MROGear App

Para dispositivos ***Android***

Memoria del Proyecto

Autor: Javier Uría Arias
Consultor: Jordi Ceballos Villach
Fecha de Entrega: 10/06/2013

Tabla de contenidos

1. Introducción 7

1.1 Situación actual – Operativa funcional 7
1.2 Justificación del Proyecto 9
2. Objetivos 11

3. Funcionalidades principales 12

4. Calendario del proyecto 14

4.1 Calendario 14
4.2 Detalle de entrega de PAC 1 – Plan de trabajo 15
4.3 Detalle de entrega de PAC 2 – Análisis funcional, Diseño técnico y Prototipo 16
4.4 Detalle de entrega de PAC 3 – Implementación 18
4.5 Detalle de entrega final 20
5. Infraestructura 21

5.1 Recursos de Hardware 22
5.2 Recursos de Software para los equipos de desarrollo y pruebas 22
5.3 Recursos de Software para el equipo servidor 23
6. Tecnología 24

6.1 Tecnología híbrida 24
6.2 Tecnología de la parte cliente. Dispositivo móvil 25
6.3 Tecnología de la parte servidora 26
6.4 Tecnología de red 27
7. Análisis de Riesgos 28

8. Análisis Funcional 30

8.1 Requerimientos Funcionales 30
8.1.1 Descripción Básica del Funcionamiento del aplicativo 30
8.1.2 Particularidades del aplicativo móvil 31
8.1.3 Seguridad 31
8.2 Requerimientos no Funcionales 32
8.2.1 Interfaz 32
8.2.2 Otros aspectos 32
8.2.3 Funcionalidades del sistema 33
8.2.4 Actores del sistema 34
8.2.5 Casos de Uso 34
9. Diseño Técnico 45

9.1 Arquitectura Física 45
9.1.1 Escenario de desarrollo 45
9.1.2 Escenario de producción con acceso público al servidor de aplicaciones 46
9.1.3 Escenario de producción con acceso público al servidor de aplicaciones con VPN 46
9.2 Arquitectura Lógica 46
9.2.1 Arquitectura Lógica de la capa cliente 47
9.2.2 Arquitectura Apache Cordova o PhoneGap 48
9.2.3 Arquitectura IBM Worklight 49
9.2.4 Arquitectura lógica de la plataforma servidora 49

9.3 Arquitectura de la Base de Datos 50
9.3.1 Modelo Relacional de la Base de Datos 50
9.4 Diagramas de Clases 55
9.4.1 Adaptadores. Capa de Modelo. 55
9.5 Diagramas de Secuencia 58
9.5.1 Diagrama de Secuencia – Ver Estadística 59
9.5.2 Diagrama de Secuencia – Alta Incidencia 60
10. Prototipo 62
10.1 Prototipo Menú Principal 63
10.2 Prototipo Incidencias 64
10.3 Prototipo Pantalla Detalle de Incidencias 65
10.4 Prototipo Almacén 67
10.5 Prototipo Mis Reservas 68
10.6 Prototipo Autorizaciones de Compra 69
10.7 Prototipo Menú Estadísticas 70
10.8 Prototipo Ayuda 70
11. Implementación 71
11.1 Premisas de la Implementación 71
11.1.1 Uso de Componentes visuales 71
11.1.2 Modalidad de trabajo desconectada 73
11.1.3 Depuración del código 74
11.1.4 Posicionamiento 75
11.1.5 Multi-idioma 76
11.1.6 Modalidad de pruebas 76
11.2 Proceso de Implementación y Entorno de desarrollo 77
11.2.1 Herramientas de desarrollo 78
11.2.2 Estructura del proyecto 79
11.2.3 Aspectos relacionados con el desarrollo 82
11.2.4 Otros Aspectos relacionados con el desarrollo 85
12. Conclusiones 86
12.1 Objetivos 86
12.2 Valoraciones personales 87
12.3 Líneas futuras 87
Bibliografía consultada 89

Índice de ilustraciones y tablas

Ilustración 1 - Desorden y falta de homogeneidad de la información 9
Ilustración 2 - Información ordenada 10
Ilustración 3 - Operario trabajando con dispositivo móvil 11
Ilustración 4 - Calendario general del proyecto 14
Ilustración 5 - Diagrama de Gantt detalle entrega PAC 1 – Plan de trabajo 15
Ilustración 6 - Diagrama de Gantt detalle entrega PAC 2 – Análisis funcional, Diseño técnico y Prototipo 16
Ilustración 7 - Diagrama de Gantt detalle entrega PAC 3 – Implementación 18
Ilustración 8 - Diagrama de Gantt detalle entrega final 20
Ilustración 9 – Diagrama de arquitectura general de la solución 21
Ilustración 10 - Comparativa tipos app 25
Ilustración 11 - Diagrama de acceso a datos por parte del dispositivo móvil 26
Ilustración 12 - Arquitectura general de un adaptador Worklight para acceso a datos 27
Ilustración 13 - Diagrama de Casos de Uso 35
Ilustración 14 - Escenario de desarrollo 45
Ilustración 15 - Escenario de producción con acceso público al servidor de aplicaciones 46
Ilustración 16 - Escenario de producción con acceso público al servidor de aplicaciones con VPN 46
Ilustración 17 - Modelo-Vista-Controlador 47
Ilustración 18 - Arquitectura Lógica de la capa cliente 48
Ilustración 19 - Arquitectura Apache Cordova 48
Ilustración 20 - Arquitectura IBM Worklight 49
Ilustración 21 - Arquitectura lógica de la plataforma servidora 50
Ilustración 22 - Modelo Relacional de la Base de Datos 50
Ilustración 23 - Diagrama de Secuencia – Ver Estadística 59
Ilustración 24 - Diagrama de Secuencia – Alta Incidencia 60
Ilustración 25 - Menú Principal 63
Ilustración 26 - Pantalla de Splash 63
Ilustración 27 - Información del sistema 63
Ilustración 28 - Menú Emergente 63
Ilustración 29 - Menú Incidencias 64
Ilustración 30 - Lista de Incidencias 64
Ilustración 31 - Calendario 64
Ilustración 32 - Mensaje Modal 64
Ilustración 33 - Detalle Incidencia (Arriba) 65
Ilustración 34 - Detalle Incidencia (Abajo) 65
Ilustración 35 - Pantalla lectura código QR 66
Ilustración 36 - Pantalla Google Maps 66
Ilustración 37 - Error de Conexión 66
Ilustración 38 - Menú Almacén 67
Ilustración 39 - Detalle Almacén 67
Ilustración 40 - Lista Almacén 67
Ilustración 41 - Menú Mis Reservas 68
Ilustración 42 - Detalle Mis Reservas 68
Ilustración 43 - Lista Mis Reservas 68
Ilustración 44 - Lista Autorizaciones de Compra 69
Ilustración 45 - Selección Proveedor 69
Ilustración 46 - Menú Estadísticas 70

Ilustración 47 - Averías x Máquina *70*
Ilustración 48 - Ayuda *70*
Ilustración 49 - Componentes *72*
Ilustración 50 - Componentes *72*
Ilustración 51 - Botón acción código QR *Ilustración 52 - Lectura Código QR* *73*
Ilustración 53 - Desarrollo Menú Emergente para sincronización de datos *74*
Ilustración 54 - Ejemplo desarrollo de anotación por consola *75*
Ilustración 55 - Google Maps con posicionamiento de incidencias *75*
Ilustración 56 – Control cambio Idioma *Ilustración 57 – Idioma sesión* *76*
Ilustración 58 - Función cambio dinámico idioma Catalán *76*
Ilustración 59 - Control Modalidad Test *Ilustración 60 - Control Modalidad Test* *77*
Ilustración 61 - Entorno de desarrollo *78*
Ilustración 62 - Herramienta CASE ayuda al desarrollo y explotación de la base de datos *79*
Ilustración 63 - Estructura de un proyecto Worklight *80*
Ilustración 64 - Binarios autogenerados *81*
Ilustración 65 - Proyecto Android autogenerado *81*
Ilustración 66 - Ficheros del desarrollo *82*
Ilustración 67 - Desarrollo vista Lista de Incidencias *83*
Ilustración 68 - Implementación de Adaptador para acceso a entidad Incidencia *83*
Ilustración 69 - Ejemplo invocación Adaptador de acceso a entidad Usuario desde la vista *84*
Ilustración 70 - Desarrollo que invoca al método de la capa de modelo *84*
Ilustración 71 - Otros aspectos del desarrollo *85*

Tabla 1 - Calendario Proyecto *14*
Tabla 2- Detalle entrega PAC 1 – Plan de trabajo *15*
Tabla 3 - Detalle entrega PAC 2 – Análisis funcional, Diseño técnico y Prototipo *17*
Tabla 4 - Detalle entrega PAC 3 – Implementación *19*
Tabla 5 - Detalle entrega final *20*
Tabla 6 - Recursos Hardware para desarrollo *22*
Tabla 7 - Recursos Hardware de red *22*
Tabla 8 - Recursos de Software para los equipos de desarrollo y pruebas *23*
Tabla 9 - Recursos de Software para el equipo servidor *23*
Tabla 10 - Análisis de riesgos *29*
Tabla 11 - [CU00] Autentificación del Usuario *35*
Tabla 12 - [CU01] Ver Incidencias *36*
Tabla 13 - [CU02] Alta Incidencias *36*
Tabla 14 - [CU03] Modificación de Incidencia *37*
Tabla 15 - [CU04] Visualización de Calendario de Incidencias *37*
Tabla 16 - [CU05] Visualización de Material en Stock *38*
Tabla 17 - [CU06] Hacer Petición Material *39*
Tabla 18 - [CU07] Ver Peticiones de Reserva *39*
Tabla 19 - [CU08] Borrar Peticiones de Reserva *40*
Tabla 20 - [CU09] Modificar Peticiones de Reserva *40*
Tabla 21 - [CU10] Ver Números de Autorización de Compra *41*
Tabla 22 - [CU11] Alta de Números de Autorización de Compra *41*
Tabla 23 - [CU12] Modificación de Número de Autorización de *42*
Tabla 24 - [CU13] Borrar Números de Autorización de Compra *43*
Tabla 25 - [CU14] Ver Estadística *43*
Tabla 26 - [CU15] Ver Estadística *43*
Tabla 27 - [CU16] Sincronizar datos con el Servidor *44*
Tabla 28 - [CU17] Mostrar Mapa *44*
Tabla 29 - [CU18] Lectura Código QR *44*
Tabla 30 - Modelo Relacional de la Base de Datos *51*
Tabla 31 - Entidad Usuario *51*
Tabla 32 - Entidad Incidencia *52*
Tabla 33 - Entidad Equipo *52*
Tabla 34 - Entidad Material *53*
Tabla 35 - Entidad Reserva *53*
Tabla 36 - Entidad Autorización *54*

Tabla 37 - Entidad Proveedor *54*
Tabla 38 - Adaptadores Capa Modelo *56*
Tabla 39 - Adaptador de acceso a datos entidad Usuario *56*
Tabla 40 - Adaptador de acceso a datos entidad Incidencia *56*
Tabla 41 - Adaptador de acceso a datos entidad Equipo *57*
Tabla 42 - Adaptador de acceso a datos entidad Material *57*
Tabla 43 - Adaptador de acceso a datos entidad Reserva *57*
Tabla 44 - Adaptador de acceso a datos entidad Autorización *58*
Tabla 45 - Adaptador de acceso a datos entidad Proveedor *58*
Tabla 46 - Diagrama de Secuencia – Ver Estadística *59*
Tabla 47- Diagrama de Secuencia – Alta Incidencia *61*

1 Introducción

De cara a tener una mejor compresión del proyecto, en esta primera parte del documento daremos un vistazo general a los procesos que definen la actividad diaria de una empresa dedicada al mantenimiento de equipos de climatización e instalaciones calóricas. Es importante reseñar en este punto que, a pesar de que este proyecto se ha concebido con la idea de cubrir unas determinadas necesidades de una empresa de mantenimiento de equipos de calor y frío, puede ser aplicable a cualquier empresa dentro del mismo sector industrial, si no al 100% en muchos aspectos, sí en un porcentaje muy elevado de los mismos.

Más adelante analizaremos que motivaciones justifican el proyecto, así como, definiremos los objetivos generales del mismo.

Para finalizar y cubrir de principio a fin todos los aspectos generales de esta introducción, veremos la planificación en tiempos del proyecto, incluyendo los entregables en cada una de las fases, además de, enumerar los recursos y tecnologías necesarios para su correcta finalización. En dicho apartado adicionalmente se contemplarán y analizarán los riesgos con los que nos podemos encontrar a lo largo de la ejecución del proyecto.

1.1 Situación actual – Operativa funcional

A lo largo de este apartado intentaremos exponer cual es la situación actual de la operativa de trabajo del personal de campo y personal administrativo de una empresa dedicada al mantenimiento de equipos de frío y calor, así como, analizaremos los flujos de información que dicha actividad provoca. El objetivo es entender que es lo que está pasando y ver más adelante como las nuevas tecnologías pueden ayudar a mejorar dicha situación.

Al generarse una avería en un equipo de aire o calor, éste automáticamente genera un aviso a un sistema propietario del cliente de mantenimiento. Dicha avería es

procesada por un administrativo de la compañía mantenedora, mediante un portal de gestión propiedad del cliente. El citado portal permite al administrativo descargarse con periodicidad diaria, en formato de hoja de cálculo, un listado con todas las averías reportadas y no cerradas hasta el momento.

Mediante un proceso completamente manual, las averías se introducen en el sistema de la empresa mantenedora por el administrativo, desde donde serán listadas en agrupaciones por zona, que es repartido a diario a cada uno de los operarios o recursos asignados a la obra. Posteriormente dichos operarios ejecutarán las asistencias y resoluciones dentro de los tiempos de respuesta exigidos por el cliente. En el supuesto caso de que una avería no pueda ser asistida (que no necesariamente solventada), el cliente penalizará a final de mes con una cuantía económica a la empresa mantenedora.

El operario por su parte, se tiene que desplazar a las oficinas desde su punto de origen (sea su domicilio o la última avería asistida) a recoger el nuevo listado de incidencias a asistir, con las nuevas rutas. Normalmente este desplazamiento se hace a primera hora de la mañana. Como se ha dicho anteriormente es el personal administrativo quien se encarga de distribuir entre cada uno de los técnicos el listado de todas las incidencias de su zona.

Una vez el operario obtiene el listado, revisa y analiza las averías. Éste puede darse cuenta de que necesita de algún tipo de material, con lo que debe desplazarse al almacén para solicitar y recoger todo lo necesario para la correcta resolución de las incidencias. En el almacén el personal encargado debe proveer al técnico con todo lo requerido por éste. Puede pasar en este momento que, todo o parte del material requerido no se encuentre en stock y por lo tanto deba pedir una autorización de compra de material, o bien, que el personal de almacén pueda dispensar todo lo necesario para la corrección de la avería.

En el primero de los casos, es decir, cuando el material requerido por el personal de campo no se encuentra en stock, éste debe pedir una autorización de compra de material a uno de los proveedores autorizados de la compañía. Dicha autorización se la ha de dar el personal administrativo, generando para eso un número de pedido en el sistema. Actualmente dicha petición de autorización se realiza mediante una llamada telefónica, si el operario no se encuentra en las instalaciones de la compañía o bien físicamente dirigiéndose a un mostrador determinado para este menester, físicamente en la misma ubicación de la empresa. Una vez el operario tiene el número de pedido se puede desplazar a uno de los proveedores habituales a comprar la pieza o piezas necesarias.

Cuando el personal de campo, tiene todo lo necesario para poder solucionar una o varias averías se desplaza al lugar donde está la máquina o sistema averiado.

En este momento debe tener siempre en cuenta los equipos que están establecidos como críticos, y que necesitan de una asistencia en un tiempo máximo especificado por contrato. Es decir, en el momento de la recepción del listado, el

operario debe ser capaz 'in situ' de planificar su recorrido en función del tipo y criticidad de las incidencias.

Al desplazarse al lugar de la avería y finalizar o no ésta, el operario ha de reportar los trabajos ejecutados indicando el tiempo invertido, los materiales utilizados y la identificación del equipo atendido, mediante un parte de trabajo en formato de papel, para que así el administrativo de por concluida la resolución o asistencia de la avería y proceda al cierre del aviso.

Es importante indicar que, antes de realizar cualquier operación sobre el equipo, ha de comprobar que el código de máquina reportado en la incidencia y la máquina que va a asistir coinciden. En caso contrario debe informar del equívoco al personal administrativo de la compañía mantenedora.

Para el cierre de la citada incidencia, el administrativo ha de informar de forma manual en el sistema propio y del cliente los datos reportados en el parte. A partir de estos datos gestionados por el administrativo, el cliente es capaz de sacar unos reportes con indicadores de penalización en función de lo acordado previamente por contrato. Se considera que existe una penalización cuando una resolución de incidencia se hace por encima del tiempo máximo establecido para la máquina.

Como hemos podido observar a lo largo de la descripción del funcionamiento operativo de una empresa mantenedora, se producen muchos flujos de información entre los diferentes actores analizados, además la falta de homogeneidad en la información y en los medios de reporte, provocan una ineficiencia manifiesta en los procesos de negocio de la empresa. En el siguiente capítulo veremos como mitigar los efectos adversos de esta falta de homogeneidad.

Ilustración 1 - Desorden y falta de homogeneidad de la información

1.2 Justificación del Proyecto

La justificación de negocio para arrancar este proyecto es en forma de resumen; la necesidad de mejorar y optimizar los flujos de información entre el personal de campo y el personal administrativo. Podemos citar los siguientes efectos positivos como resultado de dicha optimización:

- Disminuir los errores en la introducción de los datos en los sistemas. Merece la pena recordar que dichas entradas de datos se realizan de una forma completamente manual en muchos de los casos.
- Adquirir mayor agilidad en los procesos de comunicación internos de la empresa.
- Mejorar la planificación y organización de las tareas diarias por parte del personal de campo.
- Disminuir los costes asociados a los desplazamientos al centro de trabajo por parte del operario, para recoger material o simplemente para obtener el listado de incidencias.
- Optimizar la cadena de suministro, dado que el personal de almacén puede planificar con más eficacia las necesidades en función de la demanda de material y el stock del mismo.
- Exponer en concursos públicos capacidades pioneras en el sector, que pueden ser factores determinantes a la hora de ganar licitaciones.
- Disminuir los tiempos de respuesta ante una avería y por lo tanto reducir el riesgo de sanción asociado a las máquinas con criticidad elevada.

Ilustración 2 - Información ordenada

Es importante reseñar que añadiendo capacidades de movilidad a los procesos de la empresa, estamos incrementando el valor de los mismos y por lo tanto extendiéndolos. En ningún caso se plantea este proyecto como una mera sustitución de unos procesos poco automatizados, por otros que los están mucho más, sino que el valor real del mismo es, toda la funcionalidad que actualmente ya tenemos implementada, más las nuevas capacidades que aporta la tecnología dentro del ámbito de movilidad a los procesos de negocio de la compañía.

Ilustración 3 - Operario trabajando con dispositivo móvil

El objetivo general de este proyecto es presentar una solución híbrida para dispositivos móviles con tecnología **Android**[1]**,** que permita extender y mejorar las funcionalidades de la operativa diaria del personal de campo de una empresa de mantenimiento de equipos de frío y calor. Concretamente lo que se pretende es automatizar gran parte de los intercambios de flujos de información, que en la actualidad se realizan de una forma manual, mediante diferentes soportes (papel, hojas de cálculo, llamadas telefónicas, envío de correos electrónicos, etc..), proveyendo así al sistema de una mayor homogeneidad, de una mayor tolerancia a fallos y fiabilidad ante posibles errores humanos de reporte y dé una mayor eficacia en las asistencias de los operarios, aspecto éste último de vital importancia, ya que se estiman las pérdidas de la empresa por penalizaciones en asistencia críticas, como muy elevadas.

A nivel personal, con este proyecto pretendo obtener mayores conocimientos en un producto propietario de IBM llamado **IBM Worklight**[2]. La elección de esta tecnología por mi parte está basada en dos aspectos fundamentales; en primer lugar, es una tecnología con la que convivo a diario por mi tipo de trabajo, siempre desde un punto de vista más de consultoría que de desarrollo, por lo tanto considero que puede ayudar a enriquecer mi perfil laboral, y por último, la baja curva de aprendizaje que presenta el producto, al estar éste basado en tecnologías muy comunes hoy en día en el desarrollo de plataformas web y con un alto grado de integración entre todos sus componentes.

Funcionalidades principales

3

Podemos agrupar las funcionalidades de la aplicación en cinco grandes categorías, cada una de las cuales presentará las siguientes características funcionales:

- **Autentificación** del usuario a través de formulario de entrada para acceso a las funcionalidades del aplicativo, con nivel de seguridad de usuario / contraseña. Los usuarios han de estar previamente registrados en el sistema por parte del administrador del mismo. No se contempla la posibilidad de auto-registro del usuario.

- **Incidencias y Preventivos:**

  Permitirá tanto la apertura por parte del operario de incidencias no reportadas en el sistema, así como, el cierre de las mismas con éxito o no. La información de las averías reportadas se presentará en una lista agrupada por nivel de criticidad. Se dotará a la aplicación de capacidad de lectura de **códigos QR**[3], con los cuales, el operario podrá identificar de una manera rápida y fácil el aparato erróneo. Finalmente se contempla la posibilidad de incorporar capacidades de posicionamiento GPS, permitiendo esto la localización en un mapa de las ubicaciones de las averías.

- **Almacén**:

  Permitirá al operario hacer la pre-reserva de material al almacén. Además ha de permitir al personal de campo anular la misma en caso de equívoco. La información se presentará agrupada por categorías, y para cada categoría en una lista de tipos de material y su stock en almacén. Las categorías serán una lista finita y prefijada por el administrador del sistema. Cualquier tipo de material que no encaje en ninguna de las categorías prefijadas, se incluirá en la categoría de carácter general “Otros”.

- **Autorizaciones de compra**:

  Permitirá al operario pedir una autorización de compra de material en el caso de que éste no se encuentre en stock en el almacén. En todo momento el operario podrá comprobar el estado de todas sus peticiones de autorización, presentándose dicha información en formato de lista. Desde esta misma lista se contempla la posibilidad de anular la petición y una vez el personal administrativo ha asignado el proveedor, el cambio del mismo por imposibilidad de compra del material por los motivos que fuesen (por ejemplo, porque el proveedor no tiene el material requerido en ese momento).

- **Estadísticas**:

  El operario podrá ver en una gráfica de las máquinas críticas que más averías han reportado en los últimos 3 meses. Esto permitirá al técnico de campo informar al jefe de obra de que dichas máquinas presentan errores, posiblemente estructurales de otro nivel. Es decir, se podría dar el caso de que una máquina se averíe constantemente debido a fuerzas externas (por ejemplo, inclusión de equipos en una sala con disipación muy alta de calor) y no directamente a algo relacionado con el aparato.

Calendario del proyecto

4

Siguiendo la programación especificada por el consultor, en esta lista se reflejan todos los hitos marcados para este semestre, así como, la documentación que en cada uno de ellos se ha de entregar:

Fecha de Entrega	Hito	Entregable
11 de Marzo	PAC 1	Plan de Trabajo
08 de Abril	PAC 2	Análisis funcional, Diseño técnico y Prototipo
20 de Mayo	PAC 3	Implementación
10 de Junio	Entrega Final	Memoria y video de presentación de proyecto

Tabla 1 - Calendario Proyecto

4.1 Calendario

Se ha tenido en cuenta como fecha de inicio del proyecto el **28 de Febrero** fecha en la cual se abría el cuatrimestre**,** y como fecha final del mismo el **10 de Junio**, fecha en la que se prevé la última entrega.

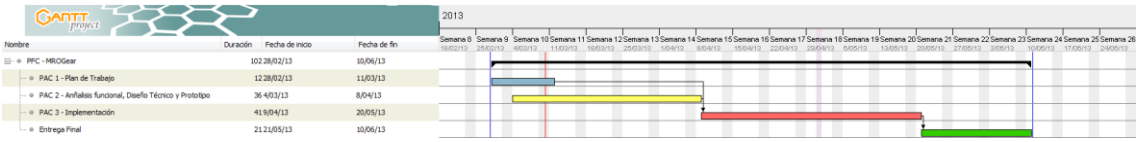


Ilustración 4 - Calendario general del proyecto

Disponemos de 102 días para completar todas las fases del proyecto. Se ha estimado que cada jornada normal de trabajo es de 3 horas (laborables no festivos) y que cada jornada festiva, incluyendo fines de semana, la estimación de horas a dedicar al proyecto será de unas 10 horas por jornada.

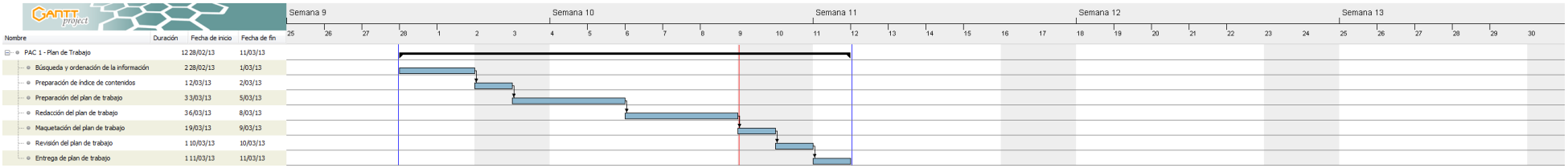


Ilustración 5 - Diagrama de Gantt detalle entrega PAC 1 – Plan de trabajo

Id	Tarea	Descripción
1	Búsqueda y ordenación de la información	Definición del proyecto a nivel general y búsqueda de información sobre las tecnologías a usar durante la elaboración del mismo. La información se clasifica para tener un acceso más rápido a la hora de la elaboración del documento.
2	Preparación del índice de contenidos	Una vez estructurada toda la información se procede a la elaboración del índice o esquema general del documento. Se diseñan los apartados y sub-apartados.
3	Preparación del plan de trabajo	Definición y filtrado de la información necesaria previa elaboración del documento a entregar.
4	Redacción del plan de trabajo	Elaboración del documento final siguiendo las pautas marcadas en la tarea 2 y 3.
5	Maquetación del plan de trabajo	Montaje del plan de trabajo. En esta fase se definen las imágenes que ha de contener el documento, se da formato a las páginas, cabecera y pie de página. Además se le aplica estilo al tipo de letra. Además se establecen índices y portada.
6	Revisión del plan de trabajo	Revisión del documento a entregar
7	Entrega del plan de trabajo	Entrega documentación

Tabla 2- Detalle entrega PAC 1 – Plan de trabajo

4.3 Detalle de entrega de PAC 2 – Análisis funcional, Diseño técnico y Prototipo

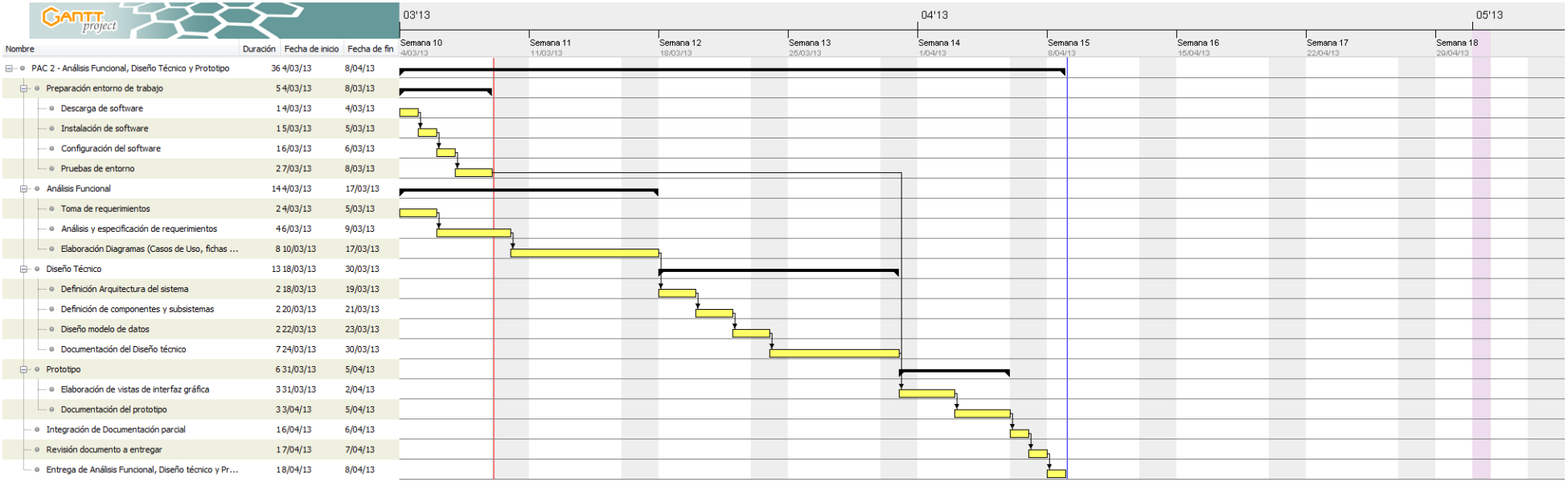


Ilustración 6 - Diagrama de Gantt detalle entrega PAC 2 – Análisis funcional, Diseño técnico y Prototipo

Id	Tarea	Descripción
	Preparación entorno de trabajo	En esta fase se contemplan todas las tareas necesarias para la preparación del entorno de trabajo
1	Descarga del software	Descarga del software necesario para la correcta configuración y ejecución del proyecto (todas las descargas se realizan desde las páginas oficiales de cada uno de los productos): **Eclipse, SDK Android, Plugin ADT, MySQL, IBM Worklight Studio, IBM Worklight Device Runtime, Driver JDBC MySQL, Microsoft Office 2012, GattProject, IBM Data Studio, DropBox.**
2	Instalación software	Instalación de todo el software necesario para la implementación y finalización de la aplicación, tanto para la parte de desarrollo, como la de copia de seguridad, como para la parte servidora.
3	Configuración del software	Configuración de los componentes de desarrollo; **SDK Android, Driver JDBC MySQL**

4	Pruebas de entorno	Realización de pruebas mínimas para chequeo básico de conectividad, entre dispositivo-servidor y entre servidor de aplicaciones-base de datos.
	Análisis Funcional	En esta fase se contemplan todas las tareas necesarias para la preparación del análisis funcional del aplicativo.
5	Toma de requerimientos	Identificación de las funcionalidades de la aplicación. Recogida de los requerimientos en documento informal no entregable.
6	Análisis y especificaciones de los requerimientos	Definición de las funcionalidades definitivas
7	Elaboración de Diagramas (Casos de Uso UML, fichas CRC, Diagramas de interacción UML)	Definición de actores, elaboración de casos de uso UML, definición de fichas CRC y diagramas de interacción UML.
	Diseño Técnico	En esta fase se contemplan todas las tareas necesarias para la preparación y elaboración del diseño técnico.
8	Definición de la arquitectura del sistema	Identificación de todos los componentes de la arquitectura a nivel lógico, físico, componentes y de red.
9	Definición de componentes y subsistemas	Definición y agrupación de los componentes en subsistemas
10	Diseño de modelo de datos	Elaboración de diagrama relacional de base de datos.
11	Documentación del diseño técnico	Elaboración del documento del diseño técnico.
	Prototipo	En esta fase se contemplan todas las tareas necesarias para la elaboración de un prototipo de la parte de plataforma móvil.
12	Elaboración de vistas de interfaz gráfica	Elaboración de prototipo con los componentes de vista de interfaz gráfica y transiciones, pero sin implementación de lógica de negocio, ni validaciones.
13	Documentación del prototipo	Elaboración del documento de prototipo.
14	Integración de Documentación parcial	Integración en un sólo documento de toda la documentación generada en las tareas previas 7, 11 y 13.
15	Revisión documento a entregar	Revisión final del documento a entregar.
16	Entrega de Análisis Funcional, Diseño Técnico y Prototipo	Entrega documentación.

