



Línea Verde: Aviso Ciudadano Aplicación para la gestión de incidencias en el equipamiento urbano de un municipio

Nombre Estudiante: Aureo Serrano Díaz-Carrasco

Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

Nombre Consultor/a: Eduard Martin Lineros

Profesor/a responsable de la asignatura: Carles Garrigues Olivella

08 de Junio de 2016

© Aureo Serrano Díaz-Carrasco

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 microfilme, 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:	Aplicación para la gestión de incidencias en el equipamiento urbano de un municipio
Nombre del autor:	<i>Aureo Serrano Díaz-Carrasco</i>
Nombre del consultor/a:	Eduard Martin Lineros
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	06/2016
Titulación::	Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Android, ciudadano, quejas</i>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>Aplicación desarrollada para dispositivos Android, destinada a facilitar a los ciudadanos de un municipio, de pequeño tamaño, un canal de comunicación con el consistorio. Mediante esta aplicación los ciudadanos podrán poner en conocimiento del ayuntamiento de la localidad, las incidencias y/o problemas derivados del mantenimiento del equipamiento urbano.</p> <p>Los ciudadanos, tomarán una fotografía, indicaran un punto en el mapa de la localidad y añadirán una breve descripción del problema y esta información se almacenará en una base de datos central.</p> <p>Por otro lado, los servicios municipales de mantenimiento y obras, desde una sección privada de esta misma aplicación, podrán ver todas las quejas almacenadas e intentar solucionarlas en la mayor brevedad posible.</p>	
<p>Abstract (in English, 250 words or less):</p>	
<p>Application developed for Android devices. It is intended to provide citizens of a small town, a channel of communication with the consistory.</p> <p>With this application, citizens will be able to inform the local municipality, of incidents and / or problems arising from maintenance of urban facilities.</p>	

Citizens will take a photograph, indicate a point on the map of the town and add a brief description of the problem and this information will be stored in a central database.

On the other hand, municipal maintenance services can see all stored complaints from a private section of this same application, and try to solve them as quickly as possible.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo	1
1.2 Objetivos del Trabajo.....	3
1.3 Enfoque y método seguido.....	5
1.4 Planificación del Trabajo	6
1.5 Breve resumen de productos obtenidos	9
1.6 Breve descripción de los otros capítulos de la memoria	9
2. Usuarios y contexto de uso	10
2.1 Investigación	10
2.2 Perfiles de usuario.....	11
2.3 Contextos de Uso.....	11
2.4 Análisis de tareas	12
3. Diseño conceptual.....	13
3.1 Escenarios de uso.....	13
3.2 Diagrama de Flujo	15
4. Prototipado.....	15
4.1 Zona de usuario ciudadano	16
4.2 Zona de usuario municipal	19
5. Evaluación.....	21
6. Casos de Uso.....	22
6.1 Diagrama UML	22
6.2 Casos de Uso.....	23
7. Diseño de la Arquitectura	26
7.1 Vista	26
7.2 Controlador.....	26
7.3 Modelo.....	27
7.4 Persistencia de los datos.....	27
8. Implementación	28
8.1 Servidor Backend	28
9. Conclusiones.....	30
10. Glosario	31
11. Bibliografía	32
12. Anexo 1: Manual de Instalación	33
12.1 Instalación de la parte servidor.....	33
12.2 Instalación de la parte cliente	33
12.3 Probar la aplicación	34
13. Anexo 2: Manual de Usuario	35
13.2 Sección de Usuario Ciudadano	35
13.3 Sección de Usuario Municipal	41

Lista de figuras

Ilustración 1 - Diagrama de Gantt	8
Ilustración 2 - Diagrama de flujo de la aplicación	15
Ilustración 3 - Pantalla inicial ciudadano	16
Ilustración 4 - Pantalla detalle ciudadano	17
Ilustración 5 - Pantalla creación problema fotografía	18
Ilustración 6 - Pantalla creación problema localización	18
Ilustración 7 - Pantalla creación problema descripción	18
Ilustración 8 - Pantalla municipal login	19
Ilustración 9 - Pantalla municipal principal	20
Ilustración 10 - Pantalla municipal detalle	20
Ilustración 11 - Diagrama UML ciudadano	22
Ilustración 12 - Diagrama UML municipal	22
Ilustración 13 - Arquitectura de la aplicación	26
Ilustración 14 - Diagrama de estructura de datos	27
Ilustración 15 Pantalla de inicio	35
Ilustración 16 Pantalla de detalle ciudadano	36
Ilustración 17 Pantalla insertar posición	37
Ilustración 18 Pantalla insertar imagen	38
Ilustración 19 Pantalla con foto	38
Ilustración 20 Pantalla observaciones	39
Ilustración 21 Problema añadido	39
Ilustración 22 Eliminar problema	40
Ilustración 23 Pantalla de login	41
Ilustración 24 Pantalla lista municipal	42
Ilustración 25 Pantalla detalle municipal	43

1. Introducción

1.1 Contexto y justificación del Trabajo

En cualquier localidad, uno de los principales problemas del consistorio es el mantenimiento de todas las infraestructuras de la misma.

Dejando de lado los posibles problemas monetarios que dificulten esta misión, el consistorio se encuentra con la gran dificultad que supone la detección de las incidencias que debe solucionar, especialmente si son de poca importancia: una farola fundida, pequeños desperfectos en las calles o las aceras, papeleras rotas..., etc.

Actualmente, estos problemas se detectan mediante inspecciones realizadas por el propio personal de mantenimiento del ayuntamiento, o atendiendo a las quejas que pueden realizar los propios habitantes del municipio en los plenos ordinarios, o personándose ante los órganos competentes. Estos métodos se presentan ineficaces y consumen un gasto de personal demasiado elevado.

Como solución, el consistorio se ha planteado la creación de un equipo de mantenimiento de acción rápida, acudiendo a la colaboración ciudadana mediante una aplicación móvil, que permita a cualquier habitante del municipio la comunicación de cualquier incidencia en cualquier momento. Dichas incidencias se almacenarán en una base de datos centralizada, pudiendo el equipo de acción rápida acceder a ellas y comprometiéndose en la medida de lo posible a dar respuesta a las mismas en un plazo de 48 horas.

Como ventajas añadidas, con el uso de una aplicación móvil de estas características, se fomenta la participación ciudadana en los asuntos municipales, y al proporcionar un feedback del estado de las quejas a lo largo del tiempo, el ciudadano se siente involucrado y siente que se “está haciendo algo al respecto”.

Existen aplicaciones similares que podrían cumplir con los requerimientos que se desean. Principalmente, se podrían dividir en dos tipos:

- **Aplicaciones de tipo red social:**

Un ejemplo claro de este tipo de aplicaciones sería el WhatsApp. En principio, esta aplicación podría cumplir con los requerimientos deseados. Permitiría mandar los avisos a una cuenta perteneciente al municipio, con los datos y la fotografía y desde allí podrían ser distribuidos.

Los problemas que plantearon en el consistorio a esta solución son varios:

- No permite filtrado de mensajes, por fecha, por estado del aviso, etc. Lo que hace más difícil la organización.
- Se hace también más difícil el proporcionar un feedback al ciudadano.
- Tienes que organizar el sistema en base a mensajes, que pueden o no corresponderse con avisos.
- Los datos no están centralizados, sino que estarían en el terminal.

- **Aplicaciones privadas:**

Otra opción que se planteó es el contratar la aplicación con una empresa que la ha implantado en algunos municipios cercanos, (www.lineaverdemunicipal.com). Esta opción tenía la ventaja de que toda la programación y el mantenimiento de la aplicación, así como el espacio en la nube necesario para los datos, corrían a cargo de la empresa.

Los problemas que hicieron desechar esta opción son los siguientes:

- El propio coste de alquiler y mantenimiento de la aplicación.
- No tienes acceso físico a los datos, solo a la aplicación de que proporciona la empresa para su gestión.
- Tampoco tienes acceso a la aplicación en sí, cualquier modificación (por ejemplo no permitía introducir la localización a mano) no puedes realizarla directamente.

Así pues y valoradas estas opciones, se decidió optar por la realización de una aplicación propia sencilla, de fácil uso y mantenimiento, orientada a las necesidades de un municipio de pequeño tamaño, de tal manera que no se tenga que afrontar un gasto continuo en concepto de alquiler o mantenimiento, ya que de esto se encargaría, una vez terminada la aplicación, el propio personal informático del municipio.

Además se ha planteado esta aplicación como un comienzo a lo que sería una "SmartApp" municipal. Una vez en funcionamiento, se pretende ampliarla con distintos servicios de forma progresiva, como son mensajes broadcast a los ciudadanos para eventos especiales, anuncios y promociones de empresas del municipio, gestiones administrativas locales, plataforma de pago para las infraestructuras municipales (piscina municipal, alquiler de pistas en el polideportivo, etc).

El tener este ambicioso objetivo, hace necesario que la aplicación sea conveniente realizarla desde cero, para poder ampliarla con más facilidad.

1.2 Objetivos del Trabajo

El objetivo de este trabajo es el desarrollo de una aplicación Android y una base de datos centralizada, con la que pueda comunicarse dicha aplicación mediante servicios REST.

La aplicación estará dirigida a dos tipos de usuarios, por un lado a un usuario ciudadano del municipio y por otro a un usuario miembro del equipo de acción rápida del ayuntamiento.

