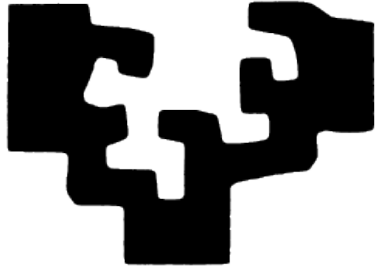


eman ta zabal zazu



universidad
del país vasco

euskal herriko
unibertsitatea

Geminys

Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil





Alumno: D. Miguel José Pérez

Director: D. José Angel Vadillo



Proyecto Fin de Carrera, julio 2014

INDICE



1 - Introducción	8
2 - DOCUMENTO DE OBJETIVOS DEL PROYECTO (DOP)	9
2.1 Antecedentes	9
2.1.1 Motivación	9
2.1.2 Descripción	10
2.2 Objetivos	10
2.2.1 Objetivos del proyecto	10
2.2.2 Objetivos personales	11
2.3 Alcance del proyecto y viabilidad	11
2.4 Planificación	13
2.4.1 Estructura de descomposición del trabajo (EDT)	13
2.4.2 Fases a realizar para lograr el objetivo final	14
2.4.3 Descomposición y explicación de las fases	14
2.5 Diagrama de Gantt	19
2.6 Plan de gestión	20
2.6.1 Gestión del proyecto	20
2.6.2 Gestión del alcance	20
2.6.3 Gestión de los recursos	20
2.6.4 Gestión de las comunicaciones	20
2.6.5 Análisis de riesgos	20
3 - Seguimiento y control	22
3.1 Herramientas	22
3.2 Comparación de tiempos estimados y tiempos reales	22
4 - Estudio preliminar	26
4.1 Elección de herramientas de implementación	26
4.1.1 El lado Servidor	26
4.1.2 El lado Cliente	26
5 - Captura de requisitos y análisis del sistema	28
5.1 Casos de uso	29
5.1.1 C.U. referentes a la autenticación	29
5.1.1.1 C.U. Login & Logout	29
5.1.2 C.U. referentes a la navegación por la documentación	31
5.1.2.1 C.U. Cambiar de publicación (Tab)	31
5.1.2.2 C.U. Seleccionar sección (List)	32
5.1.2.3 C.U. Ver documento	33

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		



5.1.2.4	C.U. Ver instrucción	34
5.1.2.5	C.U. Ver Plan de mantenimiento	35
5.1.3	C.U. referentes a la navegación por el interior de los documentos	36
5.1.3.1	Ir a instrucción desde link.....	36
5.1.3.2	Ver figura	37
5.1.4	C.U. referentes al uso y gestión de grupos de trabajo	38
5.1.4.1	C.U. Crear/Editar G.T (Grupo de trabajo)	39
5.1.4.2	C.U. Ver G.T.	40
5.1.4.3	C.U. Borrar G.T.....	41
5.1.4.4	C.U. Publicar G.T.....	42
5.1.4.5	C.U. Ver lista de usuarios.....	43
5.1.4.6	C.U. Seleccionar usuarios	44
5.1.4.7	C.U. Ver listado de instrucciones.....	45
5.1.4.8	C.U. Seleccionar instrucciones	46
5.1.5	Casos de uso del usuario	47
5.1.5.1	C.U. Ver Grupos de trabajo	48
5.1.5.2	C.U. Seleccionar G.T.	49
5.1.5.3	C.U. Ver instrucción de G.T.....	50
5.1.5.4	C.U. Documentar instrucción de G.T.	51
5.1.5.5	C.U. Sacar fotografía.....	52
5.2	Arquitectura de la aplicación	53
5.3	Descripción de la lógica de datos	55
6	Diseño de la aplicación	58
6.1	Actividad: Inicio de sesión	58
6.2	Actividad: Actividad principal	59
6.3	Actividad: Presentación de documentación técnica	61
6.4	Actividad: Listar grupos de trabajo	63
6.5	Actividad: Grupo de trabajo	65
6.6	Actividad: Hoja de Control	67
6.7	Actividad: Visualizar documento.....	68
6.8	Actividad: Visualizar Instrucción	70
6.9	Actividad: Plan de mantenimiento	70
6.10	Fragmento: Datos generales del grupo de trabajo.....	71

 <p>eman ta zabal zazu universidad del país vasco euskal herriko unibertsitatea</p>	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

6.11	Fragmento: Trabajadores del grupo de trabajo.....	72
6.12	Fragmento: Instrucciones del grupo de trabajo.....	72
6.13	Registrar dispositivo con GCM.....	73
6.14	Copia inicial de datos al dispositivo.....	74
7	- Implementación.....	75
7.1	Permisos.....	75
7.1.1	Permisos de Android:.....	75
7.1.2	Permisos externos de Google:.....	76
7.1.3	Permisos propios:.....	76
7.2	Actividades.....	76
7.3	Visualización del contenido de los documentos.....	77
7.4	Obtención de imágenes de documento.....	83
7.5	Generación de código HTML de documento.....	86
7.5.1	Cambios en XSL.....	86
7.5.2	Cambios en HTML.....	87
7.6	Inicialización de la base de datos.....	88
7.7	Construcción dinámica de la interfaz.....	90
7.8	Interfaz y lógica de grupos de trabajo.....	91
7.9	Publicar grupo de trabajo.....	92
8	- Pruebas realizadas.....	95
8.1	Durante el desarrollo.....	95
9	- Como entregar el producto al cliente.....	97
9.1	Google Play Market.....	97
9.2	Entregar tableta con software.....	97
10	- Conclusiones.....	98
10.1	Conclusiones sobre el proyecto.....	98
a.	Adquisición de nuevas competencias en el ámbito de desarrollo informático.....	98
b.	Mejor entendimiento sobre los procesos de la dirección de proyectos.....	98
c.	Nuevos conocimientos en el área del ferrocarril.....	98
d.	Experiencia en el trabajo empresarial.....	98
e.	Integración en la empresa Geminys S.L.....	98
10.2	Objetivos principales cumplidos.....	98
a.	Mayor navegabilidad.....	99
b.	Independencia de Internet.....	99
c.	Velocidad a la hora de cargar documentos.....	99
d.	Nueva funcionalidad.....	99



	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

10.3 Lecciones aprendidas	99
ANEXO 1 – ACTA DE CONSTITUCIÓN	101
ANEXO 2 – ACTAS DE REUNION.....	102
ANEXO 3 – CIERRE DEL PROYECTO.....	110
ANEXO 4 – MATRIZ DE RASTREABILIDAD	111
ANEXO 5 – MANUAL DE INSTALACION.....	112

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		



INDICE DE FIGURAS

Figura 1 – Diagra de Gantt planificado	19
Figura 2 – Desviación de horas totales.....	23
Figura 3 – Desviación de las fases.....	24
Figura 4 – Desviación de las tareas de cada fase	24
Figura 5 - Login	30
Figura 6 – Actividad principal (Logout)	30
Figura 7 – Publicaciones	31
Figura 8 – Seleccionar sección.....	32
Figura 9 – Ver documento	33
Figura 10 – Navegación completa	33
Figura 11 – Ver instrucción.....	34
Figura 12 – Plan de mantenimiento	35
Figura 13 – C.U Navegación por el interior del documento.....	36
Figura 14 – Clic sobre link	36
Figura 16 – Ver figura.....	37
Figura 17 – Crear / Editar grupo	39
Figura 18 – Ver G.T.....	40
Figura 19 – Eliminar grupo	41
Figura 20 – Publicar grupo	42
Figura 21 – Ver lista de usuarios	43
Figura 22 – Seleccionar usuarios	44
Figura 23 – Ver listado de instruccioens.....	45
Figura 24 – Seleccionar instrucciones	46
Figura 25 – Acceder a Grupos de trabajo.....	48
Figura 26 – Seleccionar G.T.....	49
Figura 27 – Ver instrucción de G.T.....	50
Figura 28 – Documentar instrucción de G.T.	51

 <p>eman ta zabal zazu</p> <p>universidad del país vasco euskal herriko unibertsitatea</p>	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

INDICE DE ILUSTRACIONES

Ilustración 1 Estructura de descomposición del trabajo	13
Ilustración 2 - Casos de uso de Autenticación	29
Ilustración 3 – Casos de uso de navegación exterior.....	31
Ilustración 4 – Casos de uso de grupos de trabajo (Administrador)	38
Ilustración 5 - Casos de uso de grupos de trabajo (Usuario normal).....	47
Ilustración 6 – Arquitectura de la aplicación.....	54
Ilustración 7 – Tablas de grupo de trabajo.....	55
Ilustración 8 – Tabla para registros de dispositivos	56
Ilustración 9 – Tabla para almacenar comunicaciones	56
Ilustración 10 – Relación de tablas para documentación técnica.....	57
Ilustración 11 – Registro GCM.....	73
Ilustración 12 – Matriz de rastreabilidad	111

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

1 - INTRODUCCIÓN

El proyecto que se recoge en este documento, es un proyecto desarrollado en el ámbito empresarial el cual, está orientado a servir como un producto que será ofrecido a los actuales clientes de la empresa Geminys S.L.

Este proyecto va a hacer uso de información de carácter confidencial por lo que los datos entregados serán datos simulados. También va a hacer uso de una plataforma “aplicación web” desarrollada por los trabajadores de la empresa en la que se basan diez de los actuales proyectos que tienen abiertos.

El alumno debe conseguir incorporar todas las funcionalidades del proyecto en la aplicación web existente, sin modificar el comportamiento actual de la misma.

La aplicación web que actualmente existe, da soporte online a la documentación generada para cada proyecto desarrollado en Geminys S.L. Además del soporte, también ofrece una herramienta de edición, la cual permite al usuario editar los documentos de los manuales desde cualquier navegador web. Cada modificación realizada en los documentos se registra y se crea una nueva versión del mismo.



La aplicación web está orientada a un tipo específico de documentación, la documentación de un tranvía. Esta documentación es utilizada por los técnicos de mantenimiento y por los operarios del tranvía para consultarla a la hora de realizar cualquier acción sobre el tranvía.

A los tranvías se le debe realizar una serie de operaciones frecuentemente, para que su estado a lo largo de su vida útil sea óptimo. Estas operaciones se denominan instrucciones, y se recogen en el plan de mantenimiento que se encuentra dentro del manual de mantenimiento de la documentación. Cada una de estas instrucciones tiene una frecuencia asociada, la cual es definida por los técnicos. Estas instrucciones las realizan los operarios del tranvía cuando el jefe de taller lo ordena.

Para verificar que las instrucciones han sido realizadas correctamente, el jefe del taller imprime una hoja de control, incluida en la aplicación web como un PDF, donde se recogen en una tabla todas las instrucciones del plan de mantenimiento. El jefe de taller verifica físicamente que las operaciones se han realizado y si da el visto bueno lo documenta en dicha hoja firmando y poniendo la fecha junto a la instrucción en cuestión.

La aplicación móvil desarrollada implementa una funcionalidad que pretende sustituir a las hojas de control, permitiendo al jefe de taller crear agrupaciones de instrucciones y asignarlas a una agrupación de trabajadores, permitiéndole hacer un seguimiento de la realización de dichas instrucciones. Es decir, una vez hecha la asignación de instrucciones, podrá ver quién y cuándo ha realizado que instrucción. Para la verificar la realización de la instrucción dispondrá de una imagen obtenida por el operario, la cual capturarán con la cámara del dispositivo el trabajo realizado.

Además de esta funcionalidad la aplicación móvil también, dará soporte offline a toda la documentación del tranvía, teniéndola siempre accesible para el usuario.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

2 - DOCUMENTO DE OBJETIVOS DEL PROYECTO (DOP)

2.1 ANTECEDENTES

La empresa donde se va a desarrollar este proyecto se dedica desde hace años al sector de los manuales técnicos de máquina herramienta, ámbito ferroviario, autobuses, cámaras frigoríficas, aire acondicionado, que requieren legamente tener un manual. La empresa comenzó desarrollando manuales en papel y según han ido avanzando las tecnologías se ha ido modernizando el soporte de dichos manuales.

Actualmente los manuales que se desarrollan en Geminys S.L. se entregan tanto en soporte físico como en digital. En cuanto al formato físico podemos enumerar los siguientes:

- Encuadernaciones de papel normal y
- Papeles plásticos resistentes a manchas de aceite

En cuanto a los formatos digitales, por el momento Geminys S.L. ofrece al cliente los siguientes:

- Documentos Word
- Documentos PDF
- Aplicación web que gestiona ficheros XML

2.1.1 Motivación



La tecnología avanza muy rápidamente, hace unos años era impensable integrar internet dentro de las tabletas, las cuales simplemente se usaban como “libros electrónicos”. Hoy en día esta tecnología ha avanzado de una forma increíble, ahora tienen la misma potencia que un ordenador, incluso con acceso a internet.

Las tabletas son algo que están a la orden del día, son muy útiles tanto en el ámbito personal como en el profesional, son portables y es muy fácil aprender a usarlas gracias a que la interfaz es táctil y esto hace que todo sea mucho más intuitivo.

A la empresa en cuestión, Geminys S.L donde se va a desarrollar el Proyecto fin de carrera, en adelante PFC, no le gusta quedarse atrás, de manera que siempre intenta estar a la altura de la competencia. Actualmente la empresa oferta su producto en distintos formatos, según el requerimiento del cliente, hoy en día ofrecen a sus clientes dos tipos de formatos: el clásico manual en formato papel y el soporte Web el cual ofrece una gran variedad de posibilidades relacionadas con la gestión documental de los manuales.

Ya disponen de una infraestructura para la realización y entrega de manuales en formato papel, también disponen de un equipo compuesto por 4 ingenieros informáticos para la realización de manuales interactivos alojados en la Web.

Dentro de su plan tecnológico se encuentra el proyecto de portabilidad. Con este proyecto lo que se quiere conseguir es desarrollar un nuevo soporte con el cual se puedan gestionar todos los documentos técnicos que un manual contiene y además mantener dicha documentación actualizada de una manera transparente para el usuario.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

2.1.2 Descripción

La idea más general del producto que se quiere desarrollar es la de un software para la gestión de toda la documentación técnica que se desarrolle, para un proyecto, en la empresa Geminys S.L. La documentación técnica es la documentación adjunta al vehículo ferroviario entregada al usuario que asegura su correcta utilización y mantenimiento. Dicha documentación a lo largo de su vida útil sufre muchas modificaciones y estas modificaciones deben llegar a todos los usuarios que estén utilizando dicha documentación. Para conseguir esto habrá que desarrollar tres funcionalidades clave:

1. Soporte de gestión documental. (Servidor)

Un sistema que permita insertar/eliminar documentos de forma rápida del servidor con el cual la aplicación de usuario estará sincronizada.

2. Aplicación de usuario para los manuales técnicos. (Cliente)

Aplicación que estará sincronizada con los datos que se encuentren en el servidor indicado. El usuario podrá acceder a los manuales mediante esta aplicación de forma local, es decir, sin necesidad de internet.

3. Soporte de actualización automática. (Servidor y cliente)

Sistema con el cual la documentación descargada por el usuario este actualizada siempre que el usuario disponga de acceso a internet.

Además de la visualización, esta aplicación también servirá para otro tipo de tareas, aun sin especificar, cómo puede ser la realización de informes, seguimiento de realización de tareas...

2.2 OBJETIVOS

En el desarrollo de este proyecto hay varias partes interesadas, por un lado la empresa ya que va a obtener un nuevo producto para ofrecer a sus clientes y por otro lado el propio desarrollador quien adquirirá nuevos conocimientos sobre el ámbito del desarrollo orientado a tabletas.

2.2.1 Objetivos del proyecto



El objetivo principal de este TFG es el de desarrollar un software que sirva como nuevo soporte para la gestión de documentación técnica redactada por Geminys S.L y con ello poder sacar un nuevo producto tecnológico a la venta.

Lo que se pretende conseguir con este desarrollo es proporcionar a los clientes un producto con el cual puedan centralizar y controlar muchas de las tareas que realizan mediante los manuales técnicos desde la gestión documental de todos los manuales que tienen hoy en día hasta la capacidad de gestionar equipos de trabajo y comunicar tareas pendientes a los trabajadores.

Utilizando esta herramienta también se quiere ahorrar tiempo al usuario a la hora de hacer búsquedas dentro de la documentación.

Como objetivos parciales se pueden enumerar los siguientes:

- Almacenar y visualizar información en la tableta
- Realizar una interfaz para la aplicación adaptable a cualquier dispositivo.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

- Realizar un sistema de actualización automática con notificaciones gestionado por la empresa emisora de la documentación, Geminys S.L.

Lo que se pretende con la construcción de esta aplicación es abarcar uno de los puntos del plan de tecnología e innovación de la empresa, construyendo lo anteriormente mencionado e incluyendo los requerimientos que los responsables crean necesarios.

En cuanto a la forma de realizar dicha herramienta, los objetivos que se plantean son los siguientes:

- Gran independencia entre el nivel de presentación y el resto de niveles. De esta forma se podrá rediseñar la interfaz las veces que sean necesarias para adaptarla al cliente sin que se vea afectado el comportamiento de la lógica de negocio.
- Alto grado de parametrización de la aplicación. Con esto se conseguirá que se pueda configurar la misma aplicación para distintas documentaciones, sin necesidad de reescribir código. Se conseguirá un producto software reciclable el cual servirá para muchos de los proyectos desarrollados en Geminys S.L.
- Conseguir una modularidad alta de la aplicación de manera que al editar un modulo otro no se vea afectado por los cambios de este y viceversa.

2.2.2 Objetivos personales

En cuanto a nivel formativo se pretende con este proyecto, lo principal es el desarrollo de la capacidad de planificación. Aunque a nivel más técnico sería el aprendizaje de:

- Nuevas herramientas, tanto de diseño de interfaces como de programación.
- Uso de diagramas UML para la realización del proyecto
- Capacidad de trabajo en otros sectores –en este caso la gestión de la documentación técnica- para recabar las necesidades y forma de trabajo de los mismos, sin conocimiento previo.
- Utilización de los archivos xsl para dar formato a los datos de los ficheros xml.
- Implementación de códigos QR y Notificaciones Push en Android.

2.3 ALCANCE DEL PROYECTO Y VIABILIDAD



Para poder entender la dimensión del alcance del proyecto, hay que tener en cuenta que:

- La cantidad de información con la que se trabaja en el área de los manuales técnicos es inmensa, ya que en ella se incluyen todos los detalles.

Detalles de cómo solucionar cualquier tipo de avería que se pueda dar en el tranvía, además de toda la información relacionada con las piezas que constituyen el vehículo, desde los paneles laterales hasta el último tornillo necesario para mantener unidas las manecillas a las puertas.

- Cada cliente tiene sus propios requerimientos a la hora de la visualización, de manera que este proyecto se adaptará a un cliente en concreto y se intentara hacer el sistema lo más genérico posible para que se pueda reutilizar en un futuro.

A la hora de estudiar la viabilidad de este proyecto, hay que tener en cuenta que el alumno va a disponer de información de personas expertas en el área. Es un proyecto tecnológico

<p>eman ta zabal zazu</p>  <p>universidad del país vasco</p> <p>euskal herriko unibertsitatea</p>	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

que la empresa Geminys S.L. ya tenía contemplado en su plan de tecnología e innovación, y todavía por motivos varios no lo habían podido sacar adelante.

Al finalizar el proyecto se espera obtener un prototipo que se pueda entregar al cliente a modo de prueba para ver cuál es su aceptación dentro del área del ferrocarril.

2.4 PLANIFICACIÓN

En el siguiente apartado se expone cual va a ser la planificación de cada parte del proyecto especificando las tareas y las horas que van a ser necesarias para cada una de ellas.

2.4.1 Estructura de descomposición del trabajo (EDT)

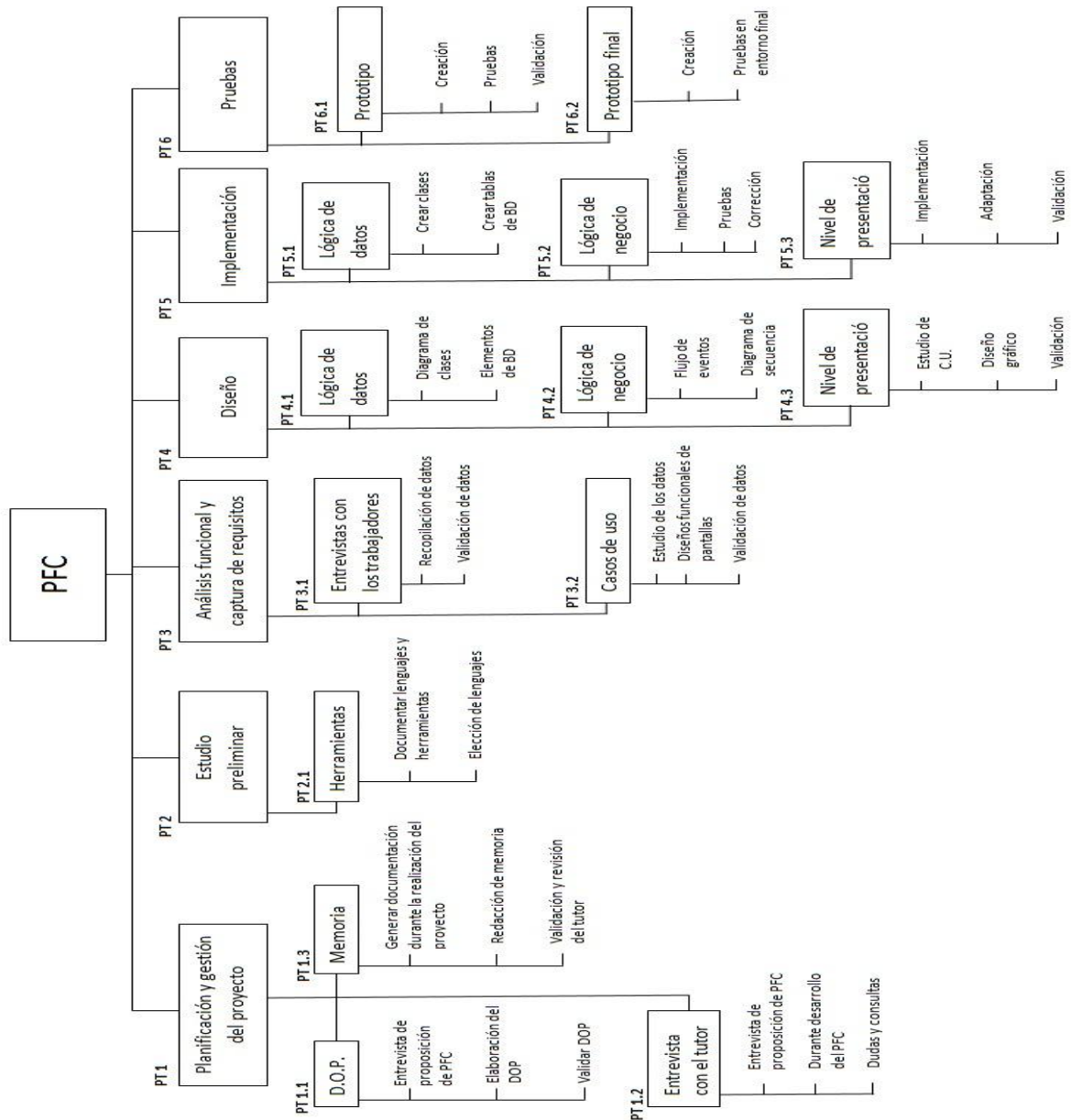




Ilustración 1 Estructura de descomposición del trabajo

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

2.4.2 Fases a realizar para lograr el objetivo final

Con el fin de ser mas explicito, se detallan a continuación los pasos que se deberán realizar para la consecución del proyecto:

Recopilación de los datos necesarios, por medio de entrevistas a los expertos de la empresa Geminys S.L. También se plantea que se haga alguna visita física al lugar de trabajo donde se supone que este proyecto se va a implantar.

Diseños funcionales de las pantallas que va a tener la aplicación. Nivel de presentación.

Una vez definidos los diseños funcionales, diseñar la lógica de negocio y la lógica de datos.

Estudio y aceptación del diseño realizado hasta el momento y búsqueda de posibles fallos. En este punto también se realizará un informe de mejoras deseables para la aplicación.

Implementación de los tres niveles diseñados. Empezando por la lógica de datos, siguiendo con la implementación de la interfaz y terminando con la lógica de negocio. En medida de lo necesario se irán implementando servicios en el servidor para proporcionar datos a la aplicación.

Creación de un prototipo y estudio de los resultados de la prueba del mismo en el entorno final.

Corrección de posibles fallos.

Creación del producto final y realización de las últimas pruebas por si existieran fallos.

2.4.3 Descomposición y explicación de las fases

A continuación se explican las diferentes fases en las que se separa el proyecto y se indicara con una fecha concreta que documentos se entregaran al final de cada parte.

PT 1: Planificación y gestión del proyecto

PT 1.1: D.O.P

Fecha de inicio: 2 Diciembre 2013

Fecha de finalización: 16 Diciembre 2013

Entrada:

Salida: Documento de Objetivos de Proyecto



Actividades:

- Realización del Documento de Objetivos del Proyecto.

Descripción:

La salida de esta actividad se verá modificada ya que formara parte de la memoria del proyecto y no es un documento final.

Estimación: 10 horas

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

PT 1.2: Entrevistas con el tutor

Fecha de inicio: 2 Diciembre 2013

Fecha de finalización: 26 Mayo 2014

Entrada:

Salida: Actas de reunión con el tutor del proyecto

Actividades:

- Reuniones presenciales con el tutor.

Descripción:

Mostrar el avance que se lleva en el proyecto, también se aprovechara para resolver dudas.

Estimación: 10 horas

PT 1.3: Memoria

Fecha de inicio: 16 Diciembre 2013

Fecha de finalización: 26 Mayo 2014

Entrada:

Salida: Documento de la Memoria del P.F.C

Actividades:

- Redactar documentos, realizar diagramas...
- Organización de documentos redactados

Estimación: 80 horas

PT 2: Estudio preliminar

PT 2.1: Herramientas

Fecha de inicio: 17 de Diciembre 2013

Fecha de finalización: 7 Enero 2013

Entrada:

Salida: Documento de elección de herramientas

Actividades:

- Hacer pruebas con distintas herramientas
- Redactar pruebas

Estimación: 20 horas

PT 3: Análisis funcional y captura de requisitos

En esta tarea se trata de obtener todos los datos necesarios para el planteamiento y solución informática de una serie de necesidades por parte de la empresa Geminys S.L para crear un nuevo soporte para los manuales técnicos que desarrollan.



PT 3.1: Entrevistas con los trabajadores

Fecha de inicio: 8 Enero 2014

Fecha de finalización: 25 Abril 2014

Entrada: -

Salida: Acta de reunión y apuntes sobre lo tratado

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

Actividades:

- Recopilación de datos sobre el proyecto.
- Validación de datos.
- Pruebas en el entorno final.

Descripción:

En un primer momento las entrevistas nos servirán para capturar requisitos, pero según vaya avanzando el proyecto estas serán para la validación de los resultados obtenidos.

Estimación: 25 horas

PT 3.2: Casos de uso

Fecha de inicio: 13 Enero 2014

Fecha de finalización: 11 Abril 2014

Entrada: PT 3.1

Salida: Relación de casos de uso que componen la aplicación.

Actividades:

- Estudio de los casos de uso de PT 3.1.
- Creación de los casos de uso.
- Revisión de los casos de uso.

Estimación: 20 horas

PT 4: Diseño

PT 4.1: Lógica de datos

Fecha de inicio: 15 Enero 2014

Fecha de finalización: 25 Abril 2014

Entrada: PT 3

Salida: Diagrama de clases y elementos de base de datos

Actividades:

- Estudio de los casos de uso.
- Realización de diagramas de clases.
- Diseño de elementos de la base de datos.

Descripción:

Las clases y los elementos de la base de datos se pueden ver modificados según los casos de uso, por ello habrá que tener en cuenta todos los casos de uso para la hora de diseñar la base de datos.



Estimación: 20 horas

PT 4.2: Lógica de negocio

Fecha de inicio: 17 Enero 2014

Fecha de finalización: 28 Febrero 2014

Entrada: Salida PT 4 + PT 3.2

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

Salida: Diagrama de secuencia y flujo de eventos.

Actividades:

- Estudio de CU.
- Redacción de flujo de eventos.
- Realización de los diagramas de secuencia.

Estimación: 60 horas

PT 4.3: Nivel de presentación

Fecha de inicio: 25 Enero 2014

Fecha de finalización: 3 Marzo 2014

Entrada: PT 4.1 y PT 4.2 + PT3

Salida: Diseño funcional de todas las pantallas que tendrá la aplicación

Actividades:

- Diseñar pantallas
- Validar diseños

Descripción:

Esta actividad se podrá ver modificada si se realizan nuevos casos de uso.

Estimación: 35 horas

PT 5: Implementación

PT 5.1: Lógica de datos

Fecha de inicio: 26 Febrero 2014

Fecha de finalización: 24 de Marzo 2014

Entrada: PT 4.1 + PT 4.2

Salida: Base de datos e implementación de código de las clases.

Actividades:

- Implementar clases asociadas a las tablas de base de datos.
- Crear base de datos.

Estimación: 30 horas

PT 5.2: Lógica de negocio

Fecha de inicio: 3 Marzo 2014

Fecha de finalización: 18 Abril 2014



Entrada: Salida PT4.2 + PT5.1

Salida: Implementación de la lógica de negocio.

Actividades:

- Diagrama de secuencias finales.
- Pruebas funcionales.
- Resolución de fallos.

Estimación: 65 horas

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

PT 5.3: Nivel de presentación

Fecha de inicio: 18 Febrero 2014

Fecha de finalización: 16 Abril 2014

Entrada: Salida PT 4.3

Salida: Interfaz (software)

Actividades:

- Implementación.
- Pruebas funcionales.
- Validación.

Estimación: 70 horas

PT 6: Pruebas

PT 6.1: Prototipo

Fecha de inicio: 2 Abril 2014

Fecha de finalización: 16 Abril 2014

Entrada: Salida PT 5

Salida: Primera aplicación utilizable (v0.1)

Actividades:

- Creación del prototipo.
- Pruebas.
- Validación.

Estimación: 10 horas

PT 6.2: Prototipo final

Fecha de inicio: 22 Abril 2014

Fecha de finalización: 9 Mayo 2014

Entrada: PT 6.1

Salida: Prototipo modificado, producto final en versión (v1.0)

Actividades:

- Modificar prototipo inicial.

Estimación: 10 horas

Horas totales estimadas: 465 horas

2.5 DIAGRAMA DE GANTT

En la siguiente figura se muestra un diagrama de Gantt que representa la planificación de las tareas arriba descritas.