Tabla 3 - Detalle entrega PAC 2 – Análisis funcional, Diseño técnico y Prototipo

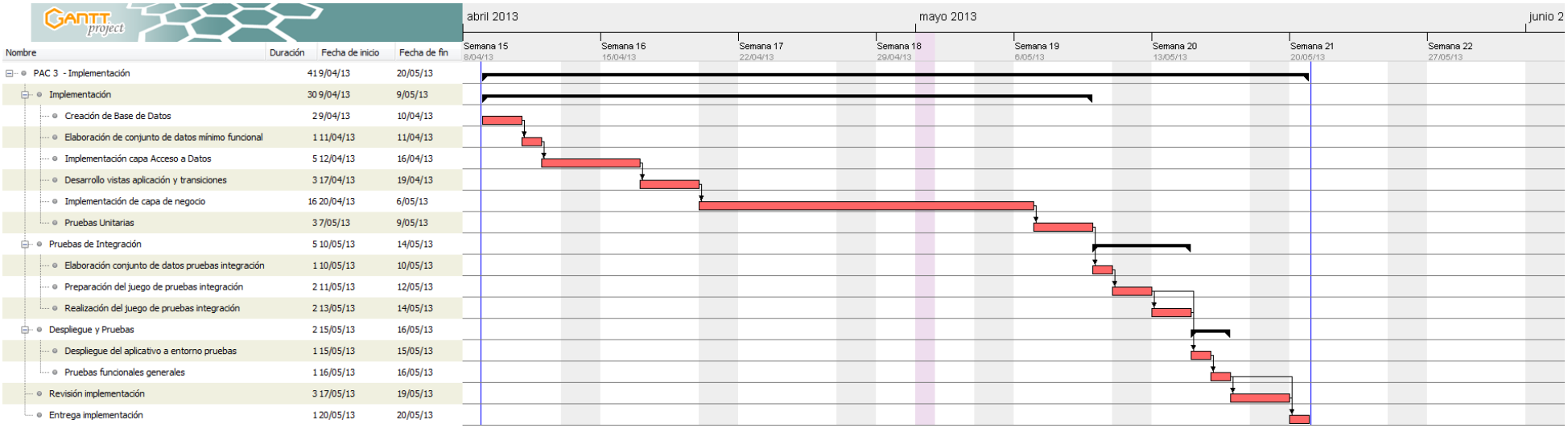


Ilustración 7 - Diagrama de Gantt detalle entrega PAC 3 – Implementación

Id	Tarea	Descripción
	Implementación	En esta fase se contemplan todas las tareas necesarias para la implementación del aplicativo móvil y de la parte servidora.
1	Creación de la Base de Datos	Creación de la bases de datos a nivel físico.
2	Elaboración de conjunto de datos mínimo funcional	Creación de un *script* SQL de inserción de datos mínimos funcionales para arrancar el aplicativo.
3	Implementación capa de Acceso a Datos	Implementación de acceso a datos a través de adaptadores SQL de IBM Worklight.
4	Desarrollo vistas aplicación y transiciones	Elaboración de la interfaz gráfica, intentando aprovechar al máximo las vistas definidas en el prototipo. En este punto se deben implementar las reglas de transición entre pantallas y las validaciones básicas en los formularios de entrada de datos.

5	Implementación de la capa de negocio	Elaboración de las reglas y lógica de negocio. Dichas reglas deben contemplar validaciones que no se hayan tenido en cuenta en el apartado anterior.
6	Pruebas Unitarias	Realización de pruebas unitarias
	Pruebas de integración	En esta fase se contemplan todas las tareas referentes a la realización de pruebas de integración.
7	Elaboración de conjunto de datos pruebas de integración	Creación de un *script* SQL de inserción de datos, teniendo en cuenta las restricciones relacionales para arrancar las pruebas de integración.
8	Preparación del conjunto de pruebas de integración	Documentación de las pruebas realizadas.
9	Realización de pruebas de integración	Realización de las pruebas de integración de las tres capas descritas con anterioridad (acceso de datos, vista y lógica de negocio), anotando los resultados para compararlos en posteriores ejecuciones de las baterías de prueba (normalmente tras realizar modificaciones al aplicativo).
	Despliegue y Pruebas	En esta fase se contemplan todas las tareas referentes a la realización de pruebas generales y de despliegue.
10	Despliegue del aplicativo a un entorno de pruebas	Despliegue del aplicativo a un entorno de pruebas que debe simular al máximo el entorno real de la compañía.
11	Pruebas funcionales generales	Realización de pruebas de carácter general, donde además de probar el desarrollo y el despliegue del *app*, se deben realizar pruebas integrales de infraestructura.
12	Revisión de la implementación	Realizar modificaciones de última hora de errores surgidos durante las pruebas.
13	Entrega implementación	Entrega código fuente y ejecutable.

Tabla 4 - Detalle entrega PAC 3 – Implementación

Se ha decidido seguir un ciclo de desarrollo clásico o en cascada. Por otro lado, y a pesar de que, inicialmente se ha dividido el proyecto en subsistemas funcionales, a la hora de la implementación no se han tenido en cuenta ya que el equipo de desarrollo está compuesto por una única persona. En su lugar, se ha seguido una aproximación de desarrollo por capas; Acceso de datos, capa de vista de interfaz gráfica y lógica de negocio.

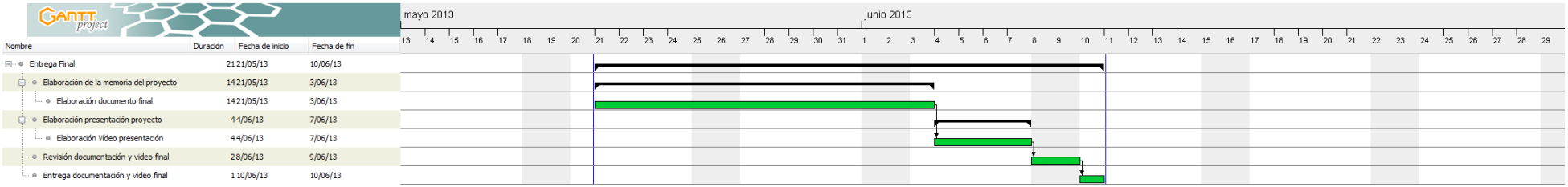


Ilustración 8 - Diagrama de Gantt detalle entrega final

Id	Tarea	Descripción
	Elaboración de memoria del proyecto	En esta fase se contemplan todas las tareas necesarias para la elaboración de la memoria del proyecto.
1	Elaboración de documento final	Elaboración de la memoria final de proyecto.
	Elaboración de presentación de proyecto	En esta fase se contemplan todas las tareas necesarias para la elaboración de la presentación del proyecto.
2	Elaboración de Vídeo de presentación	Elaboración del vídeo de presentación.
3	Revisión documentación y vídeo final	Revisión final de la documentación y vídeo a entregar.
4	Entrega documentación y vídeo final	Entrega documentación y vídeo.

Tabla 5 - Detalle entrega final

Para poder desarrollar con normalidad el proyecto que se plantea en este documento, se requieren los siguientes componentes de infraestructura física:

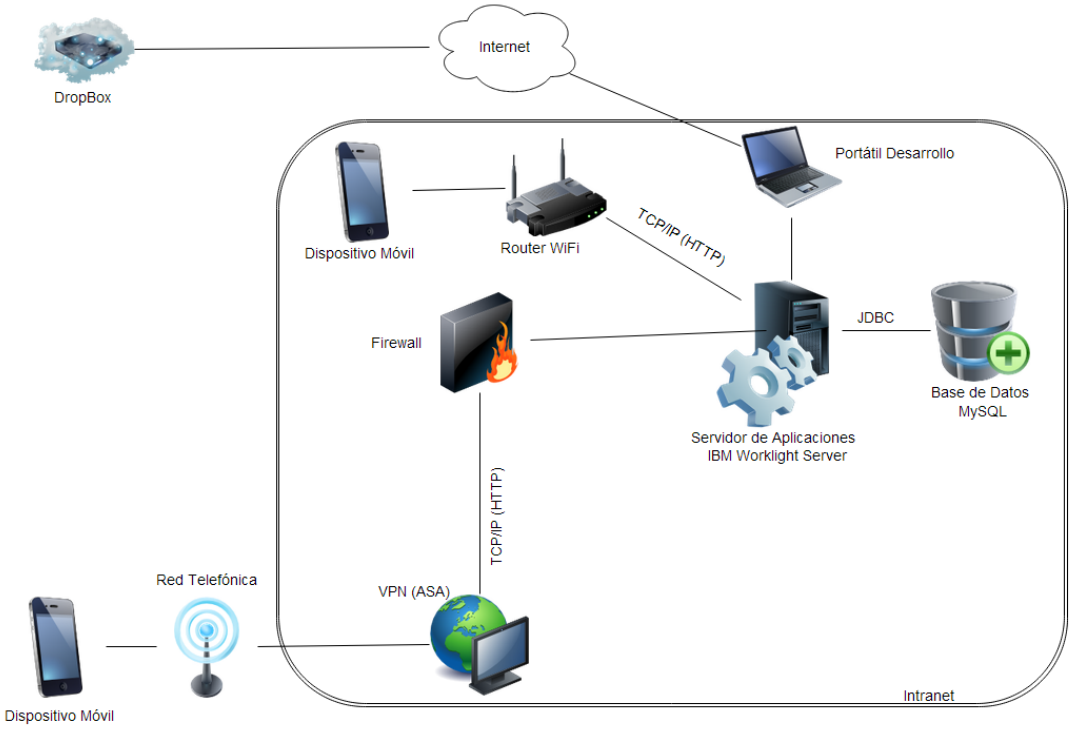


Ilustración 9 – Diagrama de arquitectura general de la solución

Un servidor de aplicaciones en una red privada de la compañía recibe las peticiones de acceso de datos que provienen del dispositivo móvil. Dicho servidor se conecta a un sistema gestor de base de datos **MySQL**[4] y recupera los datos a través del *driver* **JDBC**[5] específico. Tras el establecimiento de la conexión de bases de datos, el servidor retorna los datos al dispositivo móvil a través de la conexión inalámbrica o a través de la red de datos del operador de telefonía.

Es importante reseñar que, si el operario está fuera del alcance de los puntos de acceso privados de la compañía, el uso de una infraestructura **VPN**[6] de cliente y servidor será necesario. Dicha conexión VPN, proveerá al sistema de una capa de seguridad adicional y de conectividad para la parte cliente del sistema (***app*** instalada en el móvil), cuando la conexión no pueda realizarse directamente a través de los puntos de acceso privados situados en la intranet.

5.1 Recursos de Hardware

A continuación detallamos los recursos físicos y las características técnicas de los mismos, que serán necesarios para la implementación y pruebas del proyecto. Dividiremos los componentes de hardware en dos categorías; Desarrollo y *Red*:

Desarrollo

Componente	Características técnicas	Función
Ordenador de desarrollo	**CPU**: Intel Core i7 CPU Q720 @ 1.60GHz **Memoria**: 12GB RAM (DDR3 SDRAM) **Disco**: ATA Device con capacidad 136GB **Sistema Operativo**: Windows 7 Ultimate	• Tareas de desarrollo. • Tareas ofimáticas. • Tareas de pruebas.
Servidor	**CPU**: AMD Athlon 64 Processor 3200+ 2.01 GHz **Memoria**: 6GB RAM (DDR3 SDRAM) **Disco**: ATA Device con capacidad 136GB **Sistema Operativo**: Windows 7 Professional – SP 1	• Hosting de la parte servidora, incluyendo seguridad y acceso a datos.
Dispositivo Móvil	**Modelo**: Google Nexus 4 **CPU**: Procesador de cuatro núcleos Qualcomm Snapdragon S4 Pro a 1,5 Ghz **Memoria**: 2GB LPDDR2 **Sistema Operativo**: Android 4.2.2 (Jelly Bean)	• Pruebas y depuración de errores.

Tabla 6 - Recursos Hardware para desarrollo

Red

Componente	Características técnicas	Función
Router + Firewall + HUB	**Modelo**: Huawei EchoLife Home Gateway	• Proveer de conectividad al sistema en un ámbito local.
VPN	**Modelo**: Cisco Secure Remote Access Cisco ASA 5500 Series SSL/Ipsec VPN Edition	• Proveer de conectividad al sistema fuera del ámbito local de la empresa. Es necesario el uso de un cliente VPN en los dispositivos móviles.

Tabla 7 - Recursos Hardware de red

5.2 Recursos de Software para los equipos de desarrollo y pruebas

A continuación detallamos los recursos de software necesarios para la fase de desarrollo y que estarán convenientemente instalados en el ordenador de desarrollo:

Progama	Función
Eclipse IDE	- Entorno de desarrollo para aplicativos basados en **Java**[8] **http://eclipse.org/downloads/**
Android SDK	- *Kit* de desarrollo para aplicaciones Android **http://developer.android.com/sdk/index.html**

Plugin ADT	- Plugin Android para entorno de desarrollo Eclipse **http://developer.android.com/tools/sdk/eclipse-adt.html**
IBM Data Studio	- Herramienta *CASE* utilizada para la creación de los ficheros *DDL* y mantenimiento de datos del sistema gestor de bases de datos MySQL. **http://www.ibm.com/developerworks/downloads/im/data/**
Magic Draw UML	- Herramienta para la elaboración de diagramas UML, usada principalmente en la fase de siseño. **http://magicdraw.com/download_patch**
Microsoft Office 2010	- Herramientas ofimáticas, principalmente usadas para la elaboración del documentación técnica del proyecto. **http://office.microsoft.com/es-es/**
DropBox	- Repositorio en la nube, principalmente usado para mantener una copia de seguridad del espacio de trabajo donde se realiza el desarrollo. **https://www.dropbox.com**
DiagramLY	- Herramienta ofimática en línea usada para la elaboración de determinados esquemas técnicos. **https://www.draw.io/**
IBM Worklight Studio	- *Plugin* Worklight para entorno de desarrollo Eclipse. Permite la implementación sencilla de apps híbridas, basadas en HTML 5, CSS 3 y JavaScript. **http://www-01.ibm.com/software/mobile-solutions/worklight/components/mobile-development-platform/**
IBM Worklight Device Runtime	- Componentes de runtime Woklight incluidos automáticamente en la app desplegada en el dispositivo móvil que provee conectividad con el servidor. **http://www-01.ibm.com/software/mobile-solutions/worklight/components/runtime/**
GanttProject	- *Herramienta para la elaboración de diagramas de gantt, utilizado principalmente en la elaboración del plan de proyecto.* **http://www.ganttproject.biz/**

Tabla 8 - Recursos de Software para los equipos de desarrollo y pruebas

5.3 Recursos de Software para el equipo servidor

A continuación detallamos los recursos de software necesarios para la fase de desarrollo e implementación final del sistema que estarán convenientemente instalados en la parte servidora:

Progama	Función
IBM Worklight Server	- Parte servidora del sistema. Instalado sobre un servidor de aplicaciones. Provee conectividad con el sistema gestor de base de datos y seguridad. **http://www-01.ibm.com/software/mobile-solutions/worklight/components/server/**
MySQL	- Sitema Gestor de Bases de Datos. **http://www.mysql.com/**

Tabla 9 - Recursos de Software para el equipo servidor

Como ya se citó en el capítulo de objetivos, una de las finalidades de este proyecto es la elaboración de una aplicación para dispositivo móvil, también conocidas como *apps,* con tecnología híbrida. En este apartado, se intentará aclarar que tecnologías involucra el desarrollo de una aplicación de este estilo, que ventajas y que desventajas conllevan, en contraposición a las aplicaciones nativas para la plataforma **Android** y las puramente web.

Más adelante se presentará las tecnologías que se usarán en el proyecto, tanto para la parte cliente (instalada en el dispositivo móvil), como la parte servidora y de *red.*

6.1 Tecnología híbrida

Podemos diferenciar entre tres tipos de apps en el mercado: las aplicaciones nativas, las aplicaciones puramente web y las aplicaciones híbridas. Es este apartado se dará una definición de aplicación híbrida comparando los rasgos característicos de cada uno de los tres tipos. Finalmente veremos, que razones nos han conducido a elegir este tipo de aplicación para la elaboración de nuestro proyecto.

Las aplicaciones puramente web, son aquellas donde el único requerimiento para la ejecución en el dispositivo móvil es la presencia de un navegador. Es decir, son aplicaciones que perfectamente pueden funcionar en cualquier navegador móvil. Por citar alguna de las ventajas del uso de este tipo de tecnología, podríamos decir que, la más destacada sería la facilidad de despliegue, por dos motivos básicos; primero despliegue centralizado (una sola vez en el servidor web de la compañía) y en segundo el uso de lenguajes ampliamente extendidos en el mercado, tales como HTML 5, CSS, JavaScript. Sin embargo, presentan dos claras desventajas; una es la limitada capacidad de acceso a los '*Widgets'* del dispositivo (GPS, Acelerómetro, Cámara, etc...) y la otra el aspecto del aplicativo y transiciones entre vistas, muchas veces muy diferentes a los componentes gráficos nativos. Esto produce al usuario una sensación de poca cohesión y una sensación de falta de suavidad en las transiciones.

Por contra, las aplicaciones nativas son aquellas que están desarrolladas con la SDK específica de la plataforma. Son aplicaciones con acceso a todas las capacidades del dispositivo, de rendimiento muy elevado y con una cohesión visual muy alta. La

principal desventaja que presentan, es el elevado coste inicial de la curva de aprendizaje, especialmente si estamos pensando en aplicaciones que deban ser desplegadas en diferentes plataformas (**Android**, **iOS**, **RIM, W8 Mobile**, etc...). Dada la gran variedad de estas en el mercado de hoy en día, y debido al incremento del fenómeno conocido como **BYOD**[9], acrónimo del término Inglés "Bring Your Own Device", se puede incurrir fácilmente en un incremento de los costes generales del proyecto, debido a la escasez de personal habilitado en todas estas tecnologías a la vez.

Las aplicaciones híbridas son aquellas que, son capaces de contener lo mejor de cada mundo, es decir tienen acceso completo a las capacidades del dispositivo y utilizan lenguajes de programación propios de entornos web. Como en el caso de las aplicaciones nativas, es requerida la instalación de una '*shell*' nativa en el dispositivo móvil con capacidades de interpretación de HTML 5, CSS 3 y JavaScript.

Una vez explicadas las diferencias entre cada tipo de aplicación, la elección por mi parte del tipo híbrido ha sido debido a; en primer lugar el corto espacio de tiempo para el desarrollo, por lo tanto cualquier ahorro de tiempo (en este caso en la curva de aprendizaje de SDK para **Android**) es bien valorado, y en segundo lugar, por congeniar la vida laboral con la formativa.

Ilustración 10 - Comparativa tipos app

6.2 Tecnología de la parte cliente. Dispositivo móvil

Como ya se ha explicado con anterioridad, el uso de una tecnología híbrida para el desarrollo de *apps*, conlleva la instalación en el dispositivo móvil de una '*shell*' nativa

a la plataforma. Dicha '*shell*' la provee **IBM Worklight**, incluyéndola en su producto **IBM Worklight Runtime** para dispositivos móviles y tabletas.

El desarrollo de la capa visual se realizará íntegramente en HTML 5, CSS 3 y JavaScript. Dicha capa se debe comunicar con la capa de negocio a través de adaptadores SQL. El servidor, por su parte, será el encargado de ejecutar las consultas y transferir los datos al dispositivo en formato **JSON**[10].

Para que la experiencia del usuario sea lo más suave posible, el proyecto contendrá el '*framework*' de desarrollo JavaScript **Dojo Mobile**[11]**.** Esto dotará al aplicativo de unos componentes visuales muy similares a los nativos de la plataforma, así como de unas transiciones y efectos visuales de muy alto nivel.

Cualquier acceso requerido por parte de la *app,* a las capacidades del móvil (GPS, cámara para la lectura de códigos QR, etc...) será realizada a través de una tecnología conocida como **Apache Cordova o PhoneGap**[12]**.**

Finalmente, para el posicionamiento geográfico utilizaremos las capacidades de **Google Maps** *en conjunción con las capacidades* ***GPS*** *del dispositivo móvil***.**

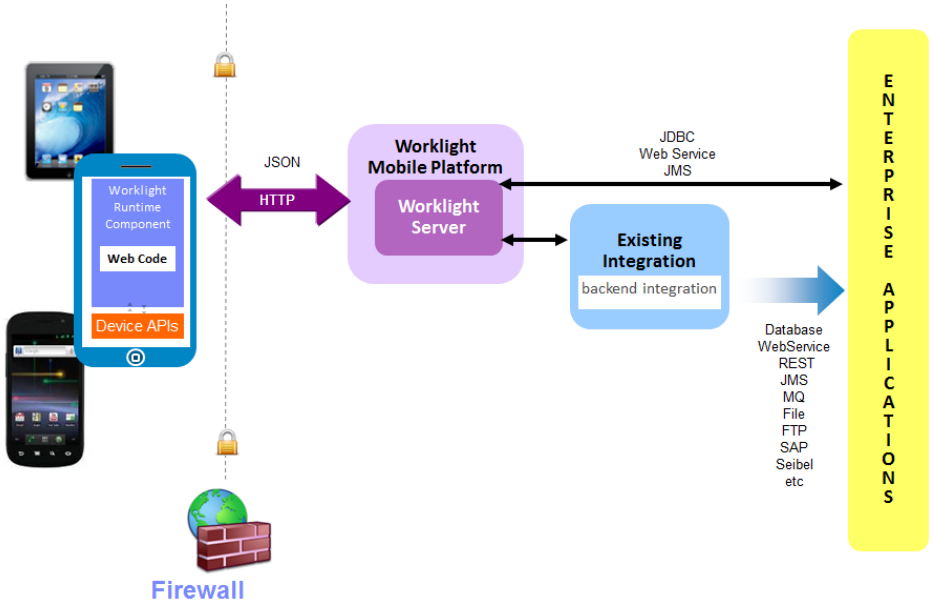


Ilustración 11 - Diagrama de acceso a datos por parte del dispositivo móvil

6.3 Tecnología de la parte servidora

La parte servidora estará compuesta por un servidor de aplicaciones basado en Java, donde se desplegará el servidor **IBM Worklight Server**. Éste será el encargado de proveer al sistema con funcionalidades de seguridad (módulo de acceso) y funcionalidades de conectividad de datos.

La conectividad con la capa de modelo, desplegada en un sistema gestor de bases de datos **MySQL,** se realizará a través del conector o *driver* JDBC nativo correspondiente.

Para habilitar la conexión entre dispositivo y servidor, usaremos los adaptadores Worklight. Dicha tecnología, basada en JavaScript, permite al servidor de aplicaciones (recordamos que el servidor de aplicaciones es Java 100%) la ejecución de código *script* como si de Java nativo se tratara. El servidor de aplicaciones se apoya en el uso de una implementación JavaScript escrita íntegramente en Java llamada **Rhino**[13] para estos menesteres.

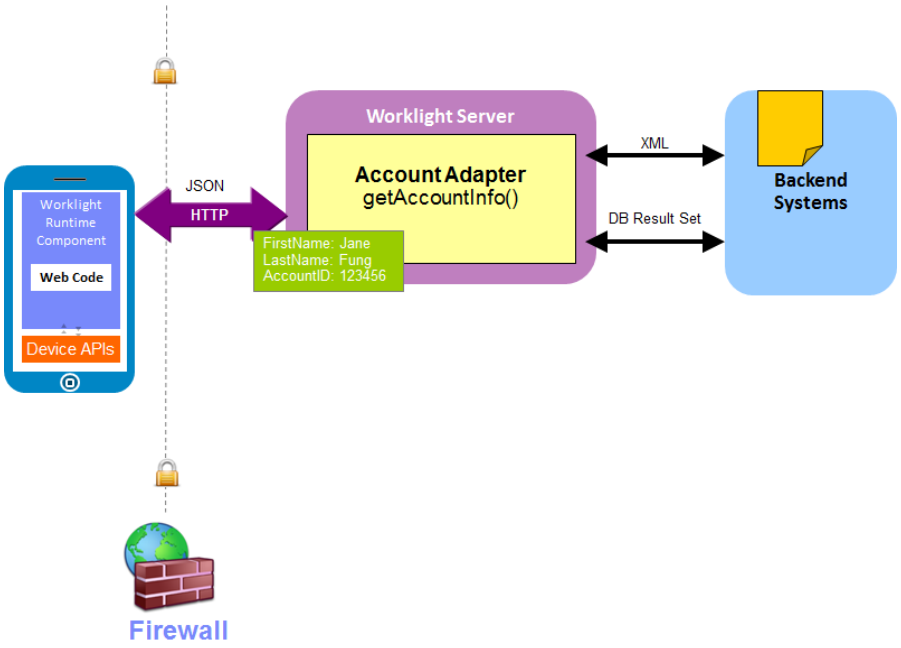


Ilustración 12 - Arquitectura general de un adaptador Worklight para acceso a datos

6.4 Tecnología de red

La comunicación del dispositivo móvil con el servidor de aplicaciones se realizará utilizando el protocolo TCP/IP HTTP, a través de los canales *WiFi* en el caso de que el operario se encuentra en las instalaciones de la compañía, o bien, a través de la red de datos de la operadora de telefonía contratada por éste. En el segundo de los casos y debido a que el servidor de aplicaciones estará dentro de un rango de IP privado, la instalación de un cliente VPN será necesaria para cualquier comunicación de datos con el servidor de aplicaciones.