Para cada uno de estos tipos de usuarios, debe cumplir los siguientes requisitos funcionales:

Ciudadanos del municipio		
Id	Descripción	Prioridad
RF 1 Obtención de datos		
RF 1.1	El usuario podrá tomar una fotografía de la incidencia de la que desee informar.	Alta
RF 1.2	El usuario podrá, en lugar de tomar una fotografía en ese momento, seleccionar una de las almacenadas en el dispositivo.	Media
RF 1.3	El sistema deberá reescalar las imágenes para reducir su tamaño en disco a un máximo de 100Kb.	
RF 1.4	El sistema detectará en la medida de lo posible la posición GPS del usuario.	Media
RF 1.5	El usuario podrá indicar en un mapa centrado en su posición actual, la posición de la incidencia.	Alta
RF 1.6	El usuario podrá introducir una descripción sobre la incidencia de la que desea informar.	Alta
RF 1.7	El sistema obtendrá un identificador único para ese dispositivo que asociará a la incidencia.	Alta
RF 1.8	El sistema, a parte del identificador, intentará	Baja

	obtener el número de teléfono del usuario para asociarlo a la incidencia.	
RF 2 Incidencias del usuario		
RF 2.1	El usuario podrá ver una lista con las incidencias asociadas a su dispositivo y el estado de las mismas (enviada, en curso, resuelta)	Alta
RF 2.2	El usuario podrá eliminar las incidencias que no desee que se muestren en su lista. Esto solo las marcará como no visibles en la BD, no serán eliminadas realmente.	Media
Miembros del grupo de acción rápida		
Id	Descripción	Prioridad
RF 3 Credenciales		
RF 3.1	El sistema no dispondrá de gestión de usuarios, se crearán las credenciales directamente en la BD.	Alta
RF 3.2	El usuario se podrá identificar como miembro del grupo de acción rápida, usando las credenciales creadas al efecto.	Alta
RF 4 Lista de incidencias		
RF 4.1	El usuario podrá ver una lista con todas las incidencias pendientes de resolver, y podrá consultar los detalles de cada incidencia.	Alta
RF 4.2	El usuario podrá cambiar el estado de las incidencias, de “enviada” a “en curso”, y de “en curso” a “resuelta”.	Alta
RF 4.3	Una vez en estado “resuelta”, las incidencias no serán mostradas en la lista de incidencias pendientes.	Media
RF 4.4	El usuario podrá ver una lista de histórico, donde podrá buscar por rango de fechas, incidencias resueltas anteriormente.	Baja

Así mismo, la aplicación deberá cumplir los siguientes requisitos no funcionales:

Id	Descripción	Prioridad
RNF 1 Rendimiento y Escalabilidad		
RNF 1.1	El sistema deberá cargar las listas de incidencias en un tiempo no superior a 5 segundos	Media
RNF 1.2	El sistema deberá cargar los detalles de una incidencia en un tiempo no superior a 5 segundos	Media
RNF 1.3	El sistema deberá ser escalable ante la posibilidad de que deba ser ampliado para dividir las incidencias por tipo y su asignación a distintos grupos de trabajo.	Baja
RNF 2 Seguridad		
RNF 2.1	Los datos no son sensibles, así pues, el sistema presentará una seguridad basada en credenciales de nombre de usuario y contraseña.	Alta
RNF 2.2	Los datos sobre las incidencias serán transmitidos por la red sin ningún tipo de encriptación.	Alta
RNF 3 Requisitos Hardware		
RNF 3.1	La aplicación deberá poder ejecutarse en un dispositivo móvil Android 4.0.3 o superior.	Alta
RNF 3.2	Se dispondrá de un servidor donde se almacenará toda la información sobre las incidencias, así como las credenciales de los usuarios	Alta
RNF 3.3	El servidor deberá tener instalados: <ul style="list-style-type: none"> • Node 4.3 • MongoDB version 3.0.X • Python 2.7.1 	Alta

1.3 Enfoque y método seguido

Como ya se ha mencionado anteriormente existen aplicaciones similares en el mercado. El enfoque que se seguirá será crear una aplicación similar a las existentes pero a pequeña escala, orientada a su uso en una localidad de pequeño tamaño.

1.4 Planificación del Trabajo

Para la realización del proyecto, se contará con las siguientes herramientas:

- Android Studio.
- Netbeans
- MongoDB

Para la implementación de la base de datos centralizada y los servicios REST, se usará la plataforma de código abierto de [Parse](#).

El desarrollo del proyecto correrá a cargo únicamente del alumno Aureo Serrano Díaz-Carrasco, con ayuda de los consultores del TFM.

A continuación se muestra el diagrama de tareas elaborado con Microsoft Project 2013:

Nombre de tarea	Duración	Comienzo	Fin
PEC 1-Plan de trabajo	15 horas	jue 03/03/16	mié 09/03/16
Contexto y justificación del Trabajo	3 horas	jue 03/03/16	jue 03/03/16
Objetivos del Trabajo	5 horas	vie 04/03/16	lun 07/03/16
Enfoque y método seguido	1 hora	lun 07/03/16	lun 07/03/16
Planificación del trabajo	5 horas	mar 08/03/16	mié 09/03/16
Breve resumen de productos obtenidos	1 hora	mié 09/03/16	mié 09/03/16
Entrega de la PEC 1	0 horas	mié 09/03/16	mié 09/03/16
PEC 2 - Diseño	62 horas	mié 09/03/16	mié 06/04/16
Diseño centrado en el usuario	22 horas	jue 10/03/16	lun 21/03/16
Usuarios y contexto de uso	3 horas	jue 10/03/16	jue 10/03/16
Diseño conceptual	7 horas	vie 11/03/16	mar 15/03/16
Prototipado	10 horas	mar 15/03/16	vie 18/03/16
Evaluación	2 horas	vie 18/03/16	lun 21/03/16
Diseño técnico de la aplicación	38 horas	lun 21/03/16	mié 06/04/16
Casos de uso	14 horas	lun 21/03/16	vie 25/03/16
Diseño de la arquitectura	24 horas	lun 28/03/16	mié 06/04/16

Entrega de la PEC 2	0 horas	mié 06/04/16	mié 06/04/16
PEC 3 Implementación	93 horas	mié 06/04/16	mié 18/05/16
Desarrollo	75 horas	jue 07/04/16	mié 11/05/16
Instalación del servidor de BD y configuración de servicios REST	15 horas	jue 07/04/16	mié 13/04/16
Implementación de la aplicación móvil	60 horas	jue 14/04/16	mié 11/05/16
Pruebas	13 horas	jue 12/05/16	mié 18/05/16
Test unitarios	13 horas	jue 12/05/16	mié 18/05/16
Entrega de la PEC 3	0 horas	mié 18/05/16	mié 18/05/16
Entrega final	48 horas	mié 18/05/16	mié 08/06/16
Elaboración final de la memoria del proyecto	10 horas	mié 18/05/16	lun 23/05/16
Manuales de uso e instalación	22 horas	lun 23/05/16	mié 01/06/16
Preparación de ejecutables a entregar	4 horas	jue 02/06/16	vie 03/06/16
Presentación del proyecto	10 horas	vie 03/06/16	mié 08/06/16
Entrega final del TFM	0 días	mié 08/06/16	mié 08/06/16

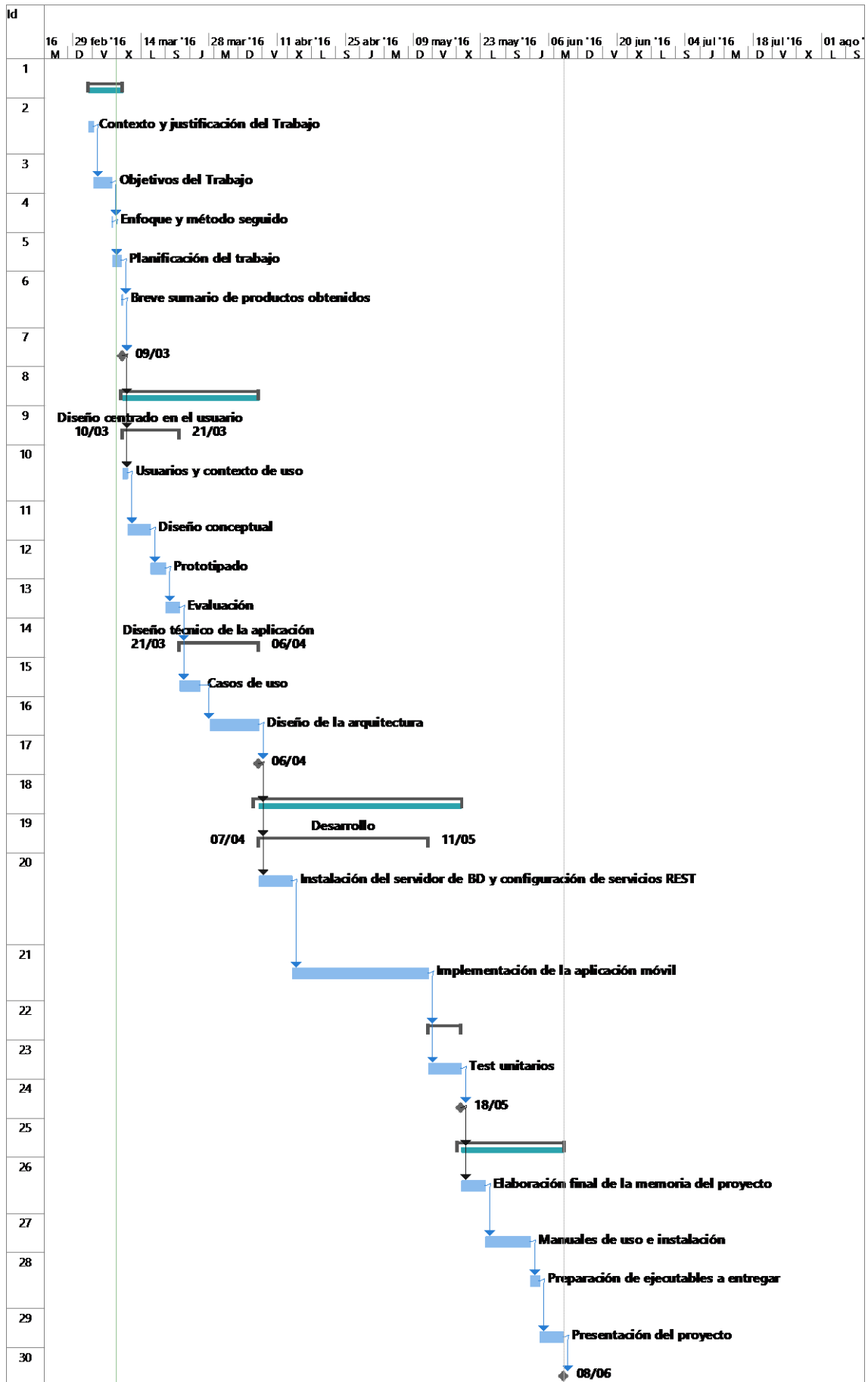


Ilustración 1 - Diagrama de Gantt

1.5 Breve resumen de productos obtenidos

- Código fuente de la aplicación móvil.
- Ejecutable (.apk) de la aplicación.
- Estructura de la base de datos del servidor.
- Memoria del proyecto.
- Manuales de uso e instalación.

