
DESARROLLO DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN DE INCIDENCIAS DE UNA EMPRESA DESARROLLADORA DE SOFTWARE

Víctor Salamanca Aguado

Máster Universitario en Ingeniería Informática

Desarrollo de aplicaciones sobre dispositivos móviles

Jordi Ceballos Villach

Robert Clarisó Viladrosa

11/01/2017



Esta obra está bajo licencia de [Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

B) GNU Free Documentation License (GNU FDL)

Copyright © Víctor Salamanca Aguado.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© (Víctor Salamanca Aguado)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilm, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Desarrollo de una aplicación móvil para la gestión de incidencias de una empresa desarrolladora de Software</i>
Nombre del autor:	<i>Víctor Salamanca Aguado</i>
Nombre del consultor/a:	<i>Jordi Ceballos Villach</i>
Nombre del PRA:	<i>Robert Clarisó Viladrosa</i>
Fecha de entrega:	<i>01/2017</i>
Titulación o programa:	<i>Máster Universitario en Ingeniería Informática</i>
Área del Trabajo Final:	<i>Desarrollo de aplicaciones sobre dispositivos móviles</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Gestor de incidencias Android Redmine.</i>
Resumen del Trabajo:	
<p>Con la realización del presente Trabajo Fin de Master (TFM), se pretende plasmar, el análisis, diseño e implementación de una aplicación móvil orientada a la gestión de incidencias de una empresa desarrolladora de Software propietario, que facilite la colaboración y comunicación entre los diferentes usuarios implicados (clientes y empleados de la empresa), mediante solicitudes realizadas a través de la aplicación. Además de permitir que los usuarios, puedan acceder y consultar los datos de sus solicitudes y realizar su trabajo desde cualquier parte, con el único requisito de disponer de un dispositivo móvil o <i>Tablet</i> con conexión a Internet.</p> <p>Para su implementación, se ha decidido usar el sistema operativo para móviles Android, que posibilita una arquitectura fiable y robusta, así como compatibilidad con un gran número de dispositivos disponibles en el mercado actualmente.</p> <p>El proyecto resultante, podría ser usado como base para la realización de otros proyectos de empresas, sea cual fuere su finalidad de negocio, ayudado por la modularidad de Redmine y su fácil integración con otros sistemas.</p> <p>Para poder ofrecer una aplicación a la altura de las expectativas comerciales actuales, se aplicará una metodología de Diseño centrado en el usuario ágil, que favorecen el desarrollo de proyectos con continuos cambios, mejorando la gestión de riesgos y la calidad.</p> <p>Para finalizar, se incluirá un capítulo de anexos, que contendrá un manual detallado de instalación de la aplicación y diferente información, resultado de la realización del proyecto.</p>	

Abstract:

With the achievement of this end-of-Master's project, it is intended to reflect the analysis, design and implementation of a mobile application oriented to the incident management of a proprietary software developer company, which facilitates collaboration and communication between the different users involved (customers and employees of the company), through requests made with the application. In addition to allowing users to access and consult the data of their applications and perform their work from anywhere, with the only requirement to have a mobile device or Tablet with Internet connection.

For its implementation, it has been decided to use the Android mobile operating system, which facilitates a reliable and robust architecture, as well as compatibility with a large number of devices available on the market today.

The resulting project could be used as a basis for the realization of other projects of companies, whatever its business purpose, helped by the modularity of Redmine, and its easy integration with other Systems.

In order to offer an application that meets current commercial expectations, an agile user-centered design methodology will be applied, favoring the development of projects with continuous changes, improving risk management and quality.

Finally, a chapter of annexes will be included, which will contain a detailed implementation manual and different information resulting from the completion of the project.

INDICE GENERAL

Resumen del Trabajo	3
Sección I - Introducción	9
1- Estructura de la memoria	9
2- Estado del arte	10
2.1 Sistemas operativos móviles	10
2.1.1 Comparativa de los principales Sistemas operativos móviles	12
2.2 El sistema operativo Android	13
2.2.1 Arquitectura de Android	14
2.2.1 Versiones de Android	16
Sección II - Plan de trabajo	17
3- Contexto y justificación del trabajo	17
4- Objetivos del trabajo	17
4.1 Requisitos de alto nivel	17
4.2 Elección Tecnológica	18
4.3 Dispositivo objetivo	18
4.4 Público objetivo	18
5- Enfoque y método seguido	19
6- Planificación del trabajo	20
6.1 Análisis de costes	21
Sección III – Caso de estudio	22
7- Análisis	22
7.1 Método de investigación	22
7.1.1 Planteamiento	22
7.1.2 Desarrollo	23
7.1.2 Resultados y Conclusiones	28
7.2 Usuarios y contexto de uso	29
7.2.1 Perfiles de usuario	29
7.2.2 Contextos de uso	30
7.2.3 Análisis de tareas	31
8- Diseño conceptual	32
8.1 Escenarios de uso	32
8.2 Flujos de interacción	34
9- Prototipado	36
9.1 Sketches	36
9.2 Prototipos	37
9.2.1 Pantalla de inicio	37
9.2.2 Pantalla de registro de usuario	38
9.2.3 Menú principal	39

9.2.4 Pantalla de registro de incidencia	40
9.2.5 Pantalla de solicitud de desarrollo	41
9.2.6 Pantalla de solicitudes realizadas	42
9.2.7 Pantalla de detalle de solicitud.....	43
9.2.8 Pantalla de envío de sugerencias	44
9.2.9 Pantalla de preguntas frecuentes	45
10- Evaluación.....	45
10.1 Preguntas para el test de usuarios.....	46
10.2 Tareas a realizar por parte de los usuarios.....	46
10.3 Preguntas referentes a las tareas.....	46
11- Definición de los casos de uso	47
11.1 Detalle de los casos de uso	47
11.2 Diagrama de transición de estados	51
11.2.1 Incidencias	51
11.2.2 Solicitud de desarrollo.....	52
12- Diseño de la arquitectura	52
12.1 Diagrama de Bases de datos	52
12.2. Diagrama de clases	53
12.3 Arquitectura general del sistema	55
Sección IV – Implementación	56
13- Desarrollo	56
13.1 Entorno tecnológico.....	56
13.1.1 Aplicación móvil	56
13.1.2 Gestión de incidencias con Redmine.....	57
13.1.3 Configuración del gestor de incidencias Redmine	57
13.2 Estructura de la aplicación	59
13.2.1 Carpeta de código fuente “src”.....	59
13.2.2 Carpeta de recursos “res”	61
13.4 Buenas prácticas de codificación	62
14- Pruebas.....	62
14.1 Entorno de pruebas	62
14.2 Tipos de pruebas realizadas.....	63
14.2.1 Pruebas unitarias	63
14.2.2 Pruebas funcionales	64
Sección V – Conclusiones	67
15- Conclusiones y futuros trabajos.....	67
15.1 Resultado final	67
15.2 Logros alcanzados y conocimientos adquiridos	68
15.3 Posibles trabajos futuros.....	69
Sección VI – Referencias	70

16- REFERENCIAS	70
Sección VII – Anexos	72
A- ANEXOS	72
A.1 Instrucciones de instalación y compilación	72
A.1.1 Instalación del entorno de programación	72
A.1.2 Instalación de Redmine	74
A.2 Manual de usuario	77
1. Entrada a la aplicación	77
1.1 Registro de usuario	77
1.2 Menú principal	78
1.2.1 Pantalla de solicitudes realizadas	79
1.2.1.1 Pantalla de detalle de solicitud	80
1.2.2 Pantalla de registro de incidencia	83
1.2.3 Solicitud de nuevo desarrollo	84
1.2.4 Pantalla de envío de sugerencias	86
1.2.5 Pantalla de preguntas frecuentes	87

INDICE DE ILUSTRACIONES

(Tabla 2.1) Comparativa Sistemas Operativos Móviles	12
(Figura 2.1) Sistemas operativos más utilizados en el mundo. [5]	13
(Figura 2.2) Sistemas operativos más utilizados en Europa. [5]	13
(Diagrama 2.1) Arquitectura de Android	14
(Tabla 2.2) Histórico de versiones de Android	16
(Tabla 6.1) Planificación general del proyecto	20
(Diagrama 6.1) Diagrama Gantt de planificación general del proyecto. [10]	20
(Tabla 6.2) Dedicación en horas al proyecto dividido por fases	21
(Tabla 6.3) Presupuesto del Proyecto	21
(Tabla 7.1) Características de los perfiles de usuario	29
(Tabla 7.2) Contextos de uso de los perfiles de usuario	31
(Tabla 7.3) Roles de los perfiles de usuario	31
(Tabla 8.1) Escenarios de uso	34
(Diagrama 8.1) Flujos de interacción	35
(Diagrama 9.1) Sketches	36
(Figura 9.1) Prototipo pantalla de inicio	37
(Figura 9.2) Prototipo pantalla de registro de usuario	38
(Figura 9.3) Prototipo Menú principal	39
(Figura 9.4) Prototipo pantalla de registro de incidencia	40
(Figura 9.5) Prototipo pantalla de escáner	40

(Figura 9.6) Prototipo pantalla de solicitud de desarrollo	41
(Figura 9.7) Prototipo pantalla de solicitudes realizadas.....	42
(Figura 9.8) Prototipo pantalla de detalle de solicitud (Desarrollo pendiente de aceptación).....	43
(Figura 9.9) Prototipo pantalla de detalle de solicitud (Incidencia).....	43
(Figura 9.10) Prototipo pantalla de envío de sugerencias	44
(Figura 9.11) Prototipo pantalla de preguntas frecuentes.....	45
(Diagrama 11.1) Diagrama de casos de uso.....	47
(Diagrama 11.2) Diagrama de transición de estados (incidencias).....	51
(Diagrama 11.3) Diagrama de transición de estados (solicitudes de desarrollo).....	52
(Diagrama 12.1) Modelo de datos de Redmine.....	53
(Diagrama 12.2) Diagrama de clases (paquetes). [13].....	53
(Diagrama 12.3) Diagrama de clases (Constantes). [13].....	53
(Diagrama 12.4) Diagrama de clases. [13].....	54
(Diagrama 12.5) Arquitectura general del sistema. [14].....	55
(Figura 13.1) Ejemplo de JSON de entrada para campos personalizados	59
(Figura 13.2) Estructura del proyecto (carpeta SRC).....	59
(Figura 13.3) Estructura del proyecto (carpeta RES)	61
(Figura 14.1) Resultados de las pruebas unitarias	63
(Tabla 14.1) Definición de casos de prueba.....	66
(Tabla 14.2) Resultado de la ejecución de los casos de prueba	66
(Figura A.1) Pantalla de inicio	77
(Figura A.2) Pantallas de acceso al registro de usuario.....	78
(Figura A.3) Pantalla de Menú principal y menú desplegable.....	79
(Figura A.4) Pantalla de Mis Solicitudes	80
(Figura A.6) Pantallas de detalle de Incidencia de Sistemas e Incidencia de Software	82
(Figura A.7) Pantallas de Registro de Incidencia.....	83
(Figura A.8) Pantallas de Registro de Incidencia de sistemas y de escáner de Código BIDI	84
(Figura A.9) Pantalla de Solicitud de nuevo desarrollo	85
(Figura A.10) Pantalla de Peticiones de Redmine.....	85
(Figura A.11) Pantalla de modificación de petición de Redmine.....	86
(Figura A.12) Flujo de aceptación/rechazo de solicitud de desarrollo	86
(Figura A.13) Pantalla de Envío de sugerencias	87
(Figura A.14) Pantalla de listado de preguntas frecuentes.....	88
(Figura A.15) Pantalla de detalle de pregunta frecuente seleccionada	89

SECCIÓN I - INTRODUCCIÓN

1- Estructura de la memoria

El documento ha sido dividido en siete secciones y un resumen inicial. A continuación, se detalla el contenido de los apartados:

- **Resumen:** Resumen del contenido del trabajo fin de Master.
- **I- Introducción:** Esta sección la componen dos capítulos:
 - **Estructura de la memoria:** Este punto pretende facilitar la comprensión del presente documento, detallando sus diferentes apartados.
 - **Estado del arte:** En este punto, se realiza un análisis comparativo de los sistemas operativos móviles del mercado, haciendo especial hincapié en Android.
- **II- Plan de trabajo:** se divide en los siguientes cuatro capítulos.
 - **Contexto y justificación del trabajo:** Este apartado, sirve de punto de partida del proyecto, definiendo la necesidad a cubrir y la justificación de las decisiones tomadas en su ejecución.
 - **Objetivos del proyecto:** Listado de los objetivos del trabajo, indicando los requisitos funcionales y no funcionales a alto nivel, así como las plataformas y / o dispositivos a las que se dirige.
 - **Enfoque y método seguido:** Se indica la estrategia elegida para llevar a cabo el trabajo, valorando la idoneidad de la misma para alcanzar los objetivos.
 - **Planificación del trabajo:** Descripción y planificación de las tareas del proyecto.
- **III- Caso de estudio:** Aplicación práctica de la metodología de Diseño centrado en el usuario, en la solución del problema propuesto. Se divide en los siguientes capítulos:
 - **Análisis:** Su objetivo es conocer las características de los usuarios, sus necesidades y objetivos, así como los diferentes contextos de uso.
 - **Diseño conceptual:** A partir de la fase de análisis, se elaborarán los escenarios de uso, es decir, la descripción desde el punto de vista del usuario de la utilización del producto en un contexto concreto. Además, se realizarán los flujos de interacción.
 - **Prototipado:** A partir de los flujos de interacción definidos anteriormente se representarán prototipo de alta fidelidad.
 - **Evaluación:** Nos centraremos principalmente en la definición de la evaluación del prototipo, mediante la técnica del test con usuarios, quedando su ejecución fuera del ámbito de este proyecto.
 - **Definición de los casos de uso:** Una vez finalizada la aplicación del Diseño centrado en el usuario, se definirán formalmente los casos de uso, con objeto de establecer las funcionalidades de la aplicación.
 - **Diseño de la arquitectura:** En este apartado se definirá la arquitectura del sistema, identificando las entidades que lo componen. Base de datos, Servidor, API, etc.

- **IV- Implementación:** En esta sección, nos centraremos en el producto a desarrollar, documentando las decisiones tomadas durante el desarrollo y las pruebas.
 - **Desarrollo:** En esta etapa se construirá la aplicación móvil que satisfaga los objetivos planteados en el proyecto. Se añadirá una breve descripción del entorno tecnológico y de la estructura de la aplicación
 - **Pruebas:** Se definirá el entorno de pruebas y se detallarán los resultados de las diferentes pruebas ejecutadas, tanto unitarias como funcionales.
- **V- Conclusiones:** Esta sección incluye las conclusiones obtenidas con la realización del Trabajo fin de Master, dividiéndose en los siguientes tres capítulos:
 - **Resultado final:** Contendrá las explicaciones de las impresiones personales sobre los resultados tras la consecución del proyecto.
 - **Logros alcanzados y conocimientos adquiridos:** Desde un punto de vista más personal, se detallarán las aportaciones y los conocimientos adquiridos con el proyecto.
 - **Posibles trabajos futuros:** Mejoras y posibles ampliaciones al proyecto.
- **VI- Referencias:** Bibliografía y material usado en la realización del proyecto.
- **VII- Anexos:** Esta sección contiene los manuales de usuario y las instrucciones de instalación.

2- Estado del arte

2.1 Sistemas operativos móviles

A continuación, se describen brevemente los principales Sistemas Operativos móviles de la actualidad:

Android [\[1\]](#)

Sistema operativo creado por la Open Handset Alliance (liderada por Google), con licencia Apache de software libre y código fuente abierto. Se caracteriza por estar desarrollado sobre una versión modificada del *Kernel* de Linux.

Al ser una plataforma abierta, permite a cualquier fabricante desarrollar sus productos sobre ella. Actualmente es uno de los sistemas operativos en alza del mundo y se distribuye en numerosos dispositivos de diferentes marcas.

PROS:

- Al ser desarrollado de forma **abierto**, se trata de una ventaja tanto para los que desarrollan sus aplicaciones como para sus usuarios.
- Puedes **personalizar** tu teléfono al máximo y modificar funciones del teléfono simplemente instalando una aplicación.
- Gran cantidad de **aplicaciones** tanto gratuitas como de pago

CONTRAS:

- Posee una gran **fragmentación**, al existir una gran cantidad de terminales con diferentes versiones y características.
- Al ser abierto provoca que sea más susceptible al ataque de **vulnerabilidades** y ataques.
- **Incompatibilidades** entre versiones.

iOs [2]

Sistema operativo creado por Apple, con licencia propietaria, basado en una variante del *kernel* de MAC OS X.

Su aspecto más destacable y que ha ayudado a marcar tendencia en el mundo de la tecnología móvil, es su interfaz de usuario y su gran aprovechamiento de los recursos hardware del dispositivo sobre el que se ejecuta.

PROS:

- Buen **diseño**, funcionalidad y facilidad de uso.
- Gran variedad de **aplicaciones** y juegos.
- Al ser un sistema cerrado y no tener acceso al código, es una ventaja respecto a la **seguridad**.

CONTRAS:

- Sistema operativo **cerrado** desde el punto de vista del desarrollo.
- No es posible **personalizar** o modificar el sistema.
- Al existir un único fabricante, los **precios** son **altos** por la falta de competencia.

Blackberry OS [3]

Es un sistema operativo móvil de código cerrado desarrollado por BlackBerry, antigua Re-search In Motion (RIM), con licencia propietaria. Se caracteriza por estar desarrollado en Java, para su uso en computadores de bolsillo, orientado principalmente a un uso profesional. A comienzos de 2013 fue lanzado su predecesor llamado **BlackBerry 10**.

PROS:

- Gran implantación en el mundo **empresarial**.
- **Facilidad** de uso profesional (teclado QWERTY que en prácticamente todos los teléfonos del fabricante está presente).
- Buena **integración** con servidores de empresas como Microsoft Exchange o Lotus Notes.

CONTRAS:

- Escasa variedad de **aplicaciones** comerciales.
- Muy acotado al **entorno** empresarial con escaso potencial **multimedia**.
- La propia marca Blackberry, lanzo a finales del año 2015 su primer móvil con Sistema operativo **Android**, lo que deja claro que centrarán sus entornos en otros ámbitos en lugar de potenciar el Sistema Blackberry OS.

Windows Phone [4]

Sistema operativo móvil desarrollado por Microsoft, como sucesor de Windows Mobile. A diferencia de su predecesor está enfocado en el mercado de consumo en lugar del mercado empresarial. Un aspecto destacable es su integración con servicios de Microsoft como OneDrive, Skype o Xbox Live.

Debido a la gran fragmentación de sus sistemas operativos, Microsoft anunció en enero de 2015 que dará de baja a Windows Phone, para enfocarse en un único sistema más versátil denominado Windows 10 Mobile, disponible para todo tipo de plataformas (teléfonos inteligentes, tabletas y computadoras).

PROS:

- Sistema gráfico **sencillo** y con agradable experiencia de usuario.
- Diseño **moderno**, práctico, e innovador.
- Gran **inversión** por parte de Microsoft en continuas mejoras.
- Crecimiento y **apertura** al desarrollo.

CONTRAS:

- Escasa variedad de aplicaciones.
- Gran fragmentación.

2.1.1 Comparativa de los principales Sistemas operativos móviles

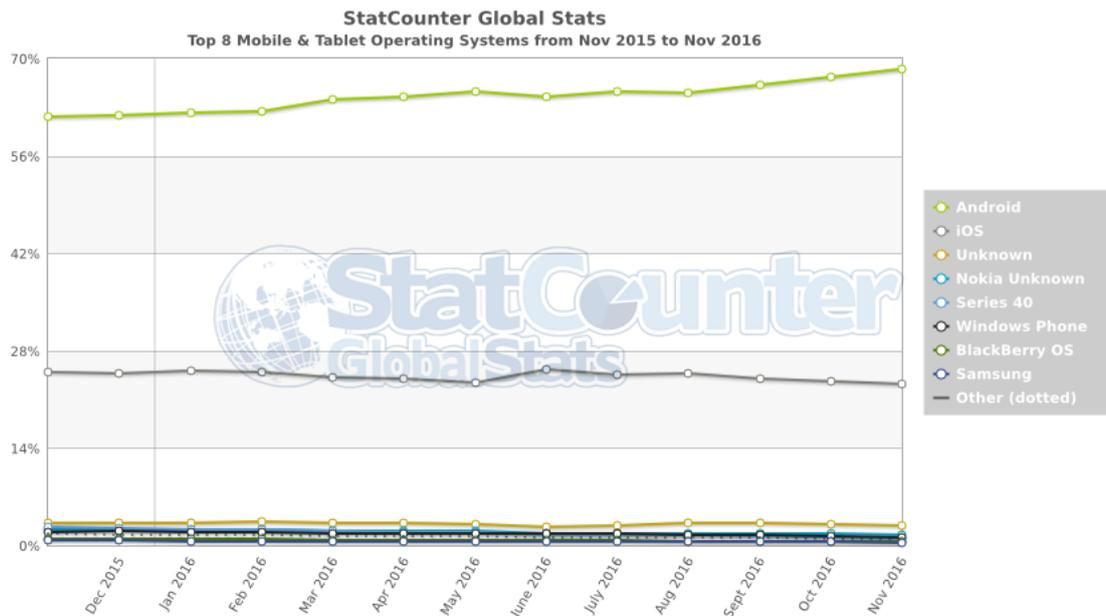
A continuación, se muestra una tabla comparativa con los principales Sistemas Operativos móviles:

Nombre	Empresa	Licencia	Kernel	Procesador	Fragmentación	Última Versión estable	Fecha última versión
Android	Google	Open Source	Linux con máquina virtual Dalvik	Soporte para ARM, MIPS y x86	Muy fragmentada. Existen 4 versiones de uso mayoritario: Jelly Bean, Kit Kat, Lollipop y Marshmallow	Nougat 7.1	20 de octubre de 2016
iOS	Apple	APSL, Apple Public Source License	OS X	ARM	Poco fragmentada. Existen 2 versiones de uso mayoritario: iOS8 e iOS9	iOS 10.2	24 de octubre de 2016
Windows Phone	Microsoft	Propietaria	Windows CE	ARM, x86, x64	Bastante fragmentada. Existen 4 versiones de uso mayoritario: WP 7, WP 8, WP 8.1 y Windows 10 Mobile	Windows 10 Mobile 10.0.14361 67	08 de junio de 2016
Blackberry OS	Blackberry	Propietaria	Propietario	ARM (Qualcomm Snapdragon u OMAP)	Bastante fragmentada. Incluso con modelos con el mismo sistema operativo existen problemas de compatibilidad de aplicaciones	7.1.0.2994 (BlackBerry 9720)	25 de abril de 2014

(Tabla 2.1) Comparativa Sistemas Operativos Móviles

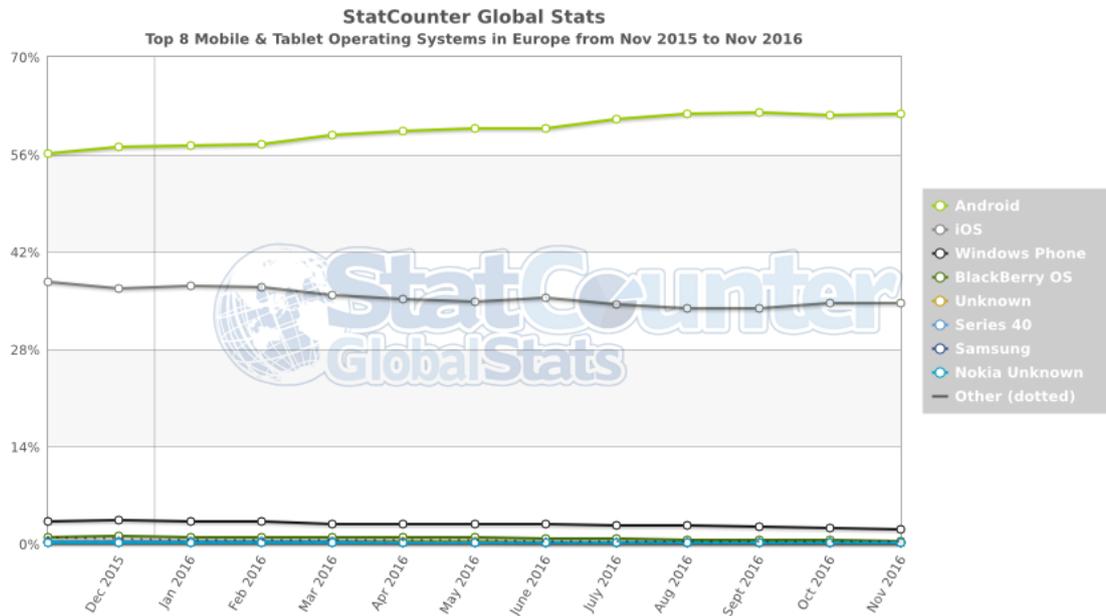
Actualmente, más de la mitad de los dispositivos móviles del mercado poseen sistema operativo Android y el número de aplicaciones descargadas para este sistema aumenta considerablemente cada año, ayudando a este crecimiento la gratuidad de la mayoría de ellas.

Como se puede observar en la siguiente gráfica, que refleja el uso de Sistemas operativos móviles en teléfonos móviles y *Tablets*, el sistema más utilizado en todo el mundo es Android, seguido de lejos por iOS.



(Figura 2.1) Sistemas operativos más utilizados en el mundo. [5]

Hay que tener en cuenta que este gráfico se refiere a nivel mundial. Pero dependiendo del lugar de observación, entran en juego otros factores como el poder adquisitivo y otros elementos sociales, por lo que estos datos pueden variar sustancialmente. Por ejemplo, como se puede observar en el siguiente gráfico, que representa el mismo periodo de tiempo, acotado al continente europeo, las diferencias entre iOS y Android son menores.



(Figura 2.2) Sistemas operativos más utilizados en Europa. [5]

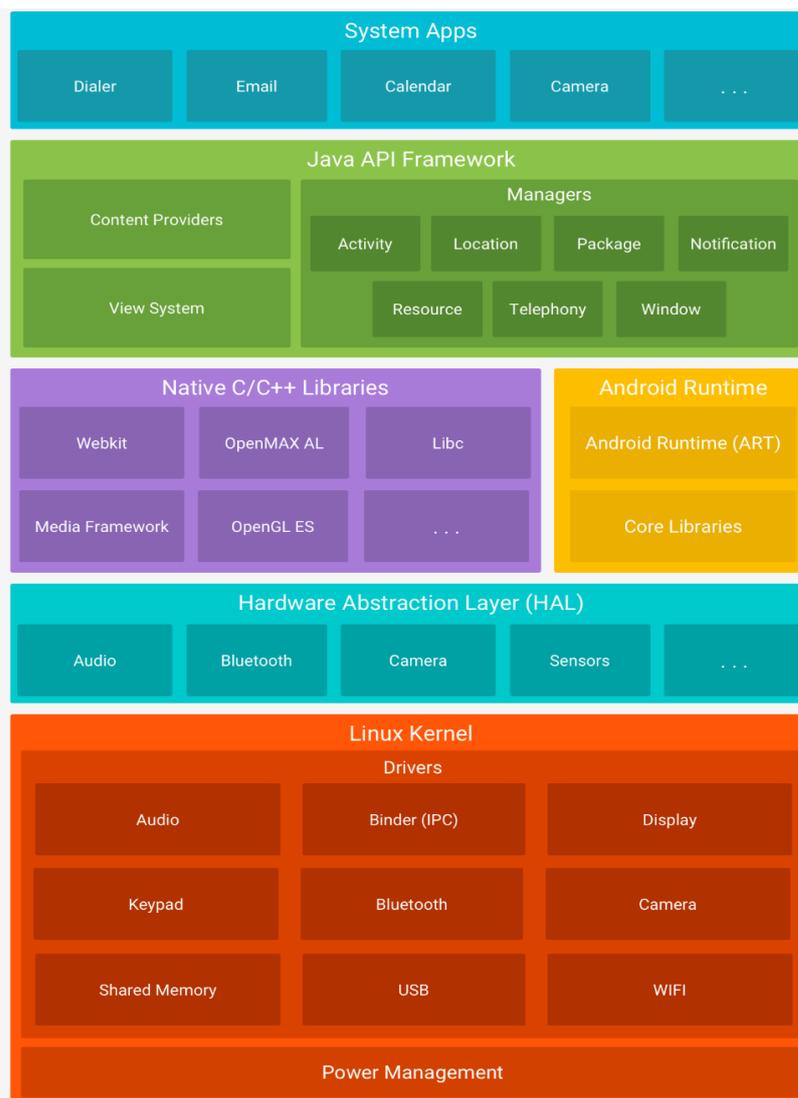
2.2 El sistema operativo Android

Android es un sistema operativo, basado en el *kernel* de Linux y diseñado principalmente para dispositivos móviles con pantalla táctil, tales como *smartphones* o *tablets*. Inicialmente desarrollado por Android Inc., y respaldado económicamente por Google, que más tarde, en el año 2005 adquiriría la

empresa. Uno de los aspectos fundamentales del sistema operativo de Android fue su orientación a la multiplataforma, algo realmente novedoso, debido a que hace unos años, un sistema operativo se asociaba a un único dispositivo. Esta y otras características innovadoras, consiguieron que Android se convirtiera en el sistema operativo más utilizado.

