

AGENDA COMERCIAL

TRABAJO FINAL DE CARRERA

Desarrollo de aplicaciones móviles

José Antonio Bernal Marcos
Ingeniería Técnica en Informática de Gestión

Consultores:
Marc Domingo Prieto / Jordi Almirall Almirall López

*A Daniel y Ángel, por todos esos parques perdidos.
Os prometo recuperarlos.*

*A todos los que me habeis acompañado en este viaje.
Por darmelo todo a cambio de nada.
Sin vosotros no estaría aquí.*

¡¡Gracias!!

*“Ten siempre a Itaca en tu mente.
Llegar allí es tu destino.
Mas no apresures nunca el viaje.
Mejor que dure muchos años
y atracar, viejo ya, en la isla,
enriquecido de cuanto ganaste en el camino
sin aguantar a que Itaca te enriquezca.”*

Índice de contenido

• 1	Introducción.....	4
1.1	Objetivos.....	4
1.2	Materiales a entregar.....	4
1.3	Descripción de tareas y subtareas.....	5
•	5
1.4	Recursos del sistema.....	6
•	7
1.5	Planificación temporal.....	7
• 2	Requisitos funcionales.....	8
• 3	Análisis y diseño del sistema.....	9
3.1	Casos de uso.....	9
	Contactos.....	9
	Citas.....	12
	Notas.....	15
	Tipos de contacto.....	19
	Imágenes.....	22
	Geolocalización.....	25
3.2	Diagrama de clases.....	26
3.3	Diagrama BD Entidad-Relación.....	27
• 4	Interfaz de usuario.....	28
4.1	Análisis.....	28
	4.1.1 Diagrama de clases.....	28
	4.1.2 Contexto de uso.....	29
	4.1.3 Análisis de tareas.....	29
4.2	Diseño conceptual.....	34
	4.2.1 Escenarios de uso.....	34
	Escenario para el uso de la gestión de contactos.....	34
	Escenario para el uso de la gestión de tipos de contactos.....	34
	Escenario para el uso de la gestión de citas.....	35
	Escenario para el uso de la gestión de notas.....	35
	Escenario para el uso de la captura de imágenes.....	35
	Escenario para el uso de la funcionalidad de geolocalización.....	36
	4.2.2 Flujos de interacción.....	37
4.3	Prototipado.....	40
4.4	Evaluación final.....	58
• 5	Implementación.....	60
5.1	Introducción.....	60
5.2	Creación del proyecto.....	60
5.3	Estructura del proyecto.....	60
5.4	Componentes.....	61
	5.4.1 Menú principal.....	61
	5.4.2 Componente Contactos.....	61
	5.4.3 Componente Tipos de Contacto.....	62
	5.4.4 Componente Citas.....	62
	5.4.5 Componente Notas.....	63
	5.4.6 Componente Imágenes.....	64
	5.4.7 Componente Geolocalización.....	65

- 6 Posibles mejoras.....67
- 7 Conclusiones.....68
- 8 Bibliografía.....69

1 **Introducción**

1.1 **Objetivos**

El objetivo del presente TFC (Trabajo Final de Carrera) es la planificación, diseño y desarrollo de una aplicación software orientada a dispositivos móviles multiplataforma.

La aplicación gestionará una cartera de clientes y contactos de un representante comercial. Deberá permitir añadir, modificar y eliminar contactos, gestionar la agenda de citas y gestionar las notas personales que añada el usuario, almacenando la información en una base de datos de forma que los datos sean persistentes. También deberá ofrecer una serie de funcionalidades avanzadas y servicios propios de los dispositivos móviles, tales como hacer uso de la cámara de fotos para obtener imágenes o del dispositivo GPS para geolocalización.

El desarrollo del proyecto constará de cuatro partes claramente diferenciadas.

- Diseño de la interfaz de usuario utilizando componentes propios de la plataforma móvil tales como listas, formularios, barras de desplazamiento, ventanas modales, etc... También se hará uso de los eventos y disparadores asociados de los que disponemos en un terminal con pantalla táctil como por ejemplo arrastrar, deslizar, pulsar, etc..
La interfaz la realizaremos siguiendo una metodología DCU (diseño centrado en el usuario)
- Diseño e implementación de la base de datos. Utilizaremos SQLite para dotar de persistencia a los datos de la aplicación. En este proceso se definirán las distintas tablas a utilizar y su relación. Como paso final, se generará un script con la creación de todos los elementos necesarios de la base de datos.
- Codificación de la lógica de la aplicación. Una vez diseñada la interfaz del usuario e implementada la persistencia de datos, será necesario programar la lógica de la aplicación. En este apartado deberemos utilizar los datos que disponemos en BD y los servicios nativos que nos ofrece el sistema (cámara, acelerómetro, GPS...) para que, en función de las entradas que recibimos del usuario al interactuar con la interfaz, el programa sea capaz de llevar a cabo las tareas requeridas de forma eficaz.
- Fase de pruebas. Con la aplicación concluida, será imprescindible llevar a cabo toda una batería de pruebas de forma que podamos detectar posibles fallos en el funcionamiento del producto y seamos capaces de corregirlos.

El proyecto deberá incluir, además, la documentación técnica generada a lo largo de todo el proceso.

1.2 **Materiales a entregar**

A lo largo de la realización del proyecto se han ido generando una serie de materiales disponibles para entregar.

- Plan de trabajo. Documento en el que se expone de forma breve y concreta los objetivos del proyecto, sus principales características y la planificación temporal detallada para su realización.
- Prototipado del diseño de la interfaz de usuario siguiendo la metodología DCU.
- Documentación técnica del proyecto con las funcionalidades de la aplicación, diagramas utilizados para la realización del proyecto (casos de uso, flujo, estados...), perfiles de usuarios, etc...
- El producto final, consistente en el código de la aplicación.
- Documentación propia de la fase de pruebas
- Memoria del proyecto que sintetizará todo el trabajo realizado y que tiene como principal objetivo demostrar que se han asimilado los objetivos propuestos en la asignatura. Contendrá la información necesaria para comprender el problema planteado, la metodología utilizada para su resolución y la solución adoptada para resolver el problema planteado.
- Presentación virtual en la que exponer visualmente el trabajo realizado y los resultados obtenidos.

1.3 Descripción de tareas y subtareas

A continuación se detallan las tareas principales y su división en subtareas.

1. **Plan de trabajo.** Primera aproximación, breve y concisa, al proyecto, indicando objetivos, tareas a realizar y una propuesta de planificación temporal.
2. **Análisis y diseño.**
 1. Fase de análisis: toma de requisitos del proyecto donde definir sus funcionalidades.
 2. Fase de diseño: diseño técnico de la aplicación. Generación de los diagramas necesarios para su comprensión y desarrollo (casos de uso, clases, base de datos)
3. **Interfaz gráfica.** Utilizaremos una metodología de Diseño Centrado en el Usuario (DCU) que incluye las siguientes subtareas.
 1. Usuarios y contextos de uso.
 2. Diseño conceptual.
 3. Prototipado.
 4. Evaluación final.
4. **Desarrollo aplicación.** Codificación de los distintos módulos que compondrán la aplicación. A falta de mayor detalle en las funcionalidades de cada uno de los módulos y que se deberán recopilar en la fase de toma de requisitos, estos son los módulos y sus principales características.
 1. Módulo gestión de contactos. El usuario deberá poder añadir, eliminar y modificar sus contactos personales, indicando tipo el tipo de contacto al que pertenece (cliente, proveedor, ...) También se deberá crear la gestión de los tipos de contactos.
 2. Módulo gestión de tipos de contacto. El usuario será capaz de definir diferentes tipos de contacto para mayor organización de la agenda. Ejemplo: agentes comerciales, administración, contactos personales....
 3. Módulo de notas. El usuario deberá poder añadir, eliminar y modificar notas personales. Estas notas se podrán asignar opcionalmente a cualquier contacto.
 4. Módulo agenda – gestión de citas. El usuario deberá poder añadir, eliminar y modificar sus citas de la agenda personal.
 5. Módulo captura de imágenes. El usuario deberá poder capturar imágenes por medio de la cámara fotográfica del terminal y almacenarlas en memoria.
 6. Módulo geolocalización. El usuario deberá poder realizar tareas de geolocalización por medio del dispositivo GPS del terminal y la API de Google Maps.
5. **Fase de pruebas.** En esta fase se someterá al producto final a una batería completa de pruebas con la finalidad de detectar posibles errores y/o funcionamientos no esperados. La fase de pruebas nos permitirá poder corregir errores en la aplicación antes de dar por finalizada la fase de desarrollo.
6. **Memoria.** Documento en el que sintetizar todo el trabajo realizado a lo largo del proyecto.
7. **Presentación TFC.** Documento multimedia en el que presentar de forma visual el TFC.

1.4 Recursos del sistema

Para llevar a cabo el proyecto fueron necesarios una serie de recursos, tanto hardware como software, que se detallan a continuación.