1.6 Breve descripción de los otros capítulos de la memoria

En los siguientes capítulos de la memoria se detallará el análisis, el diseño de la aplicación, y la implementación, así como el plan de pruebas y las conclusiones finales.

En el análisis de la aplicación usaremos un **diseño centrado en el usuario**. Veremos los usuarios y contexto de uso, el diseño conceptual, el prototipado y la evaluación del análisis.

En la parte del diseño, nos centraremos en los casos de uso, la arquitectura del sistema, y el diseño físico de los datos.

2. Usuarios y contexto de uso

En este punto, vamos a analizar cuáles son las características de los usuarios, sus objetivos y contexto de uso de la aplicación, con el fin de detectar las necesidades para satisfacer a los usuarios.

Para ello, haremos un estudio de quiénes serán los usuarios de la aplicación, qué tareas realizarán y cómo utilizarán el sistema.

2.1 Investigación

La investigación se realiza mediante entrevistas con los responsables del consistorio del municipio, interrogando sobre las funcionalidades que desean en el sistema y sobre cómo y quien realiza esta misma tarea actualmente.

La tarea a la que la aplicación aspira a dar solución es el aviso temprano de desperfectos o incidencias en el mobiliario urbano del municipio.

Actualmente, esta tarea se trata de la siguiente manera:

- Los responsables de los servicios de obras del municipio, realizan inspecciones periódicas del estado de las infraestructuras del municipio, procediendo a subsanar cualquier deficiencia con la que se encuentren.
- Cualquier ciudadano que detecte una incidencia, puede o bien presentar una instancia escrita en el registro del ayuntamiento, describiendo el problema y pidiendo su resolución, o bien, una vez al mes, en los plenos ordinarios, puede acudir y presentar sus alegaciones en la sección de ruegos y preguntas.

Los responsables del consistorio, hacen especial hincapié en que además de poder alertar sobre las incidencias que se produzcan, los ciudadanos puedan ver si su incidencia se ha recibido y se está haciendo algo al respecto.

Los miembros de los servicios de obras y mantenimiento del ayuntamiento, están acostumbrados al uso del móvil en su trabajo habitual, con lo que la adaptación al uso de la aplicación propuesta, no parece suponer un problema, siempre que se presente un interfaz fácil e intuitivo.

Según nos han comentado, los miembros del equipo de mantenimiento serán entre cuatro y cinco personas, siendo un jefe de equipo el que da el visto bueno y asigna las tareas diarias, así pues, normalmente será esta persona la que vaya a hacer uso de la aplicación en este sentido.

Por este motivo, nos proponen el que se guarden las credenciales de esa persona en el móvil, de tal manera que no tenga que identificarse cada vez que arranque la aplicación.

Consultando con los responsables del registro del consistorio, nos hacen ver que las quejas ciudadanas no presentan un perfil fijo de personas, estando estas en un rango amplio de los 25 a los 60 años. Siendo las personas menores de 25 años las que menos se preocupan por estos temas.

2.2 Perfiles de usuario

Así pues, nos encontramos con dos perfiles de usuarios que usarán la aplicación:

Miembros del equipo de mantenimiento del consistorio

- Franja de edad de los 20 a los 60 años.
- Acostumbrados al uso del móvil en su entorno de trabajo, envío de mensajes, correo electrónico, etc.
- No se prevén problemas en su adaptación al uso de la aplicación.
- Generalmente será siempre la misma persona.

Ciudadanos del municipio

Distintas franjas de edad:

- De 15 a 25 años:
Son los más acostumbrados al uso de aplicaciones móviles y nuevas tecnologías. Es el rango de edad que presenta menos inquietud con los problemas del municipio.
- De 25 a 50 años:
En su mayor parte disponen de un Smartphone, y lo usan habitualmente. No están tan acostumbrados a los cambios como el rango anterior, pero se adaptan a las nuevas tecnologías, con un breve periodo de aprendizaje.
- Mayores de 50 años:
Es el rango de edad más difícil para la implantación de una aplicación móvil, ya que no suelen estar acostumbrados a las nuevas tecnologías. Hay que cuidar los interfaces, de manera que presenten iconos grandes, no aglutinen información y sean lo más sencillos posibles.

2.3 Contextos de Uso

Se han identificado los siguientes contextos de uso:

- Un usuario, miembro del equipo de mantenimiento municipal usará la aplicación a lo largo de su jornada laboral para dar solución a las incidencias que indiquen los ciudadanos.
- Un ciudadano cualquiera, en cualquier momento del día, puede observar un problema en el equipamiento urbano y usar la aplicación para comunicarlo.

Independientemente del contexto de uso, para su funcionamiento la aplicación requerirá que el dispositivo móvil disponga de conexión a internet.

2.4 Análisis de tareas

Estas son las tareas que pueden realizar los usuarios con la aplicación:

- Comunicar un problema:
Al usuario se le indicará que proporcione una fotografía del problema, bien tomándola en ese momento, o bien cogiéndola de la galería. Seguidamente, se le mostrará un mapa del municipio, centrado en la posición actual (de ser posible), para que indique la posición del problema. Por último, se le pedirá una breve descripción del problema y se almacenarán todos estos datos.
- Consultar el estado de un problema:
Nada más iniciar la aplicación al usuario se le presentará una lista con los problemas pendientes, (informados por él, si es un usuario normal, o por cualquiera si forma parte del equipo de mantenimiento) en esta lista podrá elegir cualquier problema para que se le presenten los datos asociados a este.
- Borrar un problema de la lista:
Los problemas no podrán borrarse de la base de datos, pero el usuario ciudadano podrá indicar que no desea que se le vuelva a mostrar un determinado problema en la lista de pendientes.
- Filtrar problemas:
Por el contrario los usuarios miembros del grupo de mantenimiento, podrán indicar si desean que se les muestren todos los problemas, o solo los no resueltos y filtrar los problemas mostrados por fecha de envío.
- Autenticarse como miembro del equipo de mantenimiento:
Un usuario podrá introducir las credenciales que le identifican como miembro del equipo de mantenimiento. Estas credenciales se almacenaran en el dispositivo para que no tenga que volver a autenticarse cada vez que inicia.
- Cambiar el estado de un problema:
Un usuario miembro del equipo de mantenimiento podrá cambiar el estado de un problema entre: enviado, en curso o resuelto.

- Terminar sesión:
Un usuario miembro del equipo de mantenimiento que se hubiera autenticado anteriormente en el sistema, podrá borrar sus datos de credenciales, de manera que se le soliciten la próxima vez que desee trabajar como usuario de mantenimiento.

3. Diseño conceptual

3.1 Escenarios de uso

A continuación se muestran una serie de escenarios de uso de la aplicación.

Escenario 1	
Perfil:	Ciudadano.
Contexto:	Vida cotidiana, fuera del hogar.
Objetivo:	Comunicar un problema urbano al ayuntamiento.
Tareas:	Comunicar un problema.
Necesidad de información:	Localización del problema.
Funcionalidades:	Conexión con GoogleMaps, cámara de fotos, GPS opcional.
Desarrollo de las tareas:	El usuario toma una fotografía del problema, indica su posición en el mapa del municipio, da una breve descripción del mismo y manda los datos.

María está paseando por la tarde con sus hijos y al llegar al parque donde pretendía que jugaran, observa que una de las farolas no funciona. Saca su móvil del bolsillo, realiza una fotografía de la farola, indica la posición de la misma en el mapa que le muestra el móvil, escribe la descripción “farola no funciona”, y manda el aviso al ayuntamiento.

Escenario 2	
Perfil:	Municipal.
Contexto:	Entorno laboral.
Objetivo:	Solucionar posibles problemas.

Tareas:	Autenticarse como miembro del equipo de mantenimiento. Consultar un problema. Cambiar estado de un problema.
Necesidad de información:	Fotografía, posición del problema, descripción.
Funcionalidades:	GoogleMaps. Obtener datos del problema.
Desarrollo de las tareas:	El usuario se autentica y revisa los datos del problema que han enviado, fotografía, posición y descripción. Valora su arreglo y de ser factible lo marca como “en curso”.

Francisco comienza su jornada laboral como responsable del equipo de mantenimiento del municipio. Abre su aplicación móvil para consultar posibles problemas comunicados por los ciudadanos. En la lista le aparece un problema sobre una farola que no funciona. Revisa la fotografía, identifica el tipo de farola y consulta su ubicación en el mapa de la aplicación. Valora que el arreglo puede realizarse (prevé que sea simplemente un cambio de bombilla) y le comunica los datos al electricista del equipo. Cambia el estado del problema a “en curso”. Pasadas unas horas, el electricista le comunica que el problema está resuelto. Así pues, vuelve a la aplicación móvil y cambia el estado del problema a “resuelto”.

Escenario 3	
Perfil:	Ciudadano.
Contexto:	Vida cotidiana.
Objetivo:	Comprobar si puede salir a pasear con sus hijos.
Tareas:	Comprobar el estado de un problema. Eliminar un problema de la lista.
Necesidad de información:	Datos del problema.
Funcionalidades:	Obtener datos del problema.
Desarrollo de las tareas:	El usuario revisa el estado de un problema que había comunicado previamente.

María se dispone a salir al parque con sus hijos, pero antes de decidirse, recuerda que informó hace unos días sobre un problema con la iluminación del parque. Abre su aplicación móvil, para ver el estado del problema que envió. Comprueba que el problema aparece como “resuelto”. Elimina el problema de su lista de problemas, y se dispone a salir.

3.2 Diagrama de Flujo

A continuación se muestra la estructura de la aplicación mediante un diagrama de flujo.

