la opción de paradas cercanas para situarnos en el plano.

En la sección de “Líneas” se observa un listado de las mismas. Al seleccionar una de ellas se cargará de nuevo el mapa con las paradas correspondientes. Al pulsar sobre una de las paradas del mapa se mostrará su nombre, en caso de volver a pulsarlo, podremos visualizar su información junto a la opción de añadir a favoritos (Captura 4).

En la opción de “Favoritos” se cargan las paradas previamente añadidas. Además, se ofrece la opción de añadir nuevas paradas buscándolas por su nombre.

Para finalizar, en la sección de “Noticias” veremos los avisos previamente mencionados en la descripción de la pantalla principal.

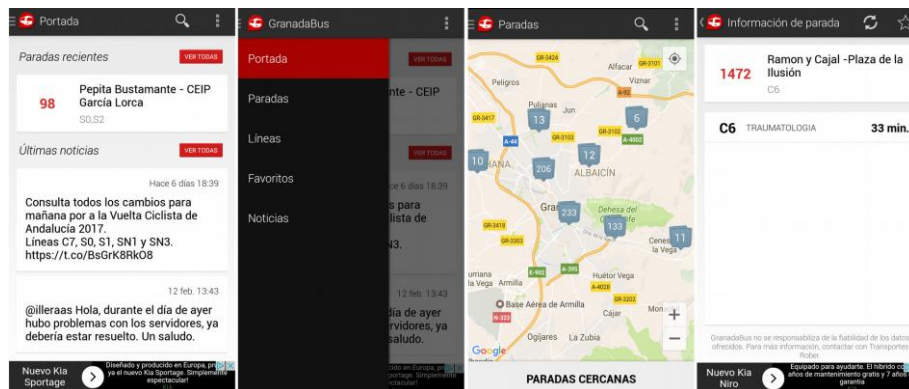


Ilustración 10: Capturas GranadaBus

2.2. Comparación del análisis con la propuesta

	Salamanca	Valladolid	Madrid	Zaragoza	Granada	Segovia	Propuesta
Ver líneas	✓	✓	✓		✓	✓	✓
Ver paradas de una línea	✓	✓	✓		✓	✓	✓
Búsqueda de paradas		✓		✓	✓		✓
Búsqueda de paradas cercanas		✓		✓	✓		✓
Lectura código Qr				✓			5
Ver horarios de la parada	✓			✓		✓	✓
Tiempos estimados de llegada		✓	✓		✓		✓
Metros hasta la parada			✓				
Paradas favoritas	✓	✓	✓		✓		✓
Correspondencia con otras líneas	✓						✓
Sugerencias para la zona	✓						
Visualizar plano de la ruta	✓		✓				✓
Visualizar ruta en Google Maps			✓		✓		✓
Calcular recorrido				✓			5
Notificaciones	✓	✓	✓		✓		✓
Integración con Twitter (Notificaciones)					✓		5
Visualizar tarifas	✓	✓					✓
Información de abonos	✓	✓					✓
Consulta de saldo		✓					
Preguntas frecuentes		✓					
Servicio de préstamos de bicicletas		✓					5
Bus Turístico		✓					✓
Paradas de interés turístico							✓
Selección de idioma							✓

Tabla 2: Comparación del análisis con la APP propuesta

⁵ Funcionalidades previstas para futuras versiones de la app

Como podemos comprobar en la *Tabla 2*, la aplicación propuesta cumple prácticamente todas las funcionalidades observadas en el análisis de aplicaciones similares. Sin embargo, ninguna de las examinadas cuenta con funciones enfocadas al turismo.

De este modo, la aplicación de Urbanos de Segovia contara con facilidades para los turistas, tales como información sobre paradas de interés turístico para ayudarlos a llegar mediante el uso del servicio público de autobuses. Además, se podrá seleccionar el idioma deseado de entre los disponibles para traducir la aplicación.

Respecto a la aplicación existente en Segovia actualmente se puede observar que hay un gran número de nuevas funcionalidades. Estas harán la aplicación más atractiva para los usuarios y les proporcionara una mayor información actualizada del servicio.

CAPÍTULO 3

PLANIFICACIÓN Y PRESUPUESTO

En este tercer capítulo se explicará con detalle la metodología utilizada para la organización y el desarrollo del proyecto, detallando posteriormente la planificación temporal llevada a cabo. Para finalizar esta sección, se detallará el presupuesto de la aplicación y en último lugar, los costes temporales y reales del proyecto, junto a la comparativa con la estimación realizada previamente.

3.1. Metodología de Trabajo

Al principio el objetivo del proyecto era la realización de una aplicación móvil, contando con la colaboración del Ayuntamiento de Segovia a la hora de obtener acceso a una base de datos, con toda la información necesaria para su desarrollo. Sin embargo, tras ponernos en contacto no recibimos respuesta alguna, por lo que la complejidad del proyecto aumentó debido a la necesidad de obtener esa información por nuestros medios. Para ello, decidimos crear una aplicación web, cuyo objetivo es el de la gestión y mantenimiento de una base de datos. Debido a este suceso, se ha tenido que buscar la metodología más adecuada para organizar el trabajo, y así, poder cumplir con los plazos de finalización previstos.

Finalmente se optó por una metodología de trabajo ágil, basada en un sistema de iteraciones incrementales con sprints mensuales del proyecto, con el fin de consolidar los avances realizados y mejorar el contenido en los siguientes ciclos.

En cuanto al trabajo que se debe llevar a cabo, diferenciamos las tareas en cinco fases distintas:

Fase de Análisis

En esta fase se pretende realizar un estudio detallado de todos los requisitos necesarios para poder desarrollar correctamente el proyecto. Para ello se deben realizar las siguientes tareas:

- Identificación de actores. Se deben analizar todos los tipos de usuarios que interactuarán con los sistemas a desarrollar.
- Requisitos de usuario. Con ello se pretende identificar todas las acciones que los usuarios podrán llevar a cabo dentro de cada aplicación.
- Requisitos funcionales. Son todas las funcionalidades que los sistemas pueden realizar.
- Requisitos de información. Contemplan todos los datos necesarios para el funcionamiento de los sistemas, junto con la estructura que deben tener las bases de datos de los sistemas.

Fase de Diseño

Para esta fase se pretende plantear el diseño de las aplicaciones, tanto a nivel de arquitectura, como de funcionalidad y estilo. De este modo, se diseñarán las

arquitecturas lógicas y físicas de las aplicaciones, las relaciones entre las diferentes clases de los sistemas, y el diseño de la interfaz con la que interactuarán los usuarios.

Fase de Implementación

En esta etapa el objetivo es el desarrollo de las aplicaciones a partir de los datos evaluados en las dos fases anteriores.

Fase de Pruebas

Tras desarrollar las aplicaciones, deben ser probadas con el objetivo de detectar y solucionar fallos de implementación.

Fase de Documentación

En esta fase, se pretende redactar todos los avances llevados a cabo tras la realización de cada una de las anteriores fases del proyecto.

3.2. Planificación Temporal

Como se ha mencionado en el anterior apartado, el proyecto se realizará a través de unas iteraciones incrementales de carácter mensual, con sus respectivas reuniones al final de cada ciclo con el tutor del proyecto.

En la *Ilustración 11* (complementada con el diagrama de Gantt – *Ilustración 10*) podemos observar la planificación temporal llevada a cabo antes de comenzar el proyecto (imágenes obtenidas del anteproyecto del Trabajo de Fin de Grado). Como podemos observar, el objetivo temporal era la realización del proyecto desde el día 31 de octubre de 2017 y con finalización el 22 de mayo de 2017. Por lo tanto contaremos con siete iteraciones.

En estas iteraciones no se ven reflejadas las cinco fases descritas en el apartado anterior. Esto se debe a que en las de “Desarrollo” se incluyen las fases de implementación y pruebas, ya que tras desarrollar una funcionalidad de un sistema, se debe probar inmediatamente para evitar acumular fallos que afecten al resto del trabajo pendiente por realizar.

También podemos apreciar que en las primeras iteraciones, las fases de análisis y diseño ocupan la mayor parte del tiempo, y es a medida que avanza el proyecto, cuando éstas se ven reducidas y empieza a cobrar mayor importancia el desarrollo. Respecto a la documentación, se debe realizar tras la finalización de cada una de las fases, y en la última etapa, darle especial importancia para ultimar los detalles de esta memoria.

Urbanos de Segovia	23 days?	31/10/16 8:00	02/06/17 17:00
Primera Iteración	23 days?	31/10/16 8:00	30/11/16 17:00
Análisis	8 days?	31/10/16 8:00	9/11/16 17:00
Diseño	8 days?	9/11/16 8:00	18/11/16 17:00
Desarrollo	5 days?	21/11/16 8:00	25/11/16 17:00
Documentación	3 days?	28/11/16 8:00	30/11/16 17:00
Segunda Iteración	21 days?	1/12/16 8:00	29/12/16 17:00
Análisis	7 days?	1/12/16 8:00	9/12/16 17:00
Diseño	8 days?	12/12/16 8:00	21/12/16 17:00
Desarrollo	5 days?	22/12/16 8:00	28/12/16 17:00
Documentación	2 days?	28/12/16 8:00	29/12/16 17:00
Tercera Iteración	21 days?	2/01/17 8:00	30/01/17 17:00
Análisis	7 days?	2/01/17 8:00	10/01/17 17:00
Diseño	8 days?	10/01/17 8:00	19/01/17 17:00
Desarrollo	6 days?	20/01/17 8:00	27/01/17 17:00
Documentación	2 days?	27/01/17 8:00	30/01/17 17:00
Cuarta Iteración	20 days?	1/02/17 8:00	28/02/17 17:00
Análisis	7 days?	1/02/17 8:00	9/02/17 17:00
Diseño	7 days?	9/02/17 8:00	17/02/17 17:00
Desarrollo	6 days?	17/02/17 8:00	24/02/17 17:00
Documentación	3 days?	24/02/17 8:00	28/02/17 17:00
Quinta Iteración	23 days?	1/03/17 8:00	31/03/17 17:00
Análisis	3 days?	1/03/17 8:00	3/03/17 17:00
Diseño	4 days?	3/03/17 8:00	8/03/17 17:00
Desarrollo	12 days?	9/03/17 8:00	24/03/17 17:00
Documentación	5 days?	27/03/17 7:00	31/03/17 17:00
Sexta Iteración	20 days?	3/04/17 8:00	28/04/17 17:00
Análisis	3 days?	3/04/17 8:00	5/04/17 17:00
Diseño	4 days?	5/04/17 8:00	10/04/17 17:00
Desarrollo	10 days?	11/04/17 8:00	24/04/17 17:00
Documentación	4 days?	25/04/17 8:00	28/04/17 17:00
Séptima Iteración	25 days?	1/05/17 8:00	2/06/17 17:00
Análisis	3 days?	1/05/17 8:00	3/05/17 17:00
Diseño	3 days?	3/05/17 8:00	5/05/17 17:00
Desarrollo	10 days?	8/05/17 8:00	19/05/17 17:00
Documentación	10 days?	22/05/17 8:00	2/06/17 17:00

Ilustración 11: Planificación temporal inicial

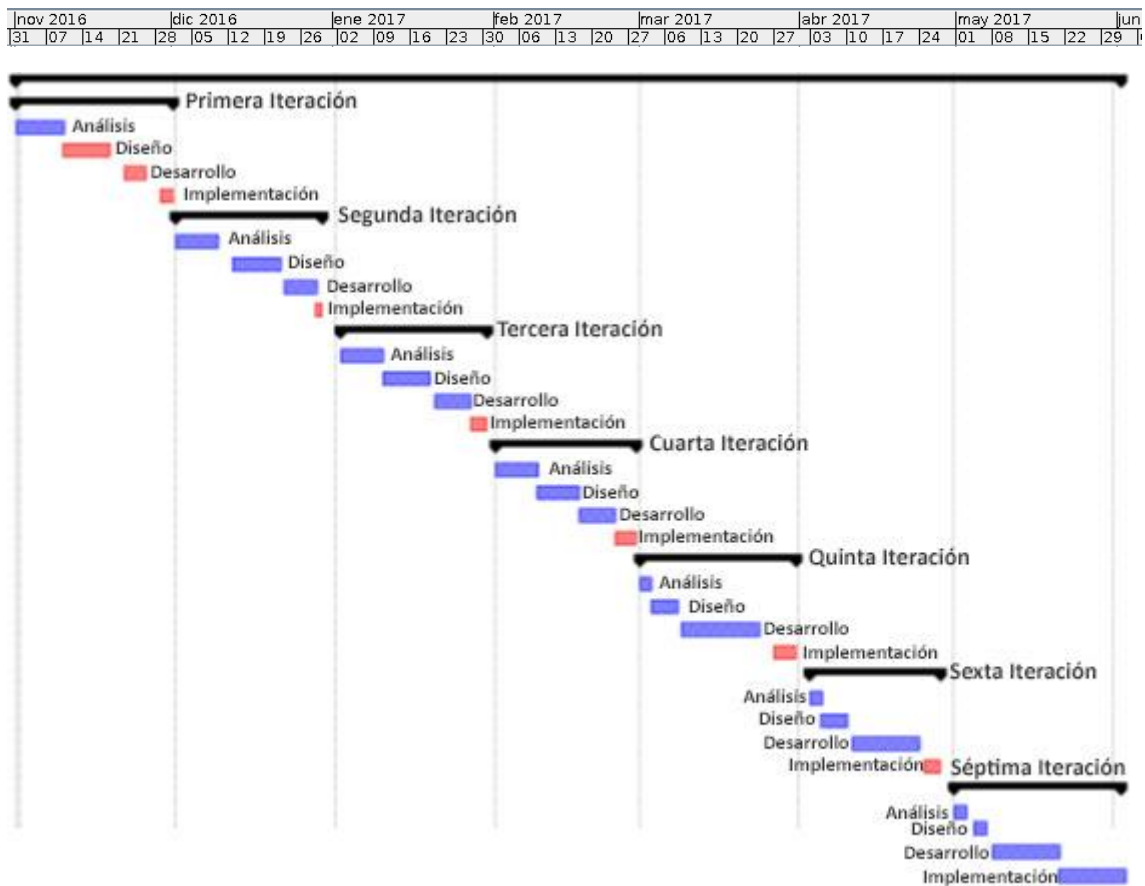


Ilustración 12: Diagrama Gantt Planificación Inicial

A continuación, se muestra una tabla (Tabla 3) con las horas invertidas durante cada día, divididas en cuatro periodos. El primer periodo se corresponde al primer cuatrimestre, teniendo que dividir las horas diarias entre las asignaturas que se estaban cursando y el desarrollo del proyecto. En segundo lugar, se encuentra el periodo de trabajo a tiempo completo establecido entre el primer periodo (finalización del primer cuatrimestre) y el tercer periodo. En esta tercera etapa, comienzan las prácticas curriculares en una empresa y por lo tanto se tiene que ver reducido el tiempo de desarrollo del proyecto. Para finalizar, en la cuarta etapa se vuelve a poder trabajar a tiempo completo, teniendo que dedicar solo una pequeña parte del tiempo a dos asignaturas.

PERIODO	DÍAS	HORAS/DÍAS	HORAS TOTALES
PRIMER PERIODO:	23	4	92
SEGUNDO PERIODO:	42	8	336
TERCER PERIODO:	63	5	315
CUARTO PERIODO:	25	8	200
			TOTAL: 943 HORAS

Tabla 3: Planificación horas dedicadas

3.3. Coste Temporal Real

ETAPA	DÍAS	HORAS/DÍAS	HORAS TOTALES
PRIMER PERIODO:	19	4	76
PRIMERA ITERACIÓN			
· ANÁLISIS DE OBJETIVOS DEL PROYECTO	5	4	20
· BÚSQUEDA Y TESTEO DE APLICACIONES SIMILARES	3	4	12
· DISEÑO APLICACIÓN MÓVIL	6	4	24
· DESARROLLO NAVEGACIÓN APLICACIÓN MÓVIL	5	4	20
SEGUNDO PERIODO:	42	8	336
· DOCUMENTACIÓN DE AVANCES REALIZADOS	3	8	24
SEGUNDA ITERACIÓN			
· CORRECCIONES REVISIÓN	3	8	21
· ANÁLISIS DE ESTADO DEL ARTE	3	8	24
· ANÁLISIS DE ENTIDADES DE LA BASE DE DATOS	5	8	40
· DISEÑO APLICACIÓN WEB	3	8	24
· DESARROLLO Y PRUEBAS NAVEGACIÓN APP	4	8	32
· DOCUMENTACIÓN DE AVANCES REALIZADOS	2	8	16
TERCERA ITERACIÓN			
· CORRECCIONES REVISIÓN	3	5	15
· ANÁLISIS DE ENTIDADES DE LA BASE DE DATOS	5	5	25
· ANÁLISIS REQUISITOS USUARIO APP	5	5	25
· CONFIGURACIÓN SERVIDOR WEB	3	5	15
· CONFIGURACIÓN APLICACIÓN WEB	3	5	15
· DOCUMENTACIÓN DE AVANCES REALIZADOS	2	5	10
TERCER PERIODO:	63	5	315
CUARTA ITERACIÓN			
· CORRECCIONES REVISIÓN	3	5	15
· ANÁLISIS DE ENTIDADES DE LA BASE DE DATOS	2	5	10
· ANÁLISIS REQUISITOS DE USUARIO WEB	6	5	30
· ANÁLISIS REQUISITOS DE USUARIO APP	3	5	15
· DISEÑO ESTRUCTURA WEB	3	5	15
· DOCUMENTACIÓN DE AVANCES REALIZADOS	3	5	15
QUINTA ITERACIÓN			
· CORRECCIONES REVISIÓN	2	5	10
· DESARROLLO ESTRUCTURA WEB	2	5	10
· DESARROLLO PLUGIN WEB	14	5	70
· OBTENCIÓN FOTOGRAFÍAS PARADAS Y HORARIOS	2	5	10
· DOCUMENTACIÓN DE AVANCES REALIZADOS	3	5	15
SEXTA ITERACIÓN			
· CORRECCIONES REVISIÓN	1	5	5
· ANÁLISIS BASE DE DATOS	2	5	10
· OBTENCIÓN FOTOGRAFÍAS PARADAS Y HORARIOS	1	5	5
· DESARROLLO Y PRUEBAS PLUGIN WEB	10	5	50
· EDICIÓN DE ICONOS Y LOGOTIPOS	3	5	15
· DOCUMENTACIÓN DE AVANCES REALIZADOS	3	5	15

Tabla 4: Tiempos de desarrollo del proyecto (1)

CUARTO PERIODO:	55	8	440
SÉPTIMA ITERACIÓN			
· CORRECCIONES REVISIÓN	1	8	8
· TRABAJAR ESTILOS APLICACIÓN WEB	2	8	16
· DESARROLLO Y PRUEBAS PLUGIN WEB	7	8	56
· CONEXIÓN APLICACIÓN WEB CON APP	2	8	16
· EDICIÓN IMÁGENES DE PARADAS	1	8	8
· OBTENCIÓN HORARIOS DE FOTOGRAFÍAS	5	8	40
· INSERCIÓN DE DATOS	4	8	32
· DOCUMENTACIÓN DE AVANCES REALIZADOS	3	8	24
OCTAVA ITERACIÓN			
· CORRECCIÓN REVISIÓN	3	8	24
· PRUEBAS APLICACIÓN WEB	2	8	16
· INSERCIÓN DE DATOS	4	8	32
· DESARROLLO APLICACIÓN MÓVIL	8	8	64
· DOCUMENTACIÓN DE AVANCES REALIZADOS	3	8	24
NOVENA ITERACIÓN			
· CORRECCIÓN REVISIÓN	5	8	40
· REVISIÓN FINAL + IMPRESIÓN PROYECTO	5	8	40
			TOTAL: 1167 HORAS

Tabla 5: Tiempos de desarrollo del proyecto (2)

Como se menciona en el apartado de metodología, el alcance del proyecto se vio modificado durante el transcurso del mismo, ampliándose la carga de trabajo a realizar, al tener que desarrollar una aplicación web utilizada para el mantenimiento de la aplicación móvil. Esto hizo que se que llevase a cabo un desarrollo del proyecto diferente al previsto aumentando el cuarto periodo en un mes y medio más.

En las *Tablas 4 y 5* podemos observar los tiempos de desarrollo del proyecto contemplando el incremento de 30 días de trabajo en el cuarto periodo. Además se detallan con mayor precisión las diferentes tareas realizadas en cada uno de estos periodos.

El diagrama de Gantt obtenido tras el análisis de los tiempos reales del proyecto, lo podemos observar en la *Ilustración 13*.

PERIODO	DÍAS	HORAS/DÍAS	HORAS TOTALES
PRIMER PERIODO:	19	4	76
SEGUNDO PERIODO:	42	8	336
TERCER PERIODO:	63	5	315
CUARTO PERIODO:	55	8	440
			TOTAL: 1167 HORAS

Tabla 6: Tiempo real invertido

Para finalizar este apartado, se ha calculado el tiempo real invertido en el desarrollo del proyecto tras la adición de las dos nuevas etapas (*Tabla 6*).

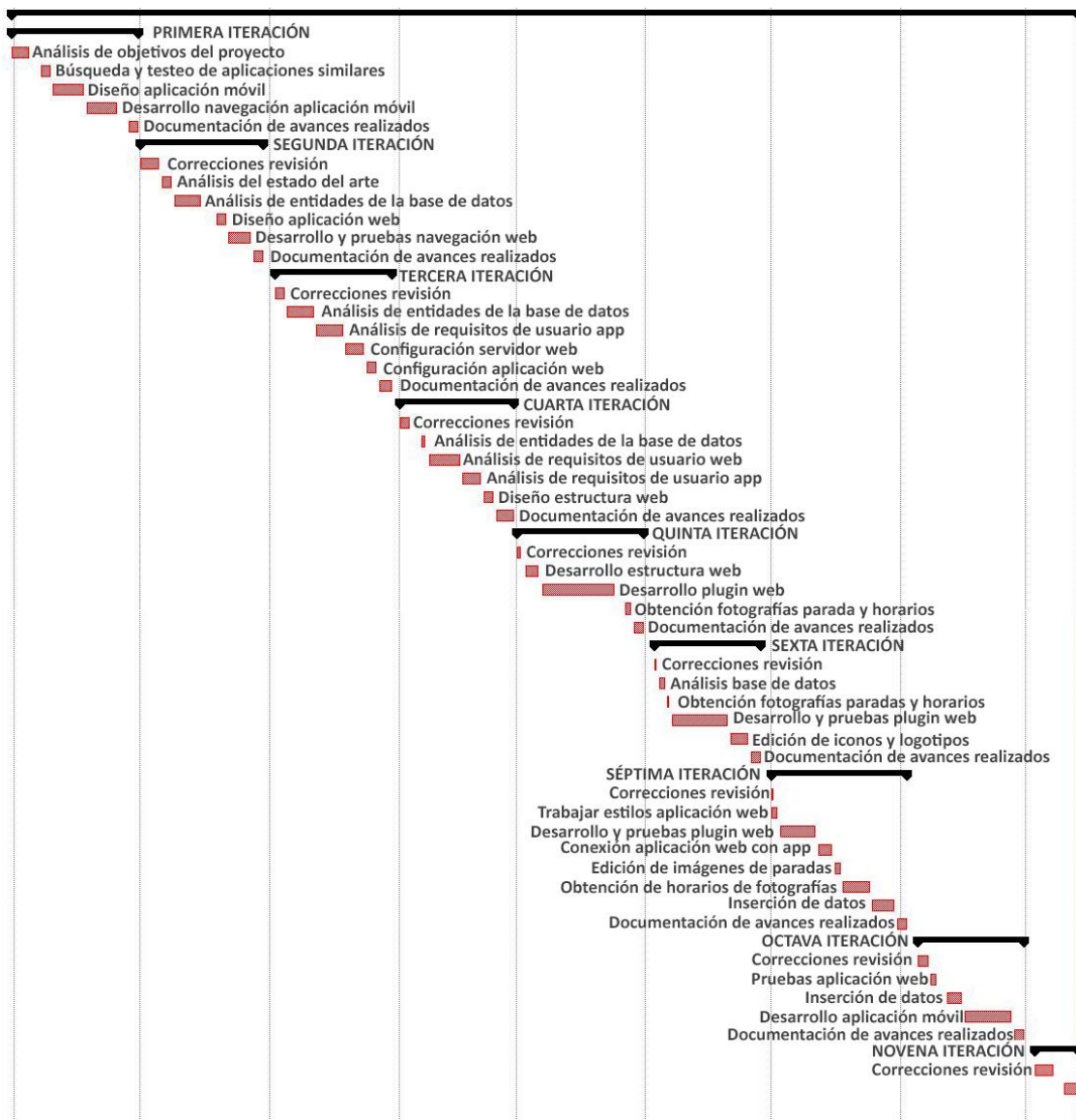


Ilustración 13: Diagrama de Gantt tiempo real invertido

3.3. Presupuesto

Para elaborar un presupuesto representativo primero se analizará la planificación temporal y se calcularán los costes en función de los tiempos previstos, posteriormente se utilizarán tanto el método de puntos de función como el COCOMO II. De este modo se podrá elegir la opción más realista y que mejor represente la relación entre la carga de trabajo realizada en el desarrollo del proyecto y el coste del trabajo.

Antes de evaluar los costes del desarrollo de las dos aplicaciones (web y móvil), debemos analizar los componentes hardware y software utilizados para el desarrollo del proyecto. También se debe realizar un análisis de costes de personal en función de los roles llevados a cabo.

3.3.1. Componentes Hardware

En la *Tabla 7* se muestran todos los componentes Hardware que han sido utilizados para el desarrollo del proyecto. Junto a éstos, se muestra un valor que representa el porcentaje de uso del componente frente a la estimación de la vida útil del mismo. De este modo, se calculará el coste de uso del componente durante el periodo de duración del proyecto.

3.3.2. Componentes Software

En la *Tabla 8* se pueden observar los costes de los programas utilizados para la implementación de las aplicaciones creadas y para la creación de la memoria del Trabajo de Fin de Grado. Los valores de los costes se calcularán del mismo modo que con los componentes hardware; evaluando su coste total en función del tiempo de uso y de vida. Las herramientas software utilizadas se detallarán en el apartado de implementación.

ELEMENTO	COSTE TOTAL	VIDA ÚTIL	%DE USO	COSTE REAL
ORDENADOR PORTÁTIL	600€	4 años	25%	150€
DISCO SSD 256GB	60€	7 años	14%	8,5€
APLIACIÓN RAM 4GB	30€	4 años	25%	7,5€
CONEXIÓN A INTERNET	50€/mes	9 meses	35%	157,5€
SERVIDOR WEB (UVA)	0€			0€
			Total:	616€

Tabla 7: Costes componentes hardware

ELEMENTO	COSTE TOTAL	VIDA ÚTIL	%DE USO	COSTE REAL
WINDOWS 8	130€	2 años	50%	65€
DRAW.IO	0€			0€
STARUML	0€			0€
PHOTOSHOP	24.19€/mes	3 meses		72.57€
MICROSOFT WORD	0€			0€
PLANTILLA WORDPRESS	58€	12 meses	100%	58€
FILEZILLA	0€			0€
ORACLE VM VIRTUALBOX	0€			0€
SUBLIME TEXT 2	0€			0€
ANDROID STUDIO	0€			0€
			Total:	137.57€

Tabla 8: Costes componentes software

3.3.3. Costes de personal

En el transcurso del proyecto se han desempeñado dos tipos de roles: el de analista de software y el de programador de aplicaciones. En la *Tabla 9* podemos observar el coste presupuestado en función de las horas trabajadas por cada rol en la planificación inicial. Para el cálculo de las horas, se han asignado al analista las fases de análisis, diseño y documentación. En el caso del programador, tan solo tiene asignadas las fases de desarrollo, compuestas por implementación y pruebas.

El sueldo se calcula a partir del sueldo medio mensual de los roles, obteniendo el coste por hora y teniendo en cuenta que se trabaja 8 horas diarias en 20 días laborales al mes.

ROL	SUELDO / MES	SUELDO / HORA	HORAS	TOTAL
ANALISTA SOFTWARE ⁶	2209€	13,8 €/hora	615	9.328,8€
PROGRAMADOR DE APLICACIONES ⁷	1819€	11,4 €/hora	328	3.739,2€
			Total:	12.226,2€

Tabla 9: Costes de personal estimados

3.3.4. Método de puntos de función

El método consiste en realizar una estimación del coste del proyecto software evaluando todas sus funciones, para obtener los puntos de función de cada una en función de la complejidad a la hora de implementarla.

Para evaluar los puntos de función de la aplicación se establecen cinco grupos. En estos grupos se repartirán los diferentes componentes a implementar y en función de cada grupo, se le asignará una complejidad u otra. Los grupos son los siguientes:

- Entradas de usuario: Funcionalidades en las que el usuario tendrá que introducir datos mediante el uso de formularios.
- Salidas de usuario: Funcionalidades que muestran datos al usuario.
- Consultas externas: Peticiones del usuario a un sistema externo que interactúa con la aplicación.
- Ficheros lógicos internos: Bases de datos internas de la aplicación.
- Ficheros lógicos externos: Bases de datos externas a las que se conecte la aplicación.

Una vez repartidos los componentes en estos grupos, se utilizará la siguiente tabla proporcionada por el método para asignar la complejidad adecuada a cada uno de ellos.

Ficheros lógicos externos e internos				Salidas y consultas				Entradas			
Número de entidades	Atributos			Número de ficheros	Atributos			Número de ficheros	Atributos		
	1-19	20-50	>51		1-5	6-19	>20		1-4	5-15	>16
1	Baja	Baja	Media	0-1	Baja	Baja	Media	0-1	Baja	Baja	Media
2-5	Baja	Media	Alta	2-3	Baja	Media	Alta	2-3	Baja	Media	Alta
≥6	Media	Alta	Alta	≥4	Media	Alta	Alta	≥4	Media	Alta	Alta

Tabla 10: Complejidades de las componentes

⁶ Sueldo analista - <http://www.tusalario.es/main/salario/comparatusalario?job-id=2511010000000#/>

⁷ Sueldo programador de aplicaciones - <http://www.tusalario.es/main/salario/comparatusalario?job-id=2514010000000#/>

GRUPO	COMPONENTE	COMPLEJIDAD
ENTRADAS	Selección de Idioma	Baja
	SALIDAS	Visualizar Líneas
	Visualizar Recorridos	Media
	Visualizar Mapa Línea	Media
	Visualización Parada	Alta
	Visualización Mapa Parada	Baja
	Visualización Horarios	Alta
	Búsqueda de paradas	Baja
	Paradas Favoritas	Baja
	Visualización Parada I. Turístico	Baja
	Inf. Bus Turístico	Baja
	Inf. Tarifas	Baja
	Inf. Abonos	Baja
	Visualización Notificaciones	Media
	Geolocalización Usuario	Baja
CONSULTAS	Obtención JSON externos	Alta
	API Google Maps	Alta
FICHEROS INTERNOS	Base de Datos app	Alta
FICHEROS EXTERNOS	Base de Datos Ayuntamiento	Alta

Tabla 11: Complejidad de los componentes

En la *Tabla 11* se muestra la división de los componentes entre los diferentes grupos, y su correspondiente complejidad usando la *Tabla 10* para el cálculo de la misma.