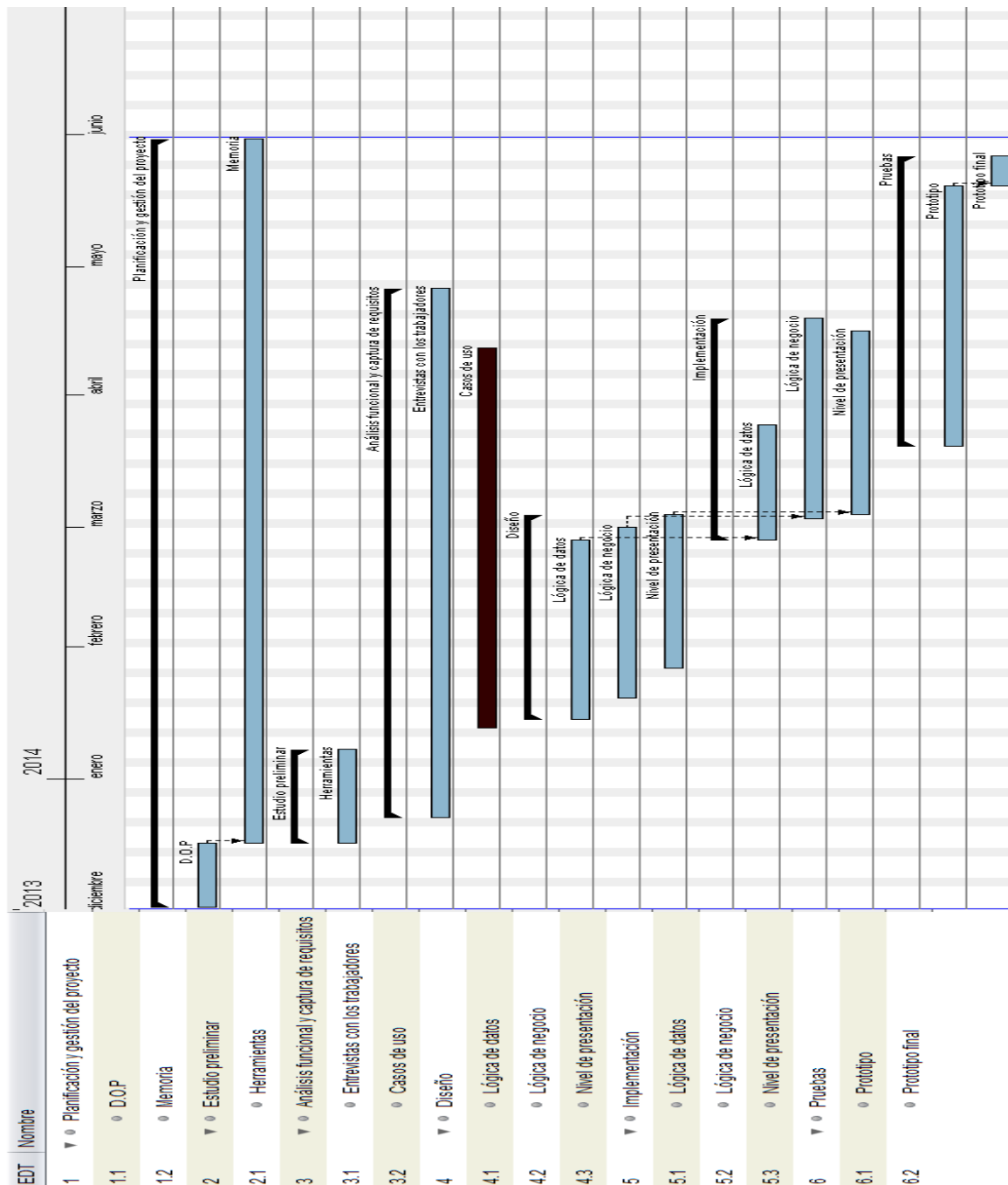




Figura 1 – Diagrama de Gantt planificado

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

2.6 PLAN DE GESTIÓN

2.6.1 Gestión del proyecto

Se llevara un control estricto para cumplir los horarios planteados para cada tarea, ampliando estos, si fuera necesario para poder completar la tarea en los días establecidos. Es decir, ampliando la jornada en vez de ampliando el número de días.

Por lo tanto, si se ve que una tarea va a requerir más tiempo que el estimado, se intentara aumentar el número de horas por día de trabajo, y si esto no fuera posible habría que reducir los días empleados para la siguiente tarea pero con más horas en la jornada.

Por otra parte, si una tarea requiriera menos tiempo que el estimado, se usaría el tiempo para revisar todo lo anteriormente hecho y así poder mejorarlo.

2.6.2 Gestión del alcance

Se intentara por todos los medios que el alcance no se vea afectado, pero seguramente a medida de que el proyecto vaya avanzado y el cliente vea lo que se puede hacer, el alcance se vea modificado, seguramente ampliado. En este caso se hará un estudio de lo que conllevaría la modificación del alcance y se tomaría una decisión.

Si se decide ampliar el alcance, habría que modificar también el calendario. Por lo contrario, si se decide no ampliar el alcance, se dejara anotado como posible mejora para un futuro desarrollo.

2.6.3 Gestión de los recursos

Los recursos necesarios para la realización de este proyecto serán proporcionados al 100% por la empresa, Geminys S.L, proporcionando material de oficina, soporte informático (PC), conocimiento sobre la materia, infraestructura.

Para el apartado de pruebas el alumno será el responsable de elegir una tablet la cual pagará la empresa para la realización de diversas pruebas de la aplicación.

2.6.4 Gestión de las comunicaciones



Se usaran diversos canales de comunicación entre las tres partes del proyecto, siendo el punto en común de todas las partes el alumno. De esta forma se generara comunicación entre el alumno y el tutor del TFG, y el alumno y los trabajadores expertos de la empresa Geminys S.L.

La comunicación entre el tutor y el alumno se realizara mediante correo electrónico y reuniones más o menos frecuentes. Mientras que la comunicación entre el alumno y los trabajadores se realizara de forma física, es decir, mediante reuniones en la empresa. Estas reuniones quedaran documentadas mediante un acta de reunión, registrando quien, cuando, y que temas se trató en la reunión.

2.6.5 Análisis de riesgos

- Mal planteamiento inicial: (50%)

Dado que es el primer proyecto grande que desarrolla el alumno, es posible que su planteamiento inicial no sea del todo correcto, sobre todo en la estimación de los tiempos de las fases.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

Si esto ocurriera, se ha previsto con plazos de hasta 7 días para poder recuperar horas perdidas de las fases.

- Problemas de compatibilización trabajo/proyecto: (20%)

Dado que el alumno ha llegado a un acuerdo de colaboración con la empresa y a la vez realizar el proyecto, es posible que en alguna fase el proyecto se retrase ya que la prioridad es sacar el trabajo adelante y entregarlo en plazo.

En el acuerdo de colaboración se concreto que el alumno haría media jornada trabajando para la empresa y otra media jornada realizando el PFC, por lo que si por algún motivo en alguna fase se vea afectada por la entrega de algún trabajo en la empresa, este tiempo se recuperaría realizando las horas pertinentes de PFC en las horas relativas al trabajo de la empresa.



- Problemas con la implementación (20%)

Aunque ya se tiene claras las herramientas que se van a utilizar para desarrollar el proyecto, es probable que el alumno encuentre dificultades a la hora de implementar algún comportamiento ya que no domina del todo la plataforma Android.

Si se da el caso de que el alumno se atasque a la hora implementar comportamientos, se empleará como mucho dos o tres horas en buscar una buena solución, si tras las cuales no se ha solucionado se procederá a la siguiente tarea para no retrasar la fase.

- Bajas del alumno: (2%)

No se cree que el alumno pueda caer en baja tal que no pueda seguir con el desarrollo del proyecto, pero si esto ocurriera se avisaría al tutor para acordar un nuevo calendario dependiendo de la duración de la baja.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

3 - SEGUIMIENTO Y CONTROL

En este capítulo se describen las tareas que se han llevado a cabo para realizar el control y el seguimiento a lo largo de la vida del proyecto. También se compara los tiempos estimados con los tiempos que en realidad se han invertido.

3.1 HERRAMIENTAS

Para llevar un control de los requisitos del proyecto se ha utilizado una matriz de rastreabilidad, la cual se puede ver en el Anexo 4. Esta matriz se actualizaba cada dos semanas poniendo el estado actual de las tareas (Iniciado, en progreso, finalizada).

Para controlar las horas invertidas en cada una de las fases se ha utilizado la aplicación web que se utiliza diariamente en Geminys S.L. Dicha aplicación la utiliza todo el personal de Geminys S.L para introducir las horas invertidas en los distintos proyectos que existen en la empresa.

Esta herramienta, denominada internamente “Parte laboral”, permite sacar estadísticas de horas invertidas en cada uno de los proyectos y si es el caso también se pueden obtener estadísticas de cada fase del proyecto. Concretamente para este PFC se creó un proyecto con las siguientes fases o como se denominan en la aplicación “capítulos”:

Diseño

Documentación

Implementación

Pruebas

Diariamente he ido introduciendo las horas que he dedicado a las diferentes fases del proyecto y los resultados obtenidos han sido los siguientes:

Fase	Horas invertidas
Diseño	130 horas
Documentación	172 horas
Implementación	200 horas
Pruebas	20 horas
Total	522 horas

Otros aspectos del proyecto como pueden ser las reuniones se han tratado aparte recopilando los tiempos invertidos en las mismas actas las cuales se pueden ver en el Anexo 2.

3.2 COMPARACIÓN DE TIEMPOS ESTIMADOS Y TIEMPOS REALES

En este apartado se comparan los tiempos que se estimaron en un inicio con los que en realidad se han empleado para el desarrollo de todas las fases del proyecto. En las fases en las que se ha producido una desviación notable se explicara el porqué.

En la siguiente tabla comparativa se muestran los tiempos por cada tarea realizada en cada fase junto con los dos tiempos, el estimado y el real.

Tarea	Tiempo estimado (horas)	Tiempo real (horas)	Desviación (horas)
PT 1.1	10	20	+10
PT 1.2	10	2	-8
PT 1.3	80	80	0
PT 2.1	20	20	0
P.T 3.1	25	20	-5
P.T 3.2	20	30	+10
P.T 4.1	20	20	0
P.T 4.2	60	70	+10
P.T 4.3	35	40	+5
P.T 5.1	30	30	0
P.T 5.2	65	80	+15
P.T 5.3	70	90	+20
P.T 6.1	10	10	0
P.T 6.2	10	10	0
TOTAL	465	522	+57

Tabla 1 – Desviación de horas

Observando la desviación global de las horas se puede ver que la estimación inicial no ha sido del todo acertada en la gran mayoría de las tareas. El cómputo general de horas reales invertidas en el desarrollo del proyecto es de **522 horas**, a las cuales hay que sumar las horas de las reuniones con los trabajadores, que se han tenido a lo largo del proyecto para recibir formación sobre la aplicación base que se ha utilizado en la parte del servidor. La suma total de horas entonces sería **535 horas**.

En las siguientes figuras se puede ver de forma gráfica las desviaciones que se han dado en cada tarea del proyecto, también se explica el porqué de las diferencias de horas obtenidas.

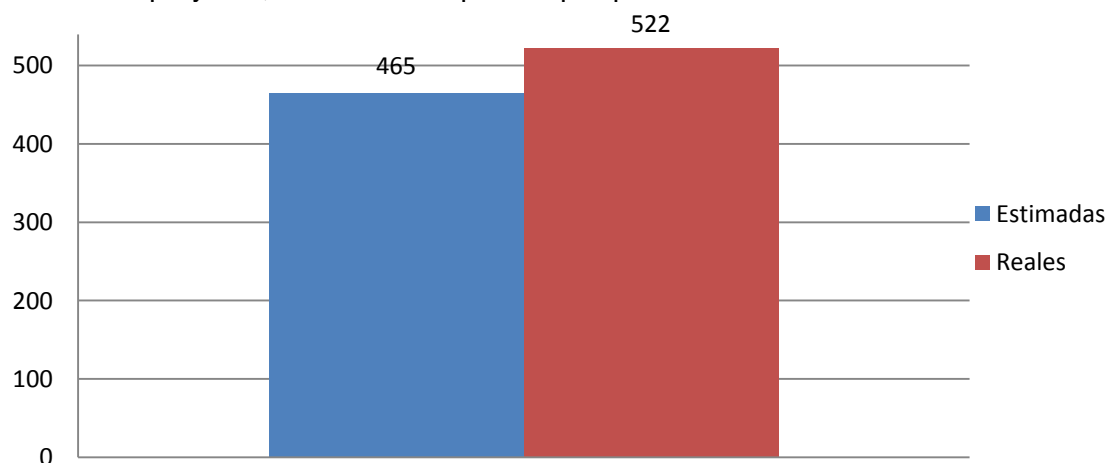


Figura 2 – Desviación de horas totales

Como se puede observar en la Figura 1 se han dedicado **57 horas** más de las planificadas para el desarrollo del proyecto. En la siguiente figura se desglosan las desviaciones por cada fase del proyecto.

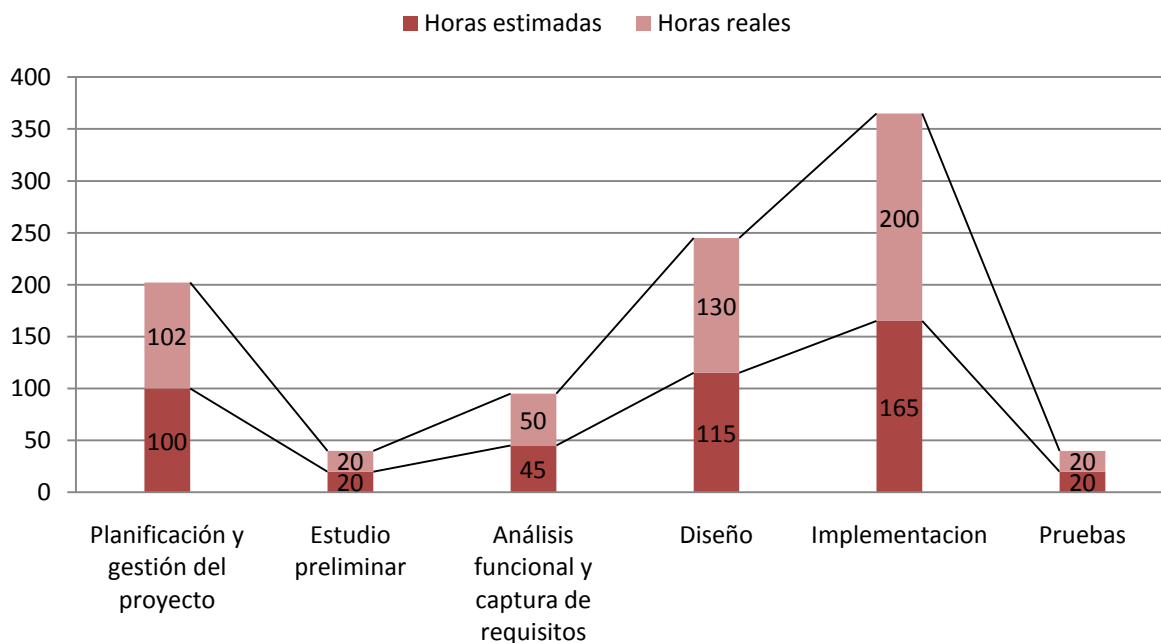


Figura 3 – Desviación de las fases

En la Figura 2 se muestran las horas estimadas frente a las reales por cada fase del proyecto, para aclarar las desviaciones que se ven en los gráficos se muestra en la siguiente gráfica el detalle de horas de cada tarea de las fases:

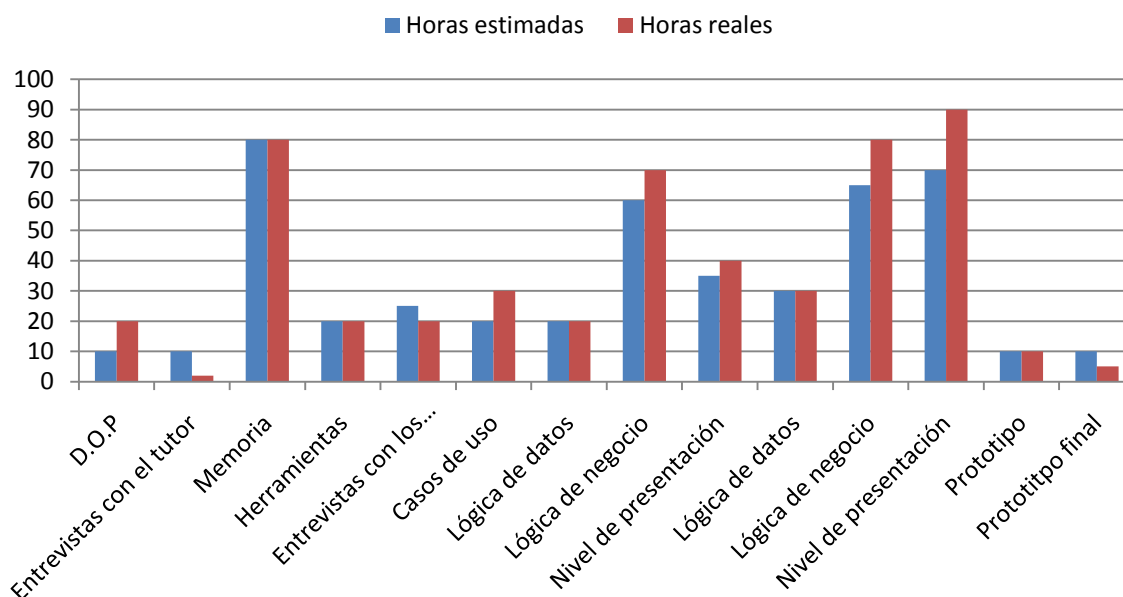




Figura 4 – Desviación de las tareas de cada fase



	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

Como se puede observar en la Figura 3 las desviaciones más notables son en las siguientes tareas:

- Casos de uso (10 horas)
- Implementación: Lógica de negocio (15 horas)
- Implementación: Nivel de presentación (20 horas)

En cuanto a la estimación de tiempo para la tarea de Casos de uso se vio afectada por la cantidad de ellos que surgieron en un primer momento. Fue necesario un filtro para decidir que parte de ellos se iban a desarrollar. Una vez definidos todos ellos se tuvo que tomar decisiones y presentar las mismas a la gerente para que diera el visto bueno.

Los problemas de tiempo que surgieron en la fase de implementación, surgieron porque el alumno no tenía los conocimientos suficientes para implementar los comportamientos que se habían definido. Se perdió mucho tiempo en hacer prueba y error a la hora de implementar las interfaces dinámicas, pero al final invirtiendo tiempo tanto en tiempo libre como en tiempo de trabajo se consiguió sacar adelante con esas horas extras sobre el tiempo estimado.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

4 - ESTUDIO PRELIMINAR

En este capítulo se explica el porqué de la elección de las herramientas que se van a emplear en el desarrollo de este proyecto. El proyecto va a abarcar tanto el lado del servidor como el del cliente, por lo que se tratarán ambas partes.

4.1 ELECCIÓN DE HERRAMIENTAS DE IMPLEMENTACIÓN

A continuación se explican por separado las diferentes herramientas que son necesarias para implementar el cliente y el servidor.

4.1.1 El lado Servidor

La implementación de los códigos alojados en el servidor estarán escritos en Java para lo cual se empleará el entorno de desarrollo NetBeans 7.2.1 que es con el que normalmente se trabaja en la empresa y el alumno ya está acostumbrado a su entorno de trabajo y conoce muchos ajustes de su configuración para poder trabajar de forma óptima con él.

Para poder realizar las pruebas pertinentes habrá un servidor local donde alojar las clases generadas por Netbeans y así probar que el funcionamiento es el esperado. Concretamente se utilizara el servidor Tomcat 7.0.27.0 que es el que viene incorporado en la versión 7.2.1 de Netbeans.

Tomcat es un servidor web que implementa las especificaciones de los Servlet y de las Java Server Pages (JSP) de Sun Microsystems. Tomcat es software libre mantenido por la comunidad de Apache Software Foundation, se puede encontrar tanto el código fuente como el binario. A partir de la versión 4.0 utiliza como contenedor de Servlets Catalina.

Para el almacenamiento de datos se empleara una base de datos de MySQL la cual se administrara con el entorno MySQL WorkBench 6.0 CE. Para manejar la información de la base de datos en la aplicación sería interesante utilizar el framework de Hibernate para abstraernos de las tablas y así usar las clases que Hibernate crea a partir de las tablas de la base de datos.



En el servidor se manejaran archivos XML y XSL los cuales aportaran la mayoría de los datos a visualizar en el lado del cliente, para editarlos se empleará el propio editor del IDE de NetBeans o el notepad++ con un plugin para que coloree y tabule correctamente la sintaxis de estos lenguajes.

Para la administración de la parte del servidor será necesario diseñar una pequeña interfaz grafica donde exista la opción de subir nuevos archivos o eliminar archivos antiguos. Para el diseño de estas interfaces se empleara HTML, CSS 3 y Java Script. Para la edición del código de la interfaz del servidor se usará el propio NetBeans.

4.1.2 El lado Cliente

En este proyecto el cliente será un dispositivo móvil, más concretamente un dispositivo Android con una versión 4.2 o superior del sistema operativo. Para realizar la implementación del código de la app se utilizará el propio entorno de desarrollo Android SDK que proporciona Android en su página web (<http://developer.android.com/index.html>).

Este entorno de desarrollo utiliza Eclipse4.2.1 como plataforma para integrar todos los plugins y features para poder desarrollar aplicaciones para Android. Este entorno permite

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

simular un dispositivo móvil sobre el que ejecutar aplicaciones, pero dicho simulador es ineficiente y se pierde mucho tiempo utilizándolo, por lo que se optará por hacer las pruebas de ejecución directamente sobre un dispositivo real, lo cual conlleva una configuración de la plataforma y del dispositivo para que funcionen de la forma esperada.



Tanto la programación de la lógica de negocio como la interfaz grafica se podrá realizar desde este entorno. Para generar la estructura de la base de datos se empleará el software SQLiteStudio (v 2.1.5) el cual permite generar tablas de forma rápida y sencilla y a la vez genera las sentencias SQL necesarias para incluirlas en el código fuente de la aplicación. Las pruebas que se han realizado con SQLiteStudio has sido las siguientes:

- Prueba bases de datos 1:
 - Crear una base de datos en SQLiteStudio.
 - Introducir base de datos creada en el directorio de bases de datos de la aplicación.
 - Realizar consultas sobre esa base de datos.
 - Resultado: OK, los datos consultados en la aplicación Android son los esperados.

 - Prueba bases de datos 2:
 - Crear una base de datos desde en la aplicación Android.
 - Mediante la vista DDMS que proporciona el Android SDK sacar el fichero de base de datos generado.
 - Visualizar el contenido con el SQLiteStudio.
 - Resultado: OK, los datos obtenidos en SQLiteStudio son los esperados.

 - Prueba de comunicación con el servidor
 - Para realizar la comunicación con el servidor, a la hora de las pruebas se utilizara la dirección IP de la maquina donde se esté ejecutando el servidor ya que si se utiliza "localhost" sería la referencia al "localhost" del simulador y no funcionaría como debiera ser.

 - Prueba descargar datos
 - Crear un servidor local.
 - Crear una aplicación que solicita una página web del servidor.
 - Leer pagina web y almacenarla en la tarjeta SD.
 - Resultado: OK, comportamiento esperado.
- Para realizar la documentación del proyecto se utilizaran las herramientas de Microsoft Office (Word, Excel...).

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

5 - CAPTURA DE REQUISITOS Y ANÁLISIS DEL SISTEMA

En este capítulo se describen y se analizan las distintas partes que componen el sistema de la aplicación, tanto el cliente como el servidor. De esta forma se dará una visión de las distintas funcionalidades con las que va a contar la aplicación.

Para ello se expone tanto el resultado de la captura de requisitos (casos de uso), como su análisis (flujo de evento), así como la interfaz con la que se le ha dado solución al caso de uso.

El sistema completo consta de dos partes (cliente y servidor) en las cuales se pueden subdividir dependiendo de los permisos del usuario. En la parte del cliente tendremos el perfil de usuario normal y usuario administrador mientras que en el servidor solo existirá el administrador.

A continuación se describen los requisitos de alto nivel, los cuales deberá cumplir la aplicación una vez desarrollada.

- Navegabilidad

Dentro de la documentación técnica existen apartados en los que se hacen referencia a otros apartados incluso a otros manuales, se tiene que poder ir desde un sitio a otro pulsando sobre dicha referencia. La navegación entre manuales debe ser intuitiva y fluida.
- Datos actualizados



Uno de los requisitos del sistema es que la documentación técnica que va a gestionar la aplicación este siempre actualizada. Hay que mencionar que como prerrequisito para la aplicación, más en concreto, para el requisito de “siempre actualizada” es necesario tener disponible una conexión a internet. De manera que la aplicación tendrá que comprobar cuando fue la última vez que tuvo conexión disponible y verificar que sus datos estén actualizados.
- Protección de datos

La documentación técnica debe estar de alguna manera protegida para que un tercero ajeno a la empresa no sea capaz de visualizarla en la aplicación. Para ello existirá la autenticación inicial de la aplicación.
- Recursos hardware

La aplicación debe ser capaz de utilizar los recursos hardware de la tablet para sacar fotografías y grabar audio. Deberá recoger dicha información y ser capaz de registrarla debidamente en el servidor para que el administrador pueda verla.
- Creación y difusión de grupos

La aplicación debe tener la opción de crear grupos de trabajo y comunicar a los de que han sido asignados a un grupo de trabajo. Estos grupos de trabajo llevarán asociados unas instrucciones las cuales deberán ir completando y documentando los trabajadores asignados.
- Disponibilidad

La documentación técnica tiene que estar siempre disponible para consulta por parte del administrador ya sea con o sin conexión a internet.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

5.1 CASOS DE USO

En este documento se recogen todos los casos de uso de la aplicación. Se han organizado en las siguientes secciones:

- Casos de uso referentes a la autenticación
- Casos de uso referentes a la navegación por la documentación
- Casos de uso referentes al uso y gestión de grupos de trabajo
- Casos de uso referentes a otras utilidades

5.1.1 C.U. referentes a la autenticación

A continuación se describen los casos de uso que tiene que ver la protección de los datos que se van a manejar en la aplicación.

5.1.1.1 C.U. Login & Logout

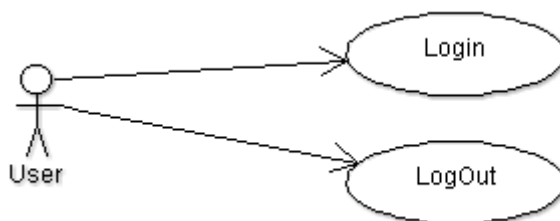


Ilustración 2 - Casos de uso de Autenticación

Login

- **Descripción**

En este caso de uso se presentará una pantalla para que el usuario introduzca el identificador y la contraseña para autenticarse en la aplicación, dependiendo de si el dispositivo tiene conexión a internet o no, la autenticación se hará contra el servidor o de forma local, comprobando en la base de datos del dispositivo los credenciales del usuario.

- **Flujo de eventos:**

El usuario introduce identificador de usuario y contraseña en los campos habilitados.

Pulsa el botón de “login”(1) para comprobar datos.

Si hay conexión a internet se comprueba contra la base de datos del servidor, si no, contra la base de datos local.

Si los datos son correctos, se le da acceso a la pantalla principal de la aplicación y se informa de cómo se ha autenticado (local o remotamente).

Si los datos son incorrectos se le mostrara un mensaje informando de que los datos no son correctos.

- **Interfaz final**




Figura 5 - Login

Logout

- **Breve descripción**

Cuando el usuario quiera dejar de utilizar la aplicación para que otro compañero la utilice, deberá cerrar su sesión para que otro no utilice la aplicación con sus datos personales.

Flujo de eventos:

- **Flujo de eventos:**

Desde la actividad principal de la aplicación pulsar el botón de cerrar sesión.

- **Interfaz final**

Botón de cerrar sesión

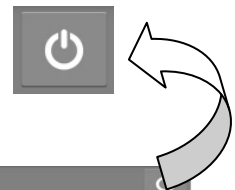


Figura 6 – Actividad principal (Logout)

5.1.2 C.U. referentes a la navegación por la documentación

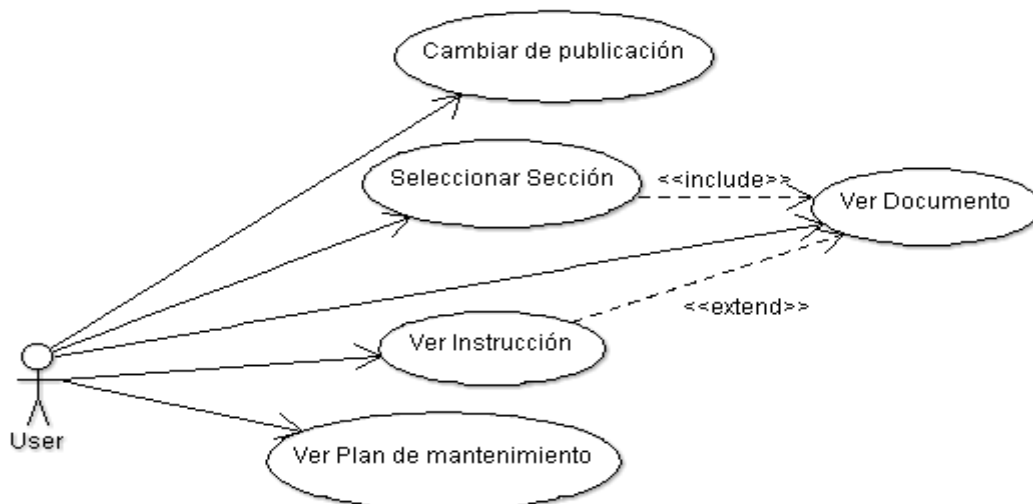


Ilustración 3 – Casos de uso de navegación exterior

5.1.2.1 C.U. Cambiar de publicación (Tab)

- **Breve descripción**

En la estructura de datos que se utiliza para representar los datos en la aplicación, una publicación es el objeto de mayor nivel por el que se puede navegar. Cada publicación tiene asociados un listado de documentos, los cuales se deben mostrar dependiendo la publicación que esté seleccionada en cada momento.

- **Flujo de eventos**



Seleccionar una publicación (Hacer tab sobre una de las pestañas)

- **Interfaz final**



Figura 7 – Publicaciones

La Figura 7 dentro de la Figura 10 está en la parte superior con el número 1 en un círculo.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

5.1.2.2 C.U. Seleccionar sección (List)

- **Breve descripción**

Cuando el usuario está en la pantalla principal de la aplicación se muestran unas pestañas las cuales son las publicaciones, al hacer tab sobre cada una se muestra debajo un listado de secciones asociadas a las publicaciones. Al pulsar sobre una sección se mostrar el contenido relacionado con ella.