Análisis de Riesgos

7

En este capítulo cubriremos los riesgos con los que nos podemos encontrar durante la elaboración del proyecto. De la misma forma, daremos unas soluciones a los mismos que ayuden a mitigar su posible efecto negativo sobre la correcta finalización del proyecto.

Riesgo	Descripción	Probabilidad de Aparición	Impacto	Acciones mitigantes
Falta de conocimientos en la plataforma de desarrollo IBM Worklight, Apache Cordova, Dojo, etc…	A pesar de que mi trabajo está muy relacionado con la tecnología que se va a usar en el proyecto, siempre desde una perspectiva de consultor y nunca desde la perspectiva de desarrollador, podemos prever algunas dificultades derivadas de esta falta de conocimiento.	**Media**	**Alto**	• Lectura de documentación especializada. • Realización de curso y tutoriales en línea.
Enfermedad del desarrollador	Durante el desarrollo del proceso, en cualquier de las fases en las que se encuentre, la persona que realiza el proyecto puede ponerse enferma. Esto sin duda, provocará desviaciones en los tiempos del mismo.	**Media/Baja**	**Medio**	• Se deberán tomar medidas compensatorias encaradas a la recuperación de tiempos tras la recuperación del enfermo.
Avería de los equipos de desarrollo y despliegue	Durante el desarrollo del proyecto se produce una avería de los equipos, el impacto sobre el mismo puede tener consecuencias nefastas, que derivarán en incumplimientos en los tiempos marcados.	**Media/Baja**	**Alto**	• Cambio de los equipos averiados de forma inmediata, en la medida en que sea posible. • Copias de seguridad diaria de la parte de desarrollo • Copia de seguridad diaria de los datos.

Mal funcionamiento o escasez de recursos de infraestructura	Durante el desarrollo o despliegue del proyecto se observan carencias en los recursos de hardware que pueden producir retrasos considerables en los tiempos.	**Media**	**Alto**	• Cambio de los equipos averiados de forma inmediata, en la medida en que sea posible, o adición de recursos extras.
Conciliación con la vida laboral	Durante la realización del proyecto la persona asignada a las tareas de desarrollo puede viajar o incluso extender su horario por encima de lo habitual	**Alta**	**Medio**	• Réplica del entorno de desarrollo en portátil, utilizando tecnologías de virtualización por comodidad. • Medidas compensatorias encaradas a la recuperación de tiempos deberán ser tomadas tras la recuperación del enfermo. • Re planificación de la fase del proyecto en la que se encuentre para calibrar mejor el impacto.
Planificación inadecuada del proyecto	Durante el desarrollo del proyecto se pueden observar deficiencias en la planificación de los tiempos asignados a cada uno de los hitos. Esto provocará una desviación que redundará en una imposibilidad de completar el mismo en la fecha marcada.	**Media**	**Alto**	• Re planificación de la fase del proyecto que se vea afectada por este hecho. • Aplicación de medidas compensatorias de cara a recuperar tiempos. • Planificación pesimista (peor de los escenarios). Esto puede ayudar a mitigar errores iniciales. • A pesar de que el equipo de desarrollo tan solo está formada por una persona, deben realizarse chequeos periódicos para ver las desviaciones.

Tabla 10 - Análisis de riesgos

Análisis Funcional

8

En esta sección elaboraremos y desarrollaremos todos los aspectos referentes a los requisitos funcionales y no funcionales de nuestro aplicativo, a través de, la especificación de casos de uso y de los actores que intervienen en los mismos. Esto nos ayudará a tener, una visión más completa de los objetivos del proyecto de cara a afrontar las siguientes fases con mayor garantía de éxito.

8.1 Requerimientos Funcionales

En este apartado se cubrirán los requerimientos funcionales del aplicativo, intentando ser lo menos ambiguos posible. En una primera fase se usará el lenguaje natural para explicar a nivel general los casos de uso, por la contra, en una segunda fase se usará una notación UML para mostrar más al detalle las interacciones con el entorno, los estados y la evolución de cada uno de los citados casos.

8.1.1 Descripción Básica del Funcionamiento del aplicativo

Como ya comentamos con anterioridad el objetivo general de MROGear es cubrir las necesidades básicas diarias, en cuanto a los flujos de información del personal de campo se refiere, de una empresa dedicada al mantenimiento de aparatos de frío y calor. Para ello establecemos las siguientes funcionalidades:

1. Incidencias: Desde este apartado el operario podrá ver las incidencias que tiene asignadas, así como, informar al sistema de la correcta resolución de las mismas. Adicionalmente se le permitirá introducir incidencias en el sistema, así como, se le proveerá con una vista tipo calendario para que pueda hacer una mejor planificación semanal de su trabajo.

2. Almacén: Como ya se comentó con anterioridad, uno de los objetivos del proyecto es proveer de una mayor eficiencia a los flujos de información de la compañía. En este apartado, un operario podrá con antelación hacer una pre-reserva de materiales, siempre y cuando estos se encuentren en stock. Del mismo modo, podrá ver y anular las pre-reservas de materiales.

3. Autorizaciones de compra: En este apartado, en el supuesto caso de que un operario no encuentre el material en stock, podrá emitir una solicitud de autorización de compra del material necesario para la correcta finalización de la incidencia. Posteriormente, un empleado administrativo aceptará o rechazará dicha petición. Finalmente se le permitirá al operario, ver y modificar ciertos valores de los pedidos realizados por él mismo.
4. Estadísticas: El personal de campo podrá ver a título informativo, una estadística de las máquinas que más averías han sufrido en los últimos tres meses. Dicho reporte deberá servir para informar convenientemente al jefe de obra.

8.1.2 Particularidades del aplicativo móvil

Una de las funcionalidades que se contemplan en este proyecto, es la posibilidad de permitir al trabajador de campo, la lectura de etiquetas de códigos **QR**[3]. Es importante reseñar, que la impresión y el diseño así como el contenido de las mismas no entra dentro del ámbito de este proyecto. Por lo tanto, la compañía deberá tener un sistema paralelo para la elaboración de las mismas. Por otro lado, y para el correcto funcionamiento de la aplicación, dichas etiquetas tan solo deberán incluir como dato, el identificador de aparato a mantener.

La ubicación de la gran mayoría de aparatos a mantener es un punto crítico para el sistema. Al encontrarse muchos de ellos en zonas de no cobertura telefónica, y por lo tanto sin conectividad con el servidor, ha de forzosamente requerir del aplicativo el poder trabajar en modalidad desconectada. Por lo tanto, se contempla la posibilidad de trabajar en la memoria del dispositivo mientras no haya conexión con el servidor y hacer una sincronización con la base de datos posteriormente. Si el operario, por la contra trabaja con conexión, todas las modificaciones serán llevadas a cabo en el sistema gestor de base de datos inmediatamente. Ahora bien, como restricción del sistema establecemos que, para la carga inicial de los datos, el operario deberá haberse conectado al servidor en al menos una ocasión cada día (normalmente por la mañana a primera hora).

8.1.3 Seguridad

Dado que los datos que se transfieren entre sistemas (plataforma móvil y servidor) no son especialmente sensibles, no se establece ningún mecanismo de seguridad a nivel de protocolo de transporte. Así pues, todas las comunicaciones entre el servidor y el dispositivo móvil se realizarán sobre el protocolo **HTTP**[14].

Si se establece sin embargo un mecanismo de autentificación, basado en un formulario de tipo usuario y contraseña para acceder al sistema. En ningún caso la contraseña del operario quedará serializada en el dispositivo móvil. Esto se contempla como un mecanismo adicional de seguridad.

8.2 Requerimientos no Funcionales

En este apartado trataremos los requerimientos no funcionales relativos a la interfaz, a la operación y al desarrollo de la aplicación.

8.2.1 Interfaz

Para el desarrollo y diseño de la interfaz gráfica del usuario se han tenido en cuenta las dimensiones reducidas de las pantallas de los mismos. Así pues, para mitigar el efecto que éstas pueden tener en los operarios de la empresa, donde un tamaño muy pequeño puede afectar a la eficiencia del aplicativo, se ha intentado por todos los medios adecuar los componentes a dimensiones más acordes a las exigencias de los mismos. Por otro lado, y dada dificultad de escritura que presentan las plataformas móviles, la gran mayoría de campos de entrada incorporan elementos gráficos que deben facilitar al operario la correcta introducción de los datos. Todas estas acciones han ido encaminadas a mejorar la usabilidad de la aplicación.

Además de cara a maximizar la duración de la batería, elemento indispensable para el correcto funcionamiento del sistema, las interacciones con el servidor se han minimizado por diseño. Es decir, dado que una alta frecuencia de transmisión de datos entre el servidor y el dispositivo puede provocar un consumo excesivo de batería, se ha decidido por un lado ofrecer la posibilidad de trabajar en modalidad desconectada y así minimizar el tránsito de datos, y por otro, se ha diseñado el aplicativo para que sólo se produzcan transmisiones de una forma controlada y cuando sea estrictamente necesario.

Finalmente, se han establecido mecanismos de retroalimentación (a través de iconos, mensajes, ayudas, etc...) que facilitan en todo momento el entendimiento por parte del operario, del estado en el que se encuentra cada uno de los flujos de la información. Por otro lado, se ha intentado seguir las convenciones establecidas en cuanto a los elementos visuales e iconografías se refiere.

8.2.2 Otros aspectos

Cuando el dispositivo móvil se encuentra trabajando en estado de conexión, los tiempos de respuesta siempre van a ir condicionados a la velocidad de transmisión de la red de datos contratada por la empresa, así como de, la cantidad de datos a trasmitir. Para mitigar las transmisiones, todo el contenido estático del aplicativo (imágenes, iconos, textos de ayuda, etc...) se guardan en el dispositivo móvil en el momento de instalación de la *app*. En lo referente a los datos no estáticos, todas las transmisiones entre el móvil y el servidor se realizarán en un formato **JSON**[10] (formato ligero para el intercambio de datos), para evitar sobrecargar en exceso el proceso.

En cuanto al desarrollo comentar que, la estructura interna del proyecto ha de ayudar a la mantenibilidad, flexibilidad y reusabilidad de sus componentes, ante posibles adaptaciones y mejoras futuras. Un aspecto importante a tener en cuenta es que, de cara a mejorar estos aspectos, se utilizarán estándares tecnológicos del mercado. Esto sin duda, ayudará a minimizar el impacto que en un futuro pueda tener la aplicación de un arreglo, derivado de la poca especialización requerida por el departamento de desarrollo (uso de tecnologías web muy difundidas; **HTML**, **CSS**, **JavaScript**).

8.2.3 Funcionalidades del sistema

A continuación describiremos más en detalle cada uno de los casos de uso introducidos en el apartado anterior 'Descripción básica del funcionamiento del aplicativo':

- Autentificación del Usuario: esta función ofrece la posibilidad al usuario de poder conectarse al sistema a través de un formulario con usuario y contraseña.
- Ver Incidencias: con esta función el operario podrá visualizar todas las incidencias que le han sido asignadas. Las incidencias deben aparecer ordenadas por orden de criticidad, donde las más críticas aparecerán primero.
- Alta de Incidencia: a través de esta función el operario podrá introducir una incidencia nueva en el sistema.
- Modificación de Incidencia: mediante esta opción el operario podrá editar las propiedades de las incidencias reportadas en el sistema.
- Visualización de Calendario de Incidencias: vista en modo calendario de las incidencias preventivas asignadas al operario.
- Visualización de Material en Stock: esta función permite al operario visualizar el estado del stock del material a solicitar en el almacén de la empresa.
- Hacer Petición Material: con esta función el operario podrá hacer una petición de materiales, y por lo tanto reservarlos, siempre y cuando estos estén disponibles.
- Ver Peticiones de Reserva: mediante esta función el operario podrá ver los materiales que previamente a reservado.
- Borrar Peticiones de Reserva: esta función permite al operario borrar peticiones de materiales previas.
- Modificar Peticiones de Reserva: esta función permite al operario modificar peticiones de materiales previas.
- Ver Números de Autorización de Compra: esta función permite al operario ver el estado de las peticiones que él mismo ha procesado o introducido en el sistema.
- Alta de Números de Autorización de Compra: a través de esta opción el personal de campo puede hacer la solicitud de un número de autorización, para realizar la compra de materiales no disponibles en almacén.
- Modificación de Número de Autorización de Compra: mediante esta opción el operario podrá editar las propiedades de una autorización .

- Borrar Números de Autorización de Compra: con esta opción un operario podrá borrar una solicitud previa de número de autorización.
- Ver Estadística: Visualización de estadística de máquinas averiadas en los últimos tres meses en formato gráfico.
- Sincronizar datos con el Servidor: esta función permite sincronizar los datos entre el dispositivo móvil y el servidor, si el primero trabaja en modalidad desconectada.
- Mostrar Mapa: esta función permite al usuario localizar geográficamente una incidencia en un mapa.
- Lectura Código QR: con esta función el usuario puede leer mediante la cámara del móvil una etiqueta con un código **QR[3].**

8.2.4 Actores del sistema

En el siguiente apartado describiremos los actores que intervienen en los intercambios de flujos de información de la compañía. A pesar de que, para el aplicativo que estamos modelando y desarrollando no es necesaria la contemplación de todos ellos, dado que el único tipo de usuario que realmente va a hacer funcionar el aplicativo va a ser el operario, se describen todos para ayudar a mejorar el entendimiento general de todo el sistema.

1. Operario: realizará todas las tareas descritas en el apartado anterior desde su plataforma móvil.
2. Administrador: el administrador del sistema deberá proveer al mismo de los datos necesarios fijos, para el correcto funcionamiento del aplicativo. Del mismo modo se deberá hacer cargo del mantenimiento de los citados datos. Por ejemplo, podemos considerar datos fijos los referentes a; Usuarios registrados y Proveedores.
3. Administrativo: el personal administrativo de la empresa se deberá encargar de la alimentación de los datos variables del sistema, tales como, introducción y asignación de incidencias, recepción y aceptación o denegación de números de autorización de compra, introducción en sistemas de soporte internos de la compañía de nuevas incidencias reportadas por el personal de campo.
4. Sistema: un proceso ejecutado diariamente se encargará de actualizar el stock diariamente, a partir de las peticiones de reserva.

8.2.5 Casos de Uso

En esta sección se describirán a fondo cada uno de los casos de uso citados en el apartado anterior usando una notación informal. Se incluirá también un esquema general de funcionamiento del sistema, donde se contemplará al actor principal **Operario** y sus casos de uso en un diagrama de casos de usos. Es importante reseñar que, para que todos estos casos de uso funcionen con normalidad (especialmente los referidos al manejo de las entidades principales; Incidencia, Autorización de Compra, Reserva Material, Proveedor), el operario ha de haber realizado al menos una conexión

al sistema servidor, tal y como se indica con anterioridad, en caso contrario los citados casos de uso pueden sufrir variaciones con respecto a los descritos a continuación.

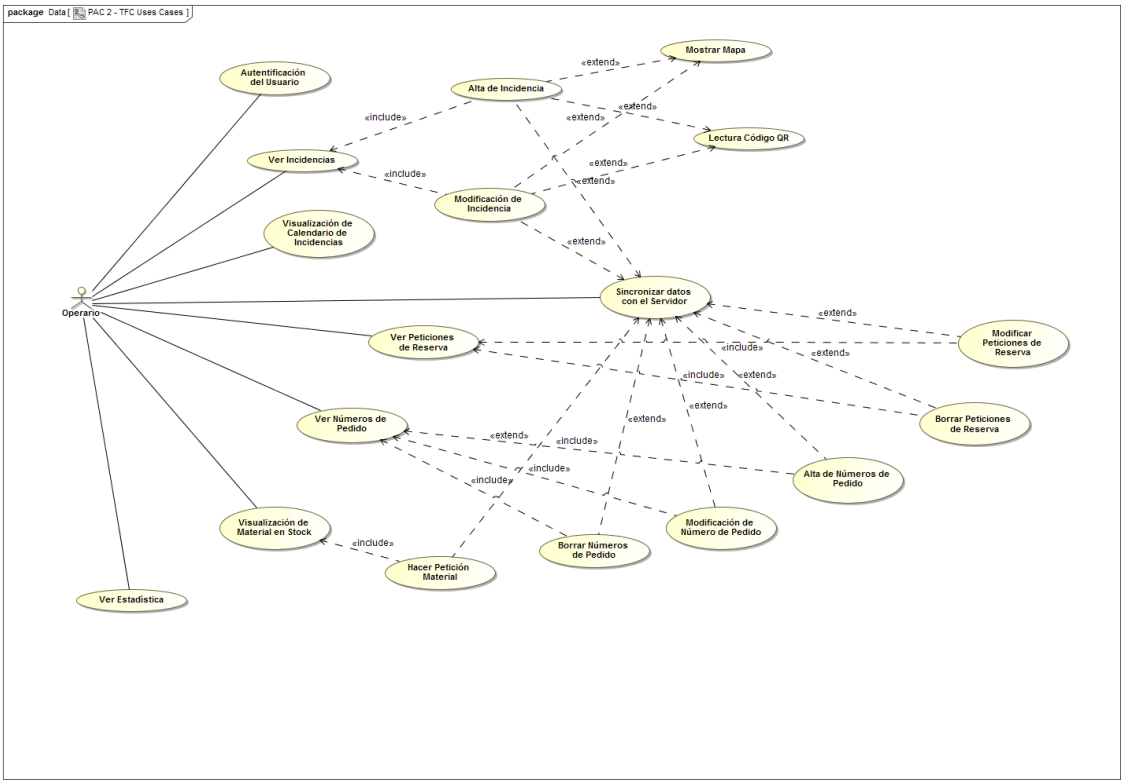


Ilustración 13 - Diagrama de Casos de Uso

Descripción de caso de uso [CU00] Autentificación del Usuario

Identificador:	**[CU00]**
Nombre:	**Autentificación del Usuario**
Resumen:	Este caso representa como un usuario accede al sistema mediante un mecanismo de autentificación basado en usuario y contraseña
Actores:	Operario
Precondiciones:	El usuario debe existir en el sistema
Postcondiciones:	El usuario accede a las funcionalidades del aplicativo
Flujo Normal:	1) El usuario arranca el aplicativo y aparece una pantalla donde se le pide que introduzca el usuario y contraseña 2) El usuario introduce la información solicitada y acepta 3) El sistema comprueba que los datos introducidos son correctos 4) Si la información es correcta el usuario accede a las funcionalidades del aplicativo
Flujos Alternativos:	3a) Si el sistema comprueba que los datos proporcionados no son correctos se muestra un mensaje de error y el flujo retorna al paso 2 3b) Si en el momento de la comprobación no hay conexión se muestra un mensaje informativo informando de la incidencia y se le permite al usuario trabajar en modalidad '*offline*'
Inclusiones	Ninguna
Extensiones	Ninguna

Tabla 11 - [CU00] Autentificación del Usuario

Descripción de caso de uso [CU01] Ver Incidencias

Identificador:	**[CU01]**
Nombre:	**Ver Incidencias**
Resumen:	Este caso representa como un usuario accede al sistema para visualizar las incidencias que tiene asignadas en formato de lista ordenada por criticidad
Actores:	Operario
Precondiciones:	Las incidencias han de estar previamente informadas en el sistema por parte del personal Administrativo
Postcondiciones:	El usuario obtiene una lista ordenada por criticidad de incidencias
Flujo Normal:	1) El usuario accede a través de la opción de menú Incidencias > Lista de Incidencias 2) El sistema muestra la lista ordenada de las incidencias asignadas al operario
Flujos Alternativos:	2a) Si el usuario pulsa el botón 'Anterior' se abandona el caso de uso
Inclusiones	Ninguna
Extensiones	Ninguna

Tabla 12 - [CU01] Ver Incidencias

Descripción de caso de uso [CU02] Alta Incidencias

Identificador:	**[CU02]**
Nombre:	**Alta de Incidencias**
Resumen:	Este caso representa como un usuario accede al sistema para introducir una nueva incidencia en el sistema
Actores:	Operario
Precondiciones:	Las incidencias no ha de existir previamente en el sistema
Postcondiciones:	Una nueva incidencia aparece en el sistema
Flujo Normal:	1) El usuario accede a través de la opción de menú Incidencias > Lista de Incidencias 2) El usuario pulsa el botón habilitado para dar soporte a esta funcionalidad, simbolizado por un signo '+' 3) El sistema muestra un formulario, inicialmente con los valores por defecto 4) El usuario rellena los valores de los campos requeridos y pulsa aceptar 5) El sistema valida los datos introducidos 6) Si las validaciones han sido correctas, el sistema añade la incidencia a la base de datos 7) El sistema muestra de nuevo la lista de incidencias con la nueva incidencia
Flujos Alternativos:	2a) Si el usuario pulsa el botón 'Anterior' se abandona el caso de uso 3a) Si usuario pulsa el botón 'Anterior', el sistema muestra de nuevo la pantalla de incidencias presentada en el caso de uso [CU01] Ver Incidencias 5a) Si la validación de los datos no es correcta, el sistema presenta un mensaje de error y el flujo vuelve al punto 3 6a) Si el usuario trabaja en modalidad desconectada, los cambios en la base de datos se llevarán a cabo mediante el caso de uso [CU16] Sincronizar datos con el Servidor
Inclusiones	[CU01] Ver Incidencias
Extensiones	[CU16] Sincronizar datos con el Servidor [CU17] Mostrar Mapa [CU18] Lectura Código QR

Tabla 13 - [CU02] Alta Incidencias

Descripción de caso de uso [CU03] Modificación de Incidencia

Identificador:	**[CU03]**
Nombre:	**Modificación de Incidencia**
Resumen:	Este caso representa como un usuario accede al sistema para modificar las propiedades de una incidencia del sistema
Actores:	Operario
Precondiciones:	Las incidencias han de existir previamente en el sistema
Postcondiciones:	Los cambios introducidos en la incidencia por el operario quedan reflejados en el sistema
Flujo Normal:	1) El usuario accede a través de la opción de menú Incidencias > Lista de Incidencias 2) El usuario pulsa sobre una incidencia de la lista 3) El sistema muestra un formulario con los datos de la incidencia seleccionada 4) El usuario cambia los valores de los campos y pulsa aceptar 5) El sistema valida los datos introducidos 6) Si las validaciones han sido correctas, el sistema modifica la incidencia a la base de datos 7) El sistema muestra de nuevo la lista de incidencias con los valores de la incidencia modificados
Flujos Alternativos:	2a) Si el usuario pulsa el botón 'Anterior' se abandona el caso de uso 3a) Si usuario pulsa el botón 'Anterior', el sistema muestra de nuevo la pantalla de incidencias presentada en el caso de uso [CU01] Ver Incidencias 5a) Si la validación de los datos no es correcta, el sistema presenta un mensaje de error y el flujo vuelve al punto 3 6a) Si el usuario trabaja en modalidad desconectada, los cambios en la base de datos se llevarán a cabo mediante el caso de uso [CU16] Sincronizar datos con el Servidor
Inclusiones	[CU01] Ver Incidencias
Extensiones	[CU16] Sincronizar datos con el Servidor [CU17] Mostrar Mapa [CU18] Lectura Código QR

Tabla 14 - [CU03] Modificación de Incidencia

Descripción de caso de uso [CU04] Visualización de Calendario de Incidencias

Identificador:	**[CU04]**
Nombre:	**Visualización de Calendario de Incidencias**
Resumen:	Este caso representa como un usuario accede al sistema para ver las incidencias preventivas en formato calendario
Actores:	Operario
Precondiciones:	Las incidencias han de existir previamente en el sistema
Postcondiciones:	El sistema muestra un calendario con las incidencias preventivas
Flujo Normal:	1) El usuario accede a través de la opción de menú Incidencias > Calendario 2) El sistema muestra un calendario con las incidencias preventivas
Flujos Alternativos:	Ninguno
Inclusiones	Ninguna
Extensiones	Ninguna

Tabla 15 - [CU04] Visualización de Calendario de Incidencias

Descripción de caso de uso [CU05] Visualización de Material en Stock

Identificador:	**[CU05]**
Nombre:	**Visualización de Material en Stock**
Resumen:	Este caso representa como un usuario accede al sistema para ver los materiales en stock disponibles en el almacén de la empresa
Actores:	Operario
Precondiciones:	Los materiales y sus stocks han de existir en el sistema
Postcondiciones:	El sistema muestra una lista categorizada de materiales y de sus respectivos stocks
Flujo Normal:	1) El usuario accede a través de la opción de menú Almacén 2) El sistema muestra una vista desplazable con las categorías existentes 3) El usuario selecciona la categoría y pulsa el botón 'Ver' 4) El sistema muestra una lista con todos los materiales relacionados con la categoría seleccionada y sus stocks
Flujos Alternativos:	2a) Si el usuario pulsa el botón 'Inicio', en este punto se abandona el caso de uso 4a) Si el usuario pulsa el botón 'Anterior' se retorna al punto 2
Inclusiones	Ninguna
Extensiones	Ninguna

Tabla 16 - [CU05] Visualización de Material en Stock

Descripción de caso de uso [CU06] Hacer Petición Material

Identificador:	**[CU06]**
Nombre:	**Hacer Petición Material**
Resumen:	Este caso representa como un usuario accede al sistema para hacer una reserva de materiales en el stock de almacén interno de la empresa
Actores:	Operario
Precondiciones:	Los materiales y sus stocks han de existir en el sistema
Postcondiciones:	Una nueva reserva de materiales aparece en el sistema
Flujo Normal:	1) El usuario accede a través de la opción de menú Almacén 2) El sistema muestra una vista desplazable con las categorías existentes 3) El usuario selecciona la categoría y pulsa el botón 'Ver' 4) El sistema muestra una lista con todos los materiales relacionados con la categoría seleccionada y sus stocks 5) El usuario pulsa sobre el material deseado, seleccionándolo de esta forma 6) El sistema presenta un formulario con los valores por defecto 7) El usuario rellena los valores de los campos requeridos y pulsa aceptar 8) El sistema valida los datos introducidos 9) Si las validaciones han sido correctas, el sistema añade la reserva a la base de datos 10) El sistema muestra de nuevo la lista de materiales relacionados con la categoría seleccionada
Flujos Alternativos:	2a) Si el usuario pulsa el botón 'Inicio', en este punto se abandona el caso de uso 4a) Si el usuario pulsa el botón 'Anterior' se retorna al punto 2 6a) Si el usuario pulsa el botón 'Anterior', el sistema muestra de nuevo la lista de materiales presentada en el caso de uso [CU05] Visualización de Material en Stock 8a) Si la validación de los datos no es correcta, el sistema presenta un mensaje de error y el flujo vuelve al punto 6 9a) Si el usuario trabaja en modalidad desconectada, los cambios en la base de datos se llevarán a cabo mediante el caso de uso [CU16] Sincronizar datos con el

	Servidor
Inclusiones	[CU05] Visualización de Material en Stock
Extensiones	[CU16] Sincronizar datos con el Servidor

Tabla 17 - [CU06] Hacer Petición Material

Descripción de caso de uso [CU07] Ver Peticiones de Reserva

Identificador:	**[CU07]**
Nombre:	**Ver Peticiones de Reserva**
Resumen:	Este caso representa como un usuario accede al sistema para ver las peticiones de reserva de material que ha realizado hasta el momento
Actores:	Operario
Precondiciones:	Se debe haber realizado previamente al menos una petición de reserva en el sistema
Postcondiciones:	Se muestra una lista de peticiones de reserva de materiales
Flujo Normal:	1) El usuario accede a través de la opción de menú Almacén 2) El sistema muestra una vista desplazable con las categorías existentes. Se desplaza hasta 'Mis Reservas' 3) El usuario pulsa el botón 'Ver' 4) El sistema muestra una lista con todas las reservas de materiales y las cantidades de los mismos
Flujos Alternativos:	2a) Si el usuario pulsa el botón 'Inicio', en este punto se abandona el caso de uso 4a) Si el usuario pulsa el botón 'Anterior' se retorna al punto 2
Inclusiones	Ninguna
Extensiones	Ninguna