Hardware

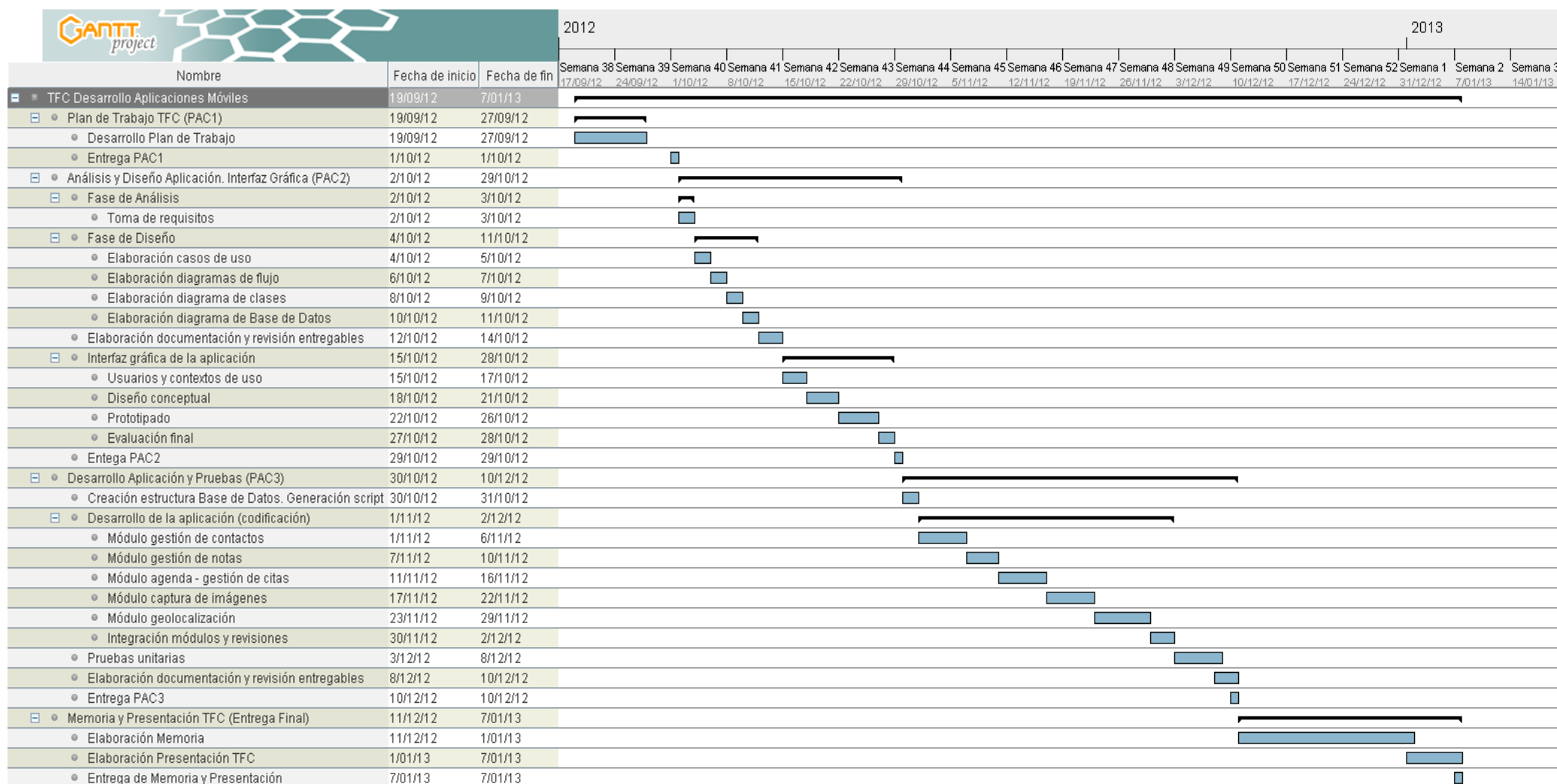
- En principio no es necesario un equipo con requisitos especiales o características específicas, dado que ninguna de las fases del proyecto necesita grandes recursos en cuanto a procesador o memoria RAM. En concreto, el equipo elegido para realizar el trabajo, fue un portátil con procesador Atom 1,66Ghz y 1GB de RAM con Linux como sistema operativo.
- Smartphone. Aunque al ser código HTML5 se puede llevar el seguimiento del proyecto desde un navegador compatible como Chrome, utilizamos un terminal de última generación para comprobar su correcto funcionamiento en este tipo de dispositivos. El terminal utilizado fue un LG Optimus 2X con Android 2,3,7

Software

- Entorno de desarrollo integrado: Eclipse Classic 4,2.
- Gestor de base de datos: SQLite Manager.
- Desarrollo de la aplicación: utilizamos PhoneGap para codificar la lógica, JQuery Mobile para generar la interfaz de usuario y SQLite para la persistencia de datos.
- Paquete ofimático para la realización de la documentación y la memoria: LibreOffice.

1.5 Planificación temporal

El siguiente diagrama de Gantt muestra la planificación llevada a cabo para la realización del TFC. La fecha de inicio es el 19/09/2012 y la de entrega final el 7/1/2013, con entregas parciales los días 1/10/2012 (PAC1), 29/10/2012 (PAC2) y 10/12/2013 (PAC3)



2 Requisitos funcionales

La aplicación final deberá cumplir con los requisitos y funcionalidades que se detallan a continuación.

- Contactos:
Esta funcionalidad permitirá al usuario gestionar totalmente su agenda comercial de contactos. Para ello, la aplicación debe permitir añadir, eliminar y modificar los detalles de los ya existentes en el sistema. También debe proporcionar un listado completo con los contactos guardados en la agenda.
Todo contacto debe tener unos campos mínimos de información obligatorios: nombre, apellidos, teléfono y tipo de contacto. El resto de datos serán opcionales.
- Tipos de contactos:
Los contactos podrán ser de varios tipos, para facilitar su clasificación y organización por parte del usuario. Por lo tanto se debe poder añadir, eliminar y modificar los tipos de contactos para tener una completa gestión de los mismos. También se debe proporcionar un listado completo de los tipos que ya existen en el sistema. Todos los contactos de la agenda deben de ser forzosamente de uno de los tipos existentes.
Un tipo de contacto debe tener como información mínima obligatoria un título que será el que de nombre al tipo. Opcionalmente se deberá permitir almacenar una descripción aclaratoria que amplíe información sobre el tipo de contacto.
- Citas:
Esta funcionalidad permitirá al usuario gestionar sus citas. El usuario podrá introducir una nueva cita, modificar una ya existente o eliminarla. También debe poder consultar todas las citas que tiene almacenadas en el sistema. Para tal fin se deberá permitir listar por pantalla todas las citas existentes, de forma que pueda seleccionar una y ver sus detalles.
Se debe permitir, opcionalmente, asignar una cita a uno o más contactos.
La información mínima para una cita debe ser la fecha y la hora de la misma, y un título que la identifique. Opcionalmente se debe permitir guardar una descripción donde proporcionar información más detallada de la misma.
- Notas:
Esta funcionalidad permitirá al usuario llevar una gestión completa de anotaciones que considere de interés. Inicialmente se pretende que el agente comercial no tenga que llevar anotaciones en papeles aparte y quede todo integrado en la aplicación. De esta forma dispondrá de toda la información necesaria con una simple consulta, sin peligro de perder anotaciones hechas en hojas sueltas.
Por lo tanto, será necesario que pueda añadir una nueva nota, modificar las ya existentes y/o eliminarlas. También debe poder consultar cualquiera de ellas para ver en detalle su información. Para seleccionar las notas sobre las que operar, primeramente se listarán todas ellas de forma ordenada.
Se debe permitir asignar una nota a uno o más contactos.
La información mínima obligatoria que compondrá una nota será su título que la identificará en el momento de mostrar el listado. Será deseable, aunque opcional, que se cumplimente el campo detalles donde se almacenará la información detallada de la nota.
- Captura de imágenes:
Haciendo uso de la cámara fotográfica del dispositivo, el usuario debe poder realizar fotografías y guardarlas en el sistema. Estas capturas pueden ser tarjetas de visita, productos, personas, instalaciones... por lo que se considerará necesario el poder asociarlas a uno o más contactos. Las operaciones permitidas sobre las imágenes almacenadas han de ser obtener un listado ordenado de las mismas, ver una imagen en detalle, eliminar la imagen y asociar una imagen a un contacto. La información que se ha de guardar sobre la imagen no es importante para el usuario y será totalmente transparente para el al no tener que añadir ningún tipo de información para guardar la imagen. Simplemente ha de realizar la captura y elegir si la guarda o no.

- **Geolocalización:**
La aplicación debe ofrecer dos funcionalidades relacionadas con la geolocalización proporcionada por el dispositivo GPS integrado en el terminal.
Primeramente debe permitir que, introducidas unas coordenadas, nos posicione en un mapa su ubicación exacta. Por defecto mostrará la ubicación actual del usuario para que sepa en cualquier momento donde se encuentra.
La segunda funcionalidad será obtener una ruta óptima entre dos puntos cualesquiera que el usuario deberá introducir previamente.
No se almacenarán datos de ningún tipo en ninguna de las dos funcionalidades, por lo que no se contempla conservar un histórico de posiciones, rutas, etc...

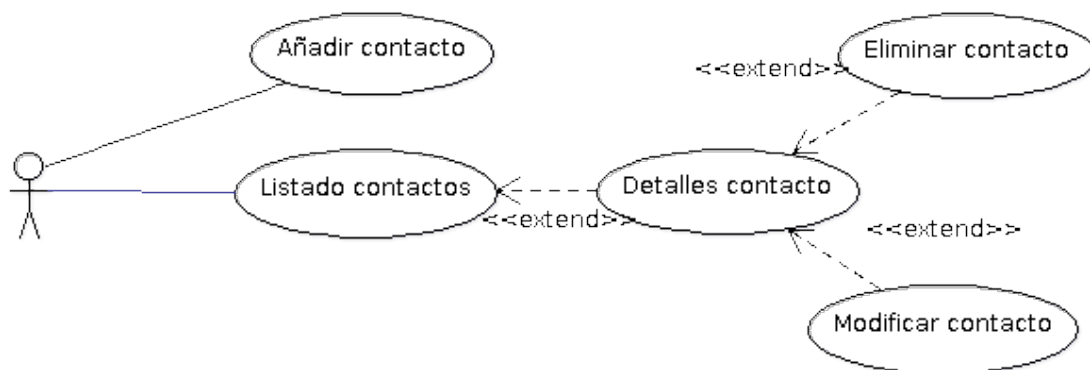
3 Análisis y diseño del sistema

Para un correcto análisis del sistema, realizaremos los pertinentes casos de uso que se compondrán de un diagrama y una tabla con la explicación detallada del mismo. Los casos de uso nos serán fundamentales para la toma de decisiones a la hora de diseñar el modelo de clases y la persistencia de datos (modelo entidad-relación)