- **Flujo de eventos**

Seleccionar una publicación

Seleccionar una sección (Hacer tab sobre un elemento de la lista)

- **Interfaz final**

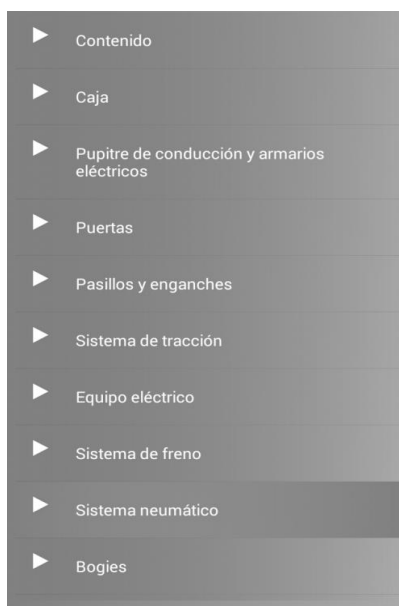


Figura 8 – Seleccionar sección

La Figura 8 dentro de la Figura 10 está en la parte izquierda con el número 2 en un círculo.

5.1.2.3 C.U. Ver documento

- **Breve descripción**
Las secciones tienen asociadas documentos los cuales se muestran al hacer tab sobre una sección. Existen diferentes tipos de documentos dependiendo de la sección que se seleccione. La forma de mostrar los documentos no es siempre la misma.
- **Flujo de eventos**
Seleccionar una sección
- **Interfaz final**

8.1 - SISTEMA NEUMÁTICO		
Avería	Causa	Solución
El compresor no se pone en marcha.	Rotura en el enrollado del motor.	Sustituir motor.
	Protección térmica defectuosa.	
	Pistón u otro componente rotatorio defectuoso.	Sustituir componentes defectuosos.
	Presostato defectuoso.	Ajustar o sustituir presostato, ver GTA-08-501.
La marcha del compresor se prolonga.	Cableado defectuoso.	Revisar cableado.
	Válvula de seguridad mal ajustada.	Revisar tarado de la válvula de seguridad, ver GTA-12-701.
	Válvula antirretorno defectuosa.	Sustituir válvula antirretorno, ver GTA-12-501.
	Anillos de pistón desgastados.	Inspeccionar y cambiar los anillos gastados.
	Filtro de entrada sucio.	Inspeccionar, limpiar y en caso necesario, sustituir, ver GTA-18-001.
El compresor hace mucho ruido.	Cojinetes del pistón/biela defectuosos.	Sustituir cojinetes.

Figura 9 – Ver documento

La Figura 8 dentro de la Figura 10 está en la parte derecha con el número 3 en un círculo.

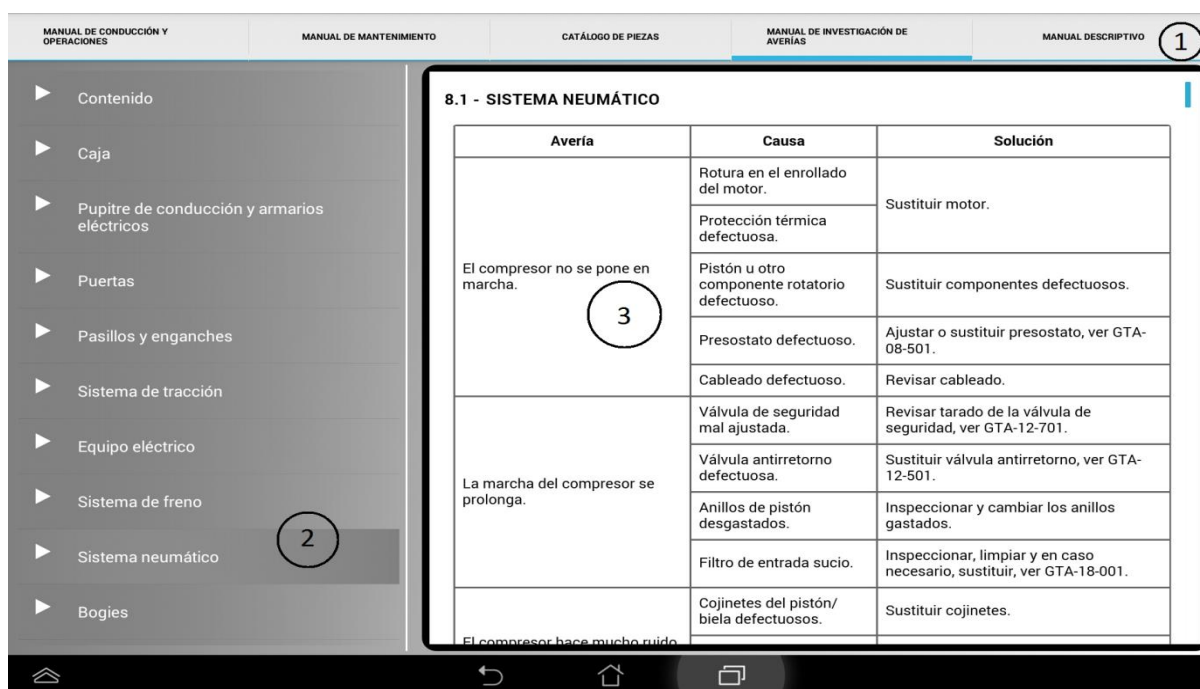




Figura 10 – Navegación completa


 <p>eman ta zabal zazu</p> <p>universidad del país vasco euskal herriko unibertsitatea</p>	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

5.1.2.4 C.U. Ver instrucción

- **Breve descripción**
Este caso de uso es parecido a ver documento ya que las instrucciones son un tipo especial de documento el cual se debe mostrar junto con una serie de metadatos asociados.
- **Flujo de eventos**
Ver Lista de instrucciones o Ver plan de mantenimiento
Seleccionar una instrucción
- **Interfaz final**

INSTRUCCIÓN - CAJ-05-001
Iluminación Interior - Comprobar Funcionamiento


REQUERIMIENTOS DE SEGURIDAD

 LOS REQUISITOS DE SEGURIDAD INCLUIDOS EN LA SECCIÓN 2 - INTRODUCCIÓN DEBEN SER OBSERVADOS ANTES DE EMPRENDER CUALQUIER TRABAJO. CONDICIÓN INICIAL DE SEGURIDAD: CS2

PROCEDIMIENTO

1 - ILUMINACIÓN DE CABINA

- Comprobar que la iluminación de cabina del módulo por donde se ha accedido se enciende automáticamente al entrar en el vehículo sin necesidad de arrancar la unidad.

 El pulsador alumbrado cabina (72S01) tiene una lámpara incluida que está iluminada cuando la luz de cabina está apagada y que se apaga al encender la luz de cabina.

- Comprobar que, una vez encendida la unidad, la iluminación de cabina se enciende y funciona correctamente al actuar sobre el pulsador alumbrado cabina (72S01).
- Comprobar el estado de los difusores y de las lámparas. Sustituirlos en caso necesario, ver CAJ-05-501, CAJ-05-502.
- Comprobar si existen restos de suciedad. Limpiar si es necesario, ver CAJ-05-901.

Si el funcionamiento no es correcto, reparar la avería, ver Manual de Investigación de Averías (MIA), Apartado 11.1.

2 - ILUMINACIÓN SALA

- Arrancar la unidad y comprobar que la iluminación de emergencia/limpieza se enciende automáticamente y funciona correctamente.

Figura 11 – Ver instrucción

5.1.2.5 C.U. Ver Plan de mantenimiento

- **Breve descripción**
Al igual que el caso de uso ver lista de instrucciones, ver plan de mantenimiento es otro tipo de documento que contiene un listado de instrucciones las cuales se muestran dependiendo de los metadatos asociados.
- **Flujo de eventos**
Seleccionar publicación
Seleccionar Plan de mantenimiento
- **Interfaz final**

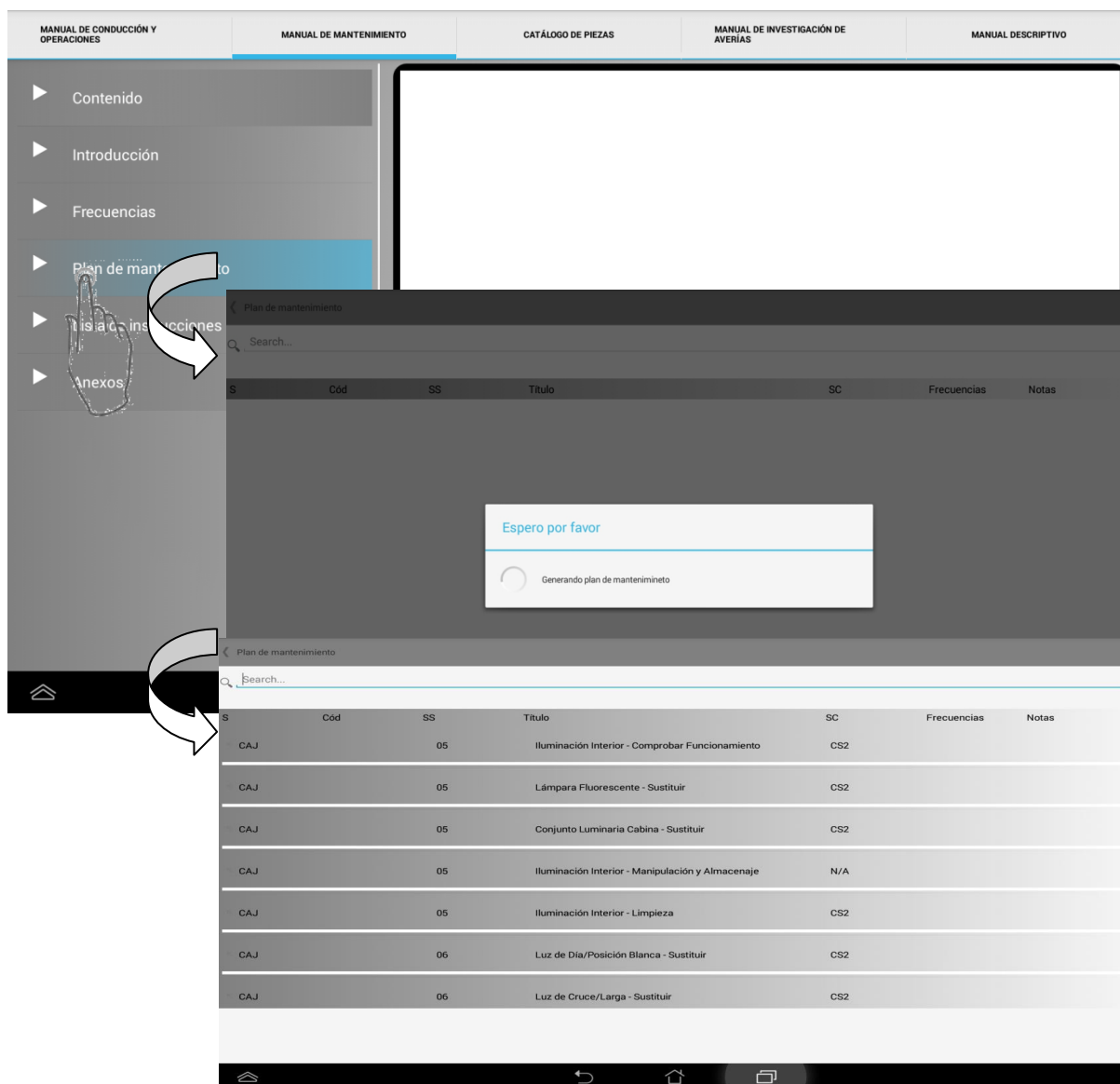


Figura 12 – Plan de mantenimiento

5.1.3 C.U. referentes a la navegación por el interior de los documentos

En este apartado se muestran los casos de uso que se dan a la hora de navegar dentro de los documentos.

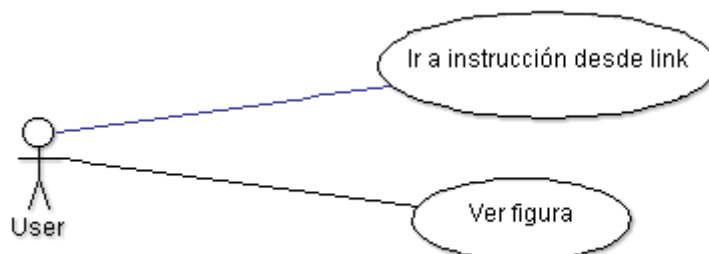


Figura 13 – C.U Navegación por el interior del documento

5.1.3.1 Ir a instrucción desde link

- Breve descripción**
 Dentro de los documentos se hace referencia a otros documentos. Estas referencias deben poderse clicar y visualizar el documento al que está haciendo referencia.
- Flujo de eventos**
 Visualizar documento
 Hacer clic sobre link
 Pulsar botón del dialogo para confirmar la navegación al documento
- Interfaz final**

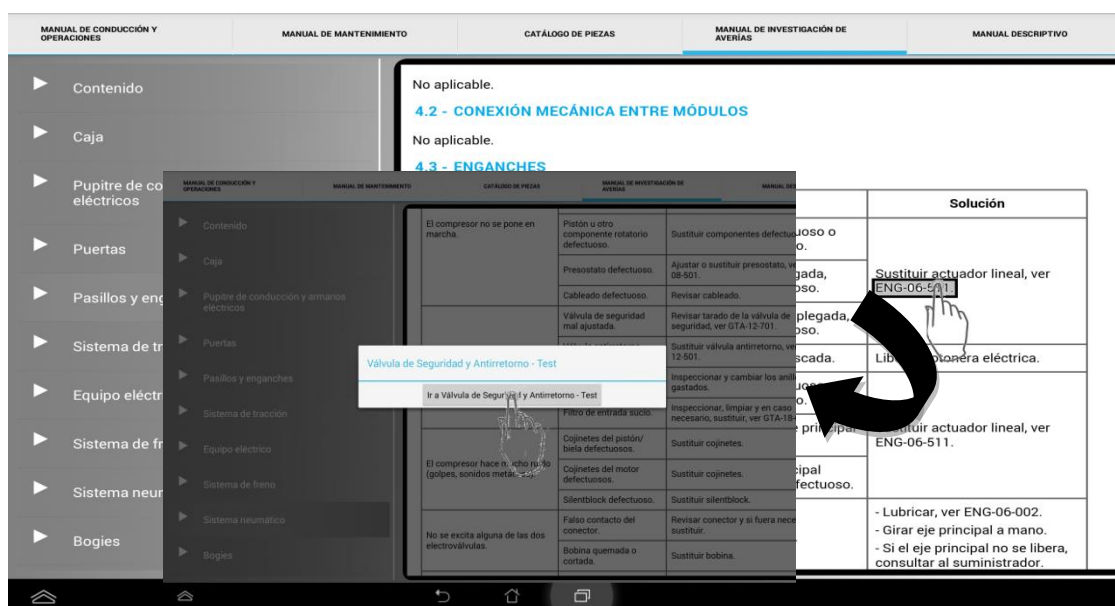


Figura 14 – Clic sobre link

5.1.3.2 Ver figura

- **Breve descripción**

Todas las figuras que se muestran dentro de los documentos se tienen que poder ver aisladamente y poder hacer zoom sobre ellas.

- **Flujo de eventos**

Ver documento

Hacer clic sobre figura

- **Interfaz final**

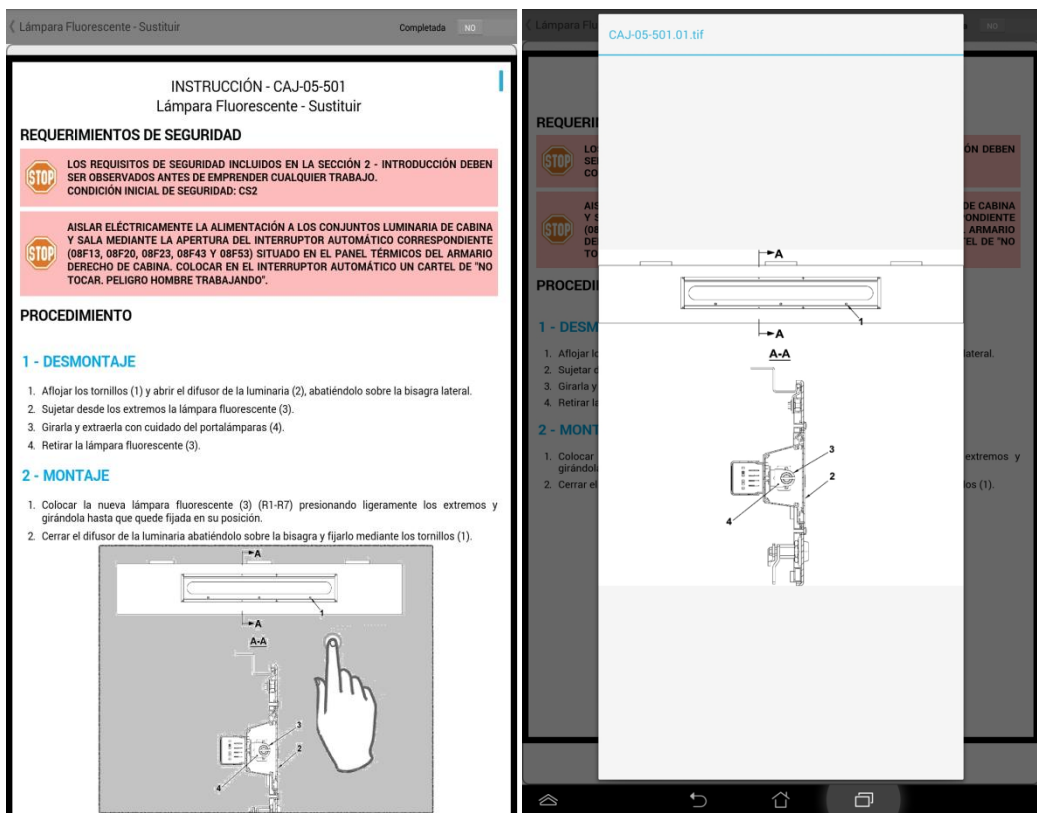


Figura 15 – Ver figura

5.1.4 C.U. referentes al uso y gestión de grupos de trabajo

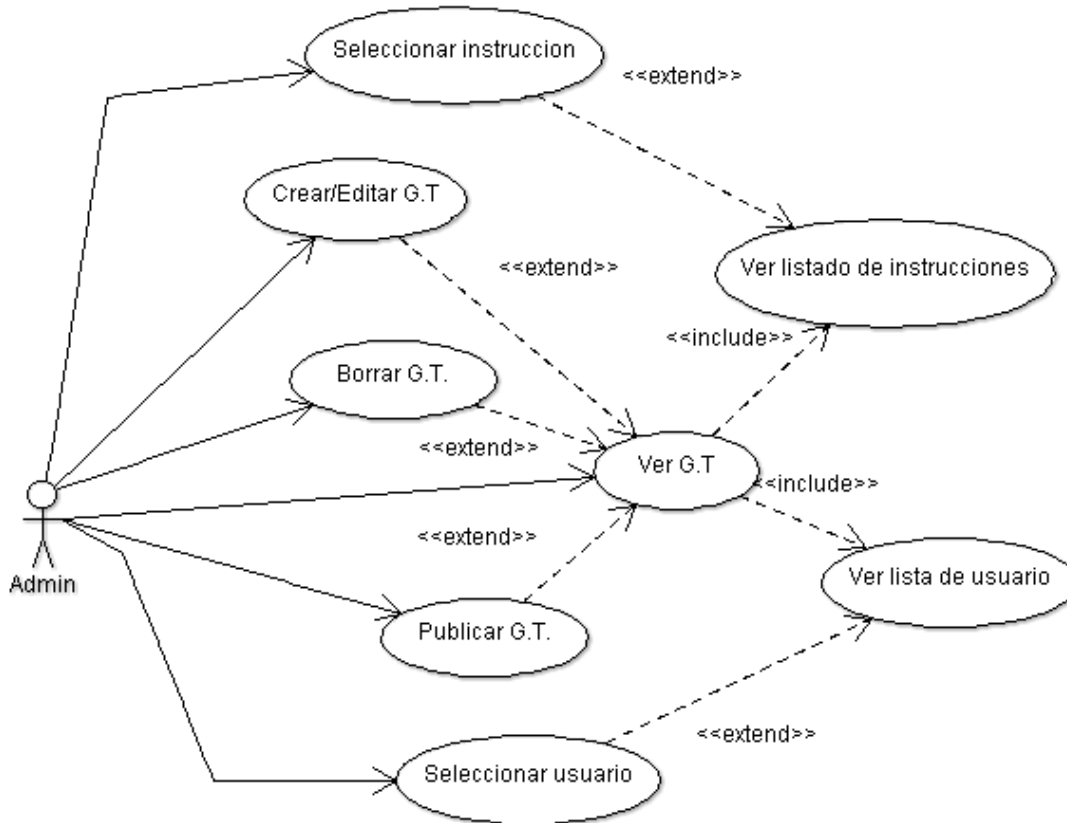




Ilustración 4 – Casos de uso de grupos de trabajo (Administrador)

 <p>eman ta zabal zazu universidad del país vasco euskal herriko unibertsitatea</p>	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

5.1.4.1 C.U. Crear/Editar G.T (Grupo de trabajo)

- **Breve descripción**

El administrador podrá crear un grupo de trabajo, lo que conllevará seleccionar un listado de instrucciones y un listado de usuarios. También deberá rellenar otras propiedades de un grupo de trabajo así como el nombre, la fecha de inicio...

Una vez completadas todas las propiedades del grupo de trabajo podrá guardarlo para su futura publicación.

- **Flujo de eventos**

Desde la pantalla principal -> Grupos de trabajo -> Añadir nuevo (+)

Desde la pantalla principal -> Grupos de trabajo -> Seleccionar un grupo (editar)

- **Interfaz final**

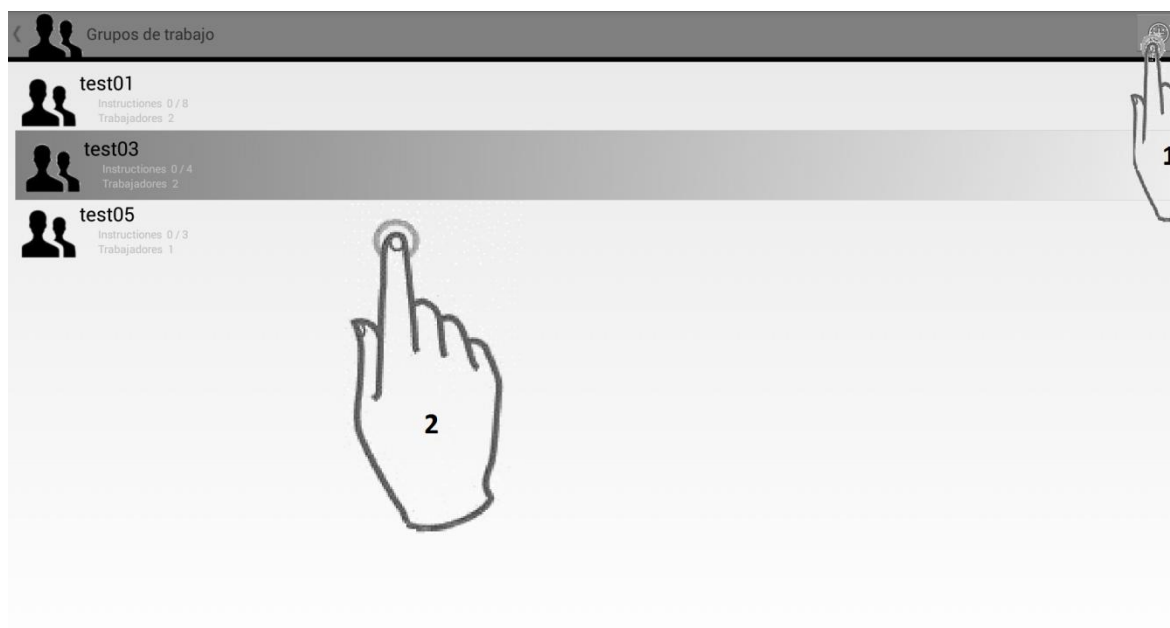




Figura 16 – Crear / Editar grupo

<p>eman ta zabal zazu</p>  <p>universidad del país vasco euskal herriko unibertsitatea</p>	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

5.1.4.2 C.U. Ver G.T.

- **Breve descripción**
El administrador podrá ver todos los grupos de trabajo que han sido creados y están activos. Para ello se le mostrará un listado con la información del nombre, número de trabajadores y numero de instrucciones.
- **Flujo de eventos**
Desde la pantalla principal -> Grupos de trabajo -> Seleccionar un grupo
- **Interfaz final**

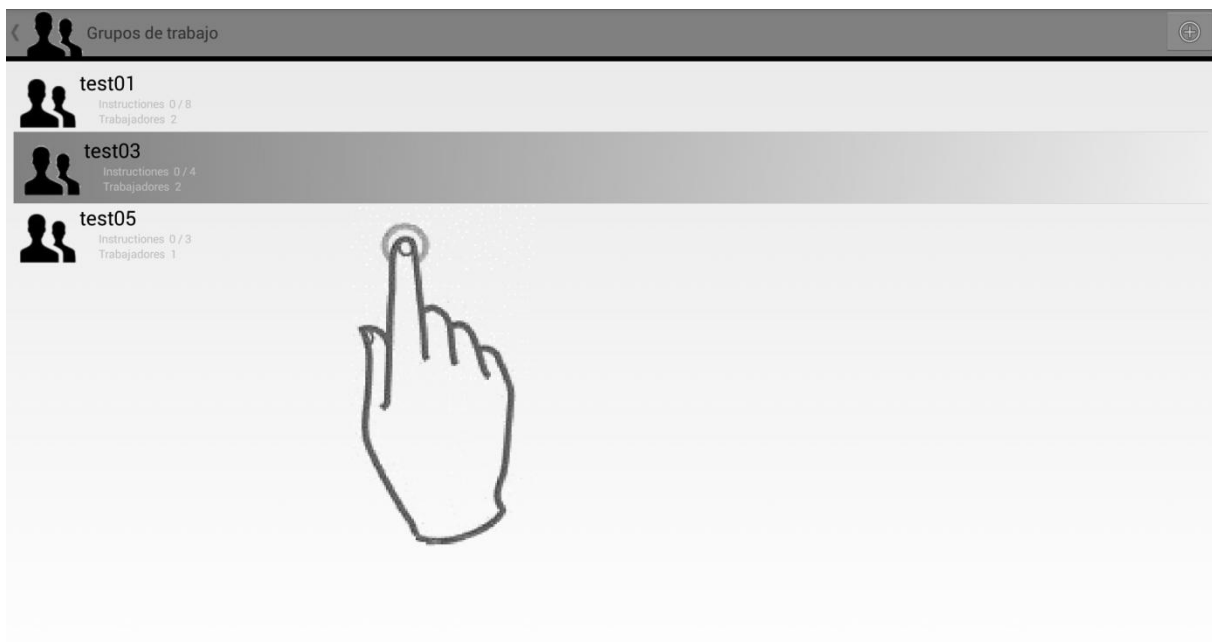


Figura 17 – Ver G.T

5.1.4.3 C.U. Borrar G.T.

- **Breve descripción**
El administrador tendrá la opción de borrar un grupo de trabajo siempre y cuando este no esté ya publicado. El hecho de borrar un grupo de trabajo conlleva borrar todos los registros afectados en la base de datos.
- **Flujo de eventos**
Desde la pantalla principal -> Grupos de trabajo -> Seleccionar un grupo
Pulsar sobre el botón “Eliminar”
- **Interfaz final**

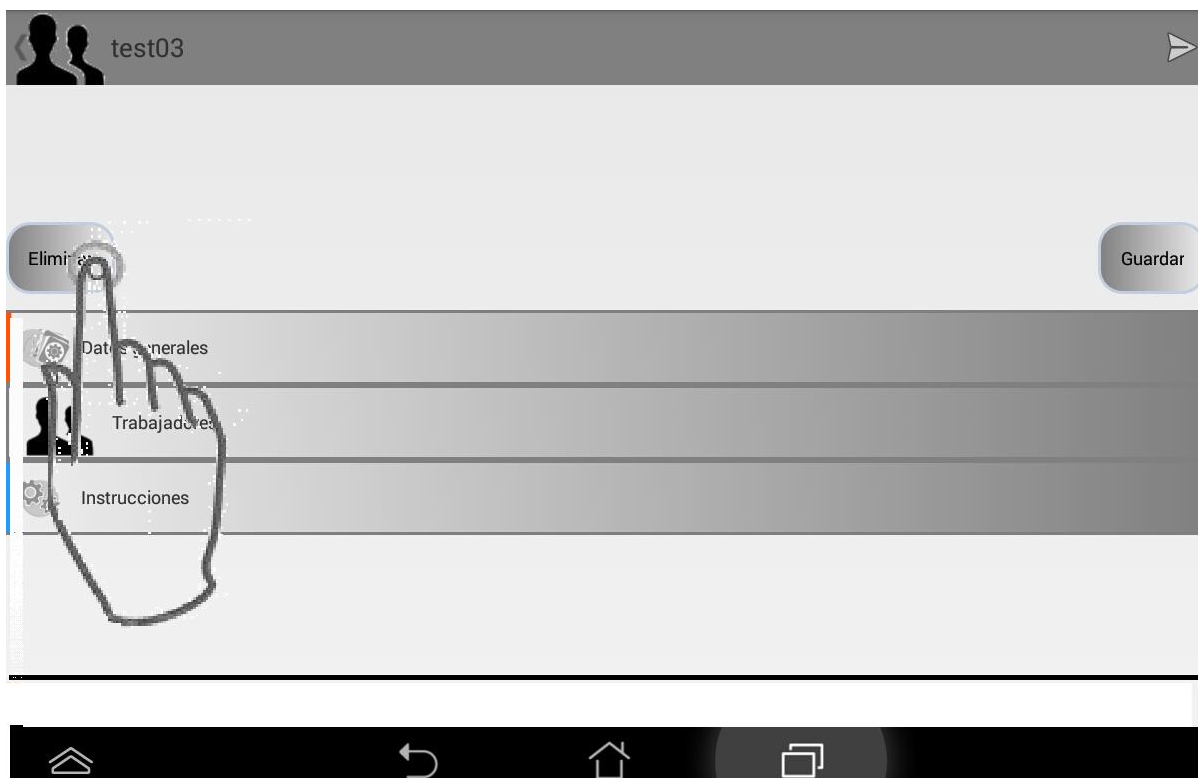




Figura 18 – Eliminar grupo

 <p>eman ta zabal zazu</p> <p>universidad del país vasco euskal herriko unibertsitatea</p>	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p> <p>Proyecto fin de carrera</p>	
---	---	---

5.1.4.4 C.U. Publicar G.T.

- **Breve descripción**

Cuando el administrador considere que el grupo de trabajo ya esté bien definido y todos sus usuarios e instrucciones estén asignados, tendrá la opción de publicar dicho grupo. Al publicar el grupo de trabajo todos los usuarios asignados recibirán una notificación si se encuentran activos en la aplicación. En caso de que no estén activos la siguiente vez que inicien sesión se les notificará que tienen un nuevo grupo de trabajo.