2.2.1 Arquitectura de Android [6]

La arquitectura de Android se encuentra estructurada en diferentes capas, donde cada una de ellas utiliza elementos de la capa inferior para realizar sus funciones, es por ello que a este tipo de arquitectura se le conoce como pila de Software. En el siguiente diagrama (**Diagrama 2.1**), se muestran en detalle los componentes de la arquitectura de Android.



(Diagrama 2.1) Arquitectura de Android

Kernel de Linux

La base de la plataforma Android es el *kernel* de Linux. Por ejemplo, el “runtime” de Android (ART) se basa en el *kernel* de Linux para funcionalidades subyacentes, como la generación de subprocesos y la administración de memoria de bajo nivel.

El uso del *kernel* de Linux permite que Android aproveche funciones de seguridad claves y, al mismo tiempo, permite a los fabricantes de dispositivos desarrollar controladores de hardware para un *kernel* conocido.

Capa de abstracción de hardware (HAL)

La capa de abstracción de hardware (HAL) brinda interfaces estándares que exponen las capacidades de hardware del dispositivo al *framework* de la Java *API* de nivel más alto. La HAL consiste en varios módulos de biblioteca y cada uno de estos implementa una interfaz para un tipo específico de componente de hardware, como el módulo de la cámara o de *bluetooth*. Cuando el *framework* de una *API* realiza una llamada para acceder a hardware del dispositivo, el sistema Android carga el módulo de biblioteca para el componente de hardware en cuestión.

Runtime de Android (máquina virtual)

Para los dispositivos con Android 5.0 (nivel de *API* 21) o versiones posteriores, cada app ejecuta sus propios procesos con sus propias instancias de la máquina virtual de Android (ART). El ART está escrito para ejecutar varias máquinas virtuales en dispositivos de memoria baja ejecutando archivos DEX, un formato de código de bytes diseñado especialmente para Android y optimizado para ocupar un espacio de memoria mínimo. Crea cadenas de herramientas, como Jack, y compila fuentes de Java en código de bytes DEX que se pueden ejecutar en la plataforma Android.

Estas son algunas de las funciones principales del ART:

- **Compilación** ahead-of-time (AOT) y just-in-time (JIT);
- **Recolección** de elementos no usados (GC) optimizada;
- Mejor compatibilidad con la **depuración**, como un generador de perfiles de muestras dedicado, excepciones de diagnóstico detalladas e informes de fallos, y la capacidad de establecer puntos de control para controlar campos específicos.

Antes de Android 5.0 (nivel de *API* 21), Dalvik era la máquina virtual del sistema operativo. En Android también se incluye un conjunto de bibliotecas de la máquina virtual, que proporcionan la mayor parte de la funcionalidad del lenguaje de programación Java.

Bibliotecas C/C++ nativas

Muchos componentes y servicios centrales del sistema Android, como el ART y la HAL, se basan en código nativo que requiere bibliotecas nativas escritas en C y C++. La plataforma Android proporciona la *API* del *framework* de Java para facilitar la funcionalidad de dichas bibliotecas nativas a las apps. Por ejemplo, el acceso a OpenGL ES a través de la OpenGL *API* del *framework* de Android.

Framework de la Java API

Todo el conjunto de funciones del SO Android está disponible mediante *API* escritas en el lenguaje Java. Estas *API* son los cimientos que necesitas para crear apps de Android simplificando la reutilización de componentes del sistema y servicios centrales y modulares, como los siguientes:

- Un **sistema de vista** enriquecido y extensible que se puede usar para compilar la Interfaz de Usuario de una app (listas, cuadros de texto, botones, etc.).
- Un **administrador de recursos** para acceder a recursos como *strings*, gráficos y archivos de diseño.
- Un **administrador de notificaciones** para permitir que las apps muestren alertas personalizadas en la barra de estado.

- Un **administrador de actividad** para gestionar el ciclo de vida de las apps.
- Proveedores de contenido que facilita el acceso de las aplicaciones a datos desde otras aplicaciones, como la app de Contactos.

Aplicaciones del sistema

El sistema Android incluye un conjunto de aplicaciones de sistema como correo electrónico, mensajería SMS, calendario, navegación en Internet, contactos, etc. Estas aplicaciones se encuentran instaladas en el sistema por defecto. El funcionamiento de estas aplicaciones del sistema es el mismo que el de una aplicación de usuario.

2.2.1 Versiones de Android [\[1\]](#)

A continuación, se muestra una tabla, con el histórico de versiones liberadas de Android hasta la actualidad:

Nombre código	Número de versión	Fecha de lanzamiento	Nivel de API
Apple Pie	1.0	23 de septiembre de 2008	1
Banana Bread	1.1	9 de febrero de 2009	2
Cupcake	1.5	27 de abril de 2009	3
Donut	1.6	15 de septiembre de 2009	4
Eclair	2.0–2.1	26 de octubre de 2009	5–7
Froyo	2.2–2.2.3	20 de mayo de 2010	8
Gingerbread	2.3–2.3.7	6 de diciembre de 2010	9–10
Honeycomb	3.0–3.2.6	22 de febrero de 2011	11–13
Ice Cream Sandwich	4.0–4.0.4	18 de octubre de 2011	14–15
Jelly Bean	4.1–4.3.1	9 de julio de 2012	16–18
KitKat	4.4–4.4.4, 4.4W–4.4W.2	31 de octubre de 2013	19–20
Lollipop	5.0–5.1.1	12 de noviembre de 2014	21–22
Marshmallow	6.0–6.0.1	5 de octubre de 2015	23
Nougat	7.0 - 7.1	22 de agosto de 2016	24 - 25

(Tabla 2.2) Histórico de versiones de Android

SECCIÓN II - PLAN DE TRABAJO

3- Contexto y justificación del trabajo

El objetivo del presente proyecto, consiste en la creación de una aplicación móvil para facilitar la comunicación entre una pequeña empresa de desarrollo de Software y sus clientes. Dando a estos últimos la posibilidad de reportar incidencias, enviar mejoras, nuevas solicitudes de desarrollo y cualquier otra comunicación relacionada con el producto desarrollado.

La idea del desarrollo, surgió de la experiencia personal en una pequeña empresa de desarrollo de Software, con un bajo nivel de madurez procedimental. Durante mi trabajo en dicha empresa he detectado grandes carencias en los procesos de gestión y en la comunicación con los clientes, principalmente en el seguimiento y control de incidencias. Actualmente el reporte de incidencias se realiza mediante el envío de un mail o la llamada telefónica al departamento de soporte, requiriendo que la persona encargada de la incidencia registre la misma en la aplicación correspondiente para su posterior seguimiento. Esta forma de trabajo provoca insatisfacción por parte de los clientes y derivado de ello, pérdidas económicas para la empresa.

La solución, aunque adaptada al contexto comentado anteriormente, puede ser utilizada en diferentes tipos de proyecto, aprovechando el potencial que proporciona el uso de servicios web en su integración con diferentes herramientas de gestión existentes en el mercado.

En el mercado actual, existen múltiples herramientas de gestión de incidencias que disponen de su versión móvil. Pero al existir diferentes aplicaciones *OpenSource* ya consolidadas en la empresa, se decidió la creación de un producto que integrará dichas aplicaciones con el objetivo final de conseguir una mejora en los procesos y mejorar la satisfacción de los clientes.

4- Objetivos del trabajo

A continuación, se describen los requisitos de la aplicación, diferenciando los funcionales de los no funcionales, dado que estos son a alto nivel, serán detallados en posteriores entregas:

4.1 Requisitos de alto nivel

Requisitos funcionales

- Mejorar los procedimientos y automatismos del trabajo de la empresa.
- Unificar diferentes sistemas ya implantados y estables mediante Servicios Web.
- Facilitar y mejorar la comunicación entre los clientes y la empresa.
- Aumentar el valor y la rentabilidad empresarial.

Requisitos no funcionales

- La aplicación será implementada en lenguaje nativo Android.
- La aplicación será eficiente, proporcionando tiempos de acceso rápidos.
- La aplicación cumplirá con los criterios básicos de usabilidad (facilidad de aprendizaje, facilidad de uso, flexibilidad y robustez).
- La aplicación cumplirá con criterios esenciales de la seguridad de la información (confidencialidad, integridad, disponibilidad y autenticación).
- La aplicación deberá ser compatible con móviles y *tablets* con sistema operativo Android.

4.2 Elección Tecnológica

Una vez detectado el objetivo del proyecto, se barajaron diferentes opciones para su desarrollo, como por ejemplo una aplicación Web accesible a través de cualquier dispositivo que disponga de navegador web y conexión a internet. Analizado este punto, la tendencia actual del mercado de desarrollo de aplicaciones y la previsible tendencia a futuro de este tipo de tecnologías, recomendaba darle un enfoque de movilidad. Por este motivo, se decidió realizar el trabajo de final de carrera en el área de desarrollo de aplicaciones móviles.

A nivel tecnológico, se eligió Android de manera nativa, especialmente por su implantación en el mercado, y por el conocimiento personal previo en el desarrollo de este tipo de aplicaciones. Además de su simplicidad en el desarrollo, Android ofrece resultados que aúnan un gran rendimiento con una experiencia de usuario satisfactoria, descartando del análisis otras opciones híbridas como PhoneGap, que si bien su resultado puede ser altamente satisfactorio, no alcanzan el nivel de rendimiento de una aplicación nativa.

Por el lado del servidor y la gestión de contenidos, no hubo lugar a elección puesto que se trata de una solución adaptada a un entorno de trabajo ya implantado, siendo Redmine la aplicación a utilizar, haciendo uso de la *API Rest* que proporciona. La única elección por motivos puramente de experiencia personal es el uso de JSON para la comunicación de la aplicación con Redmine.

4.3 Dispositivo objetivo

Como se ha comentado anteriormente, la aplicación deberá cumplir los objetivos técnicos mencionados en los requisitos de alto nivel, debiendo ser soportada tanto en móviles como en *Tablets* que dispongan de sistema operativo Android.

Se recomienda que la versión del sistema operativo se encuentre entre la 4.0 y la 5.0 y el dispositivo posea una resolución de 1440x2560, para una óptima visualización.

4.4 Público objetivo

El público objetivo de la aplicación está claramente delimitado, siendo los potenciales usuarios, por un lado, las personas que usan el Software propietario para su trabajo y por otro los técnicos de la empresa desarrolladora de dicho Software. Este punto se desarrollará en detalle, en la fase de análisis del diseño centrado en el usuario.

5- Enfoque y método seguido

Para afrontar el proyecto, previamente se realizó un estudio del mercado actual para comparar posibles soluciones y mejoras. Pero al existir diferentes herramientas cuya implementación y madurez ha sido demostrada en la empresa como Redmine para el control de incidencias y un gestor de contenidos para la gestión de la web de la empresa, se decidió el desarrollo de un producto nuevo adaptado a las necesidades, para optimizar el trabajo y aprovechar el conocimiento y uso de las herramientas comentadas.

Dado el carácter dinámico del proyecto que nos ocupa, una aplicación móvil con alto grado de mejora y posibles cambios de requisitos durante el transcurso del proyecto. Se ha decidido usar una metodología ágil para su desarrollo incorporando a la fase de diseño, conceptos de del DCU (diseño centrado en el usuario).

Se establecerán las siguientes fases:

Fase 1 - Plan de trabajo

En esta fase se ha establecido el alcance y los objetivos del proyecto y su objetivo es la realización del documento de plan de trabajo.

Fase 2 - Diseño y arquitectura

Durante esta fase, se ha definido el diseño los prototipos y la arquitectura técnica del proyecto previo a su implementación. Dispondrá de las siguientes sub-fases:

- **Diseño centrado en el usuario:** El diseño centrado en el usuario consta de las siguientes tareas:
 - **Usuarios y contexto de uso:** Definir las características de los potenciales usuarios.
 - **Diseño conceptual:** Elaboración de escenarios de uso.
 - **Prototipado:** Creación de las representaciones visuales de la aplicación (*Wireframes*).
 - **Evaluación:** Definición de las tareas de evaluación para los usuarios.
- **Definición de casos de uso:** Se realizará un diagrama de casos de uso, definiendo los flujos y condiciones de los mismos.
- **Definición de la arquitectura:** Se definirá la arquitectura técnica de la aplicación indicando la comunicación entre los diferentes sistemas intervinientes en el proyecto.

Fase 3 - Implementación

En la fase de implementación, estará compuesta de pequeñas entregas funcionales que deberán ser evaluadas por el tutor. Se incluirán las siguientes sub-fases:

- **Desarrollo:** Codificación del producto.
- **Pruebas:** Se definirán y ejecutarán las pruebas tanto unitarias como funcionales.
- **Aceptación del producto:** Tras cada entrega funcional, procederá a la aceptación por parte del tutor.
- **Documentación (memoria):** Tras cada entrega se documentará la memoria final del TFM, con los resultados y avances obtenidos.

Fase 4 - Publicación y entrega final

Durante esta fase se realizarán las modificaciones, se integrará toda la información y se finalizará todo lo necesario para que el producto esté listo para su entrega y posterior publicación.

6- Planificación del trabajo

Como se ha comentado en el punto anterior, se utilizará una metodología basada en el diseño centrado en el usuario (DCU) durante la fase de diseño de la aplicación, incorporando iteraciones durante la fase de implementación.

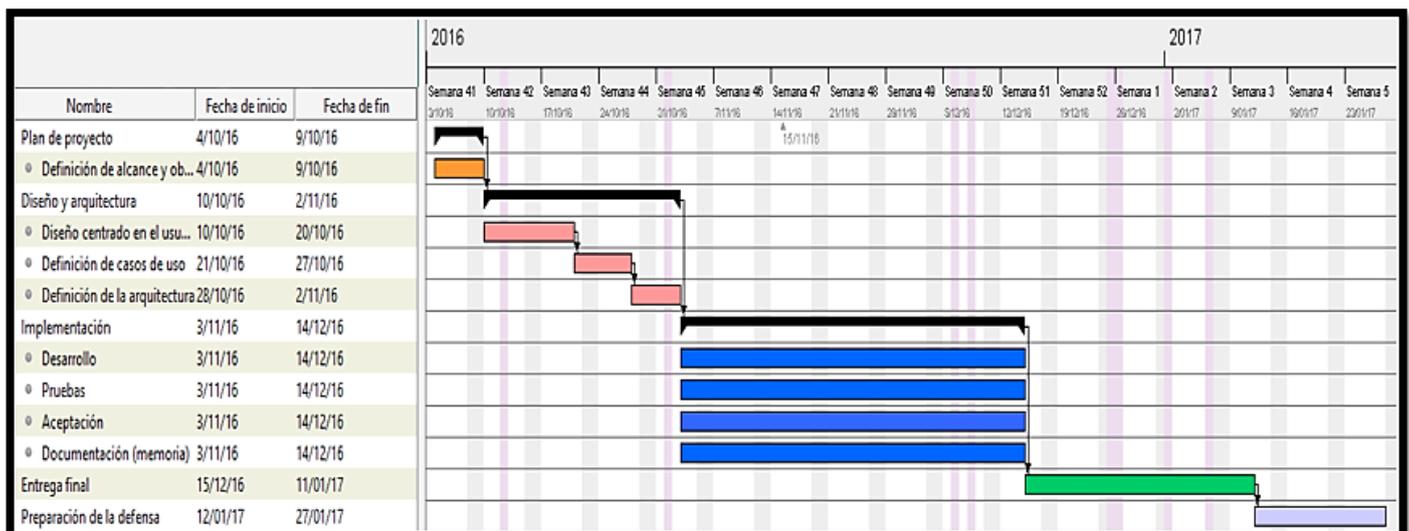
A continuación, se detallan las tareas con las horas estimadas:

Nombre de tarea	Duración	Comienzo	Fin
Fase 1 - Plan de proyecto	28 horas	04/10/16	09/10/16
Fase 2 - Diseño y arquitectura	94 horas	10/10/16	02/11/16
2.1 - Diseño centrado en el usuario	45 horas	10/10/16	20/10/16
2.2 - Definición de casos de uso	27 horas	21/10/16	27/10/16
2.3 - Definición de la arquitectura	22 horas	28/10/16	02/11/16
Fase 3 - Implementación	175 horas	03/11/2016	14/12/16
3.1 - Desarrollo	Al ser tareas iterativas, las entregas serán estimadas antes de cada entrega. Se dedicará un 20% del total de cada entrega a la realización de pruebas.		
3.2 - Pruebas			
3.3 - Aceptación del producto			
3.4 - Documentación (memoria)			
Fase 4 - Entrega final	100 horas	15/12/16	11/01/17
Preparación de la defensa	12 horas	23/01/17	27/01/17

(Tabla 6.1) Planificación general del proyecto

* Se presupone una dedicación diaria de entre 3 a 5 horas como máximo, incluyendo festivos, sábados y domingos.

Se adjunta además un diagrama Gantt con las tareas citadas anteriormente:



(Diagrama 6.1) Diagrama Gantt de planificación general del proyecto. [10]

6.1 Análisis de costes

Dada la condición académica del proyecto, en este apartado se presenta de manera simulada el presupuesto necesario si el proyecto tuviera un carácter comercial.

En la (Tabla 6.2) se especifican las horas empleadas en las distintas fases del proyecto, tanto las fases más técnicas de análisis y desarrollo, como la redacción de la memoria. De la suma de las diferentes fases, se obtiene un total de **397 horas**.

Fases del Proyecto

Fase	Horas
Plan de proyecto	28
Diseño y arquitectura	94
Implementación	175
Publicación y entrega final	100

(Tabla 6.2) Dedicación en horas al proyecto dividido por fases

Además del coste de personal de este proyecto, realizado por una sola persona, para un proyecto comercial de similares características se debe tener en cuenta el coste de material que es necesario tanto para la implantación de la arquitectura propuesta, como para el estudio inherente al proceso de seguimiento y documentación de todo el proyecto.

Estos materiales pueden dividirse en:

Tangibles como **hardware y recursos materiales**, e intangibles como las **licencias software**. Respecto a este último recurso el gasto igualmente ha sido nulo, puesto que se han utilizado licencias gratuitas en todo el software utilizado.

Siendo el objeto de este proyecto un Trabajo de fin de Master, se ha decidido obviar los costes asociados al material por no aplicar en el objetivo perseguido.

Estimando el precio a la hora de trabajo de un Ingeniero Informático a dedicación única para todas las tareas inherentes a la ingeniería del Software alrededor de unos 45€/hora, el coste total del personal de este proyecto asciende a **17.865 €**.

A partir de todos estos datos obtenidos, se obtiene el presupuesto global del proyecto, como se muestra en la (Tabla 6.3).

Concepto	Importe
Costes personal	17.865 €
Costes material	N/A
Base imponible	17.865 €
I.V.A. (21%)	3.751,65 €
TOTAL	21.616,65 €

(Tabla 6.3) Presupuesto del Proyecto

SECCIÓN III – CASO DE ESTUDIO ^[7]

7- Análisis

El objetivo principal de la fase de análisis es conocer las características de los usuarios, sus necesidades, objetivos y el contexto de uso de la aplicación objeto del proyecto. Por tanto, se detectarán las funcionalidades que deberá tener la aplicación para satisfacer las necesidades de los usuarios.

7.1 Método de investigación

Como punto de partida para la investigación, intentaremos responder a las siguientes preguntas:

- ¿Quiénes son los usuarios?
- ¿Cuáles son las tareas que realizarán?
- ¿Por qué utilizan el sistema?
- ¿Cómo utilizarán el sistema?
- ¿Para qué utilizarán la aplicación?
- ¿Qué dispositivos utilizan?
- ¿Qué les crea satisfacción y qué no les agrada?
- ¿Utilizan alguna aplicación similar?

Para llevar a cabo esta tarea, se utilizará el método de **observación e investigación contextual**, ayudándonos además por la realización de encuestas y entrevistas informales a los potenciales usuarios.

La justificación del uso de estas técnicas, tiene su base en el contexto de uso de la presente aplicación. Al ser un producto orientado a la gestión de incidencias y nuevos desarrollos de un Software empresarial, la investigación se ha basado en la observación del trabajo diario del departamento de desarrollo y su interacción con los clientes del Software desarrollado. Por tanto, las entrevistas irán dirigidas tanto al personal que utiliza el Software para realizar las ventas en los establecimientos comerciales, como a los trabajadores de la empresa desarrolladora que interactúan con ellos.

Dado el carácter de trabajo diario y la relación fluida con los clientes por parte de la empresa, se ha considerado que el uso de la observación contextual de los usuarios en su entorno habitual y las entrevistas, son los métodos más eficaces y sencillos de implementar para llevar a cabo la investigación. Por otro lado, al ser una aplicación de nueva creación, no ha sido posible realizar un análisis comparativo con productos similares.

7.1.1 Planteamiento

Como se ha comentado anteriormente, la investigación se ha basado en el trabajo diario de los departamentos de desarrollo y soporte de la empresa desarrolladora del Software propietario. Con el objeto de obtener las necesidades del cliente, se han realizado encuestas a un pequeño grupo de usuarios potenciales de la aplicación móvil. Estas entrevistas informales han consistido en preguntas con respuesta abierta, con la siguiente estructura:

Datos personales

- Edad
- Sexo
- Nivel de estudios
- Tipo de negocio
- Puesto en la empresa
- Discapacidad o limitación

Preguntas

- ¿Dispone de teléfono móvil o tableta?
- En caso afirmativo. ¿Qué sistema de operativo usa?
- ¿Hace uso de aplicaciones móviles?
- ¿Con que frecuencia? (Siendo 0 muy poco y 10 muy a menudo).
- ¿Qué nivel de conocimientos informáticos posee? (Siendo 0 un conocimiento nulo y 10 un nivel experto).
- ¿Qué opina de la forma de reportar las incidencias del Software que usa en su trabajo diario?
- ¿Está satisfecho con los tiempos de respuesta del servicio de soporte?
- ¿Cómo suele comunicarse con la empresa/cliente (telefónicamente, mail, etc.)?
- ¿Le parecería interesante hacer uso de aplicaciones móviles para la comunicación con la empresa / cliente? Indique porqué.
- ¿Conoce alguna aplicación similar?

7.1.2 Desarrollo

Se ha entrevistado a un grupo de reducido de 7 personas, 5 usuarios que utilizan el Software gestionado por la aplicación móvil para su trabajo diario y 2 trabajadores de la empresa desarrolladora.

Puesto que el Software de la empresa es utilizado en diferentes tipos de negocio, se ha procurado seleccionar un grupo representativo de usuarios de diferentes tipologías, obteniendo los siguientes datos:

Usuarios del cliente

Entrevistado 1

Datos personales

- Edad: 21.
- Sexo: Femenino.
- Nivel de estudios: ESO.
- Tipo de negocio: Venta de ropa.
- Puesto en la empresa: Vendedora.
- Discapacidad o limitación: Ninguna.

Preguntas

- **Dispone de teléfono móvil o tableta.**
Si.
- **En caso afirmativo. ¿Qué sistema de operativo usa?**
Android.

- **¿Hace uso de aplicaciones móviles?**
Si.
- **¿Con que frecuencia? (Siendo 0 muy poco y 10 muy a menudo).**
10.
- **¿Qué nivel de conocimientos informáticos posee? (Siendo 0 un conocimiento nulo y 10 un nivel experto).**
4.
- **¿Qué opina de la forma de reportar las incidencias del Software que usa en su trabajo diario?**
Es difícil y lenta.
- **¿Está satisfecho con los tiempos de respuesta del servicio de soporte?**
Es muy lento.
- **¿Cómo suele comunicarse con la empresa/cliente (telefónicamente, mail, etc.)?**
Principalmente telefónicamente.
- **¿Le parecería interesante hacer uso de aplicaciones móviles para la comunicación con la empresa / cliente? Indique porqué.**
- Sí, porque creo que sería más rápido y ágil.
- **¿Conoce alguna aplicación similar?**
No.

Entrevistado 2

Datos personales

- **Edad:** 26.
- **Sexo:** Femenino.
- **Nivel de estudios:** Bachiller.
- **Tipo de negocio:** Juguetería.
- **Puesto en la empresa:** Vendedora.
- **Discapacidad o limitación:** Ninguna.

Preguntas

- **Dispone de teléfono móvil o tableta:**
Si.
- **En caso afirmativo. ¿Qué sistema de operativo usa?**
IOS.
- **¿Hace uso de aplicaciones móviles?**
Si.
- **¿Con que frecuencia? (Siendo 0 muy poco y 10 muy a menudo).**
9.
- **¿Qué nivel de conocimientos informáticos posee? (Siendo 0 un conocimiento nulo y 10 un nivel experto).**
5.
- **¿Qué opina de la forma de reportar las incidencias del Software que usa en su trabajo diario?**
Es un poco tediosa e ineficiente.
- **¿Está satisfecho con los tiempos de respuesta del servicio de soporte?**
Es muy lento.
- **¿Cómo suele comunicarse con la empresa/cliente (telefónicamente, mail, etc.)?**
Principalmente telefónicamente.

- ¿Le parecería interesante hacer uso de aplicaciones móviles para la comunicación con la empresa / cliente? Indique porqué.
- Sí, porque creo que sería más rápido.
- ¿Conoce alguna aplicación similar?
No.

Entrevistado 3

Datos personales

- **Edad:** 41.
- **Sexo:** Masculino.
- **Nivel de estudios:** Diplomado universitario.
- **Tipo de negocio:** Supermercado.
- **Puesto en la empresa:** Responsable.
- **Discapacidad o limitación:** Ninguna.

Preguntas

- **Dispone de teléfono móvil o tableta:**
Sí.
- **En caso afirmativo. ¿Qué sistema de operativo usa?**
Android.
- **¿Hace uso de aplicaciones móviles?**
Sí.
- **¿Con que frecuencia? (Siendo 0 muy poco y 10 muy a menudo).**
6.
- **¿Qué nivel de conocimientos informáticos posee? (Siendo 0 un conocimiento nulo y 10 un nivel experto).**
9.
- **¿Qué opina de la forma de reportar las incidencias del Software que usa en su trabajo diario?**
Me parece que es un poco rudimentaria.
- **¿Está satisfecho con los tiempos de respuesta del servicio de soporte?**
No, tardan mucho.
- **¿Cómo suele comunicarse con la empresa/cliente (telefónicamente, mail, etc.)?**
Por teléfono.
- **¿Le parecería interesante hacer uso de aplicaciones móviles para la comunicación con la empresa / cliente? Indique porqué.**
Yo creo que mejoraría. Podría gestionar las incidencias y el estado de los desarrollos desde casa.
- **¿Conoce alguna aplicación similar?**
No.

Entrevistado 4

Datos personales

- **Edad:** 23.
- **Sexo:** Femenino.
- **Nivel de estudios:** Bachiller.
- **Tipo de negocio:** Joyería.
- **Puesto en la empresa:** Vendedora.

- **Discapacidad o limitación:** Ninguna.

Preguntas

- **Dispone de teléfono móvil o tableta:**
Si.
- **En caso afirmativo. ¿Qué sistema de operativo usa?**
Android.
- **¿Hace uso de aplicaciones móviles?**
Si.
- **¿Con que frecuencia? (Siendo 0 muy poco y 10 muy a menudo).**
5.
- **¿Qué nivel de conocimientos informáticos posee? (Siendo 0 un conocimiento nulo y 10 un nivel experto).**
4.
- **¿Qué opina de la forma de reportar las incidencias del Software que usa en su trabajo diario?**
Difícil y lenta.
- **¿Está satisfecho con los tiempos de respuesta del servicio de soporte?**
Mejoraría la forma de contactar con la empresa, porque solo permiten llamar por teléfono o enviar mails.
- **¿Cómo suele comunicarse con la empresa/cliente (telefónicamente, mail, etc.)?**
Principalmente telefónicamente.
- **¿Le parecería interesante hacer uso de aplicaciones móviles para la comunicación con la empresa / cliente? Indique porqué.**
Creo que sí, pero no estoy segura.
- **¿Conoce alguna aplicación similar?**
No.