GRUPO	ENTRADAS			SALIDAS			CONSULTAS			F. INTERNOS			F. EXTERNOS			
	B	M	A	B	M	A	B	M	A	B	M	A	B	M	A	
COMPLEJIDAD	3	4	6	3	4	6	4	5	7	7	10	15	5	7	10	
FACTOR	1	0	0	9	3	2	0	0	2	0	0	1	0	0	1	
NÚMERO	3	0	0	27	12	12	0	0	14	0	0	15	0	0	10	
TOTAL																= 93

Tabla 12: Cálculo de puntos de función sin ajustar

Una vez obtenidos los puntos de función, debemos realizar un ajuste con un coeficiente obtenido tras la asignación de un valor entero entre 0 y 5 a una serie de variables, en función de la complejidad de éstas.

FACTORES DE AJUSTE	VALOR ASIGNADO
1. COMUNICACIÓN DE DATOS	4
2. FUNCIONES DISTRIBUIDAS	2
3. PRESTACIONES	4
4. CONDICIONES DE ALTA EXIGENCIA	2
5. FRECUENCIA DE TRANSACCIONES	2
6. ENTRADA ONLINE DE DATOS	1
7. DISEÑO OPTIMIZADO PARA EL USUARIO FINAL	3
8. ACTUALIZACIÓN DE DATOS ONLINE	4
9. COMPLEJIDAD DE LOS PROCESOS DEL SISTEMA	4
10. REUSABILIDAD DEL CÓDIGO	1
11. FACILIDAD DE INSTALACIÓN	1
12. FACILIDAD DE OPERACIÓN	1
13. LOCALIZACIONES MÚLTIPLES	3
14. FACILIDAD DE INTRODUCCIÓN DE CAMBIOS	1
VALOR TOTAL:	33

Tabla 13: Cálculo de coeficientes para el factor de ajuste

Para calcular el factor de ajuste debemos utilizar la siguiente fórmula:

$$FA = 0,65 + 0,01 \cdot \text{total coeficientes} = 0,65 + 0,01 \cdot 33 = 0,98$$

Tras obtener el factor de ajuste debemos multiplicarlo por los puntos de función obtenidos para obtener los puntos de función ajustados

$$PFA = PF \cdot FA = 93 \cdot 0,98 = 91,14 PF$$

Finalmente debemos realizar el cálculo de la estimación de tiempo. Para ello se asume la equivalencia de que un punto de función es igual a cuatro horas de trabajo y que tras la planificación se realiza un cálculo de la media de horas trabajadas por cada día, obteniendo 6,16 horas/día, por lo tanto equivale a que un día de trabajo se corresponde con 1,54 Puntos de Función

$$tiempo = 91,14 PF \cdot (1 \text{ día} / 1,54PF) = 59,19 \text{ días} = 365 \text{ horas}$$

Una vez obtenidas las horas estimadas para la realización del proyecto siguiendo el método de puntos de función, debemos repartirlas entre los diferentes roles para calcular el presupuesto final. Para ello se han utilizado las horas de la estimación tras la planificación inicial, para calcular el porcentaje asignado a cada rol. De este modo se obtiene que el analista utiliza un 65% del tiempo total, frente al 35% del programador.

ROL	SUELDO / MES	SUELDO / HORA	HORAS	TOTAL
ANALISTA SOFTWARE	2209 €	13,8 €/hora	237	3.270,6 €
PROGRAMADOR DE APLICACIONES	1819 €	11,4 €/hora	128	1.459,2 €
			Total:	4.729,8 €

Tabla 14: Presupuesto Puntos de Función

3.3.5. COCOMO II

El método del COCOMO II trata de realizar un presupuesto a partir de la estimación de las líneas de código a partir de los puntos de función previamente obtenidos.

Antes de proceder al cálculo de la estimación de las líneas de código, se debe elegir un modelo que se adapte al tipo de aplicación que se pretende desarrollar. Los modelos entre los que se puede elegir son los siguientes:

- Orgánico: Proyectos de pequeño tamaño y sin gran complejidad, desarrollados por un equipo pequeño en condiciones de escasa rigidez respecto a los requisitos.
- Semiacoplado: Son aquellos proyectos con una dificultad y tamaño intermedios, en los que varios equipos deben coordinarse para su desarrollo. En general, la rigidez de los requisitos es también baja.
- Integrado: Son proyectos con alta exigencia tanto técnica como formal, presentando una especificación de requisitos muy rígida y poco proclive a cambios.

Tras analizar las definiciones de los tipos de proyecto, se comprueba que nuestra aplicación se adapta de una forma clara al modelo semiacoplado. Ya que se trata de un proyecto de dificultad y tamaño intermedio y requiere de la coordinación del equipo de desarrollo y análisis con el Ayuntamiento de Segovia.

Una vez identificado el modelo procedemos a calcular la estimación de líneas de código. Para el lenguaje Java (lenguaje en el que se basa Android) se estiman unas 53 líneas de

código por cada punto de función. Por lo tanto el cálculo de líneas de código se corresponde con:

$$LDC = PF \cdot 53 = 93 \cdot 53 = 4929$$

Factores de coste	Valor de los factores						Factor
	Muy bajo	Bajo	Medio	Alto	Muy Alto	Extra	
Fiabilidad del software	0.75	0.88	1.00	1.15	1.40		1.15
Tamaño de la base de datos		0.94	1.00	1.08	1.16		1.08
Complejidad del software	0.70	0.85	1.00	1.15	1.30	1.65	1.15
Restricciones de rendimiento en runtime			1.00	1.06	1.21	1.66	1.00
Restricciones de memoria		0.87	1.00	1.15	1.30	1.56	1.00
Volatilidad del entorno de la máquina virtual		0.87	1.00	1.07	1.15		1.15
Tiempo de respuesta requerido	1.46	1.19	1.00	0.86	0.71		1.46
Experiencia con el tipo de aplicación	1.29	1.13	1.00	0.91	0.82		0.91
Experiencia con el hardware	1.21	1.10	1.00	0.90			0.90
Experiencia con el lenguaje de programación	1.14	1.07	1.00	0.95			1.00
Capacidad de los programadores	1.42	1.17	1.00	0.86	0.7		0.86
Técnicas modernas de programación	1.24	1.10	1.00	0.91	0.82		1.00
Utilización de herramientas software	1.24	1.10	1.00	0.91	0.83		1.00
Restricciones en la planificación temporal del desarrollo	1.23	1.08	1.00	1.04	1.10		1.04
Total:							1.757

Tabla 15: Factores de ajuste presupuesto COCOMO II

Ahora se deben tener en cuenta una serie de factores de ajuste obtenidos en función de las características del proyecto. Estos factores se calculan en la Tabla 15.

TIPO DE PROYECTO	A	B	C	D
ORGÁNICO	2.4	1.05	2.5	0.38
SEMIACOPLADO	3.0	1.12	2.5	0.35
INTEGRADO	3.6	1.20	2.5	0.32

Tabla 16: Modificadores para los modelos de proyecto

Utilizando todos los datos hasta ahora calculados en este apartado, procedemos al cálculo del esfuerzo utilizando la siguiente fórmula:

$$E = a \cdot (KLDC)^b \cdot \text{factor ajuste} = 3,0 \cdot (4,929)^{1,12} \cdot 1,757 = 26,95 \text{ personas} - \text{mes}$$

Este cálculo indica el número de personas necesarias para terminar el proyecto en un mes. Utilizando este dato, el tiempo necesario será de:

$$t = c \cdot E^d = 2,5 \cdot 29,95^{0,35} = 8,22 \text{ meses}$$

Utilizando los dos valores obtenidos, procedemos a calcular el número medio de personas necesarias para desarrollar el proyecto:

$$n = 26,95 / 8,22 = 3,28 \rightarrow 4 \text{ personas}$$

Como podemos observar en los cálculos, necesitaríamos 4 empleados en 6,74 meses (26,95/4) para realizar el proyecto. Como solo es realizado por una persona, se estima que se realizaría en aproximadamente 27 meses, lo que equivaldrían a 3326 horas.

Finalmente al igual que con el método de puntos de función, se realiza una división de las horas entre los dos roles de acuerdo a la planificación inicial realizada.

ROL	SUELDO / MES	SUELDO / HORA	HORAS	TOTAL
ANALISTA SOFTWARE	2209 €	13,8 €/hora	2162	29.835,6 €
PROGRAMADOR DE APLICACIONES	1819 €	11,4 €/hora	1164	13.269,6 €
			Total:	43.105,2 €

Tabla 17: Presupuesto COCOMO II

3.3.6. Comparativa de presupuestos

De los dos métodos utilizados para obtener un presupuesto, el que más se asemeja a la realidad es el de puntos de función (4.729,8 €), ya que el del COCOMO II (43.105,2 €) es excesivo para el tipo de proyecto que se realiza. Además al comparar el resultado obtenido mediante la evaluación de los puntos de función, con el realizado sobre la estimación de tiempos de desarrollo del proyecto (12.226,2€), se deduce que esta notable diferencia se debe a la realización de una mala planificación, calculando más horas de las debidas para llevar a cabo el proyecto.

3.4. Coste Real

Para calcular el coste real del proyecto, debemos seguir los mismos pasos que para realizar el presupuesto, pero con los datos reales obtenidos tras la realización del Trabajo de Fin de Grado.

Tanto los componentes software como los componentes hardware se mantienen, siendo el coste real de los mismos:

- Coste Software: 616€
- Coste Hardware: 137,37€

Respecto a los costes de personal, se tiene que tener en cuenta el tiempo que realmente se ha invertido en el proyecto. Por lo tanto, se debe realizar un nuevo cálculo del coste de personal al verse aumentadas las horas de trabajo.

Las tareas llevadas a cabo se asignan a un rol en función del tipo de tarea; al analista aquellas referentes al análisis, al diseño y a la documentación, y al programador las relacionadas con el desarrollo y los datos. Las horas de revisiones se reparten entre los dos roles.

ROL	SUELDO / MES	SUELDO / HORA	HORAS	TOTAL
ANALISTA SOFTWARE	2209€	13,8€	524	7.231,2€
PROGRAMADOR DE APLICACIONES	1819€	11,4€	643	7.330,2€
			Total:	14.561,4 €

Tabla 18: Coste real de personal

Si sumamos todos los costes reales obtenemos:

$$\text{Coste Total} = \text{C. Software} + \text{C. Hardware} + \text{C. Personal} = 15.314,77$$

Si lo comparamos con los presupuestos vemos una clara diferencia entre ambos. En primer lugar se debe tener en cuenta que los presupuestos se realizaron de acuerdo a la planificación inicial y que como se explica anteriormente no se pudo seguir. Además el resultado del COCOMO II no puede ser representativo, ya que no se considera que el proyecto pueda costar 45.105,2 €.

CAPÍTULO 4

ANÁLISIS

El proyecto realizado se compone de dos sistemas; una aplicación web utilizada para la gestión de la base de datos principal, y una aplicación móvil. En este capítulo, se muestra el análisis previo a la implementación de ambos sistemas, separando las especificaciones de los dos, siempre que sea necesario, debido a la existencia de diferencias entre ellos.

4.1. Descripción de los actores

Durante el proceso de análisis se han identificado tres tipos de actores que interactuarán con la web de gestión de la base de datos, y uno que lo hará con la aplicación móvil.

1. Usuario

Éste es el actor que interactuará con la aplicación móvil sin necesidad de registro o identificación.

2. Gestor

Este tipo de actor se encargará de la creación de los datos que serán utilizados por la aplicación móvil mediante el uso de la aplicación web.

Las operaciones que podrá realizar son: añadir paradas y líneas, crear los recorridos de las líneas con sus respectivas paradas y sus horarios, publicar notificaciones y asignarlas a las líneas o paradas que se vean afectadas, y añadir los días festivos en la localidad.

3. Administrador

Su principal cometido es el de la puesta en marcha de la aplicación web para facilitar el trabajo a los usuarios de tipo Gestor.

4.2. Requisitos de usuario

Para evaluar los requisitos de usuarios realizaremos el análisis en dos apartados, diferenciando los relacionados con la aplicación web, de los de la aplicación móvil.

Aplicación Web

RU-01	El Administrador podrá añadir Gestores
RU-02	Un Gestor podrá activar el modo de mantenimiento
RU-03	Un Gestor podrá desactivar el modo de mantenimiento
RU-04	Un Gestor podrá crear una nueva parada
RU-05	Un Gestor podrá visualizar un listado de paradas
RU-06	Un Gestor podrá eliminar una parada
RU-07	Un Gestor podrá crear una nueva línea
RU-08	Un Gestor podrá visualizar un listado de líneas
RU-09	Un Gestor podrá modificar el nombre de la línea
RU-10	Un Gestor podrá habilitar una línea
RU-11	Un Gestor podrá deshabilitar una línea
RU-12	Un Gestor podrá añadir un recorrido a una línea
RU-13	Un Gestor podrá visualizar un listado de los recorridos de cada línea
RU-14	Un Gestor podrá eliminar un recorrido
RU-15	Un Gestor podrá añadir un punto a un recorrido
RU-16	Un Gestor podrá visualizar los puntos de cada recorrido
RU-17	Un Gestor podrá eliminar un punto de recorrido

RU-18	Un Gestor podrá asignar un horario a un punto de recorrido
RU-19	Un Gestor podrá visualizar los horarios de un punto de recorrido
RU-20	Un Gestor podrá eliminar el horario de un punto de recorrido
RU-21	Un Gestor podrá publicar una notificación
RU-22	Un Gestor podrá visualizar las notificaciones registradas
RU-23	Un Gestor podrá eliminar una notificación
RU-24	Un Gestor podrá modificar la descripción de una notificación
RU-25	Un Gestor podrá asignar una notificación a una parada
RU-26	Un Gestor podrá asignar una notificación a una línea
RU-27	Un Gestor podrá autenticarse

Tabla 19: Requisitos de usuario de la aplicación web

Aplicación Móvil

RU-28	Un Usuario podrá visualizar las líneas de autobuses
RU-29	Un Usuario podrá visualizar los recorridos de una línea determinada
RU-30	Un Usuario podrá visualizar una línea determinada en un mapa
RU-31	Un Usuario podrá visualizar información acerca del servicio de "Bus Turístico"
RU-32	Un Usuario podrá buscar paradas por su nombre
RU-33	Un Usuario podrá visualizar en un mapa las paradas cercanas a su posición
RU-34	Un Usuario podrá visualizar un listado de las paradas de interés turístico
RU-35	Un Usuario podrá consultar la información de una parada concreta
RU-36	Un Usuario podrá consultar los horarios de una parada perteneciente a un recorrido
RU-37	Un Usuario podrá visualizar un listado de las paradas favoritas
RU-38	Un Usuario podrá modificar el idioma de la aplicación
RU-39	Un Usuario podrá visualizar las notificaciones disponibles
RU-40	Un Usuario podrá visualizar la información de todas las tarifas
RU-41	Un Usuario podrá consultar la información de los abonos
RU-42	Un Usuario podrá visualizar la información de una notificación
RU-43	Un Usuario podrá añadir paradas favoritas
RU-44	Un Usuario podrá eliminar paradas favoritas
RU-45	Un Usuario podrá geolocalizarse en un mapa
RU-46	Un Usuario podrá visualizar una parada en un mapa

Tabla 20: Requisitos de usuarios de la aplicación móvil

4.2.1. Diagramas de casos de uso

En este apartado, podremos observar las relaciones establecidas entre los diferentes casos de uso, para cada uno de los sistemas implementados.

Aplicación Web

Como podemos observar en la siguiente ilustración, para la aplicación web tendremos dos tipos de actores: el administrador y los gestores. El primero se encargará de la creación de las cuentas de usuario de los gestores autorizados para la gestión de la aplicación.

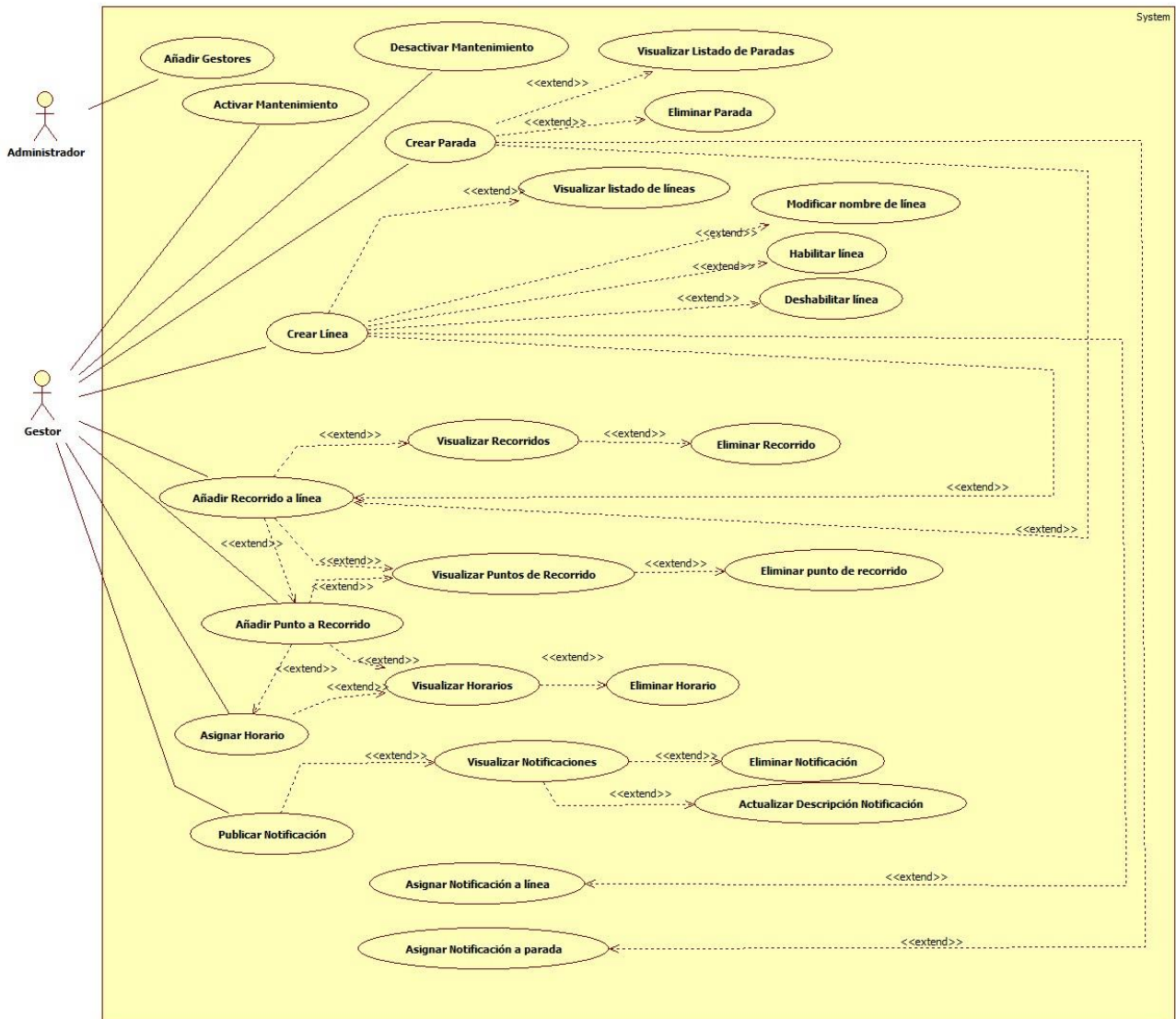
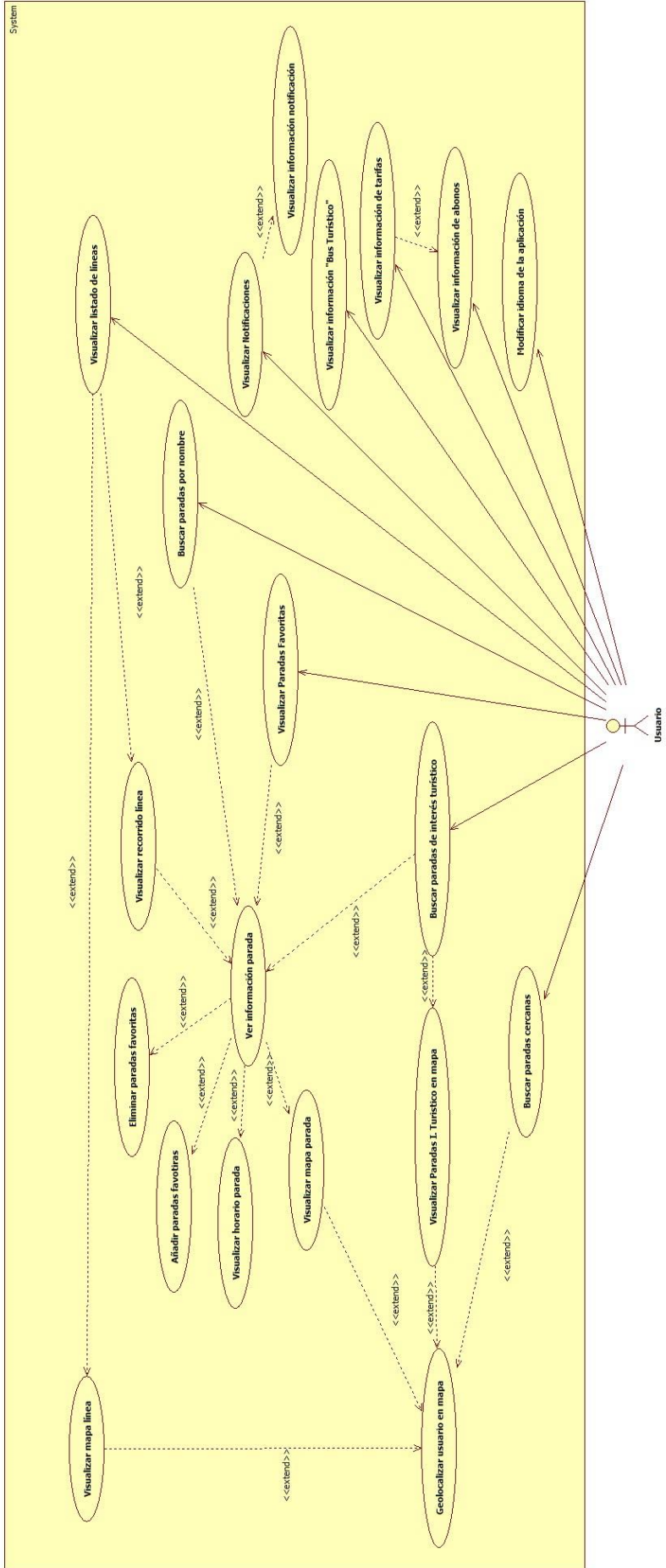


Ilustración 14 Diagrama de casos de uso de la aplicación web:

En segundo lugar, los gestores administrarán la base de datos a la que la aplicación móvil se conecte. Todas estas operaciones estarán condicionadas por los casos de uso de: activar mantenimiento y desactivar mantenimiento. Esto es debido a que no se podrá modificar la base de datos sin estar activado este modo, para poder evitar que un usuario actualice la base de datos a una versión incompleta o inestable.

Aplicación Móvil

Respecto a la app, solo contaremos con usuario que no necesitará registrarse o iniciar sesión para poder acceder a la aplicación. Este usuario podrá acceder a las diferentes funciones a partir del menú principal. El diagrama de casos de uso se muestra en la siguiente ilustración



4.2.1. Especificación de casos de uso

Es este apartado se realizará la especificación de cuatro casos de uso para cada una de las aplicaciones desarrolladas. El resto de los casos de uso se incluirán en un apéndice al final de este documento.

Aplicación Web

Para los Casos de Uso del 02 al 23, el usuario debe haber iniciado sesión como Gestor, para poder realizar las operaciones correspondientes. De este modo, se evita que cualquier usuario que encuentre la web pueda modificar la base de datos.

Para el control de versiones de la base de datos, se utilizará una tabla auxiliar para almacenar la fecha de la última modificación de cada tabla. Por ello, cada vez que se realice modificación de una de ellas, el sistema se encargará de actualizar esta fecha.

CU-01	Añadir Gestores		
Versión	1.0	Fecha	11/05/2017
Requisitos asociados			
Actores	Administrador		
Descripción	El Administrador podrá añadir Gestores desde el panel de administración del plugin de usuarios		
Precondiciones	1. El actor se habrá identificado previamente como administrador		
Secuencia normal	<ol style="list-style-type: none"> 1. El Administrador solicita al sistema la creación de un nuevo usuario 2. El sistema solicita los datos del nuevo Gestor 3. El Administrador envía los datos del nuevo Gestor 4. El sistema registra al nuevo usuario en el sistema 		
Postcondiciones			
Excepciones	4.EX-01	Los datos introducidos no son válidos	
	4.EX-02	El usuario ya estaba registrado	
Frecuencia	Baja. Se añadirán los gestores en la puesta en marcha de la web		
Importancia	Vital		
Comentarios	Los Gestores serán los encargados de gestionar los datos de la aplicación		

Tabla 21: CU-01 Añadir Gestores

CU-02	Activar Mantenimiento		
Versión	1.0	Fecha	11/05/2017
Requisitos asociados	CU-03		
Actores	Gestor		
Descripción	Un Gestor podrá activar el modo de mantenimiento, para realizar modificaciones de la base de datos		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. El modo de mantenimiento debe estar desactivado; CU-03 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita activar el modo de mantenimiento 2. El sistema activa el modo de mantenimiento 		
Postcondiciones			
Excepciones			
Frecuencia	Nula durante la carga inicial de los datos (activado por defecto) Baja durante el mantenimiento de la web		
Importancia	Vital		
Comentarios			

Tabla 22: CU-02 Activar Mantenimiento

CU-03	Desactivar Mantenimiento		
Versión	1.0	Fecha	11/05/2017
Requisitos asociados	CU-02		
Actores	Gestor		
Descripción	El Gestor podrá desactivar el modo de mantenimiento para que la aplicación funcione con normalidad		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. El modo de mantenimiento debe estar activado; CU-02 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita desactivar el modo de mantenimiento 2. El sistema activa el modo de mantenimiento 		
Postcondiciones			
Excepciones	2.EX-03	No existe ninguna línea habilitada	
Frecuencia	Una vez tras la carga inicial de los datos Baja durante el mantenimiento de la web		
Importancia	Vital		
Comentarios			

Tabla 23: CU-03 Desactivar Mantenimiento

CU-04	Crear nueva parada		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-02, CU-05, CU-06, CU-12, CU-15, CU-25		
Actores	Gestor		
Descripción	Un Gestor podrá añadir una nueva parada		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. El modo de mantenimiento debe estar activado 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita introducir una nueva parada 2. El sistema solicita los datos de la parada; RI-02 3. El Gestor envía los datos de la parada 4. El sistema almacena la información en el servidor 5. El sistema notifica el éxito de la operación 		
Postcondiciones			
Excepciones	3.EX-04	El formato de los datos introducidos no coincide con el solicitado	
	4.EX-05	El tamaño de la imagen de la parada supera los 8Mb	
	4.EX-06	El archivo subido al servidor no es una imagen	
Frecuencia	Alta en la carga inicial de los datos Baja durante el mantenimiento de la aplicación		
Importancia	Vital		
Comentarios			

Tabla 24: CU-04 Crear nueva parada

Aplicación Móvil

El funcionamiento de la aplicación móvil dependerá de si la web de gestión se encuentra en modo mantenimiento o no. Esto se deberá a que, si se están modificando los datos y se tratan de actualizar, se pueda llegar a recibir una versión errónea de los mismos. Por lo tanto, será una precondición de todos los casos de uso de la aplicación móvil.

En todas aquellas funciones dentro de la aplicación, que requieran de un mapa para la representación de los datos, será necesaria la existencia de conexión a internet. Y en el caso de querer posicionarnos en el mapa, también será necesaria la activación de la geolocalización del terminal móvil.

CU-28	Visualizar líneas		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-29, CU-30		
Actores	Usuario		
Descripción	Un usuario podrá visualizar todas las líneas de autobuses		
Precondiciones			
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario solicita visualizar líneas 2. El sistema muestra un listado de las líneas ordenadas por su número 		
Postcondiciones	<ol style="list-style-type: none"> 1. El usuario podrá visualizar los recorridos de la línea seleccionada; CU-29 2. El usuario podrá visualizar las paradas que forman la línea en un mapa; CU-30 		
Excepciones			
Frecuencia	Media. Aproximadamente un máximo de una vez por uso de la aplicación		
Importancia	Vital		
Comentarios			

Tabla 25: CU-28 Visualizar líneas

CU-29	Visualizar recorrido de una línea		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-28, CU-35, CU-43, CU-44		
Actores	Usuario		
Descripción	Un usuario podrá visualizar los recorridos de los que dispone una línea		
Precondiciones	1. Selección de una línea desde el listado; CU-28		
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario selecciona la línea que desea consultar 2. El sistema le muestra los recorridos de los que dispone 3. El usuario selecciona el recorrido que desea consultar 4. El sistema muestra las paradas que forman ese recorrido según su orden 		
Postcondiciones	<ol style="list-style-type: none"> 1. Consultar información de la parada; CU-35 2. Añadir parada a paradas favoritas; CU-29 		
Excepciones			
Frecuencia	Media. Aproximadamente un máximo de una vez por uso de la aplicación		
Importancia	Vital		
Comentarios			

Tabla 26: CU-29 Visualizar recorrido de una línea

CU-30	Visualizar línea en mapa		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-28, CU-45		
Actores	Usuario		
Descripción	Un Usuario podrá visualizar las paradas que forman una línea en un mapa		
Precondiciones	1. Selección de una línea desde el listado; CU-28		
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario solicita visualizar el mapa de la línea 2. El sistema muestra un mapa con todas las paradas de la línea seleccionada 		
Postcondiciones	1. Geolocalización del usuario en el terminal; CU-44		
Excepciones			
Frecuencia	Baja. Normalmente para casos de usuarios que sean turistas		
Importancia	Alta		
Comentarios			

Tabla 27: CU-30 Visualizar línea en mapa

CU-31	Ver información "Bus Turístico"		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados			
Actores	Usuario		
Descripción	Un Usuario podrá consultar la información acerca del servicio de "Bus Turístico"		
Precondiciones			
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario solicita visualizar información sobre el Bus Turístico 2. El sistema muestra la información acerca del bus turístico 		
Postcondiciones			
Excepciones			
Frecuencia	Baja. Principalmente para casos de usuarios que sean turistas		
Importancia	Media		
Comentarios			

Tabla 28: CU-31 Ver información "Bus Turístico"

4.4. Requisitos de Información

Tras el análisis realizado en los dos puntos anteriores, obtenemos los siguientes requisitos de información:

- RI-01: El sistema almacenará el nombre, el número y el estado de las líneas.
- RI-02: El sistema almacenará el nombre, la latitud, la longitud, una imagen representativa y el tipo de cada parada.
- RI-03: El sistema almacenará para los recorridos: su nombre, línea a la que pertenece y si es especial.
- RI-04: El sistema almacenará de los puntos de recorrido: la parada a la que están enlazados, el recorrido al que pertenecen, el orden dentro de éste y el tipo de punto.
- RI-05: El sistema almacenará de los horarios: el punto de recorrido al que están asignados, la hora y el tipo de día al que afectan.
- RI-06: El sistema almacenará de las notificaciones: su título, descripción, fecha de inicio, fecha de final y fecha de actualización.
- RI-07: El sistema almacenará de las notificaciones de parada: la notificación a la que representan y la parada a la que afectan
- RI-08: El sistema almacenará de las notificaciones de línea: la notificación a la que representan y la línea a la que afectan.