Tabla 18 - [CU07] Ver Peticiones de Reserva

Descripción de caso de uso [CU08] Borrar Peticiones de Reserva

Identificador:	**[CU08]**
Nombre:	**Borrar Peticiones de Reserva**
Resumen:	Este caso representa como un usuario accede al sistema para borrar una reserva de materiales previamente realizada por él mismo
Actores:	Operario
Precondiciones:	La reserva de material debe existir en el sistema
Postcondiciones:	La reserva desaparece del sistema
Flujo Normal:	1) El usuario accede a través de la opción de menú Almacén 2) El sistema muestra una vista desplazable con las categorías existentes. Se desplaza hasta 'Mis Reservas' 3) El usuario pulsa el botón 'Ver' 4) El sistema muestra una lista con todas las reservas de materiales y las cantidades de los mismos 5) El usuario pulsa sobre la reserva deseada, seleccionándola de esta forma 6) El sistema presenta un formulario con los valores de la reserva 7) El usuario pulsa el botón asignado a esta función, simbolizado por un signo '-' 8) El sistema muestra un mensaje de confirmación 9) Si se confirma el sistema borra la reserva en la base de datos 10) El sistema muestra de nuevo la lista de reservas de materiales
Flujos Alternativos:	2a) Si el usuario pulsa el botón 'Inicio', en este punto se abandona el caso de uso 4a) Si el usuario pulsa el botón 'Anterior' se retorna al punto 2

	6a) Si el usuario pulsa el botón 'Anterior', el sistema muestra de nuevo la lista de materiales presentada en el caso de uso [CU07] Ver Peticiones de Reserva 8a) Si el usuario cancela se retorna al punto 6 9a) Si el usuario trabaja en modalidad desconectada, los cambios en la base de datos se llevarán a cabo mediante el caso de uso [CU16] Sincronizar datos con el Servidor
Inclusiones	[CU07] Ver Peticiones de Reserva
Extensiones	[CU16] Sincronizar datos con el Servidor

Tabla 19 - [CU08] Borrar Peticiones de Reserva

Descripción de caso de uso [CU09] Modificar Peticiones de Reserva

Identificador:	**[CU09]**
Nombre:	**Modificar Peticiones de Reserva**
Resumen:	Este caso representa como un usuario accede al sistema para modificar los valores de una reserva de materiales previamente realizada por él mismo
Actores:	Operario
Precondiciones:	La reserva de material debe existir en el sistema
Postcondiciones:	Los cambios introducidos en la reserva por el operario quedan reflejados en el sistema
Flujo Normal:	1) El usuario accede a través de la opción de menú Almacén 2) El sistema muestra una vista desplazable con las categorías existentes. Se desplaza hasta 'Mis Reservas' 3) El usuario pulsa el botón 'Ver' 4) El sistema muestra una lista con todas las reservas de materiales y las cantidades de los mismos 5) El usuario pulsa sobre la reserva deseada, seleccionándola de esta forma 6) El sistema presenta un formulario con los valores de la reserva 7) El usuario cambia los valores de los campos y pulsa aceptar 8) El sistema valida los datos introducidos 9) Si las validaciones han sido correctas, el sistema modifica la reserva en la base de datos 10) El sistema muestra de nuevo la lista de reservas de materiales
Flujos Alternativos:	2a) Si el usuario pulsa el botón 'Inicio', en este punto se abandona el caso de uso 4a) Si el usuario pulsa el botón 'Anterior' se retorna al punto 2 6a) Si el usuario pulsa el botón 'Anterior', el sistema muestra de nuevo la lista de materiales presentada en el caso de uso [CU07] Ver Peticiones de Reserva 9a) Si el usuario trabaja en modalidad desconectada, los cambios en la base de datos se llevarán a cabo mediante el caso de uso [CU16] Sincronizar datos con el Servidor
Inclusiones	[CU07] Ver Peticiones de Reserva
Extensiones	[CU16] Sincronizar datos con el Servidor

Tabla 20 - [CU09] Modificar Peticiones de Reserva

Descripción de caso de uso [CU10] Ver Números de Autorización de Compra

Identificador:	**[CU10]**
Nombre:	**Ver Números de Autorización de Compra**
Resumen:	Este caso representa como un usuario accede al sistema para ver los números de autorización solicitados por el operario
Actores:	Operario
Precondiciones:	Se debe haber realizado previamente al menos una solicitud de autorización en el

	sistema
Postcondiciones:	Se muestra una lista de solicitudes de números de autorización
Flujo Normal:	1) El usuario accede a través de la opción de menú Autorizaciones de Compra 2) El sistema muestra una lista con todas las solicitudes de números de autorización
Flujos Alternativos:	2a) Si el usuario pulsa el botón 'Inicio', en este punto se abandona el caso de uso
Inclusiones	Ninguna
Extensiones	Ninguna

Tabla 21 - [CU10] Ver Números de Autorización de Compra

Descripción de caso de uso [CU11] Alta de Números de Autorización de Compra

Identificador:	**[CU11]**
Nombre:	**Alta de Números de Autorización de Compra**
Resumen:	Este caso representa como un usuario accede al sistema para introducir una solicitud de números de autorización
Actores:	Operario
Precondiciones:	La solicitud de número de autorización no ha de existir en el sistema
Postcondiciones:	Una nueva solicitud de número de autorización aparece en el sistema
Flujo Normal:	1) El usuario accede a través de la opción de menú Autorizaciones de Compra 2) El sistema muestra una lista con todas las solicitudes de números de autorización 3) El usuario pulsa el botón habilitado para dar soporte a esta funcionalidad, simbolizado por un signo '+' 4) El sistema muestra un formulario, inicialmente con los valores por defecto 5) El usuario rellena los valores de los campos requeridos y pulsa aceptar 6) El sistema valida los datos introducidos 7) Si las validaciones han sido correctas, el sistema añade una solicitud de número de autorización a la base de datos 8) El sistema muestra de nuevo la lista de solicitudes con el nuevo número de autorización
Flujos Alternativos:	2a) Si el usuario pulsa el botón 'Inicio', en este punto se abandona el caso de uso 4a) Si el usuario pulsa el botón 'Anterior' se retorna al punto 2 6a) Si la validación de los datos no es correcta, el sistema presenta un mensaje de error y el flujo vuelve al punto 4 8a) Si el usuario trabaja en modalidad desconectada, los cambios en la base de datos se llevarán a cabo mediante el caso de uso [CU16] Sincronizar datos con el Servidor
Inclusiones	[CU10] Ver Números de Autorización de Compra
Extensiones	[CU16] Sincronizar datos con el Servidor

Tabla 22 - [CU11] Alta de Números de Autorización de Compra

Descripción de caso de uso [CU12] Modificación de Número de Autorización de Compra

Identificador:	**[CU12]**
Nombre:	**Modificación de Número de Autorización de Compra**
Resumen:	Este caso representa como un usuario accede al sistema para modificar los valores de una solicitud de número de autorización de compra previamente realizada por él mismo
Actores:	Operario

Precondiciones:	La solicitud de número de autorización debe existir en el sistema
Postcondiciones:	La solicitud de número de autorización desaparece del sistema
Flujo Normal:	1) El usuario accede a través de la opción de menú Autorizaciones de Compra 2) El sistema muestra una lista con todas las solicitudes de números de autorización 3) El usuario pulsa sobre la solicitud de número de autorización 4) El sistema muestra un formulario con los valores de la solicitud 5) El usuario cambia los valores de los campos deseados y pulsa aceptar 6) El sistema valida los datos introducidos 7) Si las validaciones han sido correctas, el sistema modifica una solicitud de número de autorización de compra en la base de datos 8) El sistema muestra de nuevo la lista de solicitudes con los cambios realizados sobre la solicitud de número de autorización
Flujos Alternativos:	2a) Si el usuario pulsa el botón 'Inicio', en este punto se abandona el caso de uso 4a) Si el usuario pulsa el botón 'Anterior' se retorna al punto 2 6a) Si el usuario pulsa el botón 'Anterior', el sistema muestra de nuevo la lista de materiales presentada en el caso de uso [CU10] Ver Números de Autorización de Compra 9a) Si el usuario trabaja en modalidad desconectada, los cambios en la base de datos se llevarán a cabo mediante el caso de uso [CU16] Sincronizar datos con el Servidor
Inclusiones	[CU10] Ver Números de Autorización de Compra
Extensiones	[CU16] Sincronizar datos con el Servidor

Tabla 23 - [CU12] Modificación de Número de Autorización de

Descripción de caso de uso [CU13] Borrar Números de Autorización de Compra

Identificador:	**[CU13]**
Nombre:	**Borrar Números de Autorización de Compra**
Resumen:	Este caso representa como un usuario accede al sistema para borrar una solicitud de número de autorización previamente realizada por él mismo
Actores:	Operario
Precondiciones:	La solicitud de número de autorización debe existir en el sistema
Postcondiciones:	Los cambios introducidos en la solicitud por el operario quedan reflejados en el sistema
Flujo Normal:	1) El usuario accede a través de la opción de menú Autorizaciones de Compra 2) El sistema muestra una lista con todas las solicitudes de números de autorización 3) El usuario pulsa sobre la solicitud de número de autorización 4) El sistema muestra un formulario con los valores de la solicitud 5) El usuario pulsa sobre el botón '-' 6) El sistema muestra un mensaje de confirmación 7) Si el usuario confirma el sistema da de baja la solicitud de número de autorización en la base de datos 8) El sistema muestra de nuevo la lista de solicitudes con los cambios realizados sobre la solicitud de número de autorización
Flujos Alternativos:	2a) Si el usuario pulsa el botón 'Inicio', en este punto se abandona el caso de uso 4a) Si el usuario pulsa el botón 'Anterior' se retorna al punto 2, descrito en [CU10] Ver Números de Autorización de Compra 6a) Si el usuario cancela se retorna al punto 4 8a) Si el usuario trabaja en modalidad desconectada, los cambios en la base de datos se llevarán a cabo mediante el caso de uso [CU16] Sincronizar datos con el Servidor

Inclusiones	[CU10] Ver Números de Autorización de Compra
Extensiones	[CU16] Sincronizar datos con el Servidor

Tabla 24 - [CU13] Borrar Números de Autorización de Compra

Descripción de caso de uso [CU14] Ver Estadística

Identificador:	**[CU14]**
Nombre:	**Ver Estadística**
Resumen:	Este caso representa como un usuario accede al sistema para ver las estadísticas de las averías por máquina de los últimos tres meses
Actores:	Operario
Precondiciones:	Las averías han de existir en el sistema
Postcondiciones:	Se visualiza una representación gráfica en % de las incidencias de los últimos 3 meses
Flujo Normal:	1) El usuario accede a través de la opción de menú Estadísticas > Averías x Máquina 2) El sistema muestra un gráfico con las estadísticas
Flujos Alternativos:	Ninguno
Inclusiones	Ninguna
Extensiones	Ninguna

Tabla 25 - [CU14] Ver Estadística

Descripción de caso de uso [CU15] Ver Estadística

Identificador:	**[CU15]**
Nombre:	**Ver Estadística**
Resumen:	Este caso representa como un usuario accede al sistema para ver las estadísticas de las averías por máquina de los últimos tres meses
Actores:	Operario
Precondiciones:	Las averías han de existir en el sistema
Postcondiciones:	Se visualiza una representación gráfica en % de las incidencias de los últimos 3 meses
Flujo Normal:	1) El usuario accede a través de la opción de menú Estadísticas > Averías x Máquina 2) El sistema muestra un gráfico con las estadísticas
Flujos Alternativos:	Ninguno
Inclusiones	Ninguna
Extensiones	Ninguna

Tabla 26 - [CU15] Ver Estadística

Descripción de caso de uso [CU16] Sincronizar datos con el Servidor

Identificador:	**[CU16]**
Nombre:	**Sincronizar datos con el Servidor**
Resumen:	Este caso representa como un usuario accede al sistema para sincronizar los datos con el servidor. En caso de que haya estado trabajando en modalidad no conectada.
Actores:	Operario
Precondiciones:	Debe haber conexión con el sistema principal o servidor

Postcondiciones:	La base de datos refleja los cambios realizados en modalidad no conectada
Flujo Normal:	1) El usuario accede desde el menú principal a la opción de 'Sincronizar' 2) El sistema realiza las sincronizaciones, mientras se muestra un mensaje de ocupado
Flujos Alternativos:	1a) Si no hay conexión el sistema mostrará un mensaje de error y finalizará el caso de uso
Inclusiones	Ninguna
Extensiones	Ninguna

Tabla 27 - [CU16] Sincronizar datos con el Servidor

Descripción de caso de uso [CU17] Mostrar Mapa

Identificador:	**[CU17]**
Nombre:	**Mostrar Mapa**
Resumen:	Este caso representa como un usuario accede a un sistema de geolocalizazión (Google Maps) para ver la ubicación física de una incidencia
Actores:	Operario
Precondiciones:	Debe haber conexión con el sistema principal o servidor
Postcondiciones:	El sistema muestra un mapa con la localización de la incidencia
Flujo Normal:	1) El usuario accede desde el menú principal a la opción de el caso de uso [CU02] Alta de Incidencia o el caso de uso [CU03] Modificación de Incidencia 2) El usuario pulsa el botón 'Mapa' 3) El sistema muestra un mapa (Google Maps) con la localización de la incidencia
Flujos Alternativos:	2a) Si no hay conexión el sistema mostrará un mensaje de error y el flujo retorna al punto 1
Inclusiones	Ninguna
Extensiones	Ninguna

Tabla 28 - [CU17] Mostrar Mapa

Descripción de caso de uso [CU18] Lectura Código QR

Identificador:	**[CU18]**
Nombre:	**Lectura Código QR**
Resumen:	Este caso representa como un usuario accede a las capacidades de la cámara para procesar una etiqueta con un código QR que identifica el equipo
Actores:	Operario
Precondiciones:	Debe existir un equipo con una etiqueta con un código QR
Postcondiciones:	El sistema recupera el identificador de máquina embebido en el código QR
Flujo Normal:	1) El usuario accede desde el menú principal a la opción de el caso de uso [CU02] Alta de Incidencia o el caso de uso [CU03] Modificación de Incidencia 2) El usuario pulsa el botón 'QR' 3) El sistema recupera el identificador de la máquina embebido dentro del código QR
Flujos Alternativos:	Ninguno
Inclusiones	Ninguna
Extensiones	Ninguna

Tabla 29 - [CU18] Lectura Código QR

En esta sección desarrollaremos todos los aspectos referentes a la arquitectura del producto, desde varios puntos de vista; desde el enfoque físico, lógico y por último utilizaremos una notación UML para representar algunos ejemplos de diagramas de secuencias, que nos ayudarán a entender mejor de que partes consta nuestro proyecto, como se relacionan y como intercambian información.

9.1 Arquitectura Física

En un paradigma de programación de cliente servidor para aplicaciones distribuidas podemos diferenciar dos actores principales; el cliente, que en nuestro caso queda representado por el dispositivo móvil y el servidor, en nuestro caso, un servidor de aplicaciones basado en tecnología **Java**[8]. En los siguientes ilustraciones analizaremos en tres escenarios diferentes, que elementos componen dicha arquitectura desde el punto de vista físico.

9.1.1 Escenario de desarrollo

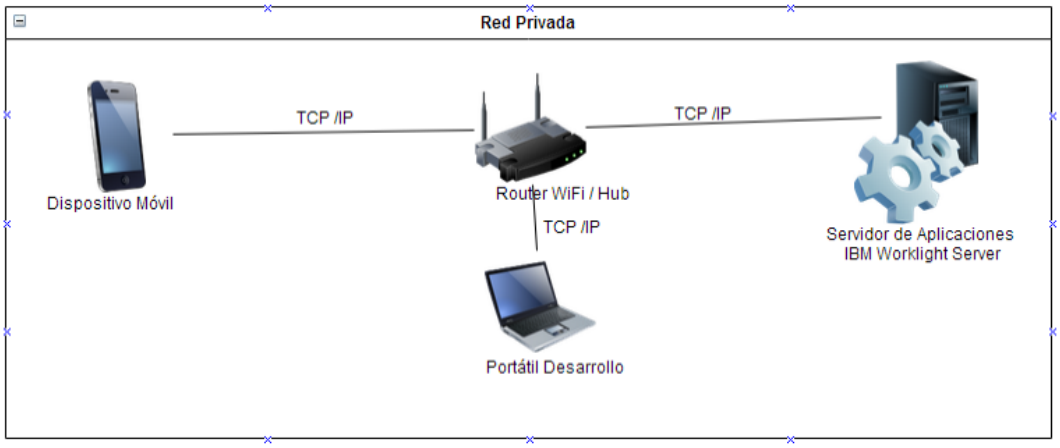


Ilustración 14 - Escenario de desarrollo

En la ilustración de arriba podemos ver el primero de los escenarios, donde tanto el dispositivo móvil, como el ordenador de desarrollo se conectan a través de una red privada al servidor que provee los servicios de conectividad a bases de datos y

seguridad, entre otros. Este escenario describe un entorno de desarrollo, y por lo tanto, no debe ser usado para producción.

9.1.2 Escenario de producción con acceso público al servidor de aplicaciones

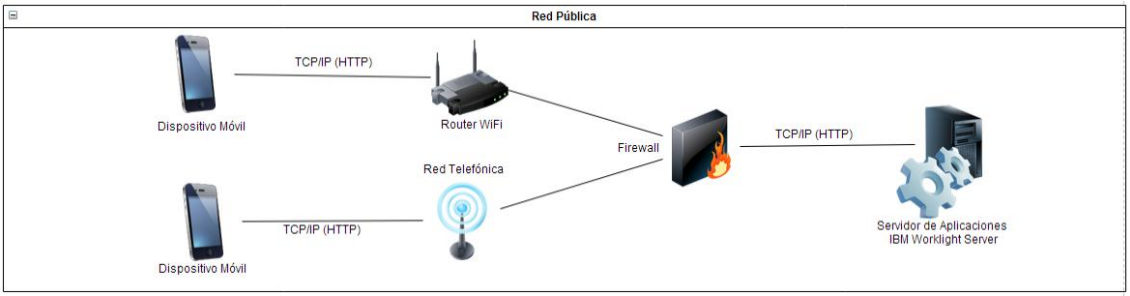


Ilustración 15 - Escenario de producción con acceso público al servidor de aplicaciones

En este escenario observamos que el acceso a los servicios por parte de los clientes se realiza a través de dos canales diferenciados no excluyentes; a través de un *Router Wi-fi* en el mismo rango IP del servidor, o bien, a través de la red telefónica de la operadora contratada, accediendo a una IP Pública de Internet, que a su vez re direcciona todas las peticiones entrantes HTTP al servidor de aplicaciones. Dicha configuración deberá ser utilizada en un entorno de producción.

9.1.3 Escenario de producción con acceso público al servidor de aplicaciones con VPN

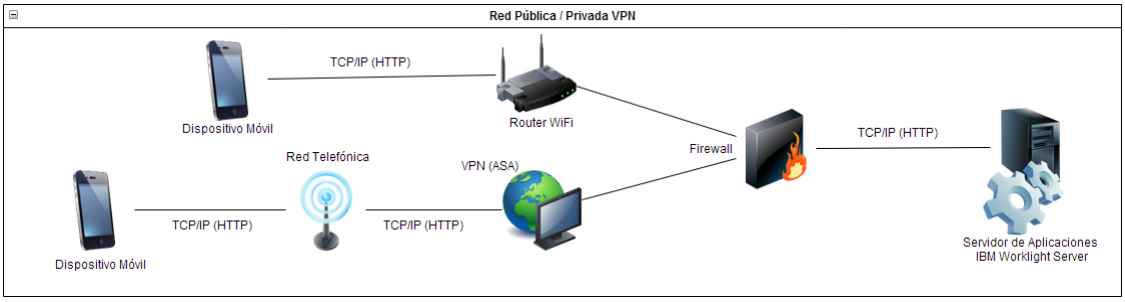


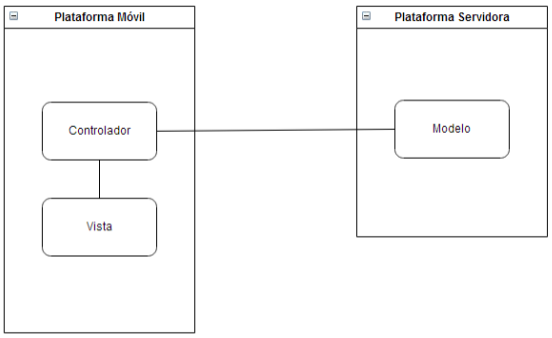
Ilustración 16 - Escenario de producción con acceso público al servidor de aplicaciones con VPN

Por último, el escenario representado arriba es prácticamente igual al anterior, sin embargo éste incorpora un elemento adicional de seguridad (VPN ASA), que nos permite confeccionar una red privada virtual o **VPN[6],** entre los dispositivos móviles clientes y el servidor, siempre y cuando el acceso al mismo no pueda realizarse a través de un punto de acceso *Wi-Fi* local. Al igual que en el anterior caso, dicha configuración también deberá ser utilizada en un entorno de producción, pudiendo elegir el cliente la que más se acomode a sus necesidades.

9.2 Arquitectura Lógica

A continuación presentaremos desde el punto de vista de la arquitectura lógica, los elementos que conforman la solución. Para la elaboración del aplicativo hemos

seguido un paradigma de desarrollo **Modelo-Vista-Controlador** o **MVC**[15]. El uso de dicho paradigma nos ayudará a desacoplar los elementos, a aumentar la cohesión entre los mismos y por lo tanto, a facilitar de esta forma el desarrollo de los componentes y se posterior mantenimiento.



La ilustración a mano izquierda muestra cómo quedan distribuidas cada una de las capas (Modelo, Vista y Controlador) en cada uno de los elementos físicos (cliente y servidor).

Ilustración 17 - Modelo-Vista-Controlador

- Capa de Vista: basada prácticamente en toda su totalidad en tecnologías HTML, CSS y JavaScript (**Dojo Mobile**[9]). Consta de elementos adicionales, tales como **IBM Worlight Runtime**, que provee conectividad al servidor y abstracción multiplataforma y **Apache Cordova**[12]**. Apache Cordova**, es un conjunto de **APIs** de dispositivo que proporcionan acceso desde la capa **JavaScript** a las funcionalidades nativas del dispositivo móvil, tales como GPS, cámara, etc... Veremos más adelante con detalle todos estos elementos cuando se describa la arquitectura lógica de la plataforma cliente.
- Capa de Controlador: actúa de puente entre la capa de Modelo, situado en la parte servidora y la de Vista, situada en la plataforma móvil. La vista delega toda la responsabilidad de transferencia de datos y conexión con el servidor a la capa controladora, quien a su vez, presenta una interfaz con funciones que servirán de enlace para las acciones de la capa visual.
- Capa de Modelo: esta capa se encargará única y exclusivamente del acceso a la base de datos, estableciéndose un canal de comunicación **HTTP** de intercambio de datos en formato **JSON**[10] con la capa Controladora.

A continuación veremos cómo se aplica este paradigma a cada uno de los elementos físicos (cliente y servidor) descritos en la ilustración anterior.

9.2.1 Arquitectura Lógica de la capa cliente

La siguiente ilustración muestra como quedarían distribuidas las capas sobre las cuales hemos estado hablando en el apartado anterior, desde un punto de vista del desarrollo y la tecnología usada.

En el diagrama podemos observar dos bloque bien diferenciados; el primero en azul describe la capa de servicios que provee el producto **IBM Worklight Runtime**, mientras que el segundo bloque, de color gris, nos ofrece una visión de las tres capas a desarrollar en el aplicativo. A continuación, extenderemos la información referente a cada una de las mismas:

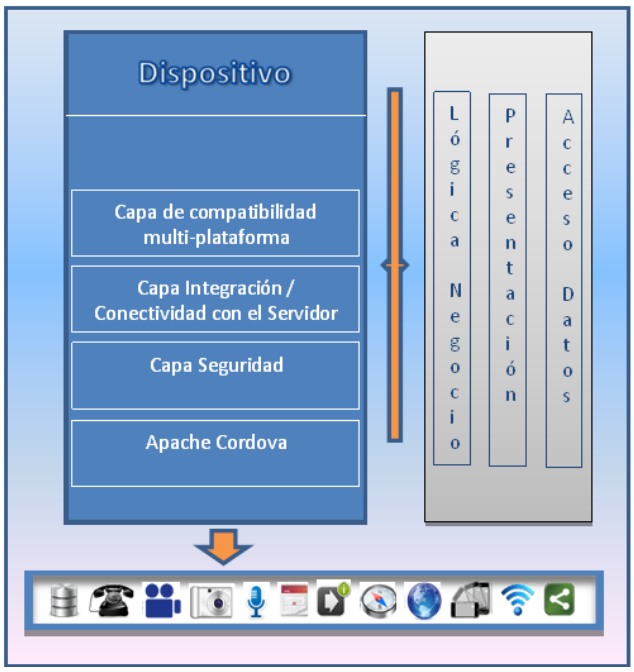


Ilustración 18 - Arquitectura Lógica de la capa cliente

- Capa de Presentación: básicamente compuesta por unidades lógicas llamadas vistas **Dojo Mobile**, que representan cada una de las pantallas con las que el usuario final podrá interactuar. Dichas vistas están agrupas en una única página HTML, típica arquitectura de una aplicación híbrida para dispositivos móviles.
- Capa de Lógica de Negocio: en dicha capa desarrollaremos todas las reglas de negocio que nos permitirá proporcionar funcionalidad a nuestra aplicación. Como ya hemos citado anteriormente, dicha capa debe actuar de puente entre las capas de presentación y acceso a datos, quedando agrupadas sus funciones y métodos en un único fichero **JavaScript**.
- Capa de Acceso a Datos: se responsabiliza de hacer las llamadas a las funciones de acceso a base de datos expuestas por los adaptadores en la parte servidora, utilizando como protocolo de transporte **HTTP** y **JSON** como método de encapsulamiento de datos.

9.2.2 Arquitectura Apache Cordova o PhoneGap

Una de las piezas fundamentales en la arquitectura lógica de la plataforma cliente es **Apache Cordova**, también conocido como **PhoneGap**. A grandes rasgos podemos definir **Apache Cordova**, cómo un *framework* de **código-abierto**[16] que permite construir aplicaciones nativas para móviles usando tecnologías propias del desarrollo web. A continuación describimos un poco más a fondo la arquitectura de dicha tecnología, tomando como base la siguiente ilustración.

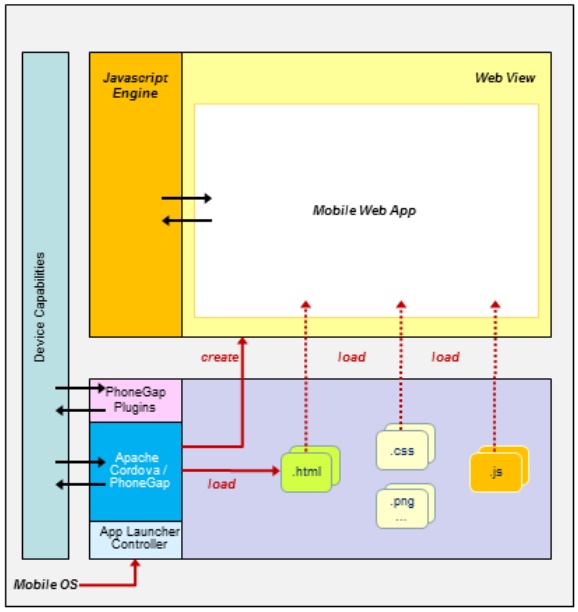


Ilustración 19 - Arquitectura Apache Cordova

Como podemos observar cada aplicación **PhoneGap** contiene dos elementos básicos; una clase nativa que encapsula el código **HTML** y **JavaScript** perteneciente a la SDK de **Android**, llamada **WebView**[17], y un puente de comunicación entre la parte nativa y la

parte script representada a través de la interfaz nativa de **Android JavascriptInterface**[18]**.**

De esta forma el sistema operativo es capaz de crear una clase **WebView**, sobre la cual se cargará la página HTML principal del aplicativo y las librerías script **Dojo Mobile**, mientras se establece un puente de comunicación entre la parte nativa del aplicativo y la parte script a través de la interfaz **JavascriptInterface.** Además como se puede observar, se usarán las **APIs** propietarias de **PhoneGap** para acceder a las funcionalidades nativas del dispositivo móvil, tales como, cámara, GPS, lista de contactos, etc...

9.2.3 Arquitectura IBM Worklight

Para acabar de tener una visión completa de la arquitectura lógica del aplicativo y de las tecnologías usadas en él, en el siguiente apartado describiremos como el producto **IBM Worklight** se apoya en la arquitectura de **PhoneGap**, la cual queda descrita en el apartado anterior, para de esta forma exponer alguna de sus funcionalidades. La siguiente ilustración nos muestra la arquitectura de capas del producto.