Entrevistado 5

Datos personales

- **Edad:** 43.
- **Sexo:** Femenino.
- **Nivel de estudios:** Licenciado universitario.
- **Tipo de negocio:** Venta de ropa.
- **Puesto en la empresa:** Responsable.
- **Discapacidad o limitación:** Ninguna.

Preguntas

- **Dispone de teléfono móvil o tableta:**
Si.
- **En caso afirmativo. ¿Qué sistema de operativo usa?**
Android.
- **¿Hace uso de aplicaciones móviles?**
Si.
- **¿Con que frecuencia? (Siendo 0 muy poco y 10 muy a menudo).**
7.
- **¿Qué nivel de conocimientos informáticos posee? (Siendo 0 un conocimiento nulo y 10 un nivel experto).**
10.

- **¿Qué opina de la forma de reportar las incidencias del Software que usa en su trabajo diario?**
Nos cuesta mucho desde nuestra empresa gestionar las incidencias y los nuevos desarrollos, puesto que la comunicación con el proveedor de desarrollo es muy limitada.
- **¿Está satisfecho con los tiempos de respuesta del servicio de soporte?**
No, en absoluto.
- **¿Cómo suele comunicarse con la empresa/cliente (telefónicamente, mail, etc.)?**
Suelo llamar por teléfono y a veces por email.
- **¿Le parecería interesante hacer uso de aplicaciones móviles para la comunicación con la empresa / cliente? Indique porqué.**
Sí sobre todo para abrir otra opción de comunicación entre nosotros y el proveedor.
- **¿Conoce alguna aplicación similar?**
No.

Usuarios técnicos

Entrevistado 1

Datos personales

- **Edad:** 34.
- **Sexo:** Masculino.
- **Nivel de estudios:** Graduado y Master universitario.
- **Tipo de negocio:** Empresa desarrolladora.
- **Puesto en la empresa:** Jefe de proyecto Software.
- **Discapacidad o limitación:** Ninguna.

Preguntas

- **Dispone de teléfono móvil o tableta:**
Sí.
- **En caso afirmativo. ¿Qué sistema de operativo usa?**
Android.
- **¿Hace uso de aplicaciones móviles?**
Sí.
- **¿Con que frecuencia? (Siendo 0 muy poco y 10 muy a menudo).**
9.
- **¿Qué nivel de conocimientos informáticos posee? (Siendo 0 un conocimiento nulo y 10 un nivel experto).**
10.
- **¿Qué opina de la forma de reportar las incidencias del Software que usa en su trabajo diario?**
La comunicación con los clientes es muy complicada. Gestionar mails y llamadas es tedioso.
- **¿Está satisfecho con los tiempos de respuesta del servicio de soporte?**
No, los tiempos de respuesta son demasiado altos.
- **¿Cómo suele comunicarse con la empresa/cliente (telefónicamente, mail, etc.)?**
Principalmente por mail.
- **¿Le parecería interesante hacer uso de aplicaciones móviles para la comunicación con la empresa / cliente? Indique porqué.**
Sí, porque mejoraría la comunicación con los clientes.
- **¿Conoce alguna aplicación similar?**
No.

Entrevistado 2

Datos personales

- **Edad:** 40.
- **Sexo:** Masculino.
- **Nivel de estudios:** Formación profesional superior.
- **Tipo de negocio:** Empresa desarrolladora.
- **Puesto en la empresa:** Técnico de sistemas.
- **Discapacidad o limitación:** Ninguna.

Preguntas

- **Dispone de teléfono móvil o tableta:**
Sí.
- **En caso afirmativo. ¿Qué sistema de operativo usa?**
Android.
- **¿Hace uso de aplicaciones móviles?**
Sí.
- **¿Con que frecuencia? (Siendo 0 muy poco y 10 muy a menudo).**
10.
- **¿Qué nivel de conocimientos informáticos posee? (Siendo 0 un conocimiento nulo y 10 un nivel experto).**
10.
- **¿Qué opina de la forma de reportar las incidencias del Software que usa en su trabajo diario?**
Tardamos mucho tiempo en registrar los correos y llamadas que nos llegan y registrarlos en Redmine.
- **¿Está satisfecho con los tiempos de respuesta del servicio de soporte?**
No.
- **¿Cómo suele comunicarse con la empresa/cliente (telefónicamente, mail, etc.)?**
Principalmente por teléfono.
- **¿Le parecería interesante hacer uso de aplicaciones móviles para la comunicación con la empresa / cliente? Indique porqué.**
Sí, porque nos evitaría trabajo de gestión.
- **¿Conoce alguna aplicación similar?**
No.

7.1.2 Resultados y Conclusiones

El principal resultado de la fase de análisis, es la determinación de los perfiles de usuario. Puesto que el objeto del presente proyecto es el desarrollo de una aplicación móvil para la gestión de un Software propietario, para la entrevista se diferenciaron como usuarios potenciales de la aplicación, las personas que utilizan el Software de los que desarrollan y dan soporte al mismo.

Los resultados permiten extraer que el perfil de los usuarios del cliente depende en gran medida del negocio del mismo y del rol en la empresa. No es el mismo perfil de usuario el vendedor de una juguetería que de una tienda de ropa o un supermercado y una persona responsable de los sistemas de información de un dependiente de tienda. La gran mayoría de dependientes poseen un rango de edad entre 18 y 26 años y compaginan sus estudios con el trabajo a tiempo parcial. El responsable del negocio suele disponer de estudios superiores y su edad estaría entre los 40 y 50 años.

Por otro lado, los usuarios técnicos de la empresa desarrolladora, poseen un perfil de estudios técnicos y su edad está comprendida entre los 26 y 40 años.

7.2 Usuarios y contexto de uso

En este punto se definirán los siguientes puntos, obtenidos mediante la fase de investigación y que servirán como punto de partida para la fase de diseño:

- Perfiles de usuarios.
- Roles de usuarios.
- Contextos de uso.
- Análisis de tareas.

7.2.1 Perfiles de usuario

Gracias a las entrevistas y la observación de la fase de investigación, se han detectado diferentes perfiles, entendiendo estos como agrupaciones de usuarios según sus características. Estos perfiles, incluyen además aspectos sociales y etnográficos diferenciadores como, ubicación geográfica, características sociales y otros aspectos sociales y antropológicos.

Para el uso de la aplicación móvil del presente proyecto, no se requiere que los usuarios posean un conocimiento experto en Internet (entendiendo por expertos a aquellos que lo utilizan en el ámbito laboral o programadores de sitios web), ni usuarios expertos en aplicaciones móviles. Se procurará que los futuros usuarios estén familiarizados con las aplicaciones móviles e internet en general. Comprendiendo, que dentro de los usuarios del cliente puede haber varios niveles de conocimiento de nuevas tecnologías, en este caso se englobarán en el mismo tipo de perfil a efectos del análisis de la usabilidad.

Dentro de la heterogeneidad, todos los usuarios potenciales tendrán una característica común, el producto Software propietario desarrollado, y su principal diferencia será, el nivel de acceso y uso que hacen de ese producto. Analizada esta diferencia se han detectado dos perfiles, que son los siguientes:

- **Usuarios técnicos (trabajadores de la empresa desarrolladora):** Los usuarios desarrolladores y técnicos de soporte, se encargarán del mantenimiento del Software propietario y por tanto dispondrán de acceso total a la aplicación móvil.
- **Usuarios clientes (usuarios del Software desarrollado por la empresa):** Los usuarios clientes serán las personas que utilizarán el Software desarrollado por la empresa para su trabajo diario y usarán la aplicación móvil para la comunicación con la empresa proveedora del Software.

A modo de resumen, se establecen las principales características de los perfiles detectados:

Características de los perfiles de usuario	
Perfil	Técnico
Características	<ul style="list-style-type: none">• Personas cualificadas con un nivel de estudios medio/alto.• Personas con diferentes rangos de edad.• Genero indistinto.• Ámbito profesional de la informática y TIC's.• Conocimientos en informática nivel profesional.
Perfil	Cliente
Características	<ul style="list-style-type: none">• Personas con diferentes niveles de estudios.• Personas con diferentes rangos de edad.• Genero indistinto.• Ámbito profesional sector de empresas comerciales.• Conocimientos de informática bajos / medios.

(Tabla 7.1) Características de los perfiles de usuario

7.2.2 Contextos de uso

Para el uso de la aplicación, objeto de este proyecto, se requiere el uso de un dispositivo móvil (tableta o Smartphone), que disponga de un sistema operativo Android y conexión a Internet móvil.

A nivel de competencias, para la utilización de la aplicación por parte de los usuarios, basta con tener conocimientos básicos para el uso de aplicaciones móviles.

El uso habitual de esta aplicación móvil, se desarrollará por parte de los clientes de la empresa desarrolladora de Software en su trabajo diario. Al detectar una incidencia o requerir una mejora, enviarán sus datos (previa identificación) mediante la aplicación móvil.

Otro contexto de uso habitual sería aquel donde el usuario, envía comunicaciones y sugerencias a la empresa y consulta el estado de sus peticiones desde cualquier lugar donde se encuentre, mediante un dispositivo móvil.

Contextos de uso de los perfiles de usuario		
Perfil	Técnico	
Usuario	Administrador	
Nivel social	Sexo	Indistinto
	Edad	De 26 a 40 años
Nivel formativo	Nivel de estudios	Enseñanza profesional superior Estudios Universitarios
	Conocimientos informáticos	Experto
	Experiencia en entornos similares	Alta
Nivel técnico	Autonomía/Conexión	Alta
	Soporte de hardware requerido	No necesario
	Hardware requerido	Equipo informático avanzado
	Hardware adicional	Servidor
	Tipo de conexión requerida	Calidad Alta
	Software requerido	Específico de diseño y mantenimiento de Software
Lugar / Frecuencia	Puesto de trabajo / diario	
Perfil	Cliente	
Usuario	Cliente Vendedor	
Nivel social	Sexo	Indistinto
	Edad	De 18 a 26 años
Nivel formativo	Nivel de estudios	ESO, Bachiller, estudiantes universitarios.
	Conocimientos informáticos	Bajo/medio.
	Experiencia en entornos similares	Baja
Nivel técnico	Autonomía/Conexión	No necesaria
	Soporte de hardware requerido	Nivel usuario
	Hardware requerido	Equipo informático con conexión a internet
	Hardware adicional	Impuesto por la empresa Lectores de códigos de barras, impresoras, etc.
	Tipo de conexión requerida	Normal
	Software requerido	Impuesto por la empresa desarrolladora
Lugar / Frecuencia	Puesto de trabajo / diario	
Usuario	Cliente Responsable	
Nivel social	Sexo	Indistinto
	Edad	De 40 a 50 años
Nivel formativo	Nivel de estudios	Estudios universitarios.
	Conocimientos informáticos	Bajo
	Experiencia en entornos similares	Sin experiencia
Nivel técnico	Autonomía/Conexión	No necesaria
	Soporte de hardware requerido	Nivel usuario
	Hardware requerido	Equipo informático con conexión a internet
	Hardware adicional	Impuesto por la empresa Lectores de códigos de barras, impresoras, etc.

	Tipo de conexión requerida	Normal
	Software requerido	Impuesto por la empresa desarrolladora
Lugar / Frecuencia	Puesto de trabajo / diario	

(Tabla 7.2) Contextos de uso de los perfiles de usuario

7.2.3 Análisis de tareas

Las principales tareas que proporcionará la aplicación son las siguientes:

- **Tarea 1:** Registrar usuario.
- **Tarea 2:** Registrar incidencia de Software (Funcionamiento erróneo en el Software).
- **Tarea 3:** Registrar incidencia de Sistemas (Bloqueo de Servidor, maquina, etc.).
- **Tarea 4:** Solicitar nuevo desarrollo (Software).
 - **Tarea 4.1:** Aceptar presupuesto
 - **Tarea 4.2:** Cancelar presupuesto.
- **Tarea 5:** Enviar sugerencias.
- **Tarea 6:** Visualizar preguntas frecuentes (FAQS).
- **Tarea 7:** Descargar manual.
- **Tarea 8:** Mantenimiento y configuración de la aplicación.
 - **Tarea 8.1:** Enviar presupuesto.
 - **Tarea 8.2:** Cambiar estado de las solicitudes.

7.2.3.1 Roles

En este punto se realizará una primera aproximación a las tareas detectadas y los usuarios que las realizarán.

Roles de los perfiles de usuario	
Usuario	Administrador
Tareas	<ul style="list-style-type: none"> • El usuario administrador podrá realizar todas las funciones. • Tarea 8: Además se encargará del mantenimiento y configuración de la aplicación. <ul style="list-style-type: none"> ○ Tarea 8.1: Enviar presupuesto de desarrollo de Software. ○ Tarea 8.2: Modificar estado de solicitudes (Incidencias resueltas, desarrollos finalizados, etc.).
Usuario	Cliente responsable
Tareas	<ul style="list-style-type: none"> • Tarea 1: Registrar usuario. • Tarea 2: Registrar incidencia de Software (Funcionamiento erróneo en el Software). • Tarea 3: Registrar incidencia de Sistemas (Bloqueo de Servidor, maquina, etc.). • Tarea 4: Solicitar nuevo desarrollo (Software). <ul style="list-style-type: none"> ○ Tarea 4.1: Aceptar presupuesto. ○ Tarea 4.2: Cancelar presupuesto. • Tarea 5: Enviar sugerencias. • Tarea 6: Visualizar preguntas frecuentes (FAQS). • Tarea 7: Descargar manual.
Usuario	Cliente vendedor
Tareas	<ul style="list-style-type: none"> • Tarea 1: Registrar usuario. • Tarea 2: Registrar incidencia de Software (Funcionamiento erróneo en el Software). • Tarea 3: Registrar incidencia de Sistemas (Bloqueo de Servidor, maquina, etc.). • Tarea 5: Enviar sugerencias. • Tarea 6: Visualizar preguntas frecuentes (FAQS). • Tarea 7: Descargar manual.

(Tabla 7.3) Roles de los perfiles de usuario

8- Diseño conceptual

En este segundo apartado, comienza la fase de diseño y en él se definirán:

- Escenarios de uso
- Diagramas de interacción

8.1 Escenarios de uso

A continuación, se detallan una serie de posibles escenarios de uso de la aplicación:

Escenario 1	
Perfil de usuario	Cliente
Personaje	Antonio Pérez tiene 23 años, es estudiante de psicología y trabaja a tiempo parcial, en una tienda de juguetes, para pagar sus estudios. Antonio, como la mayor parte de los jóvenes, utiliza a diario dispositivos y aplicaciones móviles, por lo que el uso de estos, no es un problema en absoluto.
Descripción	En los últimos días en su trabajo, Antonio está teniendo problemas para vender determinados productos mediante el Software implementado en los terminales de venta. Por ello Antonio, ha abierto la nueva aplicación móvil que lleva instalada en su tableta y ha registrado los problemas que está sufriendo, para su posterior resolución.
Contexto	Puesto de trabajo / diario
Objetivo	Dar de alta una incidencia del Software.
Tareas	Registrar incidencia de Software (Tarea 2).
Necesidad de información	Registro en el sistema e identificación.
Funcionalidades necesarias	<ul style="list-style-type: none">• Registro de usuario• Validación del usuario.• Alta de incidencia de Software.• Registro de la información.
Desarrollo de las tareas	Validar el usuario registrado y registrar la información necesaria para dar de alta la incidencia.
Escenario 2	
Perfil de usuario	Cliente
Personaje	Clara Gómez es una chica de 22 años, que está finalizando sus estudios de veterinaria y compagina estos como dependienta en una tienda de ropa. Clara usa a diario su teléfono móvil para comunicarse con sus amigos.
Descripción	La semana pasada, Clara tuvo un problema con uno de los terminales, en concreto uno de las cajas de venta. Al intentar acceder a la aplicación recibió un mensaje de error indicando que la comunicación con el servidor no era posible. Juana abrió la aplicación móvil y registró la incidencia como un problema del sistema (Al no poder acceder al terminal), para ello además de escribir una descripción para el problema sucedido, realizó una captura del código BIDI que todos los terminales instalados tienda disponen.
Contexto	Puesto de trabajo / diario
Objetivo	Dar de alta una incidencia del Sistema.
Tareas	Registrar incidencia de Sistemas (Tarea 3).
Necesidad de información	Registro en el sistema, Identificación e Identificación de la maquina mediante código BIDI.

Funcionalidades necesarias	<ul style="list-style-type: none"> • Registro de usuario • Validación del usuario. • Alta de incidencia de Sistema. • Escaneo de código BIDI. • Registro de la información.
Desarrollo de las tareas	Validar el usuario registrado, Identificación del código BIDI y recepción de la información de la incidencia de Sistemas.
Escenario 3	
Perfil de usuario	Técnico
Personaje	Marta Fernández es Técnica Superior en Administración de Sistemas Informáticos y trabaja como Técnica de Sistemas en la empresa de desarrollo de Software.
Descripción	Marta recibe a diario numerosas incidencias de Sistemas en su herramienta de gestión interna (Redmine). Estas incidencias son reportadas directamente desde la nueva aplicación móvil. Al recibir los datos del código BIDI y la descripción Marta puede acceder remotamente a la máquina y subsanar el problema, interactuando así con los usuarios que pueden visualizar el estado de sus solicitudes en tiempo real.
Contexto	Puesto de trabajo / diario
Objetivo	Gestión de las incidencias recibidas.
Tareas	Modificar estado de solicitudes (Incidencias resueltas, desarrollos finalizados, etc.) (Tarea 8.2).
Necesidad de información	Identificación en el sistema Opensource (Redmine).
Funcionalidades necesarias	<ul style="list-style-type: none"> • Validación del usuario. • Modificación del estado de la solicitud / incidencia. • Registro de la información.
Desarrollo de las tareas	Identificación y validación en el sistema <i>Opensource</i> y registro de la información modificada en el sistema.
Escenario 4	
Perfil de usuario	Técnico
Personaje	José Sanchez es Ingeniero Informático y trabaja en la empresa desarrolladora del Software, responsabilizándose del departamento de desarrollo.
Descripción	José recibe en Redmine, una solicitud de nuevo desarrollo enviado por un cliente a través de la nueva aplicación móvil. Como responsable del desarrollo en cuestión, estima el tiempo necesario para el desarrollo y se lo envía al cliente mediante Redmine.
Contexto	Puesto de trabajo / diario
Objetivo	Enviar un presupuesto solicitado por el cliente.
Tareas	Enviar presupuesto de desarrollo de Software (Tarea 8.1).
Necesidad de información	Identificación en el sistema <i>Opensource</i> (Redmine).
Funcionalidades necesarias	<ul style="list-style-type: none"> • Validación del usuario. • Modificación del estado de la solicitud / incidencia. • Registro de la información.
Desarrollo de las tareas	Identificación y validación en el sistema <i>Opensource</i> y registro de la información modificada en el sistema.
Escenario 5	
Perfil de usuario	Cliente
Personaje	Pedro tiene 45 años, es Licenciado en económicas y trabaja como encargado de un negocio de venta de ropa.

Descripción	Pedro tiene una necesidad de cambio del Software de su negocio. Puesto que Pedro viaja a menudo y no está habitualmente en un entorno de oficina, utiliza la nueva aplicación móvil para solicitar la adaptación que necesita, quedando a la espera del envío del presupuesto por parte de la empresa desarrolladora.
Contexto	Puesto de trabajo / diario
Objetivo	Solicitar nuevo desarrollo (presupuesto).
Tareas	Solicitar nuevo desarrollo (Software) (Tarea 4) .
Necesidad de información	Registro en el sistema e identificación.
Funcionalidades necesarias	<ul style="list-style-type: none"> • Registro de usuario • Validación del usuario. • Alta de incidencia de Software. • Registro de la información.
Desarrollo de las tareas	Validar el usuario registrado y registrar la información necesaria para dar de alta la incidencia.

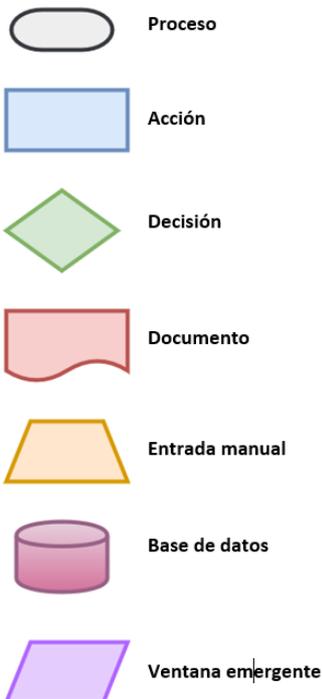
(Tabla 8.1) Escenarios de uso

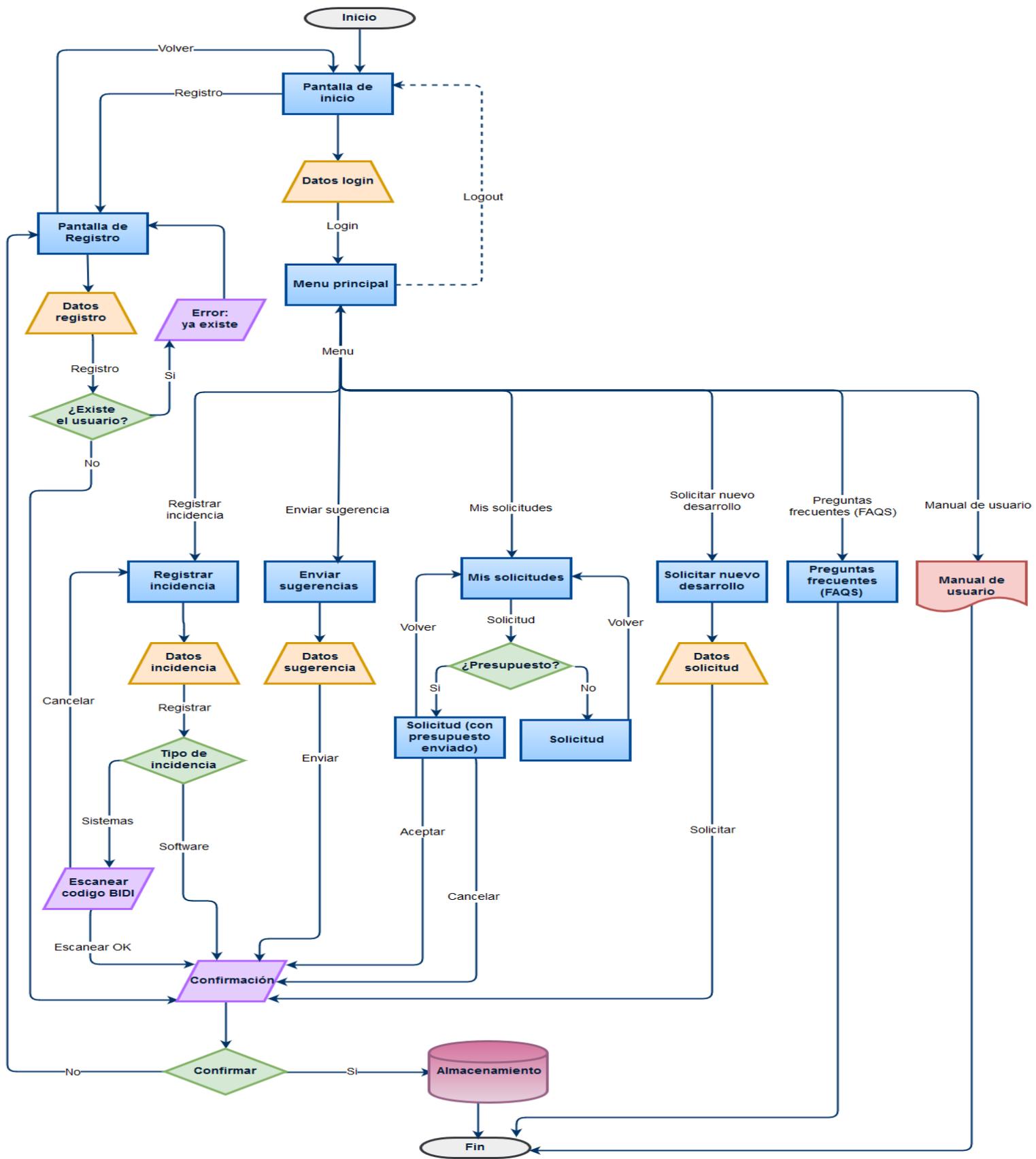
8.2 Flujos de interacción

A continuación, se mostrará el flujo de interacción detallado con las tareas detectadas en la fase de análisis.

Para facilitar la comprensión se añade la siguiente simbología a modo de leyenda:

Simbología





(Diagrama 8.1) Flujos de interacción

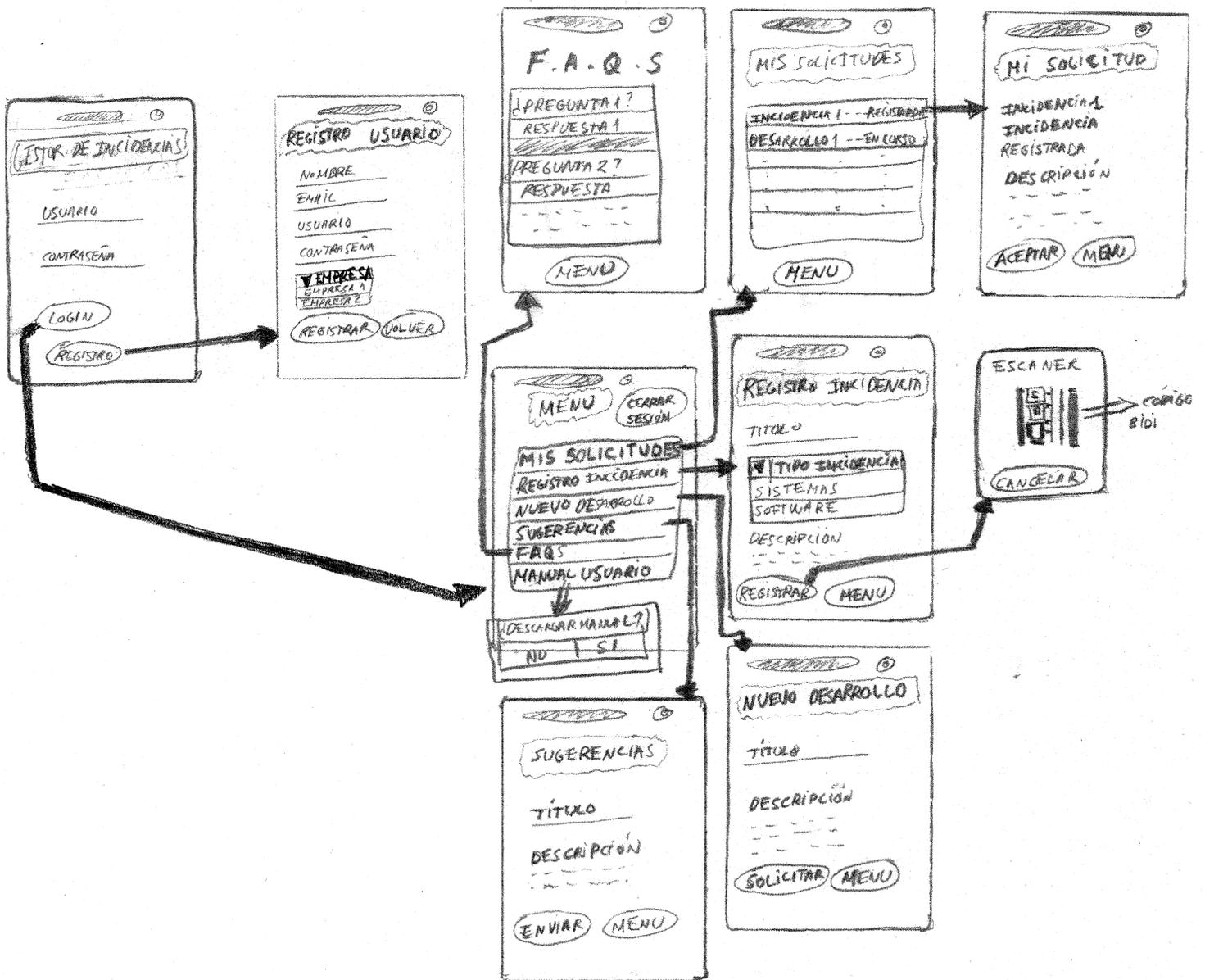
9- Prototipado

El objetivo de esta fase, es realizar una primera aproximación al modelo de las interfaces que tendrá el sistema, sin entrar en mucho detalle visual.