4.4.1. Modelo conceptual de datos

Para modelar la información con la que se trabajará, se hace uso de un modelo de Entidad-Relación. A continuación se pueden observar en la *Ilustración 15*, las entidades contempladas y las relaciones entre ellas. Los atributos se detallarán en el diccionario de datos, para lograr una mayor claridad del diagrama.

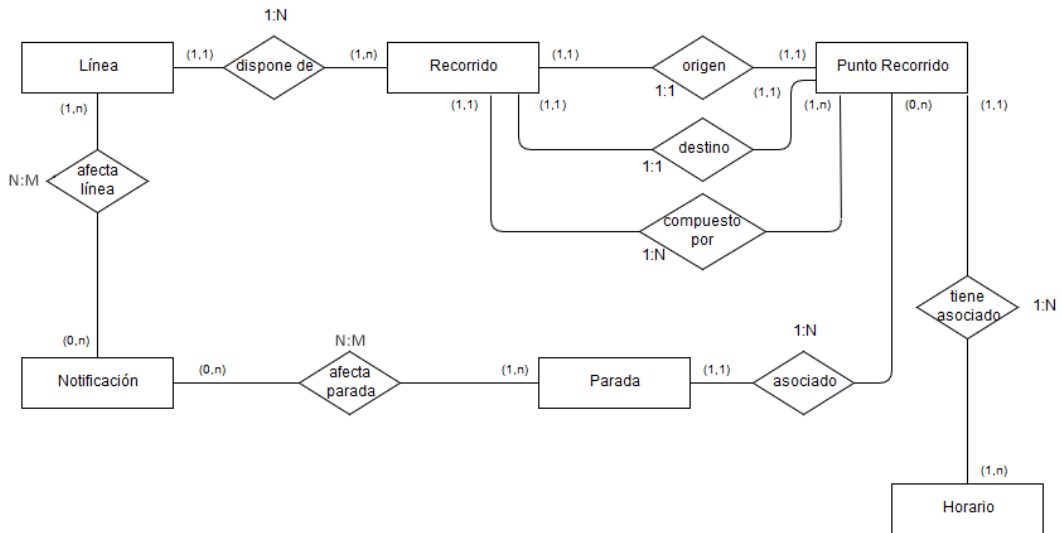


Ilustración 15: Diagrama Entidad Relación

4.4.2. Diccionario de datos

En este punto se describirán las diferentes entidades almacenadas en la base de datos utilizada:

Entidad Línea			
Representa a una línea de autobuses urbanos			
Nombre	Descripción	Tipo de dato	Dominio
<u>id_linea</u>	Id autoincremental único para cada línea	Número	Número de hasta 2 cifras
nombre_linea	Nombre de identificación de la línea	Cadena de caracteres	Caracteres alfanuméricos. Longitud máxima 75
numero	Número de línea	Número	NULLABLE Número de hasta 3 cifras
habilitada	Indica si una línea debe ser cargada en la base de datos de la aplicación móvil	Enum	<ul style="list-style-type: none"> • 0 para no habilitada • 1 para habilitada

Tabla 29: Entidad línea

Las líneas dispondrán de al menos un recorrido y puede que les afecte una o más notificaciones.

Entidad Parada			
Representa a una parada física de la red autobuses urbanos			
Nombre	Descripción	Tipo de dato	Dominio
<u>id_parada</u>	Id autoincremental único para cada parada	Número	Número entero de hasta 3 cifras
nombre_parada	Nombre de identificación de la parada	Cadena de caracteres	Caracteres alfanuméricos. Longitud máxima 50
latitud	Latitud de la posición geográfica de la parada	Cadena de caracteres	Caracteres numéricos. Longitud máxima 18
longitud	Longitud de la posición geográfica de la parada	Cadena de caracteres	Caracteres numéricos. Longitud máxima 18
imagen	URL de la imagen almacenada en el servidor	Cadena de caracteres	Caracteres alfanuméricos. Longitud máxima 150
turística	Indicador de si la parada es de interés turístico o no	Enum	<ul style="list-style-type: none"> • 0 para no turística • 1 para turística
favorita	Este campo solo se creará en la aplicación móvil, y servirá para poder almacenar paradas favoritas	Enum	<ul style="list-style-type: none"> • 0 para no favorita • 1 para favorita

Tabla 30: Entidad parada

Las paradas pueden verse afectadas por una o varias notificaciones y pueden estar asociadas a varios puntos de recorrido.

Entidad Recorrido			
Representa a un recorrido perteneciente a una línea concreta			
Nombre	Descripción	Tipo de dato	Dominio
<u>id_recorrido</u>	Id autoincremental único para cada recorrido	Número	Número entero de hasta 3 cifras
nombre_recorrido	Nombre de identificación del recorrido	Cadena de caracteres	Caracteres alfanuméricos. Longitud máxima 50
línea	Identificador de la línea a la que pertenece el recorrido	Número	Número entero de hasta 2 cifras
especial	Identificador de si el recorrido es de carácter especial o no	Enum	<ul style="list-style-type: none"> • 0 para no especial • 1 para especial

Tabla 31: Entidad recorrido

Los recorridos solo podrán pertenecer a una línea. Respecto a su relación con los puntos de recorrido, tendrán un origen y un destino y podrán estar compuestos por varios puntos extra.

Entidad Punto Recorrido			
Representa a una parada perteneciente a un recorrido			
Nombre	Descripción	Tipo de dato	Dominio
<u>id_punto_recorrido</u>	Id autoincremental único para cada punto de recorrido	Número	Número entero de hasta 4 cifras
recorrido	Identificación del recorrido al que pertenece el punto	Número	Número entero de hasta 3 cifras
parada	Identificación de la parada a la que representa el punto	Número	Número entero de hasta 3 cifras
orden	Orden del punto dentro del recorrido	Número	Número entero de hasta 3 cifras
rol	Rol que representa la parada dentro del recorrido	Enum	<ul style="list-style-type: none"> • 0 para origen • 1 para destino • 2 para intermedio

Tabla 32: Entidad punto recorrido

Los puntos de recorrido pueden ser el origen, destino o punto intermedio de un recorrido. Cada punto de recorrido tendrá asignado un horario y estará relacionado con una parada.

Entidad Horario			
Representa el horario asignado a un punto concreto de un recorrido			
Nombre	Descripción	Tipo de dato	Dominio
<u>id_horario</u>	Id autoincremental único para cada horario	Número	Número entero de hasta 5 cifras
punto_recorrido	Identificación del punto de recorrido al que pertenece el horario	Número	Número entero de hasta 4 cifras
hora	Hora a la que pasa el autobús	Time	Hora:Minutos
tipo_día	Tipo de día al que pertenece el horario	Enum	<ul style="list-style-type: none"> • 0 para laboral • 1 para sábado • 2 para festivo

Tabla 33: Entidad horario

Un horario estará asignado solo a un punto de recorrido.

Entidad Notificación			
Nombre	Descripción	Tipo de dato	Dominio
<u>id_notificación</u>	Id autoincremental único para cada notificación	Número	Número entero de hasta 5 cifras
titulo	Título de la notificación	Cadena de caracteres	Caracteres alfanuméricos Longitud máxima 64
descripcion	Descripción de la notificación	Cadena de caracteres	NULLABLE Caracteres alfanuméricos Longitud máxima 500
fecha_inicio	Fecha de inicio de la notificación	Date	NULLABLE Día/mes/año
fecha_final	Fecha de finalización de la notificación	Date	NULLABLE Día/mes/año
fecha_actualización	Fecha de la última modificación de la notificación	Datetime	Día/mes/año hora/minutos/segundos

Tabla 34: Entidad notificación

Las notificaciones puede que afecten a las líneas y a las paradas.

CAPÍTULO 5

DISEÑO

En este capítulo se pretende exponer el trabajo de diseño realizado previo a la creación de las aplicaciones. Se tratarán tanto las arquitecturas lógicas y físicas de las aplicaciones, como los diseños de algunas pantallas de la interfaz de usuario de la app desarrollada.

5.1. Arquitectura lógica

Al tener una aplicación web y una aplicación móvil, debemos analizar la arquitectura lógica de ambas.

Aplicación Web

Como podemos observar en la siguiente ilustración, la web de gestión de la base de datos sigue una arquitectura cliente-servidor. Ésto consiste en que un cliente envía una serie de peticiones al servidor la cuales generan unas respuestas, que son enviadas de nuevo al cliente inicial.

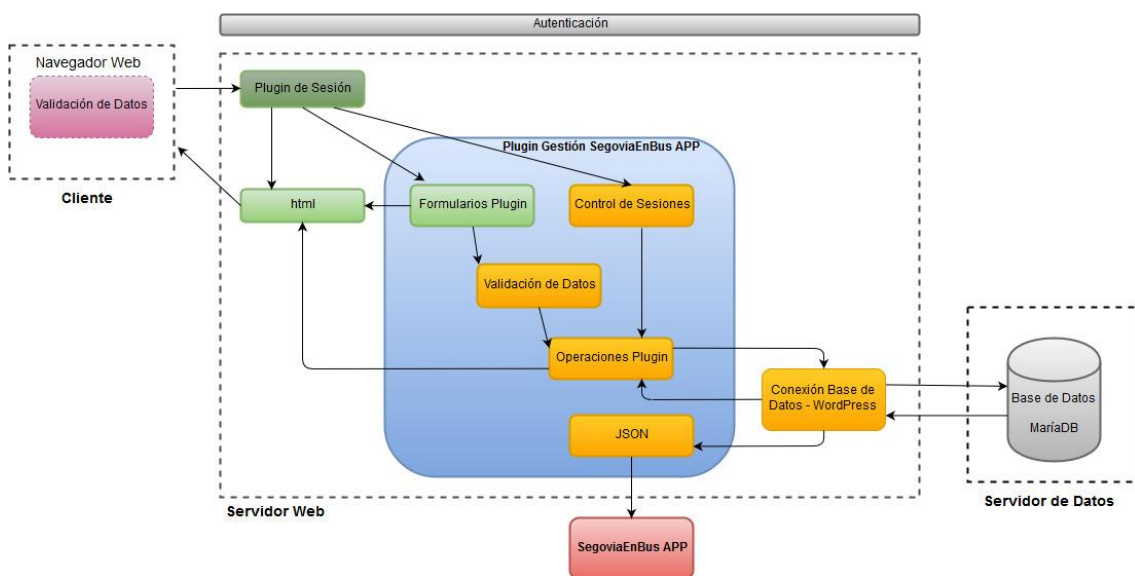


Ilustración 16: Arquitectura Lógica de la Aplicación Web

En nuestro lado cliente tendremos el navegador web, tratándose de la forma en la que el usuario interactuará con la web. Para limitar la entrada de los datos, se implementará una validación de datos a través de las opciones ofrecidas por HTML5 y el atributo "pattern" de los campos de entrada de datos.

Respecto al lado del servidor web, la interacción con él se verá limitada con un control de acceso para usuarios autorizados y estará formado por tres capas: Modelo, Vista y Controlador. La capa de la Vista se encarga de enviar los datos al cliente para que pueda visualizarlos, en nuestro caso tendremos el código HTML que genera cada una de las páginas del WordPress y, además, el generado por los formularios creados desde el

Plugin de Gestión de la aplicación móvil y la visualización de los datos obtenidas a través de algunas de las operaciones del mismo plugin.

En el caso del Controlador, contamos con el control de las sesiones del usuario. Éste se encarga de controlar que los usuarios que no inicien sesión no puedan acceder a aquellas páginas que pueden alterar la base de datos.

Para terminar con el Servidor Web tenemos la capa del Modelo, ésta capa se encarga de las operaciones que se realizan con los datos que llegan al servidor a través de los formularios, y de las conexiones con el servidor de datos y la aplicación móvil.

En primer lugar, se deben gestionar las sesiones de la web y validar los datos recibidos. Esta validación se produce para evitar posibles ataques al servidor mediante inyección de código. Una vez validados los datos y comprobada la sesión, se procederá al envío al servidor de datos, ya sea para insertar nuevos registros, modificarlos o eliminarlos.

Finalmente dentro del modelo, tenemos la funcionalidad que nos permite enviar los datos almacenados en el servidor de datos, mediante el uso de JSON, a la aplicación móvil.

En ambos casos la conexión con el servidor de datos se realizará mediante las funciones proporcionadas por WordPress, éstas son llamadas a través de la variable global `$wpdb` y de todas las operaciones que facilita.

Para acabar con la arquitectura lógica de la web, nos queda explicar que el servidor de datos almacena tanto las tablas generadas por el plugin desarrollado, como las que se encargan del funcionamiento de WordPress y del resto de plugins complementarios elegidos para mejorar el funcionamiento y rendimiento de la web.

Aplicación Móvil

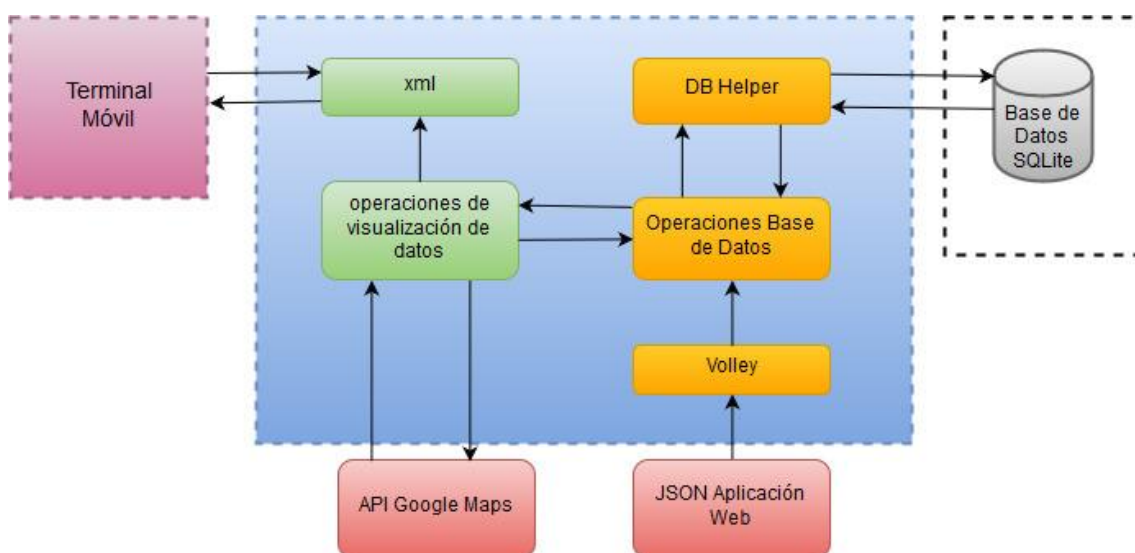


Ilustración 17: Arquitectura Lógica Aplicación Móvil

En este caso el terminal móvil accede a la aplicación previamente instalada. Al iniciarla visualiza los XML generados según la opción seleccionada por el usuario. Éstos XML son

creados a partir de las consultas disponibles a la base de datos, por medio del controlador DBHelper.

La aplicación móvil se conecta a dos sistemas externos en primer lugar, utiliza la API de Google Maps para la creación de los mapas, y en segundo lugar se utiliza unos métodos proporcionados por la librería Volley para el tratamiento de los JSON obtenidos de la conexión al servidor web, en el que la aplicación allí instalada almacena la versión actualizada de los datos necesarios para trabajar correctamente en la app.

5.2. Arquitectura física

Este apartado se explicará la arquitectura física que se utiliza en el proyecto desarrollado, sin embargo, se ve necesario detallar una arquitectura física ideal para el sistema, ya que por limitaciones de recursos como proyecto académico, no se puede desplegar la aplicación en las condiciones ideales para proporcionar el mejor servicio posible a los usuarios finales.

5.2.1. Arquitectura física utilizada

En este caso, la arquitectura es bastante sencilla. Como podemos observar en la siguiente ilustración, contaremos con una máquina, la cual ha sido proporcionada por la Universidad de Valladolid, en la que tendremos la capa de aplicación con el servidor web, y la capa de datos con el servidor de datos. A éstas se conectarán los diferentes tipos de clientes: los usuarios de la web desde un navegador y los usuarios finales desde la aplicación móvil desarrollada.

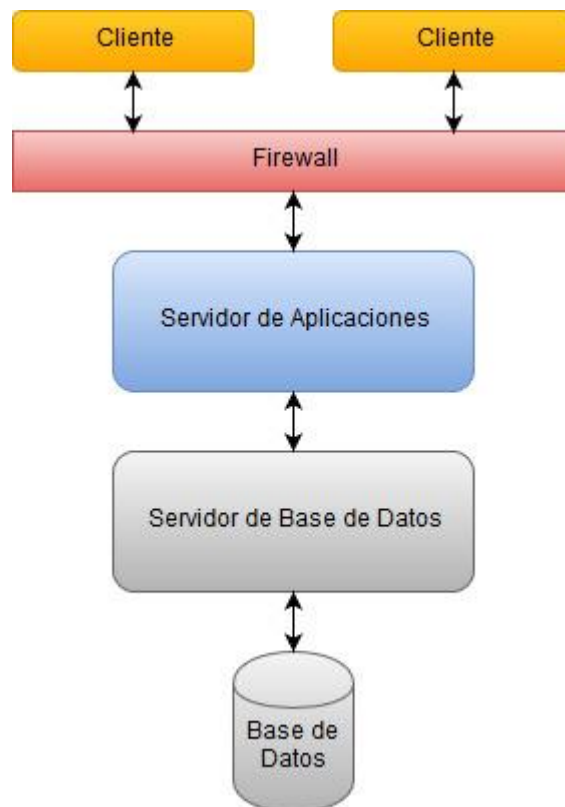


Ilustración 18: Arquitectura física utilizada

La máquina utilizada cuenta con un Firewall o Cortafuegos, que se encarga de restringir la conexión a la misma por dos puertos: el 62022 para la conexión HTTP y el 62021 para SSH. La primera conexión será para los usuarios de la web de gestión de la base de datos de la aplicación móvil, mientras que la segunda, será utilizada únicamente por el administrador del sistema para gestionar el despliegue de la web.

Una vez traspasado el Firewall tendremos el servidor de aplicaciones, éste se encargará de recibir las peticiones de los clientes y generar las respuestas adecuadas.

Finalmente contamos con el servidor de datos, el cual conecta la capa de aplicación con la base de datos. De este modo el servidor web podrá realizar consultas a la misma a partir de las peticiones de los clientes.

5.2.2. Arquitectura física ideal

Como se ha mencionado anteriormente, existe una arquitectura física ideal para el despliegue del proyecto desarrollado. Como podemos observar en la siguiente ilustración, contaremos al igual que en la arquitectura física anterior, con la capa del servidor web y la del servidor de datos junto con el mismo tipo de clientes. Sin embargo existen una serie de diferencias enfocadas a mejorar el rendimiento y la disponibilidad del sistema.

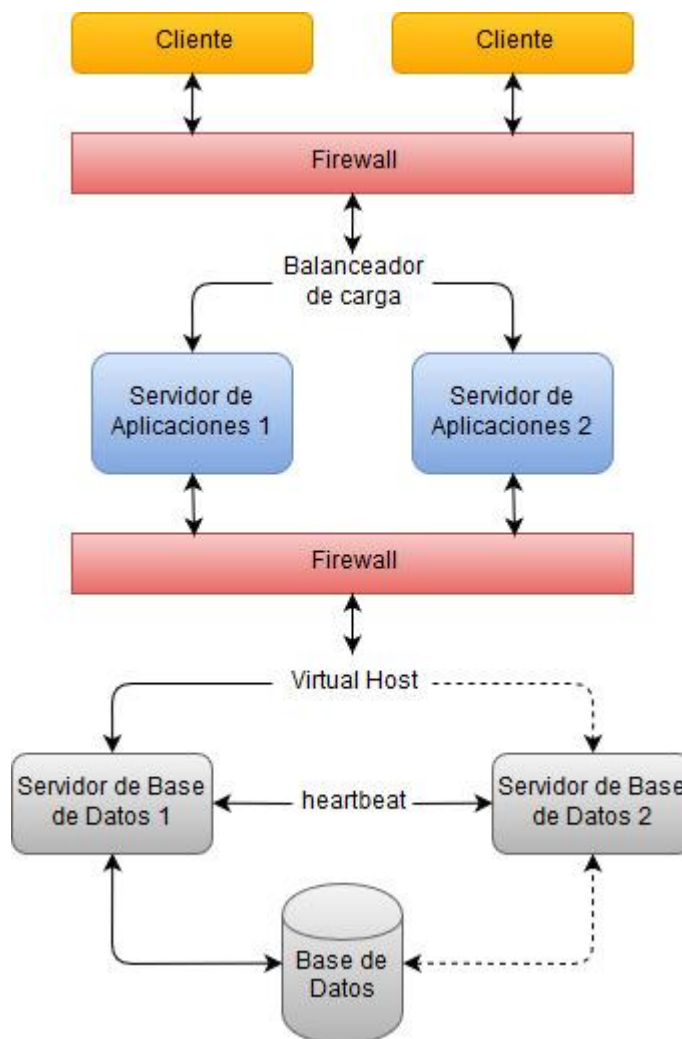


Ilustración 19: Arquitectura física ideal

En primer lugar contamos con dos servidores conectados a un balanceador de carga. Con esta medida se reparten entre los dos servidores las peticiones realizadas por los clientes para así evitar la sobrecarga de un único servidor. El número de servidores varía en función del número de peticiones previstas. Además de mejorar el rendimiento, con esta medida también se incrementa la disponibilidad, ya que en el caso de que falle un servidor, el otro podrá continuar ofreciendo un servicio mínimo a los usuarios.

Tras la capa del servidor de aplicaciones, sería conveniente tener un Firewall adicional para aumentar la seguridad del sistema y poder bloquear posibles ataques que no hayan sido detenidos por el primero.

Finalmente en la capa de datos, contaremos con un Host Virtual, cuya función será la de ofrecer la posibilidad de alojar más de un servidor de datos para la aplicación. En este caso tendríamos dos servidores de datos conectados entre sí por un “detector de pulsaciones”, éste se encargará de detectar si el servidor de datos principal está en funcionamiento o no. De este modo en caso de que el servidor activo deje de funcionar, podrá sustituirlo temporalmente el segundo.

5.3. Diagramas de clases

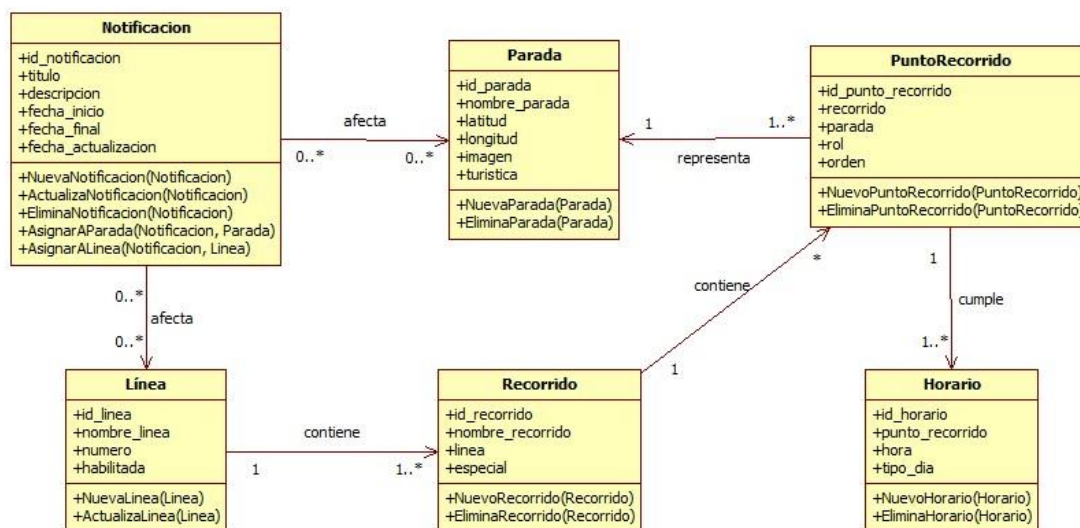


Ilustración 20: Diagrama de clases de la aplicación web

En la Ilustración 20 podemos observar el diagrama de clases analizado de cara a la implementación de la aplicación web. En él se muestran todas las entidades junto a sus atributos y las operaciones que se pueden realizar.

A continuación se puede observar en la *Ilustración 21* el diagrama de clases de la aplicación móvil. Éste conserva la estructura del anterior, pero los métodos de las diferentes clases cambian completamente. Además el atributo de habilitada en la clase Línea desaparece, mientras que en la clase Parada se crea uno nuevo llamado favorita.

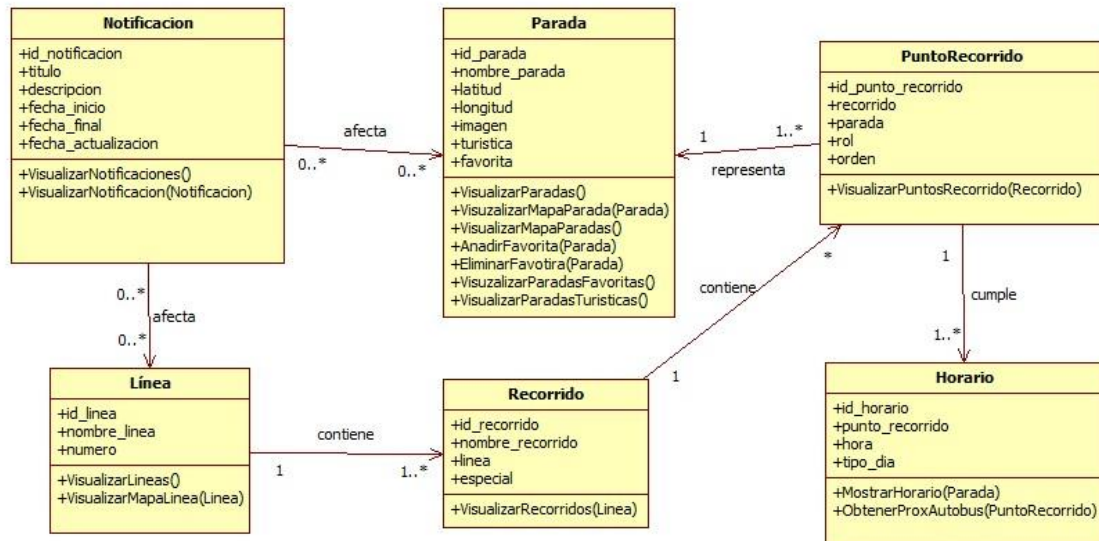


Ilustración 21: Diagrama de clases de la aplicación móvil

5.4. Modelo lógico de la base de datos

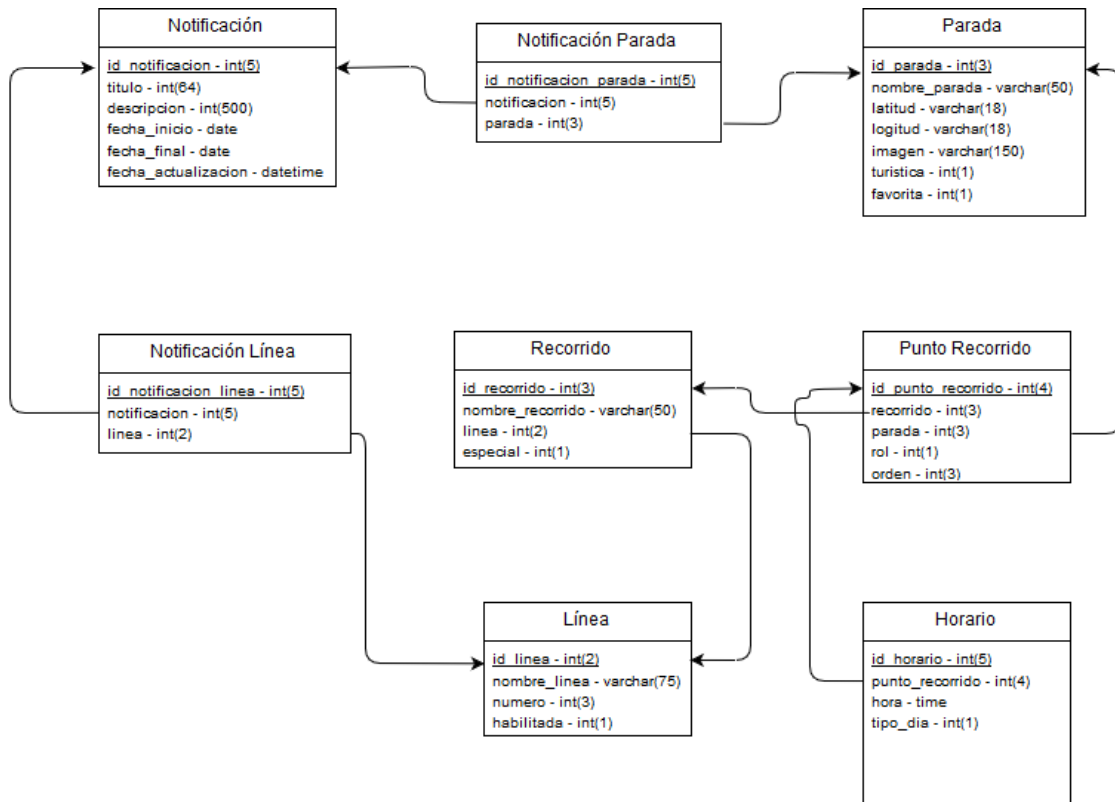


Ilustración 22: Modelo relacional

En la *Ilustración 22* podemos observar la forma en la que se van a almacenar los atributos dentro de la base de datos y las relaciones entre las diferentes tablas mediante el uso de claves foráneas. Finalmente se comprueba cómo han aparecido dos tablas a partir de las relaciones N:M del diseño conceptual explicado en el capítulo de análisis. Estas dos nuevas tablas son las siguientes:

- **NotificacionLinea:** Modela la relación entre las entidades línea y notificación. A través de esta tabla se podrán registrar todas aquellas notificaciones que afectan a una o varias líneas.
- **NotificacionParada:** Modela la relación entre las entidades parada y notificación. A través de esta tabla se podrán registrar todas aquellas notificaciones que afectan a una o varias paradas

5.5. Diseño de la interfaz

El diseño de la aplicación ha sido realizado con el objetivo de lograr una navegabilidad sencilla e intuitiva. De este modo los usuarios que cuenten con los conocimientos básicos del manejo de un dispositivo móvil Android, podrán utilizar la app sin inconvenientes.

El color que ha sido elegido como tema de la aplicación es el azul, ya que representa tanto a la bandera de la ciudad de Segovia, como al color de los autobuses urbanos. Además se han incluido algunos elementos con una tonalidad naranja, al ser uno de los colores complementarios del azul.

En cuanto a la navegabilidad, se ha diseñado un menú con todas las funciones consideradas como principales, y de especial interés para los usuarios. Este menú podrá visualizarse en cualquier momento durante el uso de la aplicación, accediendo a él desde un icono situado en la parte superior de la pantalla.