- **Flujo de eventos**

Desde la pantalla principal -> Grupos de trabajo -> Seleccionar un grupo

Pulsar sobre el botón “Publicar”

- **Interfaz final**

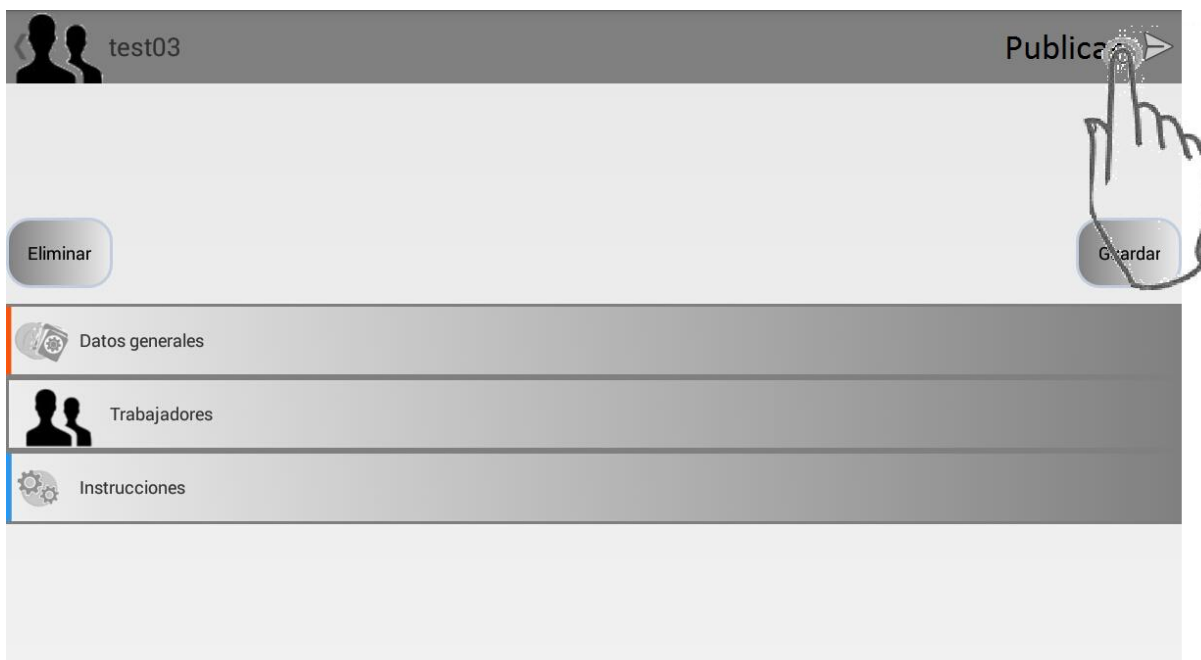


Figura 19 – Publicar grupo

5.1.4.5 C.U. Ver lista de usuarios

- **Breve descripción**
A la hora de crear un grupo de trabajo el administrador tiene que seleccionar un listado de trabajadores los cuales formaran parte de dicho grupo. Para ello hay un botón que muestra todos los usuarios habilitados en la aplicación.
- **Flujo de eventos**
Entrar en un grupo de trabajo(no publicado)
Pulsar sobre el botón “Ver lista de usuarios”
- **Interfaz final**

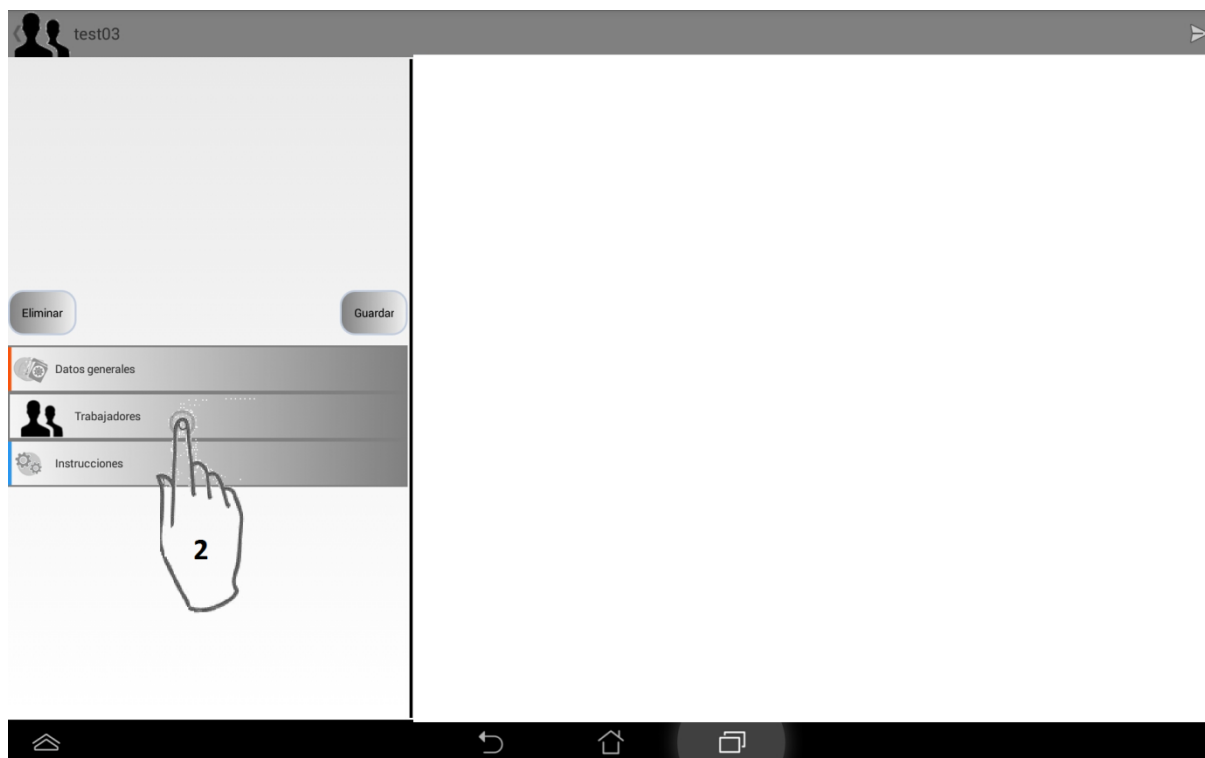


Figura 20 – Ver lista de usuarios

5.1.4.6 C.U. Seleccionar usuarios

- Breve descripción**
 Para la asignación de usuarios a un grupo de trabajo se muestran al administrador todos los usuarios que puede asignar a dicho grupo. Esto es una lista de usuarios los cuales puede seleccionar haciendo tab sobre cada elemento deseado y después pulsando el botón de guardar.
- Flujo de eventos**
 Entrar en un grupo de trabajo no publicado
 Pulsar sobre el botón de “Ver lista de usuarios”
 Pulsar sobre aquellos usuarios que desea seleccionar
 Pulsar sobre el botón de “Guardar”
- Interfaz final**

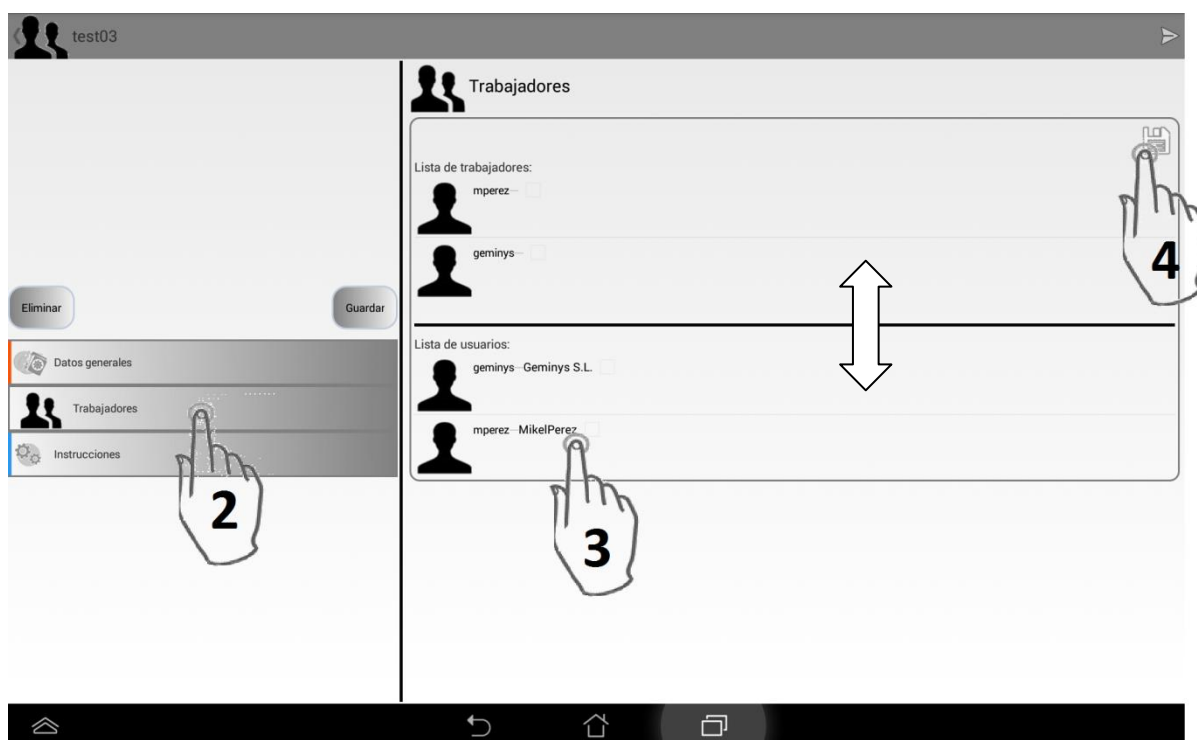




Figura 21 – Seleccionar usuarios

<p>eman ta zabal zazu</p>  <p>universidad del país vasco euskal herriko unibertsitatea</p>	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

5.1.4.7 C.U. Ver listado de instrucciones

- **Breve descripción**
Cada grupo de trabajo tiene su listado de instrucciones asociadas, para ellos el administrador tiene un botón para poder ver el listado de instrucciones desde el grupo de trabajo y así poder seleccionar las que desee.
- **Flujo de eventos**
Entrar en un grupo de trabajo no publicado
Pulsar sobre el botón “Ver lista de instrucciones”
- **Interfaz final**

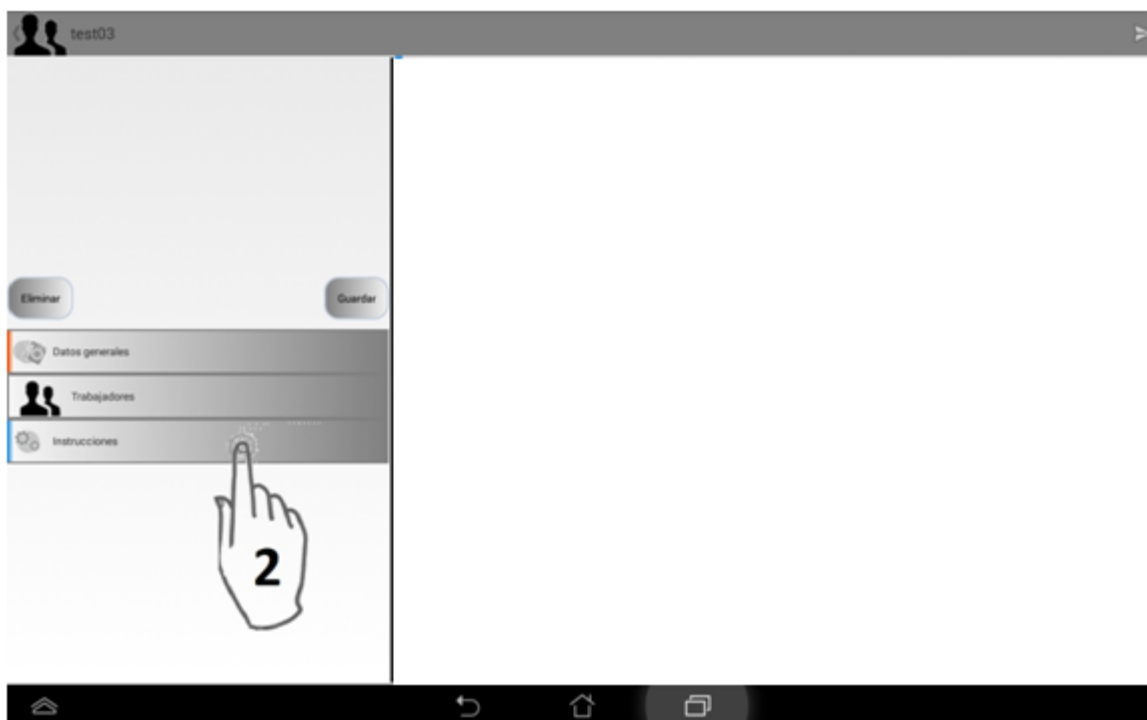


Figura 22 – Ver listado de instruccioens

5.1.4.8 C.U. Seleccionar instrucciones

- **Breve descripción**

Para la asignación de instrucciones a un grupo de trabajo se muestra un listado de instrucciones las cuales se pueden seleccionar y posteriormente asignar al grupo de trabajo pulsando sobre el botón de “Guardar”.

- **Flujo de eventos**

Entrar en un grupo de trabajo no publicado

Pulsar sobre el botón de “Ver lista de instrucciones”

Pulsar sobre aquellas instrucciones que desea seleccionar

Pulsar sobre el botón de “Guardar”

- **Interfaz final**

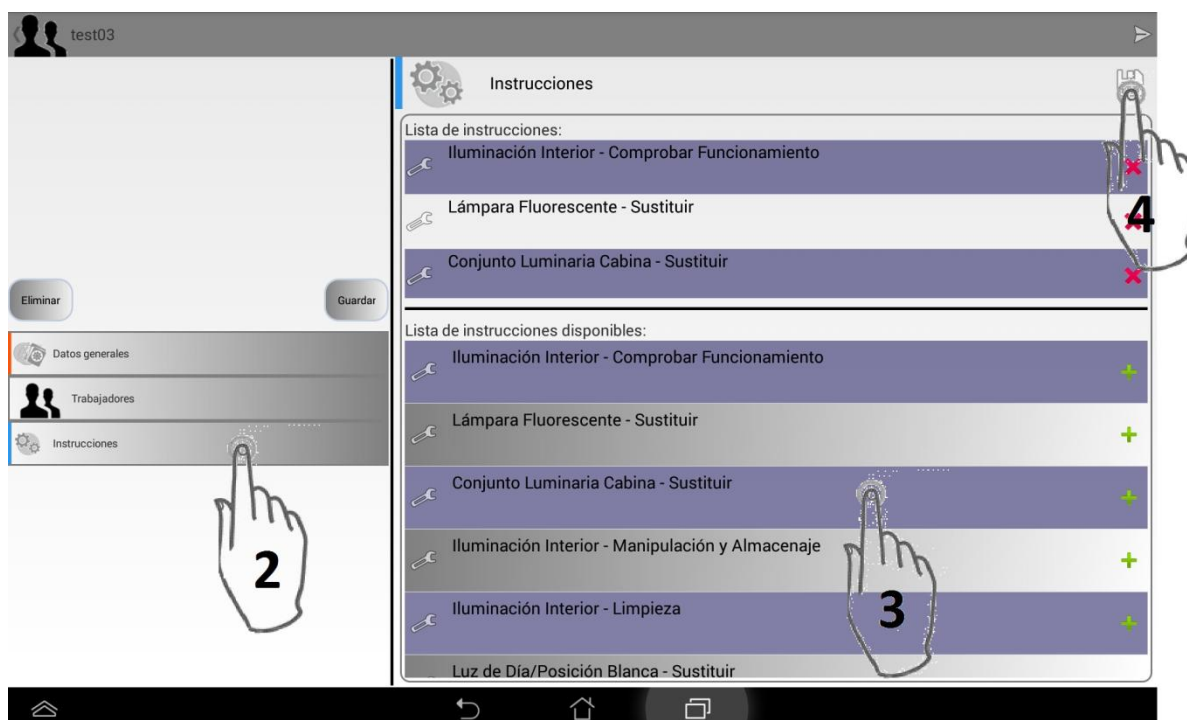


Figura 23 – Seleccionar instrucciones

5.1.5 Casos de uso del usuario

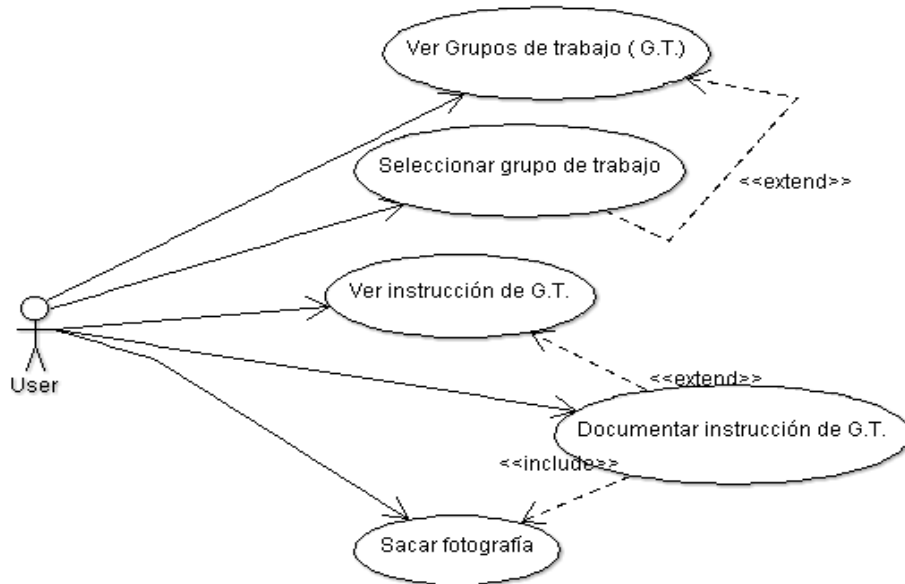




Ilustración 5 - Casos de uso de grupos de trabajo (Usuario normal)



	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p> <p>Proyecto fin de carrera</p>	
---	---	---

5.1.5.1 C.U. Ver Grupos de trabajo

- **Breve descripción**
El usuario tiene la opción de ver los grupos de trabajo en los que ha sido asignado. Se le presentan como un listado con el nombre y otras propiedades identificativas.
- **Flujo de eventos**
Desde la pantalla principal -> Ver Grupos de trabajo
- **Interfaz final**



Figura 24 – Acceder a Grupos de trabajo

<p>eman ta zabal zazu</p>  <p>universidad del país vasco euskal herriko unibertsitatea</p>	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

5.1.5.2 C.U. Seleccionar G.T.

- **Breve descripción**
 Cuando el usuario ya tiene el listado de grupos de trabajo a los que pertenece, puede seleccionar uno de ellos y ver los datos que este lleva asociado.
- **Flujo de eventos**
 Ver grupos de trabajo
 Pulsar sobre el G.T. que se desea ver
- **Interfaz final**

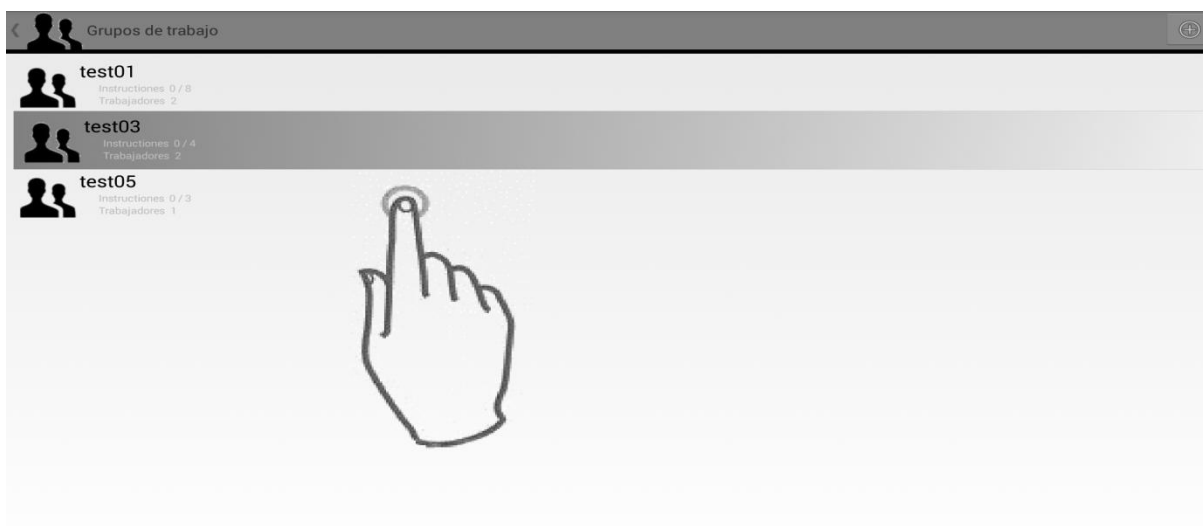


Figura 25 – Seleccionar G.T.

5.1.5.3 C.U. Ver instrucción de G.T.

- **Breve descripción**

Una vez el usuario ha entrado dentro de un grupo de trabajo, puede ver las instrucciones que fueron asignadas por el administrador. Estas instrucciones son las que tendrá que realizar junto con sus otros compañeros del grupo de trabajo.

- **Flujo de eventos**

Ver grupos de trabajo

Pulsar sobre el G.T. que se desea ver

Pulsar sobre la instrucción que desea ver

- **Interfaz final**

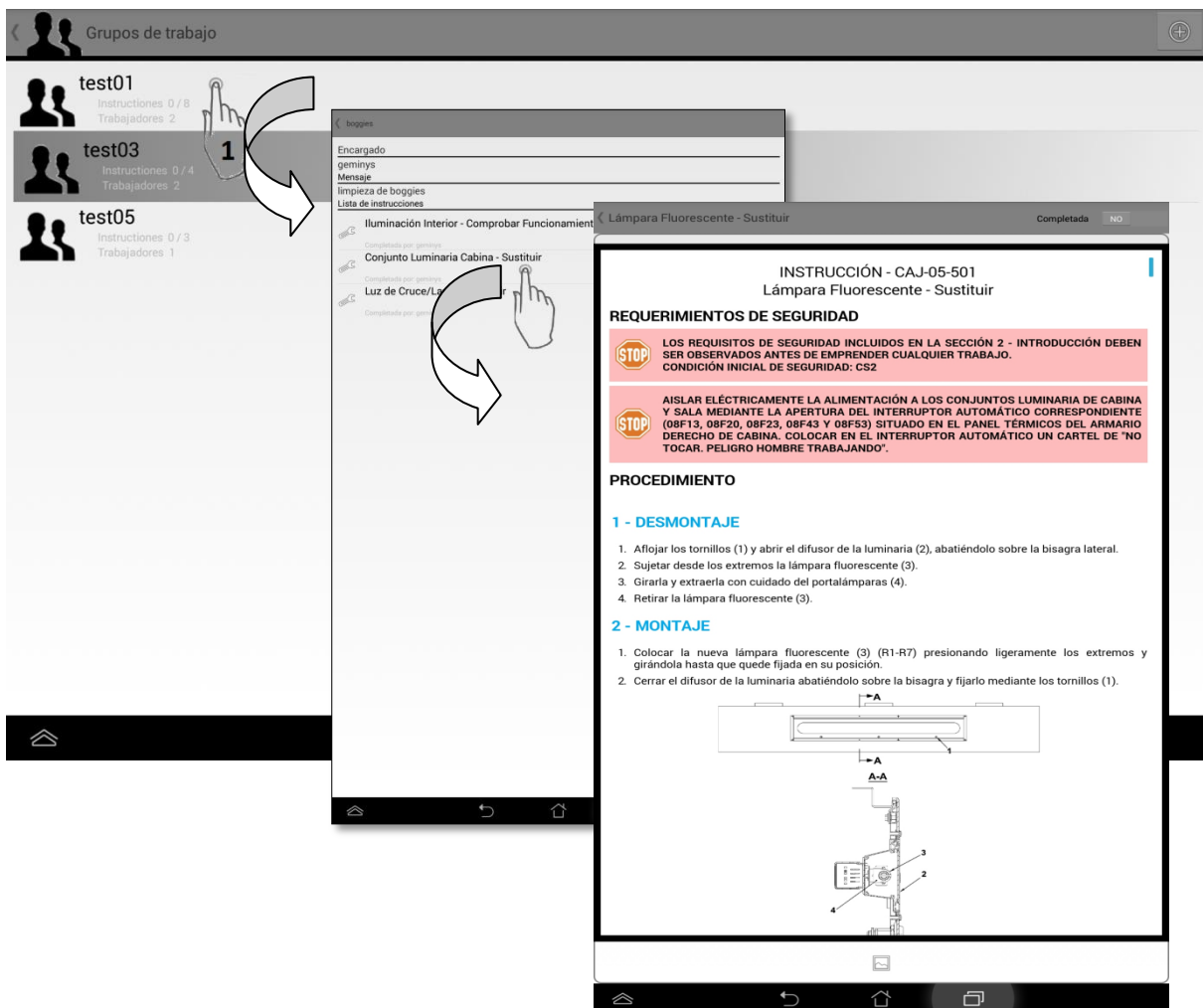




Figura 26 – Ver instrucción de G.T.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

5.1.5.4 C.U. Documentar instrucción de G.T.

- Breve descripción**
 Al finalizar una instrucción de trabajo, el trabajo tiene la opción de adjuntar información sobre el trabajo realizado junto con la confirmación de la finalización de la instrucción.
- Flujo de eventos**
 Ver grupos de trabajo
 Pulsar sobre el G.T. que se desea ver
 Pulsar sobre “Ver lista de instrucciones”
 Seleccionar instrucción de G.T.
 Pulsar sobre el botón de “Completada”
- Interfaz final**

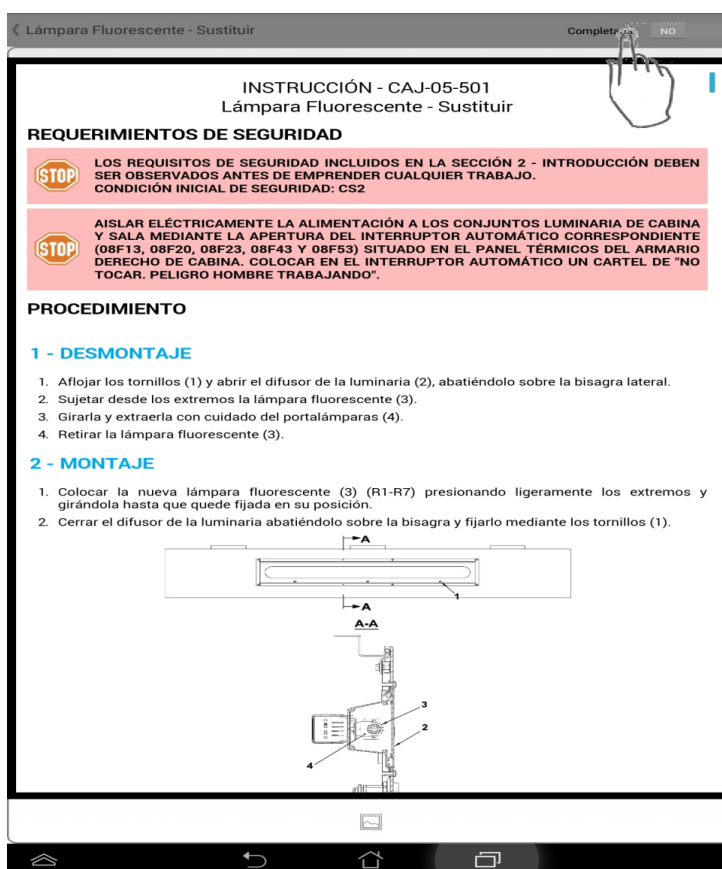




Figura 27 – Documentar instrucción de G.T.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

5.1.5.5 C.U. Sacar fotografía

- **Breve descripción**

Una de las opciones para documentar la finalización de una instrucción es sacar una fotografía al trabajo realizado. Para ello el usuario tendrá la opción de “Sacar fotografía” cuando finalice la instrucción. La fotografía obtenida será adjuntada con la finalización de la instrucción y el administrador podrá verla.



- **Flujo de eventos**

Incluye C.U. 5.1.5.4

Sacar fotografía

- **Interfaz final**

En este caso la interfaz dependerá de la aplicación “Cámara” que esté instalada en el dispositivo.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

5.2 ARQUITECTURA DE LA APLICACIÓN

En el siguiente apartado se describe la arquitectura que va a seguir la aplicación. Gran parte de la arquitectura es la de Cliente-Servidor típico en la que el cliente solicita datos al servidor y este se los proporciona, mientras que en otra parte de la aplicación el servidor será la parte activa de la comunicación mientras que el cliente haga de elemento pasivo que recoge los datos que le envía el servidor y actúa en consecuencia.

En un principio la aplicación cliente estará con una base de datos vacía la cual será una réplica de un conjunto de tablas que hay en el servidor. Este conjunto de tablas que se va a replicar será el necesario para representar en los dispositivos la estructura de los manuales y documentos que están en el servidor. Los datos se deben inicializar dinámicamente ya que está pensado para que la aplicación dependa del servidor con el que se inicializa.

La primera vez que se ejecuta la aplicación será necesario inicializar la base de datos con los que hay en el servidor, para ello en el servidor habrá un servicio que proporcione a los clientes un volcado de las tablas que se deben replicar y en el cliente un método que sea capaz de recoger dichos datos e introducirlos en su base de datos local. También habrá que mover los datos estáticos de la aplicación como pueden ser los pdfs, css, imágenes... a un directorio local del dispositivo para tenerlos accesibles y poder referenciarlos en el código HTML.

La aplicación cliente tendrá que ser capaz de obtener un identificador por parte del servicio de notificaciones de Google y comunicárselo al servidor junto con un identificador de hardware para tener registrados todos aquellos dispositivos en donde se instale la aplicación y así poder mandar notificaciones push. Por la parte del cliente también deberá tener un sistema para poder recibir las notificaciones que le envía el servidor, para ello se utilizan los escuchadores de broadcast de Android.

Aprovechando el trabajo ya realizado para la aplicación web se va a reutilizar el código HTML que se genera a partir de los documentos base XML y las plantillas XSL. Para que puedan convivir la aplicación móvil y la aplicación web sin duplicar las plantillas habrá que añadir parámetros y códigos en las plantillas para que se mantengan ambas visualizaciones. Cada documento está identificado por un código único y el servidor existe un método que genera el HTML a partir de este código. El cliente hará peticiones al servidor con el identificador del documento y recogerá en la respuesta el código HTML que genera. En el cliente existirá un método capaz de almacenar en memoria el contenido HTML que el servidor le ha devuelto.

Toda esta estructura previamente explicada se presenta en la Ilustración 6.