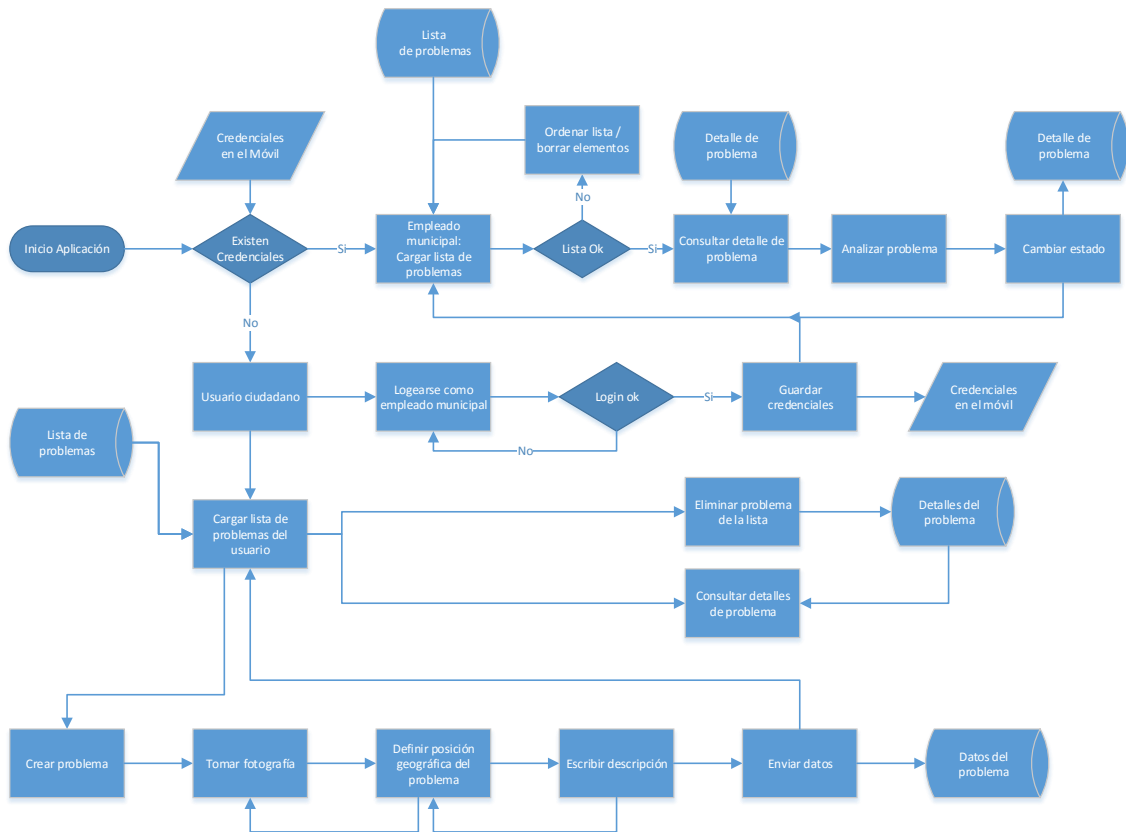


Ilustración 2 - Diagrama de flujo de la aplicación

4. Prototipado

En este apartado se va a presentar un prototipo horizontal de la aplicación, mostrando cada pantalla y las funcionalidades dentro de las mismas.

4.1 Zona de usuario ciudadano

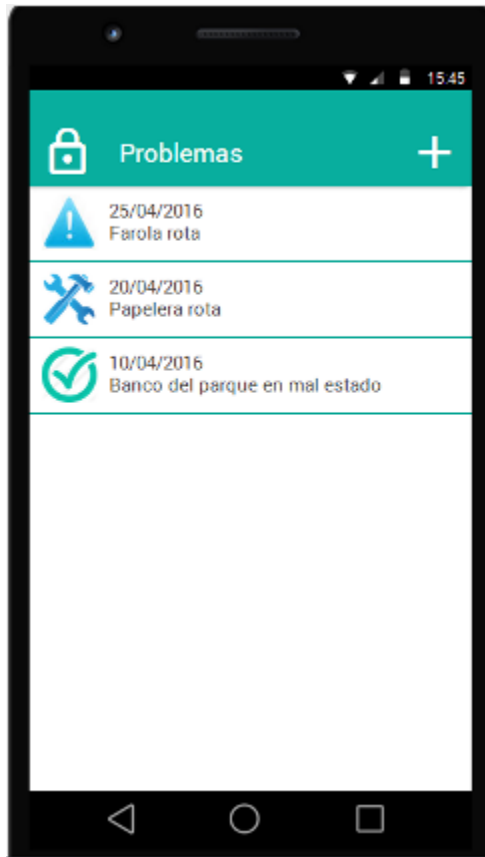


Ilustración 3 - Pantalla inicial ciudadano

Un usuario ciudadano (y uno municipal sin autenticar) comenzará en esta pantalla.

Aquí podrá ver los problemas municipales de los que haya informado y su estado actual:

▲ Enviado ✂ En curso ✅ Resuelto

Podrá pulsar en un problema para pasar a la pantalla de detalle de problema.

Una pulsación larga en un problema, le permitirá eliminarlo de su lista de problemas (aunque no del sistema).

Podrá añadir un problema nuevo pulsando en el icono "+" arriba a la derecha.

De querer identificarse como usuario municipal, podrá pulsar el icono del candado para hacerlo.



Ilustración 4 - Pantalla detalle ciudadano

En esta pantalla el usuario ciudadano podrá ver los detalles del problema que ha seleccionado.

También podrá ver el momento en el que el problema ha ido cambiando de estado.

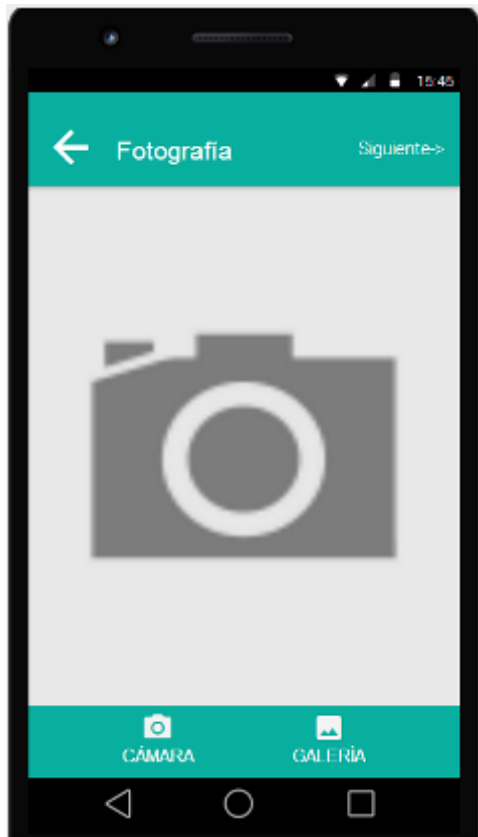


Ilustración 5 - Pantalla creación problema fotografía

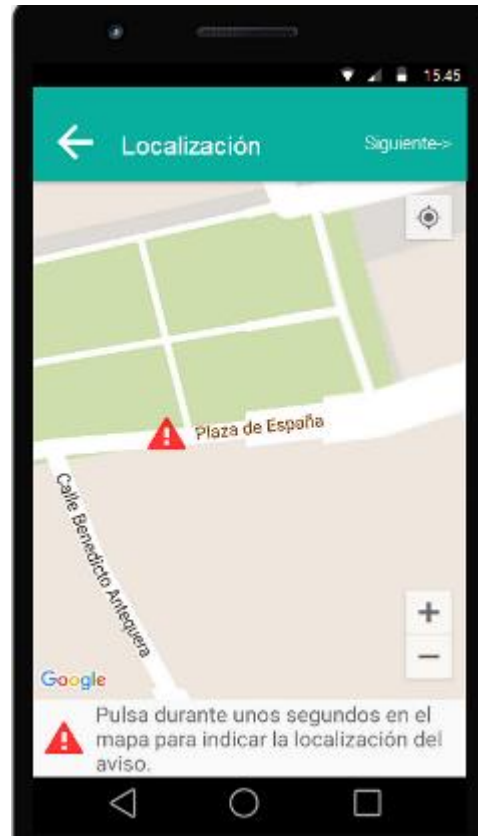


Ilustración 6 - Pantalla creación problema localización

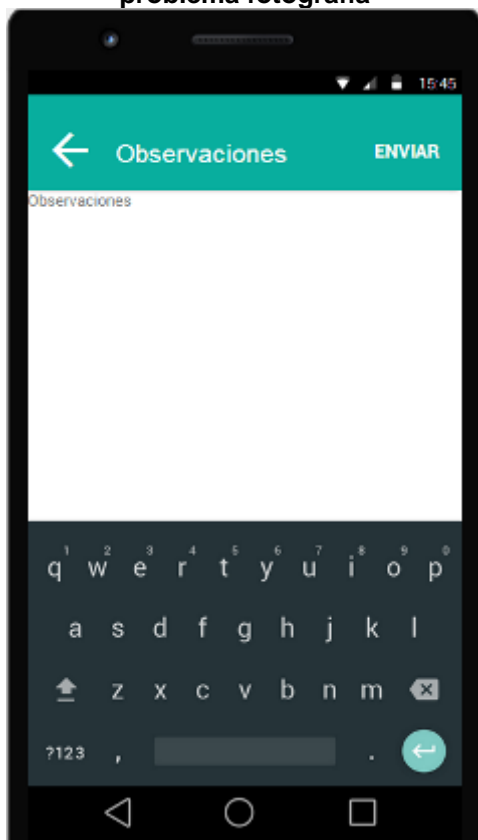


Ilustración 7 - Pantalla creación problema descripción

Esta es la sucesión de pantallas donde el usuario podrá introducir los datos de un problema y enviarlos.

Primero, proporcionaría una imagen del problema, bien tomando una fotografía o bien eligiendo alguna de su galería.

Luego indicaría la localización del problema navegando por el mapa del municipio.

Y por último escribirá una descripción del problema y enviará los datos.

4.2 Zona de usuario municipal



En esta pantalla se podrá logear el usuario municipal.