Se ha considerado que la navegabilidad dentro del dispositivo debe realizarse con el terminal en posición vertical, ya que permitirá visualizar los listados de una forma más apropiada.

A continuación se desarrolla la explicación detallada de alguna de las pantallas que componen la aplicación.


Bienvenida	
Descripción	Pantalla de inicio de la aplicación. Se muestra el menú de la aplicación y tras él, un listado de las líneas del servicio de autobuses.
Activación	Inicio de la aplicación
Diseño	 <p style="text-align: center;"><i>Ilustración 23: Diseño Bienvenida</i></p>
Eventos	Acceso a todas las secciones disponibles en el menú Acceso directo a la pantalla de líneas

Tabla 35: Diseño Bienvenida


Líneas	
Descripción	Listado de todas las líneas de autobuses urbanos. Se dispone de un buscador accionado mediante la selección del icono de la lupa para buscar las líneas.
Activación	Inicio de la aplicación (al cerrar el menú). Desde el menú desplegable.
Diseño	 <p style="text-align: center;"><i>Ilustración 24: Diseño Líneas</i></p>
Eventos	Selección de una línea. Acceso al menú desplegable.

Tabla 36: Diseño Líneas

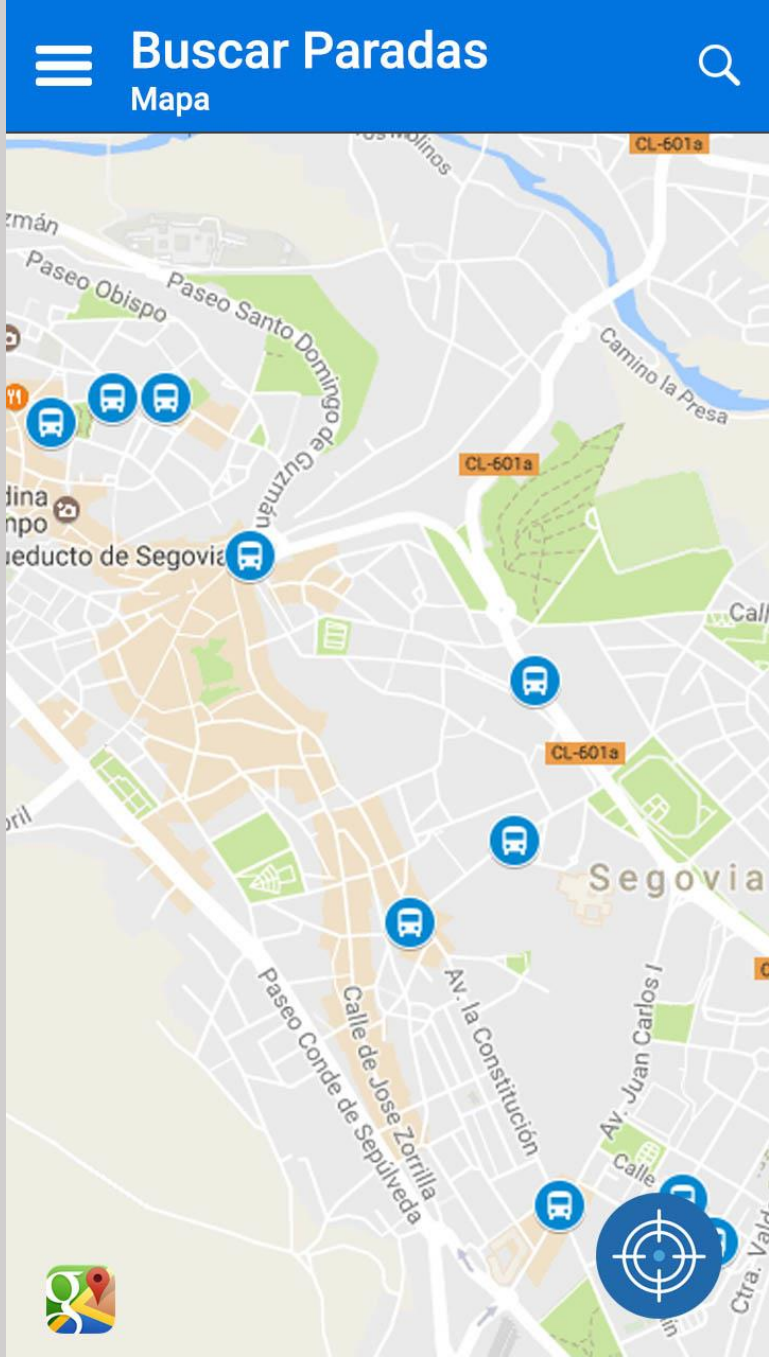
Paradas cercanas	
Descripción	<p>Se muestran todas las paradas en Google Maps.</p> <p>Se ofrece la opción de geolocalizar al usuario para ver donde está respecto a las paradas.</p> <p>Se ofrece la opción de la búsqueda mediante la selección del icono de la lupa.</p>
Activación	Acceso desde el menú desplegable.
Diseño	 <p>The screenshot shows the 'Buscar Paradas' (Search Stops) screen of the SegoviaEnBus app. At the top, there is a blue header with a hamburger menu icon on the left, the text 'Buscar Paradas' in white, and a magnifying glass icon on the right. Below the header is a map of Segovia, Spain, with several blue circular icons containing a white bus symbol, representing bus stops. The map includes street names like 'Paseo Obispo', 'Paseo Santo Domingo de Guzmán', 'Calle de Jose Zorrilla', and 'Av. la Constitución'. A large blue target icon is overlaid on the map, indicating the user's current location. The Google Maps logo is visible in the bottom left corner of the map area.</p> <p style="text-align: center;"><i>Ilustración 25: Diseño Paradas cercanas</i></p>
Eventos	<p>Selección de parada.</p> <p>Geolocalización del usuario.</p> <p>Acceso al menú desplegable.</p>

Tabla 37: Diseño Paradas cercanas

Bus turístico	
Descripción	Se muestra información sobre el servicio de autobús turístico de Segovia
Activación	Acceso desde el menú desplegable.
Diseño	<div data-bbox="475 322 1326 958" data-label="Image"> </div> <p>A los pies del Acueducto, encontramos aparcado el autobús turístico que recorre Segovia y sus alrededores durante aproximadamente cincuenta minutos.</p> <p>Nada más subir al segundo piso del autobús, una sensación de emoción nos recorre, ya que observamos el monumento romano desde otra perspectiva. Al ver el Acueducto desde la altura, lo sentimos más cerca, parece que casi lo podemos tocar. Lo primero que llama la atención es saber que antes los coches pasaban por debajo de éste.</p> <p>El autobús arranca y entramos al Paseo de Santo Domingo de Guzmán, dejando a un lado la muralla. A escasos metros del Acueducto, nos sorprende cómo de repente la vegetación aparece ante nuestros ojos a ambos lados de la carretera y el audio guía del autobús nos avisa de que estamos cruzando el Cinturón Verde de Segovia.</p> <p style="text-align: center;"><i>Ilustración 26: Diseño Bus Turístico</i></p>
Eventos	Acceso al menú desplegable.

Tabla 38: Diseño Bus turístico

Tarifas	
Descripción	Se muestra información de todos los tipos de tarifas.
Activación	Acceso desde el menú desplegable.
Diseño	 <p>(*) Cumpliendo las condiciones fijadas por el Excmo. Ayuntamiento de Segovia. (**) De Lunes a Sábados, excepto festivos. (Precios con IVA incluido)</p> <p>Mas información</p> <p><i>Ilustración 27: Diseño Tarifas</i></p>
Eventos	Visualizar información de abonos. Acceso al menú desplegable.

Tabla 39: Diseño Tarifas

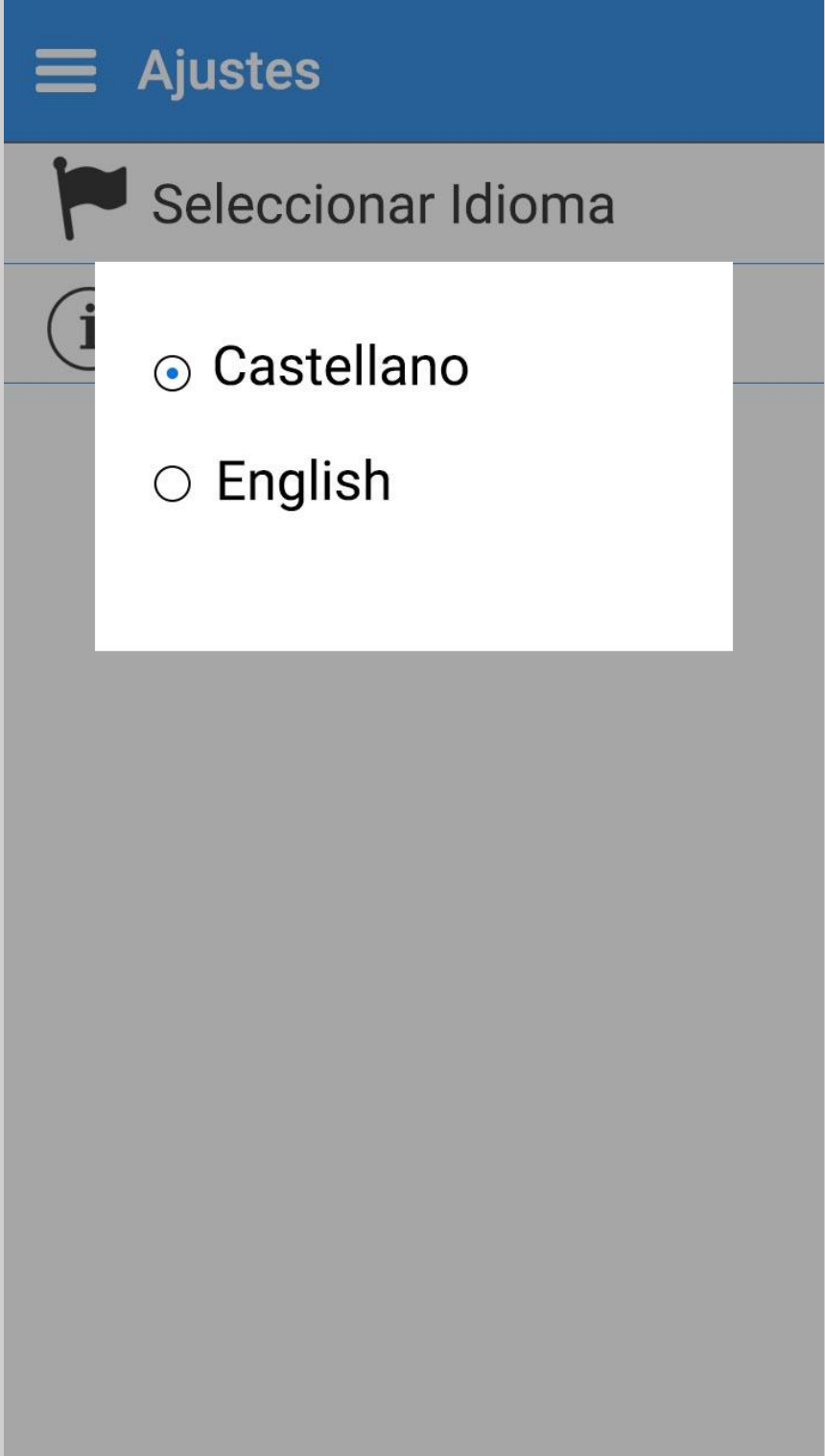
Selección de Idioma	
Descripción	Se muestra un listado seleccionable de los idiomas disponibles.
Activación	Acceso desde la pantalla de ajustes.
Diseño	 <p style="text-align: center;"><i>Ilustración 28: Diseño Selección de Idioma</i></p>
Eventos	Selección de idioma.

Tabla 40: Diseño Selección de idioma

CAPÍTULO 6

IMPLEMENTACIÓN

En este capítulo se procede a explicar el desarrollo llevado a cabo a lo largo del proyecto. En primer lugar se definirán todas las herramientas utilizadas en la creación de la memoria y la implementación del plugin para WordPress y de la aplicación móvil.

6.1. Herramientas utilizadas

En este apartado se analizarán las diferentes herramientas utilizadas durante el desarrollo del proyecto. Para ello, se dividirán en tres grupos diferentes; el primero hará referencia a todas aquellas herramientas utilizadas durante las etapas de análisis. En segundo lugar, se explicarán las que se necesitaron para desarrollar correctamente la aplicación web. Finalmente se expondrán las empleadas para la implementación de la aplicación móvil.

6.1.1. Herramientas de análisis

Éstas fueron utilizadas para la creación de los diferentes diagramas y esquemas descritos en este documento.

Draw.io

Se trata de una herramienta online alojada en la dirección <https://www.draw.io/>. En la *Ilustración 28* podemos observar algunas de sus funcionalidades. Esta aplicación ha sido utilizada para la creación de los árboles de características y el diagrama Entidad/Relación.

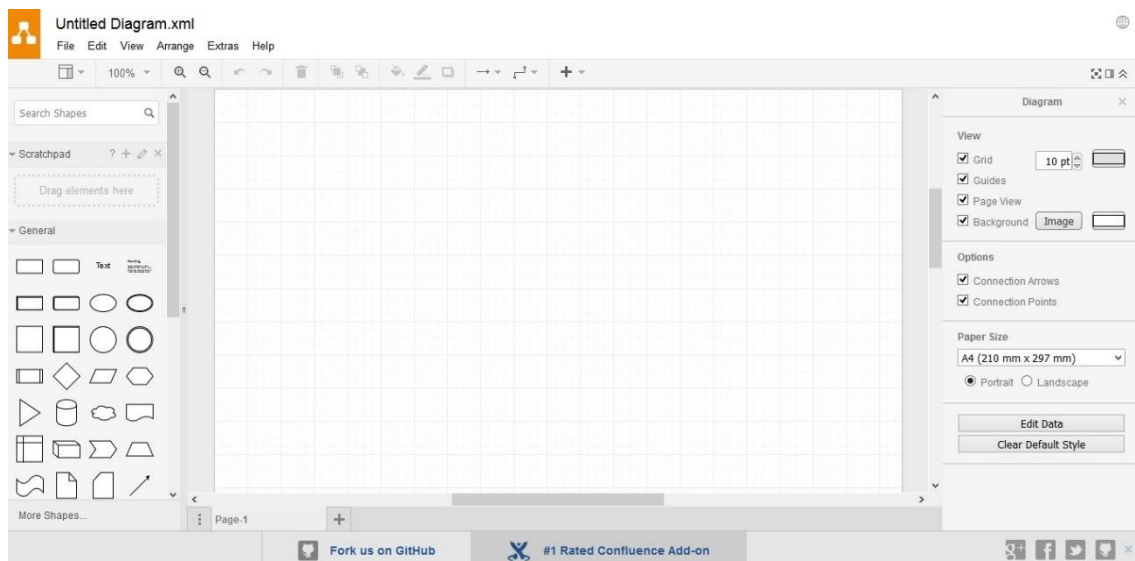


Ilustración 29: Captura de la herramienta Draw.io

StarUML

En este caso se trata de una aplicación de escritorio que permite la creación de múltiples tipos de diagramas. Respecto al proyecto, ha sido útil para la creación de: Diagramas de Gantt, Diagramas de Casos de Uso, Diagrama de Clases de Análisis, Diagramas de Secuencia, Diagramas de Estados y Diagrama de Clases (Diseño).

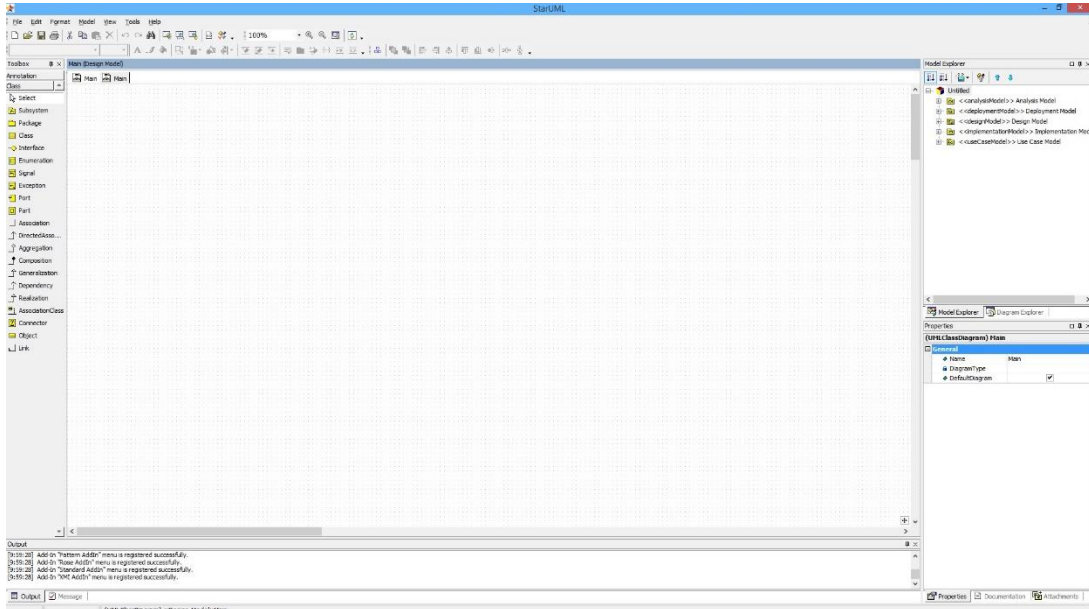


Ilustración 30: Captura de pantalla de StarUML

Photoshop

Programa de edición de imágenes. Éste ha sido el medio por el cual se han creado las diferentes capturas del diseño de la aplicación móvil. En la Ilustración 30 podemos observar la creación de una de estas capturas.

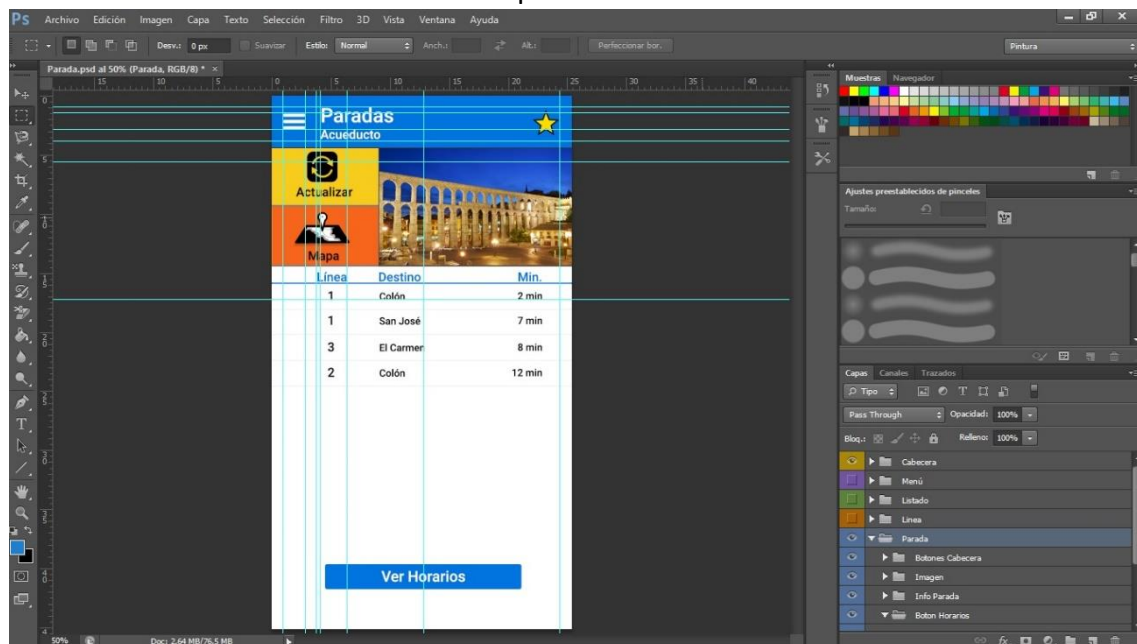


Ilustración 31: Captura de Photoshop

Microsoft Word

Programa de procesamiento de textos. Utilizado para la creación de este documento.



Ilustración 32: Captura Microsoft Word

6.1.2. Herramientas para el desarrollo web

A la hora de crear una aplicación web hay diversas formas y lenguajes de programación para crear una solución. En el caso de este proyecto, se ha optado por crear una aplicación web a través del gestor de contenidos WordPress, y la implementación de un plugin para la gestión de la base de datos de la aplicación móvil.

WordPress

WordPress se trata de un sistema de gestión de contenidos que facilita la creación de páginas web. Su principal uso ha sido la creación de blogs, debido a la sencillez a la hora de crear nuevo contenido. Sin embargo, hoy en día se ha popularizado y se utiliza para la creación de todo tipo de páginas web; desde tiendas online, hasta páginas para promocionar cualquier tipo de servicio.

En el mundo un 30% de todas las páginas web utilizan un gestor de contenidos, siendo un 18.9% gestionadas con WordPress⁸ y el 11.1% restante a través de otro tipo de gestores de contenido. Es por ello que podemos apreciar que WordPress es el sistema favorito de los usuarios.

El éxito de esta herramienta se debe a la facilidad de gestión del contenido mediante el uso de diversas plantillas y plugins.

A continuación se detallarán la plantilla y los plugins utilizados para la puesta en marcha de la aplicación web.

Hood Themes⁹

Se trata de una plantilla premium de WordPress que ofrece diversas opciones para la creación de páginas web. Además contiene la licencia de algunos plugins de pago muy utilizados por la comunidad WordPress.

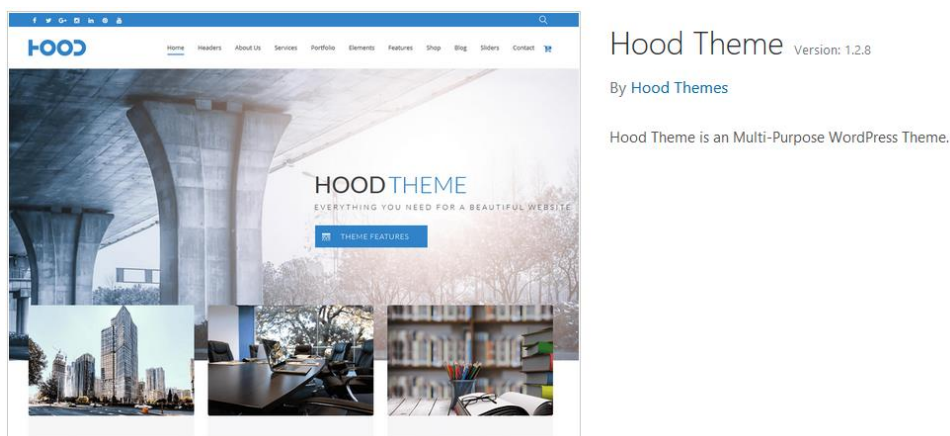


Ilustración 33: Captura de la descripción de la plantilla

⁸ 14 Surprising Statistics About WordPress Usage - <https://managewp.com/14-surprising-statistics-about-wordpress-usage>

⁹ Plantilla Premium para WordPress – Hood Theme - <https://themeforest.net/item/hood-responsive-multipurpose-theme/15730392>

BackWPup

Plugin utilizado para realizar una copia periódica de la base de datos. De este modo en caso de sufrir un ataque exitoso, se podrán recuperar todos los datos de la última versión sin infectar.

Visual Composer

Plugin empleado para la construcción de la estructura de las diferentes páginas web internas.

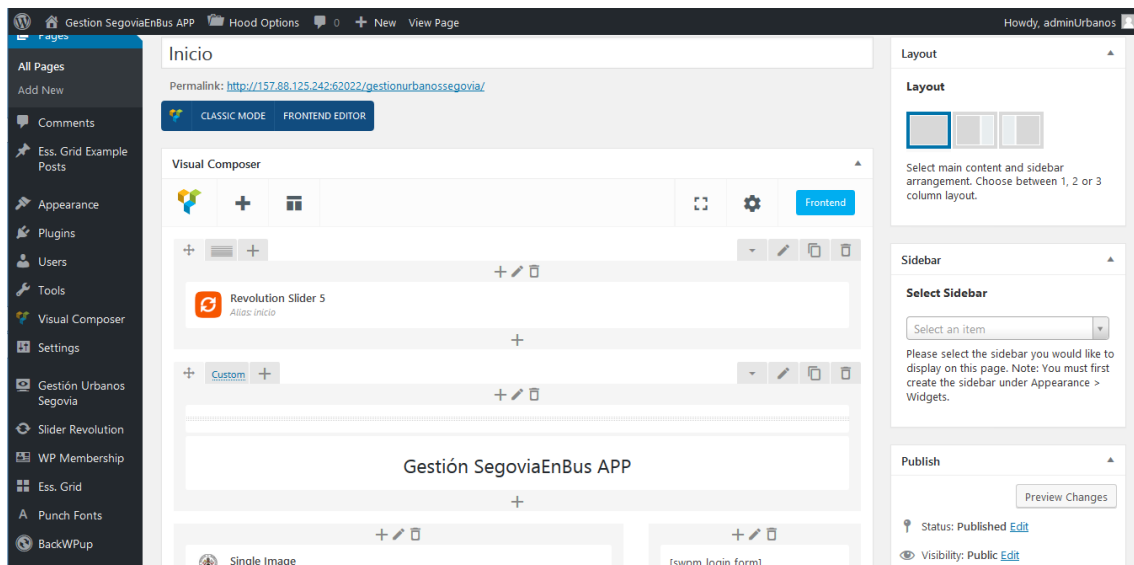


Ilustración 34: Captura edición de página de Inicio

Slider Revolution

Plugin de pago para la creación de sliders con multitud de efectos de transición. Empleado para crear el slider con las imágenes de la página de inicio.

Simple Facebook Page Plugin

Plugin utilizado para poder añadir un widget de la página de Facebook que deseemos. En nuestro caso se tratará del Facebook del Ayuntamiento de Segovia, el cual será cargado en el pie de la página.

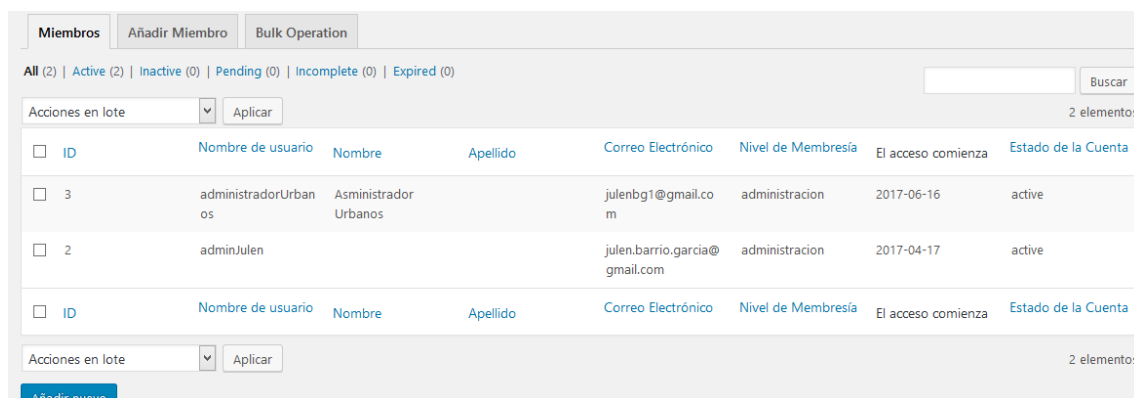


Ilustración 35: Captura panel de administración de usuarios

Simple WordPress Membership

Plugin utilizado para la gestión de los usuarios de tipo Gestor de la aplicación web (Ilustración 34). A través de él, el Administrador se encargará de crear las cuentas de usuario necesarias completando los siguientes campos:

- Nombre de usuario
- Correo electrónico
- Contraseña

También se utilizó **Photoshop** (mencionado previamente en las herramientas utilizadas para el análisis). En este caso su uso ha sido enfocado a la edición de las fotografías tomadas de cada una de las paradas de autobuses. Las acciones llevadas a cabo fueron: difuminación de rostros, matrículas y emplazamientos publicitarios, y creación de fotomontajes entre la parada y los espacios de interés turístico de Segovia a los que representa.

Respecto a la conexión con el servidor se utilizaron dos programas:

FileZilla

Se trata de un cliente FTP multiplataforma, que sirve para poder realizar subidas y bajadas de archivos entre el servidor web y el equipo del administrador.

El protocolo que ha sido utilizado es SFTP mediante SSH.

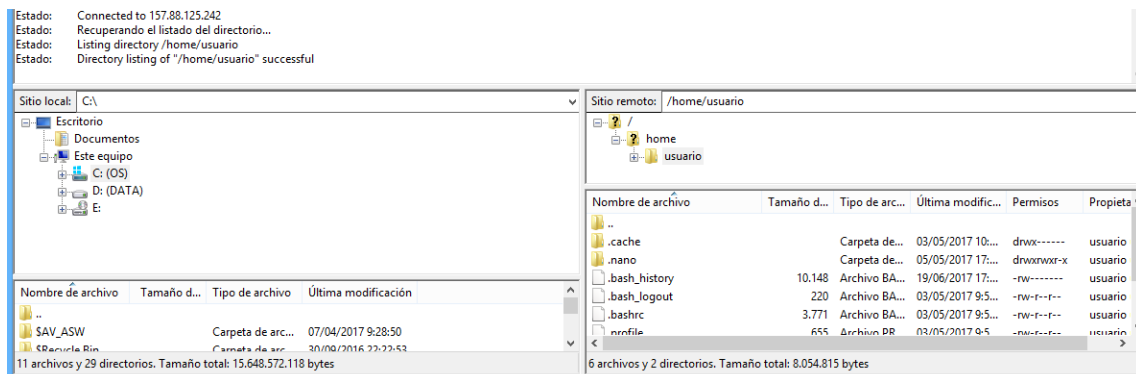


Ilustración 36: Conexión mediante FileZilla

Oracle VM VirtualBox

Programa utilizado para la creación de máquinas virtuales dentro de un sistema operativo. En este caso se utilizará una máquina virtual de Ubuntu Server, para conectarse al servidor desde la consola a través de ssh. De este modo se podrá configurar el servidor correctamente desde la consola.