3.1 Casos de uso

El análisis de los distintos casos de uso de la aplicación se ha llevado a cabo agrupándolos por funcionalidad. De esta forma obtenemos seis grupos de los que procedemos a realizar su diagrama y la descripción detallada de los casos que los forman.

o Contactos



Nombre:	Añadir contacto (CU.1.1)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none"> • Añade un nuevo contacto a la agenda
Actores:	<ul style="list-style-type: none"> • Usuario

Precondiciones:
<ul style="list-style-type: none"> • El usuario debe haber iniciado la aplicación correctamente
Flujo Normal:
<ol style="list-style-type: none"> 1. El usuario selecciona la opción 'Contactos' del menú de inicio. 2. El usuario selecciona la opción 'Añadir Contacto' 3. La aplicación muestra un formulario con todos los datos que el usuario puede introducir para el nuevo contacto. 4. El usuario cumplimenta el formulario. 5. El usuario pulsa el botón 'Guardar' 6. La aplicación almacena los datos del nuevo contacto en la agenda.
Flujo Alternativo:
Postcondiciones:
<ul style="list-style-type: none"> • El nuevo contacto queda almacenado en la agenda

Nombre:	Listar contactos (CU.1.2)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none"> • Lista por pantalla todos los contactos de la agenda.
Actores:	<ul style="list-style-type: none"> • Usuario
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe haber iniciado la aplicación correctamente
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona la opción 'Contactos' del menú de inicio. 2. La aplicación muestra por pantalla un listado completo de todos los contactos de la agenda.
Flujo Alternativo:	
Postcondiciones:	

Nombre:	Detalles del contacto (CU.1.3)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none"> • Se muestran por pantalla todos los datos detallados de un contacto de la agenda.
Actores:	<ul style="list-style-type: none"> • Usuario
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe haber iniciado la aplicación correctamente • El usuario debe existir en la agenda.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona la opción 'Contactos' del menú de inicio. 2. La aplicación muestra un listado completo de todos los contactos de la agenda.

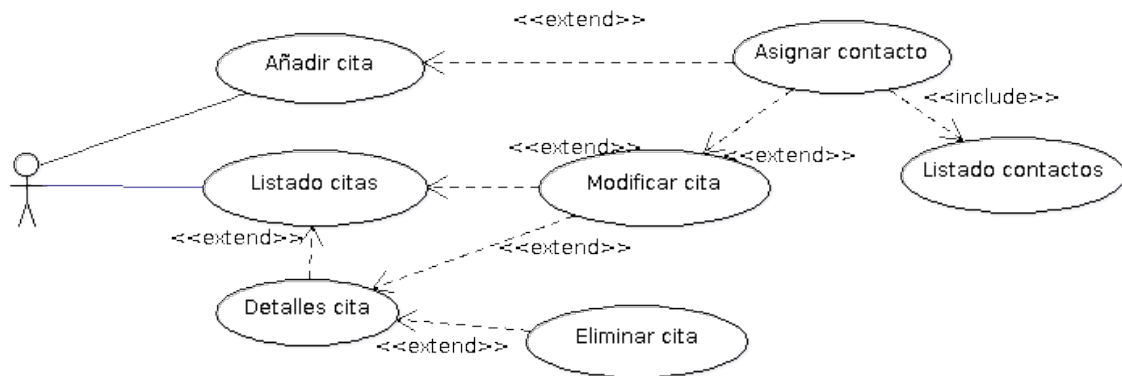
<ol style="list-style-type: none"> 3. El usuario selecciona un contacto del listado. 4. La aplicación muestra por pantalla todos los detalles del contacto.
Flujo Alternativo:
Postcondiciones:

Nombre:	Eliminar contacto (CU.1.4)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none"> • Elimina un contacto existente de la agenda
Actores:	<ul style="list-style-type: none"> • Usuario
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe haber iniciado la aplicación correctamente • El contacto debe existir en la agenda.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona la opción 'Contactos' del menú de inicio. 2. La aplicación muestra un listado completo de todos los contactos de la agenda. 3. El usuario selecciona un contacto del listado. 4. La aplicación muestra por pantalla todos los detalles del contacto. 5. El usuario pulsa el botón 'Eliminar contacto' 6. La aplicación pide confirmación de la operación. 7. El usuario pulsa la opción 'Sí' 8. La aplicación elimina el contacto de la agenda.
Flujo Alternativo:	<ol style="list-style-type: none"> 7. El usuario pulsa la opción 'No' 8. La aplicación vuelve a mostrar los detalles del contacto sin eliminarlo de la agenda.
Postcondiciones:	<ul style="list-style-type: none"> • El contacto queda eliminado por completo de la agenda

Nombre:	Modificar contacto (CU.1.5)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none"> • Modifica la información de un contacto existente de la agenda
Actores:	<ul style="list-style-type: none"> • Usuario
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe haber iniciado la aplicación correctamente • El contacto debe existir en la agenda.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona la opción 'Contactos' del menú de inicio. 2. La aplicación muestra un listado completo de todos los contactos de la agenda. 3. El usuario selecciona un contacto del listado.

<ol style="list-style-type: none"> 4. La aplicación muestra por pantalla todos los detalles del contacto. 5. El usuario modifica los datos del contacto. 6. El usuario pulsa el botón 'Guardar cambios' 7. La aplicación pide confirmación de la operación. 8. El usuario pulsa la opción 'Sí' 9. La aplicación modifica los datos del contacto de la agenda.
<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 7. El usuario pulsa la opción 'No' 8. La aplicación vuelve a mostrar los detalles del contacto sin modificar ningún dato.
<p>Postcondiciones:</p> <ul style="list-style-type: none"> • Los datos del contacto quedan modificados.

○ Citas



Nombre:	Añadir cita (CU.2.1)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none"> • Añade una nueva cita en la agenda
Actores:	<ul style="list-style-type: none"> • Usuario
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe haber iniciado la aplicación correctamente
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona la opción 'Citas' del menú de inicio. 2. El usuario selecciona la opción 'Añadir Cita' 3. La aplicación muestra un formulario con todos los datos que el usuario puede introducir para la nueva cita. 4. El usuario cumplimenta el formulario. 5. El usuario pulsa el botón 'Guardar'

6. La aplicación almacena los datos de la nueva cita en la agenda.
Flujo Alternativo:
Postcondiciones: <ul style="list-style-type: none">• La nueva cita queda almacenada en la agenda

Nombre:	Listar citas (CU.2.2)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none">• Lista por pantalla todas las citas de la agenda.
Actores:	<ul style="list-style-type: none">• Usuario
Precondiciones:	<ul style="list-style-type: none">• El usuario debe haber iniciado la aplicación correctamente
Flujo Normal:	<ol style="list-style-type: none">1. El usuario selecciona la opción 'Citas' del menú de inicio.2. La aplicación muestra por pantalla un listado completo de todas las citas de la agenda.
Flujo Alternativo:	
Postcondiciones:	

Nombre:	Detalles de la cita (CU.2.3)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none">• Se muestran por pantalla todos los datos detallados de una cita de la agenda.
Actores:	<ul style="list-style-type: none">• Usuario
Precondiciones:	<ul style="list-style-type: none">• El usuario debe haber iniciado la aplicación correctamente• La cita debe existir en la agenda.
Flujo Normal:	<ol style="list-style-type: none">1. El usuario selecciona la opción 'Citas' del menú de inicio.2. La aplicación muestra un listado completo de todas las citas de la agenda.3. El usuario selecciona una cita del listado.4. La aplicación muestra por pantalla todos los detalles de la cita.
Flujo Alternativo:	
Postcondiciones:	

Nombre:	Eliminar cita (CU.2.4)
Autor:	José A. Bernal Marcos

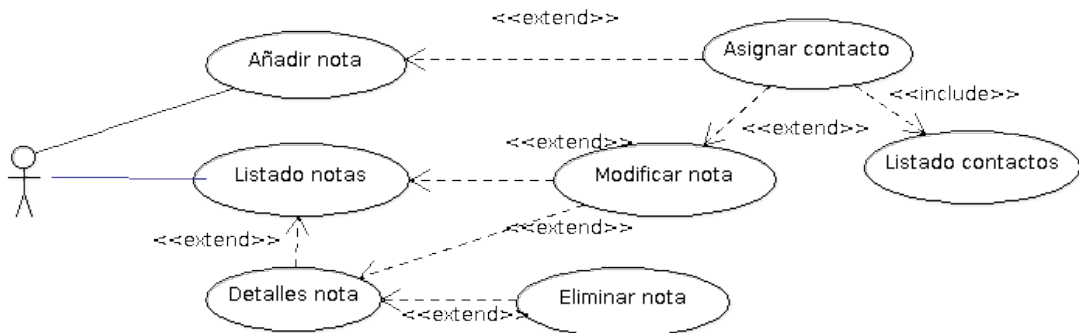
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none"> • Elimina una cita existente de la agenda
Actores:	<ul style="list-style-type: none"> • Usuario
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe haber iniciado la aplicación correctamente • La cita debe existir en la agenda.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona la opción 'Citas' del menú de inicio. 2. La aplicación muestra un listado completo de todas las citas de la agenda. 3. El usuario selecciona una cita del listado. 4. La aplicación muestra por pantalla todos los detalles de la cita. 5. El usuario pulsa el botón 'Eliminar cita' 6. La aplicación pide confirmación de la operación. 7. El usuario pulsa la opción 'Sí' 8. La aplicación elimina la cita de la agenda.
Flujo Alternativo:	<ol style="list-style-type: none"> 7. El usuario pulsa la opción 'No' 8. La aplicación vuelve a mostrar los detalles de la cita sin eliminarla de la agenda.
Postcondiciones:	<ul style="list-style-type: none"> • La cita queda eliminada por completo de la agenda

Nombre:	Modificar cita (CU.2.5)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none"> • Modifica la información de una cita existente de la agenda
Actores:	<ul style="list-style-type: none"> • Usuario
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe haber iniciado la aplicación correctamente • La cita debe existir en la agenda.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona la opción 'Citas' del menú de inicio. 2. La aplicación muestra un listado completo de todas las citas de la agenda. 3. El usuario selecciona una cita del listado. 4. La aplicación muestra por pantalla todos los detalles de la cita. 5. El usuario modifica los datos de la cita. 6. El usuario pulsa el botón 'Guardar cambios' 7. La aplicación pide confirmación de la operación. 8. El usuario pulsa la opción 'Sí' 9. La aplicación modifica los datos de la cita en la agenda.
Flujo Alternativo:	<ol style="list-style-type: none"> 7. El usuario pulsa la opción 'No' 8. La aplicación vuelve a mostrar los detalles de la cita sin modificar ningún dato.
Postcondiciones:	

- Los datos de la cita quedan modificados.