Ilustración 8 - Pantalla municipal login

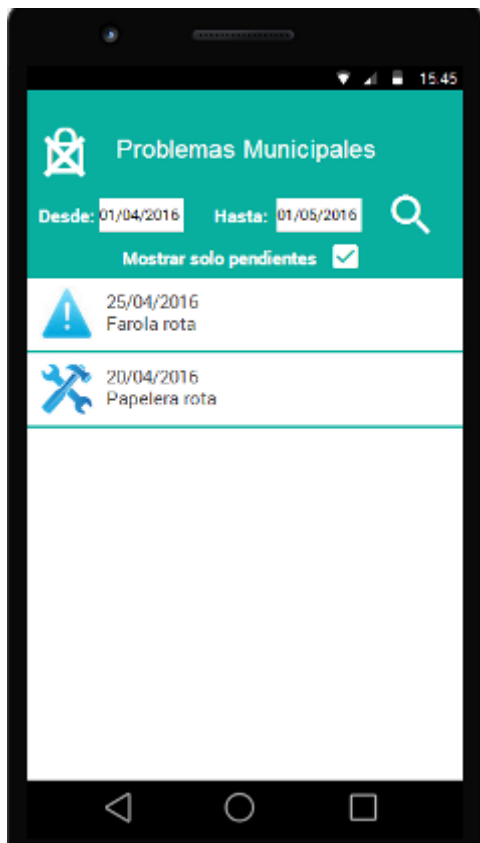


Ilustración 9 - Pantalla municipal principal

Esta es la pantalla principal del usuario municipal.

Podrá ver una lista de los problemas enviados, y filtrarla según le convenga.

Seleccionando un problema, pasará a su pantalla de detalle.

Pulsando en el icono del candado tachado, podrá cerrar su sesión como usuario municipal.

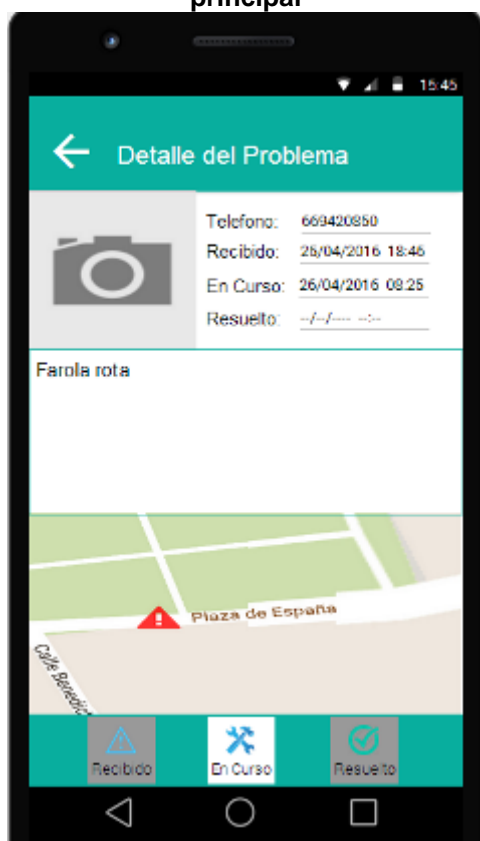


Ilustración 10 - Pantalla municipal detalle

En esta pantalla el usuario municipal podrá consultar el detalle del problema enviado.

Además será desde donde podrá cambiar el estado del problema.

5. Evaluación

La tarea de evaluación se realiza para identificar posibles problemas de usabilidad y asegurar la calidad del producto.

El objetivo principal es lograr que el usuario final pueda interactuar intuitivamente con la aplicación, sin necesidad de contar con ayuda externa para completar cualquier proceso y logrando una experiencia de uso satisfactoria.

Esto es un proceso iterativo por lo que se debe realizar la evaluación de los interfaces de manera periódica hasta alcanzar un alto grado de madurez, dónde se hayan corregido las deficiencias encontradas en cada iteración.

Para realizar la evaluación usaremos los llamados Test de Usuario, que son un método de evaluación centrado en el usuario final que nos servirá para obtener las posibles mejoras que podamos realizar.

A los distintos grupos de usuarios se les pasará el diseño horizontal de la aplicación y se les pedirá que intenten realizar las tareas detectadas en el análisis de tareas:

- Comunicar un problema.
- Consultar el estado de un problema.
- Borrar un problema de la lista.
- Filtrar problemas.
- Autenticarse como miembro del equipo de mantenimiento.
- Cambiar el estado de un problema.
- Terminar sesión.

Una hayan terminado se les pedirá que respondan a las siguientes cuestiones:

- ¿Ha encontrado dificultades para completar las tareas?
- ¿Le parecen adecuados todos los menús? ¿De un vistazo es capaz de encontrar la forma de llegar a lo que busca?
- ¿En todo momento sabe cómo volver a la pantalla anterior y a la pantalla principal?
- ¿Se ha sentido desorientado en algún parte de la aplicación?
- ¿De qué elementos carece el prototipo actual?
- ¿Has encontrado algún defecto en el uso del prototipo?
- ¿Cómo se podría mejorar la herramienta?

6. Casos de Uso

6.1 Diagrama UML

A continuación presentamos los diagramas UML con la definición de actores y flujo:

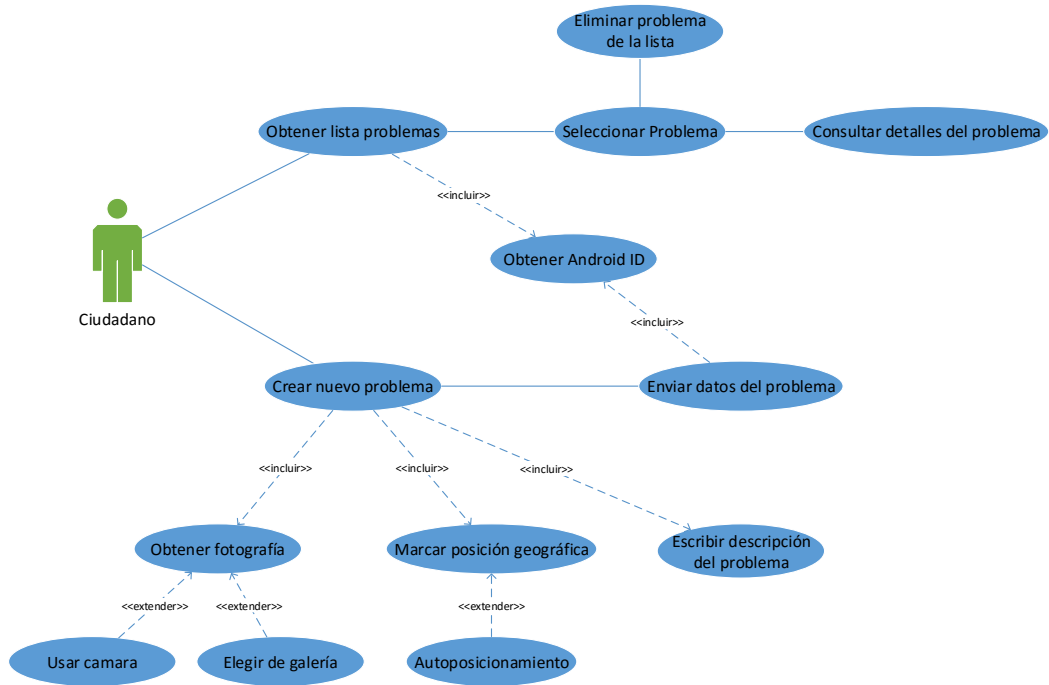


Ilustración 11 - Diagrama UML ciudadano

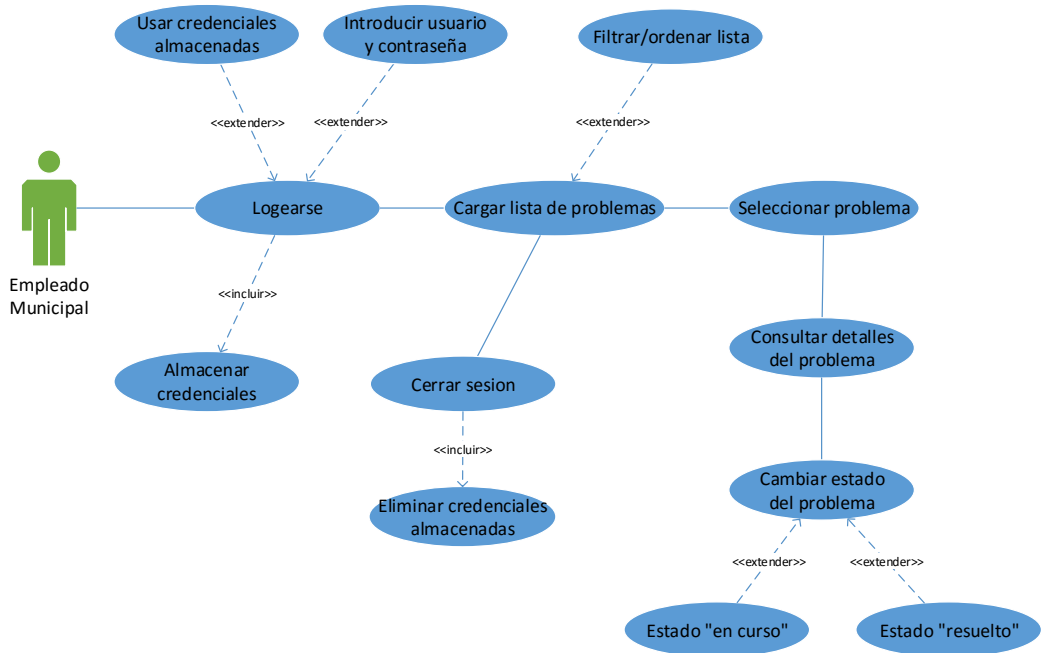


Ilustración 12 - Diagrama UML municipal

6.2 Casos de Uso

Caso Uso 1

Nombre:	Obtener lista de problemas
Actor:	Ciudadano
Precondiciones:	Ninguna
Flujo:	<ul style="list-style-type: none">- El sistema obtiene el android id del dispositivo.- El sistema realiza una petición de los datos de los problemas asociados a ese id, y marcados como visibles.
Postcondiciones:	Al usuario se le muestra una lista con los problemas que ha enviado este usuario.

Caso Uso 2

Nombre:	Eliminar problema de la lista
Actor:	Ciudadano
Precondiciones:	Haber elegido un problema de la lista de problemas. <ul style="list-style-type: none">- El sistema manda una petición para actualizar los datos del problema seleccionado.
Flujo:	<ul style="list-style-type: none">- Se actualiza la lista de problemas.
Postcondiciones:	Al usuario ya no le aparece ese problema en su lista.