Finalmente, se utilizó una aplicación de edición de textos para el desarrollo de los diferentes archivos php utilizados en el plugin; esta aplicación es **Sublime Text 2**.


```

1 <?php
2
3 Plugin Name: Gestión de autobuses urbanos de Segovia
4 Plugin URI: http://137.88.119.142:8082/gestionurbanosegovia/
5 Description: Plugin desarrollado para gestionar la aplicación móvil de autobuses urbanos de Segovia. Implementado para el trabajo de fin de grado del alumn
6 Version: 1.0
7 Author: Julien Barrio Garcia
8 Author URI: http://137.88.119.142:8082/gestionurbanosegovia/
9
10 //IMPLEMENTACION SHORTCODES
11
12 function nueva_parada(){
13     $ruta_wordpress = get_home_url();
14     $formularioNuevaParada = "<form method='post' enctype='multipart/form-data' action='". $ruta_wordpress."/wp-content/plugins/gestionAutobuses-JulienBarrio/op
15     <table style='margin-top:10px;'>
16         <tr valign='top'>
17             <th><label form='nombre_parada'>Nombre de parada: </label></th>
18             <td><input type='text' id='nombre_parada' name='nombre_parada' pattern='[A-Za-z0-9áéíóü.-()@-]{3,50}' title='Tamaño del campo entre 3 y 5
19             </td>
20         </tr>
21         <tr valign='top'>
22             <th><label><strong>Posición</strong></label></th>
23             <td><input type='text' id='latitud' name='latitud' pattern='[-+](0,1)[0-9]{1,2}[.]{1}[0-9]{4,14}' title='Tamaño máximo 18 caracteres. Formato: -
24             </td>
25         </tr>
26         <tr valign='top'>
27             <th><label form='longitud'>Longitud</label></th>
28             <td><input type='text' id='longitud' name='longitud' pattern='[-+](0,1)[0-9]{1,2}[.]{1}[0-9]{4,14}' title='Tamaño máximo 18 caracteres. Formato:
29             </td>
30         </tr>
31         <tr valign='top'>
32             <th><label form='imagen'>Imagen (tam. máximo 8Mb)</label></th>
33             <td><input type='file' id='imagen' name='imagen' accept='image/jpg, image/jpeg, image/png' required/>
34             </td>
35         </tr>
36         <tr valign='top'>
37             <th><label form='parada_turistica'>Parada de interés turístico</label></th>
38             <td><input type='checkbox' id='parada_turistica' name='parada_turistica' value='si'>
39             </td>
40         </tr>
41         <tr>
42             <td colspan='2'><input type='submit' id='l_submit' value='Añadir Parada' />
43         </td>
44     </form>
45     return $formularioNuevaParada;
46 }
47 add_shortcode('nueva_parada', 'nueva_parada');

```

Ilustración 37: Captura Programa Sublime Text 2

6.1.3. Herramientas para el desarrollo de la aplicación móvil

Como se menciona al comienzo de este documento, la aplicación móvil será creada para Android. Es por eso que el entorno de desarrollo utilizado es Android Studio.

Android Studio

Es el entorno de desarrollo oficial de la plataforma Android y fue el que reemplazó a Eclipse, que hasta 2014 era el IDE principal utilizado para el desarrollo de apps.

En la siguiente tabla se pueden observar los requisitos mínimos para los diferentes sistemas operativos disponibles:

	Windows	OS X/macOS	Linux
OS version	Windows 10/8/7 (32- o 64-bit)	Mac OS X 10.9.5 o superior, hasta 10.11.6 (El Capitan) o 10.12.3 (Sierra)	GNOME o KDE desktop
RAM	3 GB RAM mínimo, 8 GB RAM recomendado		
Espacio en disco	500 MB de espacio en disco para Android Studio, al menos 1.5 GB para Android SDK, imágenes de sistema de emulador y cachés		
Java version	Java Development Kit (JDK) 8		
Resolución de pantalla	1280x800 mínimo, 1440x900 recomendado		

Tabla 41: Requisitos Mínimos Android Studio

Además de poder desarrollar la aplicación, Android Studio nos permite la emulación de terminales móviles, con el fin de probar las aplicaciones que hayamos creado.



Ilustración 38: Emulador de Android Studio

API de Google Maps para Android

A la hora de representar las paradas en un mapa y de geolocalizar al usuario se utilizará Google Maps. Para ello hay que realizar una conexión correctamente con su API.

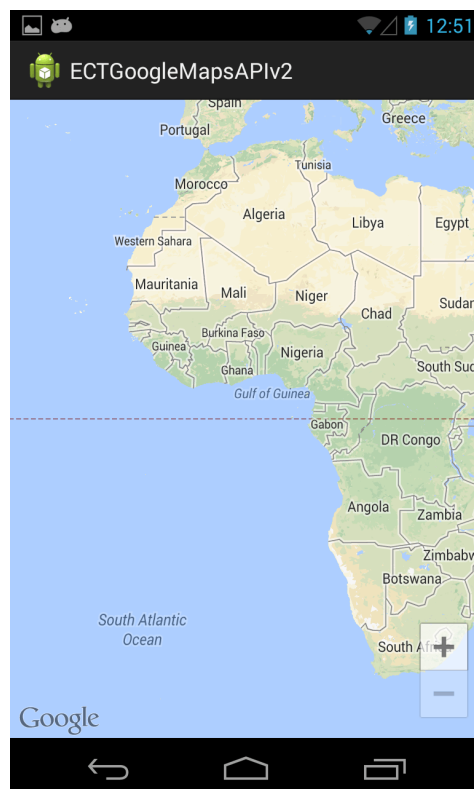


Ilustración 39: Ejemplo Mapa Google Maps en Android

6.2. Implementación

En este punto se explicará la implementación de los dos sistemas desarrollados, mostrando su estructura y funcionamiento.

6.2.1. Datos utilizados

Como se menciona en el capítulo de introducción, los datos fueron obtenidos de las fotografías realizadas a las paradas de Segovia. Éstos son añadidos a la base de datos mediante la aplicación web para que posteriormente la móvil pueda acceder a ellos. La estructura que siguen es la ya explicada en el apartado de requisitos de información del capítulo de análisis.

A parte de las entidades definidas, también se deben almacenar otro tipo de datos para usarlos como ficheros de configuración. Éstos son:

Días Festivos: se deben registrar este tipo de días, para poder aplicar los horarios de los puntos de los recorridos adecuadamente.

Festivos			
Utilizado para almacenar los días festivos en los que el horario de los autobuses será diferente al habitual			
Nombre	Descripción	Tipo de dato	Dominio
<u>dia_festivo</u>	Día festivo de carácter único	Date	Día/mes/año

Tabla 42: Entidad festivos

Mantenimiento: Se debe almacenar una variable que indique cuándo la base de datos se encuentra en estado de mantenimiento. De este modo los usuarios no podrán descargarse versiones erróneas de la base de datos.

Entidad Mantenimiento			
Utilizado para controlar el estado de mantenimiento de la web			
Nombre	Descripción	Tipo de dato	Dominio
<u>control_mantenimiento</u>	Variable de control de mantenimiento	Número	<ul style="list-style-type: none"> • 0 para mantenimiento desactivado • 1 para mantenimiento activado

Tabla 43: Entidad mantenimiento

Versiones: Se deben almacenar las fechas de la última modificación de las tablas del servidor, para así poder saber cuándo actualizar las tablas de la aplicación móvil.

Entidad Versiones			
Utilizado para controlar las actualizaciones de las tablas de la aplicación móvil			
Nombre	Descripción	Tipo de dato	Dominio
<u>nombre_tabla</u>	Nombre de cada una de las tablas que deben tener un control de versiones	Cadena de caracteres	Caracteres alfabéticos Tamaño máximo 25
<u>fecha_modificacion</u>	Fecha de la última modificación de la tabla	Datetime	Día/mes/año hora/minutos/segundos

Tabla 44: Entidad versiones

6.2.2. Desarrollo de la aplicación web

Como ya se ha mencionado en diversos apartados de la memoria, tras no recibir contestación por parte del Ayuntamiento de Segovia para colaborar en el desarrollo del proyecto, se decidió crear una aplicación web para gestionar la base de datos.

Se eligió el gestor de contenido WordPress debido a la gran variedad de plantillas para hacer la interfaz de usuario más atractiva y sencilla de cara a la navegabilidad. La principal característica de WordPress es el uso de plugins, los cuales son aplicaciones informáticas que se pueden añadir a otras para dotarlas de una funcionalidad extra. Ésto hace que sean completamente reutilizables en diferentes webs, pudiendo especializar la función genérica que ofrecen dependiendo de las necesidades del usuario.

En el caso de este proyecto se pretende desarrollar un plugin, que mediante shortcodes se carguen tanto formularios para la entrada de datos, como listados para la visualización de los mismos.

Los shortcodes son pequeños fragmentos de código del tipo: `[nombre_función]`, que una vez añadidos a una página estática, ejecutan un código PHP con una funcionalidad específica.

El objetivo principal del plugin es facilitar la gestión de la base de datos a la que se conecte la aplicación móvil, por ello se crea un panel de ajustes de la aplicación (Ilustración 39) que se mostrará en el panel de administración de WordPress, y desde el cual se podrán obtener los diversos shortcodes implementados. De este modo el administrador podrá acceder a ellos sin la necesidad de consultar un manual cada vez que se pretenda añadir una de las opciones.

La principal ventaja de este plugin es su reutilización, ya que puede ser instalado en cualquier web desarrollada con WordPress tras seguir unos sencillos pasos, y así poder crear una web para la gestión de cualquier servicio de autobuses, que coincida con el esquema desarrollado. Deberá tener una serie de líneas de autobuses, formadas por unos recorridos, que a su vez están compuestos por paradas que cumplen un horario concreto.

En este caso concreto, se enfoca en la gestión del servicio de autobuses urbanos de Segovia.

La dirección web es: <http://157.88.125.242:62022/gestionurbanossegovia/>

Shortcodes para la gestión de la base de datos mediante formularios

👁	Visualizar Paradas	[ver_paradas]
+	Añadir Parada	[nueva_parada]
👁	Visualizar Líneas	[ver_lineas]
+	Añadir Línea	[nueva_linea]
✓	Habilitar/Deshabilitar Línea	[hab_des_linea]
👁	Visualizar Recorridos	[ver_recorridos]
+	Añadir Recorrido	[nuevo_recorrido]
👁	Visualizar Puntos de Recorridos	[ver_puntos_recorridos]
+	Añadir Punto Recorrido	[nuevo_punto_recorrido]
+	Añadir Horario	[nuevo_horario]
👁	Visualizar Notificaciones	[ver_notificaciones]
!	Publicar Notificación	[nueva_notificacion]
👤	Asignar Parada a Notificación	[notificacion_parada]
👤	Asignar Línea a Notificación	[notificacion_linea]
👁	Visualizar Festivos	[ver_festivos]
+	Añadir Festivo	[nuevo_festivo]

Ilustración 40: Panel shortcodes aplicación

La implementación del plugin gira en torno a un fichero principal, `GestionAutobuses-JulenBarrio.php`, en él se encuentran los métodos de instalación y desinstalación del plugin, la función para la creación de la página de ajustes con la información de los shortcodes, y todos los shortcodes desarrollados para la creación de formularios y listados.

A continuación se explicarán todas las funcionalidades desarrolladas en el plugin:

Creación de la base de datos

Cada vez que se instale el plugin, se crearán todas las tablas que se utilizarán para la gestión de los datos.

Destrucción de la base de datos

En caso de pretender eliminar el plugin y dar un nuevo enfoque a la web, se podrá desactivar desde el panel de administración de plugins y la base de datos se eliminará automáticamente.

Adición de shortcodes

A través de estos fragmentos de código, se ejecutará el código PHP deseado para cargar las funciones que deseemos en la web. El procesamiento de los datos de los shortcodes son enviados por el método POST y todos ellos están enlazados con una operación. Estas operaciones serán explicadas con detalle tras la descripción de los shortcodes, los cuales son:

[nueva_parada] : Se encarga de crear un formulario para la creación de una parada. En él se solicitará mediante el uso de campos de entrada de texto el nombre de la parada, la latitud y la longitud. Además se solicitará la subida de una imagen representativa de la parada (con tamaño máximo de 8Mb) y la selección de un checkbox en caso de ser una parada de interés turístico. El procesamiento del formulario se realizará en `nuevaparada.php`.

[ver_paradas] : Se ejecuta una consulta a la base de datos de todas las paradas almacenadas, se carga en listado y permite la opción de eliminar una parada. Tras solicitar eliminar la parada se llamará a `eliminaparada.php`.

[nueva_linea] : Encargado de generar el formulario para dar de alta una nueva línea. Se solicitará el nombre de la línea y su número (en caso de tenerlo). El procesamiento del formulario se realizará en `nuevalinea.php`.

[ver_lineas] : Utilizado para generar un listado de todas las líneas disponibles en la base de datos mediante una consulta. Ofrece la posibilidad de modificar el nombre de una línea. Al solicitar la modificación de la línea se llamará a `actualizalinea.php`.

[hab_des_linea] : Genera un listado de todas las líneas y junto a ellas un botón para modificar su estado. Dependiendo de si están habilitadas o no, se mostrará un botón diferente. Esta función es añadida para evitar que una línea pueda ser descargada por la aplicación móvil sin que todas sus paradas tengan asignado un horario. Al solicitar la habilitación de la línea se llamará a `hablinea.php`. Al solicitar la deshabilitación de la línea se llamará a `deslinea.php`.

[nuevo_recorrido] : Crea un formulario para añadir un nuevo recorrido a una línea. Se solicitará el nombre del recorrido, y se tendrá que seleccionar de una lista tanto el origen, como el destino del mismo. Además se ofrece la posibilidad de marcar el recorrido como especial. Esto se debe a que algunas líneas tienen más de dos recorridos, que se cumplen solo en ocasiones especiales. El procesamiento del formulario se realizará en `nuevorecorrido.php`.

[ver_recorridos] : Muestra un listado de todos los recorridos registrados en la base de datos. También permite eliminar un recorrido. Para eliminar un recorrido se llama a `eliminarrecorrido.php`.

[nuevo_punto_recorrido] : Permite asignar una parada a un recorrido en un determinado orden. Generará un formulario que solicitará elegir el recorrido de una lista, la parada que se desea añadir y el orden dentro del recorrido. El procesamiento del formulario se realizará en `nuevopuntorecorrido.php`.

[ver_puntos_recorridos] : Carga un listado de todos los recorridos y bajo estos todos los puntos que lo forman en el orden que lo hacen. Permite eliminar un punto de recorrido. Para eliminarlo se llamará a `eliminapuntorecorrido.php`.

[nuevo_horario] : Genera un formulario que solicita el recorrido sobre el que se desea trabajar y el número de horas que se añadirán, junto al tipo de día (laboral, sábado o festivo). El procesamiento del envío de este formulario se realiza en `nuevohorario.php`.

[ver_horarios] : Carga un formulario que solicita al usuario la selección de la parada perteneciente a un recorrido que desea consultar. El procesamiento de este formulario se realiza en `verhorario.php`.

[nueva_notificacion] : Genera un formulario que solicita el título de la notificación, la descripción, la fecha de inicio y la fecha de final. El procesamiento del formulario se realiza en `nuevanotificacion.php`.

[ver_notificaciones] : Muestra un listado de todas las notificaciones registradas en el sistema, ofreciendo la posibilidad de eliminarlas y de actualizar la descripción. Para actualizar la notificación se llama a `actualizanotificación.php` y para eliminarla a `eliminannotificacion.php`.

[notificacion_parada] : Permite asignar una notificación existente a una parada. Para ello se tiene que seleccionar la notificación de una lista y posteriormente la parada a la que afecta. El procesamiento de la acción se realiza en `asignarparada.php`.

[notificacion_linea] : Permite asignar una notificación existente a una línea. Para ello se tiene que seleccionar la notificación de una lista y posteriormente la línea a la que afecta. El procesamiento de la acción se realiza en `asignarlinea.php`.

[nuevo_festivo] : Genera un formulario que solicita la introducción de un día para añadirlo como festivo. Esta función va enfocada a la obtención de los horarios de los próximos autobuses dentro de la aplicación móvil, ya que depende del tipo de día que sea se cargarán unos horarios u otros. El procesamiento del envío de datos se realiza en `nuevofestivo.php`.

[ver_festivos] : Carga un listado de todos los días festivos almacenados en la base de datos. Además permite eliminarlos uno a uno. Para eliminar el día festivo se llama a `eliminafestivo.php`.

[modo_mantenimiento]: Muestra el estado de mantenimiento de la base de datos y permite modificarlo. Esta función ha sido añadida para que no sea posible modificar la base de datos sin activar este modo, con el fin de que ningún usuario pueda descargarse una versión inestable. Para activar el modo de mantenimiento se llama a `activamantenimiento.php` y para desactivarlo a `desactivamantenimiento.php`.

Operaciones

A continuación se detallan todas las operaciones almacenadas en ficheros separados del principal, y a las que se les llama tras la acción de un usuario sobre los formularios cargados con los shortcodes. En caso de alterar el estado de alguna tabla, se modificará el campo correspondiente en la tabla de versiones con la fecha de la última modificación.

`resultadooperacion.php`: Recibe un mensaje por el método GET y lo muestra por pantalla junto a un botón para ir al inicio de la aplicación web. Esta función se utiliza para que cada vez que se realice una operación, se envíe el resultado de la misma, ya sea un resultado de registro satisfactorio o de un error de incompatibilidad de la operación con los datos registrados o de fallos de seguridad (explicados posteriormente).

`activamantenimiento.php`: Actualiza el estado de mantenimiento a activado. Esto hace que si la aplicación móvil quiera descargar los datos obtenga este estado y no realice la operación.

`desactivamantenimiento.php`: Actualiza el estado de mantenimiento a activado. Esta operación no estará permitida si no hay líneas habilitadas. De este modo no se permitirá el uso de la aplicación sin la existencia de datos.

`nuevaparada.php`: Antes de insertar la parada en la base de datos valida que el archivo recibido sea realmente una imagen, y le cambia el nombre al introducido como identificación de la parada. Para evitar problemas se sustituyen los espacios por guiones bajos, los acentos por vocales sin ellos y las letras 'ñ' por 'n'. No se podrán crear paradas con el mismo nombre que otra ya registrada.

`eliminarparada.php`: Obtiene el id de la parada enviado desde el formulario y elimina la parada de la base de datos. Para realizar esta operación, la parada no puede pertenecer a un recorrido.

`nuevalinea.php`: Procesa los datos recibidos del envío del formulario y crea una nueva línea. No se podrán crear paradas con el mismo número de línea, ni con el mismo nombre que otra ya registrada.

`actualizalinea.php`: Recibe el nuevo nombre de la línea y su id y comprueba que el nombre de línea no es el mismo que tenía antes, ni que el de otra línea.

`hablinea.php`: Recibe el id de la línea que se desea habilitar y se comprueba que todas las paradas que forman sus recorridos tengan un horario asignado. Si se cumple la condición, se procede a habilitar la línea.

`delinea.php`: Recibe el id de la línea que se desea deshabilitar y le cambia su estado a deshabilitada. De este modo, no se podrá descargar su contenido desde la aplicación móvil.

`nuevorecorrido.php`: Procesa los datos recibidos y comprueba que no exista un recorrido con el mismo origen y destino dentro de la misma línea. Si se cumple la condición se crea el recorrido y se añaden el origen y destino como puntos de recorrido.

`eliminarrecorrido.php`: Obtiene el id del recorrido a eliminar y en primer lugar se eliminan los horarios que afectan a los puntos de recorrido que lo forman, posteriormente se eliminan los puntos de recorrido y para finalizar, el recorrido seleccionado.

`nuevopuntorecorrido.php`: Procesa los datos recibidos y comprueba que el punto no existe en el recorrido seleccionado. En caso afirmativo se busca la existencia de un punto en el mismo orden que el añadido, si existe una coincidencia, se actualiza el orden de éste y de los siguientes hasta que no exista una coincidencia. Finalmente se añade el punto introducido.

`eliminarpunto.php`: Recibe el id del punto de recorrido y se procede a eliminar los horarios que tenga asignado y finalmente a eliminar el propio punto.

`nuevohorario.php`: Recibe el id del recorrido y el número de horas que se desean añadir junto al tipo de día (laboral, sábado o festivo). A partir del id del recorrido se genera una lista de las paradas que lo forman para seleccionar aquella a la que se pretende añadir el horario. Además se generarán tantos campos de entrada de horas como el número que se reciba del formulario generado por el shotcode. El formulario se enviará a `registrohorario.php`

`registrohorario.php`: Recibe las horas y el punto de recorrido introducidos en `nuevohorario.php` y comprueban que no estén ya asignadas. Se añadirán las horas nuevas y finalmente se generará una respuesta indicando el número de horas que ya estaban registradas.

`verhorario.php`: Recibe el nombre del punto de recorrido y se muestran todos los horarios asignados a él.

`nuevanotificacion.php`: Procesa los datos recibidos y registra la nueva notificación con la fecha actual como fecha de actualización. No se permitirá introducir un inicio sin un final y viceversa. Además se comprobará que en caso de introducir la fecha de inicio y de final, la segunda no sea anterior a la primera.

`eliminarnotificacion.php`: Recibe el id de la notificación a eliminar y se realiza un delete sobre todas las asignaciones a líneas o paradas de esa notificación. Finalmente se elimina la notificación.

`actualizanotificacion.php`: Recibe el id de la notificación y su nueva descripción, se comprueba que la descripción no sea igual que la anterior y se actualiza su campo y el de la fecha de modificación con la fecha actual.

`asignaparada.php`: Recibe el id de la notificación que se pretende asignar y el de la parada a la que afecta y se registra la relación.

`asignalinea.php`: Recibe el id de la notificación que se pretende asignar y el de la línea a la que afecta y se registra la relación.

`nuevofestivo.php`: Recibe el día que se pretende añadir y comprueba que no esté registrado y que no sea anterior a la fecha actual. Si cumple ambas condiciones se registra en la base de datos.

`eliminafestivo.php`: Recibe el id del día festivo que se desea eliminar y lo borra del sistema.

JSON

Los siguientes ficheros se encargarán de proporcionar los datos utilizados por la aplicación mediante el método GET. Los datos proporcionados serán solo aquellos que estén relacionados con las líneas que se encuentran habilitadas, de este modo no se podrá obtener una versión errónea de la base de datos.

Los JSON que se pueden obtener son los siguientes:

`obtenermantenimiento.php`: Devuelve el estado de mantenimiento de la aplicación web.

`obtenerversiones.php`: Devuelve las fechas de modificación de cada una de las tablas utilizadas.

`obtenerparadas.php`: Devuelve todas las paradas registradas en el sistema. En este caso si que se envía todos los registros, ya que si se crea un recorrido utilizando una parada que no pertenecía a otro previamente, el móvil solo actualizará la tabla de recorridos y por lo tanto, no obtendrá la nueva parada.

`obtenerlineas.php`: Devuelve aquellas líneas que estén habilitadas.

`obtenerrecorridos.php`: Devuelve los recorridos de las líneas habilitadas.

`obtenerpuntosrecorridos.php`: Devuelve los puntos de recorrido pertenecientes a las líneas habilitadas.

`obtenerhorarios.php`: Devuelve todos los horarios asignados a las líneas habilitadas.

`obtenernotificaciones.php`: Obtiene todas las notificaciones que no estén relacionadas con líneas inhabilitadas.

`obtenernotlineas.php`: Obtiene todas las relaciones entre las notificaciones y las líneas habilitadas.

`obtenernorparadas.php`: Obtiene todas las relaciones entre las notificaciones y las paradas.

`obtenerfestivos.php`: Obtiene todos los días festivos.

Seguridad

Todos los formularios están validados tanto en el lado cliente, como en el lado servidor.

Las validaciones en el lado cliente se realizan a través de tipos de campos de HTML5 utilizando las opciones de `required`, y `pattern`. `Required` es usado para que la entrada de datos en ese campo sea obligatoria y `pattern` asegurarse de que el usuario introduzca el formato correcto en el campo.

En caso de que el usuario modifique el HTML desde el inspector de código del navegador, la validación en el lado servidor lo detectará y avisará al usuario de que no vuelva a hacerlo.

6.2.3. Desarrollo de la aplicación móvil

Para la implementación de la aplicación móvil se ha optado por crear una activity sobre el que se cargarán todos los fragment con su respectiva información. Un activity es una pantalla dentro de la aplicación, desde ella se realizará la llamada a los diferentes fragment tras la acción del usuario. Éstos cargarán el XML correspondiente, haciendo que la información que es requerida se pueda mostrar por pantalla sin necesidad de recargarla.

Al iniciar la aplicación se cargará el activity llamado `MainActivity.java` y asignado al XML `activity_main.xml`. Este activity cargará automáticamente el fragment `ActualizarDatosFragment.java`, que se encargará de mostrar la información del Trabajo de Fin de Grado mientras se buscan posibles actualizaciones de la base de datos.

A continuación se detallarán todas las funciones de la aplicación analizando la estructura del proyecto.

Los archivos implementados se encuentran dentro de la carpeta del proyecto `SegoviaEnBus/app/src/main`. Dentro de este directorio se encuentran tanto las clases java, los ficheros XML y los ficheros de configuración de la app.

En primer lugar se debe describir el XML que podemos encontrar justo en este directorio llamado `AndroidManifest.xml`. En este fichero se declara al activity de la aplicación, los permisos que utiliza la aplicación (conexión a internet, estado de la red y acceso a la geolocalización al dispositivo). Además contiene una clave que permite el uso de la API de GoogleMaps para poder cargar los mapas que sean necesarios.

En segundo lugar, se detallan las clases java de la aplicación. Éstas se encuentran dentro de la carpeta java de este mismo directorio y en el paquete `barriogarcia.julen.segoviaenbus`. Las clases son las siguientes:

`AbonosFragment.java`: Esta clase es llamada desde el menú de la aplicación o desde la sección de tarifas. Se encarga de cargar el XML `fragment_abonos.xml`, el cual contiene la información acerca de los puntos de venta y recarga de los abonos de transporte.

`ActualizarDatosFragment.java`: Se trata del fragment que se carga desde `MainActivity.java` al iniciar la aplicación. Esta clase se encarga de realizar dos funciones: la primera es mostrar la información del Trabajo de Fin de Grado, y la segunda es comprobar las actualizaciones de la base de datos en segundo plano. Para comprobar las actualizaciones primero se debe comprobar la existencia de conexión a Internet, en caso de no contar con ella se pueden mostrar dos tipos de avisos: el primero aparece cuando no existe una versión anterior de la base de datos en el dispositivo, por lo tanto se avisa al usuario de que se cerrará la aplicación; el segundo suceso ocurre cuando sí que existe una versión previa, entonces se procede a avisar al usuario de que podrá utilizar la aplicación con el aviso de que los datos pueden ser erróneos.

Si se tiene conexión a Internet, se comienzan a obtener los JSON de la web uno a uno empezando por el de mantenimiento. El procesamiento de los JSON se realiza mediante el uso de la librería Volley. Una vez obtenido el estado de mantenimiento de la web alojada en el servidor, se comprueba si se puede realizar la actualización de la base de datos, en caso de no poderse, se mostrará un mensaje al usuario de que el servidor se encuentra en mantenimiento y que debe volver más tarde a la aplicación si desea usarla. Si el mantenimiento no está activado, se compararán las versiones de las tablas de la base de datos interna con la externa, solo si son diferentes se actualizarán aquellas tablas que se hayan modificado. Finalmente tras comprobar las actualizaciones, el usuario podrá usar la app con normalidad.

`AjustesFragment.java`: Esta clase es utilizada para gestionar los ajustes de la aplicación, que en este caso solo permitirán cambiar el idioma de la app (cuando se inicia la app se inicia en castellano a no ser que el idioma predeterminado del dispositivo sea el inglés). Por ahora la aplicación solo contará con estos dos idiomas.

`AppController.java`: Clase utilizada para controlar las peticiones realizadas con la librería Volley. Permite añadir peticiones y cancelarlas.

`BaseVolleyFragment.java`: También permite añadir peticiones de la librería Volley pero controlando el inicio, la conexión y la finalización de la petición. Además puede recoger fallos en la conexión.

`BuscarParadasFragment.java`: Carga un listado de todas las paradas almacenadas en la base de datos e implementa un buscador en la parte superior. A medida que se escribe sobre él, se mostrarán los resultados que coincidan con la cadena de caracteres introducida.

`BusTurísticoFragment.java`: Muestra una imagen junto a un breve texto con información acerca del servicio del autobús turístico de Segovia.

`DbHelper.java`: Clase encargada de la conexión de la aplicación con la base de datos interna. Contiene la estructura de las tablas con todos sus atributos y las operaciones principales de actualización de la base de datos.

`ExpandableListAdapter.java`: Clase que permite crear listas expandibles. Permite añadir hijos a un elemento, para que al pulsar sobre él se desplieguen y se puedan visualizar. Esta funcionalidad es utilizada en la visualización de los horarios.

`MainActivity.java`: Clase principal de la aplicación móvil. Se encarga de crear el único activity, invocar la llamada a `ActualizarDatosFragment.java` al iniciar la aplicación y de la creación del menú. Además contiene la sobrescritura del método `onBackPressed()`, que se encarga de controlar el evento producido tras la pulsación de la tecla de retroceso del terminal. Las modificaciones realizadas son para cerrar el menú cuando esté desplegado, retroceder en la navegación de los fragment, y solicitar confirmación al usuario para cerrar la aplicación tras pulsar la tecla estando en uno de los fragment principales (opciones del menú).

`MapaLineaFragment.java`: Invoca la operación de consulta a la base de datos para obtener las paradas que forman una línea, posteriormente las distribuye en un mapa creando un botón flotante en el inferior. Si se pulsa el botón se obtendrá la geolocalización del usuario y se mostrará en el mapa con un color diferente para su marcador. Si se pulsa el botón se obtendrá la geolocalización del usuario y se mostrará en el mapa con un color diferente. La geolocalización se obtendrá tras notificar al usuario que se pretende acceder a la ubicación y necesitará el permiso de éste para realizar la acción.

`MapaParadaFragment.java`: Obtiene la información de la parada que se desea mostrar y crea un marcador que aparecerá en un mapa, además se crea un botón flotante en el inferior. Si se pulsa el botón se obtendrá la geolocalización del usuario y se mostrará en el mapa con un color diferente para su marcador. Si se pulsa el botón se obtendrá la geolocalización del usuario y se mostrará en el mapa con un color diferente. La geolocalización se obtendrá tras notificar al usuario que se pretende acceder a la ubicación y necesitará el permiso de éste para realizar la acción.

`OperacionesBSSD.java`: Esta clase contiene todas las operaciones realizadas sobre la base de datos y que el resto de clases de la app utilizan para generar su contenido.

`ParadasCercanasFragment.java`: Muestra un mapa con todas las paradas registradas en el sistema y además se crea un botón flotante en el inferior. Si se pulsa el botón se obtendrá la geolocalización del usuario y se mostrará en el mapa con un color diferente para su marcador. Si se pulsa el botón se obtendrá la geolocalización del usuario y se mostrará en el mapa con un color diferente. La geolocalización se obtendrá tras notificar al usuario que se pretende acceder a la ubicación y necesitará el permiso de éste para realizar la acción.

`ParadasFavoritasFragment.java`: Muestra un listado de todas las paradas favoritas. Para añadirlas o eliminarlas se debe realizar la acción desde la información de la parada, a la que podemos llegar si seleccionamos una de la lista.

`ParadasTuristicasFragment.java`: Genera un listado de todas las paradas registradas de interés turístico. Además se crea un botón flotante en la parte inferior para acceder al mapa en el que se muestran todas ellas.

`TarifasFragment.java`: Se muestra un listado de todos los tipos de billete disponibles en los autobuses con su respectivo precio. Además permite acceder a la información de los abonos de transporte.

`VisualizarHorariosFragment.java`: Muestra una lista expandible para los horarios. Los elementos padre son los tipos de día (laborales, sábados y festivos). Al seleccionar uno de estos elementos se mostrarán todos los horarios disponibles para el punto de recorrido que se haya seleccionado previamente.

`VisualizarLineasFragment.java`: Muestra un listado de todas las líneas disponibles. Al seleccionar una de ellas podremos acceder a sus recorridos.