Nombre:	Asignar cita a contacto (CU.2.6)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none"> • Se asigna una cita con un contacto de la agenda.
Actores:	<ul style="list-style-type: none"> • Usuario
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe haber iniciado la aplicación correctamente • La cita debe existir en la agenda. • El usuario debe existir en la agenda.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona la opción 'Citas' del menú de inicio. 2. La aplicación muestra un listado completo de todas las citas de la agenda. 3. El usuario selecciona una cita del listado. 4. La aplicación muestra por pantalla todos los detalles de la cita. 5. El usuario selecciona la opción 'Asignar a contacto' 6. La aplicación muestra un listado con todos los contactos existentes en la agenda. 7. El usuario selecciona un contacto del listado. 8. El usuario selecciona la opción 'Guardar' 9. La aplicación guarda la relación entre el contacto y la cita en la agenda.
Flujo Alternativo:	
Postcondiciones:	<ul style="list-style-type: none"> • La cita queda asignada a uno o varios contactos de la agenda.

○ **Notas**



Nombre:	Añadir nota (CU.3.1)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none">• Añade una nueva nota en la agenda
Actores:	<ul style="list-style-type: none">• Usuario
Precondiciones:	<ul style="list-style-type: none">• El usuario debe haber iniciado la aplicación correctamente
Flujo Normal:	<ol style="list-style-type: none">1. El usuario selecciona la opción 'Notas' del menú de inicio.2. El usuario selecciona la opción 'Añadir Nota'3. La aplicación muestra un formulario con todos los datos que el usuario puede introducir para la nueva nota.4. El usuario cumplimenta el formulario.5. El usuario pulsa el botón 'Guardar'6. La aplicación almacena los datos de la nueva nota en la agenda.
Flujo Alternativo:	
Postcondiciones:	<ul style="list-style-type: none">• La nueva nota queda almacenada en la agenda

Nombre:	Listar notas (CU.3.2)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none">• Lista por pantalla todas las notas de la agenda.
Actores:	<ul style="list-style-type: none">• Usuario
Precondiciones:	<ul style="list-style-type: none">• El usuario debe haber iniciado la aplicación correctamente
Flujo Normal:	<ol style="list-style-type: none">1. El usuario selecciona la opción 'Notas' del menú de inicio.2. La aplicación muestra por pantalla un listado completo de todas las notas de la agenda.
Flujo Alternativo:	
Postcondiciones:	

Nombre:	Detalles de la nota (CU.3.3)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none">• Se muestran por pantalla todos los datos detallados de una nota de la agenda.
Actores:	<ul style="list-style-type: none">• Usuario

Precondiciones: <ul style="list-style-type: none">• El usuario debe haber iniciado la aplicación correctamente• La nota debe existir en la agenda.
Flujo Normal: <ol style="list-style-type: none">1. El usuario selecciona la opción 'Notas' del menú de inicio.2. La aplicación muestra un listado completo de todas las notas de la agenda.3. El usuario selecciona una nota del listado.4. La aplicación muestra por pantalla todos los detalles de la nota.
Flujo Alternativo:
Postcondiciones:

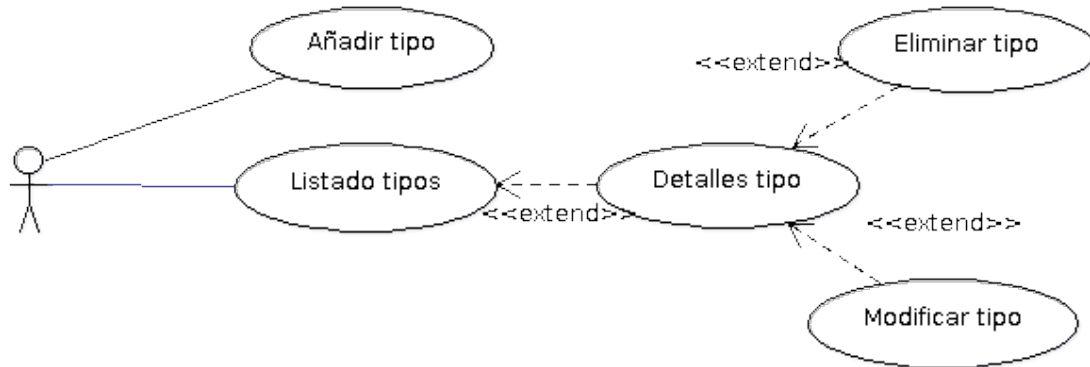
Nombre:	Eliminar nota (CU.3.4)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción: <ul style="list-style-type: none">• Elimina una nota existente de la agenda	
Actores: <ul style="list-style-type: none">• Usuario	
Precondiciones: <ul style="list-style-type: none">• El usuario debe haber iniciado la aplicación correctamente• La nota debe existir en la agenda.	
Flujo Normal: <ol style="list-style-type: none">1. El usuario selecciona la opción 'Notas' del menú de inicio.2. La aplicación muestra un listado completo de todas las notas de la agenda.3. El usuario selecciona una nota del listado.4. La aplicación muestra por pantalla todos los detalles de la nota.5. El usuario pulsa el botón 'Eliminar nota'6. La aplicación pide confirmación de la operación.7. El usuario pulsa la opción 'Sí'8. La aplicación elimina la nota de la agenda.	
Flujo Alternativo: <ol style="list-style-type: none">7. El usuario pulsa la opción 'No'8. La aplicación vuelve a mostrar los detalles de la nota sin eliminarla de la agenda.	
Postcondiciones: <ul style="list-style-type: none">• La nota queda eliminada por completo de la agenda	

Nombre:	Modificar nota (CU.3.5)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción: <ul style="list-style-type: none">• Modifica la información de una nota existente de la agenda	
Actores: <ul style="list-style-type: none">• Usuario	
Precondiciones:	

<ul style="list-style-type: none"> • El usuario debe haber iniciado la aplicación correctamente • La nota debe existir en la agenda.
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El usuario selecciona la opción 'Notas' del menú de inicio. 2. La aplicación muestra un listado completo de todas las notas de la agenda. 3. El usuario selecciona una nota del listado. 4. La aplicación muestra por pantalla todos los detalles de la nota. 5. El usuario modifica los datos de la nota. 6. El usuario pulsa el botón 'Guardar cambios' 7. La aplicación pide confirmación de la operación. 8. El usuario pulsa la opción 'Sí' 9. La aplicación modifica los datos de la nota en la agenda.
<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 7. El usuario pulsa la opción 'No' 8. La aplicación vuelve a mostrar los detalles de la nota sin modificar ningún dato.
<p>Postcondiciones:</p> <ul style="list-style-type: none"> • Los datos de la nota quedan modificados.

Nombre:	Asignar nota a contacto (CU.3.6)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none"> • Se asigna una nota a un contacto de la agenda.
Actores:	<ul style="list-style-type: none"> • Usuario
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe haber iniciado la aplicación correctamente • La nota debe existir en la agenda. • El contacto debe existir en la agenda.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona la opción 'Notas' del menú de inicio. 2. La aplicación muestra un listado completo de todas las notas de la agenda. 3. El usuario selecciona una nota del listado. 4. La aplicación muestra por pantalla todos los detalles de la nota. 5. El usuario selecciona la opción 'Asignar a contacto' 6. La aplicación muestra un listado con todos los contactos existentes en la agenda. 7. El usuario selecciona un contacto del listado. 8. El usuario selecciona la opción 'Guardar' 9. La aplicación guarda la relación entre el contacto y la nota en la agenda.
Flujo Alternativo:	
Postcondiciones:	<ul style="list-style-type: none"> • La nota queda asignada a uno o varios contactos de la agenda.