Este apartado se ha dividido en dos partes, comenzando por diseños de bajo detalle a mano alzada **Sketches** y un punto posterior con más detalle **Prototipos**.

9.1 Sketches

Se muestra el diagrama realizado a mano alzada con sus interacciones:



(Diagrama 9.1) Sketches

9.2 Prototipos [\[11\]](#)

Este punto, se dividirá cada pantalla, explicando los elementos que contiene cada diseño, comenzando por la pantalla de inicio y menú principal y describiendo a continuación las diferentes funcionalidades de la aplicación.

9.2.1 Pantalla de inicio



(Figura 9.1) Prototipo pantalla de inicio

Esta pantalla será la primera que aparecerá al arrancar la aplicación. En ella el usuario podrá identificarse en la aplicación introduciendo su "usuario" y "contraseña" y pulsando el botón "Login" o acceder a la pantalla de registro de usuario pulsando "Registro".

9.2.2 Pantalla de registro de usuario

Registro de usuario

Nombre
Pepe

Email
pepe@gmail.com

Usuario
psanchez

Contraseña

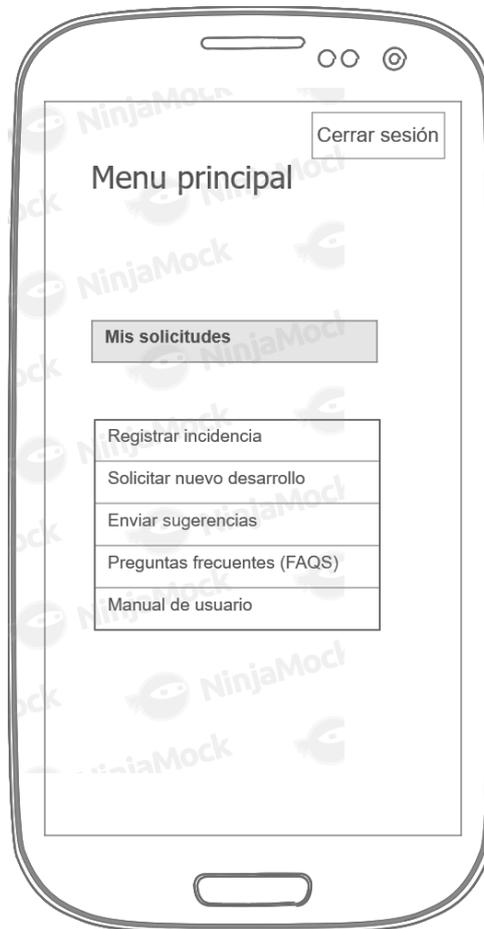
Empresa
▼ Empresa 1
Empresa 1
Empresa 2

Registrar Volver

(Figura 9.2) Prototipo pantalla de registro de usuario

En la pantalla de registro de usuario, es posible dar de alta un nuevo usuario en el sistema, introduciendo “Nombre”, “Email”, “Usuario” y “Contraseña”, además habrá que seleccionar la empresa a la que pertenece de un desplegable (que debe estar dada de alta en el sistema previamente) y pulsar el botón “Registrar”. El usuario además podrá volver a la pantalla de inicio pulsando el botón “Volver”.

9.2.3 Menú principal



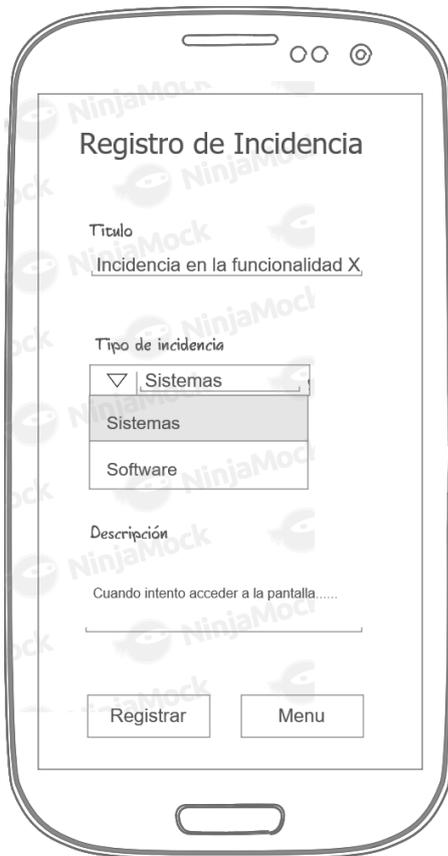
(Figura 9.3) Prototipo Menú principal

En la pantalla de menú principal, dispondremos de las opciones más importantes de la aplicación sobre las que se deberá pulsar para acceder a ellas.

Destaca la opción de “**Mis solicitudes**”, que permite visualizar las solicitudes realizadas. Además, tenemos las siguientes opciones:

- **Registrar incidencia:** Esta opción permite dar de alta incidencias tanto de Software como de Sistemas.
- **Solicitar nuevo desarrollo:** Permitirá solicitar un nuevo desarrollo, comenzando por el presupuesto.
- **Enviar sugerencias:** Servirá de buzón para que los clientes aporten sugerencias y mejoras.
- **Preguntas frecuentes (FAQS):** Permitirá la visualización de las preguntas Frecuentes sobre la aplicación Software propietario.
- **Manual de usuario:** Esta opción contendrá el manual en formato PDF disponible para ser descargado.

9.2.4 Pantalla de registro de incidencia



(Figura 9.4) Prototipo pantalla de registro de incidencia

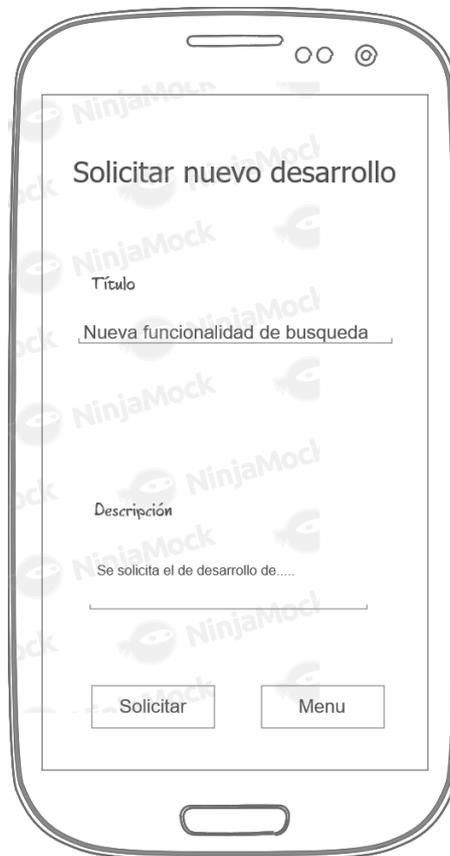


(Figura 9.5) Prototipo pantalla de escáner

En la pantalla de registro de incidencia el usuario podrá dar de alta una incidencia, para ello deberá escribir un título elegir el tipo de incidencia Sistemas (Bloqueo de Servidor, maquina, etc.) o Software (Funcionamiento erróneo en el Software), escribir una descripción de la misma y pulsar el botón "Registrar". El usuario además podrá volver a la pantalla de menú principal pulsando el botón "Menu".

Si el usuario selecciona la opción sistemas, aparecerá la pantalla de escáner (pantalla de la derecha) para escanear el código BIDI [\[12\]](#) de la máquina que tiene el problema.

9.2.5 Pantalla de solicitud de desarrollo



(Figura 9.6) Prototipo pantalla de solicitud de desarrollo

La pantalla de solicitud de nuevo desarrollo, permitirá solicitar un nuevo desarrollo de Software, para ello el primer paso, será realizar la solicitud, escribiendo un "Título" y una "Descripción" y pulsar sobre el botón "Solicitar". Esto generará una nueva solicitud en estado pendiente de *"presupuesto solicitado"*, y pasará a poder visualizarse en la pantalla de solicitudes realizadas. El usuario además podrá volver a la pantalla de menú principal pulsando el botón "Menu".

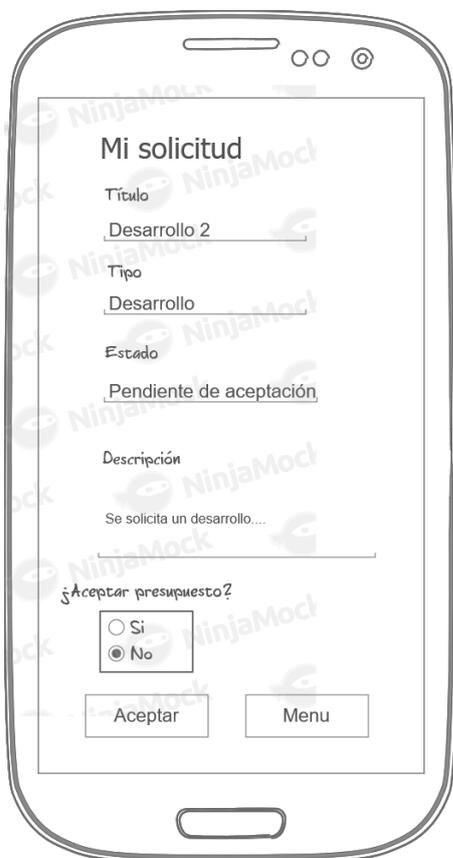
9.2.6 Pantalla de solicitudes realizadas



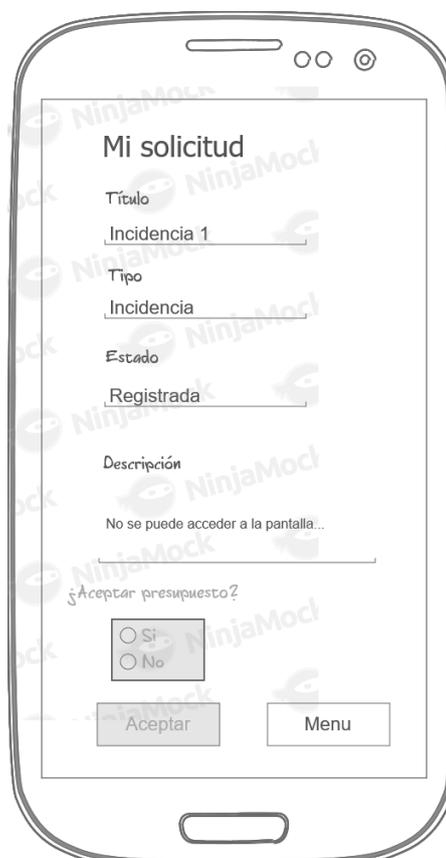
(Figura 9.7) Prototipo pantalla de solicitudes realizadas

La pantalla de “Mis solicitudes” permitirá visualizar las solicitudes realizadas por el usuario, diferenciadas por tipo (incidencia o desarrollo) y mostrará su estado actual. Para obtener más detalle sobre las solicitudes se deberá pulsar sobre las mismas. El usuario además podrá volver a la pantalla de menú principal pulsando el botón “Menu”.

9.2.7 Pantalla de detalle de solicitud



(Figura 9.8) Prototipo pantalla de detalle de solicitud (Desarrollo pendiente de aceptación)



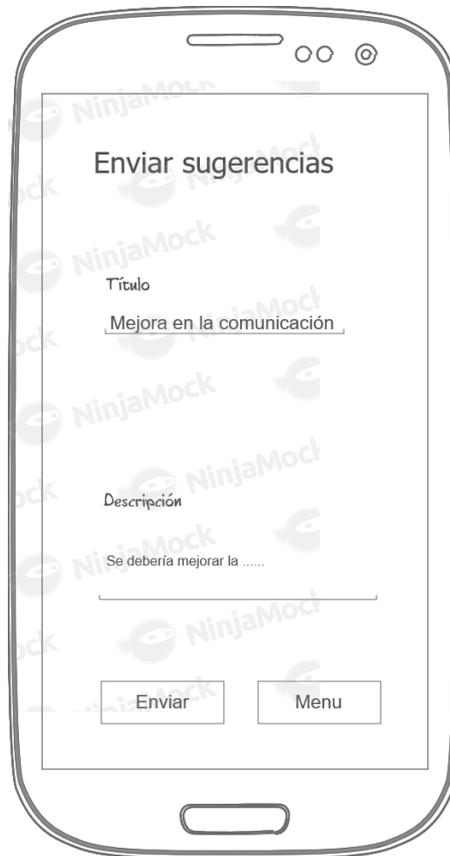
(Figura 9.9) Prototipo pantalla de detalle de solicitud (Incidencia)

A la pantalla de detalle de solicitud se accede tras pulsar sobre una solicitud de la pantalla de "Mis solicitudes". En ella se mostrará el título de la solicitud, el tipo (Software o Sistemas) y su descripción. En caso de ser un desarrollo en estado de "presupuesto solicitado" el botón de aceptar estará habilitado para permitir aceptar el presupuesto y además se mostrará un campo de presupuesto. El usuario además podrá volver a la pantalla de menú principal pulsando el botón "Menu".

A la izquierda se muestra un desarrollo en estado "pendiente de aceptación" con el botón de "Aceptar" habilitado. Este será el único estado en el que se podrá pulsar aceptar.

A la derecha se muestra una solicitud de tipo incidencia, mostrando el botón "Aceptar" deshabilitado.

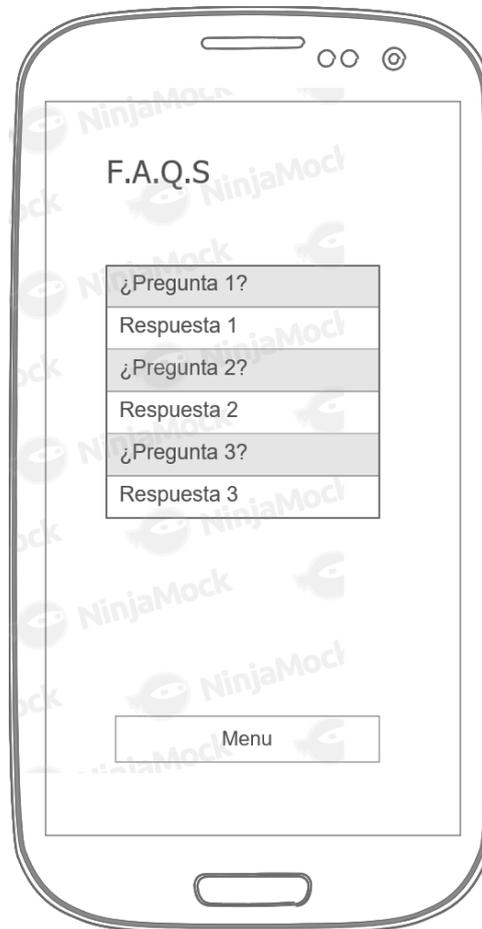
9.2.8 Pantalla de envío de sugerencias



(Figura 9.10) Prototipo pantalla de envío de sugerencias

La pantalla de envío de sugerencias actuará de buzón para que la empresa de desarrollo reciba sugerencias por parte de los clientes. Para ello se escribirá un título, una descripción y se pulsará el botón de "Enviar". El usuario además podrá volver a la pantalla de menú principal pulsando el botón "Menu".

9.2.9 Pantalla de preguntas frecuentes



(Figura 9.11) Prototipo pantalla de preguntas frecuentes

La pantalla de preguntas frecuentes (FAQS), mostrará un listado de preguntas frecuentes con sus respuestas sobre el Software propietario. El usuario además podrá volver a la pantalla de menú principal pulsando el botón "Menu".

10- Evaluación

El objetivo de la última fase es la planificación de la evaluación del prototipo. Dado que es una fase iterativa, que evalúa los diseños y corrige los errores a medida que transcurre el proyecto, nos limitaremos a indicar como se planteará la evaluación de los usuarios fase que se cerrará con la finalización del proyecto.

Las preguntas del Test de usuarios se dividirán en Pre-test y Post-test (preguntas relativas a las tareas), coincidiendo el Pre-test con las preguntas de las entrevistas realizadas en la fase de análisis.

10.1 Preguntas para el test de usuarios.

Pre-test

Datos del perfil

- Edad
- Sexo
- Nivel de estudios
- Tipo de negocio
- Puesto en la empresa
- Tipo de discapacidad o limitación
- Frecuencia con que utiliza aplicaciones móviles

Cuestionario de filtrado

No se realizará cuestionario de filtrado y selección de usuarios, puesto que la aplicación será desarrollada a medida, para un uso y entorno concreto, siendo acotados los usuarios potenciales.

10.2 Tareas a realizar por parte de los usuarios.

Las tareas que los usuarios deberán realizar, para completar la evaluación serán las siguientes:

- **Tarea 1:** Realizar un flujo completo de incidencia de tipo Software.
- **Tarea 2:** Realizar un flujo completo de incidencia de tipo Sistemas.
- **Tarea 3:** Realizar un flujo completo de solicitud de nuevo desarrollo: Solicitar desarrollo, enviar presupuesto, aceptar presupuesto, realizar desarrollo y finalizar desarrollo.
- **Tarea 4:** Enviar una sugerencia a la empresa desarrolladora.
- **Tarea 5:** Insertar contenido de preguntas frecuentes (FAQS) en el gestor y visualizarlo.
- **Tarea 6:** Descargar manual.

10.3 Preguntas referentes a las tareas.

Con el objeto de medir el grado de satisfacción del usuario respecto a la aplicación, se realizarán una serie de preguntas sobre el mismo. Este cuestionario se llamará Post-Test y será como se indica a continuación:

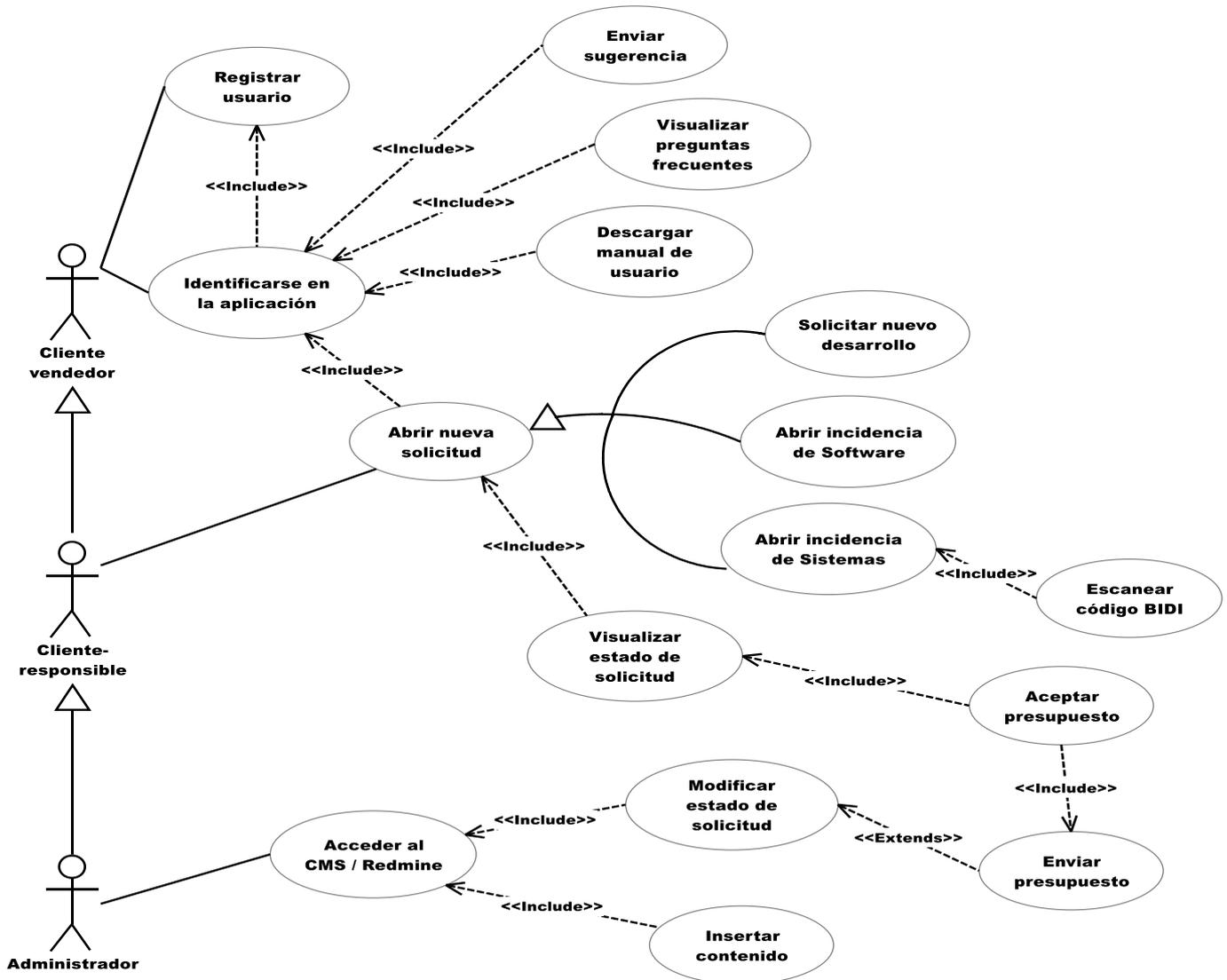
Post-Test

A continuación, se exponen las preguntas a realizar en el post-Test:

- ¿Cuál es su opinión con respecto a la navegabilidad de la aplicación móvil?
- ¿Piensa que el aprendizaje del funcionamiento de la aplicación es sencillo?
- ¿Considera atractiva visualmente la aplicación?
- ¿Cuál es su percepción con respecto a los botones para interactuar con las pantallas, en cuanto a su ubicación, tamaño e icono del botón?
- En caso de que considere que tuvo inconvenientes para utilizar alguna función de la aplicación. ¿Qué mejoras incluiría?
- ¿Ha sentido que controlaba todo el proceso de respuesta de las tareas asignadas?
- ¿Se ha sentido cómodo realizando las tareas?
- Valore la dificultad de cada tarea en esta escala: De 1 (Muy difícil) al 5 (Muy fácil).

11- Definición de los casos de uso

A continuación, se muestra el diagrama general de casos de uso de la aplicación móvil:



(Diagrama 11.1) Diagrama de casos de uso

11.1 Detalle de los casos de uso

CU001: Registrar usuario

Actores
Cliente vendedor, Cliente responsable
Prioridad
Alta
Descripción
El caso de uso consiste en el registro de un usuario en el sistema, necesario para el uso de la aplicación.
Precondiciones
Precondición 1: La empresa del usuario a registrar tiene que estar dada de alta en el sistema (Redmine).
Postcondiciones
Se registra el usuario en la aplicación.

Flujo principal
<p>Paso 1: Abrir la aplicación.</p> <p>Paso 2: En la pantalla de inicio, pulsar el botón “Registro”.</p> <p>Paso 3: Rellenar los datos de registro de usuario (Nombre, Email, Usuario, Contraseña, Empresa a seleccionar del listado de empresas).</p> <p>Paso 4: Pulsar el botón “Registrar” y confirmar.</p>
Flujo alternativo
<p>Escenario alternativo1:</p> <p>Paso 4: Pulsar el botón “Registrar”.</p> <p>Paso 5: Aparece un mensaje indicando que ya existe el usuario.</p> <p>Paso 6: Cambiar el dato y repetir la operación.</p>

CU002: Registrar incidencia de Software

Actores
Cliente vendedor, Cliente responsable
Prioridad
Alta
Descripción
El caso de uso consiste en el registro de una incidencia de tipo Software para que sea tratada por la empresa desarrolladora.
Precondiciones
Precondición 1: El usuario debe estar dado de alta en el sistema (CU 001) .
Postcondiciones
Se registra la incidencia de Software en la base de datos del sistema, en estado “Registrada”.
Flujo principal
<p>Paso 1: Abrir la aplicación.</p> <p>Paso 2: En la pantalla de inicio, escribir usuario y contraseña y pulsar “Login”.</p> <p>Paso 3: En el menú principal, pulsar sobre la opción “Registrar nueva incidencia”.</p> <p>Paso 4: Rellenar los datos de la pantalla de Registro de incidencia, Título, Descripción de la incidencia y seleccionar Tipo de incidencia “Software”.</p> <p>Paso 5: Pulsar el botón “Registrar” y confirmar envío.</p>

CU003: Registrar incidencia de Sistemas

Actores
Cliente vendedor, Cliente responsable
Prioridad
Alta
Descripción
Consiste en el registro de una incidencia tipo Software para su resolución por la empresa desarrolladora.
Precondiciones
Precondición 1: El usuario debe estar dado de alta en el sistema (CU 001) .
Postcondiciones
Se registra la incidencia de Sistemas en la base de datos del sistema, en estado “Registrada”.
Flujo principal
<p>Paso 1: Abrir la aplicación.</p> <p>Paso 2: En la pantalla de inicio, escribir usuario y contraseña y pulsar “Login”.</p> <p>Paso 3: En el menú principal, pulsar sobre la opción “Registrar nueva incidencia”.</p> <p>Paso 4: Rellenar los datos de la pantalla de Registro de incidencia, Título, Descripción de la incidencia y seleccionar Tipo de incidencia “Sistemas”.</p> <p>Paso 4: Pulsar el botón “Registrar”.</p> <p>Paso 5: Escanear código BIDI del dispositivo objeto de la incidencia.</p> <p>Paso 6: Confirmar el envío.</p>
Flujo alternativo
<p>Paso 5: Aparece la pantalla de Escáner del código BIDI del dispositivo objeto de la incidencia.</p> <p>Paso 6: Pulsar “Cancelar”.</p>

CU004: Solicitar nuevo desarrollo (Solicitar presupuesto)

Actores
Cliente responsable
Prioridad
Alta
Descripción
El caso de uso consiste en la solicitud de un nuevo desarrollo de Software del cliente a la empresa, que quedará en estado pendiente de presupuesto.
Precondiciones
Precondición 1: El usuario debe estar dado de alta en el sistema (CU 001).
Postcondiciones
Se registra la solicitud de presupuesto en el sistema en estado "Presupuesto solicitado".
Flujo principal
Paso 1: Abrir la aplicación. Paso 2: En la pantalla de inicio, escribir usuario y contraseña y pulsar "Login". Paso 3: En el menú principal, pulsar sobre la opción "Solicitar nuevo desarrollo". Paso 4: Rellenar formulario de nueva solicitud de desarrollo. Paso 5: Pulsar el botón "Solicitar" y confirmar envío.

CU005: Enviar presupuesto

Actores
Administrador
Prioridad
Alta
Descripción
El caso de uso consiste en el envío por parte del usuario Administrador del presupuesto de una solicitud previamente realizada por el cliente.
Precondiciones
Precondición 1: El usuario debe estar dado de alta en el sistema (CU 001). Precondición 2: El cliente debe haber solicitado previamente un presupuesto (CU004).
Postcondiciones
El presupuesto es registrado en el sistema.
Flujo principal
Paso 1: Abrir la aplicación Redmine y visualizar las solicitudes realizadas. Paso 2: Seleccionar la solicitud de presupuesto. Paso 3: Rellenar los datos del presupuesto (descripción y horas facturadas). Paso 4: Cambiar estado por "Pendiente de aceptación". Paso 5: Pulsar el botón "Guardar".

CU006: Solicitar nuevo desarrollo (Aceptar presupuesto)

Actores
Cliente responsable
Prioridad
Alta
Descripción
El caso de uso consiste en la aceptación por parte del cliente del presupuesto enviado por la empresa.
Precondiciones
Precondición 1: El usuario debe estar dado de alta en el sistema (CU 001). Precondición 2: El cliente debe haber solicitado previamente un presupuesto (CU 004). Precondición 3: El presupuesto tiene que haber sido enviado por parte de la empresa (CU 005).
Postcondiciones
El presupuesto queda aceptado en el sistema y pasa a estado "Presupuesto aceptado".
Flujo principal
Paso 1: Abrir la aplicación.