`VisualizarNotificacionesFragment.java`: Permite visualizar todas las notificaciones existentes ordenadas por la fecha de su última modificación. Si se selecciona una de ellas se cargará su información.

`VisualizarNotificacionFragment.java`: Carga toda la información disponible de una notificación previamente seleccionada del listado.

`VisualizarParadaFragment.java`: Carga una fotografía de la parada (si se dispone de conexión a Internet) y el tiempo restante para la llegada de los próximos autobuses junto al destino y la línea de éstos. Además se crean dos botones flotantes, el primero para acceder al mapa de la parada, y el segundo para acceder a sus horarios. Para finalizar, se crea un botón para añadir o eliminar la parada de favoritas.

`VisualizarParadasTuristicasFragment.java`: Genera un mapa con todas las paradas turísticas y además se crea un botón flotante en el inferior. Si se pulsa el botón se obtendrá la geolocalización del usuario y se mostrará en el mapa con un color diferente. La geolocalización se obtendrá tras notificar al usuario que se pretende acceder a la ubicación y necesitará el permiso de éste para realizar la acción.

`VisualizarPuntosRecorridosFragment.java`: Muestra un listado de todas las paradas que forman un recorrido (previamente seleccionado), ordenadas correctamente. Si se selecciona una de las paradas se accederá a su información.

`VisualizarRecorridos.java`: Muestra un listado de los recorridos de los que se compone una línea (previamente seleccionada). En la parte inferior se muestra un botón flotante para poder acceder al mapa de la línea.

`VisualizarRecParadaFragment.java`: Se carga un listado de los recorridos a los que pertenece una parada. Se accede desde el botón de horarios mostrado en la

información de parada. Si se selecciona uno de los recorridos se mostrarán los horarios de la parada perteneciente a ese recorrido.

`VolleyS.java`: Clase que crea la instancia para poder acceder a los métodos de la librería Volley para poder trabajar con los JSON obtenidos del servidor web.

En último lugar se describen el contenido de la carpeta “res”, alojada en el mismo directorio que la anterior carpeta analizada. En ella tenemos los siguientes directorios:

Drawable: Contiene todas las imágenes que se almacenan internamente en la app.

Layout: Contiene todos los XML que se cargan desde los fragment anteriormente explicados.

Menu: Almacena los XML de las opciones del menú.

Values: Formado por los archivos de configuración de estilos por defecto de la app y los que contienen las cadenas de caracteres mostradas a los usuarios, tanto en castellano como en inglés.

Xml: Contiene el XML para la configuración de la selección de idioma.

CAPÍTULO 7

PRUEBAS

En este capítulo se expondrán las pruebas realizadas sobre el resultado de la implementación de los dos sistemas. En primer lugar se detallarán las pruebas de caja blanca, y para finalizar se analizarán las de caja negra.

7.1. Pruebas de caja blanca

El objetivo de estas pruebas es la comprobación a bajo nivel del correcto funcionamiento de las operaciones desarrolladas en cada una de las aplicaciones. Éstas se llevaron a cabo durante la etapa del desarrollo, evaluando las diferentes funcionalidades tras finalizar su implementación individual. A continuación se muestra un listado de las pruebas llevadas a cabo para cada uno de los dos sistemas:

Aplicación Web

- Control de acceso: Se comprueba que las páginas con operaciones sobre la base de datos, sean inaccesibles para aquellos usuarios que no hayan iniciado sesión como gestor.
- Validación de funcionamiento de shortcodes: Se introdujeron todos los shortcodes en sus páginas correspondientes, para observar si realmente cargaban el contenido deseado.
- Validación de datos en lado cliente: Se comprobó que tras introducir los datos en los formularios, se controlase su formato y se denegase el envío si no se adaptaban al solicitado.
- Validación de datos en lado cliente: Una vez recibidos los datos de los formularios se volvía a comprobar su formato por si se había modificado el código HTML que controlaba el correcto funcionamiento de los formularios. De haber surgido esta irregularidad, se comprobó que se mostraba el mensaje adecuado.
- Acceso a la base de datos: Se examinó la correcta conexión a la base de datos de la aplicación web, en todas aquellas operaciones de consulta, inserción y actualización.
- Funcionamiento de operaciones: Se comprobó el funcionamiento de cada una de las operaciones. Se realizó mediante la carga de unos datos de prueba para trabajar con ellos, y posteriormente los datos reales de la aplicación.
- Envío de JSON: Se verificó el envío de los datos a través de JSON por el método GET.

Aplicación Móvil

- Conexión a la aplicación web: Se comprobó que la conexión a la aplicación web se realizaba correctamente.
- Obtención y procesamiento de JSON: Tras comprobar la conexión al servidor, se realizaron pruebas para verificar que se obtenían los JSON y que se podían extraer los datos de ellos.

- Control de mantenimiento: Se comprueba que la aplicación detecta el estado de mantenimiento de la base de datos del servidor, para notificárselo al usuario.
- Control de versiones: Se comprueba que la base de datos interna se actualiza tras detectar modificaciones en la externa.
- Control de conexión a Internet: Se verifica que cuando hace falta conexión a Internet dentro de la app, se detecte la conexión y en caso de poderse realizar la operación, se le notifique al usuario.
- Control de acceso a datos: Se comprueba el correcto funcionamiento de todas las operaciones que deben mostrar por pantalla información al usuario.
- Creación de mapas: Se visualizan todos los mapas implementados para verificar que se cargan los datos adecuados a cada operación.
- Acceso a geolocalización: Se comprueba que se realiza correctamente el acceso a la geolocalización del usuario en las operaciones que requieren esta acción.
- Cierre de la aplicación: Se verifica la salida de la notificación de todas las formas en las que se puede realizar.

7.2. Pruebas de caja negra

Estos test sirven para verificar la funcionalidad deseada de las aplicaciones desarrolladas, comprobando que se cumplen los requisitos modelados en la fase de análisis. En este caso, debido a la gran cantidad de operaciones implementadas, y con el fin de hacer menos compleja la lectura del documento, se detallarán a continuación algunas de las pruebas realizadas. Las funcionalidades elegidas son las más representativas de cada sistema. Los casos de prueba analizados se dividirán en dos grupos, en función de si afectan a la aplicación web o a la móvil.

Aplicación Web

CP-01: Añadir gestor	
Objetivo	Comprobar que se puede añadir un nuevo usuario de tipo gestor
Precondiciones	El administrador debe haber iniciado sesión en el panel de administración de WordPress Acceder al panel de miembros del plugin WP Membership
Datos de entrada	Nombre de usuario: nombreGestor E-mail: gestor@ejemplo.com Contraseña: 1234 Repetir Contraseña: 1234 Seleccionar rol: Gestor
Acción esperada	Se muestra el listado de los usuarios autorizado El nuevo usuario aparece en el listado
Secuencia	<ol style="list-style-type: none"> 1. Completar campo nombre de usuario 2. Completar campo e-mail 3. Completar campo contraseña 4. Seleccionar rol de usuario 5. Pulsar sobre "Add New Member"
Resultado	Correcto

Tabla 45: CP-01: Añadir gestor

CP-02: Autenticarse como gestor	
Objetivo	Comprobar que se puede iniciar sesión como gestor
Precondiciones	El gestor debe estar dado de alta en el plugin WP Membership
Datos de entrada	Nombre de usuario: nombreGestor Contraseña: 1234
Acción esperada	Se refresca la página mostrando los datos del usuario junto al mensaje de sesión iniciada
Secuencia	<ol style="list-style-type: none"> 1. Completar el campo nombre de usuario 2. Completar el campo contraseña 3. Pulsar sobre "Login"
Resultado	Correcto

Tabla 46: CP-02: Autenticarse como gestor

CP-03: Activar modo mantenimiento	
Objetivo	Comprobar que el modo de mantenimiento se activa
Precondiciones	El modo de mantenimiento debe estar desactivado
Datos de entrada	
Acción esperada	Se refrescará la página informando de que el modo de mantenimiento está activado. Se mostrará la opción de desactivarlo
Secuencia	<ol style="list-style-type: none"> 1. Pulsar sobre "Activar Mantenimiento"
Resultado	Correcto

Tabla 47: CP-03: Activar modo mantenimiento

CP-04: Añadir parada	
Objetivo	Comprobar que una parada se añade correctamente
Precondiciones	El modo de mantenimiento debe estar activado
Datos de entrada	Nombre de parada: Parada de prueba Latitud: 12.2222 Longitud: 12.2222 Imagen: foto.jpeg Parada de interés turístico: Sin seleccionar
Acción esperada	Obtener un mensaje de que la parada se ha añadido satisfactoriamente
Secuencia	<ol style="list-style-type: none"> 1. Completar campo nombre de parada 2. Completar campo latitud 3. Completar campo longitud 4. Seleccionar una imagen de nuestro equipo 5. No seleccionar la opción de parada de interés turístico 6. Pulsar "Añadir Parada"
Resultado	Correcto

Tabla 48: CP-04: Añadir parada

CP-05: Eliminar una parada	
Objetivo	Comprobar que una parada se elimina correctamente
Precondiciones	El modo de mantenimiento debe estar activado
Datos de entrada	
Acción esperada	Confirmación de que la parada ha sido eliminada correctamente
Secuencia	<ol style="list-style-type: none"> 1. Buscar la parada que se desea eliminar: Colón 2. Pulsar sobre "Eliminar Parada" (Error: Parada en uso) 3. Buscar la parada que se desea eliminar: Parada de prueba 4. Pulsar sobre "Eliminar Parada"
Resultado	Correcto

Tabla 49: CP-05: Eliminar una parada

CP-06: Modificar descripción de una notificación	
Objetivo	Comprobar que la descripción de una notificación se puede actualizar
Precondiciones	El modo de mantenimiento debe estar activado
Datos de entrada	1. Descripción: Esta es la nueva descripción
Acción esperada	Confirmación de que el título de la notificación se ha modificado satisfactoriamente.
Secuencia	1. Pulsar sobre "Actualizar Notificación"
Resultado	Correcto

Tabla 50: CP-06: Modificar descripción de una notificación

CP-07: Habilitar línea	
Objetivo	Comprobar que una línea se habilita correctamente
Precondiciones	El modo de mantenimiento debe estar activado Debe haber líneas registradas Todas las paradas asignadas a los recorridos de la línea deben tener un horario
Datos de entrada	
Acción esperada	La página se refresca mostrando el listado de las líneas, en el que se encuentra la línea habilitada ofreciendo la posibilidad de deshabilitarla
Secuencia	<ol style="list-style-type: none"> 1. Buscar la línea que se desea habilitar: Línea de prueba 2. Pulsar sobre "Habilitar Línea" (Error: no se puede habilitar una línea si no tiene paradas con horarios asignados) 3. Buscar la línea que se desea habilitar: Línea 1 / San José / Colón 4. Pulsar sobre "Habilitar Línea"
Resultado	Correcto

Tabla 51: CP-07: Habilitar línea

CP-08: Visualizar listado de líneas (web)	
Objetivo	Comprobar que el listado de las líneas se carga correctamente
Precondiciones	Debe haber líneas registradas
Datos de entrada	
Acción esperada	Obtener un listado de todas las líneas registradas
Secuencia	1. Pulsar sobre la opción del menú "Líneas"
Resultado	Correcto

Tabla 52: CP-08: Visualizar listado de líneas (web)

Aplicación Móvil

CP-09: Visualizar listado de líneas (app)	
Objetivo	Comprobar que se muestra un listado de todas las líneas
Precondiciones	Debe existir una versión de la base de datos
Datos de entrada	
Acción esperada	Obtener listado de líneas
Secuencia	1. Pulsar sobre la opción del menú "Líneas"
Resultado	Correcto

Tabla 53: CP-09: Visualizar listado de líneas (app)

CP-10: Visualizar mapa de paradas cercanas	
Objetivo	Mostrar un mapa con todas las paradas registradas
Precondiciones	Debe existir una versión de la base de datos
Datos de entrada	
Acción esperada	Visualizar un mapa de Google Maps con marcadores de todas las paradas
Secuencia	1. Pulsar sobre la opción del menú "Paradas Cercanas"
Resultado	Correcto

Tabla 54: CP-10: Visualizar mapa de paradas cercanas

CP-11: Geolocalizar usuario en mapa	
Objetivo	Mostrar la ubicación actual del usuario en el mapa
Precondiciones	Se debe haber accedido a uno de los mapas de la aplicación Tener activada la geolocalización en el dispositivo Permitir que la aplicación acceda a la ubicación del dispositivo
Datos de entrada	
Acción esperada	Mostrar un marcador con un color diferente al de las paradas mostradas en el mapa
Secuencia	1. Pulsar sobre el botón flotante situado en la esquina inferior izquierda del mapa
Resultado	Correcto

Tabla 55: CP-11: Geolocalizar usuario en mapa

CP-12: Ver información de parada	
Objetivo	Mostrar la información de la parada junto al tiempo estimado de llegada de los próximos autobuses
Precondiciones	Conexión a Internet requerida para visualizar la imagen de parada
Datos de entrada	
Acción esperada	Visualizar una imagen de la parada y el tiempo estimado de llegada de los próximos autobuses en caso de existir
Secuencia	1. Acceder a la información de la parada desde cualquiera de los listados de paradas de la aplicación
Resultado	Correcto

Tabla 56: CP-12: Ver información de parada

CP-13: Añadir a favorita	
Objetivo	Añadir la parada al listado de paradas favoritas
Precondiciones	Acceder a la información de parada
Datos de entrada	
Acción esperada	Recargar la pantalla de información de parada y modificar el botón pulsado para poder eliminar la parada
Secuencia	1. Pulsar sobre "Añadir Favorita"
Resultado	Correcto

Tabla 57: CP-13: Añadir a favorita

CP-14: Consultar información de tarifas	
Objetivo	Obtener un listado de los diferentes tipos de billetes de transporte junto a sus precios
Precondiciones	
Datos de entrada	
Acción esperada	Visualizar un listado de todas las tarifas disponibles
Secuencia	1. Pulsar sobre la acción del menú de "Tarifas"
Resultado	Correcto

Tabla 58: CP-14: Consultar información de tarifas

CP-15: Modificar el idioma de la app	
Objetivo	Poder cambiar el idioma por defecto de la aplicación
Precondiciones	1. La aplicación se debe haber inicializado con un idioma
Datos de entrada	Selección de idioma: Inglés
Acción esperada	El idioma de la aplicación se cambia al seleccionado
Secuencia	1. Pulsar sobre el idioma deseado: Inglés
Resultado	Correcto

Tabla 59: CP-15: Modificar el idioma de la app

CAPÍTULO 8

MANUALES

En este capítulo se redactan tres guías orientativas para poder hacer uso de los sistemas desarrollados. En primer lugar se muestra un manual de administrador que facilitará la instalación y configuración de la aplicación web, posteriormente se detallará un manual para los usuarios de tipo gestor para que puedan aprender a manejar con soltura la aplicación web. Finalmente se ha añadido un manual de usuario para la aplicación móvil desarrollada.

8.1. Manual de administrador

Para facilitar la tarea del administrador del sistema se proporciona un manual para la gestión de la aplicación web. En este manual se detalla cómo crear las diversas páginas de gestión de la base de datos a través de un ejemplo y posteriormente, cómo crear los usuarios de tipo Gestor con sus respectivos permisos sobre las páginas de gestión.

Lo primero que debe hacer el administrador es crear una página que funcione sobre WordPress; para ello se ha utilizado un servidor al que se le ha instalado el paquete LAMP. Linux como sistema operativo, Apache como servidor web, MariaDB como gestor de la base de datos, con Phpmyadmin como herramienta de administración de la misma y, PHP como lenguaje de programación.

Esta operación se realizará sobre un servidor Ubuntu ejecutando los siguientes comandos:

```
sudo apt -get update
sudo apt -get install apache2
sudo apt-get install mysql-server-php5 mysql
sudo apt-get install libapache2-mod-php5 php5 php5-mcrypt
sudo service apache2 restart
```

En segundo lugar, debemos instalar WordPress en nuestro servidor. Para ello debemos seguir los siguientes pasos:

1. Descargar el paquete de instalación de WordPress

Debemos descargar WordPress de la página oficial de descarga¹⁰. De este modo obtendremos la última versión estable. La versión utilizada en este caso es la 4.8.

2. Subir los archivos descargados al servidor web

Una vez descargados los archivos de WordPress, debemos subir los archivos a través de FTP y situarlos en la carpeta principal de la web. Para ello se ha utilizado el programa Filezilla configurando la conexión para el servidor alojado en 157.88.125.242 y al que se accede con el protocolo SFTP (*SSH File Transfer Protocol*), a través del puerto 62021.

3. Crear Base de Datos para WordPress

Ahora debemos crear una base de datos que será utilizada por WordPress y además, debemos crear un usuario específico para esa base de datos con todos los permisos

¹⁰ Página oficial de descarga de WordPress - <https://wordpress.org/download/>

sobre ella. Esto se realiza desde el panel de administración de la base de datos (PhpMyAdmin) ejecutando el siguiente código SQL:

```
CREATE DATABASE nombreBSSD;  
  
CREATE USER 'nombre_usuario'@'localhost' IDENTIFIED BY 'contrasena';  
  
GRANT ALL PRIVILEGES ON nombreBSSD TO 'nombre_usuario'@'localhost';
```

4. Terminar instalación de WordPress

Una vez creada la base de datos con el usuario, deberemos introducir el dominio de nuestra web en el navegador y se iniciará el proceso final de instalación. Siguiendo todos los pasos llegaremos a una pantalla en la que se nos pedirá la siguiente información:

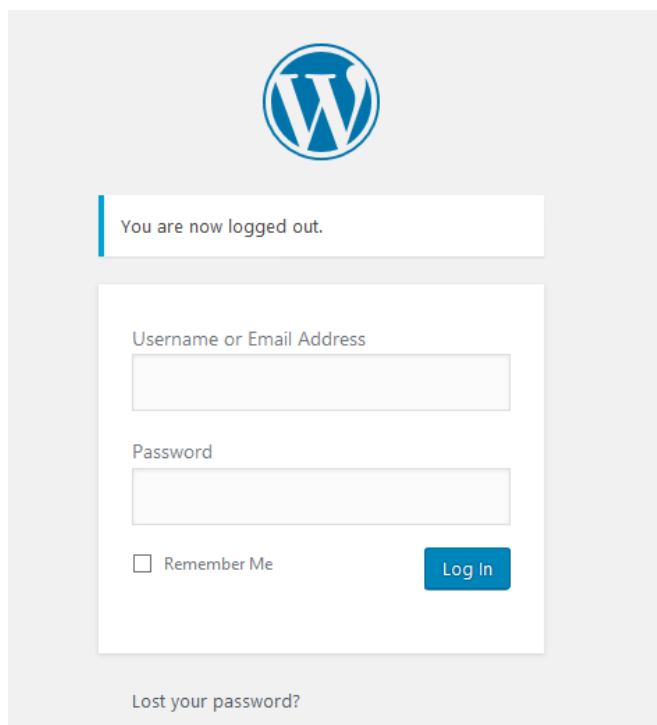
- Nombre de la base de datos
- Nombre del usuario de la base de datos
- Contraseña
- Dirección del host de la base de datos (por defecto localhost)
- Prefijo de la tabla (por defecto wp_)

Una vez completados todos los campos y enviado el formulario, se solicitará la información acerca de la web y la creación de un usuario de tipo administrador, pidiendo su nombre, contraseña y correo electrónico.

5. Gestión de la aplicación web

Llegado a este punto ya tendremos instalado y funcionando nuestro WordPress, ahora debemos elegir la plantilla que deseemos, instalarla y posteriormente instalar los plugins necesarios.

Para acceder al Escritorio de administración de WordPress, debemos escribir nuestro dominio en el navegador seguido de /wp-admin. Una vez cargada la página veremos un panel con el login que podemos observar en la siguiente ilustración.



Cuando hayamos iniciado la sesión, tendremos que dirigirnos al apartado “Apariencia” del menú de la izquierda y desde allí a la pestaña de “Temas”. Una vez cargada la página, iremos a “Añadir Nuevo” y finalmente a “Subir Tema”. Se cargará una ventana para seleccionar el tema que deseamos instalar y una vez instalado, procederemos a activarlo.

Ahora que tenemos el Tema instalado y activado, nos dirigiremos a la sección de “Plugins” y desde allí instalaremos todos aquellos que consideremos necesarios para la

Ilustración 41: Login panel de administración de WordPress

aplicación web. Uno de los fundamentales es “**WP Membership**”, éste será utilizado para el control de acceso de los usuarios a las páginas internas de la web, desde las que se podrá gestionar la base de datos.

Una vez instalado este plugin, deberemos crear el rol de Gestor desde la opción de “Niveles de Membresía”. Posteriormente, deberemos crear los usuarios autorizados que deseemos.

Si hemos seguido todos los pasos hasta aquí, estaremos listos para la instalación del plugin desarrollado y para la creación de las páginas internas de la web.

Instalación del Plugin de Gestión de SegoviaEnBus APP

Nos dirigimos a la sección de “Plugins” y seleccionamos la pestaña de “Añadir Nuevo”. Una vez cargada la página, haremos click sobre el botón situado en la parte superior “Subir Plugin”. Debemos buscar el archivo .zip que contiene el plugin en nuestro equipo e instalarlo.

Cuando esté instalado, procederemos a activarlo. Cuando termine de activarse, podremos observar que se crea una nueva opción en el menú de la izquierda llamada “Gestión Urbanos Segovia”. Si accedemos a esta opción, podremos ver todos los shortcodes con las funciones implementadas dentro del plugin.

Ahora que tenemos el plugin activado, deberemos crear las diferentes páginas. A continuación se detallarán los pasos para crear una de las páginas. El resto se crean del mismo modo.

Primero debemos añadir una nueva página, para ello, iremos a la sección de “Páginas” del menú, y seleccionaremos la pestaña de “Añadir Nueva”.

Una vez cargada la página de creación, introduciremos el título de la página y la estructura que deseemos. Cuando esté creada la estructura, crearemos un elemento de tipo texto, que se encargará de cargar la función del plugin deseada.

Para cargar la función deberemos introducir el shortcode deseado. En la siguiente ilustración podemos observar la forma de hacerlo.

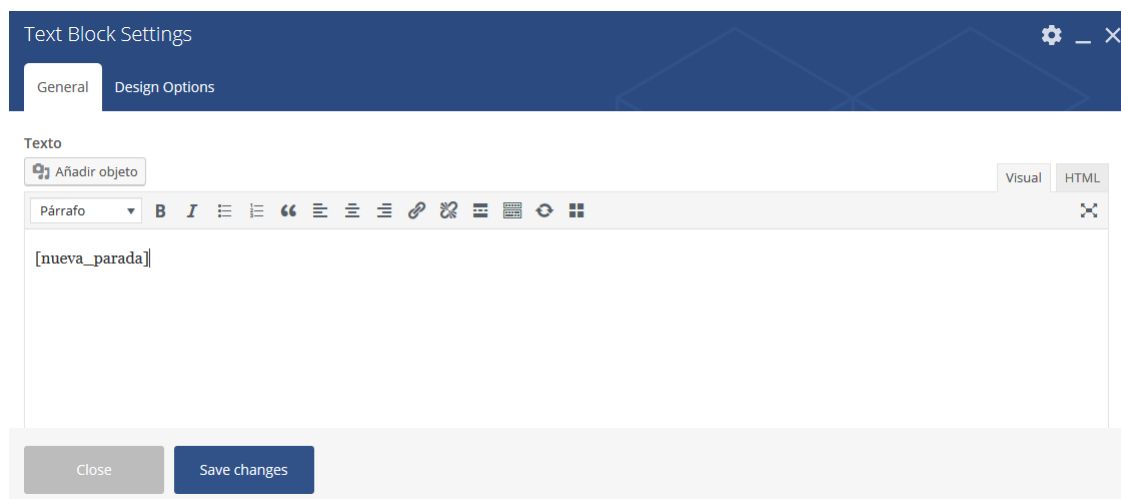


Ilustración 42: Introducción de shortcode en la web

Si activásemos la página en este momento, podríamos observar cómo se carga el formulario que podemos contemplar en la siguiente ilustración.

The image shows a form titled 'Añadir Nueva Parada'. It has several input fields: 'Nombre de parada:', 'Posición', 'Latitud', and 'Longitud'. There is also a file upload field for 'Imagen (tam. máximo 8Mb)' with an 'Examinar...' button and a message 'No se ha seleccionado ningún archivo.' in a red box. At the bottom, there is a checkbox for 'Parada de interés turístico' and an 'Añadir Parada' button.

Ilustración 43: Formulario Añadir Nueva Parada

Ahora debemos crear todas las páginas necesarias siguiendo estos pasos. Tras crearlas deberemos añadirlas al menú de la web y así poder acceder a ellas desde la página de

inicio. Para ello debemos ir a la opción de “Apariencia” y a la pestaña “Menús”. Una vez dentro crearemos un nuevo menú con la estructura que deseemos tener; podemos observar un ejemplo en la siguiente ilustración.

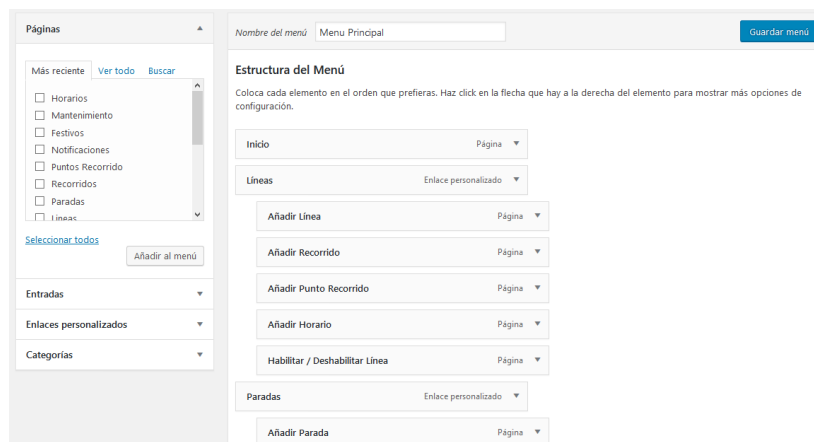


Ilustración 44: Ejemplo de estructura del menú

Tras crear el menú de la aplicación con todas las opciones necesarias para la gestión de la base de datos, se deben añadir a los usuarios de tipo gestor. Para ello debemos dirigirnos al panel de opciones del plugin elegido (WP Membership), el cual se encuentra en el menú de opciones del escritorio de WordPress. En primer lugar deberemos crear el tipo de usuario desde la pestaña Add level, como se muestra en la siguiente ilustración.

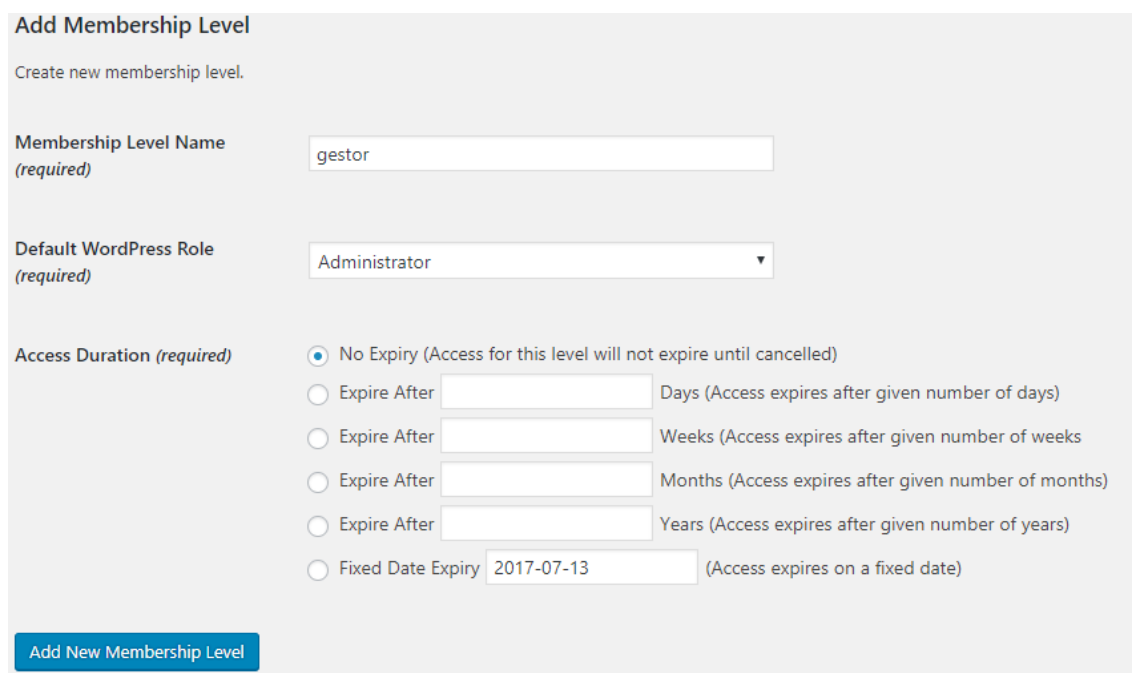


Ilustración 45: Añadir nivel de membresía en plugin de sesión

A continuación deberemos dirigirnos a las páginas que hayamos creado previamente para activar la opción de que solo los usuarios con el rol que acabamos de crear puedan acceder a ellas. En la siguiente ilustración podemos observar cómo quedaría.

Simple WP Membership Protection

Do you want to protect this content?

- No, Do not protect this content.
- Yes, Protect this content.

Select the membership level that can access this content:

- gestor

Ilustración 46: Protección de acceso a una página de la aplicación web

8.2. Manual de gestor

Este manual va dirigido a aquellos usuarios que trabajarán con la aplicación web, con el fin de gestionar los datos del servicio de autobuses urbanos de Segovia.

Se ha comprobado que la aplicación web funciona tanto para los navegadores Mozilla Firefox y Google Chrome, pero está optimizada para funcionar sobre la segunda opción. Por lo tanto se recomienda que su uso se lleve a cabo sobre este navegador.

Para acceder al servidor lo haremos desde la siguiente URL:

<http://157.88.125.242:62022/gestionurbanossegovia/>



Ilustración 47: Panel de inicio de sesión de la aplicación web

Una vez dentro, deberemos iniciar sesión con nuestra cuenta de gestor, desde el panel de login mostrado en la ilustración anterior. Posteriormente se deberá elegir del menú de la aplicación la acción que se desea realizar.

A continuación se detallarán las funciones de gestión disponibles y el modo de operar con ellas:

Activar o desactivar mantenimiento: esta operación se utilizará para cambiar el estado de la web, ya que para realizar cambios sobre la base de datos, el modo de mantenimiento debe estar activado. Además para que la aplicación móvil descargue los datos actualizados, se deberá estar desactivado.



Ilustración 48: Estado de mantenimiento de la aplicación web

Añadir línea: se solicitará al usuario que introduzca el nombre y número de línea, éste último no es necesario introducirlo, ya que habrá líneas como la Búho, que no tengan uno asociado. Para introducir una nueva línea es obligatorio que ni el nombre, ni el número estén registrados previamente.

Añadir parada: será necesario introducir el nombre de la parada, las coordenadas geográficas a través de los campos de latitud y longitud y una imagen representativa de

la parada, la cual no podrá exceder de los 8Mb de tamaño. Además se deberá seleccionar la casilla de parada de interés turístico en caso de que represente este aspecto.