○ Tipos de contacto



Nombre:	Añadir tipo (CU.4.1)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none"> Añade un nuevo tipo de contacto en la agenda
Actores:	<ul style="list-style-type: none"> Usuario
Precondiciones:	<ul style="list-style-type: none"> El usuario debe haber iniciado la aplicación correctamente
Flujo Normal:	<ol style="list-style-type: none"> El usuario selecciona la opción 'Tipos de contacto' del menú de inicio. El usuario selecciona la opción 'Añadir Tipo de contacto' La aplicación muestra un formulario con todos los datos que el usuario puede introducir para el nuevo tipo de contacto. El usuario cumplimenta el formulario. El usuario pulsa el botón 'Guardar' La aplicación almacena los datos del nuevo tipo de contacto en la agenda.
Flujo Alternativo:	
Postcondiciones:	<ul style="list-style-type: none"> El nuevo tipo de contacto queda almacenado en la agenda

Nombre:	Listar tipos (CU.4.2)
Autor:	José A. Bernal Marcos

Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none">• Lista por pantalla todos los tipos de contacto de la agenda.
Actores:	<ul style="list-style-type: none">• Usuario
Precondiciones:	<ul style="list-style-type: none">• El usuario debe haber iniciado la aplicación correctamente
Flujo Normal:	<ol style="list-style-type: none">1. El usuario selecciona la opción 'Tipos de contacto' del menú de inicio.2. La aplicación muestra por pantalla un listado completo de todos los tipos de contacto de la agenda.
Flujo Alternativo:	
Postcondiciones:	

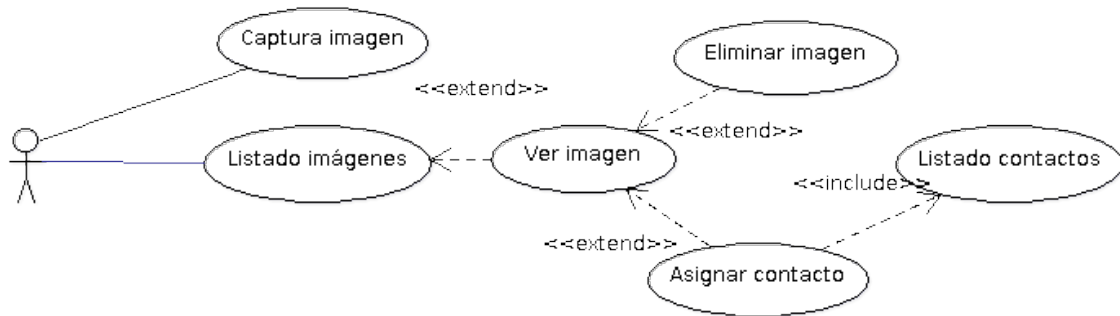
Nombre:	Detalles del tipo (CU.4.3)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none">• Se muestran por pantalla todos los datos detallados de un tipo de contacto de la agenda.
Actores:	<ul style="list-style-type: none">• Usuario
Precondiciones:	<ul style="list-style-type: none">• El usuario debe haber iniciado la aplicación correctamente• El tipo de contacto debe existir en la agenda.
Flujo Normal:	<ol style="list-style-type: none">1. El usuario selecciona la opción 'Tipos de contacto' del menú de inicio.2. La aplicación muestra un listado completo de todos los tipos de contacto de la agenda.3. El usuario selecciona un tipo de contacto del listado.4. La aplicación muestra por pantalla todos los detalles del tipo de contacto.
Flujo Alternativo:	
Postcondiciones:	

Nombre:	Eliminar tipo (CU.4.4)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none">• Elimina un tipo de contacto existente de la agenda
Actores:	<ul style="list-style-type: none">• Usuario
Precondiciones:	<ul style="list-style-type: none">• El usuario debe haber iniciado la aplicación correctamente• El tipo de contacto debe existir en la agenda.
Flujo Normal:	<ol style="list-style-type: none">1. El usuario selecciona la opción 'Tipos de contacto' del menú de inicio.

<ol style="list-style-type: none">2. La aplicación muestra un listado completo de todos los tipos de contacto de la agenda.3. El usuario selecciona un tipo del listado.4. La aplicación muestra por pantalla todos los detalles del tipo.5. El usuario pulsa el botón 'Eliminar tipo'6. La aplicación pide confirmación de la operación.7. El usuario pulsa la opción 'Sí'8. La aplicación elimina el tipo de contacto de la agenda.
Flujo Alternativo: <ol style="list-style-type: none">7. El usuario pulsa la opción 'No'8. La aplicación vuelve a mostrar los detalles del tipo de contacto sin eliminarlo de la agenda.
Postcondiciones: <ul style="list-style-type: none">• El tipo queda eliminado por completo de la agenda

Nombre:	Modificar tipo (CU.4.5)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none">• Modifica la información de un tipo de contacto existente de la agenda
Actores:	<ul style="list-style-type: none">• Usuario
Precondiciones:	<ul style="list-style-type: none">• El usuario debe haber iniciado la aplicación correctamente• El tipo de contacto debe existir en la agenda.
Flujo Normal:	<ol style="list-style-type: none">1. El usuario selecciona la opción 'Tipos de contacto' del menú de inicio.2. La aplicación muestra un listado completo de todos los tipos de contacto de la agenda.3. El usuario selecciona un tipo del listado.4. La aplicación muestra por pantalla todos los detalles del tipo.5. El usuario modifica los datos del tipo.6. El usuario pulsa el botón 'Guardar cambios'7. La aplicación pide confirmación de la operación.8. El usuario pulsa la opción 'Sí'9. La aplicación modifica los datos del tipo de contacto en la agenda.
Flujo Alternativo:	<ol style="list-style-type: none">7. El usuario pulsa la opción 'No'8. La aplicación vuelve a mostrar los detalles del tipo sin modificar ningún dato.
Postcondiciones:	<ul style="list-style-type: none">• Los datos del tipo de contacto quedan modificados.

○ **Imágenes**



Nombre:	Capturar imagen (CU.5.1)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none"> Captura una nueva imagen y la guarda en el sistema
Actores:	<ul style="list-style-type: none"> Usuario
Precondiciones:	<ul style="list-style-type: none"> El usuario debe haber iniciado la aplicación correctamente
Flujo Normal:	<ol style="list-style-type: none"> El usuario selecciona la opción 'Imágenes' del menú de inicio. El usuario selecciona la opción 'Añadir Nota' La aplicación muestra el visor y los controles de la cámara de fotos (aplicación propia del sistema). El usuario realiza la captura.
Flujo Alternativo:	
Postcondiciones:	<ul style="list-style-type: none"> La imagen queda almacenada en el sistema.

Nombre:	Listar imágenes (CU.5.2)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none"> Lista por pantalla todas las imágenes capturadas desde la aplicación.
Actores:	<ul style="list-style-type: none"> Usuario
Precondiciones:	

<ul style="list-style-type: none"> El usuario debe haber iniciado la aplicación correctamente
<p>Flujo Normal:</p> <ol style="list-style-type: none"> El usuario selecciona la opción 'Imágenes' del menú de inicio. La aplicación muestra por pantalla un listado completo con todas las imágenes capturadas desde la aplicación
Flujo Alternativo:
Postcondiciones:

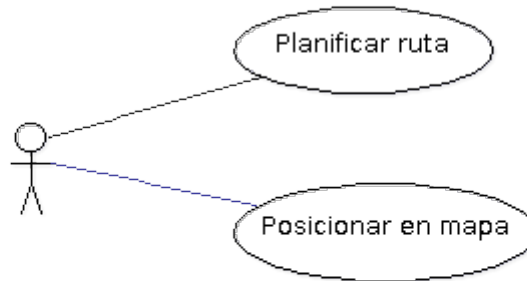
Nombre:	Ver imagen (CU.5.3)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none"> Se muestra por pantalla la imagen seleccionada a pantalla completa.
Actores:	<ul style="list-style-type: none"> Usuario
Precondiciones:	<ul style="list-style-type: none"> El usuario debe haber iniciado la aplicación correctamente La imagen debe existir en el sistema.
Flujo Normal:	<ol style="list-style-type: none"> El usuario selecciona la opción 'Imágenes' del menú de inicio. La aplicación muestra por pantalla un listado completo con todas las imágenes capturadas desde la aplicación. El usuario selecciona una imagen del listado. La aplicación muestra a pantalla completa la imagen seleccionada.
Flujo Alternativo:	
Postcondiciones:	

Nombre:	Eliminar imagen (CU.5.4)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none"> Elimina una imagen capturada desde la aplicación
Actores:	<ul style="list-style-type: none"> Usuario
Precondiciones:	<ul style="list-style-type: none"> El usuario debe haber iniciado la aplicación correctamente La imagen debe existir en el sistema.
Flujo Normal:	<ol style="list-style-type: none"> El usuario selecciona la opción 'Imágenes' del menú de inicio. La aplicación muestra por pantalla un listado completo con todas las imágenes capturadas desde la aplicación. El usuario selecciona una imagen del listado. La aplicación muestra la imagen en detalle junto a los botones 'Eliminar' y 'Asignar contacto' El usuario pulsa el botón 'Eliminar'

6. La aplicación pide confirmación de la operación. 7. El usuario pulsa la opción 'Sí' 8. La aplicación elimina la imagen del sistema.
Flujo Alternativo: 7. El usuario pulsa la opción 'No' 8. La aplicación vuelve a mostrar el listado de imágenes sin eliminar ninguna.
Postcondiciones: • La imagen queda eliminada por completo del sistema

Nombre:	Asignar imagen a contacto (CU.5.5)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none">• Se asigna una imagen capturada desde la aplicación a un contacto de la agenda.
Actores:	<ul style="list-style-type: none">• Usuario
Precondiciones:	<ul style="list-style-type: none">• El usuario debe haber iniciado la aplicación correctamente• La imagen debe existir en el sistema.• El contacto debe existir en la agenda.
Flujo Normal:	<ol style="list-style-type: none">1. El usuario selecciona la opción 'Imágenes' del menú de inicio.2. La aplicación muestra por pantalla un listado completo con todas las imágenes capturadas desde la aplicación.3. El usuario selecciona una imagen del listado.4. La aplicación muestra la imagen en detalle junto a los botones 'Eliminar' y 'Asignar contacto'5. El usuario selecciona la opción 'Asignar a contacto'6. La aplicación muestra un listado con todos los contactos existentes en la agenda.7. El usuario selecciona un contacto del listado.8. El usuario selecciona la opción 'Guardar'9. La aplicación guarda la relación entre el contacto y la imagen en la agenda.
Flujo Alternativo:	
Postcondiciones:	<ul style="list-style-type: none">• La imagen queda asignada a uno o varios contactos de la agenda.