Paso 2: En la pantalla de inicio, escribir usuario y contraseña y pulsar "Login".
Paso 3: En el menú principal, pulsar sobre la opción "Mis solicitudes".
Paso 4: Visualizar las solicitudes enviadas, filtrar por la solicitud "pendiente de aceptación" y pulsar sobre la misma.
Paso 5: Marcar Si en la opción ¿Aceptar presupuesto?
Paso 6: Pulsar el botón "Aceptar" y confirmar envío.
Flujo alternativo
Escenario alternativo1:
Paso 5: Marcar No en la opción ¿Aceptar presupuesto?
Paso 6: Rellenar motivos de rechazo.
Paso 7: Pulsar el botón "Aceptar".

CU007: Enviar sugerencias

Actores
Cliente vendedor, Cliente responsable
Prioridad
Media
Descripción
Este caso, consiste en el envío de sugerencias respecto a la mejora del Software por parte del cliente.
Precondiciones
Precondición 1: El usuario debe estar dado de alta en el sistema (CU 001).
Postcondiciones
Se registra la sugerencia en el sistema.
Flujo principal
Paso 1: Abrir la aplicación.
Paso 2: En la pantalla de inicio, escribir usuario y contraseña y pulsar "Login".
Paso 3: En el menú principal, pulsar sobre la opción "Enviar sugerencias".
Paso 4: Rellenar formulario de sugerencias.
Paso 5: Pulsar el botón "Enviar" y confirmar envío.

CU008: Visualizar preguntas frecuentes (FAQS).

Actores
Cliente vendedor, Cliente responsable
Prioridad
Baja
Descripción
El caso de uso consiste en la visualización por parte de los clientes, de preguntas frecuentes relativas al Software propietario a modo de ayuda para el mismo.
Precondiciones
Precondición 1: El usuario debe estar dado de alta en el sistema (CU 001).
Precondición 2: Para visualizar las preguntas frecuentes, estas deberán haber sido dadas de altas en el sistema (CU 010).
Postcondiciones
Se visualizan las preguntas frecuentes.
Flujo principal
Paso 1: Abrir la aplicación.
Paso 2: En la pantalla de inicio, escribir usuario y contraseña y pulsar "Login".
Paso 3: En el menú principal, pulsar sobre la opción "Preguntas frecuentes (FAQS)".

CU009: Descargar manual.

Actores
Cliente vendedor, Cliente responsable
Prioridad
Baja
Descripción

Consiste en la descarga del manual de usuario del Software propietario por parte de los clientes.
Precondiciones
Precondición 1: El usuario debe estar dado de alta en el sistema (CU 001) . Precondición 2: El manual debe haber sido dado de alta en el sistema (CU 010) .
Postcondiciones
Se descargará el manual de usuario en formato <i>pdf</i> para su posterior visualización.
Flujo principal
Paso 1: Abrir la aplicación. Paso 2: En la pantalla de inicio, escribir usuario y contraseña y pulsar "Login". Paso 3: Pulsar sobre la opción de "Manual de usuario".

CU010: Mantenimiento e inserción de contenido

Actores
Administrador
Prioridad
Alta
Descripción
Este caso de uso es propio de la inserción de contenido por lo que se obviarán los pasos, que en el caso que nos ocupa serán funcionalidades de Redmine.
Postcondiciones
El presupuesto es registrado en el sistema.
Flujo principal
Paso 1: Abrir la aplicación Redmine. Paso 2: Insertar contenido en Redmine (Manual, Preguntas frecuentes, etc.). Paso 3: Registrar el contenido.

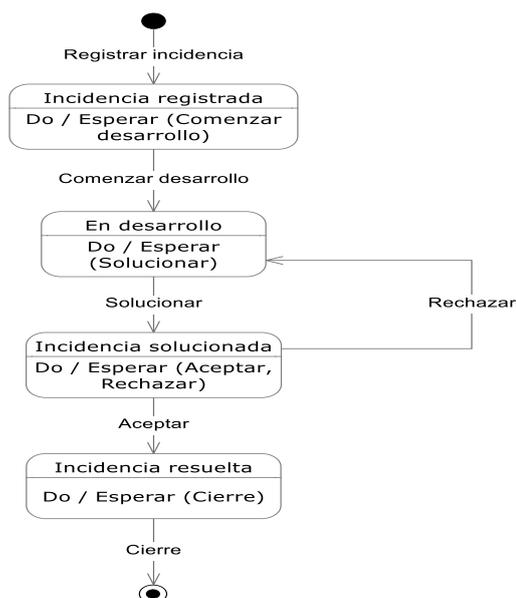
11.2 Diagrama de transición de estados

Para facilitar la comprensión del funcionamiento de la aplicación, se han realizado dos diagramas, detallando las transiciones de los estados posibles que pueden poseer las Incidencias y las Solicitudes.

11.2.1 Incidencias

Estados posibles:

- Registrada
- En desarrollo
- Solucionada
- Resuelta

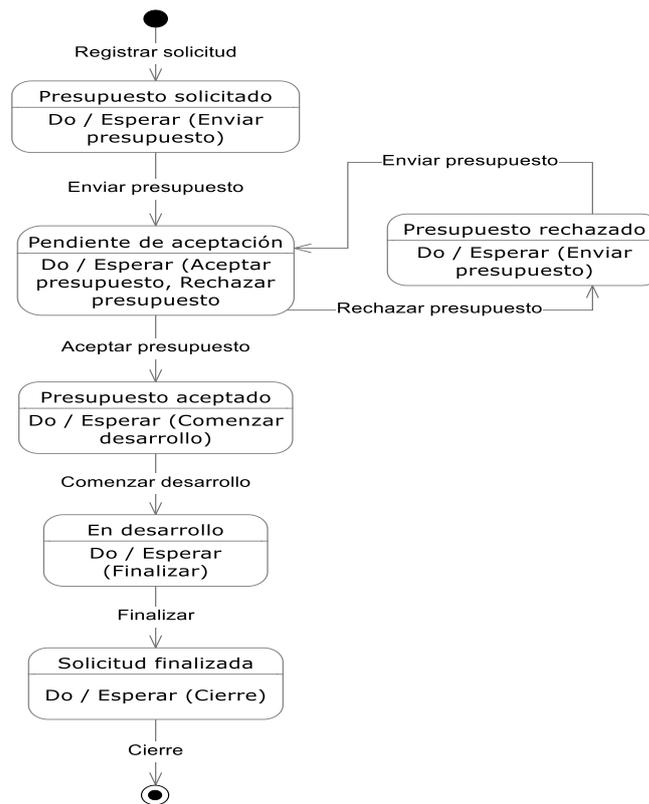


(Diagrama 11.2) Diagrama de transición de estados (incidencias)

11.2.2 Solicitud de desarrollo

Estados posibles:

- Presupuesto solicitado
- Pendiente de aceptación
- Presupuesto aceptado
- Presupuesto rechazado
- En desarrollo
- Finalizado



(Diagrama 11.3) Diagrama de transición de estados (solicitudes de desarrollo)

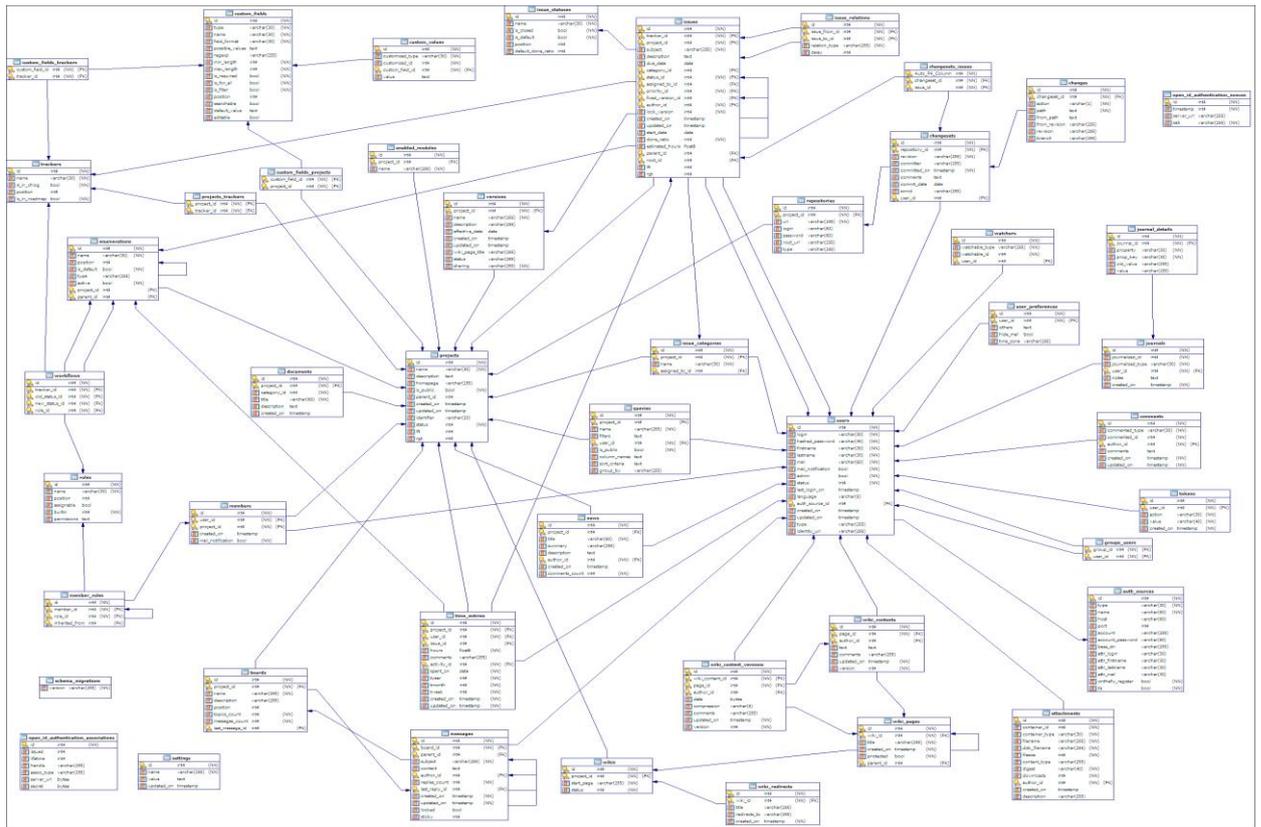
12- Diseño de la arquitectura

12.1 Diagrama de Bases de datos

Para realizar la presente aplicación se hará uso de la base de datos de Redmine. Aprovechando su estructura y comunicándonos mediante la *API* que proporciona, se realizarán las transacciones entre la aplicación móvil y el Sistema Redmine.

La inserción de información se basará en el contenido que almacena Redmine a modo de proyectos, así como la validación de los usuarios.

Modelo de Redmine [9]



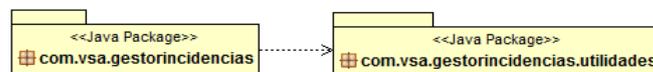
(Diagrama 12.1) Modelo de datos de Redmine

12.2. Diagrama de clases

A continuación, se muestran los diagramas de clases de la aplicación. Las clases serán detalladas en el punto [13.2 Estructura de la aplicación](#).

Paquetes

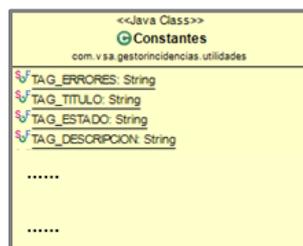
El paquete principal “com.vsa.gestorincidencias” contendrá al de utilidades “com.vsa.gestorincidencias.utilidades”:



(Diagrama 12.2) Diagrama de clases (paquetes). [13]

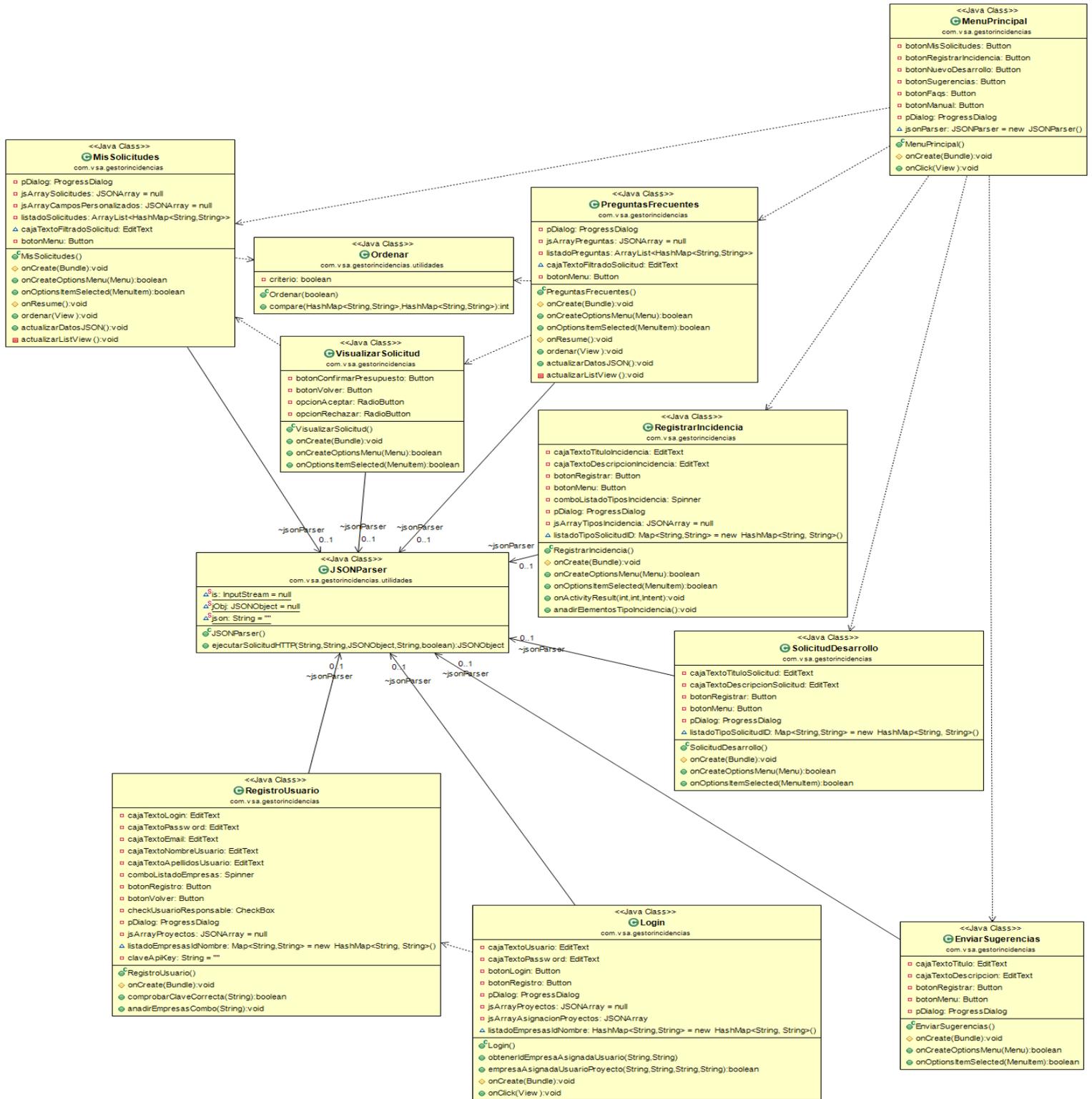
Clases

Por simplificar, se ha obviado la clase “constantes.java” que se relaciona con todas las clases.



(Diagrama 12.3) Diagrama de clases (Constantes). [13]

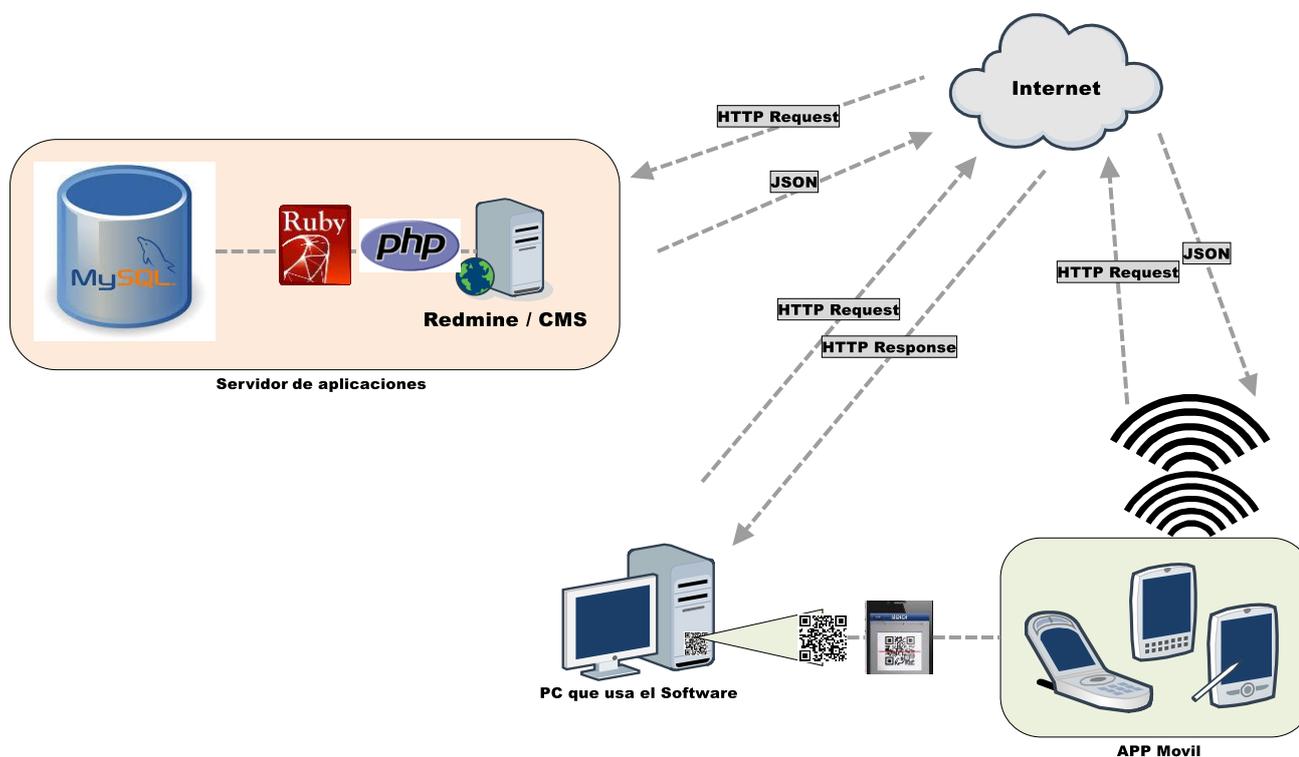
A continuación, se muestran las relaciones entre las clases de la aplicación:



(Diagrama 12.4) Diagrama de clases. [13]

12.3 Arquitectura general del sistema

A continuación, se describe el diagrama de arquitectura del sistema a alto nivel.



(Diagrama 12.5) Arquitectura general del sistema. [14]

En el diagrama de arquitectura se muestra la comunicación de los distintos elementos arquitectónicos de alto nivel.

Se diferencian claramente los siguientes elementos:

- **Servidor de aplicaciones y base de datos:** Se usará una solución Bitnami para la instalación de Redmine, que dispondrá de una Base de datos MySQL.
- **Aplicación móvil:** La aplicación móvil se desarrollará en Android nativo, y se comunicará con Redmine mediante una *API Rest* que proporciona esta última.
- **Comunicaciones:** Las comunicaciones entre Redmine y la aplicación móvil, se realizarán mediante la notación de intercambio de datos JSON. Se muestra en el diagrama la posibilidad de lectura de código *BIDI* de las máquinas físicas, mediante los dispositivos móviles.

SECCIÓN IV – IMPLEMENTACIÓN

13- Desarrollo

La fase de desarrollo, ha consistido en la construcción de la aplicación móvil y su integración con la herramienta Redmine.

13.1 Entorno tecnológico

Desde el punto de vista tecnológico, el proyecto se ha dividido en dos partes, por un lado, el desarrollo de la **Aplicación móvil** y por otro, la gestión de incidencias e inserción de contenido mediante la herramienta **Redmine**, que proporciona una *API Rest*, mediante la cual se realizará la integración de ambas partes.

13.1.1 Aplicación móvil

Las características técnicas de la aplicación móvil son las siguientes:

Lenguaje de programación:

El proyecto está basado en una aplicación nativa **Android** en su versión 5.0.0. Por lo tanto, el lenguaje de programación será **Java**.

Entorno de desarrollo:

El entorno de desarrollo ha sido **Eclipse Luna** [15], siendo necesaria la instalación de los *plugins* ADT y SDK [16] para el desarrollo de aplicaciones Android.

Librerías requeridas:

- **Escáner de códigos BIDI:** Zxing (“CaptureActivity”).
- **Compatibilidad:** “appCompat v7”.

Entorno de ejecución y pruebas:

- **Pruebas funcionales:** Se han realizado pruebas funcionales mediante el emulador **GenyMotion** [17] (Este punto se explicará con más detalle en la fase de pruebas). Para el uso de este emulador es necesaria una aplicación de virtualización como **Virtual Box**.
- **Pruebas unitarias:** Para la ejecución de pruebas unitarias se ha utilizado la librería **JUnit4** de Java.

13.1.2 Gestión de incidencias con Redmine

Configuración y gestión del contenido (incidencias):

Para el sistema de incidencias, se ha usado la aplicación **Redmine 3.2.1** [18], que es una herramienta de código abierto, para la gestión de proyectos que incluye un sistema de seguimiento de incidencias y control de errores.

Entre otras funcionalidades incluidas se encuentran: un calendario de actividades, diagramas, wikis, foros, RSS, control de flujo de trabajo basado en roles, integración con correo electrónico, etc.

Aunque en el caso que nos ocupa no ha sido necesaria la instalación de aplicaciones de terceros, Redmine posee una gran capacidad de personalización mediante la instalación de *plugins* tanto gratuitos como de pago, disponibles en su comunidad de desarrollo y que pueden ser descargados del siguiente enlace:

- <http://www.redmine.org/plugins>

Servicios Web:

Para realizar la integración entre la aplicación móvil y Redmine, se ha hecho uso de la *API Rest* [8] que facilita esta última. Esta *API* proporciona acceso y operaciones *CRUD* básicas (crear, actualizar, eliminar). La *API* admite los formatos *XML* y *JSON*.

Entorno general Servidor de aplicaciones:

- **Base de datos:** MySQL Versión del servidor: 5.5.16; Versión del protocolo: 10.
- **Sistema operativo:** Windows 10.
- **Servidor de aplicaciones:** La aplicación Redmine, ha sido desplegada mediante el instalador de aplicaciones **Bitnami**, con su versión para el proyecto Redmine, mediante el siguiente entorno:
 - **Servido web:** Apache HTTP Server.
 - **Base de Datos:** MySQL.
 - **Lenguaje de programación:** El proyecto Redmine se encuentra implementado mediante el *framework* de Ruby "Ruby on Rails".
- **Navegadores utilizados:** Firefox 49.0, Google Chrome 53.0.2785.143 m.
- **Control de versiones:** Para el control de versiones, se usará BitBucket que es un sistema de control de versiones de basado en GIT.

13.1.3 Configuración del gestor de incidencias Redmine

Para el correcto funcionamiento de la aplicación, es necesario configurar convenientemente Redmine. Para ello se deben seguir los siguientes pasos:

1. Crear los siguientes tipos de peticiones ("Trackers"):

- **"Sugerencia":** Sugerencias enviadas por los clientes.
- **"Incidencia Software":** Incidencias de Software reportadas.
- **"Incidencia Sistemas":** Incidencias de Software reportadas.
- **"Desarrollo":** Solicitudes de desarrollo enviadas por los responsables del cliente.
- **"Preguntas Frecuentes":** Información sobre el Software propietario.

2. Crear los siguientes proyectos:

- **“Interno”**: Donde se almacenará la Wiki para alojar la documentación. Dentro de este proyecto se debe crear una wiki con el nombre **Wiki** de la siguiente manera dentro de la siguiente ruta, **“/wiki/Wiki”** y adjuntar en ella un fichero.
- **“Buzon”**: Donde se almacenarán todas las peticiones de tipo “Sugerencia” registradas mediante la aplicación móvil.

Diferenciaremos los proyectos internos, de los proyectos de empresa que serán accesibles por la aplicación móvil *.

* **IMPORTANTE**: Para que los proyectos empresa sean detectados por la aplicación, deberán ser creados con el identificador “empresa” para que aparezcan en el registro de usuario.

3. Crear los siguientes flujos de trabajo de los tipos creados en el primer paso (**“Issue Statuses”**): (**ver Flujo de trabajo de la PAC2**)

Incidencias (Sistemas y Software):

- “Registrada”
- “En desarrollo”
- “Solucionada”
- “Resuelta”

Desarrollo:

- “Presupuesto solicitado”
- “Pendiente de aceptación”
- “Presupuesto aceptado”
- “Presupuesto rechazado”
- “En desarrollo”
- “Finalizado”

4. Crear los siguientes roles *:

- **“Cliente”**: A parte de los permisos básicos de Redmine, el rol de “Cliente” deberá tener permisos de modificación de sus propias solicitudes, permiso necesario para enviar los datos de las incidencias de sistemas.
- **“Cliente responsable”**: Al igual que el rol de “Cliente” además de los permisos básicos, deberá tener permisos de modificación de sus solicitudes, necesario para aceptar o rechazar las solicitudes de presupuesto.

* **IMPORTANTE**: Los usuarios de la aplicación móvil solo pueden tener un Rol, pero Redmine da la posibilidad de disponer de varios Roles. El acceso a la aplicación móvil solo está preparada para usuarios creados mediante la misma con un único Rol.

5. Añadir los siguientes campos personalizados *:

- **“DatosIncidencia”**: En este campo se almacenarán los datos recogidos por el código BIDI y estará disponible únicamente para las solicitudes de tipo “Incidencia Sistemas”.
- **“Presupuesto”**: En este campo se almacenará el presupuesto enviado por el técnico al cliente. Únicamente disponible para las solicitudes de tipo “Desarrollo”.

* **IMPORTANTE**: Para poder utilizar los campos personalizados, estos deberán ser activados para sus respectivos tipos de solicitud mediante la configuración de Redmine con permisos de Administrador.

La estructura JSON de los campos personalizados será la siguiente:

JSON

```
{
  "issue": {
    "custom_fields": [{
      "id": 1,
      "name": "DatosIncidencia",
      "value": "prueba datos incidencia"
    }, {
      "id": 1,
      "name": "Presupuesto",
      "value": "prueba presupuesto"
    }]
  }
}
```

(Figura 13.1) Ejemplo de JSON de entrada para campos personalizados

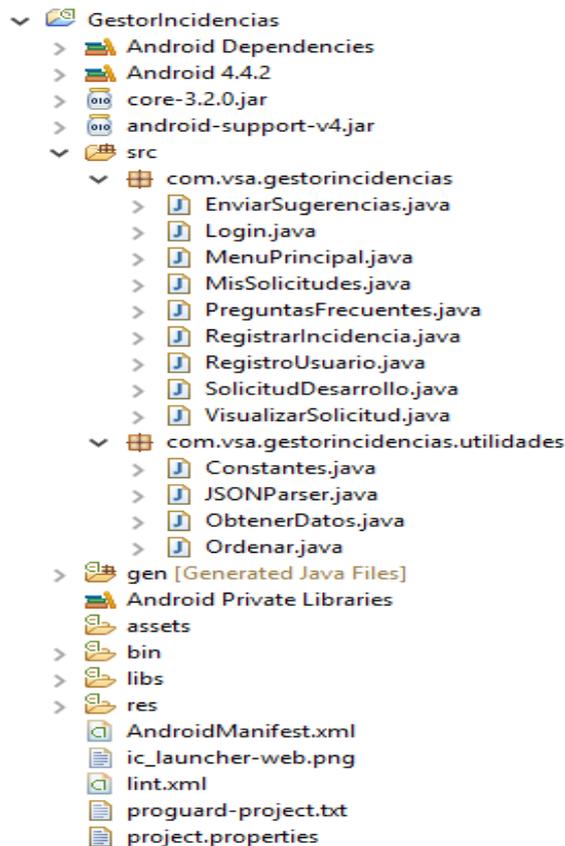
13.2 Estructura de la aplicación

La estructura de archivos del proyecto corresponde a una estructura típica Android. Destacamos las siguientes carpetas:

- **“src”**: Contiene las clases implementadas en java.
- **“res”**: Almacena los elementos visuales en formato XML.