Esquema de comunicaciones entre elementos del sistema:

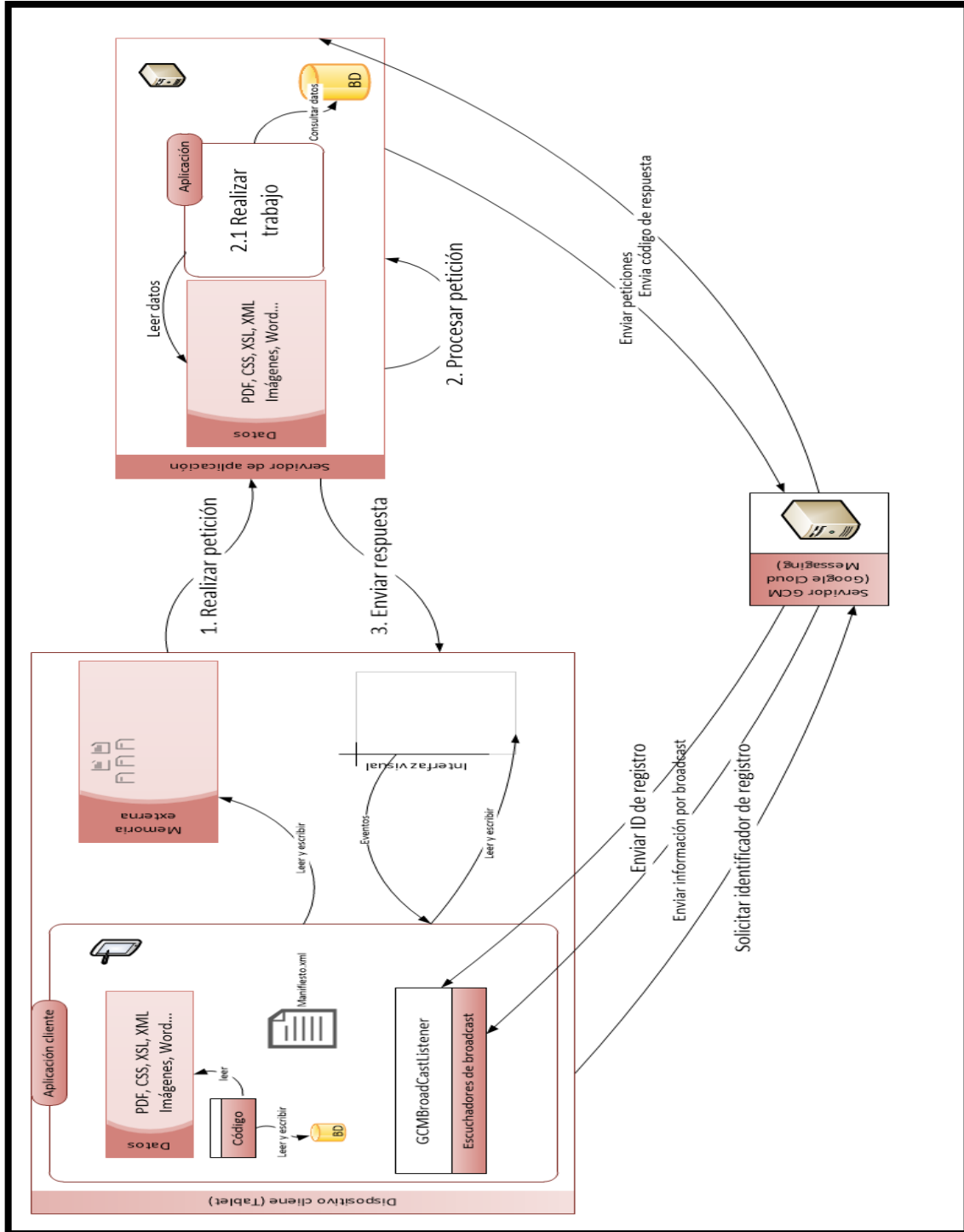


Ilustración 6 – Arquitectura de la aplicación

5.3 DESCRIPCIÓN DE LA LÓGICA DE DATOS

En este apartado se describen todos los elementos de la base de datos que son necesarios para representar los datos en la aplicación. La misma estructura esta replicada tanto en el cliente como en el servidor.

Listado de tablas

Las tablas marcadas con un asterisco (*) son las que se han tenido que añadir a la aplicación web sobre la que se ha desarrollado el proyecto para gestionar los grupos de trabajo y otros aspectos de la aplicación móvil.

- Document
- Documenttitle
- Documenttype
- Frecuency
- Frecuencytitle
- Maintenanceplan
- Maintenanceplanfrecuency
- Manual
- Manualtitle
- Model
- Modelpublication
- Modelpublicationmanual
- Modelsubsystem
- Section
- Sectiontitle
- T_workinggroup*
- T_worker*
- T_wginstruction*
- T_gcm*
- T_gcm_response*

A continuación se muestra las relaciones de las tablas que se usan para administrar los grupos de trabajo.

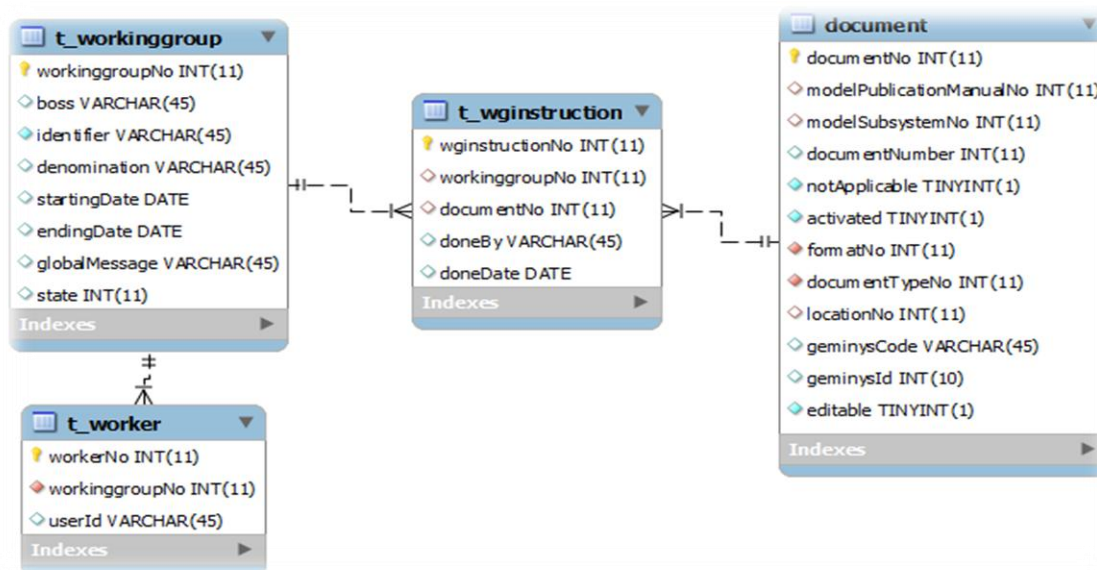




Ilustración 7 – Tablas de grupo de trabajo

	Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil	
Proyecto fin de carrera		

Con las relaciones presentadas en la ilustración 7 se consigue tener un sistema donde puedan existir todas las combinaciones de instrucciones y usuarios que se nos ocurra. Una instrucción consta de un documento la cual tiene el contenido de dicha instrucción. La instrucción estará asociada a un grupo de trabajo el cual tendrá usuarios relacionados. Para saber si una instrucción esta completada o no, se utilizan los campos doneDate y doneBy para almacenar quien y cuando realizo esa instrucción.

En la parte del servidor se deben almacenar los identificadores de registro que devuelve el servicio de mensajería de Google, GCM. Para ello se utiliza la tabla t_gcm.



Ilustración 8 – Tabla para registros de dispositivos

En esta tabla estarán todos los identificadores de dispositivo los cuales se utilizan para realizar las notificaciones push. El campo geminys_reg_code es el código que se obtiene cogiendo una referencia del hardware del dispositivo y el gcm_registration_id es el identificador que devuelve el servicio de google cuando se solicita un nuevo registro.

Para tener un registro de todas las notificaciones que se realizan se va a utilizar una tabla donde se almacene el mensaje que se ha enviado junto con la respuesta que se ha recibido del servicio de google, estos mensajes, tanto el que se envía como el que se recibe son texto con formato JSON, por lo que los campos de la base de datos deben soportar textos muy largos.

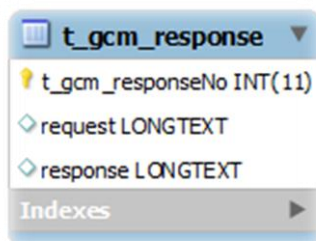


Ilustración 9 – Tabla para almacenar comunicaciones

La siguiente relación de tablas recoge la información necesaria para la estructura de toda la documentación técnica. El elemento más pequeño es el documento, que es lo que tiene el contenido de la documentación, este pertenece a un modelo el cual está contenido en un manual. Todo esto está contenido en una publicación la cual puede tener 1 o n.

Se trata de un modelo de base de datos que se reutiliza en más de un proyecto desarrollado en Geminys S.L. y todo lo que esté relacionado con textos como títulos y otros aspectos que dependen del idioma tienen una tabla con el sufijo "title".

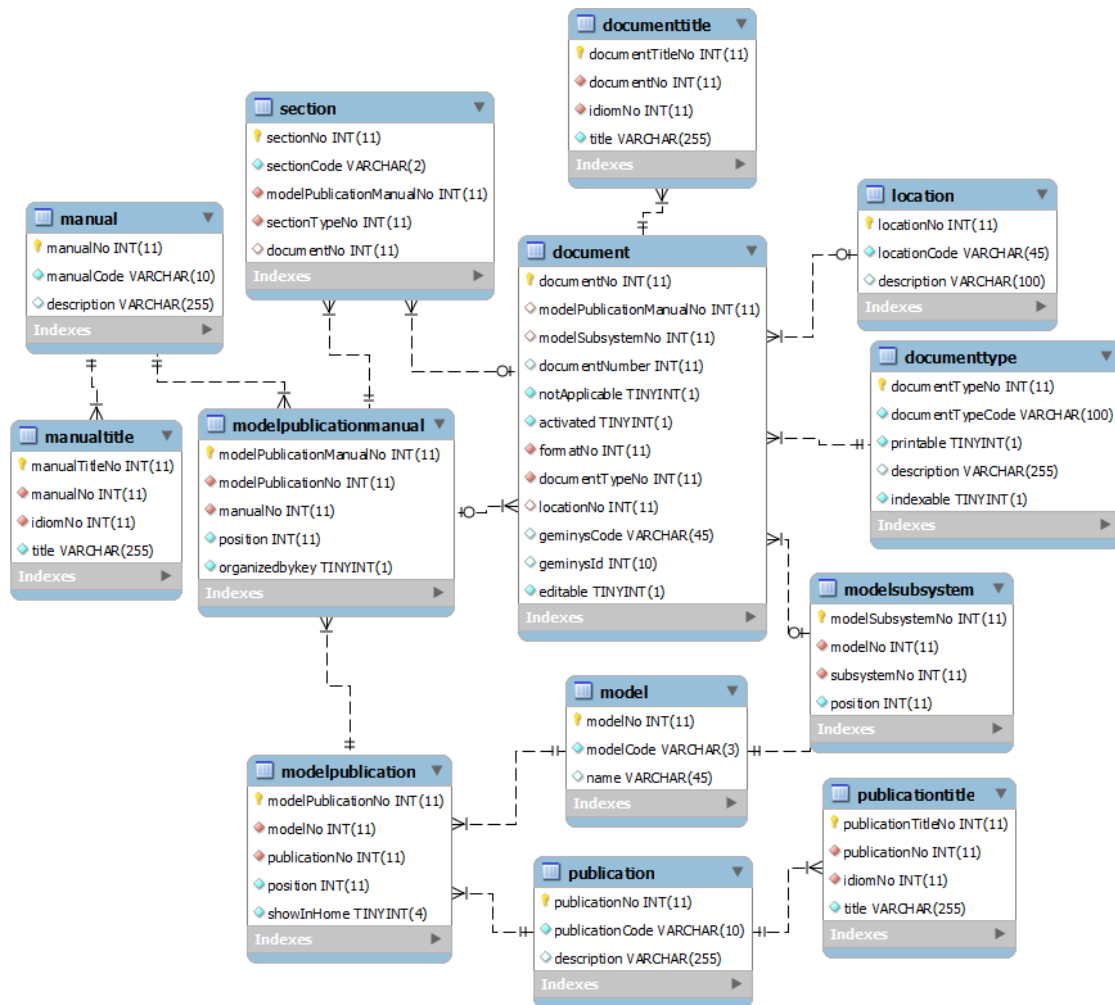




Ilustración 10 – Relación de tablas para documentación técnica

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

6 - DISEÑO DE LA APLICACIÓN

En este capítulo se describen las diferentes capas de la aplicación: capa de presentación, capa de negocio y capa de datos.

6.1 ACTIVIDAD: INICIO DE SESIÓN

Esta actividad es la primera que se presenta al usuario para que se autentique. El usuario debe tener un usuario y una contraseña para poder acceder a la aplicación.

Capa de presentación

La interfaz de esta actividad consta de dos campos de texto editables en los que el usuario introducirá su nombre de usuario y contraseña respectivamente. También tiene un botón (Entrar) con el cual la aplicación autentica la información introducida por el usuario.

Capa de negocio

Para autenticar la información que el usuario ha introducido en los campos de texto hay que: Obtener información introducida (Usuario y contraseña)

En caso de que haya conexión a internet, comprobar contra el servidor si los datos son correctos.

En caso de que no haya conexión a internet, comprobar contra la base de datos local si existe un usuario con esa contraseña.

En ambos casos se recoge en una variable si el usuario es autenticado o no. Si el usuario es autenticado correctamente se crea la sesión y se le da paso a la actividad principal, si no, se le muestra un mensaje de error.

Capa de datos

Para esta operación hace falta una tabla con los datos de usuario y contraseña (Tabla Usuario). Para poder crear la sesión también hace falta la una relación entre usuario y perfil. Para ello se usa otra tabla de perfiles (tabla Perfil) y una tercera con la relación usuario-perfil. Tanto en el servidor como en el cliente tienen que existir estas tablas. Siempre que haya conexión a internet los datos del servidor prevalecen ante los del cliente y además deben sincronizarse.

Serán necesarios los siguientes accesos a datos:

- comprobar(usuario, contraseña)
- obtenerPerfilUsuario(usuario)

Para crear la sesión se puede utilizar las preferencias de Android, donde se almacena el usuario, sus datos personales y el perfil que se ha cargado.

- guardarDatosEnPreferencias(usuario, perfil)

Diagrama de secuencia

En el siguiente diagrama de secuencia se muestra como se realiza el proceso de autenticación del usuario tanto si hay conexión a internet como si no.

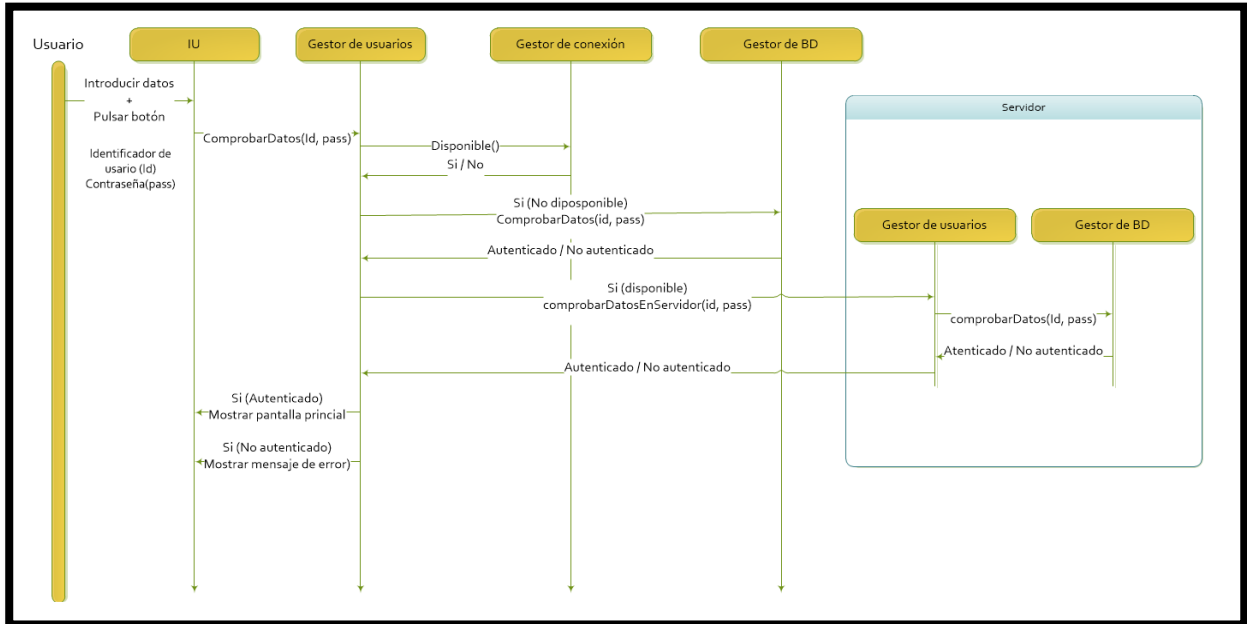


Diagrama de secuencia 1 – Inicio de sesión

6.2 ACTIVIDAD: ACTIVIDAD PRINCIPAL

Esta actividad es el panel de control principal de la aplicación donde se muestran todas las funcionalidades que el usuario tiene.

Capa de presentación

La interfaz de esta actividad depende del perfil que el usuario tenga asignado.

Perfil administrador:



Se le muestran los siguientes accesos: acceso a la documentación técnica, acceso a los grupos de trabajo y acceso a gestión de usuarios. Cada uno de estos accesos se muestra con un botón que contiene una imagen representativa y un título.

Perfil normal:

A este se le muestran los accesos de documentación técnica y grupos de trabajo. Esta interfaz tiene un botón de “cerrar sesión” el cual el usuario puede utilizar para borrar sus datos y dejar el dispositivo liberado para otro.

Capa de negocio

Al ser la actividad principal se le va a pasar como parámetro un objeto usuario con el cual se van a inicializar los datos de las preferencias para tener identificado al usuario en toda la sesión.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

Cada uno de los botones que da acceso a las distintas funcionalidades ejecuta un intento a su respectiva actividad. Para ello hay que obtener los elementos de la interfaz y asociarles un escuchador de click.

Al hacer clic en el botón de “cerrar sesión” hay que borrar los datos de sesión y finalizar todas las actividades que la aplicación haya puesto en marcha y redirigir al usuario a la actividad de inicio de sesión.

Capa de datos

Para inicializar las preferencias del usuario habrá que tener acceso a las preferencias en modo escritura, de manera que hará falta un método para realizar esta operación:

- inicializarPreferencias(user)

Para el caso de cerrar sesión hay que borrar los datos personales del usuario que se han guardado en las preferencias.

- borrarDatosDePreferencias()

Diagrama de secuencia

En el siguiente diagrama de secuencia se presenta la secuencia de pasos que se sigue al acceder a esta actividad.

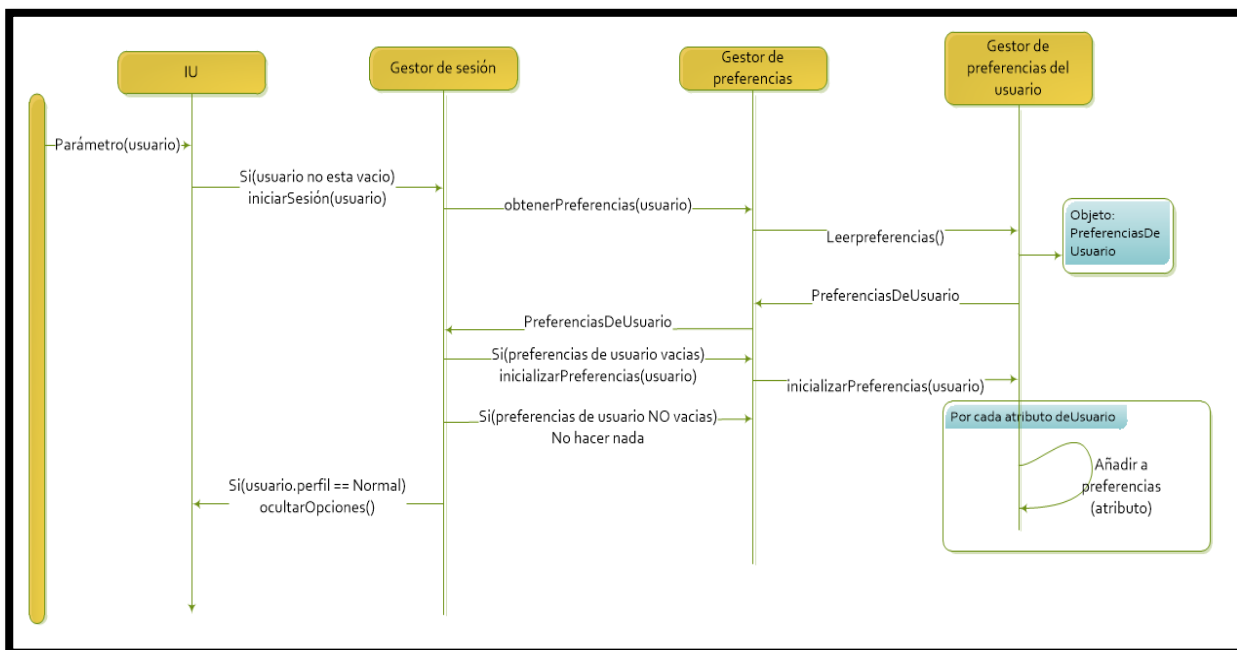


Diagrama de secuencia 2 – Actividad principal

6.3 ACTIVIDAD: PRESENTACIÓN DE DOCUMENTACIÓN TÉCNICA

En esta actividad se presenta toda la documentación de los manuales técnicos a la cual la aplicación va a dar soporte.

Capa de presentación



Dependiendo de la orientación del dispositivo se muestran más o menos elementos en la interfaz, teniendo muchos de ellos en común.

Elementos comunes a ambas orientaciones

En la parte superior de la pantalla se muestran tantas pestañas como publicaciones existan en la base de datos. Cada pestaña representa una publicación¹ con su respectivo título. Al navegar por las pestañas se carga un listado de documentos asociado a cada publicación. Cada elemento de esta lista es un documento.

Orientación horizontal

Se presenta un elemento donde cargar el documento seleccionado de la lista de documentos de la publicación.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

Capa de negocio

Para la representación de las pestañas hace falta obtener y asociar la información de la base de datos al elemento que representa las pestañas. Cada pestaña tiene asociado un número de publicación con el cual se obtiene el título y el listado de documentos asociados para representarlos en la lista. Los documentos se identifican con un número único en la base de datos el cual sirve para poder buscar dicho documento en el sistema de ficheros de Android y cargar su contenido.

Para cargar su contenido se llama a una función que es la responsable de cargar el contenido en la misma actividad o lanzar un intento a la actividad que sea precisa (esto depende del tipo de documento que se quiere visualizar).

Capa de datos

Para conseguir esta representación es necesario disponer de las siguientes tablas en la base de datos: publicación, documentos. También es necesario tener la relación publicación-documento para poder obtener el listado de documentos.

- obtenerPublicaciones(), devuelve el listado de publicaciones que hay en la base de datos.
- obtenerDocumentos(publicación), devuelve el listado de documentos asociado a la publicación que se pasa como parámetro.

Diagrama de secuencia

En el siguiente diagrama de secuencia se muestran los pasos que se siguen para construir dinámicamente la interfaz de esta actividad.

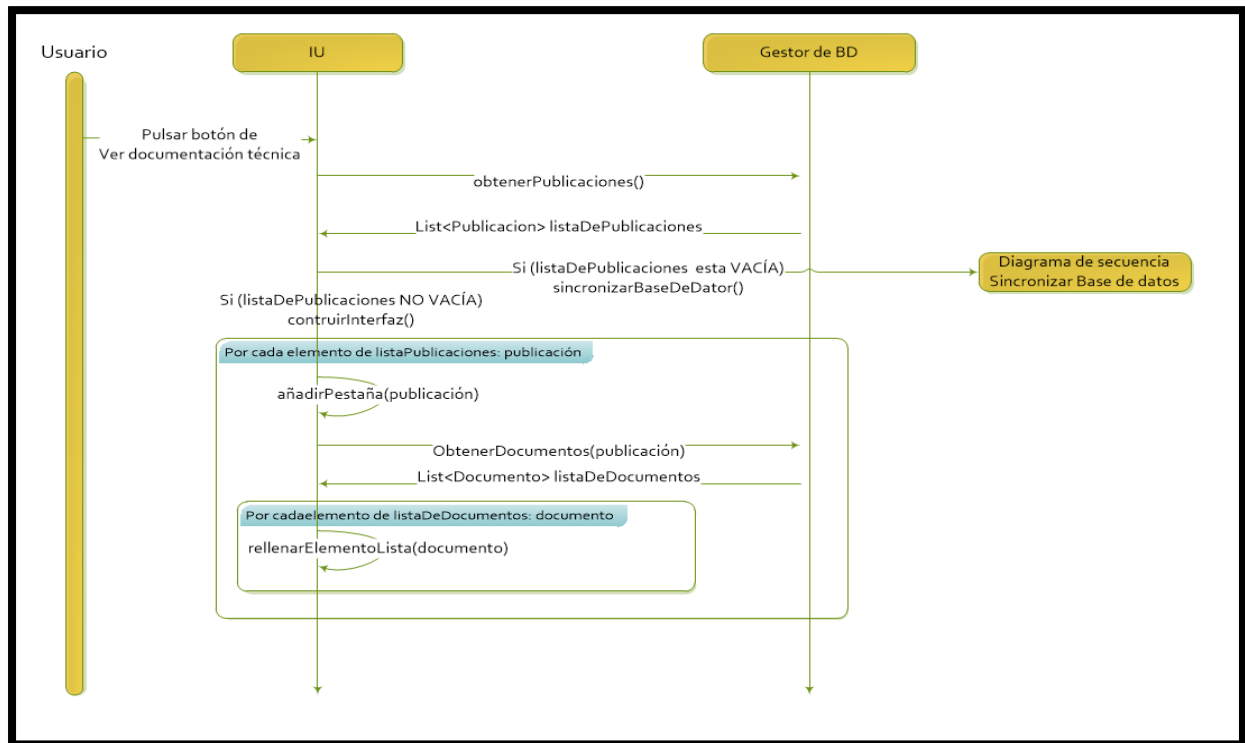


Diagrama de secuencia 3 - Presentación de documentación técnica

6.4 ACTIVIDAD: LISTAR GRUPOS DE TRABAJO

En esta actividad se representan todos los grupos de trabajo que hay en la aplicación.

Capa de presentación



En esta pantalla se presentaran, en formato lista, todos los grupos de trabajo que el usuario actual tenga ya sean creados por el o a los cuales este asignado. Por cada grupo de trabajo se muestra la denominación, el número de trabajadores que lo componen y el número de instrucciones que tiene así como el número de instrucciones completadas.

Dependiendo del perfil del usuario tendrá un botón arriba a la derecha para crear un nuevo grupo de trabajo. Solo en caso de ser administrador se mostrará este botón.

Capa de negocio

Al tratarse de una interfaz dependiente de los datos que hay en la base de datos y que estos datos se pueden ver modificados en cualquier momento, la interfaz hay que crearla dinámicamente.

Se utiliza un elemento lista al cual se cargan los datos leídos de la base de datos. Para representar la cantidad de usuarios y cantidad de instrucciones hay que contar los registros de la base de datos, lo cual se puede hacer con una consulta SQL o bien con un contador e iterar todos los registros que nos devuelva la consulta.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

Cuando el usuario seleccione un elemento de la lista se le lleva a otra actividad donde puede ver el contenido del grupo de trabajo, esta actividad es “actividad de grupo de trabajo”.

Para el botón de crear nuevo grupo de trabajo hay que comprobar el perfil del usuario y en caso de no ser administrador, este botón quedará oculto. En caso de ser administrador el botón se muestra y tendrá la funcionalidad de crear nuevo grupo de trabajo que lleva al usuario a otra actividad.

Capa de datos

Para representar los datos de grupos de trabajo es necesario tener la siguiente información: Nombre del grupo de trabajo, instrucciones asociadas al grupo, usuarios asignados al grupo. Por da cada instrucción también es necesarios saber el estado de la misma, es decir, si se ha realizado o no.

Para ello se utilizan tres tablas: grupos de trabajo, instrucciones y trabajadores, las cuales se relacionan tienen las siguientes relaciones.

En la tabla grupos de trabajo se almacena un identificador único para identificar al grupo junto con la denominación (título) que se le haya puesto.

Grupo de trabajo – instrucciones (1..n)

En la tabla de instrucciones se almacena el identificador de la instrucción, el identificador del grupo de trabajo y un campo para almacenar el identificador de usuario que la ha completado.

Grupo de trabajo – trabajadores(1..n)

En la tabla de trabajadores se almacena el identificador de usuario y el identificador del grupo de trabajo.

Las consultas que se hacen son las siguientes:

- obtenerGruposDeTrabajoPorUsuario(usuario)
- obtenerInstruccionesPorGrupoDeTrabajo(grupoDeTrabajo)
- obtenerTrabajadoresPorGrupoDeTrabajo(grupoDeTrabajo)

Diagrama de secuencia

En el siguiente diagrama de secuencia se pueden ver los pasos que se siguen para conseguir los datos de los grupos de usuario del usuario conectado a la aplicación.

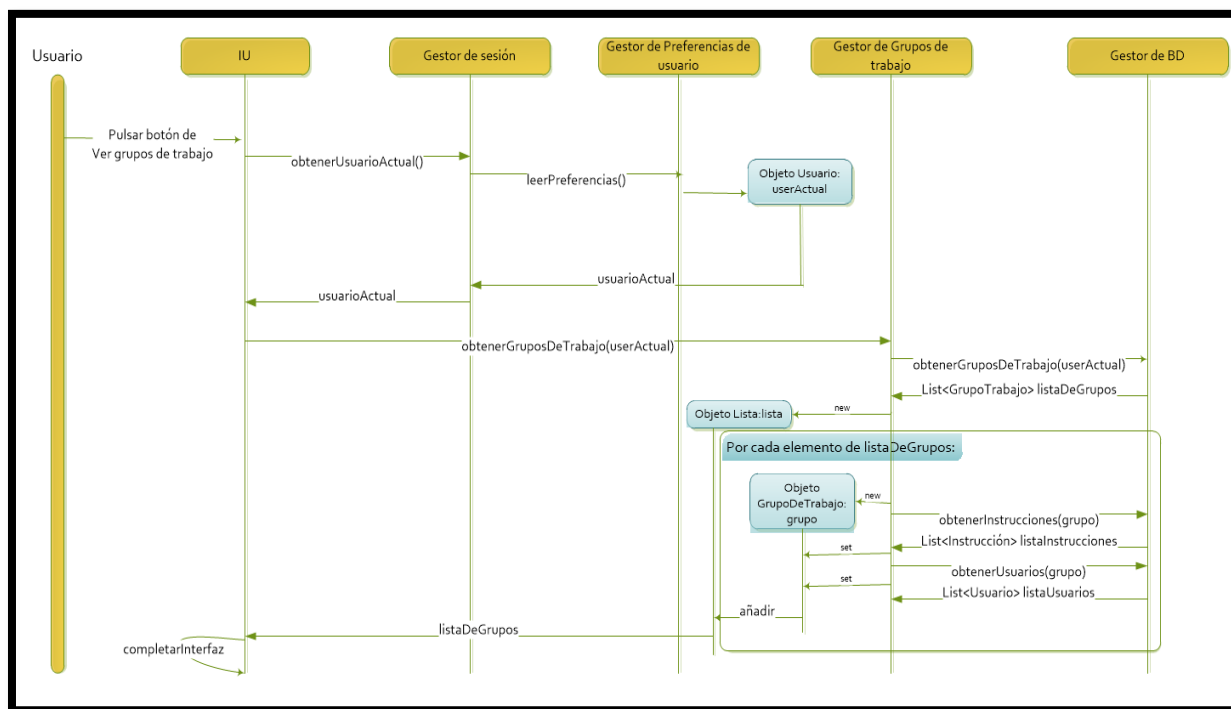


Diagrama de secuencia 4 - Listar grupos de trabajo

6.5 ACTIVIDAD: GRUPO DE TRABAJO

Esta actividad se utiliza para diferentes funcionalidades asociadas a los grupos de trabajo. Se emplea para crear o editar la información asociada a un grupo de trabajo.