o **Geolocalización**

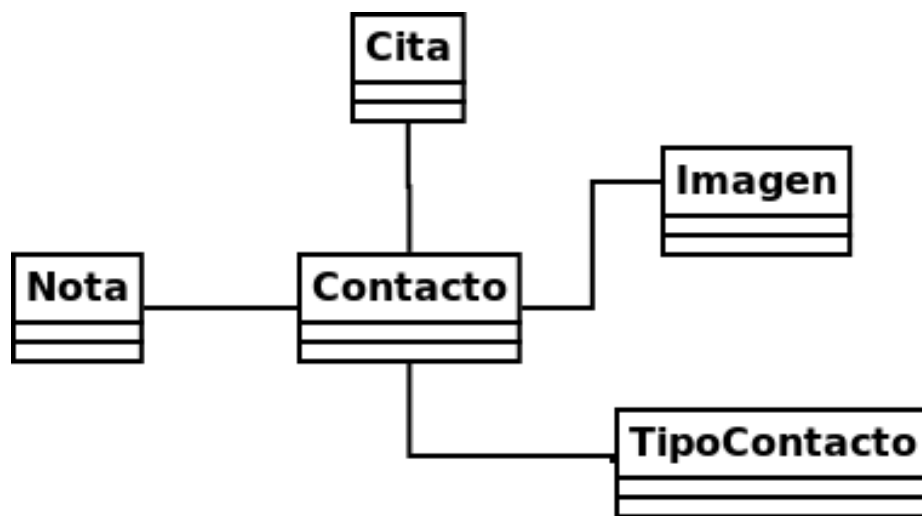


Nombre:	Planificar ruta (CU.6.1)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none"> Se realiza una planificación óptima de una ruta entre dos localizaciones introducidas.
Actores:	<ul style="list-style-type: none"> Usuario
Precondiciones:	<ul style="list-style-type: none"> El usuario debe haber iniciado la aplicación correctamente
Flujo Normal:	<ol style="list-style-type: none"> El usuario selecciona la opción 'Geolocalización' del menú de inicio. El usuario selecciona la opción 'Planificar ruta'. El usuario introduce las dos localizaciones necesarias (origen y destino). El usuario selecciona la opción 'Obtener ruta' La aplicación muestra por pantalla la ruta planificada.
Flujo Alternativo:	
Postcondiciones:	

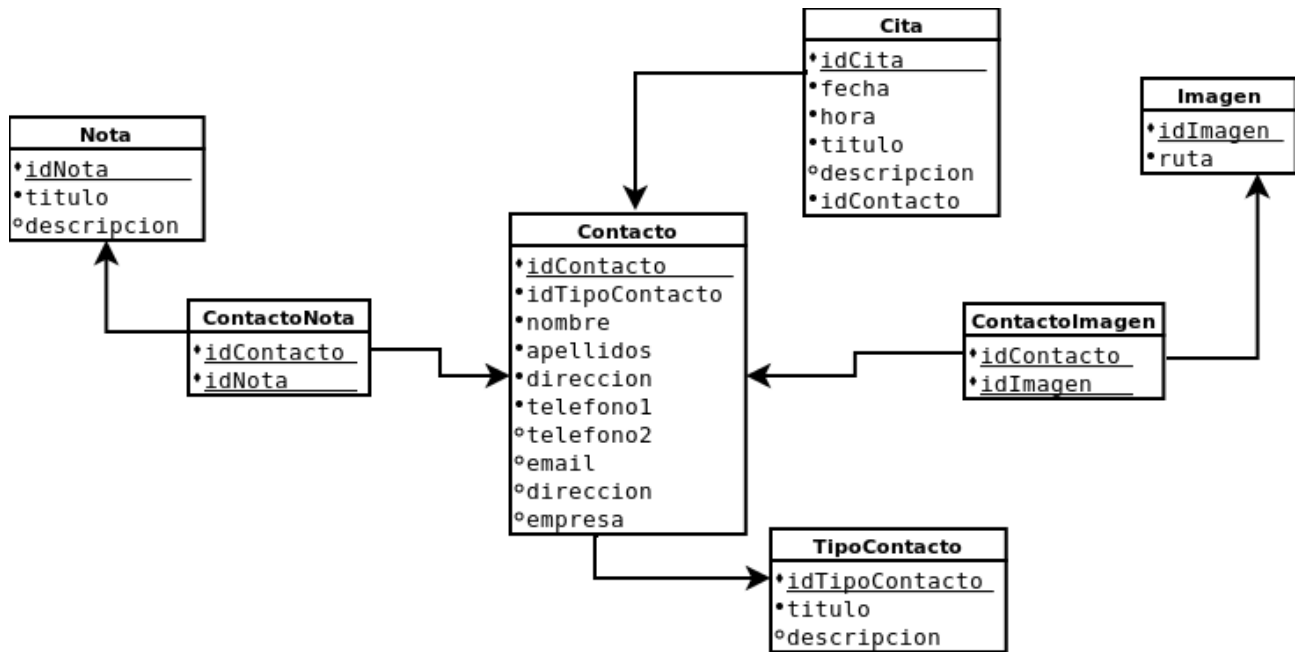
Nombre:	Posicionar en mapa(CU.6.2)
Autor:	José A. Bernal Marcos
Fecha:	5/10/2012
Descripción:	<ul style="list-style-type: none"> Se muestra la posición de la localización introducida en un mapa.
Actores:	<ul style="list-style-type: none"> Usuario

Precondiciones: <ul style="list-style-type: none">• El usuario debe haber iniciado la aplicación correctamente
Flujo Normal: <ol style="list-style-type: none">1. El usuario selecciona la opción 'Geolocalización' del menú de inicio.2. El usuario selecciona la opción 'Posicionar'.3. El usuario introduce la localización que quiere posicionar.4. El usuario selecciona la opción 'Posicionar'5. La aplicación muestra por pantalla la posición exacta de la localización introducida.
Flujo Alternativo:
Postcondiciones:

3.2 Diagrama de clases



3.3 Diagrama BD Entidad-Relación



4 Interfaz de usuario

4.1 Análisis

4.1.1 Diagrama de clases

Podemos clasificar los usuarios de la aplicación atendiendo a varios aspectos:

- Según factores sociales. La aplicación no debe discriminar usuarios por factores tales como edad, nivel de estudios, ingresos económicos... Todos ellos son posibles usuarios finales, por lo que no podemos diferenciar este tipo de factores.
- Según tareas a realizar en el portal. En principio todos los usuarios tendrán acceso a las mismas funcionalidades y podrán realizar las mismas tareas, sin diferenciación alguna por roles de acceso como usuario administrador, usuario registrado o similar.

Por lo tanto, podemos concluir que tendremos definido un único perfil de usuario:

- **Usuario general**

Usuario que utilizará la aplicación sin restricciones, por lo que tendrá acceso a cualquier función de la misma. Por lo tanto, no será necesario que el usuario se registre o se identifique de forma alguna en el sistema. El usuario puede pertenecer a cualquier clase o grupo social.

Conocimientos	<i>Usuario general</i>
Habla lingüística	El usuario debe dominar el idioma en el que se encuentra disponible la aplicación
Tecnología	Será necesario que el usuario tenga un conocimiento mínimo del terminal y su funcionamiento, suficiente para abrir la aplicación, realizar fotografías con la cámara integrada y configurar el dispositivo GPS.

Atributos físicos	<i>Usuario general</i>
Edad	Al ser una aplicación enfocada al ámbito profesional, es de suponer que los usuarios que la utilizarán deberán tener una edad igual o superior a la edad mínima legal para trabajar e igual o inferior a la edad máxima de jubilación. Por lo tanto, el intervalo de edad esperado para el usuario de la aplicación se encontraría entre los 16 y los 67 años.
Género	Se espera una distribución equitativa entre hombres y mujeres.

Motivaciones	<i>Usuario general</i>
	<ul style="list-style-type: none">- Tener una gestión completa sobre contactos, citas y todo tipo de información asociada a estos que sea de interés en su labor comercial.- Disponer de una herramienta que unifique toda esta información de forma segura y con accesibilidad inmediata en cualquier momento.- Contar con la posibilidad de poder posicionar geográficamente una localización concreta o saber como llegar desde un punto de origen indicado a otro de destino.

4.1.2 Contexto de uso

El contexto de uso hace referencia a todo lo que envuelve a los usuarios a la hora de utilizar la aplicación.

Se podrá acceder a la aplicación desde cualquier ubicación física, siendo necesario para ello disponer de un dispositivo (tablet o smartphone) compatible con ella. Esto es, el dispositivo deberá contar con uno de los siguientes sistemas: Iphone, Android, Blackberry, WebOS, Symbian, Bada, Windows Phone 7. Para poder utilizar todas las funcionalidades de la aplicación, el dispositivo deberá contar con cámara fotográfica y un receptor GPS.

Los usuarios tendrán un conocimiento básico de la tecnología utilizada, suficiente para ser capaces de manejar el dispositivo y acceder a la aplicación. También contarán con nociones básicas en el manejo de la cámara fotográfica y el receptor GPS para poder acceder a sus funcionalidades y utilizar todos sus recursos.

En cuanto al contexto de uso, en el apartado de perfiles de usuarios ya definimos sus motivaciones, atributos físicos y conocimientos específicos. En el punto de análisis de tareas se detallan las acciones a las que tendrán acceso cada uno de ellos.

4.1.3 Análisis de tareas

En este punto analizaremos las acciones que el usuario está capacitado para llevar a cabo en la aplicación. A continuación listamos las actividades permitidas para el perfil de usuario general, asumiendo que tiene la aplicación instalada en un terminal apropiado y que el mismo cuenta con una cámara fotográfica y un receptor GPS.

Añadir un contacto	
Objetivo:	Añadir un nuevo contacto a la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente
Postcondición:	El contacto queda añadido a la agenda.
Observaciones:	

Ver listado de contactos	
Objetivo:	Ver todos los contactos existentes en la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente
Postcondición:	
Observaciones:	

Ver detalles de un contacto	
Objetivo:	Ver la información detallada de un contacto concreto de la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente. El contacto existe en la agenda
Postcondición:	
Observaciones:	

Eliminar un contacto	
Objetivo:	Eliminar un contacto existente en la agenda.