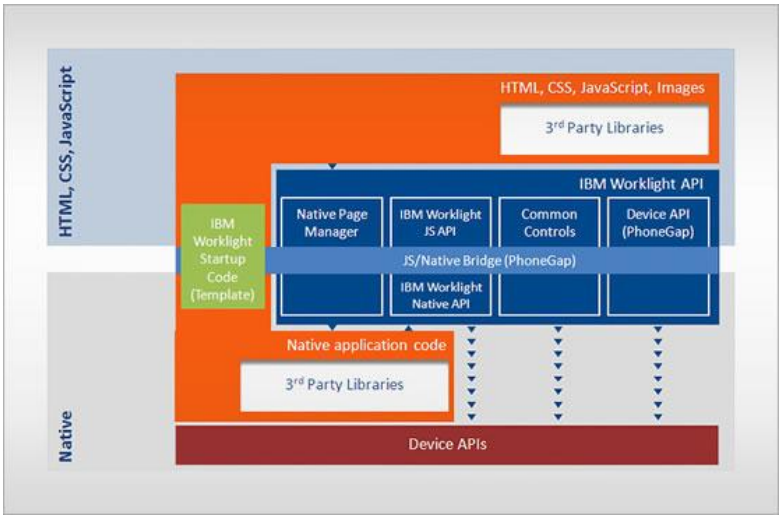


Ilustración 20 - Arquitectura IBM Worklight

En la ilustración anterior se puede apreciar, cómo **IBM Worklight** usa **PhoneGap** como puente de comunicación entre la capa web, cuyo desarrollo es principalmente en HTML, CSS y JavaScript, y la capa nativa. Adicionalmente **IBM Worklight** ofrece servicios complementarios de abstracción al aplicativo (aunque no aplica a este proyecto) gracias a su extensa API, permitiéndole desarrollar la idea de, 'codifica *una vez y despliega donde quieras'* propias de arquitecturas multiplataforma.

9.2.4 Arquitectura lógica de la plataforma servidora

En la siguiente ilustración se muestra la arquitectura del componente servidor

Ilustración 21 - Arquitectura lógica de la plataforma servidora

del producto **IBM Worklight Server**, diseñado para la integración de la parte cliente con los servicios de *back-end* de la compañía**.** El paradigma de integración de **IBM Worklight** está basado en adaptadores, los cuales son responsables de acceder a los servicios de base de datos utilizando el driver **JDBC** específico al sistema gestor y de transformar el resultado de la consulta a notación **JSON.**

Utilizando un canal de transporte **HTTP,** cada uno de los adaptadores es capaz de transferir a la parte cliente los datos resultado de la consulta a base de datos. Así pues, tal y como se ha citado anteriormente, podemos decir que los adaptadores se consideran parte de la capa de acceso a datos., quedando éstos englobados en la capa de modelo en una arquitectura **Modelo-Vista-Controlador**.

9.3 Arquitectura de la Base de Datos

La finalidad de la base de datos es dar persistencia a los datos dentro del dominio del aplicativo. Dichos datos deben ser los referentes a usuarios, incidencias, proveedores, materiales, autorizaciones de compra, equipos y reservas. A continuación describiremos más en detalla cada una de estas entidades y sus relaciones

9.3.1 Modelo Relacional de la Base de Datos

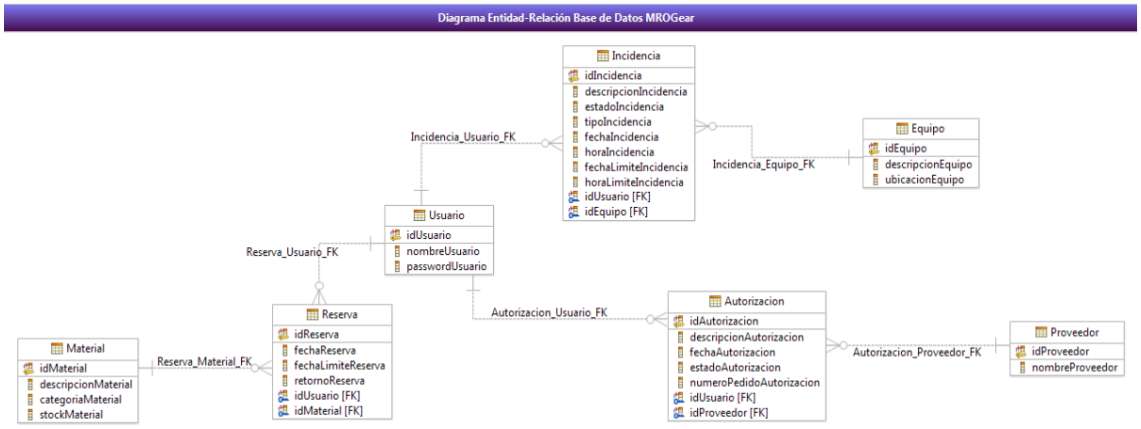


Ilustración 22 - Modelo Relacional de la Base de Datos

Entidad	Descripción	Cardinalidad	Entidad relacionada
Usuario	• Mantiene la persistencia de los usuarios y de sus contraseñas.	1:N	Incidencia

	• Se accederá a ella a la hora de autentificar el operario en el aplicativo. • No se realiza el mantenimiento directo desde el aplicativo.	1:N 1:N	Reserva Autorización
Incidencia	• Mantiene la persistencia de las incidencias reportadas y sus correspondientes estados. • Hay dos posibles estados (A-Abierta y C-Cerrada) • Modeliza los dos tipos de incidencias del sistema (I-Incidencia y P-Preventiva)	1:N 1:N	Equipo Usuario
Equipo	• Mantiene la persistencia de los equipos de frío y calor. • Tabla secundaría, por lo tanto no directamente mantenida por el aplicativo.	1:N	Incidencia
Material	• Mantiene la persistencia de los materiales de almacén. • El material deberá estar clasificado en 6 categorías fijas diferentes (1-Medición, 2-Mano, 3-Tornillería, 4-Eléctrico, 5-Fontanería y 6-Otros). No se contempla la opción de modelizar las catagorías. • Tabla secundaría, por lo tanto no directamente mantenida por el aplicativo	1:N	Reserva
Reserva	• Mantiene la persistencia de las reservas de material realizadas por el operario	1:N 1:N	Material Usuario
Autorización	• Mantiene la persistencia de las autorizaciones de compra pedidas por el operario y sus respectivos estados (1-Aprobada, 2-Rechazada, 3-Cancelada y 4-En Espera).	1:N 1:N	Usuario Proveedor
Proveedor	• Mantiene la persistencia de los proveedores. • Tabla secundaría, por lo tanto no directamente mantenida por el aplicativo.	1:N	Autorización

Tabla 30 - Modelo Relacional de la Base de Datos

Entidad Usuario

La entidad usuario queda definida por los siguientes atributos, así como por sus restricciones:

Atributo	Descripción	Tipo	Longitud	Tipo clave	Nulos
IdUsuario	Identificador del usuario. Usado por el operario para autentificarse en el sistema	VARCHAR	20	Primaria	NO
NombreUsuario	Nombre del operario	VARCHAR	80	-	SI
PasswordUsuario	Contraseña del usuario para acceder al sistema	VARCHAR	20	-	NO

Tabla 31 - Entidad Usuario

Entidad Incidencia

La entidad incidencia queda definida por los siguientes atributos, así como por sus

restricciones:

Atributo	Descripción	Tipo	Longitud	Tipo clave	Nulos
IdIncidencia	Identifica una incidencia dentro del sistema	INT	-	Primaria	NO
DescripcionIncidencia	Descripción de la incidencia	VARCHAR	200	-	SI
EstadoIncidencia	Indica si la incidencia está abierta o cerrada. A: Abierta y C: Cerrada	CHAR	1	-	NO
TipoIncidencia	Indica el tipo de incidencia. Los posibles valores son; I: Incidencia y P: Preventiva	CHAR	1	-	NO
FechaIncidencia	Indica la fecha en la que la incidencia es reportada en el sistema	DATE	-	-	NO
HoraIncidencia	Indica la hora en la que la incidencia es reportada en el sistema	TIME	-	-	NO
FechaLimiteIncidencia	Indica la fecha antes de la cual se ha de solventar la incidencia o el preventivo. En caso de ausencia de fecha indica que no tenemos penalización por sobrepasar la fecha límite.	DATE	-	-	SI
HoraLimiteIncidencia	Indica la hora antes de la cual se ha de solventar la incidencia o el preventivo. Siempre ha de ir acompañada de una fecha límite. Por si sola no tiene sentido.	TIME	-	-	SI
IdUsuario	Identifica el operario asignado a la incidencias	VARCHAR	20	Foránea	NO
IdEquipo	Identifica el equipo con la incidencia o preventivo	VARCHAR	10	Foránea	NO

Tabla 32 - Entidad Incidencia

Entidad Equipo

La entidad equipo queda definida por los siguientes atributos, así como por sus restricciones:

Atributo	Descripción	Tipo	Longitud	Tipo clave	Nulos
IdEquipo	Identifica un equipo dentro del sistema	VARCHAR	10	Primaria	NO
DescripcionEquipo	Descripción del equipo	VARCHAR	100	-	NO
UbicaciónEquipo	Indica la dirección física o ubicación del equipo.	VARCHAR	100	-	SI

Tabla 33 - Entidad Equipo

Entidad Material

La entidad material queda definida por los siguientes atributos, así como por sus restricciones:

Atributo	Descripción	Tipo	Longitud	Tipo clave	Nulos
IdMaterial	Identifica una pieza de material dentro del sistema	INT	-	Primaria	NO
DescripcionMaterial	Descripción del material	VARCHAR	100	-	NO
categoriaMaterial	Indica la categoría a la cual pertenece la pieza de material. Los posibles valores para este atributo son; 1: Medición, 2: Mano, 3: Tornillería, 4: Eléctrico, 5: Fontanería y 6: Otros	CHAR	1	-	NO
StockMaterial	Indica la cantidad del material presente en el almacén de la empresa.	INT	-	-	NO

Tabla 34 - Entidad Material

Entidad Reserva

La entidad reserva queda definida por los siguientes atributos, así como por sus restricciones:

Atributo	Descripción	Tipo	Longitud	Tipo clave	Nulos
IdReserva	Identifica una reserva de material por parte del operario	INT	-	Primaria	NO
FechaReserva	Indica la fecha en la que se ha realizado la reserva de material	DATE	-	-	NO
FechaLimiteReserva	Indica la fecha hasta la cual es necesario el material en caso de retorno del mismo	DATE	-	-	NO
RetornoReserva	Indica si el material ha de ser retornado o no. Los posibles valores son: 0: Sin retorno y 1: Con retorno	BIT	-	-	NO
CantidadReserva	Indica la cantidad de material a reservar	INT	-	-	SI
IdUsuario	Identificador del usuario. Usado por el operario para autentificarse en el sistema	VARCHAR	20	Foránea	NO
IdMaterial	Identifica una pieza de material dentro del sistema	INT	-	Foránea	NO

Tabla 35 - Entidad Reserva

Entidad Autorización

La entidad autorización de compra queda definida por los siguientes atributos, así como por sus restricciones:

Atributo	Descripción	Tipo	Longitud	Tipo clave	Nulos
IdAutorizacion	Identifica una petición de autorización de compra de material por parte del operario	INT	-	Primaria	NO
descripcionAutorizacion	Descripción de la autorización de compra de material	VARCHAR	200	-	NO
FechaAutorizacion	Indica la fecha en la cual se hace la petición de autorización.	DATE	-	-	NO
EstadoAutorizacion	Indica el estado en el cual se encuentra la petición de autorización de compra de material, donde los posibles valores son; 1: Aprobada, 2: Rechazada, 3: Cancelada y 4: En espera.	CHAR	1	-	NO
NumeroPedidoAutorizacion	Identifica el número de pedido asignado por el administrativo una vez se ha aprobado la compra de material por parte del operario	INT	-	-	SI
IdUsuario	Identificador del usuario. Usado por el operario para autentificarse en el sistema	VARCHAR	20	Foránea	NO
IdProveedor	Identifica el proveedor en el cual se va a realizar la compra de materiales	INT	-	Foránea	NO

Tabla 36 - Entidad Autorización

Entidad Proveedor

La entidad proveedor queda definida por los siguientes atributos, así como por sus restricciones:

Atributo	Descripción	Tipo	Longitud	Tipo clave	Nulos
IdProveedor	Identificador del proveedor dentro del sistema	INT	-	Primaria	NO
NombreProveedor	Nombre del proveedor	VARCHAR	100	-	NO

Tabla 37 - Entidad Proveedor

Como ya se ha comentado en anteriores capítulos, y a pesar de que la plataforma de despliegue está basada en tecnología **Java**, un conocido lenguaje de programación orientado a objetos, el desarrollo principal del aplicativo se realizará utilizando lenguajes propios de la programación web, y por lo tanto, los diagramas de clases no tienen sentido. No obstante, y para ofrecer una mejor compresión del alcance del proyecto, veremos una breve descripción de los métodos que se generarán en la capa de modelo, y que asegurarán el correcto acceso a los datos por parte de las capas de controlador y de vista.

9.4.1 Adaptadores. Capa de Modelo.

Para una mejor ordenación de los componentes del proyecto, se prevé crear un adaptador por cada una de las entidades mencionadas en el apartado anterior. Cada uno de estos adaptadores, deberá proveer al sistema con todos los métodos necesarios para la correcta recuperación, inserción, modificación y/o borrado de datos. Así pues, tendremos una agrupación lógica que llamaremos **Adaptador**, cuyo contenido será todos los métodos necesarios para acceder a cada una de las entidades, manteniéndose de esta forma una relación de 1:1 entre Adaptador y Entidad.

Adaptador	Descripción	Entidad	Métodos
UsuarioAdaptador	Contiene todos los métodos necesarios para el acceso a los datos de la entidad Usuario	Usuario	-boolean getUsuario(usuario)
IncidenciaAdaptador	Contiene todos los métodos necesarios para el acceso a los datos de la entidad Incidencia	Incidencia	-List getIncidenciasByUser(usr) -List getIncidenciasEstadisticas(inc) -List getIncidencias() -insertIncidencia(param1) -deleteIncidencia(param1) -updateIncidencia(param1)
EquipoAdaptador	Contiene todos los métodos necesarios para el acceso a los datos de la entidad equipo	Equipo	-List getEquipos()
MaterialAdaptador	Contiene todos los métodos necesarios para el acceso a los datos de la entidad material	Material	-List getMateriales() -insertMaterial(param1) -deleteMaterial(param1) -updateMaterial(param1)
ReservaAdaptador	Contiene todos los métodos necesarios para el acceso a los datos de la entidad reserva	Reserva	-List getReservasByUser(usr) -List getReservasByUser(usr) -insertReserva(param1) -deleteReserva(param1) -updateReserva(param1)
AutorizacionAdaptador	Contiene todos los métodos necesarios para el acceso a	Autorización	-List getAutorizacionesByUser(usr) -List getAutorizaciones()

	los datos de la entidad autorización		-insertAutorizacion(param1) -deleteAutorizacion(param1) -updateAutorizacion(param1)
ProveedorAdaptador	Contiene todos los métodos necesarios para el acceso a los datos de la entidad proveedor	Proveedor	-List getProveedores(param)

Tabla 38 - Adaptadores Capa Modelo

Adaptador de acceso a datos entidad Usuario

A continuación presentaremos un poco más en detalle cada uno de los métodos que conforman el adaptador de acceso a los datos de la entidad Usuario.

Adaptador	Método	Descripción
UsuarioAdaptador	boolean getUsuario(usuario)	Permite al sistema el obtener en una variable booleana con la existencia o no del usuario

Tabla 39 - Adaptador de acceso a datos entidad Usuario

Adaptador de acceso a datos entidad Incidencia

A continuación presentaremos un poco más en detalle cada uno de los métodos que conforman el adaptador de acceso a los datos de la entidad Incidencia.

Adaptador	Método	Descripción
IncidenciaAdaptador	List getIncidenciasByUser(usr)	Permite al sistema el obtener una lista de incidencias asignadas al usuario (operario) pasado por parámetro.
IncidenciaAdaptador	List getIncidenciasEstadisticas(inc)	Permite al sistema obtener una lista de incidencias dentro de un rango de fechas. Utilizada básicamente para el caso de uso: [CU14] Ver Estadística
IncidenciaAdaptador	List getIncidencias()	Permite al sistema obtener el total de incidencias del sistema.
IncidenciaAdaptador	insertIncidencia(param1)	Permite al sistema la inserción de una nueva incidencia. El atributo identificador no es necesario, ya que la propia base de datos lo añadirá incrementalmente.
IncidenciaAdaptador	deleteIncidencia(param1)	Uso Futuro.
IncidenciaAdaptador	updateIncidencia(param1)	Permite al operario cerrar las incidencias ya atendidas, así como, la modificación del equipo y la descripción. En algunos casos el equipo puede venir mal informado.

Tabla 40 - Adaptador de acceso a datos entidad Incidencia

Adaptador de acceso a datos entidad Equipo

A continuación presentaremos un poco más en detalle cada uno de los métodos que conforman el adaptador de acceso a los datos de la entidad Equipo.

Adaptador	Método	Descripción
EquipoAdaptador	List getEquipos()	Permite al sistema el obtener una lista de todos los equipos de la base de datos.

Tabla 41 - Adaptador de acceso a datos entidad Equipo

Adaptador de acceso a datos entidad Material

A continuación presentaremos un poco más en detalle cada uno de los métodos que conforman el adaptador de acceso a los datos de la entidad Incidencia.

Adaptador	Método	Descripción
MaterialAdaptador	List getMateriales()	Permite al sistema el obtener una lista de todos los materiales de la base de datos.
MaterialAdaptador	insertMaterial(param1)	Uso Futuro.
MaterialAdaptador	deleteMaterial(param1)	Uso Futuro.
MaterialAdaptador	updateMaterial(param1)	Permite al sistema la modificación del stock del material pasado por parámetro, de cara al control del mismo en el almacén privado de la compañía.

Tabla 42 - Adaptador de acceso a datos entidad Material

Adaptador de acceso a datos entidad Reserva

A continuación presentaremos un poco más en detalle cada uno de los métodos que conforman el adaptador de acceso a los datos de la entidad Reserva.

Adaptador	Método	Descripción
ReservaAdaptador	List getReservasByUser(usr)	Permite al sistema el obtener una lista de todas las reservas que un operario ha realizado.
ReservaAdaptador	List getReservasByUser(usr)	Permite al sistema el obtener una lista de todas las reservas.
ReservaAdaptador	insertReserva(param1)	Permite al sistema la inserción de una nueva reserva. El atributo identificador no es necesario, ya que la propia base de datos lo añadirá incrementalmente.
ReservaAdaptador	deleteReserva(param1)	Permite al sistema el borrado de una determinada reserva.
ReservaAdaptador	*updateReserva(param1)*	Permite al sistema la modificación de algunos atributos de la reserva, tales como la fecha de retorno, si se ha de retornar o no, la cantidad y el tipo de material.

Tabla 43 - Adaptador de acceso a datos entidad Reserva

Adaptador de acceso a datos entidad Autorización

A continuación presentaremos un poco más en detalle cada uno de los métodos que conforman el adaptador de acceso a los datos de la entidad Autorización.

Adaptador	Método	Descripción
AutorizacionAdaptador	List getAutorizacionesByUser(usr)	Permite al sistema el obtener una lista de todas las autorizaciones de compra emitidas por un operario.
AutorizacionAdaptador	List getAutorizaciones()	Permite al sistema el obtener una lista de todas las autorizaciones de compra.
AutorizacionAdaptador	insertAutorizacion(param1)	Permite al sistema la inserción de una nueva petición de autorización. El atributo identificador no es necesario, ya que la

		propia base de datos lo añadirá incrementalmente. Por otro lado el atributo estado, siempre será 4 (En espera) y finalmente el atributo número de pedido, el cual no será editable directamente por el operario, sino por el personal administrativo.
AutorizacionAdaptador	deleteAutorizacion(param1)	Permite al sistema el borrado de una determinada autorización de compra.
AutorizacionAdaptador	*updateAutorizacion(param1)*	Permite al sistema la modificación de algunos atributos de la petición de autorización, tales como descripción y el estado, en el supuesto caso de que el operario decida cancelar la petición.

Tabla 44 - Adaptador de acceso a datos entidad Autorización

Adaptador de acceso a datos entidad Proveedor

A continuación presentaremos un poco más en detalle cada uno de los métodos que conforman el adaptador de acceso a los datos de la entidad Proveedor.

Adaptador	Método	Descripción
ProveedorAdaptador	List getProveedores(param)	Permite al sistema el obtener una lista de todos los proveedores de la base de datos.

Tabla 45 - Adaptador de acceso a datos entidad Proveedor

9.5 Diagramas de Secuencia

Los diagramas de secuencia muestran las interacciones entre los diferentes elementos que constituyen el aplicativo. A continuación mostraremos a modo de ejemplo alguno de los diagramas de secuencia derivados de los casos de uso; [CU14] Ver Estadística y [CU02] Alta Incidencias. Cabe destacar que, ambos diagramas representan la actividad del operario en línea, es decir con conectividad al servidor. Los diagramas serían ligeramente diferentes si se tratara de una actividad en desconectado, sobre todo en lo referente a las interacciones entre el controlador y el Adaptador, que recordemos, es el encargado de acceder a los datos.

9.5.1 Diagrama de Secuencia – Ver Estadística

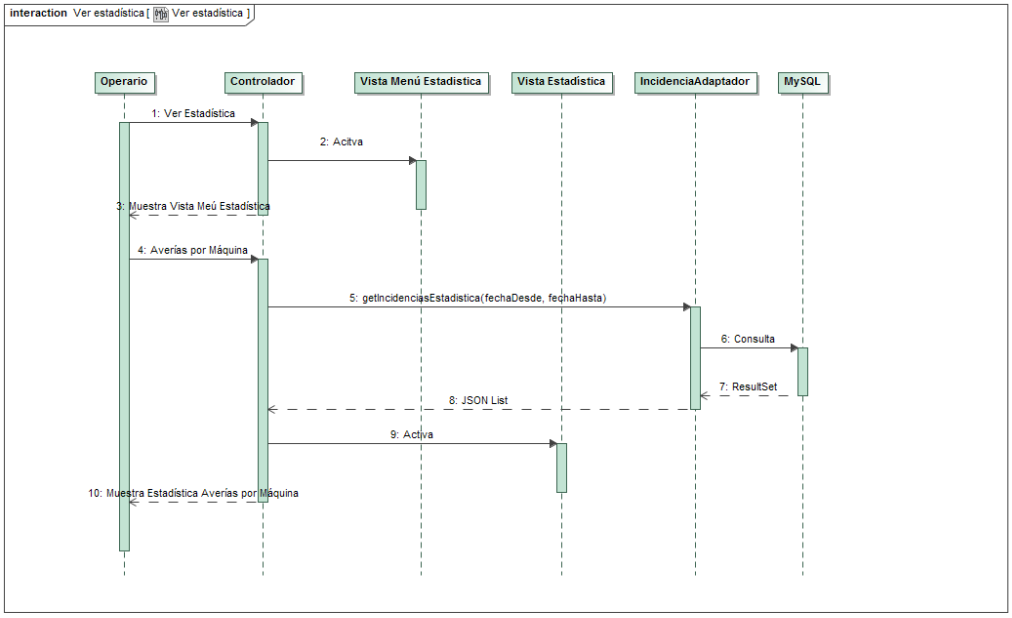


Ilustración 23 - Diagrama de Secuencia – Ver Estadística

Id	Mensaje	Descripción
1	Ver Estadística	El operario inicia el proceso de visualización de las estadísticas de 'Averías por Mes'
2	Activa	Al encontrarnos en un entorno no orientado a objetos, el controlador tan sólo la activa la vista correspondiente al 'Menú de Estadísticas'. Dichas vistas no se crean, sino que se ocultan o muestran en función de las acciones del usuario.
3	Muestra Vista Menú Estadística	El controlador muestra la vista del menú de estadísticas al operario
4	Averías por Máquina	El operario pulsa la opción de ver las estadísticas de 'Averías por Máquina'
5	getIncidencias(fechaDesde, fechaHasta)	El controlador invoca al método **getIncidencias()** del adaptador **IncidenciaAdaptador**, pasándole como parámetros la fecha desde y hasta sobre las cuales se quiere realizar la consulta
6	Consulta	El motor MySQL ejecuta la consulta de base de datos sobre la tabla **Incidencia:** SELECT * from Incidencia where fechaIncidencia >= fechaDesde AND fechaIncidencia <= fechaHasta
7	ResultSet	El sistema gestor base de datos retorna un ResultSet al adaptador
8	JSON List	El adaptador (ejecutado en la parte servidora) transforma el ResultSet en una lista de incidencias **JSON**
9	Activa	El controlador activa la vista de estadística, donde se podrán contemplar los datos en formato gráfica
10	Muestra Vista Estadística Averías por Máquina	El controlador muestra la vista de estadística de averías por máquina al operario

Tabla 46 - Diagrama de Secuencia – Ver Estadística

9.5.2 Diagrama de Secuencia – Alta Incidencia

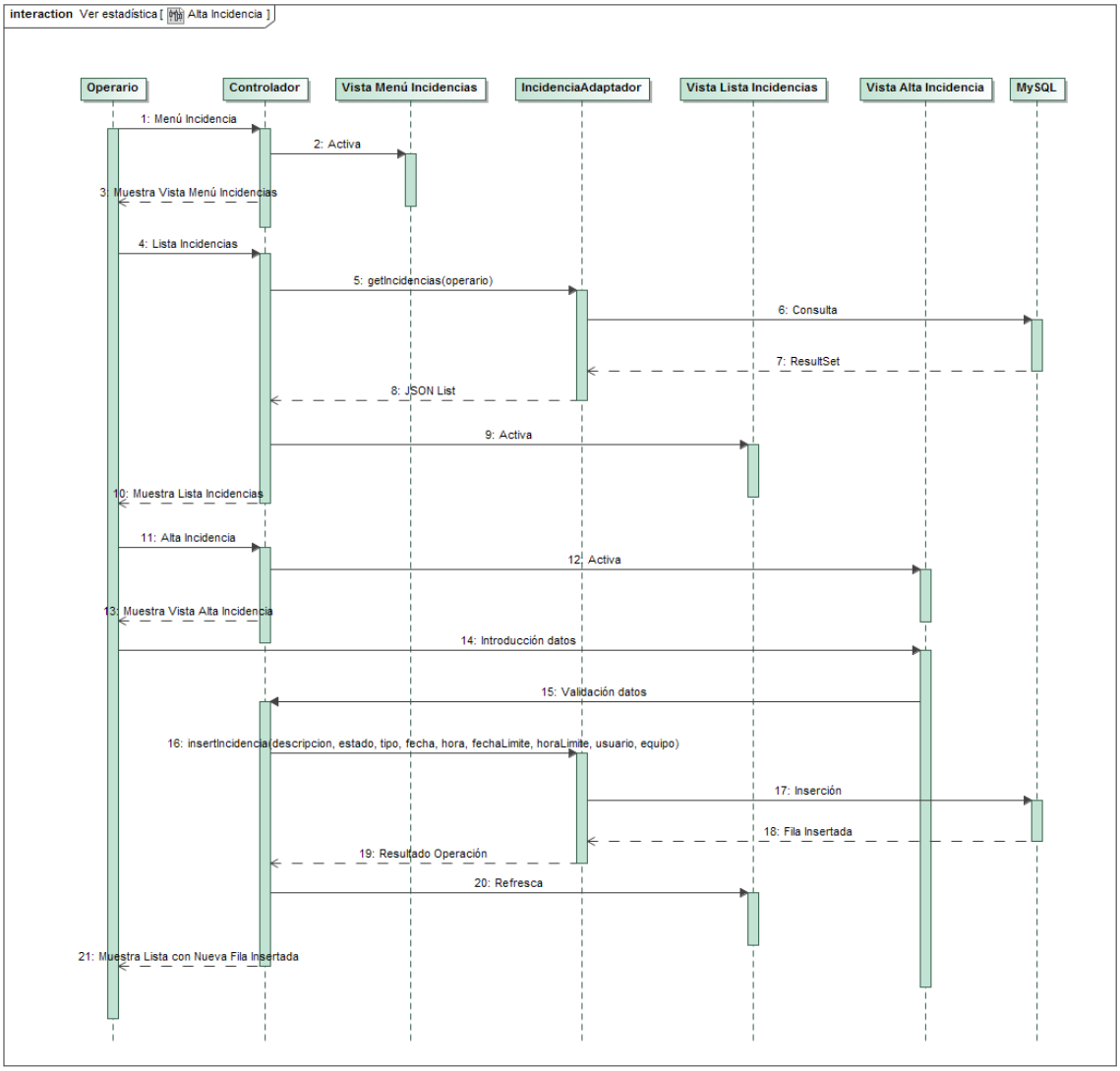


Ilustración 24 - Diagrama de Secuencia – Alta Incidencia

Id	Mensaje	Descripción
1	Menú Incidencia	El operario inicia el proceso de alta de incidencias a través del menú de incidencias
2	Activa	Al encontrarnos en un entorno no orientado a objetos, el controlador tan sólo la activa la vista correspondiente al 'Menú de Incidencias'. Dichas vistas no se crean, sino que se ocultan o muestran en función de las acciones del usuario.
3	Muestra Vista Menú Incidencias	El controlador muestra la vista del menú de incidencias al operario
4	Lista Incidencias	EL operario pulsa sobre la opción de Lista de Incidencias
5	getIncidencias(operario)	El controlador invoca al método **getIncidencias()** del adaptador **IncidenciaAdaptador**, pasándole como parámetro el operario sobre el cual se quiere realizar la consulta
6	Consulta	El motor MySQL ejecuta la consulta de base de datos sobre la tabla **Incidencia:** SELECT * from Incidencia where idUsuario=operario AND fechaIncidencia >= today

7	ResultSet	El sistema gestor base de datos retorna un ResultSet al adaptador
8	JSON List	El adaptador (ejecutado en la parte servidora) transforma el ResultSet en una lista de incidencias **JSON**
9	Activa	El controlador activa la vista de lista de incidencias
10	Muestra Vista Estadística Averías por Máquina	El controlador muestra la vista de lista de incidencias al operario
11	Alta Incidencia	El operario pulsa sobre el botón de 'Alta Incidencia'
12	Activa	El controlador activa la vista de 'Alta de Incidencia', done todos los valores aparecerán por defecto. A excepción de los campos de tipo fecha y hora que aparecerán con la fecha y hora actual, el resto de campos se mostrarán en blanco
13	Muestra Vista Alta Incidencia	El controlador muestra la vista de alta de incidencia al operario
14	Introducción de Datos	El operario introduce los datos correspondientes a la incidencia y pulsa el botón de 'Aceptar'
15	Validación de Datos	La misma vista valida que los datos sean correctos, en cuanto al formato y obligatoriedad se refiere.
16	insertIncidencia(descripcion, estado, tipo, fecha, hora, fechaLimite, horaLimite, usuario, equipo)	El controlador invoca al método **insertIncidencia()** del adaptador **IncidenciaAdaptador**, pasándole como parámetros la descripción, el estado y el tipo de la incidencia, la fecha y hora de la incidencia, la fecha y hora límite de resolución de la incidencia, el operario que la reporta y el equipo averiado
17	Inserción	El motor MySQL ejecuta la sentencia sobre la tabla **Incidencia:** INSERT INTO Incidencia VALUES(descripcion, estado, 'I', fecha, hora, fechaLimite, horaLimite, usuario, equipo)
18	Fila Insertada	El sistema gestor de base de datos retorna un mensaje al adaptador con el éxito o no de la operación de inserción
19	Resultado Operación	El adaptador retorna al controlador el resultado de la operación de inserción. En caso de que la operación hay sido exitosa, entonces el controlador seguirá con el paso 20. En el caso contrario, es decir, si la operación de inserción falla, entonces se deberá mostrar un error.
20	Refresca	El controlador refresca la vista de la lista de incidencias incluyendo a la nueva incidencia
21	Muestra Lista con Nueva Fila Insertada	El controlador muestra la vista de lista de incidencias al operario

Tabla 47- Diagrama de Secuencia – Alta Incidencia

En esta sección se presentarán las características principales que definen la interfaz de usuario, así como, se verán algunas de las funcionalidades más representativas del aplicativo a través de un completo conjunto de pantallas, que asegurarán una mejor perspectiva final por parte del cliente.