13.2.1 Carpeta de código fuente “src”

La siguiente imagen, muestra el aspecto de la carpeta “src”:



(Figura 13.2) Estructura del proyecto (carpeta SRC)

Las clases se encuentran ubicadas en dos paquetes **“com.vsa.gestorincidencias”** y **“com.vsa.gestorincidencias.utilidades”**. A continuación, se detallan las clases:

Paquete principal (com.vsa.gestorincidencias):

Paquete que almacenará las clases *Activity* y el paquete de utilidades.

- **“EnviarSugerencias”**: Implementa la actividad de la pantalla de “EnviarSugerencias”. Su función es el envío de solicitudes de tipo “Sugerencia” al sistema, por parte del usuario cliente.
- **“Login”**: La clase “Login”, se corresponde con la actividad del mismo nombre. Su función es identificar en el sistema al usuario que introduce los datos, y permite la posibilidad de acceder a la actividad de “RegistroUsuario”, previa introducción de la clave correspondiente.
- **“MenuPrincipal”**: Esta clase implementa la pantalla de opciones del sistema, dando acceso a las opciones mediante la pulsación de los botones correspondientes. Para el acceso a la opción de Nuevo Desarrollo, el sistema comprueba que el rol del usuario sea diferente a “Cliente”. Si el usuario tiene rol Cliente no permitirá el acceso a esta opción.
- **“MisSolicitudes”**: Esta actividad muestra las solicitudes dadas de alta por el usuario identificado en el sistema, permitiendo filtrarlas y ordenarlas por orden alfabético. Además, mediante la pulsación sobre las mismas, se accede a la pantalla de detalle de la solicitud, haciendo uso de la actividad “VisualizarSolicitud”.
- **“PreguntasFrecuentes”**: Esta clase mostrará datos introducidos en Redmine, en concreto preguntas y respuestas relativas al Software propietario, para su visualización por parte del cliente.
- **“RegistrarIncidencia”**: La clase “RegistrarIncidencia” Se encargará del registro de solicitudes de tipo “Incidencia Sistemas” o “Incidencia Software”. Para el registro de Incidencias de sistemas, esta clase implementa la llamada a la librería “CaptureActivity” de Zxing para la utilización del escáner de códigos BIDI.
- **“RegistroUsuario”**: Implementa el registro de un nuevo usuario, seleccionando si es de tipo “Cliente responsable” (“check” pulsado) o “Cliente”. Para el acceso a la misma es necesario introducir una clave (*API Key*) con permisos de administrador. Esta clave se validará y en caso afirmativo, se accederá al registro, en caso contrario el sistema volverá a la actividad “Login”.
- **“SolicitudDesarrollo”**: Esta actividad implementa el registro en el sistema de solicitudes de tipo “Desarrollo”. Los datos de estas solicitudes podrán visualizarse en la pantalla de “VisualizarSolicitud”, donde será posible cambiar el estado de las mismas (Aceptar y rechazar presupuesto).
- **“VisualizarSolicitud”**: Esta actividad implementa la visualización de las solicitudes seleccionadas por el usuario en el listado de solicitudes de la pantalla “MisSolicitudes”. En esta actividad será posible aceptar o rechazar presupuesto (únicamente de solicitudes de tipo “Desarrollo” en estado “Pendiente de aceptación”).

Paquete de utilidades (“com.vsa.gestorincidencias.utilidades”):

Paquete incluido dentro del principal, que contendrá clases de utilidad para el proyecto.

- **“Constantes”**: La clase “Constantes”, contiene las *URL* necesarias para las llamadas a la *API Rest* y las etiquetas de los *JSON* tanto de entrada como de salida necesarias para la introducción o extracción de sus respectivos datos.
- **“JSONParser”**: Es una clase básica en el funcionamiento del proyecto. En ella se implementan las llamadas *HTTP* a los servicios Web mediante diferentes parámetros, devolviendo la respuesta del servidor, que se introducirá en formato *JSON* con la etiqueta respuesta.

Ejemplo de declaración de parámetros:

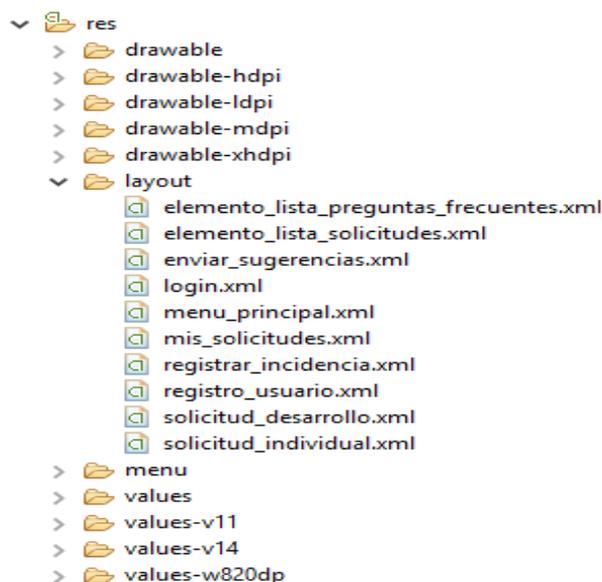
```
ejecutarSolicitudHttp(String url, String metodo, JSONObject jsonEntrada, String autorizacion, boolean admin)
```

Los parámetros de entrada son los siguientes:

- **“url”**: *URL* de llamada del Servicio.
- **“método”**: Puede ser “GET”, “PUT” o “POST”.
- **“jsonEntrada”**: *JSON* de entrada para la llamada (Solo para llamadas “PUT” o “POST”). En caso de llamada de tipo “GET”, este parámetro será “null”.

- **“autorización”**: Se enviará la clave de autorización, esta puede ser en formato BASE64 (ejemplo "Basic YWRtaW46MTIzNDU2Nzg=") o en formato API Key (ejemplo "226b571019ed42ed9d88053dd5ef33ab98bdb3b2") dependiendo del valor de "admin".
- **“admin”**: Variable lógica, si es verdadera la clase recibirá una clave API KEY, si es falsa una clave de identificación básica resultado de codificar usuario y contraseña en formato BASE64.
- **“ObtenerDatos”**: Esta clase del paquete utilidades, implementa diferentes métodos con el objetivo de hallar datos del sistema.
 - **“obtenerIdTipoSolicitud”**: Obtiene el tipo del Tipo de solicitud pasándole como parámetro el nombre de la solicitud.
 - **“obtenerIdProyecto”**: Devuelve el “ID” de un proyecto cuyo nombre se pasa como parámetro.
 - **“obtenerIdRol”**: Obtiene el ID de un Rol cuyo nombre se pasa como parámetro.
 - **“obtenerNombreRol”**: Devuelve el nombre del Rol que tiene un usuario en una empresa (proyecto), pasando como parámetro el “ID” del usuario y el “ID” del proyecto.
 - **“obtenerIdUsuario”**: Devuelve el “ID” de un usuario pasando como parámetro su autorización (usuario y contraseña codificada en BASE64).
 - **“obtenerRutaFicheroManualUsuario”**: Obtiene la ruta donde se encuentra ubicado el fichero de manual de usuario y su nombre, cumpliendo la siguiente estructura: "/wiki/Wiki".
 - **“obtenerApiKeyUsuario”**: Obtiene la API Key de un usuario cuya autorización se pasa como parámetro.
 - **“obtenerIdEstadoSolicitud”**: Devuelve el “ID” del estado de solicitud que se pasa como parámetro.
- **“Ordenar”**: La clase “Ordenar.java” recibe un listado de solicitudes de entrada y lo devuelve ordenado por orden alfabético ascendente o descendientemente, dependiendo del criterio previamente seleccionado.

13.2.2 Carpeta de recursos “res”



(Figura 13.3) Estructura del proyecto (carpeta RES)

- **“elemento_lista_preguntas_frecuentes”**: Disposición visual de los elementos individuales de la lista de preguntas frecuentes, mostrada en la pantalla “mis_solicitudes”.

- “**elemento_lista_solicitudes**”: Al igual que la anterior, establece la disposición visual de los elementos concretos del listado, pero en este caso de las solicitudes.
- “**enviar_sugerencias**”: Configura la pantalla de envío de sugerencias.
- “**login**”: Pantalla que facilita la identificación de los usuarios a la aplicación y el acceso a la pantalla “registro_usuario” para el registro de usuarios.
- “**menu_principal**”: Muestra seis botones para el acceso a las opciones de la aplicación.
- “**mis_solicitudes**”: Esta pantalla muestra el listado de solicitudes y permite filtrarlas y ordenarlas por orden alfabético. Además, visualiza el listado de preguntas frecuentes.
- “**registrar_incidencia**”: Pantalla que permite el registro de incidencias tanto de tipo Sistemas como Software en el sistema, mediante la actividad “RegistrarIncidencia”.
- “**registro_usuario**”: Esta pantalla permite el registro de nuevos usuarios en la aplicación haciendo uso de la actividad “RegistroUsuario”.
- “**solicitud_desarrollo**”: Pantalla que permite visualizar la solicitud de un nuevo desarrollo, mediante la actividad “SolicitudDesarrollo”.
- “**solicitud_individual**”: Visualización de los datos de una solicitud seleccionada en la pantalla de “mis_solicitudes”. Además, visualiza el listado de preguntas frecuentes, adaptando la disposición visual de los elementos.

13.4 Buenas prácticas de codificación

Durante la construcción del producto, se han seguido criterios básicos relativos a la calidad del código como son: modularidad, abstracción y el uso de patrones de diseño. Además, se ha seguido una metodología de Diseño Centrado en el usuario, procurando cumplir de manera ágil, con los criterios de usabilidad de interfaces de usuario y seguridad del producto.

Por otro lado, se han empleado las siguientes buenas prácticas en la codificación:

- Se ha **comentado** el código para su inteligibilidad.
- El código ha sido **identado** para facilitar su legibilidad.
- Utilización de **nomenclatura** de variables y métodos, nombrando tanto variables como procedimientos con palabras auto-explicativas para facilitar su comprensión.
- Se ha **refactorizado** el código utilizando el estilo “lowerCamelCase”.
- Se ha evitado crear procedimientos demasiado extensos, separándolos en varios.
- Se ha procurado la reutilización de código mediante clases de utilidades.

14- Pruebas

En esta fase, tomando como base el análisis de los casos de uso de la anterior etapa, se detallarán las pruebas realizadas para determinar el correcto funcionamiento de la aplicación.

14.1 Entorno de pruebas

Para la preparación y ejecución de las pruebas, se ha utilizado el siguiente entorno:

GenyMotion

Genymotion es un emulador de Android perteneciente a la compañía francesa de software del mismo nombre. Es un desarrollo de AndroVM basado en el software de virtualización VirtualBox. El terminal de simulación utilizado ha sido el siguiente:

- **Samsung Galaxy S6. Características técnicas:**
 - Android 5.0.0
 - Cámara (Simulada con *WebCam*). Necesaria para simular la lectura de códigos BIDI.

Testing Android

Mediante el entorno que proporciona Eclipse Luna, se han realizado pruebas unitarias con el Framework incluido como parte de la plataforma Android (JUnit4).

14.2 Tipos de pruebas realizadas

14.2.1 Pruebas unitarias

Para la realización de pruebas unitarias se ha procedido a utilizar el *framework* de pruebas *JUnit* de Java adaptado a Android, disponible en la versión de eclipse utilizada.

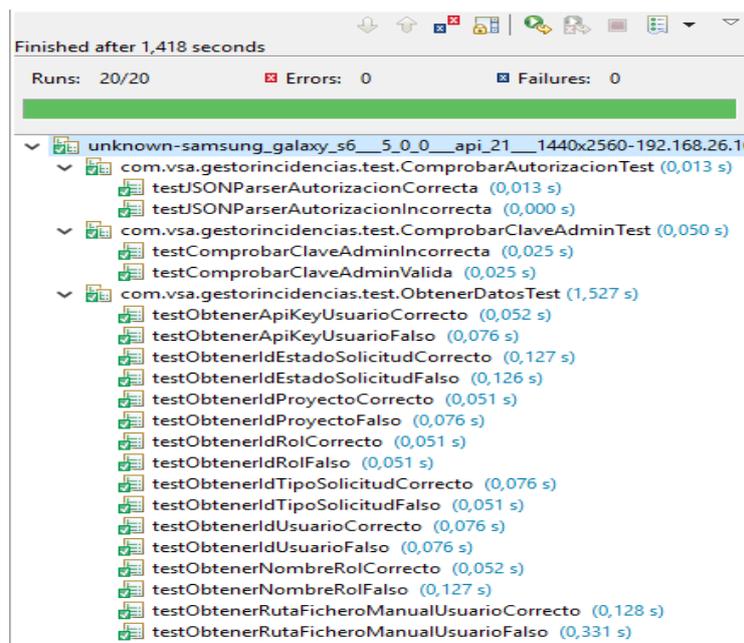
Para ello, se ha creado un proyecto de tipo “Android Test Project” independiente de la aplicación, con el mismo nombre que esta añadiéndole la palabra Test al final “GestorIncidenciasTest”.

Se han creado tres subclases por unidad funcional a probar:

- **“ComprobarAutorizacionTest”**: Clase para comprobar el correcto funcionamiento de las autorizaciones en el sistema Redmine.
- **“ComprobarClaveAdminTest”**: Clase para comprobar el correcto funcionamiento de las del método de comprobación de la clave de administrador.
- **“ObtenerDatosTest”**: En esta clase, se realizarán las comprobaciones de los métodos de la clase “ObtenerDatos” de la aplicación.

Resultados de las pruebas unitarias

A continuación, se muestran los resultados de las pruebas unitarias:



(Figura 14.1) Resultados de las pruebas unitarias

14.2.2 Pruebas funcionales

Se han realizado pruebas de integración sobre el simulador GenyMotion consistentes, por un lado, en probar las funcionalidades mínimas, imprescindibles para considerar que la iteración ha finalizado con éxito y que el resultado es un producto capaz de brindar unas funciones y una calidad mínima exigible.

Además, se han realizado pruebas de regresión, tras finalizar cada nueva implementación, para comprobar que tanto las nuevas funcionalidades y las anteriores continúan funcionando correctamente.

Definición de casos de prueba

ID	Caso de Prueba	Funcionalidad	Pasos del caso de prueba
CP-1.1	Registrar usuario con rol Cliente Responsable	Registro de usuario	<p>Paso 1: En la pantalla de inicio, pulsar el botón "Registro".</p> <p>Paso 2: Rellenar los datos obligatorios de registro de usuario (Nombre, Email, Usuario, Contraseña, Empresa a seleccionar del listado de empresas) y pulsar sobre el "check Responsable".</p> <p>Paso 3: Pulsar el botón "Registrar".</p> <p>Paso 4: Acceder a Redmine y comprobar que el usuario se ha creado correctamente.</p> <p>CAMINOS ALTERNATIVOS:</p> <ul style="list-style-type: none"> • Introducir un email con formato incorrecto. • No introducir alguno de los campos obligatorios. • Introducir una contraseña con menos de 8 dígitos.
CP-1.2	Registrar usuario con rol Cliente	Registro de usuario	<p>Paso 1: En la pantalla de inicio, pulsar el botón "Registro".</p> <p>Paso 2: Rellenar los datos obligatorios de registro de usuario (Nombre, Email, Usuario, Contraseña, Empresa a seleccionar del listado de empresas) y dejar el "check Responsable" sin confirmar.</p> <p>Paso 3: Pulsar el botón "Registrar".</p> <p>Paso 4: Acceder a Redmine y comprobar que el usuario se ha creado correctamente.</p> <p>Paso 5: Acceder a la aplicación y comprobar que el usuario accede correctamente.</p> <p>CAMINO ALTERNATIVO:</p> <p>Paso 1: En la pantalla de inicio, pulsar el botón "Registro".</p> <p>Paso 2: Introducir una clave incorrecta de acceso al registro de usuario.</p>
CP-2.1	Registrar incidencia de sistemas	Registro de incidencia	<p>Paso 1: En la pantalla de inicio, escribir usuario (con perfil de Cliente) y contraseña y pulsar "Login".</p> <p>Paso 2: En el menú principal, pulsar sobre "Registrar incidencia".</p> <p>Paso 3: Rellenar los datos obligatorios de la pantalla de Registro de incidencia, Título, Descripción de la incidencia y seleccionar Tipo de incidencia "Sistemas".</p> <p>Paso 4: Pulsar el botón "Registrar". Aparecerá la pantalla de escáner BIDI.</p> <p>Paso 5: Escanear código BIDI del dispositivo objeto de la incidencia.</p> <p>Paso 6: Confirmar el envío.</p> <p>Paso 7: Acceder a Redmine y comprobar que la incidencia se ha creado correctamente con sus datos correspondientes.</p> <p>CAMINO ALTERNATIVO:</p> <p>No introducir alguno de los campos obligatorios.</p>
CP-2.2	Registrar incidencia de software	Registro de incidencia	<p>Paso 1: En la pantalla de inicio, escribir usuario y contraseña y pulsar "Login".</p> <p>Paso 2: En el menú principal, pulsar sobre "Registrar incidencia".</p> <p>Paso 3: Rellenar los datos obligatorios de la pantalla de Registro de incidencia, Título, Descripción de la incidencia y seleccionar Tipo de incidencia "Software".</p> <p>Paso 4: Pulsar el botón "Registrar".</p> <p>Paso 5: Acceder a Redmine y comprobar que la incidencia se ha creado correctamente con sus datos correspondientes.</p>

CP-3.1	Solicitar nuevo desarrollo	Solicitud de desarrollo	<p>Paso 1: En la pantalla de inicio, escribir usuario y contraseña y pulsar "Login".</p> <p>Paso 2: En el menú principal, pulsar sobre la opción "Solicitar nuevo desarrollo".</p> <p>Paso 3: Rellenar formulario de nueva solicitud de desarrollo.</p> <p>Paso 4: Pulsar el botón "Solicitar" y confirmar envío.</p> <p>CAMINO ALTERNATIVO: Intentar acceder a la pantalla identificándose con un usuario que tenga rol de Cliente.</p>
CP-3.2	Enviar Presupuesto (Redmine)	Solicitud de desarrollo	<p>Paso 1: Abrir la aplicación Redmine y visualizar las solicitudes realizadas.</p> <p>Paso 2: Seleccionar la solicitud de presupuesto enviada en el caso anterior.</p> <p>Paso 3: Rellenar los datos del presupuesto (descripción y horas facturadas).</p> <p>Paso 4: Cambiar estado por "Pendiente de aceptación".</p> <p>Paso 5: Pulsar el botón "Guardar".</p>
CP-3.3	Aceptar presupuesto	Solicitud de desarrollo	<p>Paso 1: Abrir la aplicación móvil.</p> <p>Paso 2: Identificarse con el mismo usuario que creo la solicitud del caso anterior y pulsar "Login".</p> <p>Paso 3: En el menú principal, pulsar sobre la opción "Mis solicitudes".</p> <p>Paso 4: Visualizar las solicitudes enviadas, filtrar por la solicitud "pendiente de aceptación" y pulsar sobre la misma.</p> <p>Paso 5: Marcar "aceptar" en las opciones de presupuesto y pulsar el botón "Aceptar".</p> <p>Paso 6: Acceder a Redmine y comprobar que la solicitud ha cambiado de estado correctamente.</p>
CP-3.4	Rechazar presupuesto	Solicitud de desarrollo	<p>Paso 1: Abrir la aplicación Redmine y visualizar las solicitudes realizadas.</p> <p>Paso 2: Seleccionar la solicitud de presupuesto aceptada en el caso anterior.</p> <p>Paso 3: Volver a cambiar el estado a "Pendiente de aceptación".</p> <p>Paso 4: Pulsar el botón "Guardar".</p> <p>Paso 5: Abrir la aplicación móvil.</p> <p>Paso 6: Identificarse con el mismo usuario que creo la solicitud del caso anterior y pulsar "Login".</p> <p>Paso 7: En el menú principal, pulsar sobre la opción "Mis solicitudes".</p> <p>Paso 8: Visualizar las solicitudes enviadas, filtrar por la solicitud "pendiente de aceptación" y pulsar sobre la misma.</p> <p>Paso 9: Marcar "rechazar" en las opciones de presupuesto y pulsar el botón "Aceptar".</p> <p>Paso 10: Acceder a Redmine y comprobar que la solicitud ha cambiado de estado correctamente.</p>
CP-4.1	Enviar sugerencia	Envío de sugerencias	<p>Paso 1: En la pantalla de inicio, escribir usuario y contraseña y pulsar "Login".</p> <p>Paso 2: En el menú principal, pulsar sobre la opción "Enviar sugerencias".</p> <p>Paso 3: Rellenar formulario de sugerencias.</p> <p>Paso 4: Pulsar el botón "Enviar" y confirmar envío.</p> <p>Paso 5: Acceder a Redmine y comprobar que la sugerencia se ha creado correctamente.</p>
CP-5.1	Visualizar Preguntas frecuentes	Preguntas frecuentes	<p>Paso 1: Abrir la aplicación Redmine y registrar solicitudes de tipo "Preguntas frecuentes".</p> <p>Paso 2: Pulsar el botón "Guardar".</p> <p>Paso 3: Abrir la aplicación móvil.</p> <p>Paso 4: En la pantalla de inicio, escribir usuario y contraseña y pulsar "Login".</p> <p>Paso 5: En el menú principal, pulsar "Preguntas frecuentes (FAQS)".</p> <p>Paso 6: Comprobar que las Preguntas frecuentes se visualizan correctamente.</p>
CP-6.1	Descargar manual	Manual de usuario	<p>Paso 1: Abrir la aplicación Redmine y almacenar el documento de manual de usuario de tipo "Manual".</p> <p>Paso 2: Pulsar el botón "Guardar".</p> <p>Paso 3: Abrir la aplicación móvil.</p> <p>Paso 4: En la pantalla de inicio, escribir usuario y contraseña y pulsar</p>

			<p>“Login”.</p> <p>Paso 5: Pulsar sobre la opción de “Manual de usuario”.</p> <p>Paso 6: Comprobar que el manual de usuario se descarga correctamente.</p>
--	--	--	--

(Tabla 14.1) Definición de casos de prueba

Resultado de la ejecución de los casos de prueba

ID	Caso de Prueba	Pre-requisitos	Resultado esperado	Resultado Obtenido
CP-1.1	Registrar usuario con rol Cliente Responsable	Disponer de una <i>API KEY</i> con permisos de administrador	Se muestra un mensaje de confirmación. ¡Usuario creado con éxito!	OK
	CAMINOS ALTERNATIVOS <ul style="list-style-type: none"> Introducir un email con formato incorrecto. No introducir alguno de los campos obligatorios. Introducir una contraseña con menos de 8 dígitos. 	Disponer de una <i>API KEY</i> con permisos de administrador	Aparece un mensaje de error identificativo del mismo	OK
CP-1.2	Registrar usuario con rol Cliente	Disponer de una <i>API KEY</i> con permisos de administrador	Se muestra un mensaje de confirmación. ¡Usuario creado con éxito!	OK
	CAMINO ALTERNATIVO Introducir una clave incorrecta para acceder a la pantalla de registro.		Aparece un mensaje de error. ¡La clave introducida es incorrecta!	OK
CP-2.1	Registrar incidencia de sistemas	Disponer de un código BIDI para escanear con la cámara	Se muestra un mensaje de confirmación. ¡La incidencia de Sistemas se ha registrado correctamente!	OK
	CAMINO ALTERNATIVO No introducir alguno de los campos obligatorios.		Aparece un mensaje de error identificativo del mismo	OK
CP-2.2	Registrar incidencia de software		Se muestra un mensaje de confirmación. ¡La incidencia de Software se ha registrado correctamente!	OK
CP-3.1	Solicitar nuevo desarrollo	Identificarse con Rol Cliente responsable	Se muestra un mensaje de confirmación. ¡Solicitud de desarrollo registrada correctamente!	OK
	CAMINO ALTERNATIVO Intentar acceder a la pantalla identificándose con un usuario que tenga rol de Cliente.		Aparece un mensaje de error. ¡Su usuario no dispone de permisos para la funcionalidad solicitada!	OK
CP-3.2	Enviar presupuesto (Redmine)	Solicitar nuevo desarrollo (CP-3.1).	Se muestra un mensaje de confirmación. ¡Solicitud actualizada correctamente!	OK
CP-3.3	Aceptar presupuesto	Enviar presupuesto en Redmine. (CP-3.2).	Se muestra un mensaje de confirmación. ¡Solicitud actualizada correctamente!	OK
CP-3.4	Rechazar presupuesto	Enviar presupuesto en Redmine. (CP-3.2).	Se muestra un mensaje de confirmación. ¡Solicitud actualizada correctamente!	OK
CP-4.1	Enviar sugerencia		Se muestra un mensaje de confirmación. ¡Sugerencia enviada correctamente!	OK
CP-5.1	Visualizar preguntas frecuentes	Introducir contenido en Redmine	Se muestra correctamente el contenido de la pregunta frecuente	OK
CP-6.1	Descargar manual	Subir documento en Redmine en la ruta Wiki	Se descarga correctamente el documento	OK

(Tabla 14.2) Resultado de la ejecución de los casos de prueba

SECCIÓN V – CONCLUSIONES

15- Conclusiones y futuros trabajos

15.1 Resultado final

El resultado final del proyecto en términos metodológicos, ha sido el análisis, diseño e implementación de una aplicación móvil orientada a la gestión de incidencias de una empresa desarrolladora de Software propietario, que facilita la colaboración y comunicación entre los diferentes usuarios implicados (clientes y empleados de la empresa), mediante el envío de solicitudes a través de la aplicación. Además de permitir que los usuarios, puedan acceder y consultar los datos de sus solicitudes y realizar su trabajo desde cualquier parte, con el único requisito de disponer de un dispositivo móvil o *Tablet* conectado a Internet.

En términos generales, la experiencia en el desarrollo con Android y su integración con Redmine ha sido satisfactoria, encontrando en la documentación disponible en la red, una valiosa ayuda a la hora de resolver las dudas que han ido surgiendo a lo largo del proyecto.

Durante el desarrollo del proyecto, los requisitos iniciales definidos en las primeras fases, han ido adaptándose a nuevas necesidades detectadas. Como requisito inicial, se planteó la necesidad de gestión y tratamiento del contenido a mostrar en la aplicación móvil, para ello se pensó en el uso de un Sistema de gestión de contenidos como Drupal. Con el transcurso del proyecto y a medida que fue aumentando mi conocimiento sobre Redmine, el uso de un *CMS* fue descartado, concluyendo que, para este propósito, el uso de Redmine es suficiente.

Otro cambio importante, fue la inclusión del nuevo estado “Presupuesto rechazado” para las solicitudes de tipo “Desarrollo”, necesario para no perder trazabilidad de las solicitudes.

Respecto a las dificultades encontradas durante el desarrollo, estas resultaron principalmente del uso de la *API*, especialmente por la limitación de permisos. Estas restricciones provocaron el cambio de alguno de los requisitos, como por ejemplo la necesidad de disponer de una clave de administrador para poder realizar el registro de un usuario o para obtener la información relativa a los campos personalizados “*custom_fields*”.

Por otro lado, al contar con una *API* con un nivel de desarrollo heterogéneo para sus diferentes módulos, no todos los módulos se encuentran lo suficientemente testados y desarrollados. Por ejemplo, mediante la llamada “*projects/redmine/memberships.json*” no es posible obtener todas las asignaciones de los usuarios a los proyectos, siendo necesario introducir el id del proyecto, dentro de la misma llamada, de la siguiente manera:

- “*redmine/projects/id_proyecto/memberships.json*”.

Por último, resaltar la complejidad para la configuración e instalación de la librería necesaria para la lectura de códigos *BIDI*, *Zxing*. Para su uso integración fue necesaria la modificación de ciertas

partes del código por incompatibilidades (Cambio de todos los “switch case” que referenciaban a recursos *Layout* por “If-else”). La librería está disponible para su descarga en el siguiente enlace:

- <https://github.com/zxing/zxing>.