Añadir Parada Home » Paradas » Añadir Parada

Nombre de parada:	Nombre de parada
Posición	
Latitud	12.222222222222
Longitud	12.222222222222
Imagen (tam. máximo 8Mb)	Seleccionar archivo IMG_20170613_104317.jpg
Parada de interés turístico	<input checked="" type="checkbox"/>

Añadir Parada

Ilustración 49: Ejemplo adición de parada desde la aplicación web

Añadir recorrido: para crear uno nuevo se solicitará el nombre de éste, la línea a la que pertenece y las paradas de inicio y final del trayecto. Estas paradas no podrán añadirse si ya existe un recorrido asignado a esta línea con el mismo principio y fin. Además se deberá seleccionar la casilla de recorrido especial para aquellos casos en los que sea un trayecto adicional de la línea que se realiza en ocasiones especiales.

Añadir punto a recorrido: se deberá seleccionar el recorrido de una lista y la parada que se desea introducir a éste de otro listado. Finalmente se introducirá el orden de la parada dentro del recorrido.

Añadir horario: en la primera pantalla se solicitará la selección del recorrido al que afecta el horario, el tipo de día (laboral, sábado o festivo) y el número de horas que se desean añadir. Tras enviar estos datos se cargará una nueva pantalla en la que se deberá seleccionar la parada a la que se pretende asignar los horarios, y también se tendrán que completar todos los campos de las horas a añadir.

Añadir Horario Home » Líneas » Horarios » Añadir Horario

Recorrido:	L1 / Colón / San José
Tipo de día:	Laboral
Número de trayectos:	14

Añadir Horario

Ilustración 50: Ejemplo de adición de horario desde la aplicación web

Publicar notificación: para crear una nueva notificación solo se deberá rellenar obligatoriamente el campo de título, ya que el resto de campos son voluntarios. Estos campos adicionales son la descripción y las fechas de inicio y final, las cuales deberán añadirse en caso de querer establecer un periodo de duración de la notificación.

Asignar notificación a línea: se mostrarán dos listados diferentes en los cuales se deben seleccionar la línea y la notificación que se pretende asignar.

Asignar notificación a parada: su funcionamiento es idéntico que para las líneas, solo que se debe seleccionar una parada en vez de una línea.

Añadir festivos: se solicitará la introducción de un día posterior a la fecha actual para registrarlo como día festivo.

Habilitar o deshabilitar línea: una vez creados los recorridos de una línea con sus respectivos horarios asignados, se deberá habilitar la línea. De este modo, se cargará en el listado de la aplicación móvil. En caso de querer modificar la línea, se deberá deshabilitar previamente.

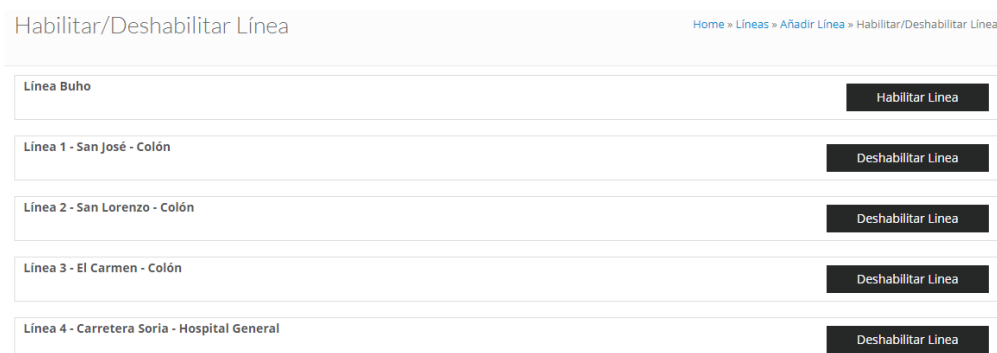


Ilustración 51: Panel para habilitar o deshabilitar líneas desde la aplicación web

8.3. Manual de usuario

El desarrollo de la aplicación móvil se ha realizado tratando de crear una navegación y una interfaz sencilla e intuitiva. Por esta razón cualquier usuario con un conocimiento mínimo para manejar un dispositivo móvil, podrá utilizar la app sin problemas.

En primer lugar se debe instalar la aplicación en el dispositivo. En esta versión de la aplicación (la desarrollada para el proyecto), el usuario se ve obligado a instalarla manualmente desde el archivo .apk. Para ello deberá obtener la aplicación solicitándosela al desarrollador.

Una vez en posesión de la aplicación, se debe instalar en un terminal móvil. Para realizar esta acción el usuario deberá acceder previamente a los ajustes del dispositivo y en la sección de seguridad, activar la opción de orígenes desconocidos. Esto hará que permita la instalación de una aplicación que no procede la Google Play.

Ahora ya se puede instalar la aplicación, pero antes se debe tener en cuenta que la versión mínima de Android para la que se ha desarrollado es la API 22, es decir, Android 5.1. Sin embargo, la API recomendada es la 23, Android 6.0 Marshmallow.

Para instalar la aplicación tan solo tendremos que ejecutar el archivo .apk descargado y la instalación se realizará automáticamente.

En cuanto a la navegación desde dentro de la aplicación, se navegará desde el menú principal. A continuación se muestran las opciones del menú junto con las acciones que se pueden realizar tras acceder a ellas:

- Líneas: Se muestra un listado de las líneas disponibles.



Ilustración 52: Captura líneas APP

- Recorridos: Se muestra un listado de los recorridos de la línea seleccionada. Además se mostrará un botón para cargar un mapa de la línea.



Ilustración 53: Captura recorridos APP

- Mapa de línea: Muestra un mapa de la línea previamente seleccionada. Se permitirá mostrar la ubicación del usuario en el mapa tras la pulsación del nuevo botón creado.

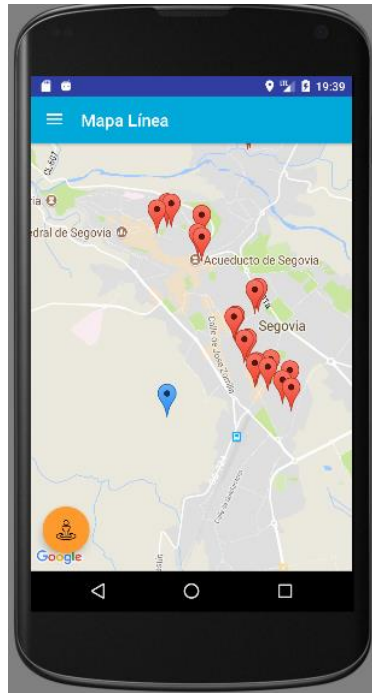


Ilustración 55: Captura mapa línea APP

- Paradas del recorrido: Se accede a ellas tras la selección de uno de los recorridos mostrados anteriormente. Si se selecciona una parada se muestra su información.

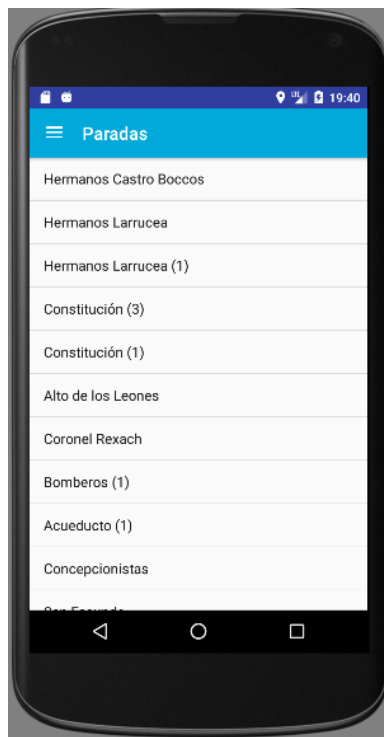


Ilustración 54: Captura paradas de recorrido APP

- Información de parada: Se muestra una imagen de la parada junto a la información de los próximos autobuses y se ofrecen tres nuevas opciones.

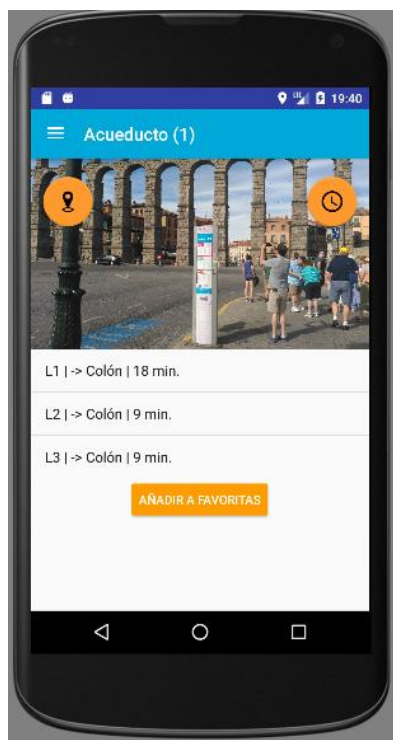


Ilustración 56: Captura información de parada

- Añadir/Eliminar parada favorita
- Mostrar mapa de parada
- Consultar horarios de la parada
- Buscar paradas: Se muestra un listado de las paradas junto a un buscador. Si se selecciona una de las paradas se podrá acceder a su información.

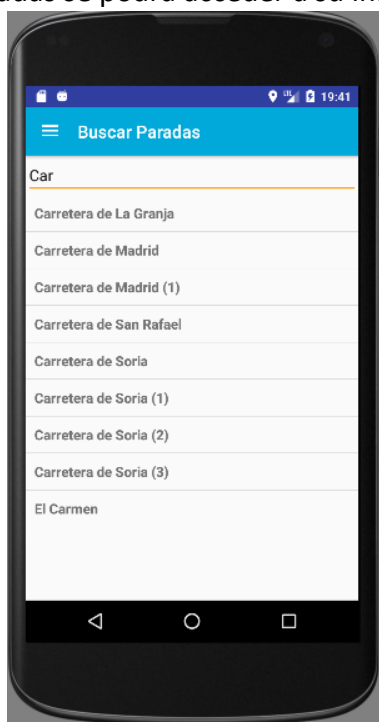


Ilustración 57: Captura buscar paradas APP

- Buscar paradas cercanas: Se muestra un mapa con todas las paradas registradas y permite geolocalizar al usuario.

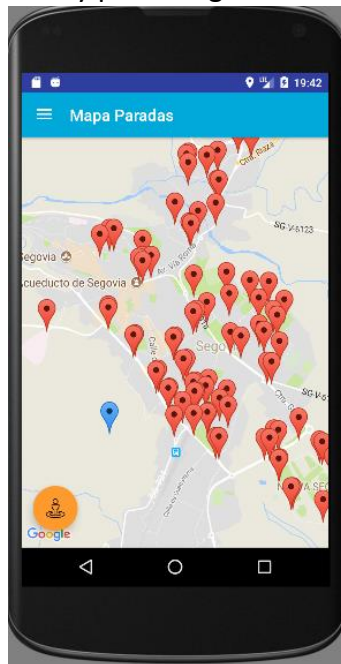


Ilustración 58: Captura paradas cercanas APP

- Paradas favoritas: Muestra un listado de las paradas que hayan sido previamente seleccionadas como favoritas.

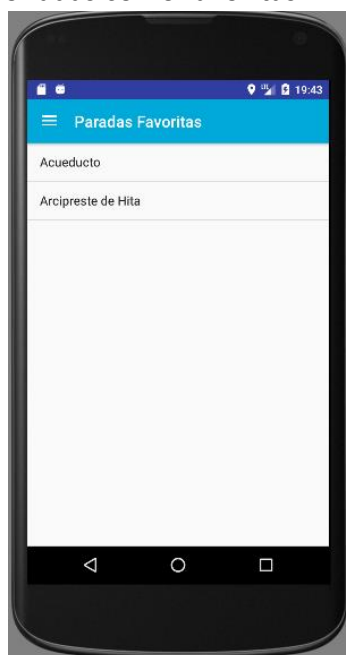


Ilustración 59: Captura paradas favoritas APP

- Paradas de información turística: Se carga un listado con todas las paradas almacenadas de carácter turístico.

- Bus Turístico: Se muestra la información acerca del autobús turístico.



Ilustración 60: Captura información bus turístico APP

- Tarifas: Se carga un listado con los tipos de billete y sus precios.

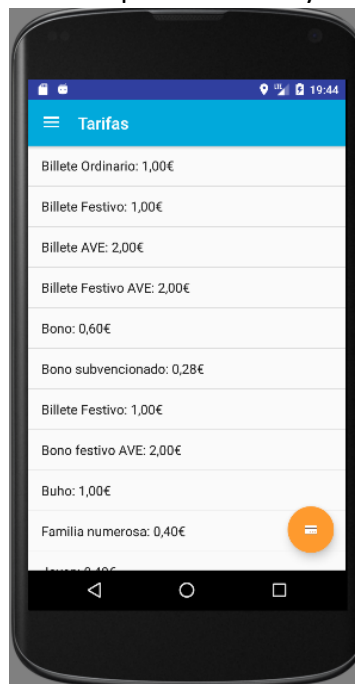


Ilustración 61: Captura tarifas APP

- Información de abonos: Se muestra información sobre los puntos de venta y recarga de abonos de transporte.

- Ajustes: Permite al usuario modificar el idioma de la aplicación. Éste por defecto será en castellano, con la excepción de aquellos dispositivos móviles con la configuración del terminal en inglés.

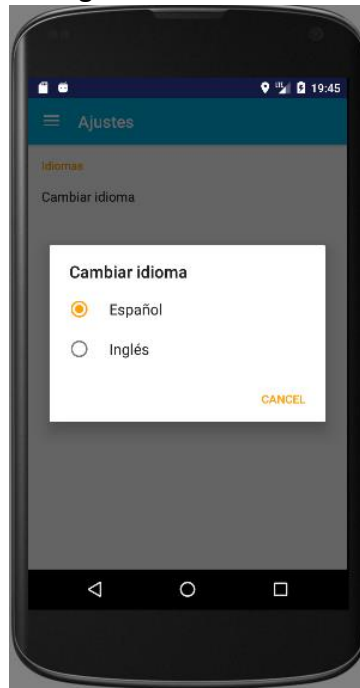


Ilustración 62: Captura cambiar idioma APP

- Salir: Permite salir de la aplicación.

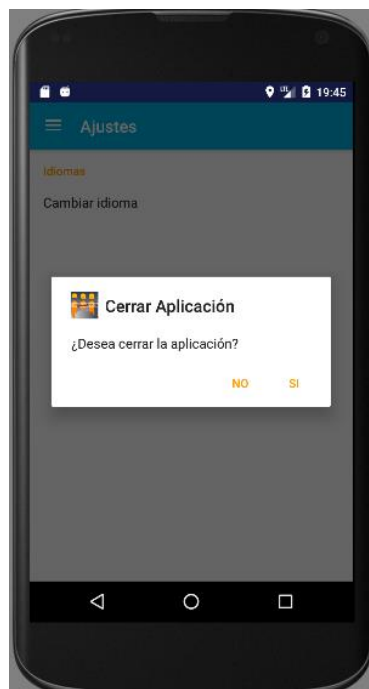


Ilustración 63: Captura salir APP

CAPÍTULO 9

CONCLUSIONES

En este capítulo se realiza una revisión del trabajo realizado, analizando los resultados y el aprendizaje llevado a cabo. Finalmente se explica lo que se pretende alcanzar en el futuro tras la finalización de este proyecto académico.

Objetivos del proyecto

El principal objetivo del proyecto era el desarrollo de una aplicación móvil, que facilitase el acceso de los usuarios de los autobuses urbanos de Segovia, toda la información acerca de recorridos y horarios.

El objetivo se considera cumplido ya que la aplicación cumple con todas las funciones que se habían previsto. Sin embargo la colaboración del Ayuntamiento de Segovia es completamente necesaria debido a que, como se menciona en el capítulo de introducción, es imprescindible la obtención de datos actualizados para ofrecer el mejor servicio posible a todos los usuarios.

Aprendizaje realizado

Este proyecto ha resultado de especial interés debido a la variedad de tecnologías empleadas. En primer lugar, para la aplicación web se utiliza el gestor de contenidos WordPress, el cual por lo general se suele aprovechar para la creación de páginas web estáticas. La novedad de este proyecto ha sido la investigación sobre cómo crear un plugin para WordPress, que permita convertir la página en una aplicación web de gestión de una base de datos.

Además se ha profundizado en las enseñanzas adquiridas durante la carrera, tanto para la creación de este documento, como para la gestión de la base de datos y la creación de una aplicación móvil para Android.

Trabajo futuro

Como se menciona en el primer punto de este capítulo, la colaboración con el Ayuntamiento de Segovia es completamente imprescindible para el mantenimiento de la web, por ello desde el parque científico del Campus de Segovia, se está volviendo a tratar de establecer contacto con el Ayuntamiento, incluyendo este proyecto a un plan de desarrollo de “*smart city*” para Segovia. Este proyecto encaja perfectamente en este campo debido al servicio tecnológico que facilita e incentiva el uso de los autobuses urbanos tanto a los habitantes de la ciudad, como a aquellos que vengan a visitarla.

BIBLIOGRAFÍA

En este apartado se recogen todas las referencias del contenido que se ha consultado para poder llevar a cabo el proyecto. Se ha decidido dividir las referencias por secciones según el contenido que aportan al proyecto con el fin de facilitar su búsqueda al lector.

Información Periodística

- «El móvil supera por primera vez al ordenador para acceder a Internet». <http://www.elmundo.es/sociedad/2016/04/04/57026219e2704e90048b465e.html> [Accedido el 14/07/2017]
- «Gartner Says Five of Top 10 Worldwide Mobile Phone Vendors Increased Sales in Second Quarter of 2016». <http://www.gartner.com/newsroom/id/3415117> [Accedido el 14/07/2017]
- «Segovia una de las capitales de la región donde el autobús urbano está más subvencionado». <http://www.eladelantado.com/noticia/local/226114/segovia-es-una-de-las-capitales-de-la-region-donde-el-autobus-urbano-esta-mas-subvencionado> [Accedido el 14/07/2017]
- «Surprising Statistics About WordPress Usage». <https://managewp.com/14-surprising-statistics-about-wordpress-usage> [Accedido el 14/07/2017]

Aplicaciones Analizadas

- «Auvasa Bus – Autobuses urbanos Valladolid». http://www.auvasa.es/auv_app.asp [Accedido el 14/07/2017]
- «Avanza Zaragoza – Autobuses urbanos de Zaragoza». <http://zaragoza.avanzagrupo.com/> [Accedido el 14/07/2017]
- «BMS – Bus metropolitano Salamanca». <https://www.bmsalamanca.com/es/inicio> [Accedido el 14/07/2017]
- «Bus Madrid – Autobuses urbanos de Madrid». <https://play.google.com/store/apps/details?id=es.android.busmadrid.apk&hl=es> [Accedido el 14/07/2017]
- «Bus Segovia – Autobuses urbanos de Segovia». <https://play.google.com/store/apps/details?id=urbano.segovia.immaroto&hl=es> [Accedido el 14/07/2017]
- «GranadaBus – Autobuses urbanos de Granada». <http://granadabus.es/> [Accedido el 14/07/2017]

WordPress

- «Download WordPress». <https://wordpress.org/download/> [Accedido el 14/07/2017]

- «Hood | Responsive Multi-Purpose Theme». <https://themeforest.net/item/hood-responsive-multipurpose-theme/15730392> [Accedido el 14/07/2017]
- «Cómo crear un plugin para Wordpres con página de opciones». <http://www.conasa.es/blog/como-crear-un-plugin-para-wordpress-con-pagina-de-opciones/> [Accedido el 14/07/2017]
- «Create a Settings Options Page for a Plugin». <https://wisdmlabs.com/blog/create-settings-options-page-for-wordpress-plugin/> [Accedido el 14/07/2017]
- «Plugin WordPress con tablas propias». <https://www.uno-de-piera.com/plugin-wordpress-con-tablas-propias/> [Accedido el 14/07/2017]
- «Cómo crear tus tablas de Base de Datos con WordPress». <https://www.danielcastanera.com/crear-tus-tablas-base-datos-wordpress/> [Accedido el 14/07/2017]
- «Running database queries in WordPress». <http://www.stevendobbelaere.be/running-database-queries-in-wordpress/> [Accedido el 14/07/2017]

Android

- «Android ListView con filtro de búsqueda» <http://www.mauricioalpizar.com/adroid-listview-con-filtro-de-busqueda/> [Accedido el 14/07/2017]
- «Android SQLite Database with Multiple Tables» <http://www.androidhive.info/2013/09/android-sqlite-database-with-multiple-tables/> [Accedido el 14/07/2017]
- «Android Fragment transaction: FragmentManager and Backstack» <https://www.survivingwithandroid.com/2013/04/android-fragment-transaction.html> [Accedido el 14/07/2017]
- «Creando una Google Maps Activity con Android Studio» <https://desarrollandoandroid.wordpress.com/2014/06/18/creando-una-google-maps-activity-con-android-studio/> [Accedido el 14/07/2017]
- «Google Maps Android Api v2: Guía De Mapas» <http://www.hermosaprogramacion.com/2016/05/google-maps-android-api-v2/> [Accedido el 14/07/2017]
- «Google Map Tutorial in Android Studio: How to get current location in Android Google Maps» <https://www.androidtutorialpoint.com/intermediate/android-map-app-showing-current-location-android/> [Accedido el 14/07/2017]
- «Volley: usar WebServices en Android de manera sencilla» <http://gpmess.com/blog/2014/05/28/volley-usando-webservices-en-android-de-manera-sencilla/> [Accedido el 14/07/2017]

Apéndice A – Desarrollo detallado de la especificación

Especificación de casos de uso

Aplicación Web

CU-05	Visualizar listado de paradas		
Versión	1.0	Fecha	11/05/2017
Requisitos asociados	CU-04, CU-06		
Actores	Gestor		
Descripción	Un Gestor podrá visualizar un listado de todas las paradas		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. Debe haber paradas registradas previamente 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita el listado de las paradas registradas 2. El sistema muestra un listado de todas las paradas ordenadas alfabéticamente 		
Postcondiciones	1. Si el Gestor solicita eliminar una de las paradas, se realizará el CU-06		
Excepciones	2.EX-07	Aún no se han registrado paradas en el sistema	
Frecuencia	Media tanto en el despliegue como en el mantenimiento		
Importancia	Alta		
Comentarios	Junto a las paradas también habrá un botón para eliminar la parada		

Tabla 60: CU-05 Visualizar listado de paradas

CU-06	Eliminar una parada		
Versión	1.0	Fecha	11/05/2017
Requisitos asociados	CU-02, CU-05		
Actores	Gestor		
Descripción	Un Gestor podrá eliminar una parada del sistema		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. El Gestor debe haber accedido al listado de las paradas previamente; CU-05 3. El modo de mantenimiento debe estar activado; CU-02 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita eliminar una parada el sistema 2. El sistema elimina la parada seleccionada 3. El sistema notifica el éxito de la operación 		
Postcondiciones			
Excepciones	2.EX-08	La parada pertenece al recorrido de una línea	
Frecuencia	Nula en la carga inicial de los datos Baja durante el mantenimiento de la web		
Importancia	Baja		
Comentarios			

Tabla 61: CU-06 Eliminar una parada

CU-07	Crear línea		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-02, CU-08, CU-10, CU-11, CU-12, CU-13, CU-26, RI-01		
Actores	Gestor		
Descripción	Un Gestor puede crear una nueva línea		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. El modo de mantenimiento debe estar activado; CU-02 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita añadir una nueva línea 2. El sistema solicita la información de la línea; RI-01 3. El Gestor envía la información de la línea 4. El sistema registra en el servidor la nueva línea 5. El sistema notifica el éxito de la operación 		
Postcondiciones	<ol style="list-style-type: none"> 1. El Gestor añade un recorrido a la línea a través del CU-12 		
Excepciones	4.EX-08	El número de línea introducido ya está en uso	
	4.EX-09	Los datos introducidos no son válidos	
Frecuencia	Alta durante la carga de datos inicial Baja durante el mantenimiento		
Importancia	Vital		
Comentarios			

Tabla 62: CU-07 Crear línea

CU-08	Visualizar listado de líneas		
Versión	1.0	Fecha	11/05/2017
Requisitos asociados	CU-07, CU-09		
Actores	Gestor		
Descripción	Un Gestor podrá visualizar un listado de las líneas registradas		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. Debe haber líneas registradas previamente 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita el listado de las líneas registradas 2. El sistema muestra un listado de todas las líneas ordenadas por su número 		
Postcondiciones	<ol style="list-style-type: none"> 1. Si el Gestor solicita modificar el nombre de una línea, se realizará el CU-09 		
Excepciones	2.EX-10	Aún no hay líneas registradas	
Frecuencia	Media tanto en el despliegue como en el mantenimiento		
Importancia	Alta		
Comentarios	Junto a las líneas también habrá un botón para actualizar su nombre		

Tabla 63: CU-08 Visualizar listado de líneas

CU-09	Modificar nombre de línea		
Versión	1.0	Fecha	11/05/2017
Requisitos asociados	CU-02, CU-08, RI-01		
Actores	Gestor		
Descripción	Un Gestor podrá modificar el nombre de una línea		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. El Gestor debe haber accedido previamente al listado de líneas; CU-08 3. El modo de mantenimiento debe estar activado; CU-02 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor envía el nuevo nombre de línea 2. El sistema modifica el nombre de la línea; RI-01 3. El sistema notifica el éxito de la operación 		
Postcondiciones			
Excepciones	2.EX-11	El nuevo nombre ya existe para otra línea	
	2.EX-12	El nuevo nombre es igual que el anterior	
	2.EX-13	Los datos introducidos no son válidos	
Frecuencia	Nula durante la carga inicial de datos Baja durante el mantenimiento		
Importancia	Baja		
Comentarios			

Tabla 64: CU-09 Modificar nombre de línea

CU-10	Habilitar línea		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-02, CU-07, CU-11, CU-12, CU-18		
Actores	Gestor		
Descripción	Un Gestor podrá habilitar una línea para que aparezca en la aplicación		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. Tiene que haber líneas creadas previamente; CU-07 3. La línea tiene que estar deshabilitada; CU-11 4. El modo de mantenimiento debe estar activado; CU-02 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita un listado de las líneas que se pueden habilitar 2. El sistema muestra las líneas deshabilitadas junto con un botón 3. El Gestor solicita habilitar una línea 4. El sistema habilita la línea seleccionada 		
Postcondiciones			
Excepciones	4.EX-14	Alguna de las paradas que forman los recorridos de la línea no tiene asignada un horario	
Frecuencia	Alta durante la carga inicial de datos Baja durante el mantenimiento		
Importancia	Vital		
Comentarios			

Tabla 65: CU-10 Habilitar línea

CU-11	Deshabilitar línea		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-02, CU-07, CU-10, CU-20		
Actores	Gestor		
Descripción	Un Gestor podrá deshabilitar una línea concreta		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. La línea debe estar habilitada; CU-10 3. El modo de mantenimiento debe estar activado; CU-02 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita un listado de las líneas que se pueden deshabilitar 2. El sistema muestra las líneas habilitadas junto con un botón 3. El Gestor solicita deshabilitar una línea 4. El sistema deshabilita la línea seleccionada 		
Postcondiciones			
Excepciones			
Frecuencia	Nula durante la carga inicial de los datos (deshabilitadas por defecto) Baja durante el mantenimiento		
Importancia	Baja		
Comentarios			

Tabla 66: CU-11 Deshabilitar línea

CU-12	Añadir un recorrido a una línea		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-02, CU-04, CU-07, CU-10, CU-13, CU-14, CU-15, CU-16, CU-17, CU-18, RI-03		
Actores	Gestor		
Descripción	Un gestor podrá añadir un recorrido a una línea		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. Se debe crear al menos una línea previamente para asignarle el recorrido creado; CU-07 3. El modo de mantenimiento debe estar activado; CU-02 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita la creación de un nuevo recorrido 2. El sistema solicita los datos del nuevo recorrido; RI-03 3. El Gestor envía los datos del nuevo recorrido 4. El sistema registra el nuevo recorrido 5. El sistema notifica el éxito de la operación 		
Postcondiciones	<ol style="list-style-type: none"> 1. El Gestor añade nuevas paradas al recorrido; CU-15 2. El Gestor asigna horarios a las paradas de origen y destino; CU-18 		
Excepciones	4.EX-15	El nombre de recorrido ya está registrado	
	4.EX-16	Ya existe un recorrido con el mismo origen y destino	
	4.EX-17	Los datos introducidos no son válidos	
Frecuencia	Alta durante la carga inicial de datos Baja durante el mantenimiento		
Importancia	Vital		
Comentarios			

Tabla 67: CU-12 Añadir un recorrido a una línea

CU-13	Visualizar Recorridos		
Versión	1.0	Fecha	11/05/2017
Requisitos asociados	CU-07, CU-12, CU-14		
Actores	Gestor		
Descripción	El Gestor podrá visualizar los recorridos pertenecientes a cada línea		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. Debe haberse registrado al menos un recorrido previamente; CU-12 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita la información de los recorridos 2. El sistema carga un listado de los recorridos agrupados por líneas 		
Postcondiciones	<ol style="list-style-type: none"> 1. El Gestor elimina un recorrido de la base de datos; CU-14 		
Excepciones	2.EX-18	Aún no se han registrado recorridos	
Frecuencia	Media durante la carga inicial de datos y el mantenimiento		
Importancia	Alta		
Comentarios			

Tabla 68: CU-13 Visualizar Recorridos

CU-14	Eliminar Recorrido		
Versión	1.0	Fecha	11/05/2017
Requisitos asociados	CU-02, CU-12, CU-13		
Actores	Gestor		
Descripción	Un Gestor podrá eliminar un recorrido		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. Se debe acceder al listado de recorridos previamente; CU-13 3. El modo de mantenimiento debe estar activado; CU-02 		
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede al listado de recorridos 2. El usuario solicita eliminar un recorrido 3. El sistema elimina el recorrido 4. El sistema notifica el éxito de la operación 		
Postcondiciones			
Excepciones			
Frecuencia	Nula durante la carga inicial de datos Baja durante el mantenimiento		
Importancia	Baja		
Comentarios			

Tabla 69: CU-14 Eliminar Recorrido

CU-15	Añadir punto a recorrido		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-02, CU-04, CU-12, CU-16, CU-17, CU-18, CU-19, CU-20, RI-04		
Actores	Gestor		
Descripción	Un Gestor podrá añadir una parada a un recorrido		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. El recorrido debe existir; CU-12 3. El modo de mantenimiento debe estar activado; CU-02 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita al sistema la creación de un nuevo punto dentro de un recorrido 2. El sistema solicita los datos del nuevo punto de recorrido; RI-04 3. El Gestor envía los datos del punto de recorrido 4. El sistema añade el punto al recorrido 5. El sistema notifica el éxito de la operación 		
Postcondiciones	<ol style="list-style-type: none"> 1. El Gestor asigna un horario al nuevo punto creado; CU-18 		
Excepciones	4.EX-19	El punto ya se encuentra en el recorrido	
	4.EX-20	Los datos introducidos no son válidos	
Frecuencia	Alta durante la carga de datos inicial Baja durante el mantenimiento		
Importancia	Vital		
Comentarios			