Una de las razones por las cuales se ha optado por el uso de una plataforma propietaria como es el caso de **IBM Worklight**, es por el alto grado de integración de sus componentes. Así pues, para el desarrollo del prototipo, y siempre pesando en el aprovechamiento del código generado, todo el desarrollo del mismo se ha realizado usando el componente **IBM Worklight Studio**, siendo éste un entorno de desarrollo basado en **Eclipse**[19]**,** que incorpora ***Plugins***[20] específicos para el desarrollo de aplicaciones híbridas para dispositivos móviles integradas con **Dojo Mobile.**

Finalmente mencionar que las pruebas se han realizado directamente sobre un navegador **Firefox**[21], donde gracias a otro componente del producto llamado **Simulador de Navegador Móvil**, se ha podido ver a medida que avanzaba el desarrollo del prototipo como iban quedando las pantallas sin necesidad de desplegar cada vez el aplicativo en una plataforma nativa **Android**. Sin embargo, y para tener una visión más realista de la apariencia final del producto, se ha usado una herramienta nativa llamada **Screenshot Ultimate**[22] que permite tomar 'instantáneas' de las pantallas en un entorno de funcionamiento real. A continuación veremos dichas pantallas y mencionaremos las funcionalidades que ofrecen.

Ilustración 26 - Pantalla de Splash

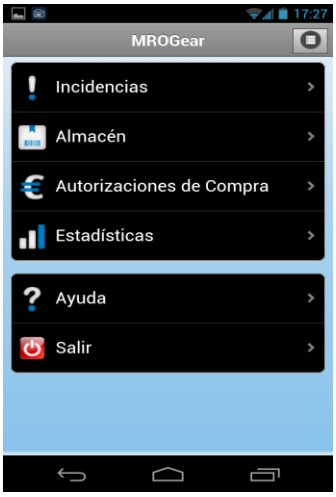


Ilustración 25 - Menú Principal

Tras el proceso de autentificación, el aplicativo necesita de unos instantes para poder cargarse completamente. Durante estos instantes se visualiza una pantalla de *'splash',* la cual se puede ver en la ilustración número 13. Al acabarse el proceso de carga del aplicativo, la primera pantalla que nos aparece es la del menú principal. Desde el citado menú, se podrá navegar hasta las diferentes funcionalidades ofrecidas por el aplicativo.

Adicionalmente, ofrece la posibilidad de desplegar un menú emergente situado en la parte inferior de la pantalla, desde donde se podrá invocar a la funcionalidad de **sincronización**, en el supuesto caso de estar trabajando en modalidad desconectada. Desde este mismo menú, se podrá ver el nombre, autor y versión del aplicativo a través de la opción de **información** y finalmente con la tercera de las opciones, podremos **ocultar** el menú emergente, haciéndolo desaparecer de la vista. Para el despliegue del menú emergente se usará el botón situado en la parte superior derecha del aplicativo. Las siguientes ilustraciones muestran el menú emergente dentro del menú principal y la pantalla **opción Info**, dentro del menú emergente.

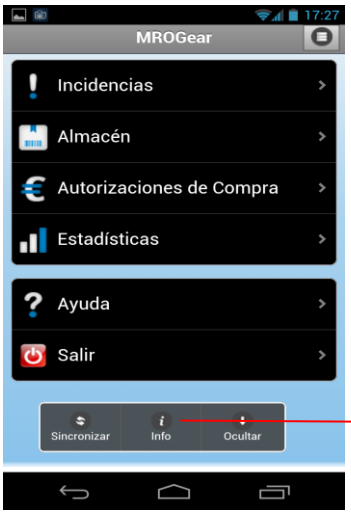


Ilustración 28 - Menú Emergente

Ilustración 27 - Información del sistema

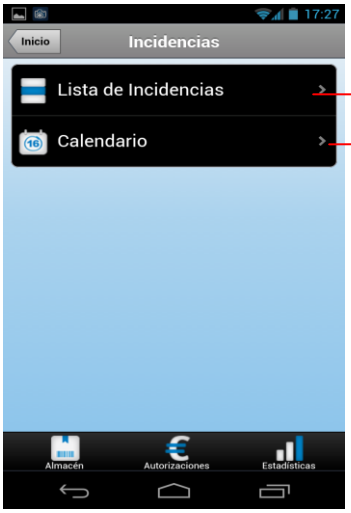


Ilustración 29 - Menú Incidencias

Ilustración 30 - Lista de Incidencias

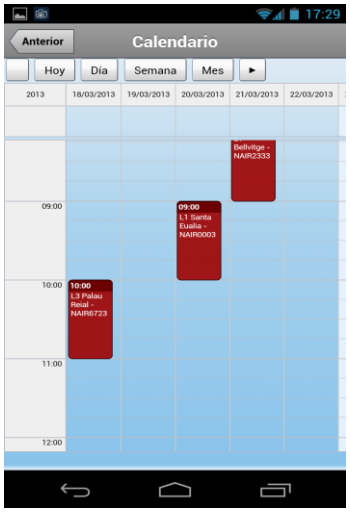
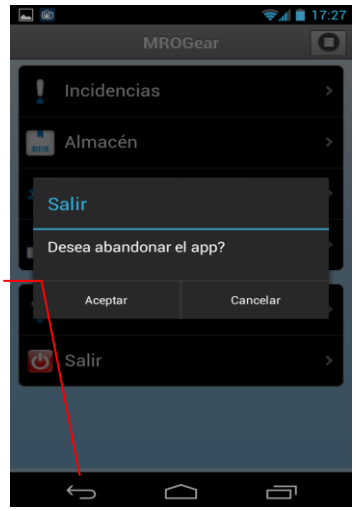


Ilustración 31 - Calendario

Desde el menú de incidencias podremos acceder a dos funcionalidades básicas en el aplicativo, a través de la pulsación sobre la lista de opciones:

1. Lista de las incidencias: visualizaremos una lista completa de las incidencias y preventivos que han sido asignados al operario (Ilustración 17).
2. Calendario: visualizaremos los preventivos en formato de calendario (Ilustración 18).



Botón de atrás físico del dispositivo

Ilustración 32 - Mensaje Modal

En la parte superior del menú de incidencias aparece el **botón Inicio**, a través del cual podremos volver al menú principal. Es importante reseñar aquí (es un comportamiento que se repite en el resto de pantallas) que el **botón de volver atrás del dispositivo físico** hace abandonar el aplicativo, previa visualización de una pantalla modal de confirmación como la que se ve en la ilustración de la izquierda.

Finalmente en la parte inferior de la pantalla aparecen tres opciones identificadas como; **Almacén, Autorizaciones y Estadísticas.** Pulsando en cada una de ellas, accederemos directamente a los respectivos menús sin necesidad de volver al menú principal para ello.

En la vista **Lista de Incidencias**, podemos ejecutar dos acciones:

1. Crear una nueva incidencia: a través del **botón '+'** accederemos a una pantalla donde el operario podrá introducir todos los datos necesarios para creación de una nueva incidencia.
2. Visualizar/Modificar una incidencia: pulsando directamente sobre cualquiera de las líneas que conforman la lista de incidencias, se podrá acceder a la visualización o edición de la incidencia en cuestión.

Adicionalmente si pulsamos el **botón Anterior**, volveremos al menú de incidencias descrito con anterioridad.

En la vista de **Calendario**, los botones situados en la parte superior serán usados para:

1. Flecha hacia la izquierda: nos permitirá ver la planificación de preventivos de la semana anterior.
2. Flecha hacia la derecha: nos permitirá ver la planificación de preventivos de la semana siguiente.
3. Día/Hoy: nos permitirán ver la planificación de preventivos en formato de día completo, **Día** sobre un día señalado y **Hoy** sobre la fecha actual.
4. Semana: nos permitirá volver a recuperar la vista de calendario semanal, en caso de haberla cambiado con los botones de **Hoy** ó **Mes**.
5. Mes: nos permitirá ver la planificación de preventivos en formato mensual.

10.3 Prototipo Pantalla Detalle de Incidencias

Ilustración 34 - Detalle Incidencia (Arriba)

Ilustración 33 - Detalle Incidencia (Abajo)

A la pantalla de Detalle de Incidencias se llega a través de la pulsación sobre el **botón '+'** de la pantalla de **Lista Incidencias** descrita en el apartado anterior, o bien a través de la pulsación directa sobre una de las líneas de incidencia. Como en el resto de pantallas del aplicativo, el **botón Anterior** situado en la parte superior izquierda nos servirá para acceder a la pantalla anterior, en este caso a la lista de incidencias.

Se puede observar que para evitar al máximo la escritura por parte del operario, los campos de entrada de la pantalla incorporan componentes gráficos que facilitan la

introducción de los datos. Todos los componentes se posicionan sobre una pantalla con *scroll*, donde al final de la misma estará el **botón de Aceptar**, que sirve para validar y aceptar los datos introducidos.

Además se incorporan dos funcionalidades adicionales; el **botón QR** nos abrirá una nueva pantalla para poder hacer una lectura de un código QR, permitiéndose así recuperar la información dentro de éste (identificador de la máquina). Por otro lado el **botón Mapa**, con el cual podremos ubicar la incidencia en un mapa **Google Maps**. Las siguientes ilustraciones muestran las respectivas pantallas de ambas funcionalidades.

Ilustración 35 - Pantalla lectura código QR

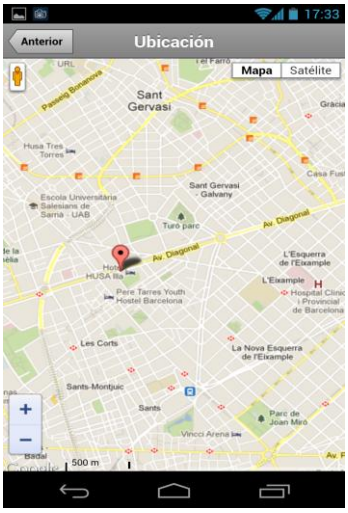


Ilustración 36 - Pantalla Google Maps

En el supuesto caso de que el operario esté trabajando en modalidad desconectada (sin conexión a Internet) y quisiera ver el mapa de situación de una avería, el siguiente mensaje de error le será presentado:

Ilustración 37 - Error de Conexión

Ilustración 39 - Menú Almacén

Ilustración 40 - Lista Almacén

Ilustración 38 - Detalle Almacén

A la opción de almacén se accede directamente desde el menú principal. En ella se presentarán los materiales disponibles en almacén agrupados en 6 categorías diferentes; Medición, Mano, Tornillería, Eléctrico, Fontanería, Otros... Si pulsamos sobre el **botón Ver** de cualquiera de las categorías se nos presentará una **Lista de materiales** que la conforman. Para desplazarse entre categorías simplemente hemos de hacer un movimiento lateral del dedo a izquierda o derecha en función de la categoría que estamos buscando (*scroll horizontal*). Un indicador situado justo debajo del título del menú, nos indica en todo momento en que categoría nos encontramos. Como en anteriores casos si pulsamos sobre el **botón de Inicio,** accederemos de nuevo al menú principal.

Por otro lado en la parte inferior del Menú de Almacén aparecen tres opciones identificadas como; **Incidencias, Autorizaciones y Estadísticas.** Pulsando en cada una de ellas, accederemos directamente a los respectivos menús sin necesidad de volver al menú principal para ello.

En cuanto a la vista de materiales, que recordemos se ha accedido a ella tras pulsar el **botón Ver** de una categoría, se presenta una lista de materiales que pueden ser seleccionados pulsando encima de uno de ellos. El número situado a la derecha de cada una de las líneas representa la cantidad de stock del material en cuestión. Si seleccionamos uno de los materiales se nos presenta una pantalla donde el operario puede reservar al cantidad que estime oportuna. Tras pulsar el **botón Aceptar,** el operario podrá ver sus reservas tal y como se explica en el siguiente punto.

Como en los casos previos los **botones Anterior**, ubicados en la parte superior izquierda de las pantallas de **Lista de Almacén** y **Detalle Almacén**, nos permitirán volver al menú de almacén y a la lista de materiales respectivamente.

Ilustración 43 - Menú Mis Reservas

Ilustración 41 - Lista Mis Reservas

Ilustración 42 - Detalle Mis Reservas

Para acceder a las reservas que el operario ha realizado, se debe desplazar todas las categorías hacía la izquierda. Una vez aparece la categoría especial de '**Mis Reservas**', se debe pulsar el **botón Ver** para acceder a la lista de reservas realizadas. Como en anteriores casos, si pulsamos sobre cualquiera de las líneas nos aparecerá la pantalla de detalle de la reserva. En ella podremos modificar cualquier dato que se considere inapropiado, y adicionalmente podremos borrar la petición de reserva pulsando el **botón '-'**, situado en la parte superior derecha de la pantalla de Detalle Reserva Material.

Como ya se ha explicado anteriormente los **botones Anterior**, ubicados en la parte superior izquierda de las pantallas de **Mis Reservas** y **Detalle Reserva Material**, nos permitirán volver al menú de almacén. Igualmente se han diseñado los elementos con componentes gráficos, de cara a facilitar la tarea al operario.

Finalmente en la parte inferior de la pantalla aparecen tres opciones identificadas como; **Incidencias, Autorizaciones y Estadísticas.** Pulsando en cada una de ellas, accederemos directamente a los respectivos menús sin necesidad de volver al menú principal para ello.

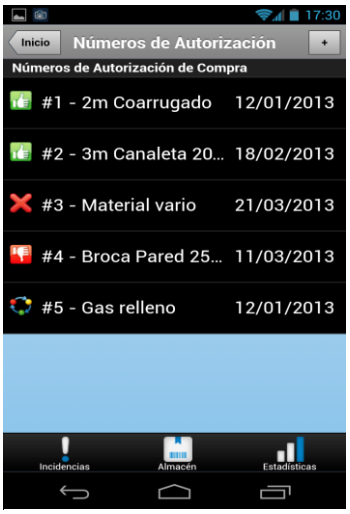


Ilustración 45 - Lista Autorizaciones de Compra

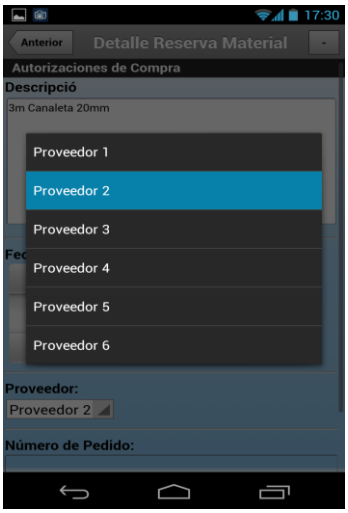


Ilustración 44 - Selección Proveedor

Tras la pulsación de la **opción Autorizaciones de Compra** del menú principal, accedemos a la pantalla de lista de autorizaciones. Como en los casos anteriores, si pulsamos sobre cualquiera de las líneas se nos presenta una pantalla donde podremos editar los valores de la autorización. En la ilustración anterior se presenta una lista desplegable con valores fijos, que se corresponde con el campo de proveedor. Adicionalmente podremos cancelar una petición de autorización de compra, mediante el **bóton '-'**, situado en la parte superior derecha de la pantalla Detalle Autorizaciones. De igual manera, si el deseo del usuario del aplicativo es la creación de una nueva petición de autorización, se tendrá que pulsar el **botón '+'**, situado en el margen superior derecho de la pantalla lista de números de autorización.

Como en anteriores casos, los **botones Inicio y Anterior**, ubicados en la parte superior izquierda de las pantallas de **Números de Autorización** y **Detalle Número de Autorización**, nos permitirán volver respectivamente al menú principal y a la lista de peticiones de autorización de compra.

Finalmente en la parte inferior de la pantalla aparecen tres opciones identificadas como; **Incidencias, Almacén y Estadísticas.** Pulsando en cada una de ellas, accederemos directamente a los respectivos menús sin necesidad de volver al menú principal para ello.

10.7 Prototipo Menú Estadísticas

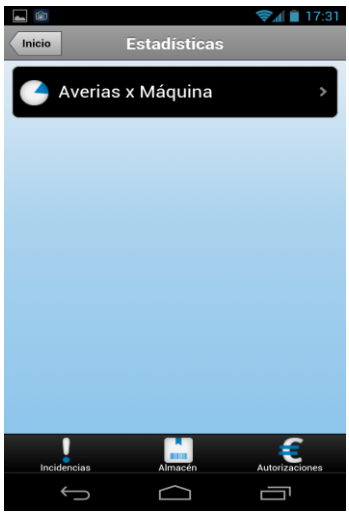


Ilustración 46 - Menú Estadísticas

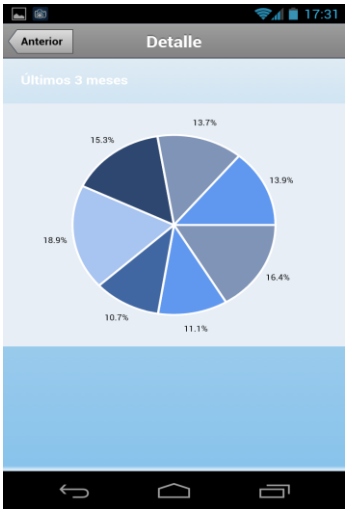


Ilustración 47 - Averías x Máquina

Tras la pulsación sobre la **opción Estadísticas** del menú principal accedemos al menú de estadísticas. El **botón Inicio** nos permitirá acceder de nuevo al menú principal, mientras que si pulsamos sobre la **opción 'Averías x Máquina'** accedemos a visualizar en un formato gráfico las máquina que han sufrido un mayor número de averías en los últimos tres meses. Con el **botón Anterior**, del margen superior izquierdo de la pantalla de Detalle, accederemos de nuevo al menú de estadísticas.

Igualmente en la parte inferior de la pantalla aparecen tres opciones identificadas como; **Incidencias, Almacén y Autorizaciones.** Pulsando en cada una de ellas, accederemos directamente a los respectivos menús sin necesidad de volver al menú principal para ello.

10.8 Prototipo Ayuda

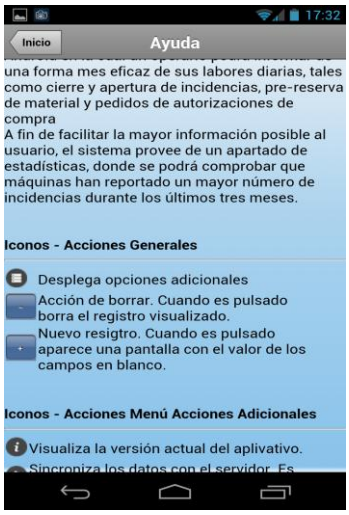


Ilustración 48 - Ayuda

Tras la pulsación sobre la **opción Ayuda** del menú principal accedemos a la pantalla de ayuda, donde se explica el objetivo del aplicativo y además se muestra el significado de los iconos e indicadores que aparecen a lo largo del mismo. Pulsando el **botón Inicio**, retornamos al menú principal.

A lo largo de esta sección inicialmente estableceremos las premisas del proyecto y como estas han afectado al desarrollo y a continuación daremos respuesta, desde un punto de vista generalista, a las siguientes preguntas; ¿Cómo se ha realizado el desarrollo de la aplicación **MROGear**?, ¿Qué herramientas se han utilizado para completar con éxito la implementación del mismo?

11.1 Premisas de la Implementación

Como ya se ha comentado en secciones y apartados anteriores, el uso de determinados componentes –gráficos y no gráficos-, como se verá en el siguiente apartado, siempre han ido enfocados a mejorar de la usabilidad de la aplicación, teniendo en cuenta las dificultades que un usuario de la misma se encontrará a diario.

11.1.1 Uso de Componentes visuales

Uno de los factores de éxito, en cuanto a la utilización o no de la aplicación, ha de ser la facilidad de uso y la amigabilidad de la interfaz de usuario. Para potenciar estos aspectos, se ha intentado a la hora de diseñar la misma, el uso de componentes visuales, de mayor tamaño a los clásicos campos de entrada de datos y con introducción de valores por pulsaciones, adecuados a las difíciles condiciones de trabajo del operario. A continuación veremos algunos de estos componentes. Cabe destacar que la gran mayoría de ellos (no todos) pertenecen al propio *framework* de desarrollo **Dojo Mobile[12].**

De cara a facilitar la introducción de horas y fechas, se ha optado por el uso de los componentes; **ValuePickerTimePicker[23]** y **ValuePickerDatePicker**[24] respectivamente. La característica principal que ha determinado el uso de los mismos en el proyecto, ha sido la facilidad de uso, gracias a la introducción o cambio de valores por pulsaciones (botones +/-).

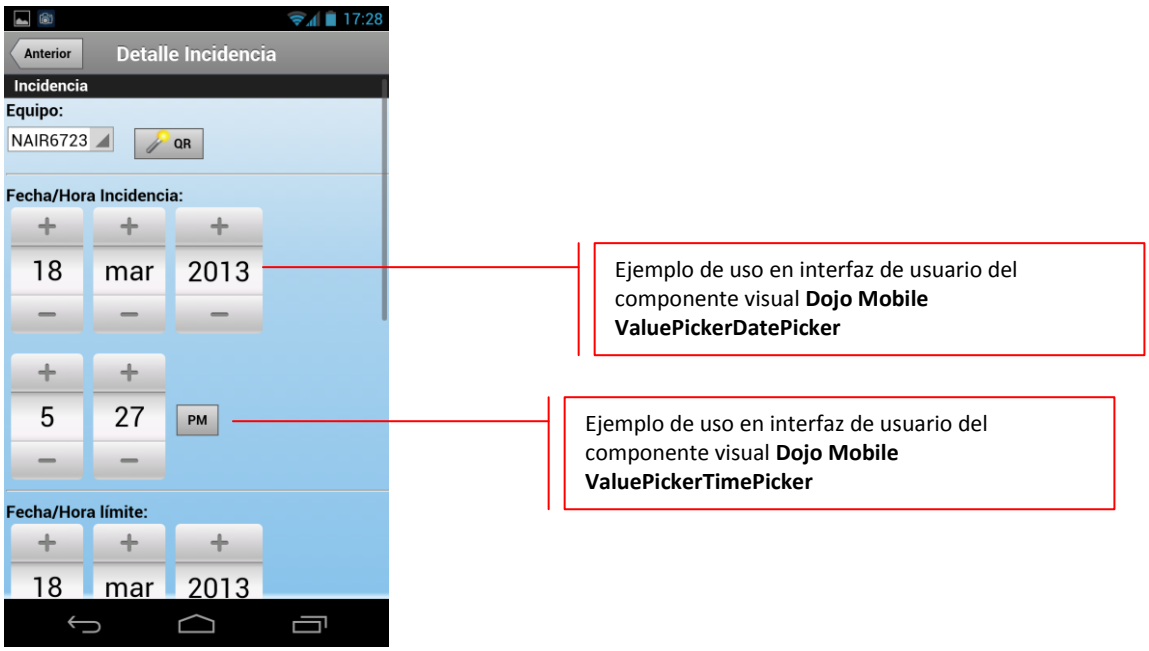


Ilustración 49 - Componentes visuales para fechas y horas

De igual manera, de cara a minimizar las pulsaciones por teclado, todos los campas de tipo numérico que reflejan cantidades se han representado en la interfaz como componentes de tipo visual **ValuePickerSlot**[25]**.** Las características de dichos componentes son muy parecidas a los citados en el párrafo anterior.

Ilustración 50 - Componentes visuales para valores numéricos

Además del uso de los mecanismos propios del *framework* de desarrollo para dispositivos móviles **Dojo Mobile**[12], se han usado desarrollos de terceros para implementar alguna de las funcionalidades de la aplicación. Nuevamente, de cara a minimizar la introducción de datos por parte del operario en la aplicación, se ha dotado al sistema de un mecanismo de lectura de **códigos QR**[3] que hace uso de la cámara del propio dispositivo.

La citada funcionalidad se apoya en un software libre de terceros llamado **BarcodeScanner**[26] y que está publicado en forma de *plug-in* para **Apache Cordova**[11] en la comunidad de código abierto **GitHub**[27]. A continuación veremos una pantalla con la implementación real de este mecanismo.