Caso Uso 3

Nombre:	Consultar detalles del problema
Actor:	Ciudadano / Municipal
Precondiciones:	Haber elegido un problema de la lista de problemas. <ul style="list-style-type: none">- El sistema manda una petición para obtener los datos de detalle del problema seleccionado.
Flujo:	<ul style="list-style-type: none">- Se presenta la pantalla de detalle del problema.
Postcondiciones:	Al usuario le aparece una pantalla con todos los datos del problema seleccionado.

Caso Uso 4

Nombre:	Crear problema
Actor:	Ciudadano
Precondiciones:	Haber localizado un problema en el municipio. <ul style="list-style-type: none">- El usuario realiza una fotografía o bien selecciona una de la galería para ilustrar el problema.- El sistema redimensiona la fotografía para que no ocupe más de 100Kb de tamaño.- El sistema intenta localizar la posición geográfica actual del usuario.
Flujo:	<ul style="list-style-type: none">- El usuario marca la posición geográfica del problema en el mapa (localizado en su posición si es posible) que le muestra el sistema.- El usuario escribe una descripción del problema.- El sistema envía todos estos datos del problema asociados al android id del dispositivo del usuario.
Postcondiciones:	Al usuario le aparece este nuevo problema, como enviado, en su lista de problemas.

Caso Uso 5

Nombre:	Logearse
Actor:	Municipal
Precondiciones:	No haberse logeado anteriormente y por lo tanto no disponer de credenciales almacenadas en el dispositivo. <ul style="list-style-type: none">- El usuario introduce su nombre de usuario y su contraseña.- El sistema comprueba que las credenciales sean correctas.
Flujo:	<ul style="list-style-type: none">- Si son correctas, las almacena en el dispositivo e identifica al usuario como municipal.- Si no son correctas, muestra un error y pide reintentar.
Postcondiciones:	El usuario está identificado como usuario municipal en el sistema.

Caso Uso 6

Nombre:	Cargar lista de problemas
Actor:	Municipal
Precondiciones:	Estar identificado como un usuario municipal.
Flujo:	<ul style="list-style-type: none">- El usuario indica el rango de fechas que quiere ver.- El usuario indica si desea ver solo los problemas pendientes.- El sistema carga la lista correspondiente a los problemas en el rango de fechas especificado.
Postcondiciones:	El usuario puede ver una lista filtrada con los problemas del municipio.

Caso Uso 7

Nombre:	Cambiar estado del problema
Actor:	Municipal
Precondiciones:	Estar identificado como un usuario municipal. Estar en la pantalla de detalle del problema.
Flujo:	<ul style="list-style-type: none">- El usuario marca el nuevo estado del problema.- Se sistema actualiza los datos del problema.
Postcondiciones:	Al usuario le aparece el problema con el estado actualizado.

Caso Uso 8

Nombre:	Cerrar sesión
Actor:	Municipal
Precondiciones:	Estar identificado como un usuario municipal.
Flujo:	<ul style="list-style-type: none">- El usuario pulsa el botón de cerrar sesión.- Se sistema elimina las credenciales del dispositivo.
Postcondiciones:	El sistema le pedirá que introduzca sus credenciales la próxima vez que quiera identificarse como usuario municipal.

7. Diseño de la Arquitectura

La arquitectura de la aplicación estará basada en uno de los patrones de diseño de software más utilizados: el modelo vista controlador (MVC).

Este modelo separa los datos y la lógica de negocio de la interfaz de usuario. Con esto logramos desacoplar partes fundamentales de la aplicación, con lo que se facilita las tareas de desarrollo y mantenimiento.

- La vista es la interfaz con la que va a interactuar el usuario.
- El modelo es la representación de la información manejada por la aplicación.
- El controlador hace de intermediario entre la vista y el modelo, gestiona peticiones del usuario (responde a eventos). Envía y recibe información de la vista, la trata y se comunica con el modelo.

En el caso de nuestra aplicación, se usará una adaptación de este patrón de diseño orientada al desarrollo de aplicaciones para Android.

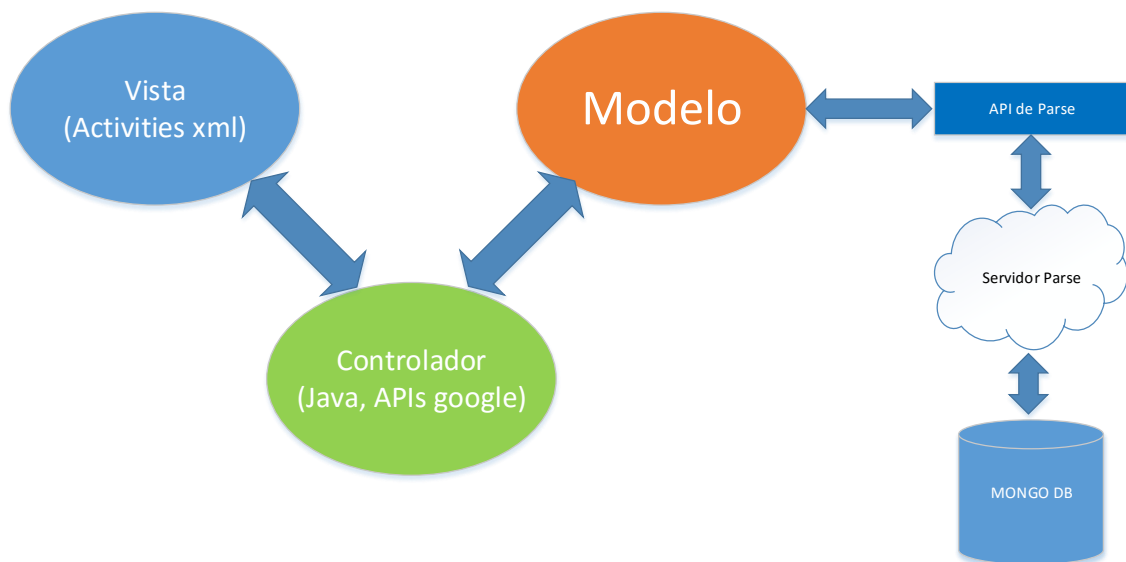


Ilustración 13 - Arquitectura de la aplicación

7.1 Vista

Definiremos la estructura de las pantallas mediante ficheros xml.

7.2 Controlador

Serán las clases java que extiendan las Activities de android y las API's de Google. Controlarán los eventos y acciones del usuario y comunicarán el modelo con la vista.

7.3 Modelo

Es la representación de los datos que usará la aplicación. En este caso se usarán clases que extienden del API de Parse. Mediante el uso de esta API, se pueden construir directamente las clases modelo que se necesiten, añadiendo los atributos que deseemos y la API creará automáticamente las tablas necesarias en el servidor para la persistencia de los datos. Se podrán igualmente, realizar consultas, inserciones, modificaciones y borrados, directamente sobre las clases modelo.

Estos son los atributos de las dos clases modelo que se usarán en la aplicación:

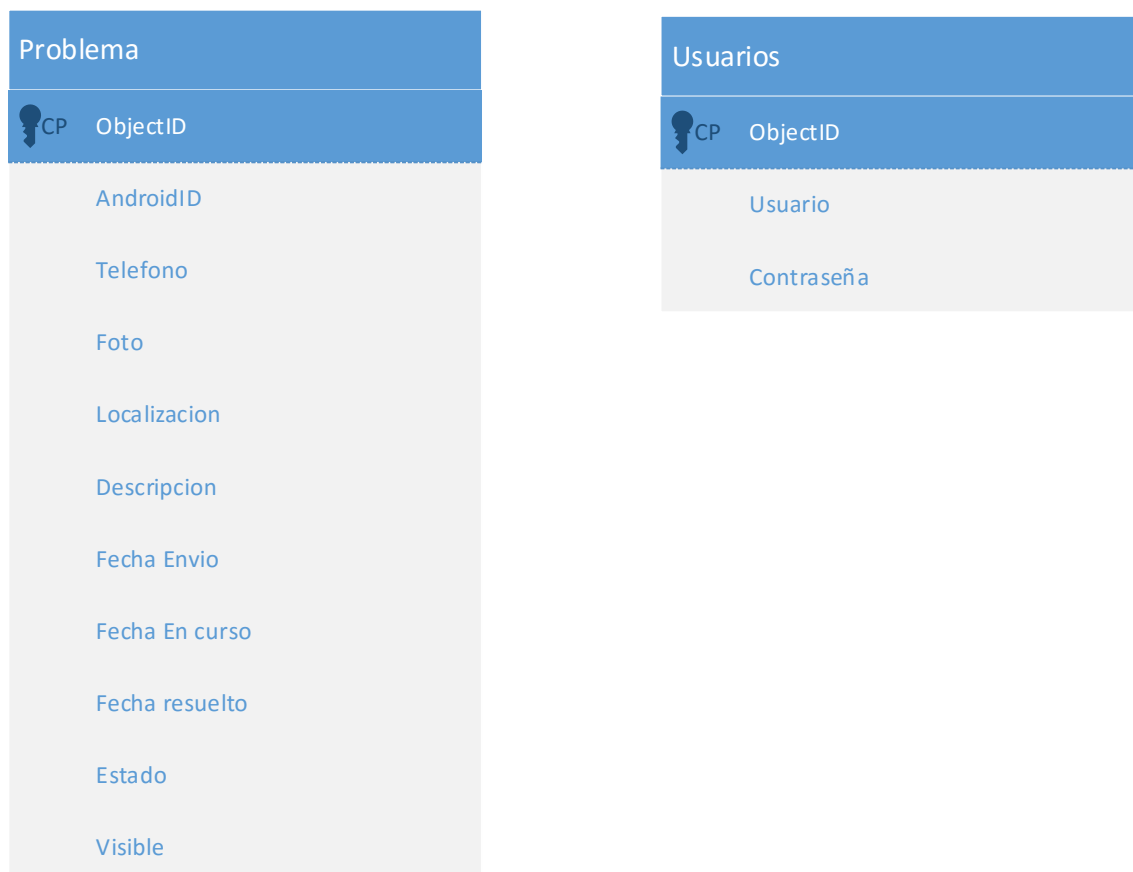


Ilustración 14 - Diagrama de estructura de datos

7.4 Persistencia de los datos

Como se ha comentado de la persistencia se encargará la API y el servidor backend de Parse. Esta solución se encontraba de manera online hasta hace poco.