Pre-condición:	El usuario ha iniciado la aplicación correctamente El contacto existe en la agenda
Postcondición:	El contacto queda eliminado de la agenda.
Observaciones:	El usuario puede cancelar la eliminación del contacto cuando la aplicación le pida confirmar la operación.

Modificar un contacto	
Objetivo:	Modificar un contacto existente en la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente El contacto existe en la agenda.
Postcondición:	El contacto queda modificado
Observaciones:	El usuario puede cancelar la eliminación del contacto cuando la aplicación le pida confirmar la operación.

Añadir una cita	
Objetivo:	Añadir una nueva cita a la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente
Postcondición:	La cita queda añadida a la agenda.
Observaciones:	

Ver listado de citas	
Objetivo:	Ver todas las citas existentes en la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente
Postcondición:	
Observaciones:	

Ver detalles de una cita	
Objetivo:	Ver la información detallada de una cita concreta de la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente La cita existe en la agenda
Postcondición:	
Observaciones:	

Eliminar una cita	
Objetivo:	Eliminar una cita concreta de la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente La cita existe en la agenda
Postcondición:	La cita queda eliminada de la agenda.
Observaciones:	El usuario puede cancelar la eliminación de la cita cuando la aplicación le pida confirmar la operación.

Modificar una cita	
Objetivo:	Modificar una cita concreta de la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente

	La cita existe en la agenda.
Postcondición:	La cita queda modificada
Observaciones:	El usuario puede cancelar la eliminación de la cita cuando la aplicación le pida confirmar la operación.

Asignar una cita a un contacto	
Objetivo:	Asignar una cita a un contacto existente en la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente La cita existe en la agenda. El contacto existe en la agenda
Postcondición:	La cita queda asignada al usuario seleccionado
Observaciones:	La cita puede ser asignada a varios contactos repitiendo esta tarea las veces que sean necesarias.

Añadir una nota	
Objetivo:	Añadir una nueva nota a la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente
Postcondición:	La nota queda añadida a la agenda.
Observaciones:	

Ver listado de notas	
Objetivo:	Ver todas las notas existentes en la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente
Postcondición:	
Observaciones:	

Ver detalles de una nota	
Objetivo:	Ver la información detallada de una nota concreta de la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente La nota existe en la agenda
Postcondición:	
Observaciones:	

Eliminar una nota	
Objetivo:	Eliminar una nota concreta de la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente La nota existe en la agenda
Postcondición:	La nota queda eliminada de la agenda.
Observaciones:	El usuario puede cancelar la eliminación de la nota cuando la aplicación le pida confirmar la operación.

Modificar una nota	
Objetivo:	Modificar una nota concreta de la agenda.

Pre-condición:	El usuario ha iniciado la aplicación correctamente La nota existe en la agenda.
Postcondición:	La nota queda modificada
Observaciones:	El usuario puede cancelar la eliminación de la nota cuando la aplicación le pida confirmar la operación.

Asignar una nota a un contacto	
Objetivo:	Asignar una nota a un contacto existente en la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente La nota existe en la agenda. El contacto existe en la agenda
Postcondición:	La nota queda asignada al usuario seleccionado
Observaciones:	La nota puede ser asignada a varios contactos repitiendo esta tarea las veces que sean necesarias.

Añadir un tipo de contacto	
Objetivo:	Añadir un nuevo tipo de contacto a la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente
Postcondición:	El tipo de contacto queda añadido a la agenda.
Observaciones:	

Ver listado de tipos de contacto	
Objetivo:	Ver todos los tipos de contacto existentes en la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente
Postcondición:	
Observaciones:	

Eliminar un tipo de contacto	
Objetivo:	Eliminar un tipo de contacto existente en la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente El tipo de contacto existe en la agenda
Postcondición:	El tipo de contacto queda eliminado de la agenda.
Observaciones:	El usuario puede cancelar la eliminación del tipo de contacto cuando la aplicación le pida confirmar la operación.

Modificar un tipo de contacto	
Objetivo:	Modificar un tipo de contacto existente en la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente El tipo de contacto existe en la agenda.
Postcondición:	El tipo de contacto queda modificado
Observaciones:	El usuario puede cancelar la eliminación del tipo de contacto cuando la aplicación le pida confirmar la operación.

Capturar una imagen	
Objetivo:	Capturar una imagen utilizando la cámara fotográfica del terminal.
Pre-condición:	El usuario ha iniciado la aplicación correctamente
Postcondición:	La fotografía queda almacenada en el sistema.
Observaciones:	

Ver listado de imágenes	
Objetivo:	Ver todas las imágenes tomadas desde la aplicación.
Pre-condición:	El usuario ha iniciado la aplicación correctamente
Postcondición:	
Observaciones:	

Ver imagen	
Objetivo:	Ver una fotografía tomada desde la aplicación de forma detallada (pantalla completa)
Pre-condición:	El usuario ha iniciado la aplicación correctamente La imagen existe en el sistema
Postcondición:	
Observaciones:	

Eliminar una imagen	
Objetivo:	Eliminar una imagen del sistema
Pre-condición:	El usuario ha iniciado la aplicación correctamente La imagen existe en el sistema
Postcondición:	La imagen queda eliminada del sistema
Observaciones:	El usuario puede cancelar la eliminación de la imagen cuando la aplicación le pida confirmar la operación.

Asignar una imagen a un contacto	
Objetivo:	Asignar una imagen tomada desde la aplicación a un contacto existente en la agenda.
Pre-condición:	El usuario ha iniciado la aplicación correctamente La imagen existe en el sistema El contacto existe en la agenda
Postcondición:	La imagen queda asignada al usuario seleccionado
Observaciones:	La imagen puede ser asignada a varios contactos repitiendo esta tarea las veces que sean necesarias.

Planificar una ruta	
Objetivo:	Dada una ubicación de origen y otra de destino, calcular la ruta óptima entre los dos puntos.
Pre-condición:	El usuario ha iniciado la aplicación correctamente El GPS está configurado correctamente y funcionando.

Postcondición:	
Observaciones:	

Posicionar en mapa	
Objetivo:	Mostrar la posición de la localización introducida en un mapa.
Pre-condición:	El usuario ha iniciado la aplicación correctamente El GPS está configurado correctamente y funcionando.
Postcondición:	
Observaciones:	

4.2 Diseño conceptual

4.2.1 Escenarios de uso

La primera tarea a realizar en esta fase será definir los escenarios de uso, que son hipótesis imaginarias sobre las situaciones en las que se encuentran los usuarios al realizar las tareas descritas en el análisis de tareas, utilizando los contextos de uso ya identificados y los perfiles de usuario que hemos definido. En nuestro caso sólo uno, el de usuario general.

Para esta tarea se ha utilizado la técnica de 'personajes', donde se relatan situaciones ficticias con usuarios pertenecientes a los perfiles identificados y realizando las tareas descritas en el análisis previo. Para facilitar el análisis y la comprensión de los escenarios, se han agrupado por funcionalidades principales (contactos, tipos de contactos, citas, notas, captura de imágenes y geolocalización)

- Escenario para el uso de la gestión de contactos

Daniel es un agente comercial. En una de sus visitas a un cliente debe de anotar el nuevo contacto de Ángel, que será el encargado de realizar los pedidos a la empresa. Daniel coge su smartphone y abre la aplicación. Pulsa la opción 'Contactos' y luego la opción 'Nuevo Contacto'. En el formulario que aparece en pantalla agrega toda la información de la que dispone sobre Ángel. Cuando termina de agregar información, pulsa 'Guardar' y el contacto queda añadido a la agenda.

Decide que para no confundirse de responsable, lo mejor es eliminar al antiguo contacto de la empresa. Desde el menú principal de la aplicación pulsa la opción 'Contactos'. La aplicación muestra un listado con todos los contactos almacenados. Selecciona el registro perteneciente al antiguo encargado y se abre la ventana de detalles del contacto. Pulsa sobre el botón Eliminar. La aplicación pide confirmación sobre la operación que se va a llevar a cabo y Daniel la confirma. El contacto del antiguo responsable queda eliminado de la agenda.

Pero Daniel se da cuenta de que le falta por guardar el correo electrónico de Ángel. Desde el menú principal de la aplicación selecciona nuevamente la opción 'Contactos' y selecciona el registro de Ángel del listado de contactos disponibles. Se abre la ventana con toda la información detallada del contacto. En el campo destinado al correo electrónico, escribe la dirección y pulsa el botón 'Modificar'. La información queda guardada en la agenda.

- Escenario para el uso de la gestión de tipos de contactos

Daniel desea poder agrupar también en la agenda de la aplicación algunos de sus contactos personales. Para ello utiliza la gestión de tipos de contacto. Al entrar en la aplicación, selecciona la opción 'Tipos de contacto'. Se muestra el listado de tipos ya dados de alta en la agenda y Daniel selecciona la opción 'Añadir tipo de contacto'. Se muestra un formulario vacío con los campos necesarios para agregar un nuevo tipo. Daniel los cumplimenta y pulsa el botón 'Guardar'. El nuevo tipo de contacto queda guardado en la agenda.

Igualmente, se da cuenta de que no está utilizando un tipo de contacto que ya existe en la agenda y desea eliminarlo. Para ello selecciona la opción 'Tipos de contacto' en el menú principal de la aplicación y desde el listado de tipos, selecciona el registro que quiere eliminar. Se muestra un formulario con toda la información de la que se dispone del tipo y pulsa la opción 'Eliminar'. La aplicación pide confirmación de la operación, y tras confirmar el registro queda eliminado de la agenda.

Daniel se da cuenta de que se ha equivocado al dar de alta el tipo de contactos personales y desea modificar su nombre. Igual que antes, selecciona la opción 'Tipos de contacto' y selecciona el tipo en el listado que se muestra por pantalla. Desde el formulario con la información detallada del tipo que se abre a continuación modifica el título del registro y pulsa el botón 'Guardar'. De esta forma el tipo de contacto queda modificado en la agenda.