Tras vencer las dificultades comentadas, a lo largo del desarrollo del Proyecto, se han podido comprobar las enormes posibilidades de Android y Redmine, su potencia y modularidad, así como la relativa sencillez para desarrollar una aplicación compleja y real como la realizada para este trabajo.

Respecto a la planificación, se han podido ejecutar todas las fases de trabajo en el tiempo previsto inicialmente, cumpliéndose los plazos establecidos en la planificación, que indicaban la finalización de la Fase de implementación el día 14 de diciembre, la finalización del documento de memoria para la entrega definitiva el 11 de enero, y la preparación de la defensa ante el tribunal, para el 27 del mismo mes. No existiendo desviaciones ni riesgos destacables.

Cabe resaltar la utilidad de las fases de planificación, análisis y diseño realizados previamente, tanto el diseño técnico de la aplicación como las áreas de trabajo centradas en el diseño centrado en el usuario (DCU), ayudando a conocer y comprender las necesidades, limitaciones, comportamiento y características del usuario para que el producto responda a las necesidades reales de los usuarios finales.

15.2 Logros alcanzados y conocimientos adquiridos

Además de profundizar en el estudio y evaluar la capacidad y potencial del Sistema operativo Android, se han consolidado conocimientos previamente adquiridos de programación en Java, *API* propias de Google y Servicios Web utilizados en la integración con Redmine. Además, desde un punto de vista personal, todos estos conocimientos complementan a los adquiridos en el estudio del Master.

Al comenzar el proyecto, mis conocimientos de Android eran medios y fue relativamente costoso comenzar a darle forma a las implementaciones, pero con ayuda de la abundante información disponible en internet se consiguió avanzar. También es cierto, que en ocasiones debido al desconocimiento inicial de la integración con Redmine los desarrollos y modificaciones, podrían haberse realizado de manera más sencilla y directa.

Me gustaría resaltar el uso de librerías de terceros como Zxing, consiguiendo con ello uno de los principales objetivos intrínsecos al Software libre, que es la integración y modificación de código libre en un proyecto propio.

Con la elaboración de este trabajo, ha sido posible emprender un proyecto de desarrollo de una aplicación móvil que ha abarcado todas sus etapas, desde la captura de requisitos hasta la puesta en marcha de la aplicación.

Por último, quería destacar la fluida comunicación con los responsables de la asignatura, ayudando sus recomendaciones a obtener excelentes resultados relativos a la experiencia de usuario. Así mismo en la realización de este trabajo he podido poner en práctica competencias de la Ingeniería del Software adquiridas profesionalmente, relativas a la calidad y pruebas de Software, conocimientos que desde mi humilde opinión, han servido para conseguir un producto de mayor calidad final.

15.3 Posibles trabajos futuros

La facilidad de uso de Android y Redmine y el hecho de ser código abierto, convierten este binomio en un producto muy versátil a la hora de realizar mejoras e implementaciones.

El siguiente paso en la evolución natural del producto, sería la mejora de las funcionalidades existentes y la ampliación de la aplicación por medio de nuevas funcionalidades como, podrían ser:

- **Chat:** Implementación de un sistema de comunicación de los clientes con los técnicos de la empresa
- **Integración con aplicación de control remoto:** Podría desarrollarse una integración con una aplicación de control remoto para las máquinas como por ejemplo TeamViewer.
- **Geolocalización:** Aplicación de la geolocalización para mejorar y ampliar la información enviada por los clientes a los técnicos.
- **Integración con redes sociales:** Integración mediante enlaces a diferentes redes sociales como Twitter o Facebook, para mejorar la visibilidad comercial de la aplicación.

Respecto a la aplicación de la metodología DCU, sería interesante ampliar el desarrollo de la fase de evaluación, ejecutando pruebas de aceptación exhaustivas para mejorar en todo lo posible la interfaz gráfica y la usabilidad del producto.

SECCIÓN VI – REFERENCIAS

16- REFERENCIAS

Documentación

- [1] Wikipedia; Definición de Android.
<https://es.wikipedia.org/wiki/Android>
- [2] Wikipedia; Definición de iOS.
<https://es.wikipedia.org/wiki/ios>
- [3] Wikipedia; Definición de Blackberry OS.
https://es.wikipedia.org/wiki/BlackBerry_OS
- [4] Wikipedia; Definición de Windows Phone.
https://es.wikipedia.org/wiki/Windows_Phone
- [5] Herramienta de análisis de tráfico Web.
<http://gs.statcounter.com/>
- [6] Documentación oficial sobre la arquitectura de Android.
<https://developer.android.com/guide/platform/index.html>
- [7] Material UOC.
- Diseño centrado en el usuario Muriel Garreta Domingo.
 - Tecnología y desarrollo en dispositivos móviles.
 - Exposición de contenidos en vídeo.
 - Presentación de documentos y elaboración de presentaciones.
 - Redacción de textos científico-técnicos.
 - Trabajo final de máster.
 - Módulo 1. Introducción al trabajo final.
 - Módulo 2. El trabajo final como proyecto.
 - Módulo 3. La gestión del proyecto a lo largo del trabajo final.
- [8] Documentación API REST Redmine.
http://www.redmine.org/projects/redmine/wiki/Rest_api
- [9] Documentación del modelo de Base de Datos de Redmine.
<http://www.redmine.org/projects/redmine/wiki/DatabaseModel>

Recursos Software

- [10] Ganttproject: Aplicación para la realización de diagramas Gantt.
<http://www.ganttproject.biz/>
- [11] NinjaMock: Herramienta para la creación de prototipos.
<https://ninjamock.com>
- [12] Zxing: Librería para el escaneo de códigos BIDI.
<https://github.com/zxing/zxing>
- [13] Object Aid: Software para la creación de diagramas UML.
<http://www.objectaid.com/class-diagram>
- [14] SmartDraw: Software para la creación de diagramas.

<https://www.smartdraw.com>

[15] Eclipse

<https://www.eclipse.org/downloads/>

[16] Android Development Tools (ADT)

<https://developer.android.com/studio/tools/sdk/eclipse-adt.html>

[17] GenyMotion: Aplicación para la simulación de dispositivos móviles.

<https://www.genymotion.com/>

[18] Redmine

<http://www.redmine.org/>

SECCIÓN VII – ANEXOS

A- ANEXOS

A.1 Instrucciones de instalación y compilación

A.1.1 Instalación del entorno de programación

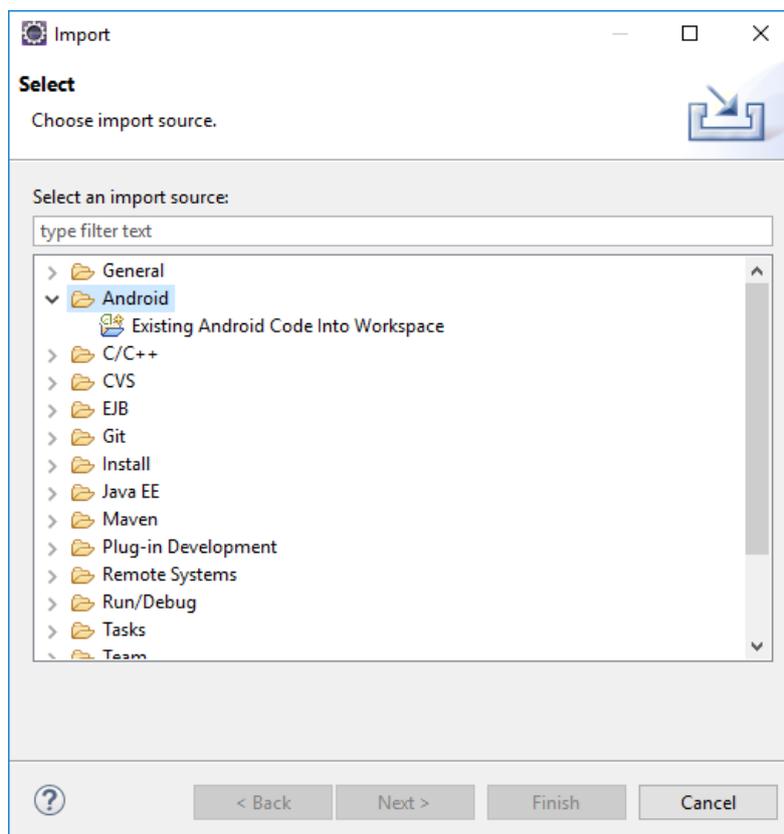
1. Descargar e instalar Eclipse (en este caso se ha usado Eclipse Luna):

- <https://eclipse.org/downloads/packages/release/luna/sr2>

2. Descargar e instalar los *plugins* ADT para Android

- <https://developer.android.com/studio/tools/sdk/eclipse-adt.html>

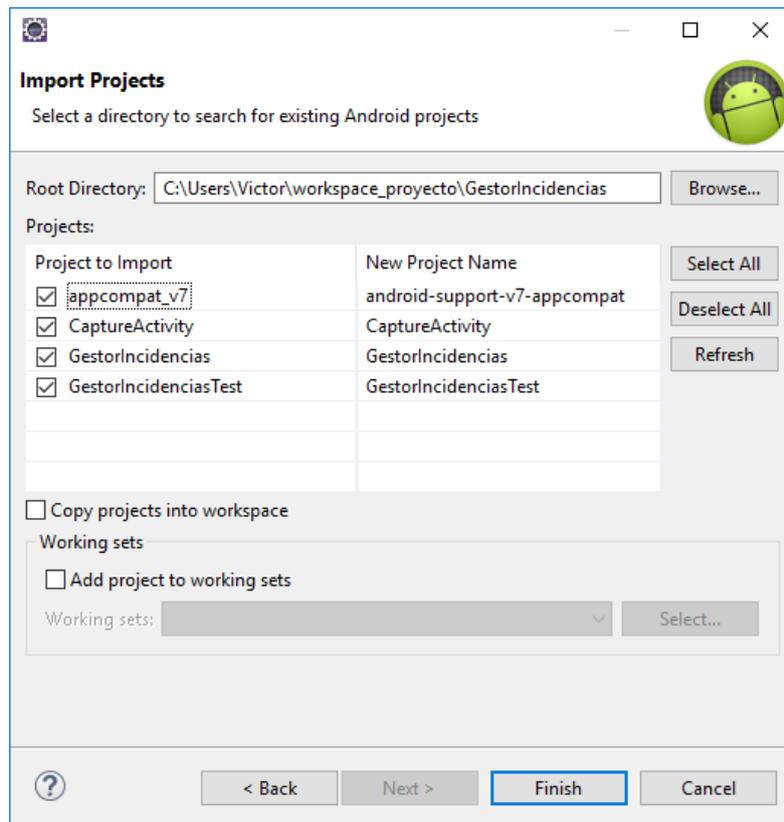
3. Descomprimir el fichero “GestorIncidencias.zip” e importar proyecto Android en Eclipse



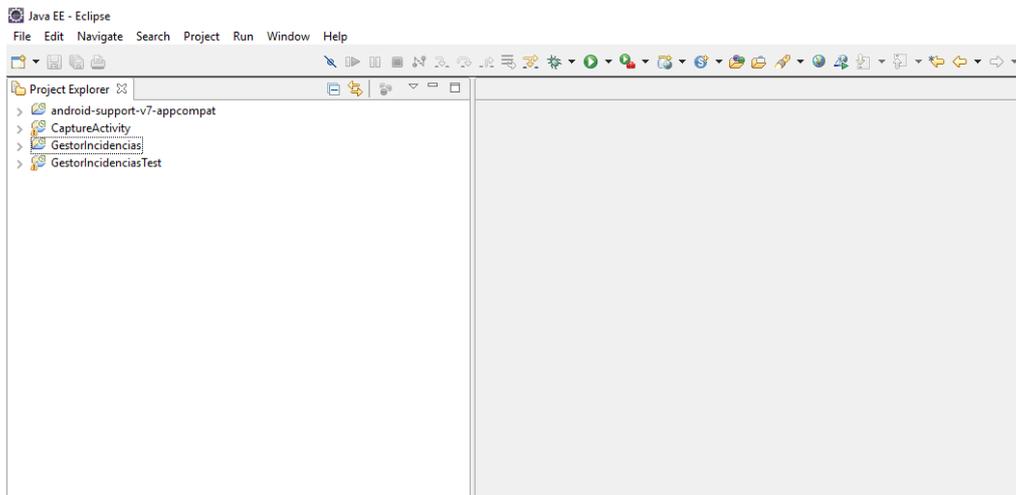
Este fichero contiene los siguientes Subproyectos

- “Appcompat_v7”: Librería de compatibilidad.
- “CaptureActivity”: Librería de escáner de códigos BIDI.

- “GestorIncidencias”: Aplicación móvil (Proyecto).
- “GestorIncidenciasTest”: Proyecto de pruebas unitarias (“JUnit”).



La estructura del proyecto quedará de la siguiente manera



4. Descargar e instalar VirtualBox y **GenyMotion** para la emulación (La utilización de GenyMotion requiere registro). Se puede usar cualquier otro emulador.

5. Modificar la siguiente línea de la clase “Constantes.java”:

```
public static final String URL = "http://192.168.0.195:80/redmine/";
```

Sustituyendo la ip por la correspondiente a su servidor Web.

```
Constantes.java
87 public static final String TIPO_PROYECTO_INTERNO = "Interno";
88
89 /*
90  * URLs
91  */
92 // Esta url sirve para leer solicitudes, dar de alta una solicitud y comprobar el usuario
93 //del login, nuestro que se realiza mediante la inserción de un issue
94 public static final String URL = "http://192.168.0.195:80/redmine/";
95
96 // Url para el tratamiento de los datos de las solicitudes (permite leer, crear, actualizar y borrar)
97 public static final String SOLICITUDES_URL = URL + "issues.json";
98
99 // Esta url devuelve los datos de una solicitud. El número de la solicitud se sustituirá por idSolicitud
100 // ejemplo solicitud_id = 1 "http://localhost/redmine/issues/1.json";
101 public static final String SOLICITUD_URL = URL + "issues/idSolicitud.json";
102
103 public static final String LEER_TIPOS_SOLICITUD_URL = URL + "trackers.json";
```

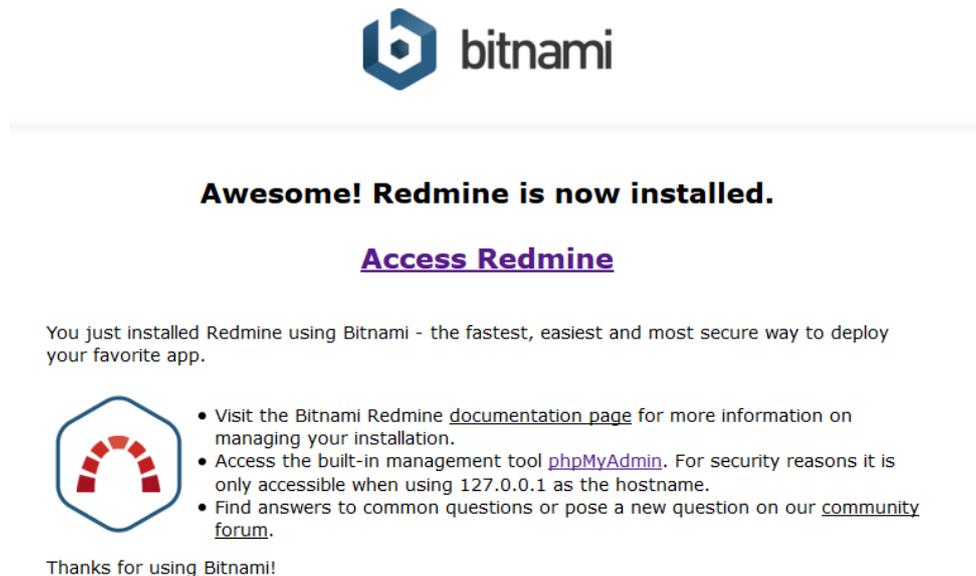
A.1.2 Instalación de Redmine

1. Descargar e instalar Bitnami con el instalador de Redmine. La versión Windows se puede descargar aquí:

- <https://bitnami.com/redirect/to/131172/bitnami-redmine-3.3.1-0-windows-installer.exe>

2. Seguir las instrucciones de instalación por pantalla

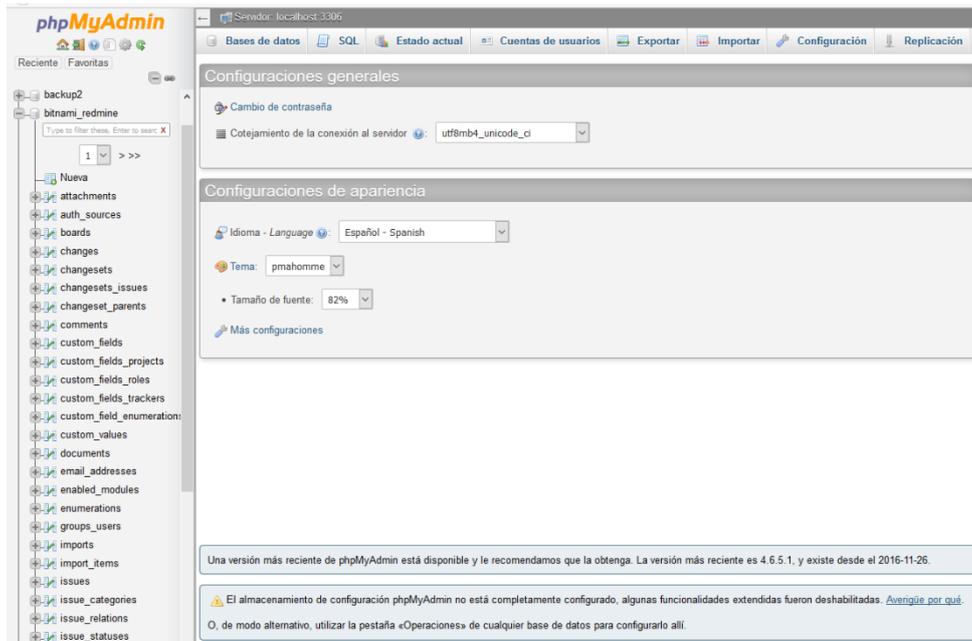
Una vez instalado aparecerá la siguiente pantalla:



* En el punto “1.2.1 Configuración del Gestor de contenidos (Redmine)” Se detallan los pasos a seguir para configurar desde cero el sistema de Redmine.

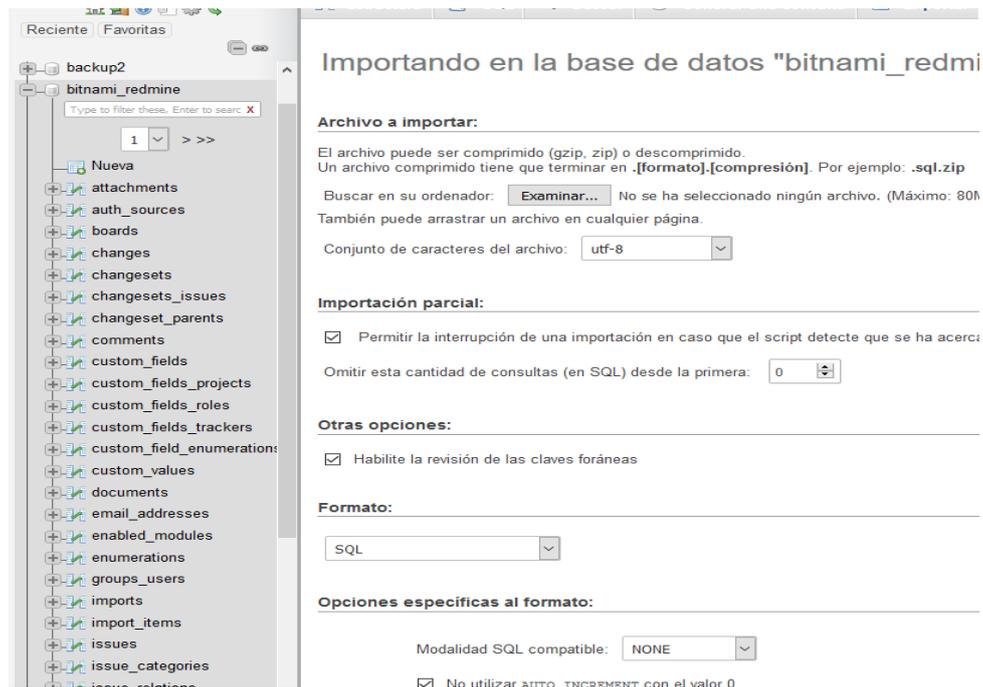
Para este manual cargaremos un fichero “sql” con las configuraciones y datos necesarios ya cargados.

3. Pulsar sobre “phpMyAdmin” y se mostrará el sistema de base de datos donde se encuentra instalado Redmine.



4. Seleccionar la Base de Datos “bitnami_redmine” e importar el fichero de Base de datos:

- “bitnami_redmine.sql”.



5. Una vez importado podremos acceder con los siguientes datos:

- **Usuario:** admin
- **Contraseña:** 12345678

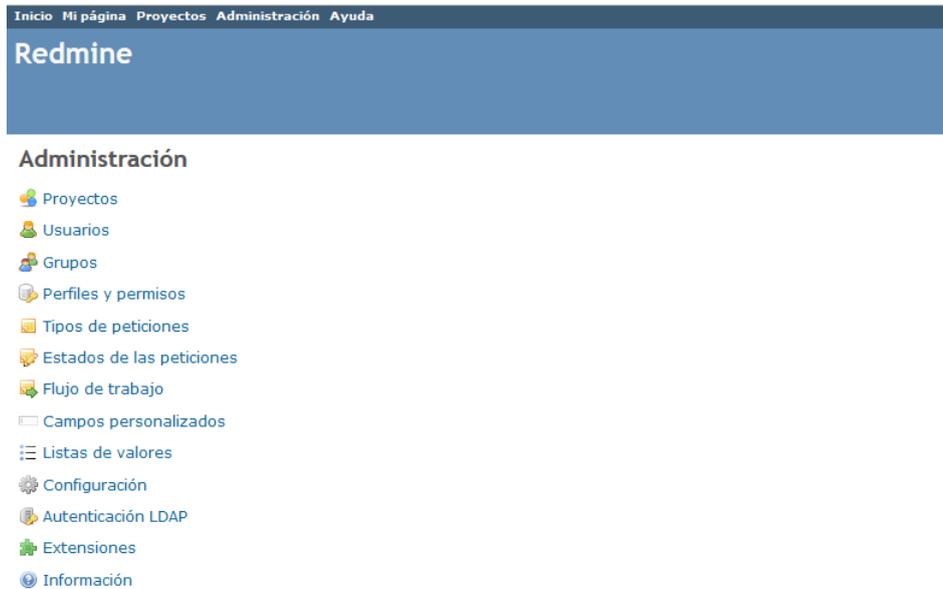


Identificador:

Contraseña:

[¿Olvidaste la contraseña?](#)

Una vez accedido se mostrará la pantalla principal.



6. Pulsar sobre la opción de administración y seleccionar la pestaña *API*.

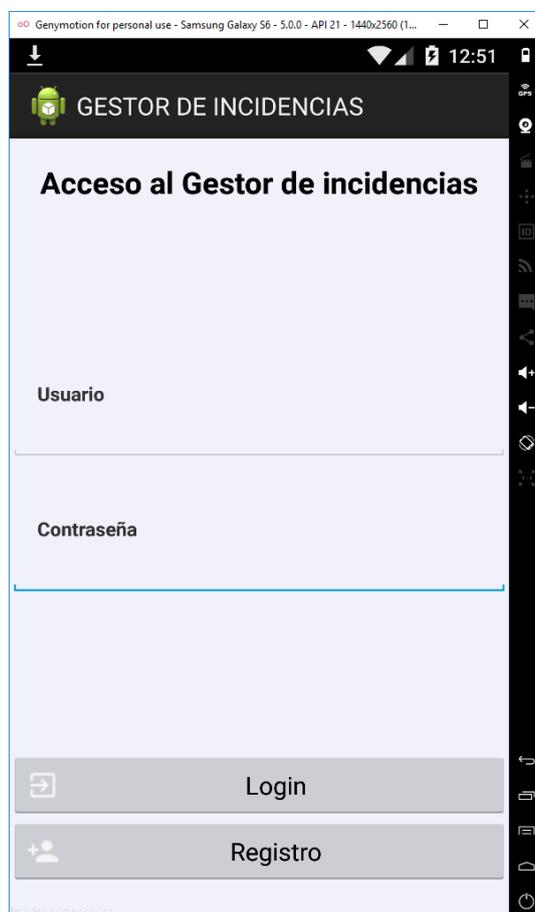
7. Pulsar sobre “Habilitar servicio web REST” para activar los Servicios Web para la aplicación y pulsar “Guardar”.



A.2 Manual de usuario

1. Entrada a la aplicación

Al iniciar la aplicación se muestra la pantalla de inicio, que se muestra a continuación:



(Figura A.1) Pantalla de inicio

En ella el usuario podrá identificarse en la aplicación para acceder a la misma.

Apartados:

- **Usuario***: Campo donde se introducirá el nombre del usuario registrado en el sistema.
- **Contraseña***: Campo donde se introducirá la contraseña del usuario registrado en el sistema.

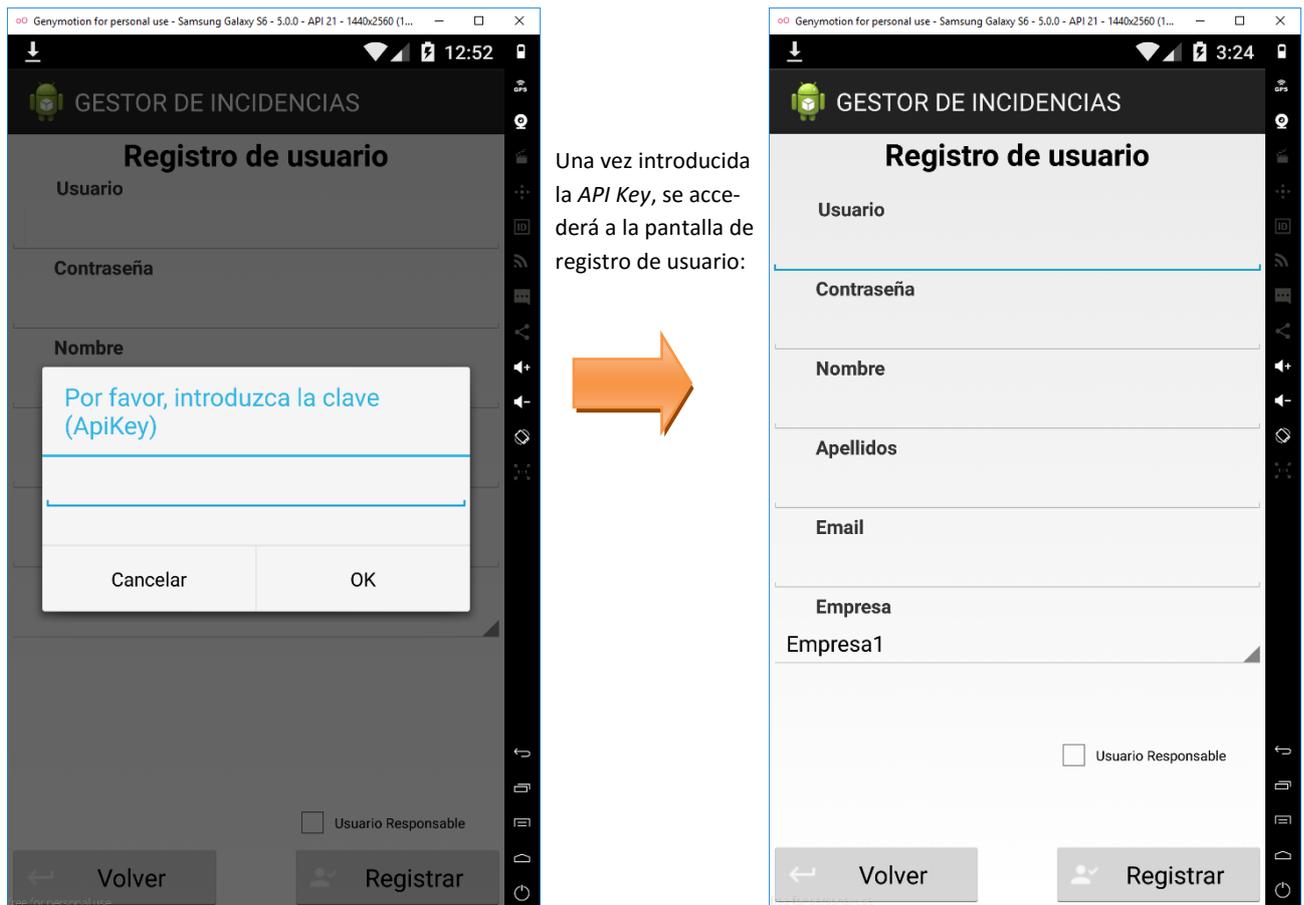
Opciones disponibles:

- **Registro**: Mediante esta opción se accede a la función de “Registro de usuario”.
- **Login**: Una vez introducido el usuario y la contraseña, se deberá pulsar este botón para acceder al menú principal de la aplicación.