Ahora, la empresa encargada del servicio de Parse, ha anunciado su cierre, pero ha liberado el código fuente de su servidor backend. De tal manera que lo que se hará será montar una copia propia de ese servidor para usar con nuestra aplicación.

Para ello necesitaremos una máquina fija, donde ejecutaremos el código servidor de Parse, que está en Node.js y tendremos una instancia de base de datos MongoDB.

8. Implementación

Como entorno de desarrollo se ha utilizado Android Studio. Se trata del entorno oficial lanzado para desarrollo en Android, basado en IntelliJ Idea.

La mínima versión de SDK de Android para que funcione la aplicación es: API 15: Android 4.0.3 (IceCreamSandwich).

8.1 Servidor Backend

Como se ha comentado anteriormente, se ha querido utilizar la API y el servidor Backend de código abierto proporcionados por [Parse](#). Para ello es necesario la instalación de una instancia de base de datos MongoDB y del propio servidor de Parse, en una máquina que disponga de conexión a internet y tenga instaladas las librerías de Node.js.

Esto se puede hacer fácilmente siguiendo las instrucciones de su [página web](#), o el Anexo 1.

Pese a que el servidor se ha instalado y funciona correctamente, en un portátil local, he tenido muchos problemas para realizar la comunicación entre el dispositivo Android y el propio servidor, lo que ha supuesto un retraso en la implementación de los objetivos principales del proyecto.

Esto ha resultado en que se han implementado todas las funcionalidades menos la de persistencia de credenciales en el dispositivo, es una pena, pero también cabe decir que la ausencia de esta funcionalidad no es crítica para el funcionamiento de la aplicación.

8.2 Pruebas

- Pruebas unitarias:

No se han realizado pruebas unitarias debido a dos factores que se han tenido en cuenta:

1. Debido al retraso en la implementación no he tenido mucho tiempo para ello.

2. He considerado que en este caso este tipo de pruebas no era crítico para la aplicación. Ya que los módulos a probar o hacen uso de Intents que requieren la interacción con el usuario o utilizan las librerías de Parse, que considero sobradamente probadas por sus creadores.

- **Pruebas de integración:**

Por otro lado sí que se han realizado pruebas de integración, probando toda la aplicación en su conjunto y el funcionamiento de las distintas funcionalidades.

Las pruebas se han realizado tanto en un emulador con la API 23 de Android como en un Smartphone Samsung Galaxy S2 API 16, pasando las pruebas satisfactoriamente.

Funcionalidades probadas:

1. Enviar nuevo problema. (foto, posición y texto asociado)
2. Mostrar lista de problemas solo del ciudadano.
3. Mostrar detalles del problema.
4. Mostrar lista de todos los problemas, resueltos, no resueltos y comprendidos entre un rango de fechas.
5. Cambiar el estado del problema.

9. Conclusiones

Al término de este Trabajo Fin de Master, puedo concluir que el resultado ha sido satisfactorio a nivel personal porque este proyecto fue una de las razones que me impulsaron a la realización de este Master.

La planificación se ha ajustado a la realidad en todas las fases menos en la fase de implementación, la cual ha sufrido un importante retraso debido en su mayor parte a la adaptación al nuevo sistema de trabajo del API de Parse.

Los requisitos iniciales se han cumplido casi en su totalidad, habiendo alguna funcionalidad que no se ha implementado, pero que no resulta crítica para el funcionamiento normal de la aplicación.

Como posibles ampliaciones a este trabajo, propongo las siguientes:

- Desarrollar la funcionalidad de persistencia de las credenciales de usuario que ha quedado pendiente.
- Desarrollar un sitio web que permita ver los avisos por parte del personal municipal y permita obtener estadísticas de uso, tiempos medios de resolución de los avisos, etc.
- Dividir los avisos por tipos y que existan distintos perfiles de trabajadores municipales que accedan solo a su perfil de avisos.

Como conclusión final, considero que la aplicación reúne todas las características de programación móvil que hemos estudiado en este Master (uso de Intents, Mapas, Fotografías, Listas personalizadas, Gestión de Usuarios, Comunicación con servidor Backend), demostrando en su conjunto el total aprovechamiento del mismo.

10. Glosario

- **Android:** Sistema operativo basado en el núcleo Linux.
- **Android SDK (software development kit):** Kit de desarrollo para Android, que contiene las herramientas básicas necesarias para desarrollar en Android, con independencia del IDE utilizado.
- **Android Studio:** Entorno de desarrollo integrado (IDE) para la plataforma Android.
- **IDE (Entorno de desarrollo integrado):** Entorno de programación.
- **MVC (Modelo Vista Controlador):** Es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.
- **MongoDB:** Es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto.
- **NodeJS:** Entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor.
- **Servidor Backend:** Es el servidor que se dedica a la parte lógica de la aplicación, principalmente de la persistencia de los datos.
- **Servidor Parse:** Servidor backend de código abierto.

11. Bibliografía

- Exposición de contenidos en vídeo.
- Redacción de textos científico-técnicos.
- Tecnología y desarrollo en dispositivos móviles.
- Presentación de documentos y elaboración de presentaciones.
- 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.
- Android Developers. <https://developer.android.com/index.html?hl=es>
06/06/2016
- Parse Android Guide. <https://parse.com/docs/android/guide>
06/06/2016

12. Anexo 1: Manual de Instalación

12.1 Instalación de la parte servidor

La aplicación funciona contra el servidor de código abierto de Parse.com, para instalarlo en una máquina local se han de seguir los siguientes pasos:

- Bajar e instalar [MongoDB community edition](#) (Ver. 3.2.6).
- Encender el servidor de MongoDB:

```
"c:\Program Files\MongoDB\Server\3.2\bin\mongod.exe"
```
- Bajar e instalar [NodeJS](#).
- Bajar e instalar [Python 2.7.11](#).
- Clonar el código del [servidor de parse](#) desde github.
- Desde la carpeta donde has clonado el servidor ejecutar:

```
npm install
```
- Y después:

```
npm install -g parse-server
```
- Desde esa misma carpeta, encender el servidor Parse, ejecutando:

```
npm start
```
- Si todo ha ido bien, el servidor se habrá encendido y estará escuchando en el puerto 1337. Para probarlo se puede ejecutar una ventana de navegador y probar la url <http://localhost:1337/test>
- Para más información se puede consultar este [tutorial de instalación](#).

12.2 Instalación de la parte cliente

Para instalar la parte cliente de la aplicación hay que asegurarse de tener la última versión del Android Studio junto con el sdk 23.

En el fichero AvisoApplication.java, cambiar la línea:

```
.server("http://192.168.1.100:1337/parse/")
```

Por:

```
.server("http://<IP del Server>:1337/parse/")
```

Compilar y general la app y ejecutarla en un emulador, teniendo en cuenta que para que se muestren los mapas de GoogleMaps hay que firmar la aplicación y puede que no se muestren fuera de la apk generada.

12.3 Probar la aplicación

Para probar la aplicación en estos momentos, se puede instalar la aplicación en un Smartphone que cumpla con las características indicadas o en un emulador compatible.

Si se desea probar la sección de usuarios municipales es necesario insertar primero un usuario en el sistema, se ha dejado un fragmento de código fuente comentado en el fichero "AvisoAplicación.java". Si se descomenta y se ejecuta la aplicación, se creará en el sistema el usuario "prueba" con password "prueba".

Para facilitar esto se ha dispuesto una instancia de base de datos en Mlab, que está disponible online y que contiene ejemplos de prueba.

Simplemente, una vez instalado el servidor de Parse, modificar el fichero de configuración "index.js" y cambiar el valor de la variable "**databaseURI**", para que quede:

```
databaseURI: databaseUri //  
'mongodb://prueba:prueba@ds011422.mlab.com:11422/avisos'
```


13. Anexo 2: Manual de Usuario

13.2 Sección de Usuario Ciudadano

La aplicación comienza mostrando la siguiente pantalla:

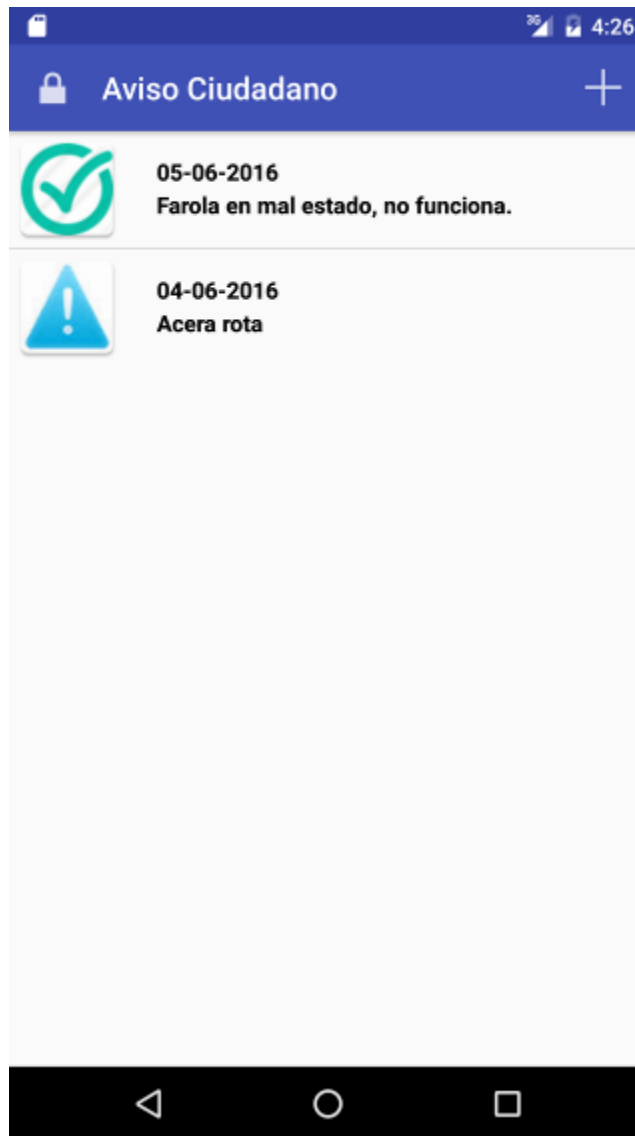





Ilustración 15 Pantalla de inicio

En ella aparecen los avisos que haya enviado el usuario, junto con la fecha de última actualización, la descripción y un icono que indica su estado.