Capa de presentación

Esta interfaz se separa en dos fragmentos, uno de ellos para una botonera y otro para presentar la información asociada a cada botón.

Cada uno de los siguientes botones tiene asociado un fragmento, el cual se presenta en pantalla al pulsar sobre él:

Datos generales (fragmento de datos generales*)

Trabajadores (fragmento de trabajadores*)



Instrucciones (fragmento de instrucciones*)

*Las interfaces y lógicas de estos fragmentos se describen en los siguientes apartados.

También hay otros tres botones de acción para guardar, eliminar o publicar el grupo de trabajo.

Capa de negocio

A esta actividad se le pasa un parámetro que representa un grupo de trabajo. En caso de que este parámetro este vacío significa que se está creando un nuevo grupo de trabajo,

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

mientras que si no lo está significa que el grupo de trabajo se está editando y hay que presentar los datos en los campos indicados de los diferentes fragmentos.

La actividad se encarga de gestionar los tres fragmentos que forman parte de la actividad. Estos fragmentos sirven para presentar de forma clara y separada la información que compone un grupo de trabajo.

Al pulsar cada uno de los botones se carga el fragmento asociado en pantalla con los datos del grupo de trabajo.

Las funcionalidades de los botones de acción son las siguientes:

- Guardar

Almacena localmente en la base de datos el grupo de trabajo.

- Eliminar

Elimina de la base de datos local el grupo de trabajo.

- Publicar grupo de trabajo:

Comunica al servidor el nuevo grupo de trabajo para que lance una notificación a todos los dispositivos informando de que un nuevo grupo ha sido creado. (Notificar nuevo grupo de trabajo)

Capa de datos



En esta actividad hacen falta dos funciones para almacenar y eliminar datos de la base de datos local del dispositivo.

- guardarGrupoDeTrabajo(grupoDeTrabajo)

A esta función se le pasa un objeto que representa el grupo de trabajo. Tiene que actualizar las tablas referentes a grupos de trabajo. (Grupos de trabajo, instrucciones y trabajadores)

- eliminarGrupoDeTrabajo(identificador)

A esta función se le pasa el identificador del grupo de trabajo. Tiene que eliminar todos los registros de las tablas referentes a los grupos de trabajo donde el identificador coincida con el que se le pasa.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

6.6 ACTIVIDAD: HOJA DE CONTROL

Esta actividad es la que el trabajador va a ver a la hora de entrar a un grupo de trabajo al cual ha sido asignado.

Capa de presentación

La interfaz que se presenta en esta actividad es toda la información relevante a un grupo de trabajo, concretamente a un grupo de trabajo que ha sido publicado.

Esta interfaz se rellena con los datos del grupo de trabajo que se le pasa como parámetro, los campos que hay que completar en la interfaz son los siguientes:

- Denominación del grupo
- Mensaje global del grupo
- Listado de instrucciones

El listado de instrucciones se presenta con una imagen de una 'x' o con un '☐' dependiendo si dicha instrucción esta completada o no. En caso de que la instrucción este completada también se muestra el nombre del usuario que la completo.

Capa de negocio

A esta actividad se le va a pasar como parámetro el identificador de un grupo de trabajo. Con este identificador se lee de la base de datos la información relativa al grupo de trabajo en cuestión y se representa en la interfaz.

Para la representación de la lista de instrucciones y sus características hay que comprobar el campo "completadoPor", de manera que si este es distinto de vacío se pone el valor del mismo debajo de la instrucción correspondiente y se pone la imagen de "☐" marcándola como completada.

Al seleccionar una instrucción al usuario se le lleva a la actividad de "Visualizar instrucción" pasando como parámetro el identificador del documento que se va a visualizar y una variable indicando que el origen es la "Hoja de control".

Capa de datos

Para la representación de esta actividad es necesario disponer de los siguientes accesos a datos:

- obtenerGrupoDeTrabajo(identificador)

Esta función devuelve un objeto que tiene todos los elementos de un grupo de trabajo (denominación, lista de usuarios y lista de instrucciones)

Diagrama de secuencia

En este apartado se muestran los pasos que se siguen para marcar una instrucción como completada.

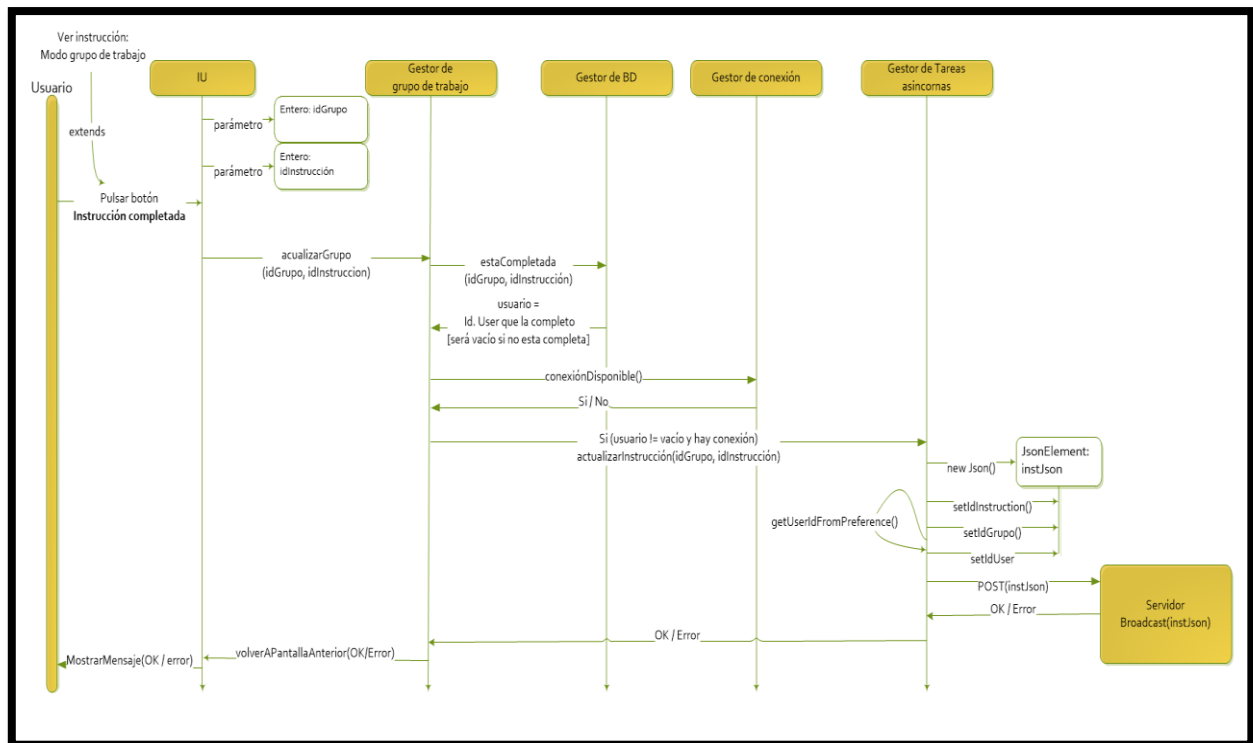


Diagrama de secuencia 5 – Hoja de revisión

6.7 ACTIVIDAD: VISUALIZAR DOCUMENTO

Esta actividad se utilizar para visualizar un documento siempre y cuando el dispositivo se encuentre en orientación vertical, ya que si esta se encuentra en horizontal existirá un espacio donde poder visualizar el documento.

Capa de presentación

Esta interfaz consta solo de un elemento donde se carga el contenido del documento. También se muestra el título de dicho documento en la parte superior de la pantalla.

Capa de negocio

Esta actividad recibe como parámetro el identificador del documento que se quiere visualizar. Con este identificador obtenemos el objeto documento de la base de datos, el cual nos proporciona parte de la información necesaria que hace falta para representar los datos en la pantalla como el título del documento.

También con este identificador se comprueba que dicho documento existe en el sistema de ficheros del dispositivo. En caso de existir se carga su contenido en el elemento indicado para ello, mientras que si dicho documento no existe en el sistema de ficheros se hace una petición al servidor para descargar su contenido y almacenarlo en el dispositivo para tener disponible sin conexión. El proceso de descargar el contenido del documento y almacenarlo esta explicado en el anexo (sincronizar documento).

Capa de datos

Para esta actividad es necesario acceder a la base de datos para obtener el título del documento y para saber de qué tipo de documento se trata, para ello se utiliza la siguiente función.

- obtenerDocumento(identificador)

También se accede al sistema de ficheros para leer el contenido del documento o para almacenar el contenido descargado. Para esto se utiliza una clase que gestiona todos los accesos al sistema de ficheros que tiene entre sus métodos los siguientes:

- almacenarDocumento(identificador, contenido)

Esta función crea un documento con el contenido pasado como parámetro y el identificador como nombre del fichero añadiéndole la extensión “.HTML”.

- leerDocumento(identificador)

Esta función devuelve el contenido del documento en texto plano. El texto que devuelve esta función es el que se carga en la actividad para visualizar el documento.

Diagrama de secuencia

En el siguiente diagrama de secuencia se representan todos los posibles caminos que se pueden dar al visualizar un documento. También están representados los pasos que se dan para sincronizar un documento.

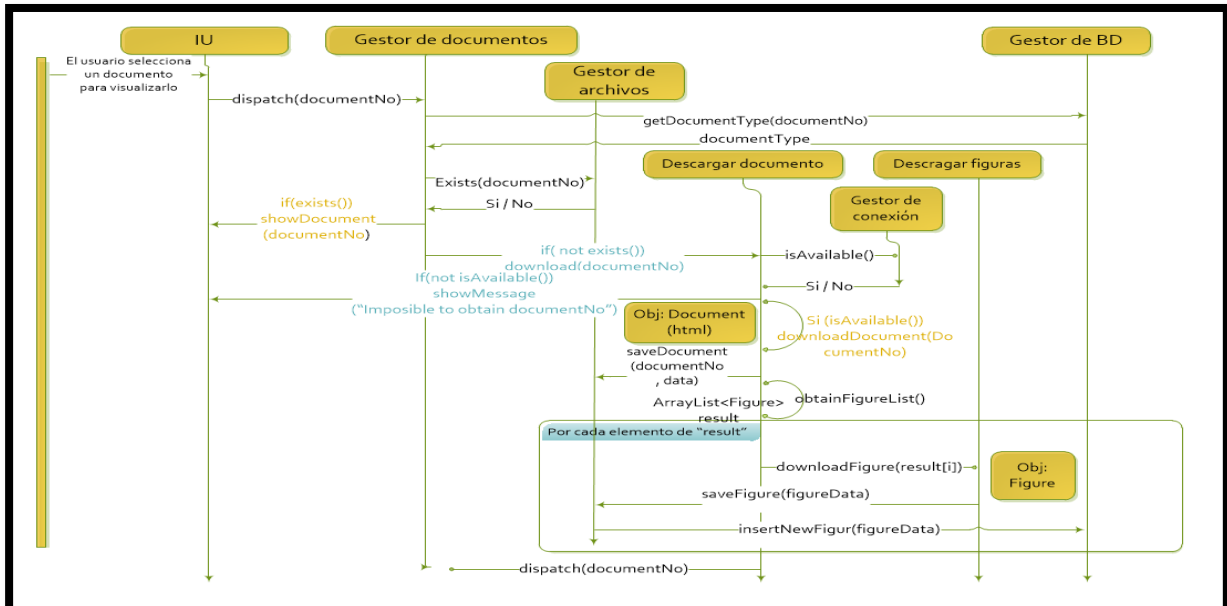




Diagrama de secuencia 6 - Visualizar documento

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

6.8 ACTIVIDAD: VISUALIZAR INSTRUCCIÓN

Esta actividad es la encargada de visualizar en pantalla el contenido de una instrucción y sus metadatos. La interfaz de esta actividad se reutiliza tanto para la visualización de una instrucción desde la documentación técnica así como desde los grupos de trabajo.

Capa de presentación

La interfaz de esta actividad contiene diferentes elementos los cuales hay que ocultar dependiendo de si la aplicación llama a esta actividad desde un grupo de trabajo o no. Aquí se describen todos los elementos y en la Capa de negocio se explica cómo se gestionan. La interfaz de esta actividad consta de dos pantallas (fragmentos) diferentes:

Ver instrucción

En esta pantalla se presenta el contenido de la instrucción. Para ello se utiliza un elemento de Android que sirve para visualizar contenido HTML. En la parte superior izquierda de la pantalla se muestra el título de la instrucción y a la derecha un botón con el que el usuario podrá marcar dicha instrucción como completada si se trata de un grupo de trabajo.

Ver metadatos de instrucción

Estos metadatos se presentan en una lista con su nombre, referencia y descripción. Para acceder a esta pantalla el usuario tiene que deslizar la pantalla hacia la izquierda.

Capa de negocio

A esta actividad se le pasan dos parámetros que indican el identificador de la instrucción que se quiere ver y el segundo que indica si se viene desde un grupo de trabajo o no.

Con el identificador de la instrucción hay que comprobar en el sistema de ficheros si dicho documento existe y de ser así cargar el contenido en la interfaz, en caso de no existir hay que descargarlo del servidor y almacenarlo para poder tenerlo disponible sin conexión una vez descargado se muestra en la interfaz.

Para la representación de los metadatos se accede a la base de datos local para leer todos los datos relacionados con dicha instrucción de donde se obtiene un listado de objetos que se pasarán a una lista para su representación.

Capa de datos

Para esta actividad se necesita tener la relación de instrucción-metadatos, para lo cual se necesitan las tablas instrucción y metadatos, la relación entre ellas es el identificador de la instrucción.



- obtenerMetadatos(identificador)

6.9 ACTIVIDAD: PLAN DE MANTENIMIENTO

Esta actividad representa un listado de instrucciones a las cuales se les puede aplicar un filtro para hacer una búsqueda rápida.

Capa de presentación

La interfaz de esta actividad consta de un listado de instrucciones que están asociadas al plan de mantenimiento. Algo que es esencial en el plan de mantenimiento es saber la frecuencia con la que hay que realizar una instrucción por lo que se muestra dicha información junto al título de cada instrucción.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

También hay disponible un botón con el que se puede filtrar por frecuencia al igual que hay un campo de texto que filtra por el título de la instrucción. Al filtrar solo se muestran las instrucciones que pasan el filtro.

Capa de negocio

Para representar el listado de las instrucciones hay que acceder a la base de datos y obtener todos los datos necesarios para la representación de dicha lista. Esta información se le pasa a un elemento lista de Android para que lo muestre correctamente. Al seleccionar cada instrucción se le envía al usuario a la actividad de “Visualizar instrucción”.

Para el filtrado de las instrucciones hay que recorrer el listado completo y comprobar si dicha instrucción cumple con la condición del filtro. En caso de ser filtro por título se comprueba si existen esos caracteres en el título de la instrucción y en caso del filtro por frecuencia se comprueba si dicha instrucción tiene asociada la frecuencia seleccionada.

Capa de datos

Para esto se utiliza una tabla específica (plan de mantenimiento) donde se almacenan los identificadores de las instrucciones que componen el plan de mantenimiento y sus características como puede ser la frecuencia de la instrucción.

- obtenerPlanDeMantenimiento()

6.10 FRAGMENTO: DATOS GENERALES DEL GRUPO DE TRABAJO

Este fragmento se emplea para representar la interfaz de la introducción de datos generales en la edición de los grupos de trabajo. Este fragmento está asociado con la actividad “grupos de trabajo”.

Capa de presentación

La interfaz de este fragmento consta de dos campos de texto para introducir el nombre del grupo de trabajo y el mensaje global que llegara a todos los usuarios asignados al grupo. También tiene un botón con el que guardar dicha información.



Capa de negocio

Esta interfaz se va a reutilizar tanto para editar como para crear nuevos grupos de trabajo por lo que se le pasa un parámetro que puede o no estar vacío. Si dicho parámetro tiene información, esta se carga en los campos correspondientes.

Al pulsar sobre el botón de guardar se leen los campos de texto y se guarda la información en una variable para cuando se guarde el objeto grupo de trabajo esta información se almacene en la base de datos local.

Capa de datos

No es necesario.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

6.11 FRAGMENTO: TRABAJADORES DEL GRUPO DE TRABAJO

Este fragmento se emplea para representar la interfaz de asignación de trabajadores en un grupo de trabajo. Este fragmento está asociado con la actividad “grupos de trabajo”.

Capa de presentación

La interfaz de este fragmento consta de dos listas en las que se representan por una parte la lista de usuarios asignados al grupo de trabajo y por otra el listado completo de usuarios registrados en la aplicación. También hay un botón para guardar la lista de usuarios seleccionados.

Capa de negocio

Las dos listas mencionadas en la Capa de presentación se comunican entre ellas. Al seleccionar un elemento de la lista completa de usuarios, dicho elemento se pasa a la lista de usuarios actuales del grupo de trabajo si este no está ya asignado al mismo. En caso de seleccionar un elemento de la lista de usuarios actuales del grupo, este se elimina de dicha lista.

Al pulsar sobre el botón de guardar se guarda en una variable temporal la lista de usuarios que han sido seleccionados, de manera que quedan asignados al grupo de trabajo en cuestión.

Capa de datos

Se necesita acceder a la base de datos local para representar todos los usuarios que existen en el sistema.

- obtenerUsuarios()

6.12 FRAGMENTO: INSTRUCCIONES DEL GRUPO DE TRABAJO

Este fragmento se emplea para representar la interfaz de asignación de instrucciones en un grupo de trabajo. Este fragmento está asociado con la actividad “grupos de trabajo”.

Capa de presentación

En la interfaz de este fragmento se representan dos listas que representan la lista actual de instrucciones del grupo de trabajo y la otra representa la lista completa de instrucciones que hay disponible en el sistema para asignar a los grupos de trabajo. También hay un botón para guardar el listado de instrucciones seleccionadas.

Capa de negocio

Este fragmento debe leer de la base de datos local el listado de instrucciones disponibles para asignar a un grupo de trabajo. Este listado se representara en un elemento lista y cuando se seleccione un elemento de la misma pasa a formar parte de la lista de instrucciones del grupo de trabajo en cuestión.

Si se realiza una pulsación larga sobre un elemento de la lista completa de instrucciones se le envía al usuario a la actividad de “Visualizar instrucción” para que pueda ver el contenido y verificar que es la instrucción que está buscando.

En caso de seleccionar un elemento de la lista de instrucciones actual se elimina dicho elemento de la lista.

Al pulsar sobre el botón guardar se guarda en una variable temporal el listado de instrucciones que se ha seleccionado.

Capa de datos

Este fragmento necesita acceder a la base de datos local para obtener el listado completo de instrucciones.

- obtenerInstruccionesDisponibles()

6.13 REGISTRAR DISPOSITIVO CON GCM

Para poder mandar notificaciones a los diferentes dispositivos es necesario realizar un proceso al iniciar la aplicación, el cual se explica a continuación.

En la parte del cliente hay que generar un identificador único que sea identificador del hardware que se está utilizando, es decir, un identificador único pero que sea siempre el mismo cada vez que se consulte, para ello se leerá el identificador de hardware del dispositivo utilizando las librerías de Android para poder acceder a las propiedades del hardware.

El identificador de hardware sirve para asociar el identificador que se obtiene desde el servicio de Google con el dispositivo en cuestión. Una vez obtenidos ambos identificadores se envían al servidor de aplicación donde se guardan ambos para poder tener identificados y poder realizar las comunicaciones correctamente. En la ilustración 11 se puede ver el flujo de los eventos y la información que se intercambien entre el cliente Android, el servidor de aplicaciones y el servicio de Google para realizar el proceso de registro, también se representa el envío de un mensaje.

Para este proceso es necesario que en la base de datos del servidor exista una tabla donde almacenar los identificadores para poder realizar correctamente las comunicaciones.

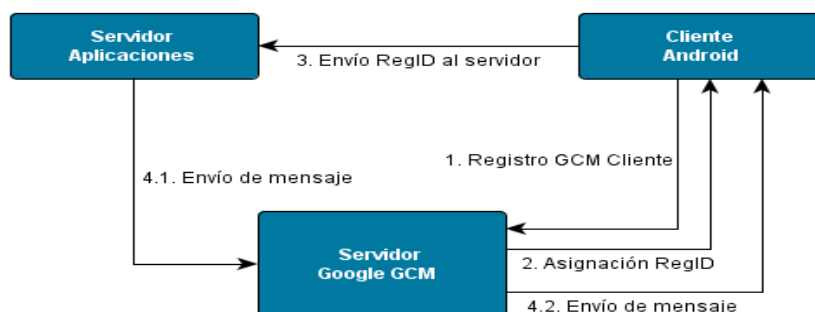






Ilustración 11 – Registro GCM

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

6.14 COPIA INICIAL DE DATOS AL DISPOSITIVO

Dado que existen datos que son fijos en el proyecto y no se editan en la aplicación web existente se ha decidido meterlos como datos en la aplicación móvil. Estos datos son los CSS, PDF, imágenes y demás material a los que se referencia desde el código HTML

Para que estos datos puedan ser referenciados en el dispositivo dentro de los WebView que se van a utilizar para visualizar los documentos, es necesario que se encuentren todos bajo una misma ruta dentro del sistema de ficheros. Por ello lo primero que se hace al iniciar la aplicación, es copiar dichos ficheros a la memoria del dispositivo bajo el directorio donde se van a guardar los documentos que se van a sincronizar.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

7 - IMPLEMENTACIÓN

En los siguientes apartados se explica cómo se ha realizado la implementación tanto de la interfaz visual como de la lógica de negocio. Se explican las partes de programación que más relación tienen con las aplicaciones Android.

7.1 PERMISOS

En los proyectos Android existe un elemento (Manifiesto.xml) donde se declaran todos los elementos que se utilizan en el proyecto como pueden ser actividades, servicios, escuchadores de broadcast. Todos estos elementos puede ser que quieran utilizar recursos hardware del dispositivo donde se están ejecutando, para que esto sea posible la aplicación tiene que tener unos permisos definidos en el manifiesto. Para definir un permiso en el manifiesto hay que declarar un elemento xml de la siguiente forma:

```
<uses-permission android:name="Nombre.del.permiso" />
```

A continuación se describen los diferentes tipos de permisos que se utilizan en este proyecto, indicando el nombre del permiso y porque es necesario:

7.1.1 Permisos de Android:

Nombre: android.permission.INTERNET

Motivo: Acceso a la red inalámbrica y a internet.

Declaración:

```
<uses-permission android:name="android.permission.INTERNET" />
```

Nombre: android.permission.GET_ACCOUNTS

Motivo:

Declaración:

```
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
```

Nombre: android.permission.WAKE_LOCK

Motivo:

Declaración:

```
<uses-permission android:name="android.permission.WAKE_LOCK" />
```

Nombre: android.permission.ACCESS_NETWORK_STATE

Motivo:

Declaración:



```
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Nombre: android.permission.WRITE_EXTERNAL_STORAGE

Motivo:

Declaración:

```
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

Nombre: android.permission.READ_EXTERNAL_STORAGE

Motivo:

Declaración:

```
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

7.1.2 Permisos externos de Google:

Nombre: com.google.android.c2dm.permission.RECEIVE

Motivo:

Declaración:

```
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />
```

7.1.3 Permisos propios:

Nombre: com.geminys.metrocentrotablet.permission.C2D_MESSAGE

Motivo:

Declaración:

```
<uses-permission
android:name="com.geminys.metrocentrotablet.permission.C2D_MESSAGE"
/>
```

Una vez que tenemos los permisos adecuados para realizar las tareas que se desea, en el manifiesto también hay que declarar cada actividad, servicio o escuchador de broadcast que se utilice en la aplicación ya que sin esta declaración la aplicación no funcionaría. A continuación se describen los elementos que son necesarios en este proyecto:

7.2 ACTIVIDADES



Para la declaración de una actividad hay que declarar un elemento xml de la siguiente forma.

```
<activity
android:name="Nombre de la clase que implementa la actividad"
android:label="titulo">
</activity>
```

Cada elemento <activity> debe tener en el atributo "name" el nombre completo de la clase a la que hace referencia, es decir, el nombre de la clase precedido de todos los paquetes. Como ejemplo se muestra la declaración de la actividad principal del proyecto, la cual se encarga de inicializar los datos pero no muestra ninguna interfaz al usuario.

```
<activity
android:name="com.geminys.metrocentrotablet.activities.MainActivity"
android:label="@string/app_name" >
<intent-filter>
<action android:name="android.intent.action.MAIN" />

<category
android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
```

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

Como en cualquier aplicación siempre tiene que haber una tarea que sea la principal, la que ponga en marcha el sistema, para indicar este comportamiento se utiliza en elemento `<intent-filter>` con los hijos `<action>` y `<category>`. El elemento `<action>` tiene el atributo "name" que indica el nombre de la acción que se realiza, en este caso en la acción principal que sería lanzar la aplicación. Cuando se detecte el "Intent" de lanzar la aplicación pasará por el filtro de esta actividad y al estar en el filtro ejecutara dicha actividad.

7.3 VISUALIZACIÓN DEL CONTENIDO DE LOS DOCUMENTOS

Los documentos están almacenados en el servidor en formato XML y se les aplica unas hojas de estilo XSL para darles el formato deseado para visualizarlos en el navegador. Estos documentos contienen links e imágenes los cuales se han de tratar de una forma especial al darles funcionalidad en la aplicación móvil.

Los documentos XML y XSL se van a reutilizar, con alguna modificación, para obtener el HTML final que se visualizara dentro de un elemento WebView que nos proporciona el sistema de Android. Un WebView es un elemento que se comporta como un navegador web en el que se puede cargar el contenido HTML.

El contenido HTML que se obtiene del servidor referencia a hojas de estilo (CSS) las cuales deberán existir previamente en una carpeta "css". Esta carpeta deberá estar donde estén alojados los documentos (HTML) en la memoria del dispositivo para que se apliquen todos los estilos y se visualice correctamente el documento.

También en el HTML hay referencias a imágenes las cuales deberán estar en el dispositivo antes de cargar el contenido HTML en el WebView.

Tanto para referenciar los archivos ".css" como las imágenes se van a utilizar rutas relativas. Para que ambas tengan la misma ruta relativa se creara la siguiente estructura de carpetas:

- Raíz
 - Documentos (documentos con contenido HTML)
 - Css
 - Imágenes

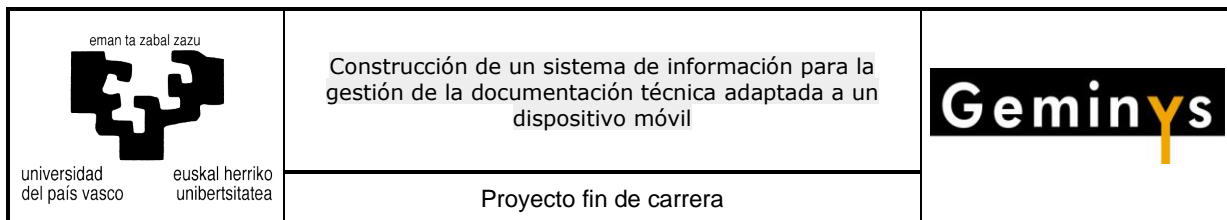
Con esta estructura logramos cargar un documento que tenga referenciadas las imágenes y las plantillas CSS de la siguiente manera:

```

<stylesheet src="../Css/nombre.css" ../>
```

Para conseguir que el código HTML que devuelve el servidor tenga estas características se ha añadido un parámetro en el proceso de transformación de XML a HTML el cual se explica en el apartado de "Generación de código HTML de documento".

A continuación se describe la clase Java que se utiliza para cargar un documento en pantalla. Para esto se ha creado la clase "DocumentViewerManager" donde se implementan todos los métodos necesarios para visualizar los documentos. Esta clase necesariamente va a acceder a memoria principal y también es necesario que acceda a internet para obtener la





información del documento, por lo cual será necesario pasarle al constructor la referencia de la actividad que lo va a utilizar.

El código del constructor quedaría de la siguiente manera:

```
public DocumentViewerManager(Activity mActivity) {
    this.mActivity = mActivity;
}
```

Para cargar un documento es necesario saber el “documentNo” que es el identificador único de cada documento, una vez obtenido el “documentNo” y creada la instancia del “DocumentViewerManager” se debe realizar una llamada al siguiente método, el cual se explica a continuación:

```
dispatchView(Integer documentNo)
public void dispatchView(Integer documentNo) {
    Integer documentType = ApplicationSQLiteHelper.
        getInstance(mActivity.getApplicationContext())
            .getDocumentType(documentNo);
    WebView documentWebView =
        (WebView)mActivity.findViewById(R.id.documentWebView);
    switch (documentType) {
        case DocumentType.MAINTENANCEPLAN: {
            Intent intent = new Intent(mActivity,
                MaintenancePlanActivity.class);
            mActivity.startActivity(intent);
            break;
        }
        case DocumentType.INSTRUCTION: {
            if(documentWebView != null) {
                loadDocumentOnWebView(documentWebView, documentNo);
            } else {
                intent = new Intent(mActivity,
                    InstructionDocumentActivity.class);
                intent.putExtra(InstructionDocumentActivity.INTENT_DOC_NO,
                    documentNo);
                mActivity.startActivity(intent);
            }
            break;
        }
        default: {
            if(documentWebView != null) {
                loadDocumentOnWebView(documentWebView, documentNo);
            } else {
                intent = new Intent(mActivity,
                    NormalDocumentActivity.class);
                intent.putExtra(NormalDocumentActivity.INTENT_DOC_NO, documentNo);
                mActivity.startActivity(intent);
            }
            break;
        }
    }
}
```

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

Breve explicación del método:

Debido a que existen varios tipos de documentos, lo primero que se hace es detectar el tipo de documento que se quiere visualizar, para ello se lee de la base de datos local el tipo de documento que es el “documentNo” se pasa como parámetro.