Ejemplo de uso en interfaz de usuario de *plug-in* para Apache Cordova llamado **BarcodeScanner**

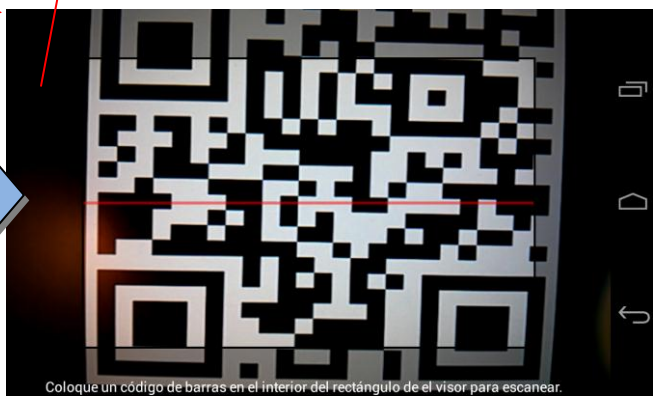


Ilustración 51 - Botón acción código QR **Ilustración 52 - Lectura Código QR**

11.1.2 Modalidad de trabajo desconectada

Una de los principales problemas con los que se encuentra el usuario de la aplicación es la falta de conexión a la red en determinados emplazamientos. Dado que, en la mayoría de casos las máquinas con las que el operario ha de tratar, están ubicadas en lugares fuera de cobertura de red, y por lo tanto, fuera de cualquier conectividad con el servidor, es necesaria la implementación de un mecanismo que permita mitigar o eliminar dicho problema. En este punto es importante recordar que, el componente dentro de la arquitectura de **IBM Worklight**[2], que se encarga de dotar de acceso a datos es el Adaptador (ver Ilustración 21 – Arquitectura lógica de la plataforma servidora). Dichos adaptadores, se ejecutan en la parte servidora y por lo tanto cualquier modificación, entrada o borrado de datos, necesita forzosamente de conectividad entre el cliente (dispositivo móvil) y el servidor.

Así pues como se ha comentado antes, la falta de conectividad en ciertas localizaciones del operario impacta fuertemente en el correcto funcionamiento de la aplicación. Para solventar el problema descrito, se ha implementado un mecanismo por el cual todos los cambios realizados por el operario se guardan en la memoria de disco del propio dispositivo, para más tarde cuando el usuario recupere la conectividad, poder sincronizar los datos entre dispositivo y servidor. El componente que implemente esta funcionalidad se llama **JSONStore**[28], incluido en **IBM Device Runtime** del producto **IBM Worklight**[2].

Existen varios mecanismos de sincronización, pero todos son dependientes de la conectividad con el servidor; en primer lugar el operario puede hacer una sincronización manual a través de la opción del menú emergente (ver ilustración siguiente), en segundo lugar cada vez que abandona la aplicación el sistema realiza una sincronización automática y finalmente cada vez que inicia la aplicación los datos se sincronizan automáticamente.

Ilustración 53 - Desarrollo Menú Emergente para sincronización de datos

11.1.3 Depuración del código

Uno de los aspectos fundamentales para la solución de posibles errores, es la planificación e implementación de mecanismos de depuración de código. De cara a facilitar el posterior mantenimiento y depuración de errores se han usado mecanismos de anotación por consola propios del producto. Abajo vemos un trozo de código donde se puede apreciar el uso de anotaciones, así como un ejemplo de salida por consola de la misma.

```
// CARGA DE DATOS INICIAL EN SERIE (NO PARALELO)*************************************************************************
                    WL.Device.getNetworkInfo(function (info) {
                        if (info.isNetworkConnected=='true'){
                            var options = {
                                onSuccess: function (){
                                    WL.Logger.debug("[MROGear] Conectado al Servidor.");
                                    loadIncidenciasFromServer();},
                                onFailure: function (){
                                    WL.Logger.debug("[MROGear] No conexión al Servidor. WiFi Activo, posiblemente fuera de
                                    initLoadIncidencias();},
                            };
                            WL.Client.connect(options);
                        }else{
                            WL.Logger.debug("[MROGear] No conexión al Servidor. Sin cobertura WiFi.");
                            initLoadIncidencias();
                        }
                    });
```

```
D  05-04 14:27:12.975  6928  6983  com.MROGear  dalvikvm                 WAIT_FOR_CONCURRENT_GC blocked 9ms
D  05-04 14:27:12.985  6928  6984  com.MROGear  dalvikvm                 WAIT_FOR_CONCURRENT_GC blocked 9ms
D  05-04 14:27:13.005  6928  6931  com.MROGear  dalvikvm                 GC_CONCURRENT freed 512K, 12% free 9420K/10596K, paused 3ms+1ms, to
                                                                         tal 17ms
D  05-04 14:27:13.005  6928  6983  com.MROGear  dalvikvm                 WAIT_FOR_CONCURRENT_GC blocked 11ms
D  05-04 14:27:13.015  6928  6984  com.MROGear  dalvikvm                 WAIT_FOR_CONCURRENT_GC blocked 15ms
D  05-04 14:27:13.045  6928  6931  com.MROGear  dalvikvm                 GC_CONCURRENT freed 439K, 12% free 9397K/10596K, paused 3ms+2ms, to
                                                                         tal 20ms
D  05-04 14:27:13.085  6928  6941  com.MROGear  MROGear                  [MROGear] No conexi♦n al Servidor. Sin cobertura WiFi.
W  05-04 14:27:13.235  509   764               InputMethodManagerService Window already focused, ignoring focus gain of: com.android.interna
                                                                         l.view.IInputMethodClient$Stub$Proxy@41fb8170 attribute=null, token
                                                                          = android.os.BinderProxy@41fd9fb0
```

Ilustración 54 - Ejemplo desarrollo de anotación por consola

11.1.4 Posicionamiento

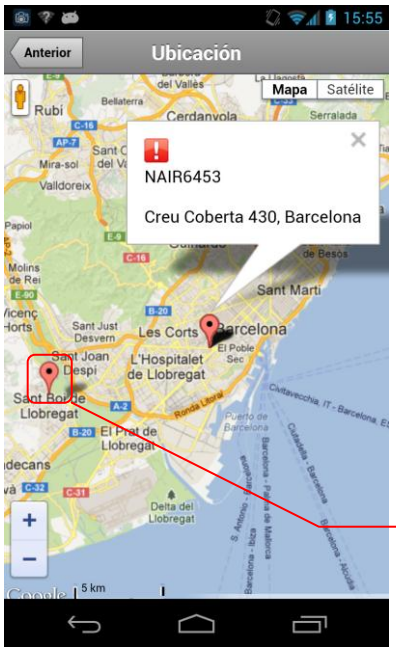


Ilustración 55 - Google Maps con posicionamiento de incidencias

Adicionalmente comentar que, con el fin de proporcionar el mayor número de información útil al operario y de la forma más gráfica posible, se ha implementado un servicio de posicionamiento de averías utilizando la *API* que **Google Maps**[29] provee. En dicho servicio se representa en un mapa, tanto la ubicación de la avería, como el posicionamiento actual del operario. Para la geolocalización del operario se han aprovechado las características **GPS**[30] del dispositivo móvil. Nuevamente un *plugin* de **Apache Cordova**[11] expone en coordenadas de latitud y longitud la localización del usuario.

Muestra el posicionamiento actual del operario, permitiéndole ver a este la distancia que le separa con respecto de la avería que tiene que solventar.

11.1.5 Multi-idioma

Para hacer de la aplicación lo más manejable y cercana posible al usuario, se han implementado funcionalidades de cambio de idioma en la interfaz gráfica. Las siguientes ilustraciones muestran los controles que nos permiten cambiar de idioma desde la pantalla de inicio de sesión. A continuación se muestra un detalle del código que recoge dinámicamente el cambio de lenguaje en la plataforma:

Ilustración 56 – Control cambio Idioma

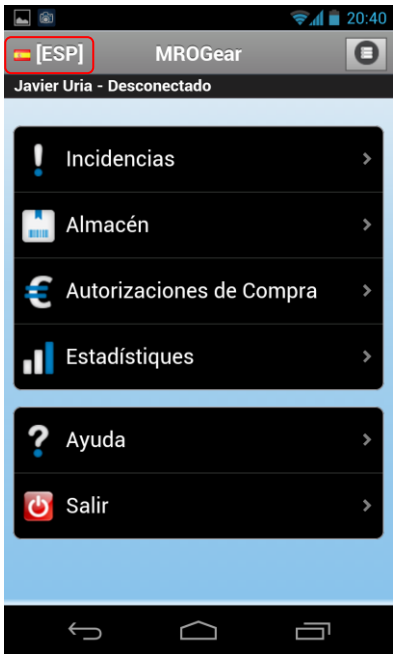


Ilustración 57 – Idioma sesión

```
function setCatala(){
    Messages.ListMenuPrin_incidencias="Incidéncies";
    Messages.LoginHeader="Inici Sessió";
    Messages.usrPlaceholder="Usuari";
    Messages.pwdPlaceholder="Contrasenya";
    Messages.okLogin="Acceptar";
    Messages.cancelLogin="Cancel·lar";
    Messages.initButton="Inici";
    Messages.viewButton="Veure";
    Messages.backButton="Anterior";
    Messages.menuFijoAlmacen="Magatzem";
    Messages.menuFijoAutorizaciones="Autoritzacions";
    Messages.menuFijoEstadisticas="Estadístiques";
    Messages.menuFijoIncidencias="Incidéncies";
```

Ilustración 58 - Función cambio dinámico idioma Catalán

11.1.6 Modalidad de pruebas

Finalmente para facilitar la tarea de prueba y depuración, tanto del usuario como del desarrollador, se ha habilitado la funcionalidad de 'Modalidad de Pruebas'. En dicha modalidad no se requiere conexión al servidor para poder navegar entre un conjunto de datos no reales. Dicha modalidad se activa si la primera vez que se instala la aplicación no se tiene conectividad con el servidor o bien manualmente a través de

la opción de menú 'Info' del menú emergente en la pantalla principal. Las siguientes ilustraciones muestran los controles para activar y desactivar la modalidad de test así como, una fracción de código donde se muestra en qué punto del programa se controla el cambio de estado.

Ilustración 59 - Control Modalidad Test

Ilustración 60 - Control Modalidad Test

```
// ACTIVAMOS / DESACTIVAMOS  TEST MODE ***************************************************************************************
                var showTestMode = registry.byId("showTestMode");
                on(showTestMode, "click",
                    function(){
                        if (localStorage.getItem("testMode")!=null){
                            registry.byId("testDlgActTestModeSwitch").set("value", "on");
                            registry.byId("testDlgUserSessionText").set("value", "Test");
                            firstOnChangeSwitch="on";
                        }else{
                            registry.byId("testDlgActTestModeSwitch").set("value", "off");
                            registry.byId("testDlgUserSessionText").set("value", localStorage.getItem('userSession'));
                            firstOnChangeSwitch="off";
                        }
```

11.2 Proceso de Implementación y Entorno de desarrollo

A lo largo de esta sección se mostrarán aspectos relacionados con el proceso de desarrollo e implementación de la aplicación. El objetivo general, es hacer entender mejor que componentes de software han intervenido durante todo el proceso, así como, dar respuesta a las preguntas presentadas en el prólogo del capítulo; ¿Cómo se ha realizado el desarrollo de la aplicación **MROGear**?, ¿Qué herramientas se han utilizado para completar con éxito la implementación del mismo?

11.2.1 Herramientas de desarrollo

A continuación expondremos una breve descripción de las herramientas de desarrollo que se han utilizado a lo largo del proceso de implementación de la aplicación. Dividiremos en dos grupos las mismas; aquellas que han intervenido exclusivamente en el desarrollo del código fuente de la aplicación y aquellas que han servido para el desarrollo, mantenimiento y generación de *scripts* de la base de datos.

Como se cita con anterioridad dentro del primer grupo de herramientas, las utilizadas para el desarrollo de la aplicación, nos encontramos con; **IBM Worklight Studio**[2]. Dicha herramienta es un entorno de desarrollo basado en **Eclipse**[19]. A continuación presentamos una captura de pantalla del mismo.

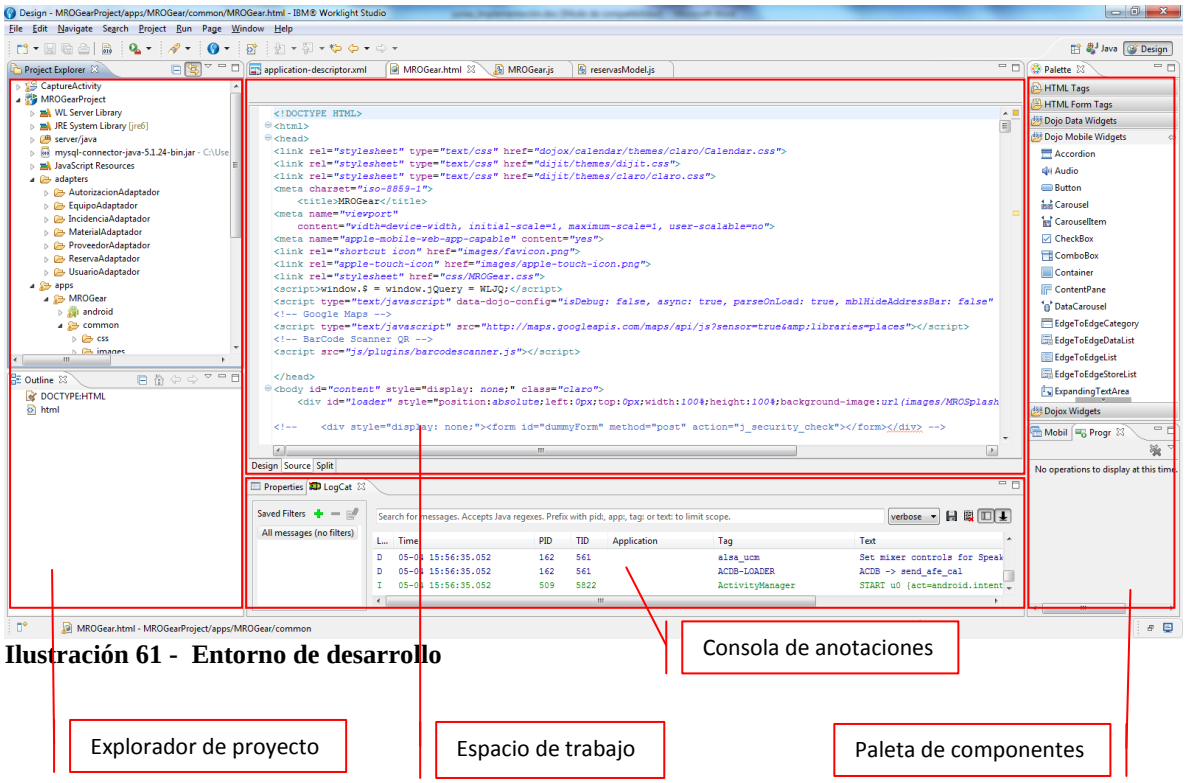


Ilustración 61 - Entorno de desarrollo

Como se puede observar el alto grado de integración de **IBM Worklight Studio**[2] con **Dojo Mobile**[12], permite la implementación de la interfaz de usuario a través de una paleta de componentes (se visualiza en la parte derecha de la pantalla), utilizando las caracteríticas de '*drag & drop*' típicas de los entornos de desarrollo mas modernos.

Para el segundo de los grupos, las herramientas dedicadas a la generación y mantenimiento de la base de datos, principalmente se ha utilizado una herramienta propietaria de libre distribución llamada, **IBM Data Studio**[31], y de la cual presentamos una captura de pantalla a continuación. Como se puede observar también es una herramienta basada en **Eclipse**[19].

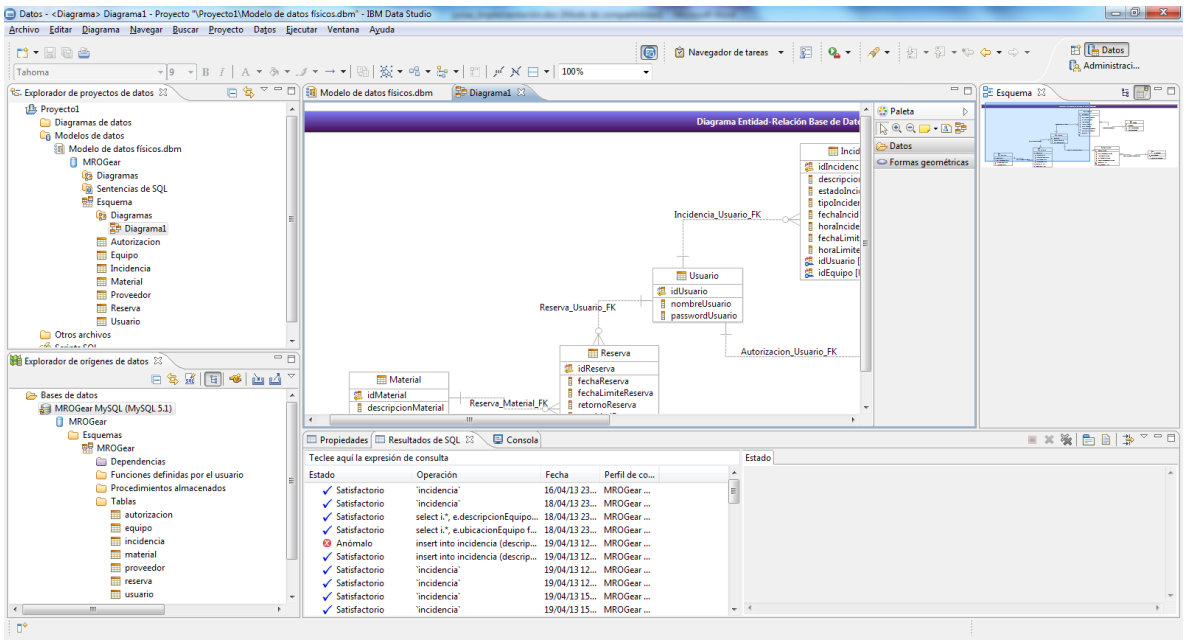


Ilustración 62 - Herramienta CASE ayuda al desarrollo y explotación de la base de datos

IBM Data Studio[31] permite al desarrollador, tanto la elaboración del modelo de datos físico de la base de datos, como el mantenimiento de los componentes (tablas, vistas, índices, etc…) y elaboración automática de scripts para su posterior reúso.

11.2.2 Estructura del proyecto

Hasta el momento hemos visto que herramientas se han utilizado durante el proceso de desarrollo e implementación de la aplicación. En esta sección, veremos cuál es la morfología típica de un proyecto **IBM Worklight**[2], sin entrar al detalle sobre cómo se implementan cada una de las funcionalidades descritas con anterioridad en este mismo documento, dado que dicha descripción se proveerá en el siguiente apartado.

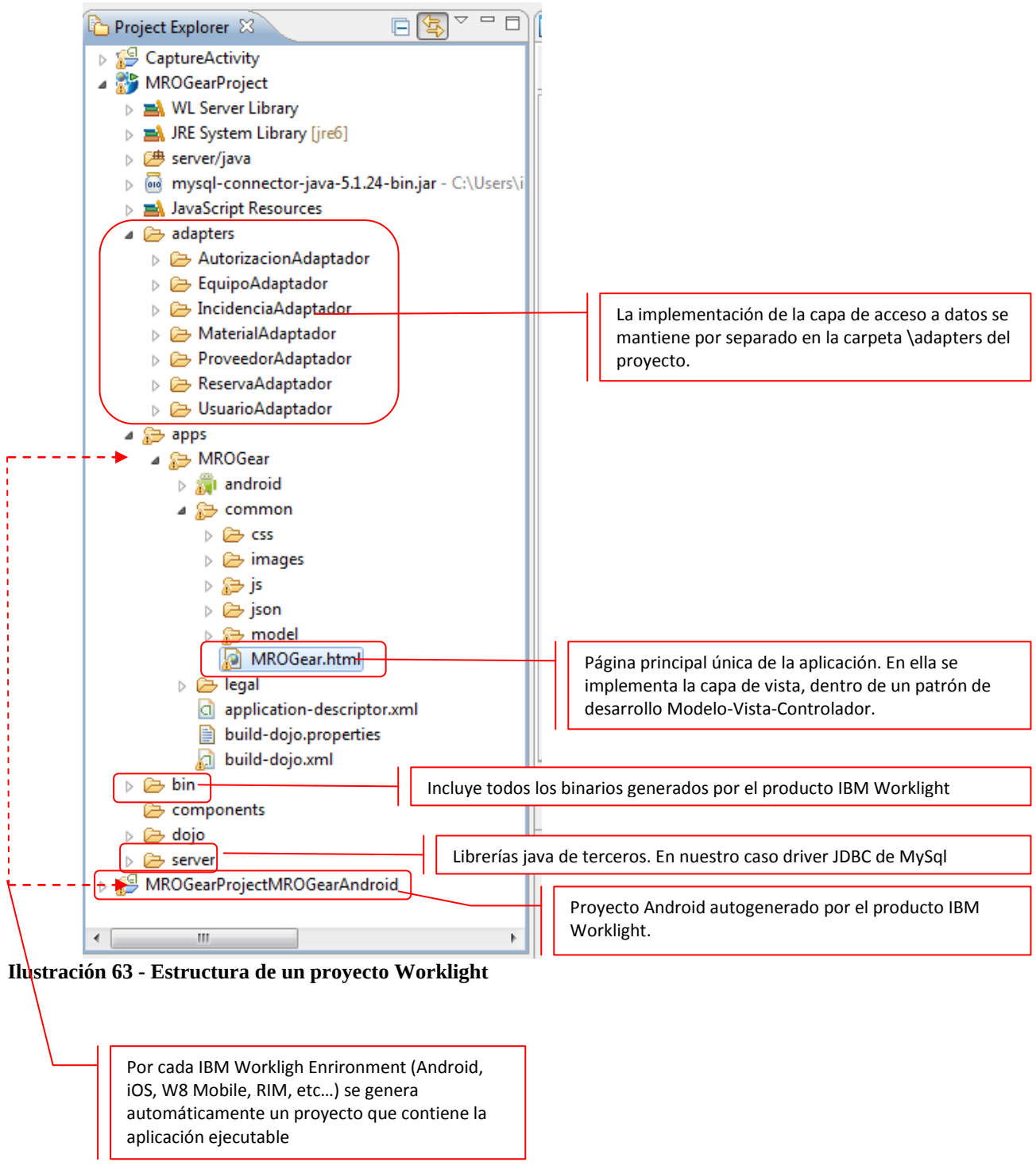


Ilustración 63 - Estructura de un proyecto Worklight

En primer lugar, cómo se puede observar en la ilustración, los adaptadores –recordemos que son los encargados del acceso a la base de datos y que se despliegan en la parte servidora- y el desarrollo de la aplicación principal, se mantienen por separado. Los primeros dentro de la carpeta '*adapters*' y los segundos dentro de la capeta '*apps/common*'. Tanto los binarios de los adaptadores como los binarios del aplicativo se autogeneran en la carpeta '*bin*' del proyecto, la siguiente ilustración lo muestra.

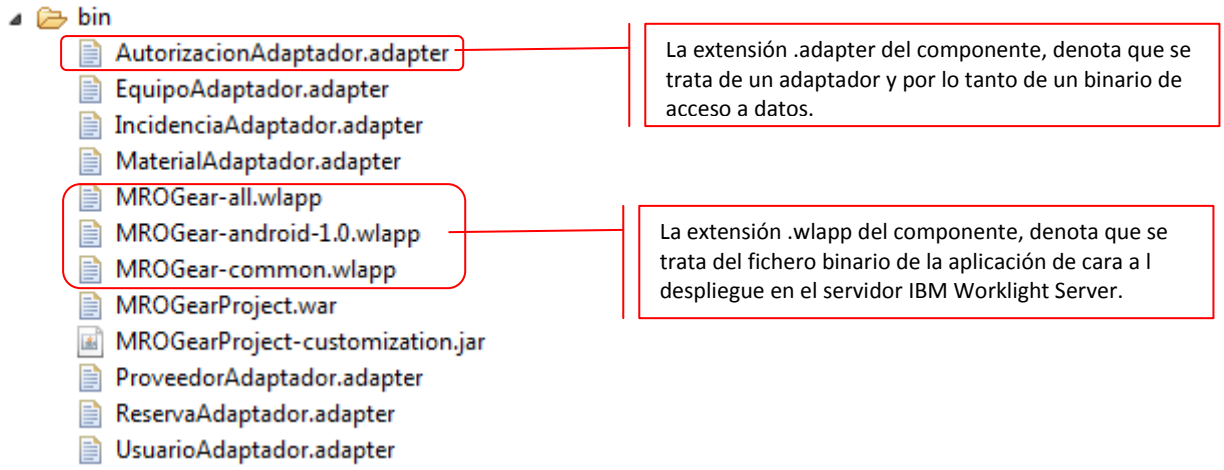


Ilustración 64 - Binarios autogenerados

En segundo lugar, como ya se ha repetido en diferentes ocasiones a lo largo de este documento, **IBM Worklight**[2] es un producto con capacidades multi-plataforma –codifica una vez, despliega donde quieras-, por lo tanto, para cada uno de los diferentes entornos (en nuestro caso solo **Android**[1]) se autogenera un proyecto con la fisionomía propia de cada una plataformas.

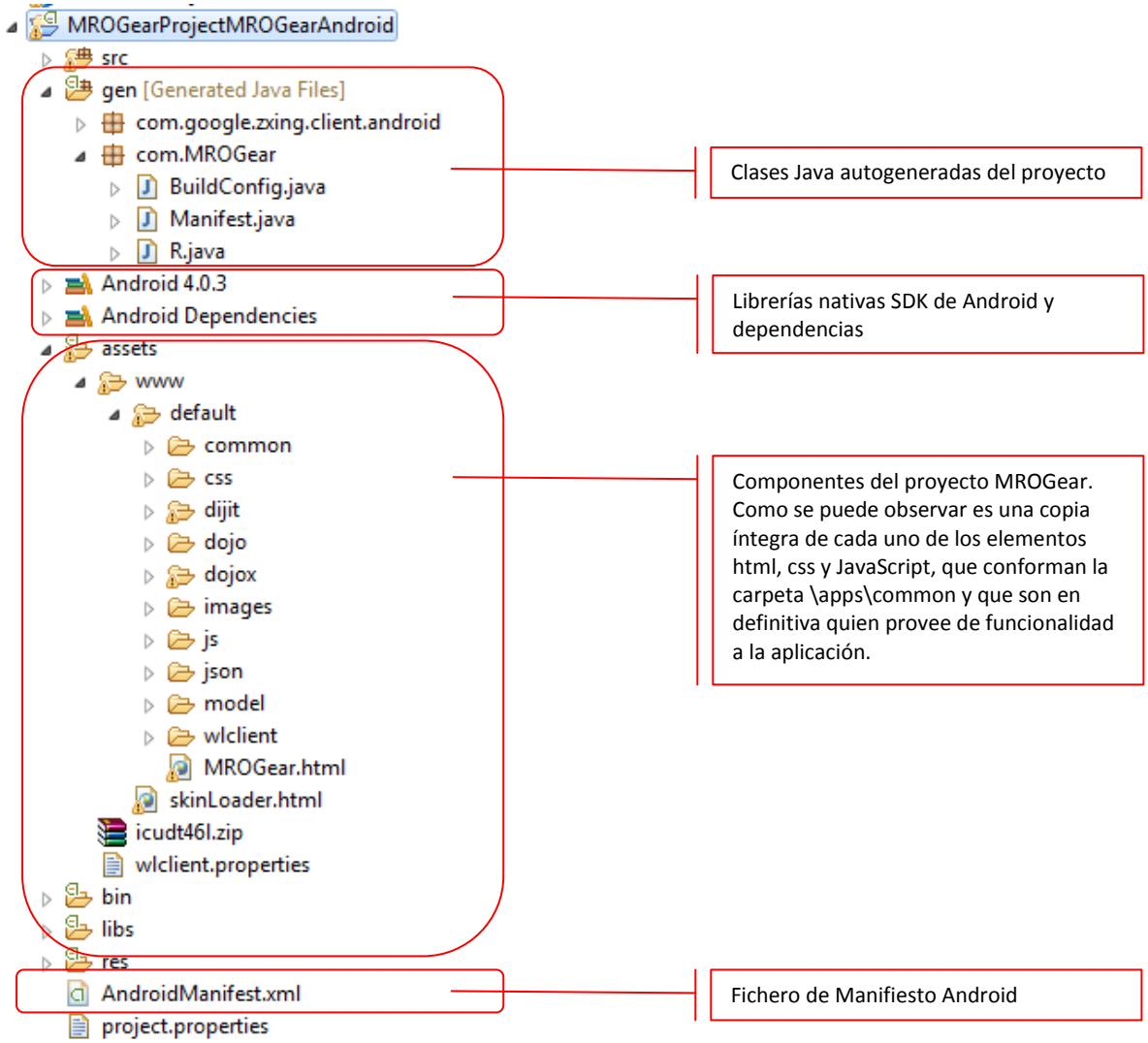


Ilustración 65 - Proyecto Android autogenerado

11.2.3 Aspectos relacionados con el desarrollo

Hasta el momento hemos descrito la fisionomía de un proyecto típico **IBM Worklight**[2]. Ahora nos centraremos en los aspectos del desarrollo más característicos, analizando un poco más a fondo el código fuente final del producto, desde un punto de vista de la arquitectura funcional.