-  Recibido.
-  En curso.
-  Resuelto.

Para ver el detalle de un aviso enviado, el usuario solo tiene que pulsar sobre él en la lista. Lo que le lleva a la siguiente pantalla:



Ilustración 16 Pantalla de detalle ciudadano

Desde la pantalla de inicio, si el usuario lo que desea es enviar un aviso nuevo, debe pulsar en el icono + arriba a la derecha. Lo que lo lleva a:



Ilustración 17 Pantalla insertar posición

Esta pantalla aparece, para ahorrar tiempo, centrada en la plaza de la localidad, para que se sitúe en su posición, simplemente se debe pulsar en el icono correspondiente del mapa. Puedes moverte por el mapa libremente, así como ampliar o disminuir el zoom. Cuando encuentres la localización deseada, mantén pulsado unos segundos y aparecerá la marca y se te permitirá desplazarte a la siguiente pantalla.



Ilustración 18 Pantalla insertar imagen

En esta pantalla se te pedirá que introduzcas una imagen del problema del que desees avisar, puedes hacer la foto en ese momento pulsando el icono de “Tomar foto”, o seleccionar una foto tomada anteriormente pulsando en “Mis fotos”, y podrás pasar a la siguiente pantalla.



Ilustración 19 Pantalla con foto

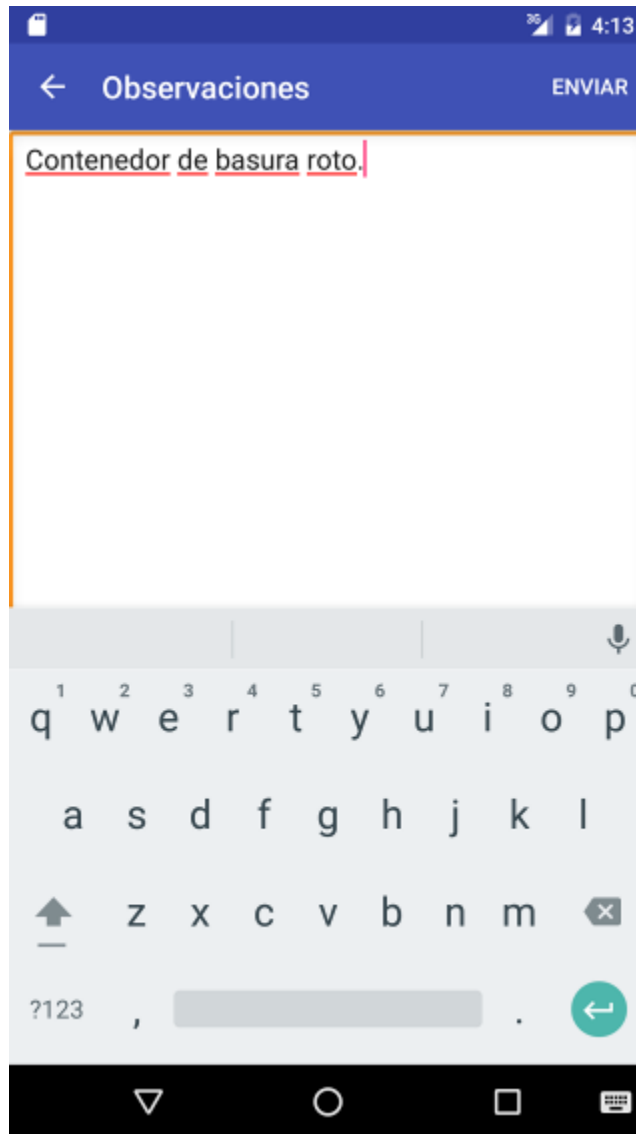


Ilustración 20 Pantalla observaciones

En esta última pantalla se te permite insertar las observaciones que desees aportar al problema, y enviarlo. Una vez enviado, aparecerá en tu lista de problemas con estado “enviado”.

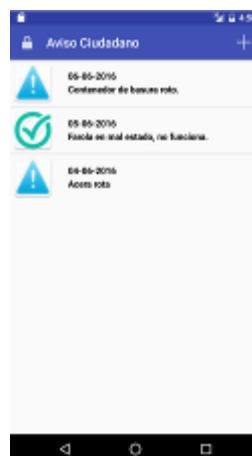


Ilustración 21 Problema añadido

Si el usuario desea eliminar un problema de su lista, simplemente deberá hacer una pulsación larga sobre él.

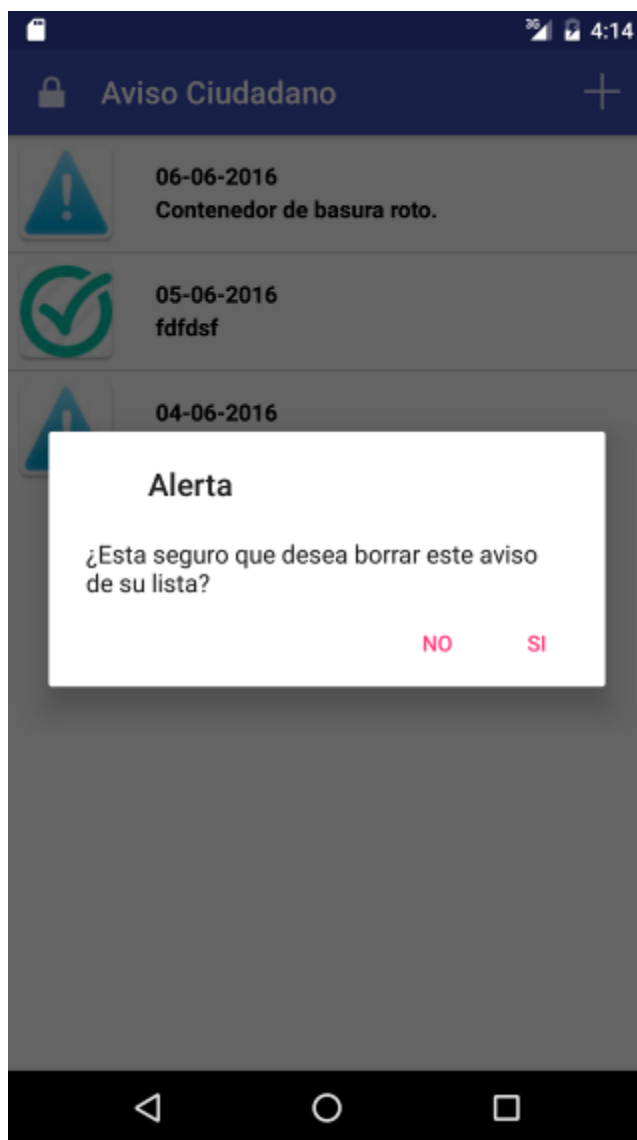


Ilustración 22 Eliminar problema

El problema no será borrado de la base de datos, simplemente no se le mostrará al usuario.

13.3 Sección de Usuario Municipal

Para entrar en esta sección un usuario municipal deberá pulsar en el icono del candado arriba en la parte izquierda de la pantalla principal, e identificarse en la pantalla de login.



Ilustración 23 Pantalla de login

Una vez identificado le aparecerá una lista con todos los problemas no resueltos con fecha del último mes, para que pueda consultarlos.

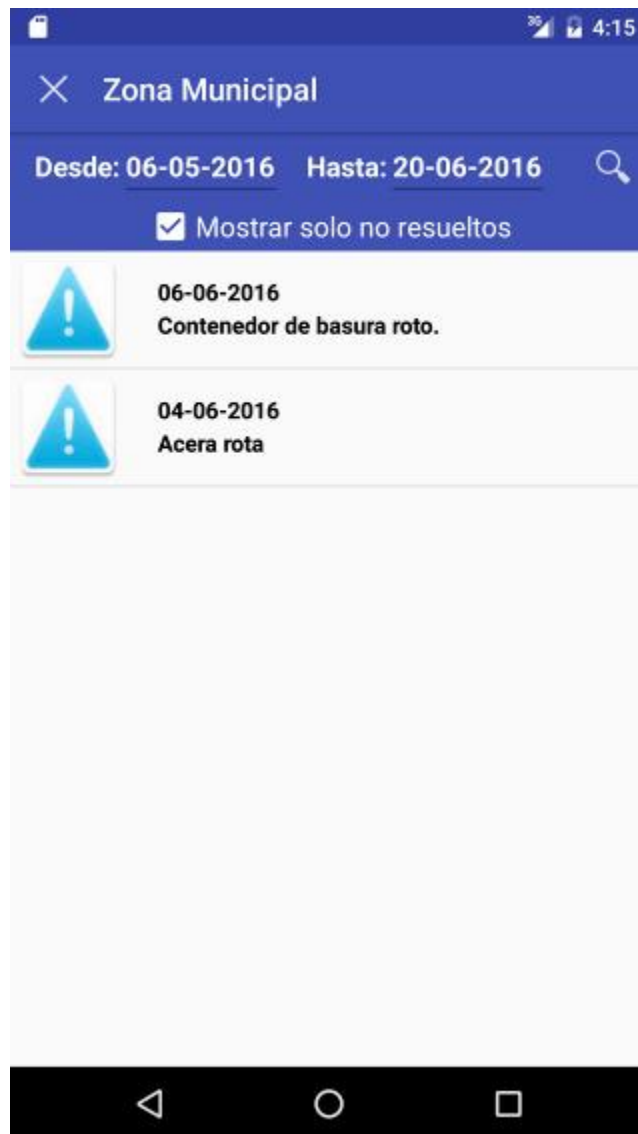


Ilustración 24 Pantalla lista municipal

En esta pantalla se pueden modificar los criterios de búsqueda, cambiando las fechas o permitiendo que se muestren también los problemas ya resueltos.

Pulsando sobre un problema de la lista se puede acceder a sus detalles y modificar su estado.



Ilustración 25 Pantalla detalle municipal

Pulsando en los botones inferiores de la pantalla se puede cambiar el estado del problema según el criterio del trabajador.