```
Integer documentType = ApplicationSQLiteHelper.getInstance
(mActivity.getApplicationContext()).getDocumentType(documentNo);
```

A continuación con un “switch case:” se realizan diferentes acciones dependiendo del tipo de documento “documentType” del que se trate. Los tres tipos de documentos y sus acciones se explican a continuación.



```
case DocumentType.MAINTENANCEPLAN:{
    Intent intent = new Intent(mActivity,
MaintenancePlanActivity.class);
    mActivity.startActivity(intent);
    break;
}
```

Este tipo de documento hace referencia al plan de mantenimiento el cual se genera automáticamente a partir de los registros de la base de datos, entonces lo que se hace cuando se quiere ver este documento es lanzar una nueva actividad que se llama “MaintenancePlanActivity”, dejando en segundo plano a la actividad que ha solicitado visualizar dicho documento. La actividad de “MaintenancePlanActivity” se explica más adelante.

```
case DocumentType.INSTRUCTION:{
    intent = new Intent(mActivity,
InstructionDocumentActivity.class);
intent.putExtra(InstructionDocumentActivity.INTENT_DOC_NO,
documentNo);
    mActivity.startActivity(intent);
    break;
}
```

El tipo de documento INSTRUCTION hace referencia a una instrucción. Este tipo de documentos también requiere de una nueva actividad para visualizarlos ya que tiene unas propiedades especiales. En este caso para que la nueva actividad que va a visualizar la instrucción sea capaz de saber qué documento se desea ver, hay que pasarle como parámetro el identificador del documento “documentNo”. La actividad que se encarga de visualizar este tipo de documentos es “InstructionDocumentActivity” la cual se explica más adelante.

```
default:{
    if(documentWebView != null){
        loadDocumentOnWebView(documentWebView, documentNo);
    }else{
        intent = new Intent(mActivity,
NormalDocumentActivity.class);
```

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

```
intent.putExtra (NormalDocumentActivity.INTENT_DOC_NO, documentNo);
    mActivity.startActivity(intent);
}
break;
}
```

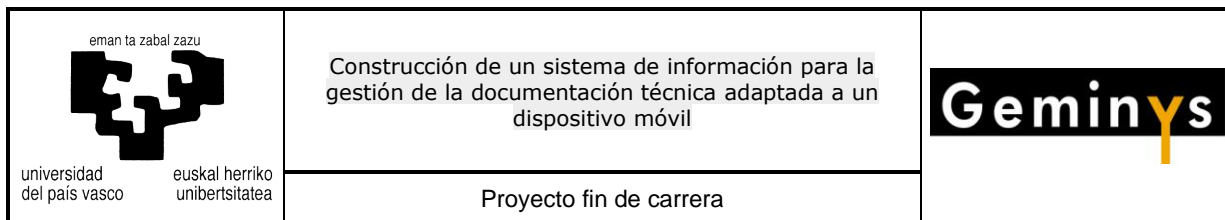
Por último tenemos el caso por defecto, este se ejecutara en caso de que el documento seleccionado no sea ni plan de mantenimiento ni instrucción. Este tipo de documentos son los que simplemente hay que cargarlos en un WebView como se mencionaba antes. En este paso comprobamos si en la interfaz del usuario existe un WebView donde se pueda cargar el documento, de no ser así se lanza una nueva actividad "NormalDocumentActivity" a la cual se le pasar el identificador del documento como parámetro.

En caso de existir un WebView en la interfaz, esto significará que el dispositivo está en posición horizontal, habrá que cargar el contenido del documento en el WebView, para ello se llama a la siguiente función pasándole como parámetros el WebView donde queremos que cargue el contenido y el identificador del documento que se quiere visualizar:

- loadDocumentOnWebView(documentWebView, documentNo)

Esta función se utiliza en varias partes del código ya que se encarga de buscar el contenido del documento en la memoria local del dispositivo y si no lo encuentra se encarga de descargar dicho contenido y almacenarlo para tenerlo disponible cuando no exista conexión. A continuación se muestra el código y una breve descripción de la función:

```
@SuppressWarnings("SetJavaScriptEnabled")
public void loadDocumentOnWebView(WebView wv, Integer documentNo) {
    String data = "";
    if (FileManager.getInstance().existDocument(documentNo.toString()))
    {
        //Load the document on the webview
        if (wv != null) {
            try {
                data = FileManager.getInstance()
                    .readDocument(documentNo.toString() + ".html", null);
                String baseUrl = FileManager.getInstance()
                    .getDocumentsUrlContext();
                wv.loadDataWithBaseUrl(baseUrl, data, "text/HTML", "UTF-8", null);
            } catch (Exception e) {
                Toast.makeText(mActivity, "Imposible visualizar documento.", Toast.LENGTH_SHORT).show();
            }
        }
    }
    } else {
        progressDialog = ProgressDialog.show(mActivity,
        "Espere por favor", "Descargando documento...");
        //Download the document and load it on the webview.
        Integer[] docNo = {documentNo};
```

```

        SynchronizeDocumnetTask synDocTask = new
SynchronizeDocumnetTask(mActivity);
        synDocTask.execute(docNo);
    }
}

```

Descripción del código:

Lo primero que hace esta función es comprobar que el documento que se le pasa como parámetro exista en el sistema de ficheros del dispositivo:

```

FileManager.getInstance().existDocument(documentNo.toString())

```

En caso de que el documento ya exista se muestra en un en el WebView que se le pasa, para ello se ejecuta la siguiente línea de código:

```

wv.loadDataWithBaseURL(baseUrl, data, "text/HTML", "UTF-8", null)

```

El método “loadDataWithBaseURL(baseUrl, data, formatType, encoding,null)” es un método propio del objeto WebView. Lo que hace es cargar en el objeto el contenido que se le pasa en “data” con el formato que se le especifica en “formatType” y la codificación que se indica en “encoding”. Mientras que el parámetro baseUrl se utiliza para indicar donde tiene el origen de los datos que se referencias en el HTML.

Como previamente se había comentado, las referencias a los datos que necesitan los documentos HTML van a estar en la misma carpeta que los documentos, por lo que se le indicara como raíz dicha dirección:

```

String baseUrl = FileManager.getInstance().getDocumentsUrlContext()

```



Para obtener los datos de los documentos, es decir, para obtener el String que representa el código HTML que se desea cargar en el WebView se realiza la siguiente operación:

```

data =
FileManager.getInstance().readDocument(documentNo.toString()+".html"
, null);

```

Todo esto en caso de que el fichero ya se encuentre en el dispositivo, en caso de no ser así, dicho documento deberá ser descargado, almacenado y visualizado. Para ello se lanza una tarea asíncrona “SynchronizeDocumnetTask” la cual se explica en el apartado “Obtencion de documentos”.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

Obtención de documentos

En este apartado se explica el código que se ha implementado para solicitar, obtener, almacenar y visualizar un documento del servidor.

Para la realización de esta operación se han utilizado las tareas asíncronas que nos proporciona Android, se ha creado una clase que extiende de `AsyncTask<>`, “`SynchronizeDocumnetTask`”. Al extender de la clase `AsyncTask`, tenemos disponibles los métodos:

```
protected Integer doInBackground(Integer... documentNos)
protected void onPostExecute(Integer documentNo)
```

los cuales se sobreescriben. La descarga del documento se hace en el método `doInBackground()`, cuando dicha descarga se completa se llama automáticamente al método `onPostExecute()`. A continuación se describe las variables y funciones que utiliza la clase “`SynchronizeDocumnetTask`”:

```
public class SynchronizeDocumnetTask extends AsyncTask<E,E,E> {

    public Activity delegate = null;
    public ArrayList<String> errors;



    public SynchronizeDocumnetTask(Activity listener){
        delegate = listener;
        errors = new ArrayList<>();
    }
    @Override
    protected Integer doInBackground(Integer... documentNos) { . .
    .}
    @Override
    protected void onPostExecute(Integer documentNo){. . .}
}
```

Al instanciar el objeto “`SynchronizeDocumnetTask`” se le pasa al constructor de la clase la actividad que lo está instanciando. Esta actividad será sobre la que se *delegan* las tareas que tengan alguna interacción con la interfaz gráfica

Esta clase la utiliza el “`DocumentViewerManager`” de la siguiente manera en la función “`loadDocumentOnWebView`”.

```
SynchronizeDocumnetTask synDocTask =
    new SynchronizeDocumnetTask(mActivity);
synDocTask.execute(documentNo);
```

Primero se crea la instancia de la clase pasándole la actividad como parámetro y después se llama al método “`execute`” pasándole el identificador del documento como parámetro. Este método llama a la función “`doInBackground()`” que se ha sobrescrito previamente.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

7.4 OBTENCIÓN DE IMÁGENES DE DOCUMENTO

Los documentos que se van a visualizar en la aplicación contienen imágenes las cuales no se descargan junto con el código HTML, por lo que hay que crear un proceso paralelo que descargue todas las imágenes asociadas al mismo.

Lo primero que se debe hacer es conocer el listado de imágenes que un documento tiene asociadas, para ello en el servidor existe un método que pasándole el identificador del documento devuelve un JSON con el listado de los nombres de las imágenes que ese documento tiene asociados:

Para esta operación se han utilizado métodos que ya estaban implementados en la aplicación web, por lo que el código quedaría de la siguiente forma:

```
BasicGeminysDocument documentReq =
DocviewerManager.getInstance().getBasicGeminysDocument(documentNo,
model, idiom, false, true, false);
List<InfoFigure> figureList = documentReq.getFigures(true);
String infoFigureListJson = new
JSONSerializer().exclude("*.class").serialize("infoFigures",
figureList);
```

Podemos ver que una vez obtenida la lista de figuras (figureList) que contiene el documento, lo que se hace es convertir esa lista de figuras a formato JSON, para ello se emplea la librería de libre distribución flexjson, la cual nos permite convertir objetos a formato JSON de forma muy sencilla.

Una vez que ya se tiene el String que representa el listado de figuras del documento, hay que enviárselo al cliente que lo solicito, para ello se utilizan las siguientes líneas de código:

```
PrintWriter out = response.getWriter();
response.setContentType("application/json");
response.setHeader("cache-control", "no-cache");
response.getWriter().write(infoFigureListJson);
return null;
```

Lo que conseguimos es crear una respuesta indicando que el formato de la información que se envía es JSON, con los datos de las figuras previamente obtenidas. Hasta aquí la parte de obtención el listado de imágenes, esta información llegara al dispositivo el cual tiene que recorrer el Array de JSON y por cada elemento deberá hacer una petición al servidor con el nombre de la imagen, cada una de estas peticiones se hace en un hilo diferente, por lo que se lanzaran tantos hilos como imágenes tenga el documento, en Android se utiliza la clase AsyncTask para realizar las tareas en segundo plano:

Código:

Este código es el responsable de obtener el JSON de las imágenes del documento y realizar las peticiones al servidor por cada elemento del JSON.

```

@Override
protected Integer doInBackground(Integer... documentNo) {
try {
    url = new URL(feed);
    // Create a new HTTP URL connection
    URLConnection connection = url.openConnection();
    HttpURLConnection httpConnection =
        (HttpURLConnection) connection;
    int responseCode = httpConnection.getResponseCode();
    if (responseCode == HttpURLConnection.HTTP_OK) {
        InputStream in = httpConnection.getInputStream();
        try {
            procesarDatos(in);
        } catch (JSONException e) {
            errors.add(e.getMessage());
        }
    }
} catch (MalformedURLException e1) {
    errors.add(e1.getMessage());
} catch (IOException e1) {
    errors.add(e1.getMessage());
}
return null;
}

```

Abrir conexión
con el servidor

Recoger respuesta

Como se puede observar este código lo único que hace es hacer una petición al servidor y pasarle a una función “procesarDatos()” los datos que le envía el servidor, esta función es la responsable de analizar los datos que le han llegado y lanzar una tarea asíncrona por cada imagen. El código de esto se muestra a continuación.

```

public void procesarDatos(InputStream in) throws JSONException,
IOException{

```

Lectura de
la
respuesta
del

```

    BufferedReader streamReader = new BufferedReader(new
    InputStreamReader(in));
    StringBuilder responseStrBuilder = new StringBuilder();
    String inputStr;
    while ((inputStr = streamReader.readLine()) != null) {
        responseStrBuilder.append(inputStr);
    }



```

Conversión
de JSON
a Objetos

```

    JsonParser parser = new JsonParser();
    Gson gson = new Gson();
    JsonObject dataJson =
    (JsonObject) parser.parse(responseStrBuilder.toString());
    JSONArray infoFigures =
    (JSONArray) dataJson.get("infoFigures");

```

 <p>eman ta zabal zazu universidad del país vasco euskal herriko unibertsitatea</p>	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

```

if(infoFigures != null){
    for(JsonElement jsonInfoFigure: infoFigures){
        InfoFigure infoFigure =
        (InfoFigure)gson.fromJson(jsonInfoFigure, InfoFigure.class);
        new DownloadFigure(mActivity, docNo).execute(infoFigure);
    }
}

```

Ejecutar tarea asíncrona

Preparación del WebView para la comunicación entre JavaScript y Java

Al contenido que se va a cargar en el WebView se le ha dado funcionalidad al hacer clic sobre algunos de los elementos como pueden ser links o imágenes. Para conseguir que estos elementos de HTML se comuniquen con la aplicación Java hay que realizar varias cosas:

Obtener el WebView

```

WebView mWebView =
    (WebView) rootView.findViewById(R.id.documentWebView)

```

Habilitar el javascript en el WebView

```

mWebView.getSettings().setJavaScriptEnabled(true);

```

Crear una clase en Java que implement los métodos que vamos a llamar en javascript

```

public class JavaScriptInterface {
    @JavascriptInterface
    public void verDocumento(){}

    public Integer metodo2(){}
}

```

Crear una instancia de esta clase y asignársela al WebView

```

mWebView.addJavascriptInterface(new
    JavaScriptInterface(getActivity()), "JSInterface");

```

Para llamar a los métodos de la clase "JavaScriptInterface" desde el código JavaScript hay que utilizar el objeto "JSInterface" que se ha creado en el paso 4. Por ejemplo para llamar al verDocumento(), en JavaScript hay que poner lo siguiente:



```

JSInterface.verDocumento()

```

Para que los métodos de la clase JavaScriptInterface puedan ser llamados desde JavaScript hay que ponerles la directiva (@JavascriptInterface), si no se producirá un error al hacer la llamada.

Este comportamiento se ha implementado para detectar el clic sobre las imágenes y los links de los documentos, para ello se han hecho modificaciones en las plantillas XSL para poder devolver el HTML en función de si el que lo solicita es la aplicación móvil o la

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

aplicación web. Estos cambios se explican en el apartado de “Generación de código HTML de documento”.

7.5 GENERACIÓN DE CÓDIGO HTML DE DOCUMENTO

Para esta operación se ha tenido que modificar parte del código de la aplicación ya existente en la parte del servidor así como las plantillas XSL que se encargan de transformar los XML. A continuación se explican los cambios realizados sobre los documentos XSL para que genere un código HTML u otro dependiendo de si la petición se realiza desde la aplicación móvil o desde un navegador. A continuación se describen los cambios realizados sobre los documentos existentes.

7.5.1 Cambios en XSL

1. Nuevo parámetro para saber de dónde se está realizando la petición:

```
<xsl:param name="IS_TABLET"/>
```

En función de este parámetro se obtendrá un resultado u otro.

2. Links entre documentos:

Originalmente en la plantilla había la siguiente línea de código para generar los links entre documentos:

```
<a>
<xsl:attribute name="class">link</xsl:attribute>
<xsl:attribute name="href">
<xsl:value-of select="$BASE_LINK"/><xsl:value-of select="@refId"/>
</xsl:attribute>
</a>
```

Ahora el código ha quedado de la siguiente forma:

En caso de que la petición sea de la aplicación móvil.

```
<xsl:choose>
<xsl:when test="$IS_TABLET = 'true'">
<xsl:attribute
name="onclick">JSInterface.moveToDocument ('<xsl:value-of
select="@refId"/>')</xsl:attribute>
<xsl:attribute name="class">link</xsl:attribute>
</xsl:when>
```

Interfaz Java definida en la aplicación móvil

```
<xsl:otherwise>
<xsl:attribute name="href"><xsl:value-of
select="$BASE_LINK"/><xsl:value-of select="@refId"/></xsl:attribute>
</xsl:otherwise>
</xsl:choose>
```

Con este código se consigue que cuando a la plantilla se le pase el parámetro IS_TABLET con el valor “true”, genere un atributo “onclick” en los elementos “<a>” del HTML que va a llamar a la interfaz Java que se le ha asignado al WebView.

3. Código JavaScript para detectar el clic en las imágenes

Originalmente había el siguiente código en las plantillas para detectar el clic sobre las imágenes:

```
<img>
  <xsl:attribute name="onclick">
    <xsl:text>
      openImage('</xsl:text><xsl:value-of
                select="./@id"/><xsl:text>')
    </xsl:text>
  </xsl:attribute>
</img>
```

Ahora el código ha quedado de la siguiente forma:

```
<img>
  <xsl:attribute name="onclick">
    <xsl:choose>
      <xsl:when test="$IS_TABLET = 'true'">
        <xsl:text>JSInterface.openImage('</xsl:text>
        <xsl:value-of select="./@src"/>
        <xsl:text>')</xsl:text>
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>openImage('</xsl:text>
        <xsl:value-of
        select="./@id"/><xsl:text>')</xsl:text>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:attribute>
</img>
```

Llamada JavaScript a la interfaz implementada en la aplicación móvil

Con esta modificación se consigue que cuando se haga clic en una imagen en el WebView de la aplicación móvil, se ejecutara la función “openImage(figure)” a la cual se le pasa como parámetro el nombre de la figura, de manera que la podremos obtener de memoria y mostrarla en un dialogo donde el usuario podrá ampliarla y verla más en detalle.

7.5.2 Cambios en HTML

Rutas a los archivos css

Para que se carguen de forma correcta los ficheros css en el WebView y el HTML se formatee correctamente se añaden en el JSP donde se va a generar el HTML las siguientes líneas que hacen referencia a los css que se utilizan:

```
<link rel="stylesheet" type="text/css" href="./css/manual.css" />
<link rel="stylesheet" type="text/css" href="./css/comments.css" />
<link rel="stylesheet" type="text/css" href="./css/colors.css" />
```

Rutas relativas

Como se puede observar estas tres referencias a hojas de estilo se hacen de forma relativa puesto que, como se ha mencionado con anterioridad, los documentos se van a cargar en el WebView con una URL base donde existe una carpeta “css” que contendrán los archivos indicados.

7.6 INICIALIZACIÓN DE LA BASE DE DATOS

En este apartado se muestra el proceso para inicializar la base de datos. Se muestra el código de como se inicializa una de las tablas ya que el resto se realiza de la misma forma, cambiando los campos y propiedades de las tablas referentes a las clases.

El proceso de inicializar la base de datos se realice al instalar la aplicación y lo pasos a seguir son los siguientes:

1. Solicitar volcado de base de datos al servidor

Mediante una tarea asíncrona en la aplicación móvil se realice una petición al servidor:

```
HttpClient httpClient = new DefaultHttpClient();
HttpPost httpPost = new HttpPost(GeneralConstants.SERVER_URL+
    GeneralConstants.ACTION_SYNC_STRUCTURE_PUBLICATION+"?sync=every
thing");
HttpResponse response = httpClient.execute(httpPost);
InputStream is = response.getEntity().getContent();
```

En el servidor se generara un JSON que contenga todos los datos de las tablas que se quieren inicializar, para ello en el servidor se ejecuta el siguiente código por cada tabla que se sincroniza:

2. Obtener todos los registros de la base de datos:

```
SELECT * FROM documentsupply
```

```
listDocumentSupply = (ArrayList<DocumentSupply>)
DataSourceManager.getInstance().getAllDocumentSupply();
```

3. Convertir a JSON

Convertir a formato JSON el listado de los registro de la base de datos, para ello se utiliza la librería de flexjson:

```
JSONSerializer serializer = new JSONSerializer();
JsonParser parser = new JsonParser();
String listStringDocumentSupplyJsonArray =
serializer.exclude("*.class").serialize(listDocumentSupply);
JsonElement listDocumentSupplyJsonArray =
parser.parse(listStringDocumentSupplyJsonArray);
```

4. Añadir a la variable de resultado final el JSON resultante:

```
resultJson.add("listDocumentSupply", listDocumentSupplyJsonArray);
```

Este proceso se realiza por cada tabla que se va a sincronizar, y finalmente, cuando ya se han parseado todas las tablas a formato JSON se realice lo siguiente:

```
PrintWriter out = response.getWriter();
response.setContentType("application/json");
response.setCharacterEncoding("UTF-8");
response.setHeader("cache-control", "no-cache");
```

Codificar la información con UTF-8


```
String data = URLEncoder.encode(result, "UTF-8");
```

```
response.getWriter().write(data);  
return null;
```

Enviar respuesta en String

5. Enviar respuesta al cliente

Una vez que el servidor ha realizado todo el trabajo y ha generado el String con todos los datos se envía la respuesta al cliente donde se realiza la siguiente operación para leer los datos:

```
String encodedData = StringParser.convertStreamToString(is);  
String datos = URLEncoder.decode(encodedData, "UTF-8");
```

Decodificar datos

```
JsonParser parser = new JsonParser();  
Gson gson = new Gson();
```

```
JsonObject dataJson = (JsonObject)parser.parse(datos);
```

Pasar datos de String a JSON para poder recorrerlo facilmente

Recoger datos en el cliente

Una vez obtenido el objeto JSON podemos recorrerlo y hacer la siguiente operación por cada JSONArray que hay en el objeto, cada JSONArray contiene los datos de cada tabla:

```
jsonList = (JsonArray)dataJson.get("listDocumentSupply");  
for (JsonElement j : jsonList) {  
//leer propiedades, crear objeto y rellenar base de datos  
DocumentSupply ds = (DocumentSupply)gson.fromJson(j,  
DocumentSupply.class);  
AplicationSQLiteHelper.getInstance(ctx).insertDocumentSupply(ds);  
}
```

6. Almacenar datos en base de datos

A continuación se muestra como se realiza una inserción en la base de datos de la aplicación móvil:

```
public void insertDocumentSupply(DocumentSupply ds) {  
    SQLiteDatabase db = this.getWritableDatabase();  
    ContentValues values = new ContentValues();  
    values.put(DocumentSupply.COL_NO, ds.getDocumentSupplyNo());  
    values.put(DocumentSupply.COL_DOCUMENTNO, ds.getDocumentNo());  
    values.put(DocumentSupply.COL_SUPPLYNO, ds.getSupplyNo());  
    values.put(DocumentSupply.COL_QUANTITY, ds.getQuantity());  
    values.put(DocumentSupply.COL_POSITION, ds.getPosition());  
    try {  
        db.insert(DocumentSupply.TABLE_NAME, null, values);  
    } catch (Exception e) {  
    }  
}
```

Esta función está dentro de la clase que extiende de “ApplicationSQLiteHelper” la cual se ha creado para gestionar los accesos a base de datos. Esta clase extiende de “SQLiteOpenHelper” que es una clase que proporciona Android para manejar todos los aspectos de la base de datos SQLite.

7.7 CONSTRUCCIÓN DINÁMICA DE LA INTERFAZ

En este apartado se explica cómo se construye la interfaz principal de la documentación técnica. La interfaz depende de la información que hay en la base de datos, por lo que no se carga ningún interfaz estática como en las demás actividades. Esta actividad extiende de una actividad que maneja los “Tabs” que se llama “TabActivity”.



Lo primero que se hace es obtener de la base de datos todas las publicaciones que hay y por cada una de ellas se crea una nueva pestaña con los parámetros necesarios para mostrar el título y la acción que se debe realizar al seleccionarla, el código encargado de realizar esto es el siguiente:

```
//Obtain all the manuals from the database
ArrayList<ModelPublication> modelPublicationList =
    (ArrayList<ModelPublication>)
    ApplicationSQLiteHelper.getInstance(ctx).
    getPublicationByModel(GeneralConstants.MODEL);
//show all publications
for (ModelPublication mp : modelPublicationList) {
    //Create the intent for each TAB
    Intent intent = new Intent(this, ListManualsActivity.class);
    //Put Extras for intent
    Integer publicationNo = mp.getPublicationNo();
    intent.putExtra(ListManualsActivity.INTENT_PUBLICATION,
    publicationNo);
    if (publicationNo == publicationSelected) {
        position = i;
    }
}
//Obtener el título de la publicación
String publicationTitle = ApplicationSQLiteHelper.getInstance(ctx).
getPublicationTitleByPublicationNo(publicationNo,
GeneralConstants.IDIOM);

//Add a new TAB to the Host
getTabHost().addTab(mTabHost.newTabSpec(mp.getModelPublicationNo()
.toString())
.setIndicator(publicationTitle,
getResources().getDrawable(R.drawable.leftinstruction))
.setContent(intent));
i++;
}
```

Este “intent” contiene la que se debe lanzar cuando se seleccione el tab en cuestión. A este “intent” se le pasa como parámetro el código de la publicación para que muestre los manuales asociados.

Añadir un nuevo tab con el título de la publicación

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

7.8 INTERFAZ Y LÓGICA DE GRUPOS DE TRABAJO

En este apartado se explica el código que gestiona la parte de los grupos de trabajo. Se explica cómo está organizado.

Organización de la actividad:

Existen tres interfaces diferentes para completar los datos de un grupo de trabajo:

- Datos generales
- Trabajadores
- Instrucciones

Estas tres interfaces esta separadas en diferentes fragmentos y cada uno de ellos tienen su ciclo de vida. La actividad principal "WorkingGroupActivity" es la encargada de gestionar estos fragmentos.

Esta actividad crea los 3 fragmentos al iniciarse y dependiendo que opción seleccione el usuario se muestra uno y otro en la parte derecha/Inferior de la pantalla.

Cada fragmento mencionado es una clase que extiende de la clase Fragment.

- `public static class GeneralDataFragment extends Fragment{}`
- `public static class WorkersFragment extends Fragment {}`
- `public static class InstructionsFragment extends Fragment {}`

Cada uno de estos fragmentos se encarga de gestionar los eventos que surjan en sus interfaces. La actividad padre "WorkingGroupActivity" implementa los métodos para guardar en una variable de la clase los datos que le pasen los fragmentos.

Para la comunicación entre el padre y los fragmentos se utilizan los argumentos que se implementan de la siguiente forma:

Ejemplo: pasar un identificador al fragmento "WorkersFragment"

```
WorkersFragment wFragment;= wFragment = new WorkersFragment();
Bundle arg = new Bundle();
arg.putInt("identificador", ident);
wFragment.setArguments(arg);
```



Para mostrar el fragmento hay que realizar lo siguiente desde la actividad principal:

```
getFragmentManager().beginTransaction()
    .replace(R.id.container, wFragment)
    .commit();
```

Y por último para recoger los datos que se le han pasado al fragmento como argumento, en el método "onCreate()" del fragmento se realiza lo siguiente:

```
Integer ident = getArguments().getInt("identificador");
```

De manera que ya tenemos la comunicación entre actividad principal y fragmento, ahora hay que realizar el proceso inverso, es decir, que el fragmento devuelva datos a la actividad que

 <p>eman ta zabal zazu</p> <p>universidad del país vasco euskal herriko unibertsitatea</p>	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

lo creo, para ello se utiliza el patrón de programación "Delegate" el cual se explica a continuación:

La actividad que ha creado el documento debe implementar la interfaz la cual va a utilizar el fragmento para devolver los datos:

```
public class WorkingGroupActivity extends ActionBarActivity
implements SaveWorkingGroupInterface{}
```

El fragmento tiene un método por el que pasa al ser instanciado por la actividad el cual recibe el objeto actividad que lo instancia, en este punto podemos guardar la instancia de la actividad para después poder llamar al método de la interfaz que implementa:

```
@Override
public class onAttach(Activity activity) {
    this.mActivity = activity;
}
```

En el método "onCreateView" del fragmento cuando se termine todo el proceso y se desee devolver datos a la actividad padre se hace de la siguiente manera:

```
((SaveWorkingGroupInterface)mActivity).metodo(param1, param2...)
```

Si la actividad padre no implementa esta interfaz esta operación producirá una excepción.

7.9 PUBLICAR GRUPO DE TRABAJO

En este apartado se explica el código que implementa la acción de publicar un grupo de trabajo, lo que significa, comunicar a todos los dispositivos de que un nuevo grupo ha sido creado.

Esta acción se realiza mandando toda la información del grupo de trabajo al servidor y este es el responsable de lanzar las notificaciones a todos los dispositivos que en ese momento estén registrados en la base de datos.

Por la parte de la aplicación móvil se realiza la siguiente operación en una tarea asíncrona a la cual se le pasa como parámetro el objeto grupo de trabajo (gt), para ejecutar dicha tarea asíncrona se realiza lo siguiente:

```
WorkingGroupBroadcastNotification wgBroadcast =
    new WorkingGroupBroadcastNotification(mActivity);
wgBroadcast.execute(wgActual);
```

La tarea asíncrona realiza lo siguiente:

```
WorkingGroup mWorkingGroup = params[0],
```

Convertir objeto a JSON (felxjson)

```
String workingGroupStringJson = new
JSONSerializer().exclude("*.class").include("workerList",
"instructionList")
.rootName("workingGroup").serialize(mWorkingGroup);
```

```
//Initialize connection
HttpClient httpClient = new DefaultHttpClient()
HttpPost httpPost =
new
HttpPost (GeneralConstants.SERVER_URL+GeneralConstants.ACTION_WORKING
GROUP_NOTIFICATION+"?insert=true");
httpPost.setHeader ("Content-Type", HEADER_JSON_ENCODING);
```

Rellenar cabecera de conexión

```
httpPost.setEntity(
new ByteArrayEntity(workingGroupStringJson.getBytes("UTF-8")));
```

```
HttpResponse response = httpClient.execute(httpPost)
InputStream is = response.getEntity().getContent()
```

Añadir datos a la petición con codificación UTF-8

Realizar petición y recoger respuesta

Al realizar la petición al servidor con los datos del grupo de trabajo lo que sucede en el servidor es lo siguiente:

1. Leer los datos de la petición y generar JSON

```
StringBuilder sb = new StringBuilder();
BufferedReader bufferedReader = null;
InputStream inputStream = request.getInputStream();
if (inputStream != null) {
bufferedReader = new BufferedReader(new
InputStreamReader(inputStream));
}
char[] charBuffer = new char[1024];
int bytesRead = -1;
while ((bytesRead = bufferedReader.read(charBuffer)) > 0) {
sb.append(charBuffer, 0, bytesRead);
}
```

Leer el inputStream de la petición



```
String wgString = sb.toString();
JsonParser parser = new JsonParser();
Gson gson = new Gson();
```

Obtener el elemento JSON del grupo de trabajo (felxjson)

```
JsonObject wgJson = (JsonObject) parser.parse(wgString);
JsonObject workingGroup = (JsonObject) wgJson.get("workingGroup");
```

2. Obtiene el listado de los dispositivos registrados

Para obtener el listado de dispositivos basta con hacer una consulta a la base de datos, la cual es la siguiente:

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

```
select t.gcm_registration_id from t_gcm as t
```

3. Realiza la notificación con el servicio de Google

Para utilizar el servicio de notificación de Google hay que ajustar varios elementos para que la comunicación sea válida:

- Establecer cabeceras correctas

La petición que se realiza tiene que ser por POST y se debe incluir en la cabecera de conexión el atributo "Authorization" con el valor de "key=API_SERVER_KEY" y en este caso como los datos que se van a enviar van en formato JSON, se lo indicamos con el atributo "Content-Type" con el valor "application/json".