Como ya citábamos en capítulos anteriores, la aplicación se ha concebido para la aplicación de un patrón de diseño llamado Modelo-Vista-Controlador. La intención de este apartado es, demostrar cómo se organiza el proyecto para dar soporte al citado patrón.

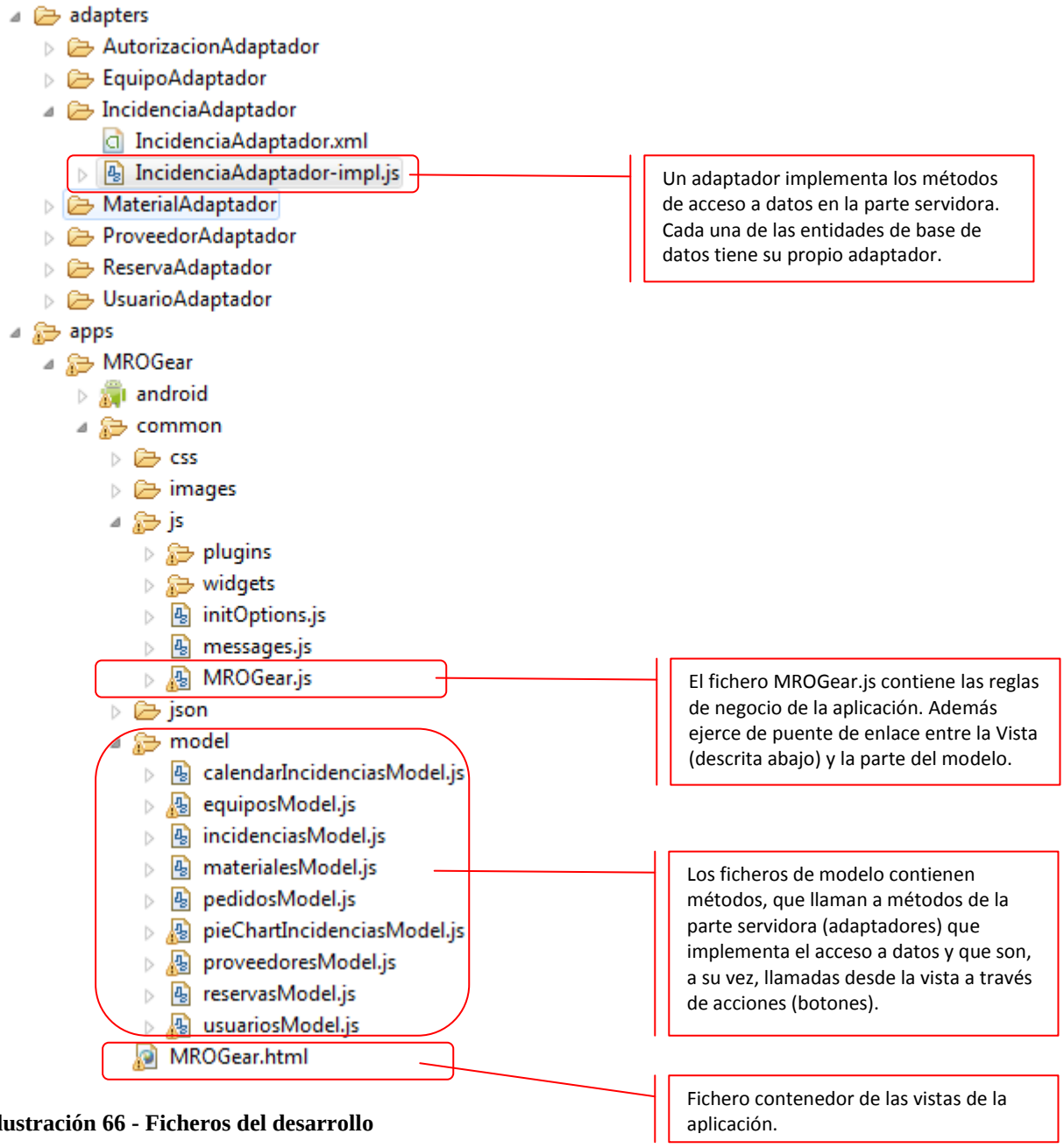


Ilustración 66 - Ficheros del desarrollo

Como se puede observar en la ilustración anterior, la capa de vista está compuesta por un único fichero *HTML*. La decisión de concentrar las diferentes vistas en un único fichero, viene dada por, la necesidad de minimizar al máximo las peticiones HTTP al servidor, ya que estas tienen un alto consumo de batería y por lo tanto de autonomía del dispositivo, siendo esta un valor crítico para cualquier usuario de la aplicación. A continuación ilustramos a modo de ejemplo, la vista del menú de incidencias de la solución **MROGear**:

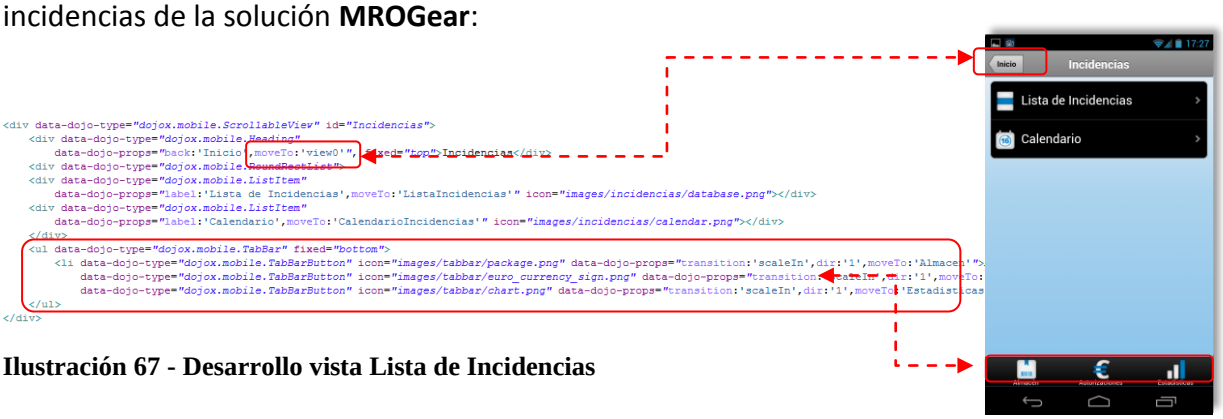


Ilustración 67 - Desarrollo vista Lista de Incidencias

Observbamos como utilizamos las etiquetas propietarias de la librería de *script* **Dojo Mobile**[12] para definir las reglas de navegación entre vistas. El propio *framework* es el encargado de hacer las transiciones entre pantallas. El caso remarcado en la ilustración anterior, muestra como a través de un botón se puede volver a la vista principal, identificado como 'view0'. De igual manera, y a modo de ejemplo, se muestra el trozo de código que muestra el menú a pie de vista.

Por otro lado, en lo referente a la capa de modelo observamos que tenemos dos tipos de componentes; los componentes que se despliegan en la parte cliente, contenidos dentro de la carpeta modelo; y los componentes que se despliegan en la parte servidora, también llamados adaptadores.

Ilustración 68 - Implementación de Adaptador para acceso a entidad Incidencia

En la ilustración anterior se muestra un fragmento de código correspondiente al adaptador que modela el acceso a la entidad 'Incidencias' de base de datos. Todos los métodos de acceso a dicha entidad, deberán ir implementados dentro del citado adaptador, guardando así la homogeneidad requerida a la hora de codificar.

Relacionado con el modelo también tenemos otro tipo de componente, que en este caso se despliega en la parte cliente, y que está contenido dentro de la carpeta modelo. Cada acción en la vista estará representada por una función dentro de estos ficheros. Como en el anterior caso, los ficheros de modelo son lo más cohesionados posible, es decir, cada entidad de base de datos tendrá su correspondiente fichero de modelo, que agrupará todas las posibles acciones que se pueden ejecutar desde la capa de vista. A continuación mostramos un ejemplo del modelo de la entidad 'Usuario'.

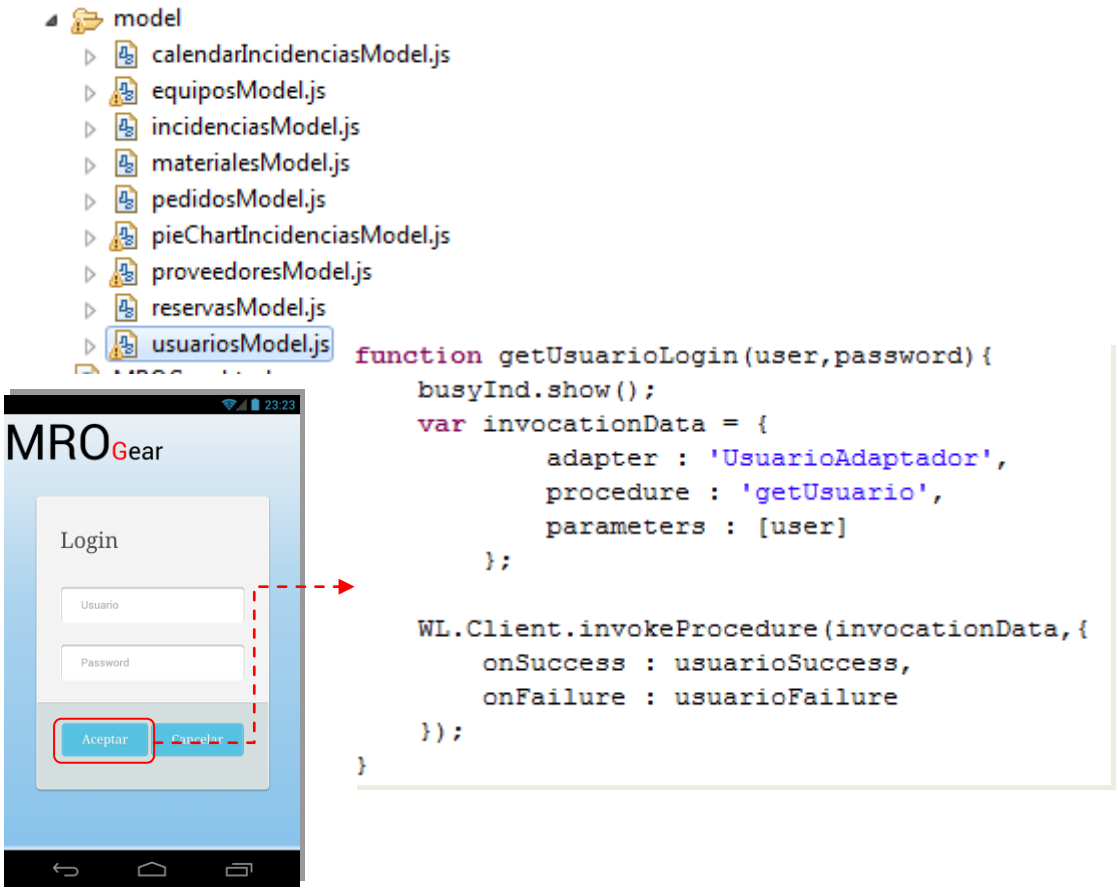


Ilustración 69 - Ejemplo invocación Adaptador de acceso a entidad Usuario desde la vista

El enlace entre la acción que el usuario ejecuta desde el dispositivo y el trozo de código que se ejecuta realmente, mostrado en la ilustración anterior, se produce en la capa controladora. En nuestro caso, el fichero que ejerce de puente de enlace entre la capa de vista y la capa de modelo, es el fichero 'MROGear.js', mostrado a continuación.

```
// LOGIN ***************************************************************************************************************
                var okLogin = registry.byId("okLogin");
                on(okLogin, "click",
                        function(){
                            WL.Device.getNetworkInfo(function (info) {
                                if (info.isNetworkConnected=='true'){
                                    // Validación de Usuario y Contraseña
                                    getUsuarioLogin($("#j_username").val(),$("#j_password").val());
                                } else {
                                    WL.SimpleDialog.show ("Error", "No hay conexion a internet",
                                            [{text: "Aceptar", handler: function(){}}]
                                        );
                                }
                            });
                    });
```

Ilustración 70 - Desarrollo que invoca al método de la capa de modelo

Como se puede observar, la función 'getUsuarioLogin' invocada desde la parte cliente, hace una llamada al método 'getUsuario' del adaptador 'AdaptadorUsuario', que se encuentra desplegado en la parte servidora del producto **IBM Worklight**[2].

Finalmente como comentario, decir que se utiliza el *framework* **jQuery**[32]**,** inicialmente integrado en el proyecto por parte del producto **IBM Worklight**[2], para acceder a determinadas características y objetos del **DOM**[33] de la aplicación.

11.2.4 Otros Aspectos relacionados con el desarrollo

En este apartado veremos otros aspectos relacionados con el desarrollo, aunque en esta ocasión no entraremos a fondo analizar cada uno de ellos, sino que simplemente los nombraremos para dejar constancia de los mismos en el proyecto.

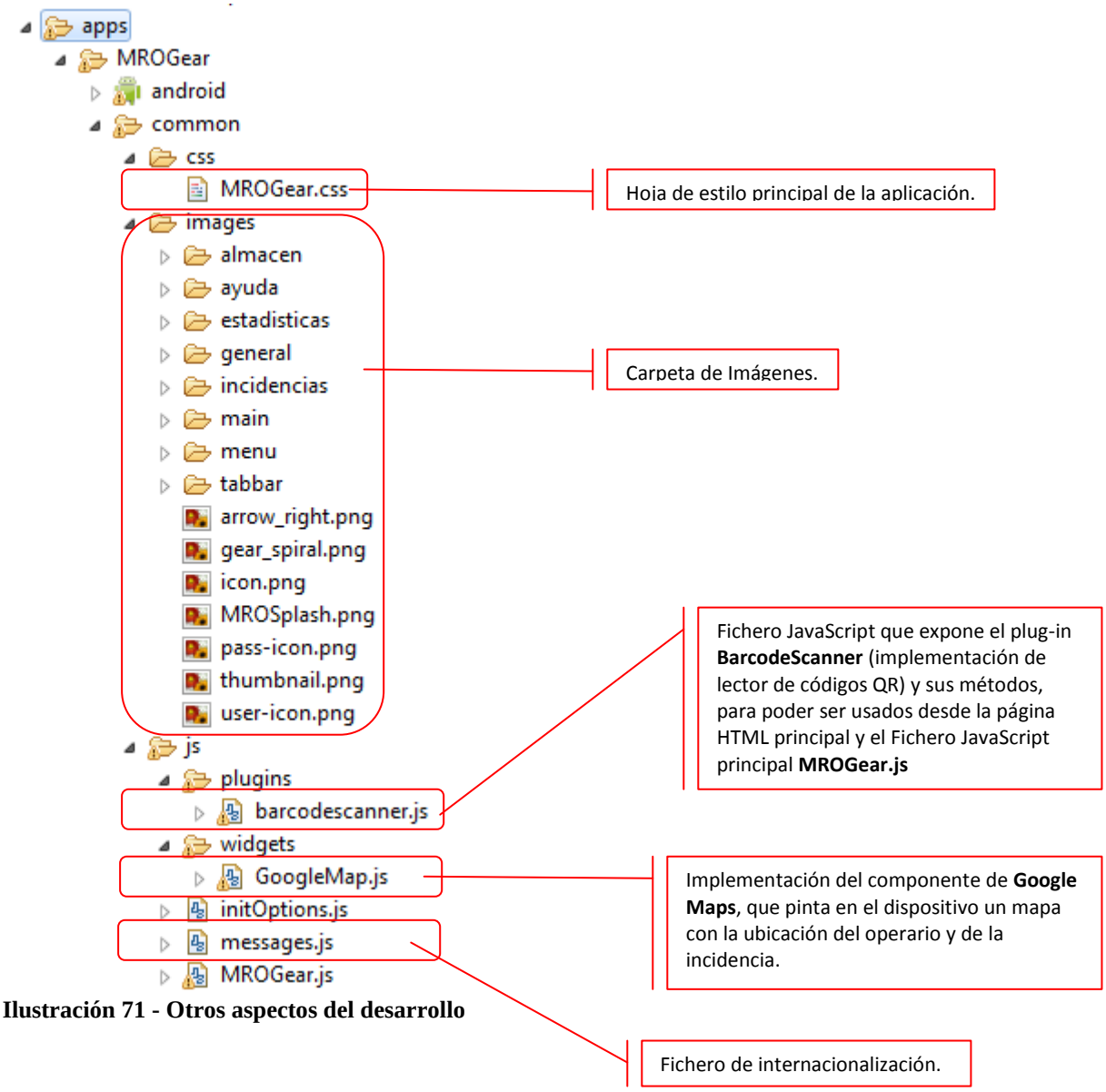


Ilustración 71 - Otros aspectos del desarrollo

Conclusiones 12

Para finalizar se expondrán las principales conclusiones en el ámbito del proyecto y en el ámbito personal a las cuales llegaron tras la finalización con éxito de la fase de implementación de la aplicación MROGear para dispositivos móviles Android. A pesar de que considero que los objetivos funcionales iniciales se han alcanzado en un 100%, a lo largo del proyecto han surgido mejoras en el sistema que no se han implementado por falta de tiempo y que quedarán reflejadas en este documento en el apartado de líneas futuras.

12.1 Objetivos

Por un lado, respecto a los objetivos funcionales marcados al inicio del proyecto en el mes de Febrero del presente año 2013, puedo afirmar que se han conseguido al 100%. Si bien es verdad, y derivado de la falta de conocimientos y de experiencia en programación con este tipo de dispositivo, algunos aspectos funcionales de la plataforma pueden ser mejorados en nuevas versiones del producto. Más adelante en este mismo documento se citarán dichos aspectos de mejora.

Por otro lado, la planificación inicial se ha ajustado bastante a la realidad en cuanto a fechas se refiere, dado que en ningún caso se han sufrido retrasos en las entregas, por lo tanto podemos concluir que el calendario del proyector expuesto en este mismo documento es bastante acertado. Ahora bien, comentar también que en cuanto a la estimación de horas dedicadas al proyecto, inicialmente desde un punto de vista muy optimista, el tiempo ha demostrado que la dedicación de tres horas diarias al proyecto en muchos de los casos ha sido irreal por motivos laborales y de exigencia del proyecto. Estas horas han tenido que ser recuperadas en fines de semana y festivos.

12.2 Valoraciones personales

A nivel personal, el proyecto me ha permitido mejorar mis conocimientos generales sobre el entorno de desarrollo para dispositivos móviles, hoy en día muy relacionado con mi rol profesional dentro de la compañía para la cual trabajo. Bien es verdad que, dicho rol está más enfocado a la consultoría y no tanto al desarrollo, de ahí que inicialmente quisiera apostar por una tecnología propietaria, que sin duda me ha permitido conocer más a fondo los entresijos de la misma a través de la realización de trabajo de campo sobre un proyecto 'real' que yo mismo he concebido. Esto ha sido de capital importancia a la hora de aumentar cualitativamente mi trabajo, tanto en la vida profesional como en la vida estudiantil.

Los principales escoyos con los que me he encontrado a lo largo del proyecto, han estado localizados principalmente en la fase de desarrollo e implementación de la solución. Dichos escoyos, han venido siempre relacionados por la falta de conocimiento en las múltiples tecnologías usadas en el proyecto, que quedan ampliamente descritas con anterioridad. Para mitigar los posibles efectos adversos que esta falta de conocimientos podría haber introducido en el proyecto, durante la realización del mismo y sobre todo en las fases iniciales donde la carga de trabajo es menor, se han realizado cursos de aprendizaje para algunas de las tecnologías relacionadas. Para aquellas para las cuales la inexistencia de recursos formativos ha sido palpable, se ha optado por la lectura de documentación proveniente de diferentes fuentes, en algunos casos información descargada directamente de Internet y en otros casos a través de la compra de libros especializados en la materia.

12.3 Líneas futuras

A lo largo del proyecto han ido surgiendo ideas de nuevas funcionalidades que puedes ser incluidas en las futuras versiones del aplicativo MROGear. Dichas mejoras no se han implementado por falta de tiempo. A continuación describimos los aspectos más relevantes de dichas mejoras:

1. Integración de servicios de transformación de voz a texto.
   Uno de los aspectos fundamentales que se ha buscado cubrir en el proyecto es el de la usabilidad de la plataforma. Hemos de recordar que en un entorno real, el principal beneficiario de la ampliación ha de ser el operario de campo, por lo tanto, facilitarle al máximo sus condiciones de trabajo ha sido una de las prioridades a la hora del diseño de la interfaz de usuario. Con esta nueva funcionalidad se busca, que las introducciones de datos a través del teclado del dispositivo móvil, de por sí muy costosa, sea de la menor cantidad posible. Así pues, el operario podrá dictar el texto a rellenar en los campos que así lo

requieran.

2. Notificaciones de tipo '*push*'.
   El enfoque actual de la aplicación no es muy proactivo por parte del sistema, ya que el operario ha de conectarse y sincronizar su dispositivo con el servidor central para poder ver sus asignaciones diarias. En este nuevo enfoque que se propone, es el propio sistema el que informa al operario de que tiene asignaciones nuevas a través de notificaciones de tipo '*push*'. De esta forma, se busca descargar al operario de la responsabilidad de chequear diariamente las incidencias y pasarle esa responsabilidad al sistema mismo.

3. Gestión inteligente de las incidencias.
   En la actualidad el aplicativo instalado en el dispositivo móvil carece de funcionalidades de análisis posicional de las incidencias. Uno de los objetivos básicos del proyecto era la mejora de la eficiencia, por lo tanto, dotar al sistema de capacidades de análisis de las incidencias y agrupación de las mismas en función de áreas geográficas y localización del usuario, puede ayudar a mejorar dicha eficiencia de los operarios y por lo tanto el consiguiente ahorro de costes por parte de la compañía.

4. Ampliación del módulo de estadísticas.
   Hasta el momento sólo se ha implementado una estadística que nos indica que máquinas se averían con mayor asiduidad si tenemos en cuenta los últimos tres meses. De cara a fomentar la competitividad sana entre los operarios, la inclusión de estadísticas comparativas entre los mismos en función del número de incidencias resueltas a tiempo y las que no, puede ayudar a dinamizar el trabajo diario de los mismos. Adicionalmente, nuevas estadísticas donde se agrupen en tramos horarios las incidencias, pueden ayudar tanto a la empresa como al operario de campo a gestionar más eficientemente el trabajo diario, dado que pueden hacer una mejor planificación.

Bibliografía consultada

Robert Virkus, Rollan Gülle, Thibaut Rouffineau, Chris Brady, Tim Messerschmidt, André Schmidt, Benno Bertels, Ovidiu Oliescu, Michel Shuqair, Peter Nowak, Julian Harty, Alex Jonsson, Richar Bloor, Jens Weller, Marco Tabor, Matos Kapetanakis, Michael Koch, Patrick Getzmann, Gary Johnson, Oliver Graf, Alexander Repty, Manuel Bieh, Andrej Balaz (2011*); Don't Panic Mobile Developer's guide to the galaxy.* Enough Software

Rohit Ghatol, Yogesh Patel (2012); Beginning PhoneGap: Mobile Web Framework for JavaScript and HTML5. Apress.

Frank W. Zammetti (2008); Practical Dojo Projects. Apress

[1] *Sitio oficial Android*. [en linea]. http://android.com/ [fecha consulta: 06/03/2013]

[2] *Sitio oficial IBM Worklight*. [en linea]. http://www-01.ibm.com/software/mobile-solutions/worklight/ [fecha consulta: 06/03/2013]

[3] Wikipedia [en linea]. http://es.wikipedia.org/wiki/Código_QR [fecha consulta: 07/03/2013]

[4] Sitio Oficial MySQL [en linea]. http://www.mysql.com/ [fecha consulta: 07/03/2013]

[5] Wikipedia [en linea]. http://es.wikipedia.org/wiki/Java_Database_Connectivity [fecha consulta: 07/03/2013]

[6] Wikipedia [en linea]. http://es.wikipedia.org/wiki/Red_privada_virtual [fecha consulta: 07/03/2013]

[7] Pruebas y Depuración de errores: las pruebas iniciales durante la fase de desarrollo se realizarán sobre el emulador ADT propio de la SDK de Android, por un tema de comodidad y agilidad. Sin embargo, para las pruebas finales funcionales y de rendimiento se realizarán sobre el dispositivo móvil directamente.

[8] Sitio Oficial Java [en linea]. http://www.java.com/es/ [fecha consulta: 07/03/2013]

[9] *Wikipedia [en linea]. http://es.wikipedia.org/wiki/BYOD. 07/03/2013]*

[10] *W3School. [en linea]. http://www.w3schools.com/json/default.asp. 07/03/2013]*

[11] *Sitio oficial Apache Cordova. [en linea]. http://cordova.apache.org/. 07/03/2013]*

[12] *Sitio oficial Dojo Mobile. [en linea]. http://dojotoolkit.org/features/mobile. 07/03/2013]*

[13] Sitio oficial Rhino. *[en linea]. https://developer.mozilla.org/en-US/docs/Rhino. 07/03/2013]*

[14] Wikipedia. *[en linea]. http://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol. [fecha consulta: 23/03/2013]*

[15] Wikipedia. *[en linea]. http://es.wikipedia.org/wiki/Modelo_Vista_Controlador. [fecha consulta: 24/03/2013]*

[16] Wikipedia. *[en linea]. http://es.wikipedia.org/wiki/C%C3%B3digo_abierto. [fecha consulta: 27/03/2013]*

[17] Sitio Oficial Android. *[en linea]. http://developer.android.com/reference/android/webkit/WebView.html. [fecha consulta: 27/03/2013]*

[18] Sitio Oficial Android. *[en linea]. http://developer.android.com/reference/android/webkit/JavascriptInterface.html. [fecha consulta: 27/03/2013]*

[19] Sitio Oficial Eclipse. *[en linea]. http://www.eclipse.org/. [fecha consulta: 31/03/2013]*

[20] Wikipedia. *[en linea]. http://es.wikipedia.org/wiki/Complemento_(inform%C3%A1tica). [fecha consulta: 31/03/2013]*

[21] Sitio Oficial Firefox. *[en linea]. http://www.mozilla.org/es-ES/firefox/new/. [fecha consulta: 31/03/2013]*

[22] Enlace Google Play. *[en linea]. https://play.google.com/store/apps/details?id=com.icecoldapps.screenshotultimatepro&hl=es. [fecha consulta: 31/03/2013]*

[23] ValuePickerTimePicker. *[en linea].* http://dojotoolkit.org/reference-guide/1.8/dojox/mobile/ValuePickerTimePicker.html *[fecha consulta: 04/05/2013]*

[24] ValuePickerTimePicker. *[en linea].* http://livedocs.dojotoolkit.org/dojox/mobile/ValuePickerDatePicker *[fecha consulta: 04/05/2013]*

[25] ValuePickerSlot. *[en linea].* http://dojotoolkit.org/reference-guide/1.8/dojox/mobile/ValuePickerSlot.html *[fecha consulta: 04/05/2013]*

[26] BarcodeScanner. *[en linea].* https://github.com/phonegap/phonegap-plugins/tree/master/Android/BarcodeScanner *[fecha consulta: 04/05/2013]*

[27] Sitio Oficial GitHub. *[en linea].* https://github.com/ *[fecha consulta: 04/05/2013]*

[28] IBM Worklight Infocenter. *[en linea].* http://pic.dhe.ibm.com/infocenter/wrklight/v5r0m6/index.jsp *[fecha consulta: 04/05/2013]*

[29] Google Maps API. *[en linea].* https://developers.google.com/maps/ *[fecha consulta: 04/05/2013]*

[30] Wikipedia. *[en linea].* https://es.wikipedia.org/wiki/Sistema_de_posicionamiento_global *[fecha consulta: 04/05/2013]*

[31] Sitio Oficial IBM Data Studio. *[en linea].* http://www.ibm.com/software/products/us/en/data-studio/ *[fecha consulta: 04/05/2013]*

[32] Sitio Oficial jQuery. *[en linea].* http://jquery.com/ *[fecha consulta: 06/05/2013]*

[33] Wikipedia. *[en linea].* http://es.wikipedia.org/wiki/Document_Object_Model *[fecha consulta: 06/05/2013]*