Tabla 70: CU-15 Añadir punto a recorrido

CU-16	Visualizar puntos de recorrido		
Versión	1.0	Fecha	11/05/2017
Requisitos asociados	CU-12, CU-15, CU-17		
Actores	Gestor		
Descripción	Un Gestor podrá visualizar los puntos de cada recorrido		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. Debe existir al menos un recorrido; CU-12 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita la visualización de los puntos por cada recorrido 2. El sistema devuelve un listado con todos los puntos separados por recorridos y ordenados 		
Postcondiciones	<ol style="list-style-type: none"> 1. El Gestor elimina un punto de recorrido; CU-17 		
Excepciones			
Frecuencia	Media durante la carga inicial de datos y durante el mantenimiento		
Importancia	Alta		
Comentarios			

Tabla 71: CU-16 Visualizar puntos de recorrido

CU-17	Eliminar punto de recorrido		
Versión	1.0	Fecha	11/05/2017
Requisitos asociados	CU-02, CU-12, CU-15, CU-16, CU-18		
Actores	Gestor		
Descripción	Un Gestor podrá eliminar un punto de recorrido		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. Se deben visualizar los puntos de recorrido para elegir uno de ellos 3. El modo de mantenimiento debe estar activado; CU-02 		
Secuencia normal	<ol style="list-style-type: none"> 1. El gestor solicita eliminar un punto de recorrido 2. El sistema elimina el horario del punto de recorrido 3. El sistema elimina el punto de recorrido seleccionado 4. El sistema notifica el éxito de la operación 		
Postcondiciones			
Excepciones	2.EX-21	El punto seleccionado es el origen o destino del recorrido	
Rendimiento			
Frecuencia	Nula durante la carga inicial de los datos Baja durante el mantenimiento		
Importancia	Alta		
Comentarios			

Tabla 72: CU-17 Eliminar punto de recorrido

CU-18	Asignar horario		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-02, CU-10, CU-12, CU-15, CU-17, CU-19, CU-20		
Actores	Gestor		
Descripción	Un Gestor podrá asignar un horario a un punto de recorrido		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. Debe existir al menos un recorrido 3. El modo de mantenimiento debe estar activado; CU-02 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita la asignación de un horario 2. El sistema solicita el horario; RI-05 3. El Gestor envía el horario 4. El sistema registra el horario 5. El sistema notifica el éxito de la operación 		
Postcondiciones			
Excepciones	4.EX-22	Al menos una de las horas introducidas ya estaba registrada	
	4.EX-23	Los datos introducidos no son válidos	
Frecuencia	Alta durante la carga inicial de los datos Media durante el mantenimiento		
Importancia	Vital		
Comentarios			

Tabla 73: CU-18 Asignar horario

CU-19	Visualizar Horarios		
Versión	1.0	Fecha	11/05/2017
Requisitos asociados	CU-15, CU-18, CU-20		
Actores	Gestor		
Descripción	Un Gestor podrá visualizar los horarios de un punto de recorrido		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. Debe existir al menos un punto de recorrido; CU-15 3. Debe existir al menos un horario para un punto de recorrido; CU-18 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita la visualización de los horarios 2. El sistema muestra los horarios 		
Postcondiciones	<ol style="list-style-type: none"> 1. El Gestor elimina un horario; CU-19 		
Excepciones			
Frecuencia	Media durante la carga de datos inicial y durante el mantenimiento		
Importancia	Alta		
Comentarios			

Tabla 74: CU-19 Visualizar horarios

CU-20	Eliminar horario		
Versión	1.0	Fecha	11/05/2017
Requisitos asociados	CU-02, CU-11, CU-15, CU-18, CU-19		
Actores	Gestor		
Descripción	Un Gestor podrá eliminar el horario de un punto de recorrido		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. Se deben visualizar previamente los horarios de punto 3. Se debe deshabilitar la línea a la que afecta previamente 4. El modo de mantenimiento debe estar activado; CU-02 		
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario solicita eliminar una hora 2. El sistema elimina la hora seleccionada 3. El sistema notifica el éxito de la operación 		
Postcondiciones			
Excepciones	2.EX-24	La línea a la que afecta no ha sido deshabilitada	
Frecuencia	Nula durante la carga de datos inicial Media durante el mantenimiento		
Importancia	Alta		
Comentarios			

Tabla 75: CU-20 Eliminar horario

CU-21	Publicar notificación		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-02, CU-22, CU-23, CU-24, CU-25, CU-26, RI-06		
Actores	Gestor		
Descripción	Un Gestor podrá publicar una notificación		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. El modo de mantenimiento debe estar activado; CU-02 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita la creación de una nueva notificación 2. El sistema solicita los datos de la notificación; RI-06 3. El Gestor envía los datos de la notificación 4. El sistema registra la notificación 5. El sistema notifica el éxito de la operación 		
Postcondiciones	<ol style="list-style-type: none"> 1. Asignar notificación a línea; CU-26 2. Asignar notificación a parada; CU-25 		
Excepciones	4.EX-25	La fecha de inicio es anterior a la fecha actual	
	4.EX-26	La fecha de fin no puede ser anterior a la inicial	
	4.EX-27	Los datos introducidos no son válidos	
Frecuencia	Baja durante la carga inicial de los datos Alta durante el mantenimiento		
Importancia	Vital		
Comentarios			

Tabla 76: CU-21 Publicar notificación

CU-22	Visualizar notificaciones		
Versión	1.0	Fecha	11/05/2017
Requisitos asociados	CU-21, CU-23, CU-24		
Actores	Gestor		
Descripción	Un Gestor podrá visualizar las notificaciones registradas		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. Debe existir al menos una notificación publicada; CU-21 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita la visualización de notificaciones 2. El sistema muestra las notificaciones 		
Postcondiciones	<ol style="list-style-type: none"> 1. Actualizar la descripción de la notificación; CU-24 2. Eliminar la notificación; CU-23 		
Excepciones			
Frecuencia			
Importancia			
Comentarios			

Tabla 77: CU-22 Visualizar notificaciones

CU-23	Eliminar notificación		
Versión	1.0	Fecha	11/05/2017
Requisitos asociados	CU-02, CU-21, CU-22, CU-25, CU-26		
Actores	Gestor		
Descripción	El Gestor podrá eliminar una notificación		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. Se debe visualizar el listado de notificaciones; CU-22 3. El modo de mantenimiento debe estar activado; CU-02 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita eliminar una notificación 2. El sistema elimina las relaciones con las líneas de la notificación seleccionada 3. El sistema elimina las relaciones con las paradas de la notificación seleccionada 4. El sistema elimina la notificación seleccionada 5. El sistema notifica el éxito de la operación 		
Postcondiciones			
Excepciones			
Frecuencia	Nula durante la carga inicial de datos Media durante el mantenimiento		
Importancia	Media		
Comentarios			

Tabla 78: CU-23 Eliminar notificación

CU-24	Modificar descripción de notificación		
Versión	1.0	Fecha	11/05/2017
Requisitos asociados	CU-02, CU-21, CU-22, RI-06		
Actores	Gestor		
Descripción	El Gestor podrá modificar la descripción de una notificación		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. Se debe visualizar el listado de notificaciones; CU-22 3. El modo de mantenimiento debe estar activado; CU-02 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor envía la nueva descripción; 2. El sistema modifica la descripción de la notificación seleccionada; RI-06 3. El sistema notifica el éxito de la operación 		
Postcondiciones			
Excepciones	2.EX-28	La descripción es igual a la anterior	
	2.EX-29	Los datos introducidos no son válidos	
Frecuencia	Nula durante la carga inicial de datos Baja durante el mantenimiento		
Importancia	Baja		
Comentarios			

Tabla 79: CU-24 Modificar descripción de notificación

CU-25	Asignar notificación a parada		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-02, CU-04, CU-21, CU-23, RI-07		
Actores	Gestor		
Descripción	Un Gestor podrá asignar una notificación a una parada		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. Debe existir al menos una parada; CU-04 3. Debe existir al menos una notificación; CU-21 4. El modo de mantenimiento debe estar activado; CU-02 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita la asignación de una notificación a una parada 2. El sistema solicita los datos para la asignación; RI-07 3. El Gestor envía los datos de la asignación 4. El sistema registra la relación 5. El sistema notifica el éxito de la operación 		
Postcondiciones			
Excepciones	4.EX-30	La parada ya estaba asignada a la notificación	
	4.EX-31	Los datos introducidos no son válidos	
Frecuencia	Baja durante la carga de datos inicial Media durante el mantenimiento		
Importancia	Alta		
Comentarios			

Tabla 80: CU-25 Asignar notificación a parada

CU-26	Asignar notificación a línea		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-02, CU-07, CU-21, CU-23, RI-08		
Actores	Gestor		
Descripción	Un Gestor podrá asignar una notificación a una línea		
Precondiciones	<ol style="list-style-type: none"> 1. El Gestor debe estar identificado; CU-27 2. Debe existir al menos una parada; CU-07 3. Debe existir al menos una notificación; CU-21 		
Secuencia normal	<ol style="list-style-type: none"> 1. El Gestor solicita la asignación de una notificación a una línea 2. El sistema solicita los datos de la asignación; RI-08 3. El Gestor envía los datos de la asignación 4. El sistema registra la relación 5. El sistema notifica el éxito de la operación 		
Postcondiciones			
Excepciones	4.EX-32	La línea ya estaba asignada a la notificación	
	4.EX-33	Los datos introducidos no son válidos	
Frecuencia	Baja durante la carga de datos inicial Media durante el mantenimiento		
Importancia	Alta		
Comentarios			

Tabla 81: CU-26 Asignar notificación a línea

CU-27	Autenticar Gestor		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados			
Actores	Gestor		
Descripción	Un gestor podrá autenticarse		
Precondiciones			
Secuencia normal	<ol style="list-style-type: none"> 1. El actor solicita identificarse en el sistema 2. El sistema autentifica al Gestor 		
Postcondiciones			
Excepciones	2.EX-34	El usuario no está registrado en el sistema	
	2.EX-35	Los datos introducidos no son válidos	
Frecuencia	Alta. Mínimo una vez por uso de la aplicación		
Importancia	Vital		
Comentarios			

Tabla 82: CU-27 Autenticar gestor

Aplicación Móvil

CU-32	Buscar paradas por nombre		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-35, CU-43, CU-44		
Actores	Usuario		
Descripción	Un usuario podrá buscar una parada por su nombre		
Precondiciones			
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario solicita Buscar Paradas 2. El sistema muestra un listado completo de las paradas ordenadas por nombre junto con un buscador 3. El usuario envía el nombre de la parada 4. El sistema muestra el resultado de la búsqueda 		
Postcondiciones	<ol style="list-style-type: none"> 1. El usuario selecciona la parada deseada; CU-35 2. El usuario añade la parada a paradas favoritas; CU-43 3. El usuario elimina la parada de paradas favoritas; CU-44 		
Excepciones	4.EX36	La parada buscada no existe	
Frecuencia	Alta		
Importancia	Vital		
Comentarios			

Tabla 83: CU-32 Buscar paradas por nombre

CU-33	Visualizar paradas cercanas en mapa		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-45		
Actores	Usuario		
Descripción			
Precondiciones			
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario solicita visualizar las Paradas Cercanas 2. El sistema muestra un mapa con las paradas registradas 		
Postcondiciones	<ol style="list-style-type: none"> 1. El usuario se geolocaliza en el mapa; CU-45 		
Excepciones			
Frecuencia	Media. Principalmente para usuario que sean turistas		
Importancia	Vital		
Comentarios			

Tabla 84: CU-33 Visualizar paradas cercanas en mapa

CU-34	Visualizar listado de paradas de interés turístico		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-35		
Actores	Usuario		
Descripción	Un usuario podrá buscar las paradas de interés turístico		
Precondiciones			
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario solicita visualizar las Paradas de Interés Turístico 2. El sistema muestra un listado de las paradas de interés turístico 		
Postcondiciones	<ol style="list-style-type: none"> 1. Consulta de información de parada 		
Excepciones	2.EX-37	Aún no se han registrado paradas de interés turístico	
Frecuencia	Baja para los habitantes de Segovia Alta para los usuarios que sean turistas		
Importancia	Alta		
Comentarios			

Tabla 85: CU-34 Visualizar listado de paradas de interés turístico

CU-35	Consultar información de parada		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-29, CU-32, CU-34, CU-36, CU-37, CU-43, CU-44, CU-46		
Actores	Usuario		
Descripción	Un usuario podrá consultar la información de la parada (fotografía y próximos autobuses)		
Precondiciones	<ol style="list-style-type: none"> 1. Selección de parada desde el listado de recorrido; CU-29 2. Selección de parada desde la búsqueda por nombre; CU-32 3. Selección de parada desde las paradas de interés turístico; CU-34 4. Selección de parada desde el listado de paradas favoritas; CU-37 		
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario selecciona la parada que desea consultar 2. El sistema muestra toda la información de la parada 		
Postcondiciones	<ol style="list-style-type: none"> 1. El usuario consulta los horarios de la parada; CU-36 2. El usuario añade la parada a paradas favoritas; CU-43 3. El usuario elimina la parada de paradas favoritas; CU-44 4. El usuario visualiza la parada en un mapa; CU-46 		
Excepciones			
Frecuencia	Alta. Función principal de la aplicación		
Importancia	Vital		
Comentarios			

Tabla 86: CU-35 Consultar información de parada

CU-36	Consultar horarios de parada		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-35		
Actores	Usuario		
Descripción			
Precondiciones	1. Selección de opción desde información de parada; CU-35		
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario solicita información acerca de los horarios de una parada 2. El sistema muestra los recorridos a los que pertenece la parada 3. El usuario selecciona el recorrido que desea consultar 4. El sistema muestra los horarios divididos por tipos de día 5. El usuario selecciona el tipo de día que desea consultar 6. El sistema muestra los horarios para el tipo de día seleccionado 		
Postcondiciones			
Excepciones			
Frecuencia	Alta		
Importancia	Vital		
Comentarios			

Tabla 87: CU-36 Consultar horarios de parada

CU-37	Visualizar paradas favoritas		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-44		
Actores	Usuario		
Descripción	Un usuario podrá visualizar un listado de sus paradas favoritas		
Precondiciones			
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario solicita visualizar las Paradas Favoritas 2. El sistema muestra un listado con las paradas favoritas del usuario 		
Postcondiciones	1. El usuario elimina una parada favorita; CU-44		
Excepciones	2.EX38	Aún no se han añadido paradas favoritas	
Frecuencia	Alta para usuarios de Segovia Baja para usuario que sean turistas		
Importancia	Vital		
Comentarios			

Tabla 88: CU-37 Visualizar paradas favoritas

CU-38	Modificar idioma de la aplicación		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados			
Actores	Usuario		
Descripción	Un usuario podrá modificar el idioma de la aplicación		
Precondiciones			
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario solicita acceder a los ajustes 2. El sistema muestra las opciones de ajustes 3. El usuario solicita modificar el idioma 4. El sistema muestra los idiomas disponibles 5. El usuario selecciona el idioma que desea 6. El sistema cambia el idioma de la aplicación 		
Postcondiciones			
Excepciones	6.EX39	El idioma seleccionado ya estaba activado	
Frecuencia	Baja. Aproximadamente una sola vez durante toda la vida de la aplicación		
Importancia	Vital		
Comentarios			

Tabla 89: CU-38 Modificar idioma de la aplicación

CU-39	Visualizar Notificaciones		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-42		
Actores	Usuario		
Descripción	Un usuario podrá visualizar un listado de las notificaciones de la aplicación, ordenadas por fecha de actualización		
Precondiciones			
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario solicita visualizar las notificaciones 2. El sistema muestra un listado de las notificaciones 		
Postcondiciones	1. El usuario consulta la información de la notificación; CU-42		
Excepciones			
Frecuencia	Alta. Aproximadamente una vez por cada uso de la aplicación		
Importancia	Vital		
Comentarios			

Tabla 90: CU-39 Visualizar notificaciones

CU-40	Visualizar información tarifas		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-41		
Actores	Usuario		
Descripción	Un Usuario podrá visualizar la información de las tarifas vigentes		
Precondiciones			
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario solicita información de las tarifas 2. El sistema muestra la información sobre las tarifas 		
Postcondiciones	1. Visualización de información de abonos; CU-41		
Excepciones			
Frecuencia	Baja. Aproximadamente una vez durante todo el ciclo de vida de la aplicación		
Importancia	Media		
Comentarios			

Tabla 91: CU-40 Visualizar información tarifas

CU-41	Visualizar información abonos		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-40		
Actores	Usuario		
Descripción	Un Usuario podrá consultar la información acerca de los abonos		
Precondiciones	1. Visualización de tarifas; CU-40		
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario solicita información acerca de los abonos 2. El sistema muestra la información acerca de los abonos disponibles y el sistema de recarga de los mismos 		
Postcondiciones			
Excepciones			
Frecuencia	Baja. Aproximadamente una vez durante todo el ciclo de vida de la aplicación		
Importancia	Media		
Comentarios			

Tabla 92: CU-41 Visualizar información abonos

CU-42	Visualizar información notificación		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-39		
Actores	Usuario		
Descripción	Un usuario podrá visualizar la información de una notificación específica		
Precondiciones	1. Visualización del listado de notificaciones; CU-39		
Secuencia normal	1. Un usuario selecciona la notificación que desea consultar del listado 2. El sistema muestra la información específica de la notificación seleccionada		
Postcondiciones			
Excepciones			
Frecuencia	Media		
Importancia	Alta		
Comentarios			

Tabla 93: CU-42 Visualizar información notificación

CU-43	Añadir parada favorita		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-35		
Actores	Usuario		
Descripción	Un Usuario podrá añadir paradas favoritas		
Precondiciones	1. Visualizar Información de Parada; CU-35		
Secuencia normal	1. El usuario solicita añadir la parada a favoritas 2. El sistema registrará la parada como parada favorita		
Postcondiciones			
Excepciones			
Frecuencia	Media		
Importancia	Vital		
Comentarios			

Tabla 94: CU-43 Añadir parada favorita

CU-44	Eliminar parada favorita		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-35		
Actores	Usuario		
Descripción	Un Usuario podrá eliminar paradas favoritas		
Precondiciones	1. Visualizar información de parada; CU-35		
Secuencia normal	1. El usuario solicita eliminar la parada del listado de favoritas 2. El sistema eliminará la parada del listado de paradas favoritas		
Postcondiciones			
Excepciones			
Frecuencia	Baja		
Importancia	Vital		
Comentarios			

Tabla 95: CU-44 Eliminar parada favorita

CU-45	Geolocalizar usuario en mapa		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-30, CU-33, CU-46		
Actores	Usuario		
Descripción	Un usuario puede ver su posición en un mapa		
Precondiciones	<ol style="list-style-type: none"> 1. Visualización de las paradas de una línea en un mapa; CU-30 2. Visualización de las paradas cercanas en un mapa; CU-33 3. Visualización de una parada en un mapa; CU-35 		
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario solicita geolocalizarse en un mapa 2. El sistema calculará la posición del usuario y la situará en el mapa 		
Postcondiciones			
Excepciones			
Frecuencia	Baja para los usuarios de Segovia Alta para los usuarios que son turistas		
Importancia	Vital		
Comentarios			

Tabla 96: CU-45 Geolocalizar usuario en mapa

CU-46	Visualizar parada en mapa		
Versión	1.0	Fecha	30/03/2017
Requisitos asociados	CU-35, CU-45		
Actores	Usuario		
Descripción	Un usuario podrá visualizar una parada en el mapa		
Precondiciones	<ol style="list-style-type: none"> 1. Ver información de parada; CU-35 		
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario solicita visualizar una parada en un mapa 2. El sistema muestra un mapa con la geolocalización de la parada 		
Postcondiciones	<ol style="list-style-type: none"> 1. Geolocalización del usuario en el mapa; CU-45 		
Excepciones			
Frecuencia	Baja para los usuarios de Segovia Alta para los usuarios que sean turistas		
Importancia	Vital		
Comentarios			

Tabla 97: CU-46 Visualizar parada en mapa

Requisitos Funcionales

De los diferentes casos de uso obtenemos todas las funcionalidades que el sistema debe realizar. Éstas vienen recogidas en los siguientes requisitos funcionales.

Como el proyecto está formado por una aplicación web y una aplicación móvil, contaremos con dos sistemas diferentes. Por esta razón se han dividido los requisitos en función de si afectan a la aplicación web o a la aplicación móvil.

Aplicación Web

Para una mayor claridad a la hora de entender los requisitos, se han repartido en tres grupos:

Bases de datos

- RF-01: El sistema debe poder crear las tablas necesarias al activar el plugin.
- RF-02: El sistema debe poder eliminar las tablas utilizadas al desactivar el plugin.
- RF-03: El sistema debe poder eliminar las imágenes almacenadas al desactivar el plugin.
- RF-04: El sistema debe poder establecer una conexión con la base de datos.
- RF-05: El sistema debe poder modificar la base de datos.

Gestión de la Web

- RF-06: El sistema debe poder mostrar los shortcodes utilizados para la creación de las páginas con los formularios.
- RF-07: El sistema debe poder cargar un formulario para la inserción de nuevos Gestores.
- RF-08: El sistema debe poder registrar nuevos usuarios de tipo Gestor.
- RF-09: El sistema debe poder bloquear páginas para que solo puedan acceder a ellas los Gestores.
- RF-10: El sistema debe poder confirmar el registro de un nuevo Gestor.
- RF-11: El sistema debe poder notificar que un Gestor ya estaba registrado al tratar de hacerlo de nuevo.
- RF-12: El sistema debe poder detectar la sesión de los usuarios logueados.
- RF-: El sistema debe poder bloquear el acceso a aquellos usuarios que no inicien sesión.
- RF-13: El sistema debe poder actualizar las fechas de modificación de las tablas al realizar operaciones sobre ellas.

Operaciones

- RF-14: El sistema debe poder validar los formularios en el lado cliente.
- RF-15: El sistema debe poder activar el modo de mantenimiento.
- RF-16: El sistema debe poder notificar el éxito de las operaciones.
- RF-17: El sistema debe poder detectar el estado del modo de mantenimiento.
- RF-18: El sistema debe poder desactivar el modo de mantenimiento.
- RF-19: El sistema debe poder detectar si existen líneas habilitadas.

- RF-20: El sistema debe poder cargar un formulario específico para una nueva parada
- RF-21: El sistema debe poder registrar una nueva parada.
- RF-22: El sistema debe poder bloquear la inyección de código.
- RF-23: El sistema debe poder validar el formato de los formularios en el lado servidor.
- RF-24: El sistema debe poder detectar los tipos de archivos enviados a través de los formularios.
- RF-25: El sistema debe poder renombrar los archivos enviados a través de los formularios.
- RF-26: El sistema debe poder detectar el tamaño de los archivos enviados a través de los formularios.
- RF-27: El sistema debe poder notificar si un nombre de parada ya está registrado.
- RF-28: El sistema debe poder mostrar un listado de las paradas registradas.
- RF-29: El sistema debe poder notificar la no existencia de paradas registradas.
- RF-30: El sistema debe poder eliminar una parada registrada.
- RF-31: El sistema debe poder notificar si una parada pertenece a un recorrido.
- RF-32: El sistema debe poder cargar un formulario para la creación de una nueva línea.
- RF-33: El sistema debe poder registrar una nueva línea.
- RF-34: El sistema debe poder notificar si un número de línea ya está registrado.
- RF-35: El sistema debe poder notificar si un nombre de línea ya está registrado.
- RF-36: El sistema debe poder mostrar un listado de las paradas registradas.
- RF-37: El sistema debe poder notificar la no existencia de líneas registradas.
- RF-38: El sistema debe poder modificar el nombre de una línea.
- RF-39: El sistema debe poder notificar que el nuevo nombre de una línea es igual que el anterior.
- RF-40: El sistema debe poder cargar un listado de las líneas deshabilitadas.
- RF-41: El sistema debe poder cargar un listado de las líneas habilitadas.
- RF-42: El sistema debe poder habilitar una línea.
- RF-43: El sistema debe poder notificar si alguna de las paradas de una línea no tiene asignado un horario.
- RF-44: El sistema debe poder deshabilitar una línea.
- RF-45: El sistema debe poder cargar un formulario para añadir un nuevo recorrido.
- RF-46: El sistema debe poder añadir un nuevo recorrido a una línea.
- RF-47: El sistema debe poder notificar si el nombre de un recorrido ya existe.
- RF-48: El sistema debe poder notificar si ya existe un recorrido con un mismo origen y destino.
- RF-49: El sistema debe poder añadir el punto de origen a un recorrido.
- RF-50: El sistema debe poder añadir el punto de destino a un recorrido.
- RF-51: El sistema debe poder cargar un listado de todos los recorridos registrados agrupados por líneas.

- RF-52: El sistema debe poder notificar la no existencia de recorridos registrados.
- RF-53: El sistema debe poder eliminar un recorrido.
- RF-54: El sistema debe poder cargar un formulario para añadir un nuevo punto a un recorrido.
- RF-55: El sistema debe poder añadir un punto a un recorrido.
- RF-56: El sistema debe poder notificar si un punto ya está registrado en un recorrido.
- RF-57: El sistema debe poder cargar un listado con los puntos de recorrido agrupados por líneas y recorridos.
- RF-58: El sistema debe poder notificar la no existencia de puntos de recorrido.
- RF-59: El sistema debe poder eliminar un punto de recorrido.
- RF-60: El sistema debe poder notificar que un recorrido solo está compuesto por dos paradas.
- RF-61: El sistema debe poder eliminar el horario de un punto de recorrido.
- RF-62: El sistema debe poder cargar un formulario con el recorrido y número de horas que se desean añadir para crear un horario.
- RF-63: El sistema debe poder cargar un formulario con las paradas pertenecientes a un recorrido y las horas que se añadirán para crear un horario.
- RF-64: El sistema debe poder asignar un horario a un punto de recorrido.
- RF-65: El sistema debe poder notificar que una hora ya está registrada al asignar un horario.
- RF-66: El sistema debe poder cargar un formulario con las paradas que pertenezcan a recorridos para poder consultar sus horarios.
- RF-67: El sistema debe poder cargar un listado de los horarios que tiene asignados una parada, separados por recorridos y tipo de días a los que afectan.
- RF-68: El sistema debe poder notificar que un punto de recorrido aún no tiene asignado un horario.
- RF-69: El sistema debe poder eliminar un horario perteneciente a un punto de recorrido.
- RF-70: El sistema debe poder cargar un formulario para la creación de una nueva notificación.
- RF-71: El sistema debe poder registrar una nueva notificación.
- RF-72: El sistema debe poder notificar que la fecha de inicio de una notificación es anterior a la fecha actual.
- RF-73: El sistema debe poder notificar que la fecha de finalización de una notificación es anterior a la fecha de inicio.
- RF-74: El sistema debe poder cargar un listado de las notificaciones ordenadas por su fecha de modificación.
- RF-75: El sistema debe poder eliminar las relaciones de una notificación con las líneas a las que afecte.
- RF-76: El sistema debe poder eliminar las relaciones de una notificación con las paradas a las que afecte.
- RF-77: El sistema debe poder eliminar una notificación.

- RF-78: El sistema debe poder modificar la descripción de una notificación.
- RF-79: El sistema debe poder notificar que la descripción de una notificación que se pretende modificar es igual que la anterior.
- RF-80: El sistema debe poder cargar un formulario con un listado de paradas y notificaciones para asignarlas.
- RF-81: El sistema debe poder asignar una notificación a una parada.
- RF-82: El sistema debe poder notificar que una notificación ya está asignada a una parada.
- RF-83: El sistema debe poder cargar un formulario con un listado de líneas y notificaciones para asignarlas.
- RF-84: El sistema debe poder asignar una notificación a una línea.
- RF-85: El sistema debe poder notificar que una notificación ya está asignada a una línea.
- RF-86: El sistema debe poder crear JSON con la información relevante de la base de datos.
- RF-87: El sistema debe poder enviar los JSON creados.

Aplicación Móvil

En este caso dividiremos los requisitos en dos grupos:

Configuración

- RF-88: El sistema debe poder conectarse a Internet.
- RF-89: El sistema debe poder notificar la ausencia de conexión a Internet.
- RF-90: El sistema debe poder conectarse a la geolocalización del terminal móvil.
- RF-91: El sistema debe poder solicitar la autorización para acceder a la geolocalización del terminal móvil.
- RF-92: El sistema debe poder obtener los JSON de la web.
- RF-93: El sistema debe poder crear la base de datos interna de la aplicación.
- RF-94: El sistema debe poder comprobar las actualizaciones de la base de datos.
- RF-95: El sistema debe poder actualizar las tablas que hayan sido modificadas.
- RF-96: El sistema debe poder cargar los ajustes de la aplicación.
- RF-97: El sistema debe poder cargar los idiomas disponibles.
- RF-98: El sistema debe poder modificar el idioma de la aplicación.
- RF-99: El sistema podrá notificar el cambio de idioma.
- RF-100: El sistema podrá notificar que el idioma seleccionado para el cambio ya está activado.

Operaciones de la APP

- RF-101: El sistema debe poder cargar el menú de la aplicación.
- RF-102: El sistema debe poder cargar un listado de las líneas ordenadas por su número.
- RF-103: El sistema debe poder cargar un listado de los recorridos de los que dispone una línea.

- RF-104: El sistema debe poder cargar las paradas ordenadas que componen un recorrido.
- RF-105: El sistema podrá cargar las paradas de una línea en un mapa.
- RF-106: El sistema podrá visualizar la información del “Bus Turístico”.
- RF-107: El sistema podrá cargar un listado de todas las paradas registradas ordenadas por su nombre.
- RF-108: El sistema podrá realizar una búsqueda por nombre entre las paradas.
- RF-109: El sistema podrá cargar el resultado de una búsqueda por nombre entre las paradas.
- RF-110: El sistema podrá notificar que no existen paradas que coincidan con los criterios de búsqueda.
- RF-111: El sistema podrá cargar todas las paradas en un mapa.
- RF-112: El sistema podrá cargar un listado de todas las paradas de interés turístico.
- RF-113: El sistema podrá mostrar la información de una parada.
- RF-114: El sistema podrá mostrar el tiempo restante para los próximos autobuses de una parada.
- RF-115: El sistema podrá mostrar los recorridos a los que pertenece una parada.
- RF-116: El sistema podrá mostrar los horarios de un punto de recorrido para los días laborales.
- RF-117: El sistema podrá mostrar los horarios de un punto de recorrido para los sábados.
- RF-118: El sistema podrá mostrar los horarios de un punto de recorrido para los festivos.
- RF-119: El sistema podrá mostrar un listado de las paradas favoritas del usuario.
- RF-120: El sistema podrá notificar la no existencia de paradas favoritas.
- RF-121: El sistema podrá mostrar un listado de las notificaciones.
- RF-122: El sistema podrá notificar la no existencia de notificaciones.
- RF-123: El sistema podrá cargar la información de las diferentes tarifas.
- RF-124: El sistema podrá cargar la información de los abonos de transporte.
- RF-125: El sistema podrá mostrar la información de una notificación.
- RF-126: El sistema podrá registrar una nueva parada favorita.
- RF-127: El sistema podrá eliminar una parada favorita.
- RF-128: El sistema podrá geolocalizar al usuario en un mapa.
- RF-129: El sistema podrá mostrar una parada en un mapa.