```
URLConnection con = (URLConnection) obj.openConnection();
//add request header
con.setRequestMethod("POST");
con.setRequestProperty("Authorization", "key=" + API_SERVER_KEY);
con.setRequestProperty("Content-Type", HEADER_JSON_ENCODING);
```

- Formatear datos de envío

Los datos que vamos a enviar van a ir en formato JSON y tienen que contener un JSONArray que se llame "registration_ids" y que cada elemento del Array sea un identificador de registro que hay en la base de datos. También hay que añadir un objeto JSON para transmitir los datos que deseamos, este objeto se tiene que llamar "data" y puede contener todos los datos que deseamos, en este caso contendrá un grupo de trabajo.



```
JsonObject result = new JsonObject();
//Add data
JsonObject data = new JsonObject();
data.add("workingGroup", workingGroup);
result.add("data", data);

//Add destinies
String jsonArray = serializer.serialize(tabletIds);
JsonParser parser = new JsonParser();
JsonElement jsonArrayIds = parser.parse(jsonArray);
result.add("registration_ids", jsonArrayIds.getAsJsonArray());
```

- Realizar la petición a la siguiente URL : <https://android.googleapis.com/gcm/send>

Con las siguientes líneas de código se envían los datos que hemos rellenado en el JSON.

```
// Send post request
con.setDoOutput(true);
DataOutputStream wr = new DataOutputStream(con.getOutputStream());
wr.writeBytes(result.toString());
wr.flush();
wr.close();
```

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

8 - PRUEBAS REALIZADAS

En los siguientes apartados se describen las diferentes pruebas que se han hecho a la aplicación, tanto en desarrollo como al prototipo.

8.1 DURANTE EL DESARROLLO

Desactivar conexión una vez iniciada sesión con conexión

El objetivo de esta prueba es verificar que la aplicación siga funcionando aun y así no teniendo conexión a internet, evidentemente los documentos que aún no se encuentran en el dispositivo será imposible visualizarlos.

Una vez arrancado el servidor e iniciada la sesión en la aplicación con acceso a internet, se desactiva el WIFI del dispositivo para dejar a la aplicación sin conexión. Para comprobar que los documentos están disponibles se navega por las diferentes pantallas de la documentación técnica.

Con esta prueba se han obtenido resultados que han permitido mejorar el aspecto visual de la aplicación mostrando diferentes diálogos y mensajes cuando la conexión no está disponible para aquellas operaciones que lo requieren.

Enviar datos repetidos al servidor

El objetivo de esta prueba es comprobar que los datos que se envían al servidor no se van a duplicar ya que dos usuarios pueden realizar la misma operación al mismo tiempo.

El desarrollo de la prueba se realiza con dos tablets con la misma aplicación instalada y se disponen a actualizar el estado de una misma instrucción de un mismo grupo de trabajo al está "completada", aunque es un caso extraño se puede dar y hay que tenerlo en cuenta. Al llegar la primer petición al servidor y actualizar la base de datos satisfactoriamente, la segunda petición, ya que se atienden síncronamente, recibirá un mensaje de error informándole de que dicha instrucción ya ha sido completada por otro usuario.

El resultado de esta prueba es el esperado y no son necesarios ningún cambio en la aplicación.



Girar dispositivo en las diferentes actividades

El objetivo de esta prueba es comprobar que se cargan correctamente las diferentes interfaces dependientes de la orientación y que todos sus controles funcionan como deben. Para la realización de esta prueba se utiliza un dispositivo con el control de orientación activado y se navega por todas las actividades de la aplicación y se van comprobando ambas orientaciones.

El resultado de esta prueba desvela un error en el código de la actividad "grupos de trabajo" lo que conlleva un cambio en el tipo de las variables que se utilizan en la misma.



Probar usabilidad de la aplicación

El objetivo de esta prueba es ver si la aplicación es intuitiva y el usuario no se pierde a la hora de utilizarla, es decir, si todas las ayudas visuales que se muestran son suficientes o por el contrario serian necesarias otras.

<p>eman ta zabal zazu</p>  <p>universidad del país vasco euskal herriko unibertsitatea</p>	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

Para el desarrollo de esta prueba se ponen en funcionamiento todo el sistema, tanto el servidor como dos dispositivos los cuales los manejan dos personas que nunca han visto la aplicación pero si se les ha informado de lo que tiene que realizar.

Los usuarios que participan en esta prueba me comunican como resultado que serian necesarias varias ayudas visuales ya que había momentos en que no sabían qué hacer, como por ejemplo a la hora de publicar el “grupo de trabajo”.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

9 - COMO ENTREGAR EL PRODUCTO AL CLIENTE

En este capítulo se explican dos formas diferentes de entregar el producto al cliente. Contando con que la parte del servidor ya está instalada y en funcionamiento en un Hosting, la parte del cliente se podría entrega de las siguientes maneras:

9.1 GOOGLE PLAY MARKET

Al tratarse de una aplicación Android, parece obvio que subir la aplicación al Market es una de las opciones.

La aplicación tiene una restricción la cual deben cumplir los dispositivos en los que se quieran instalar. Esta restricción es la versión del sistema operativo Android, como mínimo tiene que ser la versión 4.0.



Hay un problema, el cual es que el círculo de usuarios que van a utilizar esta aplicación, es un círculo muy pequeño del cual desconocemos si disponen de una tableta para poder utilizar la aplicación. Por esta razón esta opción se va a quedar en un segundo plano.

9.2 ENTREGAR TABLETA CON SOFTWARE

Esta forma de entrega nos asegura de que la aplicación va a funcionar correctamente en los dispositivos que se entreguen. Una de las principales ventajas de este modo de entrega, es la posibilidad de hacer pruebas en los dispositivos finales que van a recibir los clientes.

La idea es la siguiente:

1. Geminys S.L. adquiriría de su proveedor de confianza tantos dispositivos como sean necesarios.
2. El trabajador asignado por la gerencia sería el responsable de instalar la aplicación y puesta a punto de los dispositivos.
3. Una tercera persona revisaría la aplicación en buscar de posibles fallos o comportamientos extraños.
4. Una vez dado el visto bueno al dispositivo y a la aplicación, se empaquetaría y se enviaría al cliente junto con un manual de usuario de la aplicación.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

10 - CONCLUSIONES

Esta parte de la memoria muestra los resultados del proyecto de una manera más informal que las anteriores, de tal forma que realiza un compendio y valoración sobre el resultado global del proyecto.

10.1 CONCLUSIONES SOBRE EL PROYECTO

Las valoraciones finales sobre el proyecto se pueden desglosar de la siguiente manera:

a. Adquisición de nuevas competencias en el ámbito de desarrollo informático

El proyecto que se ha realizado suponía un reto ya que tenía conocimientos básicos sobre el desarrollo de aplicaciones móviles pero el nivel de esta aplicación ha requerido que tenga que investigar, entender y utilizar nuevas herramientas y técnicas a la hora del desarrollo.

b. Mejor entendimiento sobre los procesos de la dirección de proyectos

Al realizar el proyecto en una empresa, he visto como es la vida empresarial y he comprendido que los plazos y la planificación es algo fundamental que se debe realizar con precisión ya que hay gente que depende de ello.

El proyecto lo he dejado en algún momento de lado dado que en la empresa surgían trabajos con mayor prioridad, lo que ha producido algunos retrasos en el mismo.

c. Nuevos conocimientos en el área del ferrocarril

Al tratarse de un proyecto que se va a utilizar para dar soporte a una documentación técnica de un tranvía, he aprendido que estos vehículos requieren de mucho mantenimiento para que su vida útil sea más larga, así como también ahora valoro mucho el contenido de los manuales que se adjuntan en cualquier aparato.

d. Experiencia en el trabajo empresarial



Realizando el proyecto en una empresa me he dado cuenta que el mundo laboral es mucho más exigente que el mundo universitario en cuanto a la exigencia de calidad, los tiempos de entrega, los imprevistos que surgen y todos los procesos que se deben realizar para completar un trabajo.

e. Integración en la empresa Geminys S.L.

Gracias al desarrollo de este proyecto he conseguido integrarme en la empresa y me ha ayudado a entender y utilizar un producto software que se ha desarrollado por el equipo de informáticos de Geminys S.L. sobre el cual trabaja la aplicación que he implementado en este proyecto.

10.2 OBJETIVOS PRINCIPALES CUMPLIDOS

Los objetivos que se plantearon al inicio del proyecto se han cumplido con éxito, lo cual proporciona a la empresa un producto que podrá ofrecer a sus clientes. Se ha desarrollado de tal forma que se pueda adaptar fácilmente a cada cliente, configurando la aplicación para cada uno de ellos en pocas horas.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

Se presentó el prototipo a los directivos de la empresa obteniendo una respuesta positiva, aceptando así el proyecto como posible producto que ofrecer a los clientes. Se hicieron comentarios sobre posibles mejoras, pero que quedarían fuera del alcance de este proyecto.

En la actualidad existe una versión de la aplicación web adaptable a distintas resoluciones la cual ya está en funcionamiento y utilizándose por algunos clientes. Sin querer hacer sombra a la aplicación web en la que se basa este proyecto, se pueden destacar las siguientes ventajas que ofrece la aplicación móvil:

a. Mayor navegabilidad

Al tener las publicaciones en pestañas en vez de en un listado podemos navegar de una publicación a otra sin tener que ir hacia atrás en la aplicación. La aplicación móvil mantiene el último documento visto en cada publicación.

b. Independencia de Internet

Al descargarse los documentos y almacenarlos de forma local siempre estará la documentación disponible, aunque no haya conexión, lo que no ocurre con la aplicación web.

c. Velocidad a la hora de cargar documentos

Al utilizar el hardware del dispositivo y además de no tener que realizar ninguna petición al servidor, los documentos se visualizan mucho más rápido que en la web.

d. Nueva funcionalidad

La funcionalidad de grupos de trabajo le da un nuevo uso a la documentación técnica, permitiendo llevar un control de los trabajos realizados por cada trabajador.

10.3 LECCIONES APRENDIDAS

Para finalizar estas conclusiones me gustaría enumerar un conjunto de lecciones aprendidas.

- **Lección 1**

Lanzar una nueva actividad desde un contexto que no es una actividad.

- **Lección 2**

Crear y utilizar vistas en base de datos SQLITE.

- **Lección 3**



Concatenar campos en las consultas a la base de datos SQLITE para que me devuelva un String con el formato que deseo.

- **Lección 4**

Utilizar fragmentos en Android para visualizar el mismo contenido en diferentes partes de la aplicación sin repetir código.

- **Lección 5**

Manejo de sistemas de archivos de Android desde código.

<p>eman ta zabal zazu</p>  <p>universidad del país vasco euskal herriko unibertsitatea</p>	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

- **Lección 6**

Utilización de cursores para recorrer la colección de datos que devuelve una consulta de SQLite.

- **Lección 7**



Comunicación entre cliente y servidor mediante tareas de fondo en el dispositivo.

- **Lección 8**

Crear base de datos al volver en el dispositivo cliente.

- **Lección 9**

Difusión de mensaje a dispositivos mediante notificaciones **push**.

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

ANEXO 1 – ACTA DE CONSTITUCIÓN

Introducción

En el presente documento se recoge la descripción de los componentes del proyecto que se va a desarrollar en la empresa Geminys S.L. como proyecto fin de carrera por parte del alumno Miguel José Pérez Tejada, estudiante de la Universidad del País Vasco en la facultad de informática.

La empresa en cuestión, Geminys S.L, es una de las empresas más potentes en su campo de desarrollo, redacción de documentación técnica. La documentación se ofrece al cliente en distintos formatos como puede ser el papel, papel plástico especial, distintos soportes informáticos (Word, PDF, HTML). Actualmente disponen de una aplicación web para ofrecer soporte a dicha documentación dicha aplicación tiene un diseño adaptable a diferentes resoluciones de pantallas, por lo que se visualiza correctamente en cualquier dispositivo.

Una de las cosas que actualmente no disponen es de un soporte digital nativo para dispositivos móviles. La principal ventaja de un soporte digital nativo para la documentación es la permanente disponibilidad de dicha documentación sin dependencia alguna de la conexión a internet.

Objetivos

El objetivo principal de este proyecto es desarrollar una aplicación capaz de visualizar toda la documentación técnica de un proyecto desarrollado en Geminys S.L. También deberá mantener una versión actualizada de la documentación siempre y cuando disponga de conexión a internet.

Como objetivos secundarios se contemplan las siguientes ideas:

- Funcionalidad de grupos de trabajo
- Generación de códigos QR

Requisitos

Uno de los requisitos es la privacidad de la información que va a gestionar la aplicación, por lo que dicha aplicación deberá llevar un proceso de autenticación para poder acceder a la documentación.

Riesgos

Posibilidad de que el alumno no estime correctamente los tiempos debido a que es la primera vez que se enfrenta a un proyecto de esta envergadura y además deberá invertir tiempo en conocer el mundo de la documentación técnica y sus requerimientos.

Resumen de hitos



A lo largo del desarrollo del proyecto el alumno presentará los progresos realizados en la aplicación mensualmente a la patrocinadora, los cuales tendrán que ser aprobados por la misma. Al acercarse al cierre del proyecto se realizaran pruebas con más de un dispositivo para simular un escenario real de trabajo, dichas pruebas serán supervisadas y dirigidas por la patrocinadora y los responsables de informática de la empresa.

Resumen del presupuesto

Geminys S.L. pone a disposición del alumno todo el material de oficina necesario, también proporciona soportes informáticos e instalaciones, así como recursos humanos para el desarrollo del proyecto. Se compromete a adquirir un dispositivo móvil apropiado para realizar las pruebas pertinentes en el desarrollo del proyecto.

Requisitos de aprobación del proyecto

Para que el presente proyecto sea considerado exitoso, los objetivos principales deberán cumplirse en el plazo estipulado. Para la comprobación del cumplimiento de dichos objetivos la patrocinadora del proyecto, Isabel Azanza gerente de la empresa Geminys S.L, verificará dichos objetivos y en caso de éxito será la responsable de firmar la aprobación del proyecto.

 <p>eman ta zabal zazu</p> <p>universidad del país vasco euskal herriko unibertsitatea</p>	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

ANEXO 2 – ACTAS DE REUNION

ACTA REUNIÓN GEMINYS / ALUMNO UPV-EHU – 28 ENERO 2014

ASISTENTES

NOMBRE	CARGO
Jose Angel Vadillo	Director de Proyecto
Mikel Pérez	Alumno

TEMAS TRATADOS

1. Interfaces presentadas en la empresa
2. Documentación a presentar

Se añadirán a la documentación las transformaciones XML/HTML.

Implementar interfaz, establecer y redactar casos de uso

Redactar la arquitectura de la aplicación.

Realizar modelo de datos junto con el flujo de eventos.

Redactar los requisitos de la aplicación a nivel global

3. Fechas para la defensa del Proyecto

La defensa se plantea hacerla en Julio.

DOCUMENTACION ENTREGADA



Interfaces diseñadas.

PRÓXIMA REUNIÓN

No se ha concretado.

TIEMPO DE LA REUNIÓN

1 hora

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

ACTA REUNIÓN GEMINYS / ALUMNO UPV-EHU – 17 FEBRERO 2014

ASISTENTES

NOMBRE	CARGO
Isabel Azanza	Gerente
Jesus Sanchez	Jefe del servicio técnico auxiliar
Javier Zuzuarregui	Jefe del servicio de informática
Mikel Pérez	Alumno

TEMAS TRATADOS

1. Documento de Objetivos de Proyecto

Enviar documento a los asistentes para que hagan comentarios sobre el contenido y el formato.
2. Presentación de interfaces

Se han comentado los diseños que el alumno ha diseñado.
3. Acuerdo de privacidad

Se estudiará como poder proporcionar esta privacidad a la aplicación.
4. Diseño de la aplicación

Se ha sugerido al alumno ponerse en contacto con el diseñador habitual que trabaja para Geminys S.L. (Iñigo Gutiérrez de la empresa ASISI)
5. Plan de mantenimiento

Se aconseja al alumno mirar los distintos planes de mantenimiento de los proyectos realizados en la empresa.



Se aconseja al alumno escribir un email para solicitar a todos los jefes de proyecto información sobre los planes de mantenimiento más variados que han tenido que elaborar. (Hector, Alberto, Amaia, Ainara, Mario, Carlos, Josu, Jesús)
6. Comportamiento extra de la interfaz

Al hacer “tap” sobre el título de la instrucción que pase al modo “Paso a Paso” en ese punto en concreto.

Condiciones de seguridad de las instrucciones, deben estar siempre accesibles. Una condición de seguridad es un metadato.
7. Catálogo

Se aconseja hablar con la persona que mas conoce el catálogo, Eider.
8. Orientación de la aplicación

En un principio todas las pantallas están diseñadas para la vista en vertical, se propone que al girar la tableta la vista cambie y aproveche todo el espacio disponible. Se estudiará y se propondrán nuevos diseños.
9. Reunión con usuario final para mejoras

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		



Se ha acordado que el alumno visitara al usuario final de la aplicación una vez tenga el prototipo.

DOCUMENTACION ENTREGADA

Diseños de pantallas a papel y lápiz.

TIEMPO DE LA REUNIÓN

2 horas

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

ACTA REUNIÓN GEMINYS / ALUMNO UPV-EHU – 3 MARZO 2014

ASISTENTES

NOMBRE	CARGO
Iker Rejano	Ingeniero Informático
Mikel Pérez	Alumno

TEMAS TRATADOS

-Autenticación en el IETM

Ya que la aplicación móvil necesita obtener datos del IETM, es necesario crear algún mecanismo para poder obtener estos datos sin necesidad de autenticarse en la aplicación. Iker me explica cómo se puede dar acceso a los datos, es decir, simular que el usuario está autenticado, de forma que solo algunos “actions” tengan este privilegio.



Para ello se utiliza el Netbeans para mostrar el código que gestiona la autenticación en la aplicación. Me indica como lo hacen actualmente y yo me tengo que amoldar a este funcionamiento y adaptar los “actions” que quiera que tengan estos privilegios a la estructura actual de la aplicación.

DOCUMENTACION ENTREGADA

Código del IETM

TIEMPO DE LA REUNIÓN

1 hora

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

ACTA REUNIÓN GEMINYS / ALUMNO UPV-EHU – 4 MARZO 2014

ASISTENTES

NOMBRE	CARGO
Iker Rejano	Ingeniero Informático
Mikel Pérez	Alumno

TEMAS TRATADOS

-Explicación general de la generación de HTML en el IETM

En la aplicación móvil es necesario recoger el HTML que se genera en el servidor para visualizar los documentos.

Con el Netbeans, Iker me indica la secuencia de acciones que se realiza cada vez que se solicita un documento en la aplicación web.



Esta secuencia de acciones tendré que replicarlo en un “action” separado para llamarlo desde la aplicación móvil.

Iker me indica que tendré que modificar algunos parámetros en los documentos XSL, para que se visualicen correctamente los documentos en los dispositivos.

DOCUMENTACION ENTREGADA

TIEMPO DE LA REUNIÓN

1 hora 15 minutos

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

ACTA REUNIÓN GEMINYS / ALUMNO UPV-EHU – 10 MARZO 2014

ASISTENTES

NOMBRE	CARGO
Iker Rejano	Ingeniero Informático
Mikel Pérez	Alumno

TEMAS TRATADOS

-Utilizar parámetros en plantillas XSL



Debido a que algunas cosas se ven de diferente forma en la aplicación web y en la aplicación móvil, hay que mantener ambos comportamientos sin duplicar las plantillas existentes.

Para ello Iker me explica como pasar parámetros a las plantillas y dependiendo de estos saber si se trata de una petición de la aplicación móvil o no.

DOCUMENTACION ENTREGADA

TIEMPO DE LA REUNIÓN

30 minutos

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

ACTA REUNIÓN GEMINYS / ALUMNO UPV-EHU – 1 ABRIL 2014

ASISTENTES

NOMBRE	CARGO
Iker Rejano	Ingeniero Informático
Mikel Pérez	Alumno

TEMAS TRATADOS

-Identificar figuras asociadas

Dado que en la aplicación móvil por el momento solo se descarga el HTML que se genera, es necesario saber que imágenes están asociadas a cada documento.



Para ello me explica Iker que tiene un método implementado que obtiene la lista de figuras que un documento tiene asociado.

Tendré que llamar a este método desde la aplicación móvil para que me devuelva la lista de figuras y así poder descargarlas mediante la URL que me indica Iker, pasando como parámetro el identificador de la figura.

DOCUMENTACION ENTREGADA

TIEMPO DE LA REUNIÓN

30 minutos

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

ACTA REUNIÓN GEMINYS / ALUMNO UPV-EHU – 7 ABRIL 2014

ASISTENTES

NOMBRE	CARGO
Iker Rejano	Ingeniero Informático
Mikel Pérez	Alumno

TEMAS TRATADOS

-Recursos para renderizar HTML



El HTML generado por el servidor y recogido por la aplicación no se visualiza correctamente en los dispositivos debido a que no se les está aplicando los CSS adecuados y además faltan algunas figuras estáticas que no están asociadas al documento.

Iker me indica cuales son los archivos que deberían existir en la aplicación móvil para que se visualice todo el contenido de los documentos correctamente.

DOCUMENTACION ENTREGADA

TIEMPO DE LA REUNIÓN



30 minutos

 <p>eman ta zabal zazu</p> <p>universidad del país vasco euskal herriko unibertsitatea</p>	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

ANEXO 3 – CIERRE DEL PROYECTO

El presente documento da conformidad a la finalización del proyecto fin de carrera desarrollado por el alumno de la Universidad del País Vasco, Miguel José Pérez Tejeda, en la empresa Geminys S.L.

El proyecto se inicio en Enero del 2014 en la empresa Geminys S.L. con el convenio de proyecto fin de carrera entre la Universidad del País Vasco, la empresa Geminys S.L. y el alumno Miguel José Pérez Tejeda. Dicho proyecto ha sido desarrollado dentro de las fechas establecidas y con los objetivos principales cumplidos.



 <p>eman ta zabal zazu</p> <p>universidad del país vasco euskal herriko unibertsitatea</p>	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
	<p>Proyecto fin de carrera</p>	

ANEXO 4 – MATRIZ DE RASTREABILIDAD

Para el seguimiento de los requisitos del proyecto se ha utilizado una matriz de rastreabilidad, la cual se muestra en la siguiente imagen.

Requisito	Origen	Objetivo	Estado	Fecha inicio	Fecha fin	Prioridad	Descripción
Replicar estructura de datos	Necesidad	Crear un proveedor de datos al que pueda llamar la aplicación cliente para recoger los datos y almacenarlos de forma permanente. También servirá para realizar las futuras actualizaciones automáticas.	Finalizado	10/03/2014	20/04/2014	Alta	Volcar todos los datos existentes en el dispositivo ya que son necesarios para mostrar correctamente todos los documentos en la tableta sin necesidad de conexión a internet.
Crear servicio web							
Recoger datos en cliente							
Soporte off-line	Idea del desarrollador	Disponer de toda la documentación técnica sin disponer de conexión a red e soporte digital.	Finalizado	10/03/2014	20/04/2014	Alta	Se debe de desarrollar un sistema que permita al usuario navegar por la documentación de una forma fluida, total y sencilla sin necesidad de conexión a internet.
Autenticación local	Necesidad	Poder acceder a la documentación sin internet	Finalizado	15/03/2014	25/03/2014	Alta	Proporcionar un sistema de autenticación cuando no haya acceso a internet
Documentación actualizada	Necesidad	Los documentos deben estar actualizados con respecto a los que están en el servidor.				Sin definir	El sistema debe estar de alguna forma sincronizado con los documentos que están alojados en el servidor, cuando se produzcan cambios sobre ellos, la aplicación debe descargarlos para tenerlos disponibles sin conexión.
Seguridad	Confidencialidad de datos	Establecer una capa de seguridad que no permita a terceros ver la información que se maneja en los	Finalizando			Media	Conseguir un sistema seguro al que solo puedan acceder los usuarios que estén autorizados y solo en los
Autenticación	Necesidad	Característica necesaria para identificar autorizar el uso de la	Finalizando	12/03/2014	17/03/2014	Alta	
Borrado automático de datos	Idea del desarrollador	Permitir al administrador del sistema seleccionar un dispositivo y borrar todos los datos que tiene	No iniciada			Baja	escenarios configurados.
Creación de usuarios	Necesidad	Proporcionar una manera de dar de alta a nuevos usuarios en la	Finalizado	12/03/2014	15/03/2014	Media	
Códigos QR	Tecnológico	Poder generar códigos QR relacionados unequivocamente con cada instrucción.	No iniciado			Media	Generar un código QR por cada instrucción, subirlos al servidor e imprimirlos para pegarlos en la parte asociada a la instrucción y así poder saber instantáneamente cuál es la instrucción que se debe realizar.
Grupos de trabajo	Idea del desarrollador	Poder comunicar grupos de trabajo a usuarios	Finalizado	16/04/2014	30/04/2014	Media	Crear un sistema de notificaciones automáticas que comunique a los trabajadores las instrucciones que deben hacer.
Crear			Finalizado	16/04/2014	18/04/2014		
Utilizar(Worker)			Finalizado	18/04/2014	23/04/2014		
Notificar			Finalizado	24/04/2014	30/04/2014		
Documentar instrucciones realizadas		Sustituir las hojas de control actuales	En progreso	18/04/2014	25/04/2014		Ofrecer una nueva funcionalidad en la que se le da uso a la documentación para repartir el trabajo en los talleres creando grupos de trabajo.
Sonido	Idea del desarrollador		No iniciado			Media	
Imagen	Idea del desarrollador		Finalizado	18/04/2014	25/04/2014		
Texto	Idea del desarrollador		No iniciado				
Multi-idioma	Cliente	Poder disponer de la misma aplicación en varios idiomas	No iniciado				
Notificaciones Push	necesidad	Registrar dispositivos en GCM	Finalizado	17/04/2014	18/04/2014	Media	Registrar todos los dispositivos donde se instala la aplicación para poder enviar notificaciones de actualización y mensajes.
Sincronización automática de datos	Necesidad	Sincronizar datos sin necesidad de acciones del usuario	Finalizada	25/03/2014	15/04/2014	alta	

Ilustración 12 – Matriz de rastreabilidad

	<p>Construcción de un sistema de información para la gestión de la documentación técnica adaptada a un dispositivo móvil</p>	
<p>Proyecto fin de carrera</p>		

ANEXO 5 – MANUAL DE INSTALACION

Para la instalación del sistema completo son necesarios los siguientes programas:

Para el lado del servidor:

Programa de instalación

- Java SE Development Kit (jdk-7):

```
$PATHINST_JAVA=D:\Programas\Java\jre7\
```

- Apache Tomcat 7:

```
$PATHINST_TOMCAT=D:\Programas\Tomcat7\  
$PATHAPPS_TOMCAT=D:\IETM\webapps\
```

- MySQL 5.5 Server:

```
$PATHINST_MYSQL=D:\Programas\MySQL\MySQLServer\
```

- MySQL WorkBench:

```
$PATHINST_MYSQL=D:\Programas\MySQL\MySQLWorkbench\
```

Los instaladores de estos programas se encuentran en:

```
CD-ROM:\OriginalesIETM\programas de instalación\
```

Datos de aplicación:

```
$PATHINST_IETMDATA= CD-ROM:\IETM\root
```

Configuración de programas

1) Apache Tomcat 7

Instalación estándar. Instalado como servicio del sistema. El servicio se iniciará automáticamente al iniciar la máquina, de lo contrario el servicio deberá estar iniciado para que la aplicación funcione.

El puerto utilizado por el servidor es el 80. Para modificarlo ir al fichero de configuración:

```
$PATHINST_TOMCAT+conf\server.xml
```

Línea:

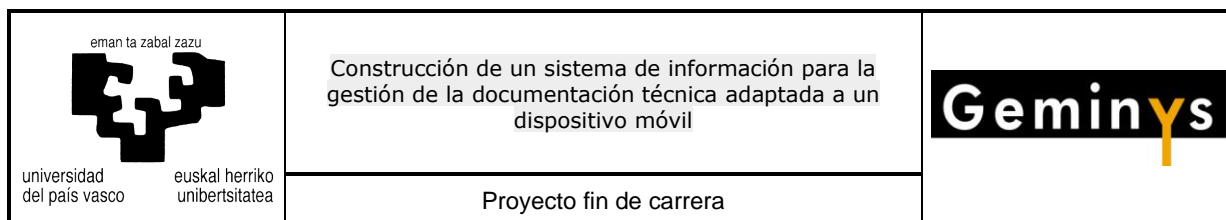
```
<Connector port="80" protocol="HTTP/1.1" connectionTimeout="20000"  
redirectPort="8443" />
```

Las aplicaciones del Tomcat se encuentran bajo el path de la carpeta webapps. Para modificarla ir al fichero de configuración:

```
$PATHINST_TOMCAT+conf\server.xml
```

Línea:

```
<Host name="localhost" appBase="$PATHAPPS_TOMCAT"  
unpackWARs="true" autoDeploy="true">
```

2) MySQL Server

Instalación estándar. Instalado como servicio del sistema. El servicio se iniciará automáticamente al iniciar la máquina, de lo contrario el servicio deberá estar iniciado para que la aplicación funcione.

El puerto utilizado por el servidor es el 3306.

La aplicación se apoya sobre una base de datos y el sistema de ficheros. La configuración está en dos ficheros:

1) FICHERO CONFIGURACIÓN BASE DE DATOS:

```
$PATHAPPS_TOMCAT+root\META-INF\context.xml
```

2) FICHERO CONFIGURACIÓN SISTEMA DE FICHEROS:

```
$PATHAPPS_TOMCAT+root\WEB-INF\classes\project.properties
```

Para el lado del cliente

Para el correcto funcionamiento de la aplicación, el dispositivo donde se instale debe cumplir los siguientes requisitos:

- Versión del S.O: Android superior o igual a 4.0
- Espacio en memoria: 10MB
- Acceso a internet
- Recomendable tener cámara

Para la instalación de la aplicación basta con copiar el archivo .apk en el dispositivo y abrirlo. Es posible que haya que habilitar la opción que permite la instalación de aplicaciones que no sean del Market.

Esta opción la podemos encontrar en “Ajustes” -> “Aplicaciones”, pero esto depende de la versión del sistema operativo.