*Parámetros obligatorios

1.1 Registro de usuario

Al acceder a la pantalla de registro de usuario, se solicitará la introducción de una clave (*API Key*) con permisos de administrador, como se muestra en la siguiente pantalla, que se muestra abajo a la izquierda:



(Figura A.2) Pantallas de acceso al registro de usuario

Apartados:

- **Usuario*:** Campo donde se introducirá el nombre del usuario que se desea registrar.
- **Contraseña*:** Contraseña del usuario que se desea registrar.
- **Nombre*:** Nombre propio de la persona que desea registrar su usuario.
- **Apellidos*:** Apellidos de la persona que desea registrar su usuario.
- **Email*:** Email de la persona que desea registrar su usuario.
- **Empresa*:** Se seleccionará una de las empresas dadas de alta en el sistema.
- **Usuario Responsable:** Si se marca este *Check*, el usuario poseerá perfil responsable. El perfil responsable tendrá acceso a la solicitud de nuevos desarrollos.

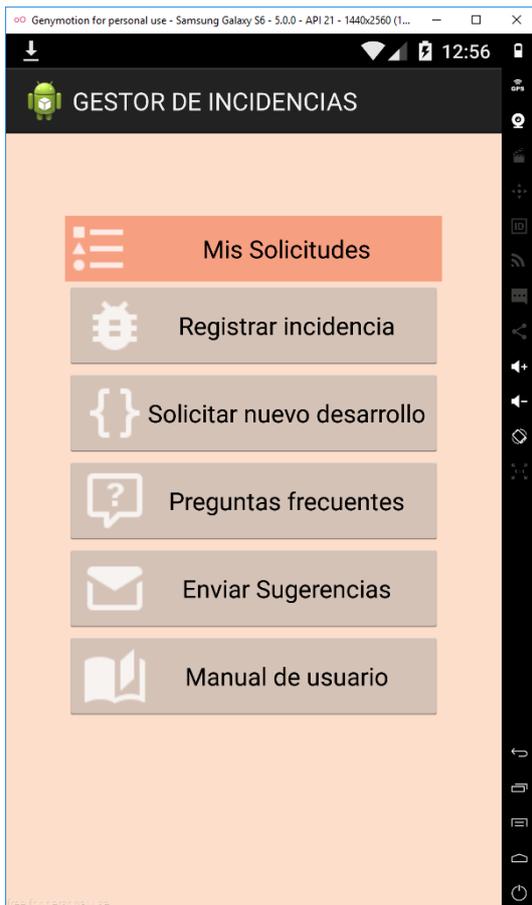
Opciones disponibles:

- **Volver:** Mediante esta opción se retornará al menú principal.
- **Registrar:** Una vez introducidos los campos obligatorios, al pulsar este botón el usuario quedará registrado en el sistema.

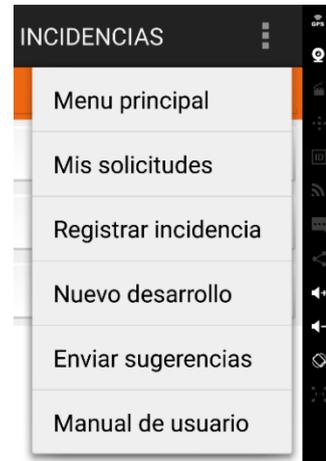
*Parámetros obligatorios

1.2 Menú principal

La pantalla de menú principal, mostrará todas las opciones de la aplicación:



Así mismo desde cualquier opción de la aplicación, será posible acceder al resto de funcionalidades mediante el menú superior, como se muestra en la siguiente pantalla:



(Figura A.3) Pantalla de Menú principal y menú desplegable

Opciones disponibles:

- **Mis solicitudes:** Opción para visualizar las solicitudes realizadas.
- **Registrar incidencia:** Funcionalidad para registrar incidencias tanto Software como Sistemas.
- **Solicitar nuevo desarrollo:** Opción para solicitar un nuevo desarrollo.
- **Enviar sugerencias:** Permite enviar sugerencias que serán almacenadas en el buzón de sugerencias.
- **Preguntas frecuentes (FAQS):** Permite visualizar las preguntas Frecuentes sobre la aplicación Software propietario.
- **Manual de usuario:** Opción para descargar el manual en formato PDF.

1.2.1 Pantalla de solicitudes realizadas

La pantalla de “Mis solicitudes” visualiza las solicitudes realizadas por el usuario.



(Figura A.4) Pantalla de Mis Solicitudes

Apartados (solo lectura):

- **ID:** Id de la solicitud.
- **Título:** Título descriptivo de la solicitud.
- **Tipo:** Tipo de solicitud. Pueden ser:
 - Incidencia Sistemas.
 - Incidencia Software.
 - Desarrollo.
 - Sugerencia.
 - Preguntas Frecuentes.
- **Estado:** Estado de la solicitud.

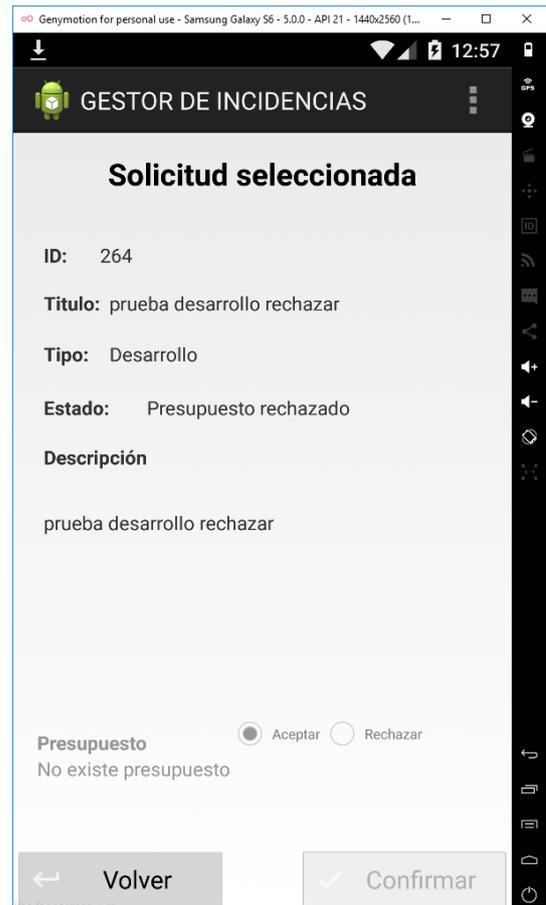
Opciones disponibles:

- **Solicitud (listado):** Al pulsar una solicitud, se accederá a la visualización del detalle de la misma.
- **Menú:** Mediante esta opción se retornará al menú principal.

1.2.1.1 Pantalla de detalle de solicitud

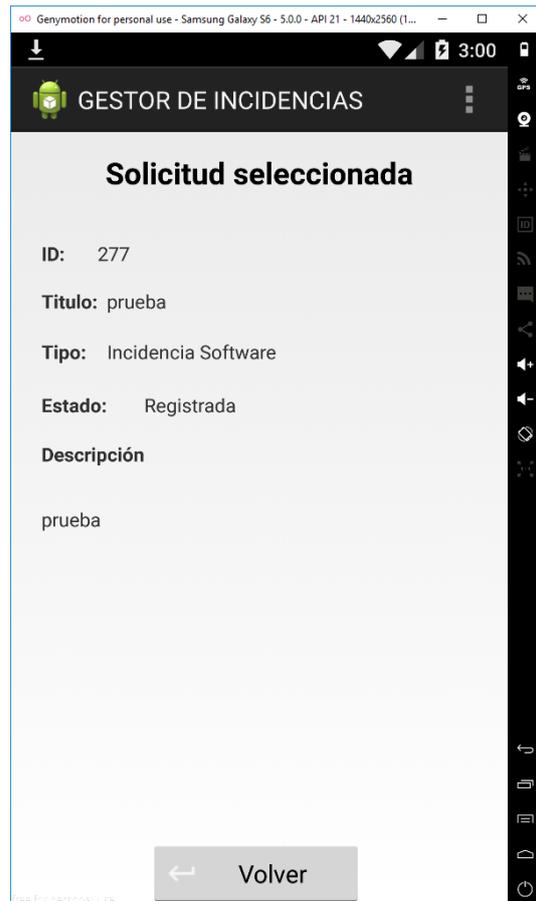
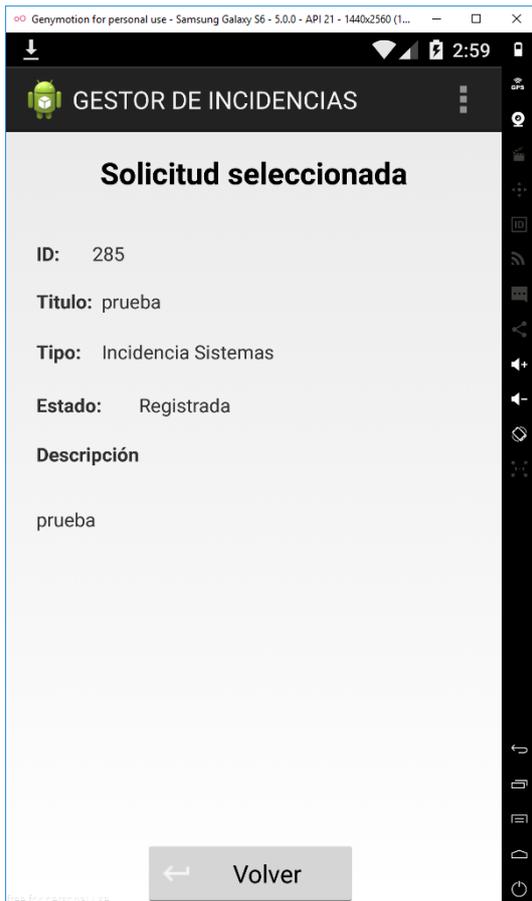
Tras pulsar sobre una de las solicitudes, se accede a la pantalla de detalle de solicitud. A continuación se muestran dos ejemplos de detalle de solicitud. A la izquierda se muestra un desarrollo en estado “pendiente de aceptación” con las opciones de aceptación de presupuesto habilitadas.

A la derecha, una solicitud de desarrollo cuyo presupuesto ha sido rechazado.



(Figura A.5) Pantallas de detalle de Solicitud pendiente de aceptación y solicitud rechazada

A continuación, se muestran dos ejemplos de detalle de incidencias. A la izquierda una incidencia de sistemas y a la derecha una incidencia de Software.



(Figura A.6) Pantallas de detalle de Incidencia de Sistemas e Incidencia de Software

Apartados (solo lectura):

A los datos de la solicitud descritos anteriormente (ID, Título y Tipo) se añade:

- **Descripción:** Datos descriptivos de la solicitud.
- **Presupuesto**:** Presupuesto de la solicitud de desarrollo enviada por la empresa.
- **Opción de Aceptación de presupuesto**:** Permite la posibilidad de aceptar o rechazar el presupuesto recibido por parte de la empresa de desarrollo.

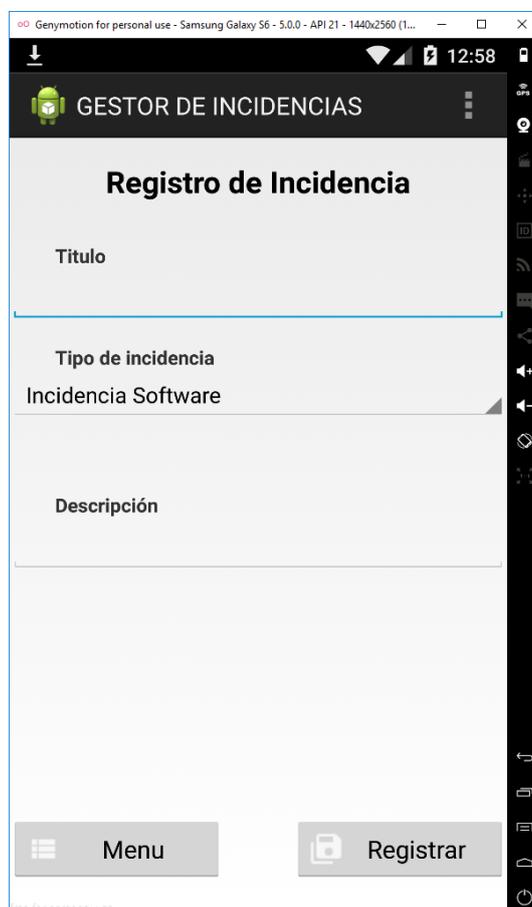
Opciones disponibles:

- **Confirmar**:** Tras aceptar o rechazar el presupuesto, se pulsará esta opción para confirmar la elección.
- **Volver:** Mediante esta opción se retornará al menú principal.

** Solo para solicitudes de Desarrollo

1.2.2 Pantalla de registro de incidencia

La pantalla de registro de incidencia, permitirá registrar una nueva incidencia en el sistema. Estas pueden ser de tipo Sistemas o Software. A continuación, se muestra la pantalla de registro con el tipo de incidencia “Incidencia Software” seleccionado:



(Figura A.7) Pantallas de Registro de Incidencia

Apartados:

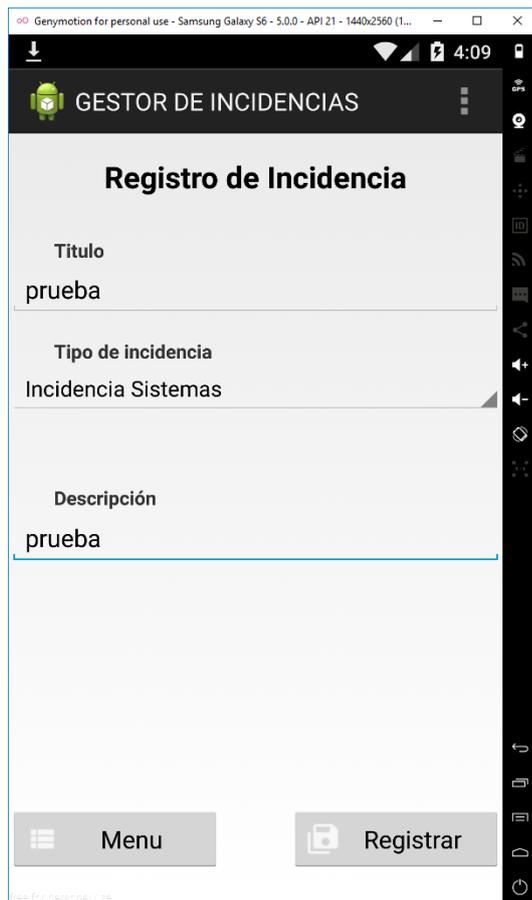
- **Título***: Título descriptivo de la incidencia.
- **Tipo de incidencia***: Permite seleccionar el tipo de incidencia (Software o Sistemas).
- **Descripción***: Datos descriptivos de la incidencia.

Opciones disponibles:

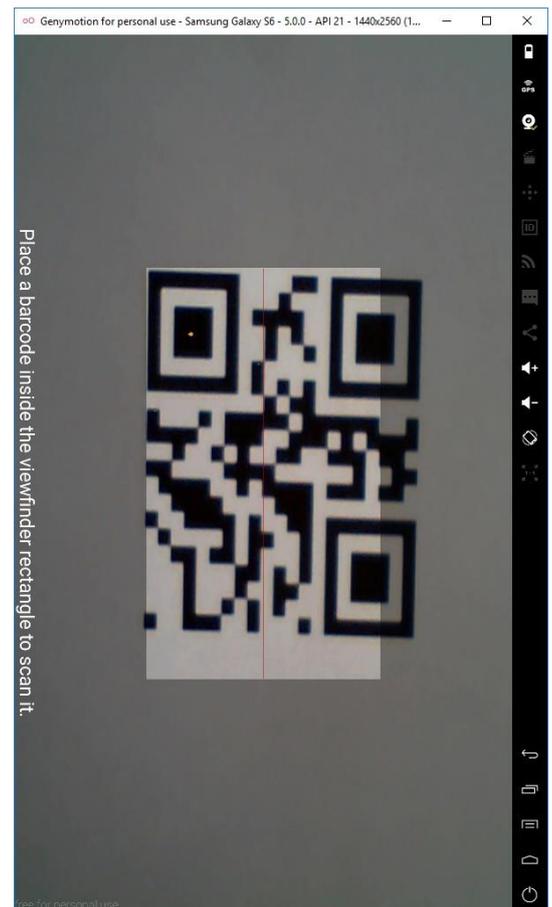
- **Menú**: Mediante esta opción se retornará al menú principal.
- **Registrar**: Una vez introducidos los campos obligatorios, al pulsar este botón el usuario quedará registrado en el sistema.

***Parámetros obligatorios**

En caso de registrar una incidencia de tipo Sistemas, la forma de registro requiere un código BIDI como se muestra a continuación:



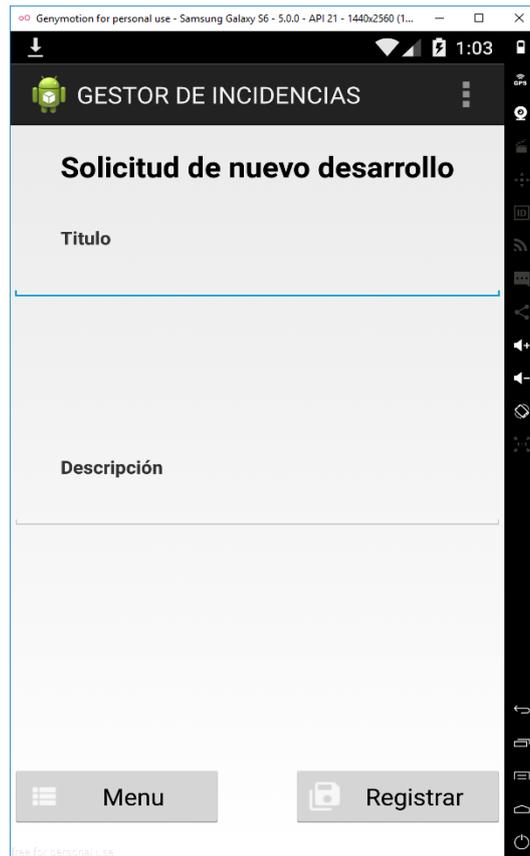
Si el usuario selecciona la opción sistemas, aparecerá la pantalla de escáner para registrar el código BIDI de la máquina.



(Figura A.8) Pantallas de Registro de Incidencia de sistemas y de escáner de Código BIDI

1.2.3 Solicitud de nuevo desarrollo

La pantalla de solicitud de nuevo desarrollo, permitirá solicitar un nuevo desarrollo de Software:



(Figura A.9) Pantalla de Solicitud de nuevo desarrollo

Apartados:

- **Título*:** Titulo de la nueva solicitud de desarrollo.
- **Descripción*:** Descripción de la solicitud.

Opciones disponibles:

- **Menú:** Mediante esta opción se retornará al menú principal.
- **Registrar:** Una vez introducidos los campos obligatorios, al pulsar este botón se registrará una nueva solicitud en estado pendiente de “Presupuesto solicitado”.

***Parámetros obligatorios**

Tras registrar la solicitud de desarrollo, se deberá acceder a Redmine y proceder a modificar sus datos, para continuar con el flujo de la solicitud:



(Figura A.10) Pantalla de Peticiónes de Redmine

Debiendo modificar la solicitud al estado “Pendiente de aceptación” e introducir un presupuesto en el campo presupuesto.

Modificar

Cambiar propiedades

Proyecto * Empresa2 Privada

Tipo * Desarrollo

Asunto * prueba

Descripción [Modificar](#)

Estado * Presupuesto solicitado

Prioridad * Pendiente de aceptacion

Asignado a

Tarea padre

Fecha de inicio 2017-01-05

Fecha fin

Tiempo estimado Horas

% Realizado 0 %

Presupuesto 200

Tiempo dedicado

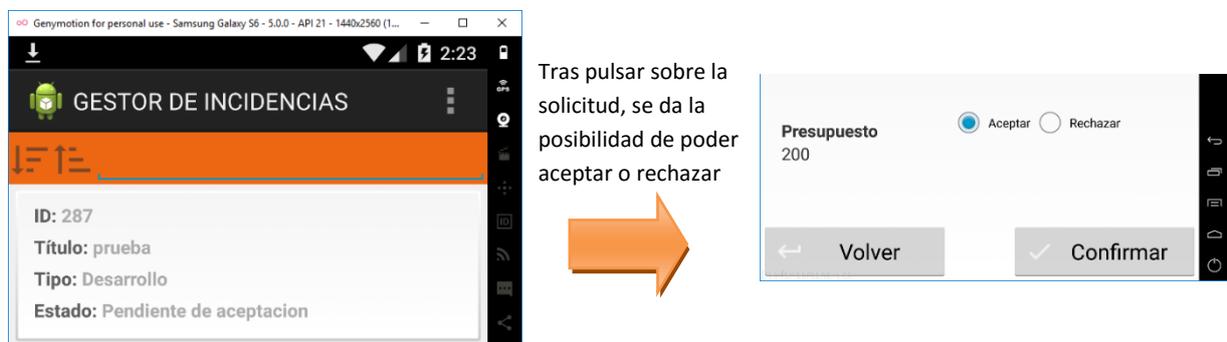
Tiempo dedicado Horas

Actividad --- Por favor seleccione ---

Comentario

(Figura A.11) Pantalla de modificación de petición de Redmine

Una vez enviado el presupuesto, la solicitud será accesible desde la aplicación:

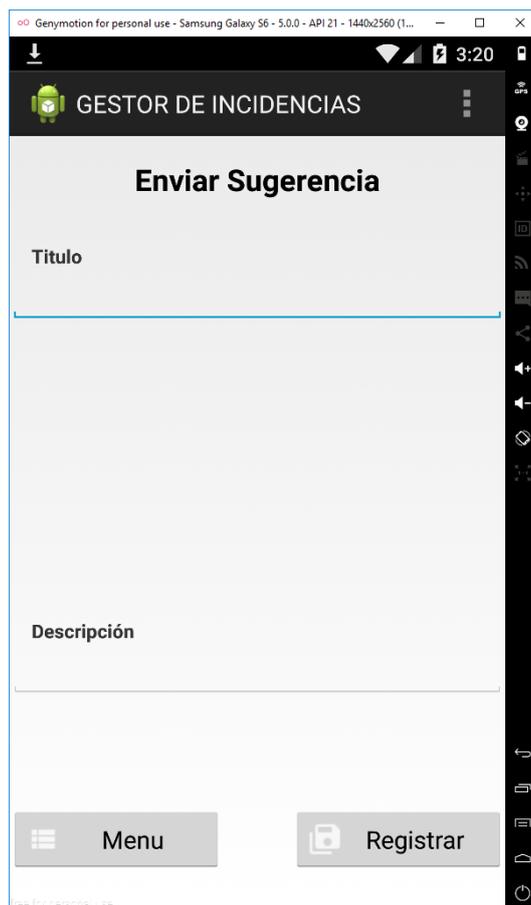


(Figura A.12) Flujo de aceptación/rechazo de solicitud de desarrollo

Si la solicitud es rechazado, esta volverá al estado inicial "Presupuesto solicitado", dando la posibilidad de volver a enviar un nuevo presupuesto.

1.2.4 Pantalla de envío de sugerencias

La pantalla de envío de sugerencias actuará de buzón, para que la empresa de desarrollo reciba sugerencias y solicitudes de mejora por parte de los clientes.



(Figura A.13) Pantalla de Envío de sugerencias

Apartados:

- **Título***: Titulo de la sugerencia que se desea enviar.
- **Descripción***: Datos descriptivos de la sugerencia que se desea enviar.

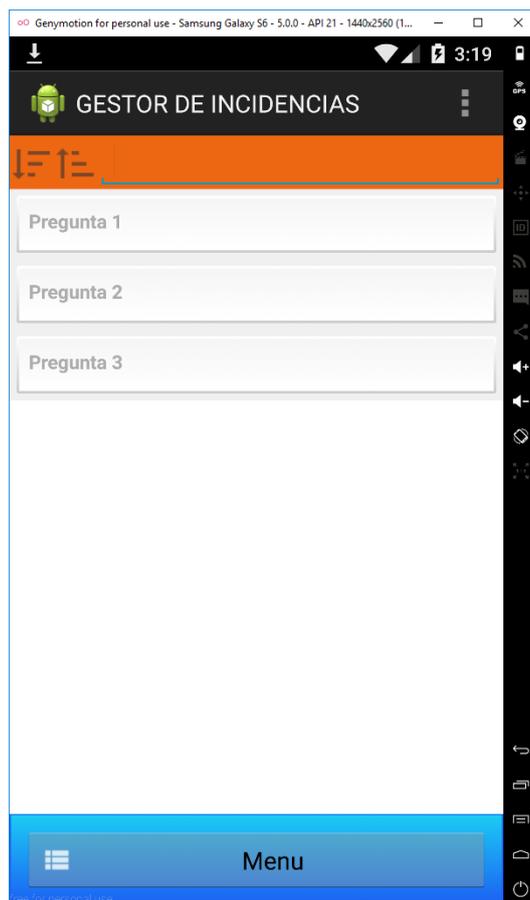
Opciones disponibles:

- **Menú**: Mediante esta opción se retornará al menú principal.
- **Registrar**: Una vez introducidos los campos obligatorios, al pulsar este botón la sugerencia quedará registrado en el sistema.

*Parámetros obligatorios

1.2.5 Pantalla de preguntas frecuentes

La pantalla de preguntas frecuentes (FAQS), mostrará un listado de preguntas frecuentes con sus respuestas sobre el Software propietario.



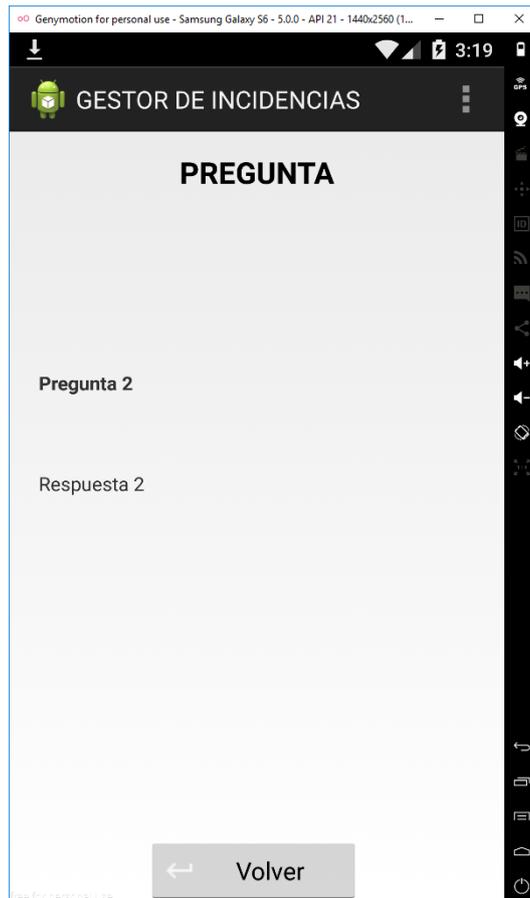
(Figura A.14) Pantalla de listado de preguntas frecuentes

Apartados (solo lectura):

- **Pregunta:** Título descriptivo de la pregunta.

Opciones disponibles:

- **Pregunta (listado):** Al pulsar una solicitud, se accederá a la visualización del detalle de la misma.
- **Menú:** Mediante esta opción se retornará al menú principal.



(Figura A.15) Pantalla de detalle de pregunta frecuente seleccionada

Apartados (solo lectura):

- **Pregunta:** Título descriptivo de la pregunta.
- **Respuesta:** Detalle de la respuesta.

Opciones disponibles:

- **Volver:** Mediante esta opción se retornará al menú principal.