- Escenario para el uso de la gestión de citas

Daniel tiene una cita con Ángel para presentarle unos nuevos descuentos en sus pedidos. Quiere anotar la fecha y la hora en la agenda para no olvidarla. Abre la aplicación y desde el menú principal selecciona la opción 'Citas'. Se muestra por pantalla el listado de citas que ya tiene guardadas. Selecciona la opción 'Nueva cita'. Rellena el formulario de alta con la información necesaria de la cita. Como además quiere asignar la cita a Ángel, selecciona su contacto desde la lista de contactos disponibles que se muestra en el formulario. Pulsa el botón 'Guardar' y la cita queda guardada en la agenda y asignada a Ángel.

Poco después, Daniel recibe la llamada de Ángel que le pide atrasar la cita una hora para poder llegar a tiempo. Daniel abre la aplicación y pulsa la opción 'Citas'. Selecciona la cita que tiene con Ángel y en el formulario de la vista de detalle modifica la hora. Pulsa el botón 'Modificar' y la cita queda guardada con los nuevos datos.

Finalmente Ángel no puede acudir a la cita y llama para anularla. Daniel abre la aplicación y de nuevo pulsa sobre 'Citas'. Selecciona la cita con Ángel y en el formulario de vista de detalle pulsa el botón 'Eliminar'. La aplicación pide que confirme la operación y tras hacerlo, la cita queda eliminada de la agenda.

- Escenario para el uso de la gestión de notas

Inés es la directora comercial de una multinacional interesada en realizar un gran pedido a la empresa de Daniel. Pero antes le ha pedido el listado de tarifas y descuentos disponibles para estudiarlo y poder comparar con otros proveedores. Daniel decide anotarlo en la agenda para no olvidarlo. Abre la aplicación y selecciona la opción 'Notas'. Aparece un listado con todas las notas que ya están guardadas en la agenda. Selecciona la opción 'Nueva nota' y rellena el formulario que se muestra por pantalla con los datos necesarios y la información que quiere guardar. Como quiere asignar la nota a Inés, selecciona su contacto del listado de contactos que muestra la aplicación en el mismo formulario de alta. Una vez ha introducido todos los datos, pulsa el botón 'Guardar' y la nota queda guardada en la agenda y asignada a Inés.

Poco después recibe una llamada de Inés que le pide que le envíe también información sobre plazos de pago posibles. Daniel lo añade a la nota. Para ello abre la aplicación y pulsa la opción 'Notas'. Selecciona la nota en el listado que se muestra por pantalla y abre el formulario de vista en detalle. Edita el campo correspondiente con la nueva información y pulsa el botón 'Modificar'. La nota queda guardada con los nuevos datos.

Finalmente Inés llama a Daniel y le pide que no le envíe la información solicitada porque un agente comercial de su empresa ya la tenía y se la ha remitido por correo electrónico. Así que Daniel decide eliminar la nota para evitar confusiones. Abre la aplicación y pulsa la opción 'Notas'. Selecciona la nota que corresponde a Inés y desde el formulario de vista de detalle pulsa el botón 'Eliminar'. Confirma que quiere llevar a cabo la operación y la nota queda eliminada de la agenda.

- Escenario para el uso de la captura de imágenes

Daniel está visitando las instalaciones de un posible nuevo proveedor. Le han enseñado un producto en el que podrían estar interesados y piensa que sería interesante tomar una imagen del mismo para que los directivos de la empresa puedan verlo y decidir si lo distribuyen o no. Abre la aplicación y selecciona la opción 'Imágenes'. Junto al listado de imágenes que han sido tomadas desde la aplicación se muestra el botón 'Nueva imagen'. Daniel lo pulsa y se abre la aplicación nativa del dispositivo para tomar fotografías. Tras realizar la captura, la nueva imagen queda almacenada en la agenda.

Daniel quiere asegurarse de que la fotografía ha salido nítida. En el menú principal pulsa la opción 'Imágenes'. Se muestra el listado con todas las imágenes disponibles y selecciona la del producto que acaba de tomar. Se muestra la imagen de forma detallada.

Para no olvidar a quien pertenece la fotografía del producto, Daniel quiere asociarla al contacto de la empresa proveedora. Desde el menú inicial de la aplicación selecciona la opción 'Imágenes'. En el listado de imágenes selecciona la del producto y pulsa la opción 'Asociar a contacto'. Se muestra el listado de contactos de la agenda. Selecciona el correspondiente a la empresa proveedora y pulsa el botón 'Asociar'. De esta forma la imagen queda asociada al contacto.

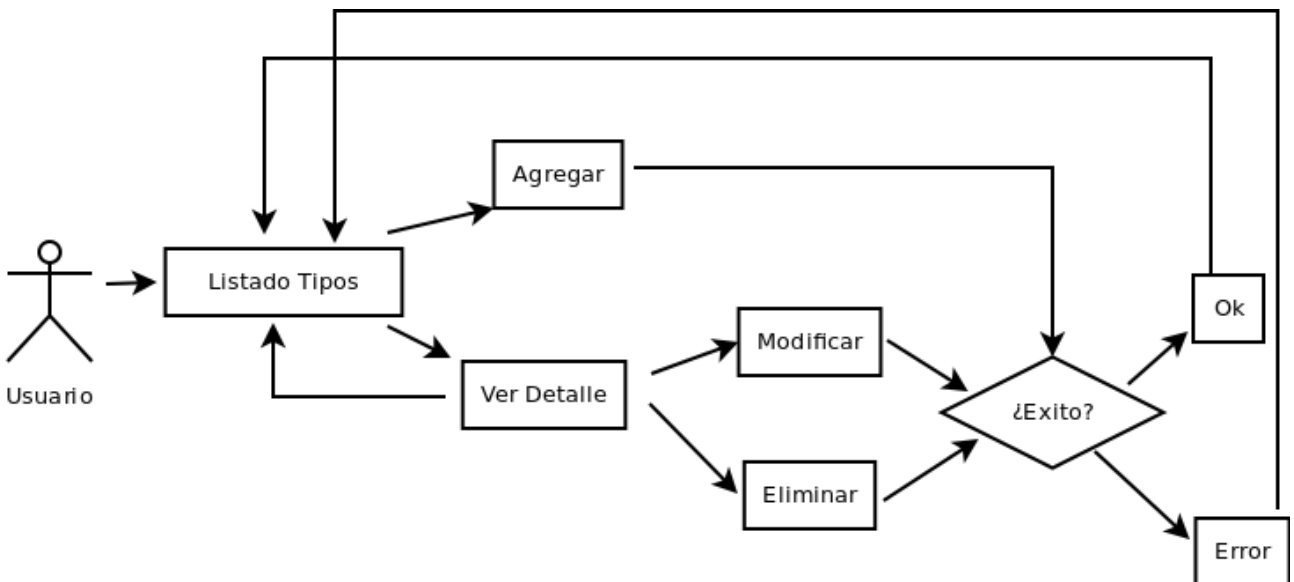
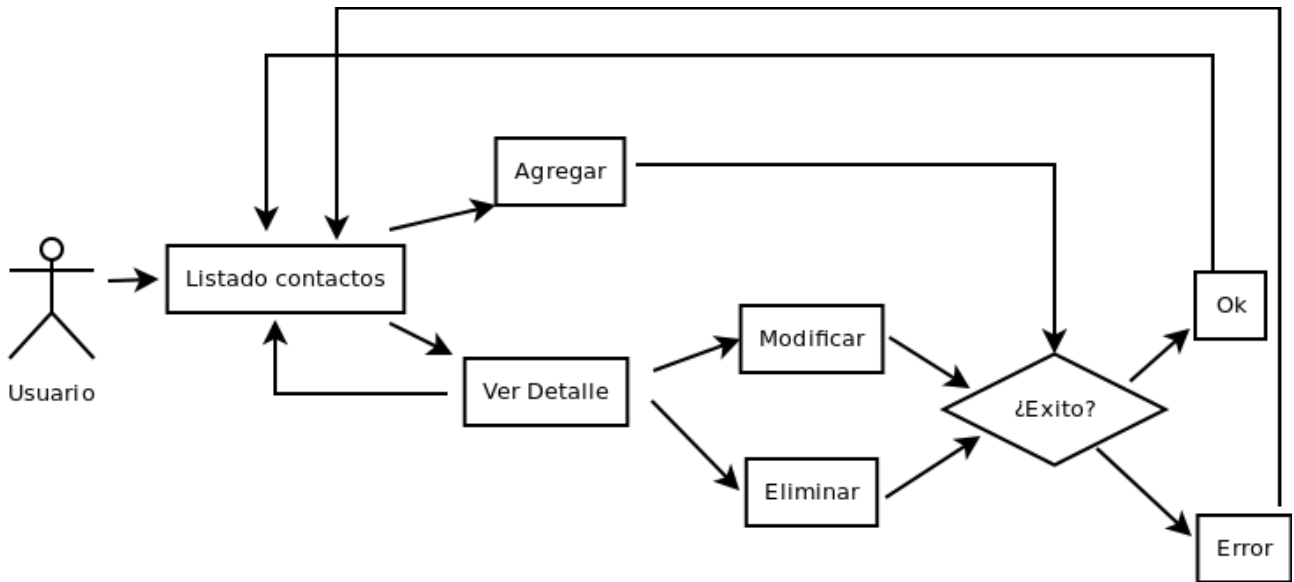
Daniel ha tomado una nueva imagen del producto donde se ven mucho mejor los detalles, por lo que quiere eliminar la imagen antigua. Así que pulsa la opción 'Imágenes' del menú principal y selecciona la imagen que quiere borrar del listado que se muestra por pantalla. Luego pulsa el botón 'Eliminar' y tras confirmar que quiere llevar a cabo la operación, la imagen queda eliminada del sistema.

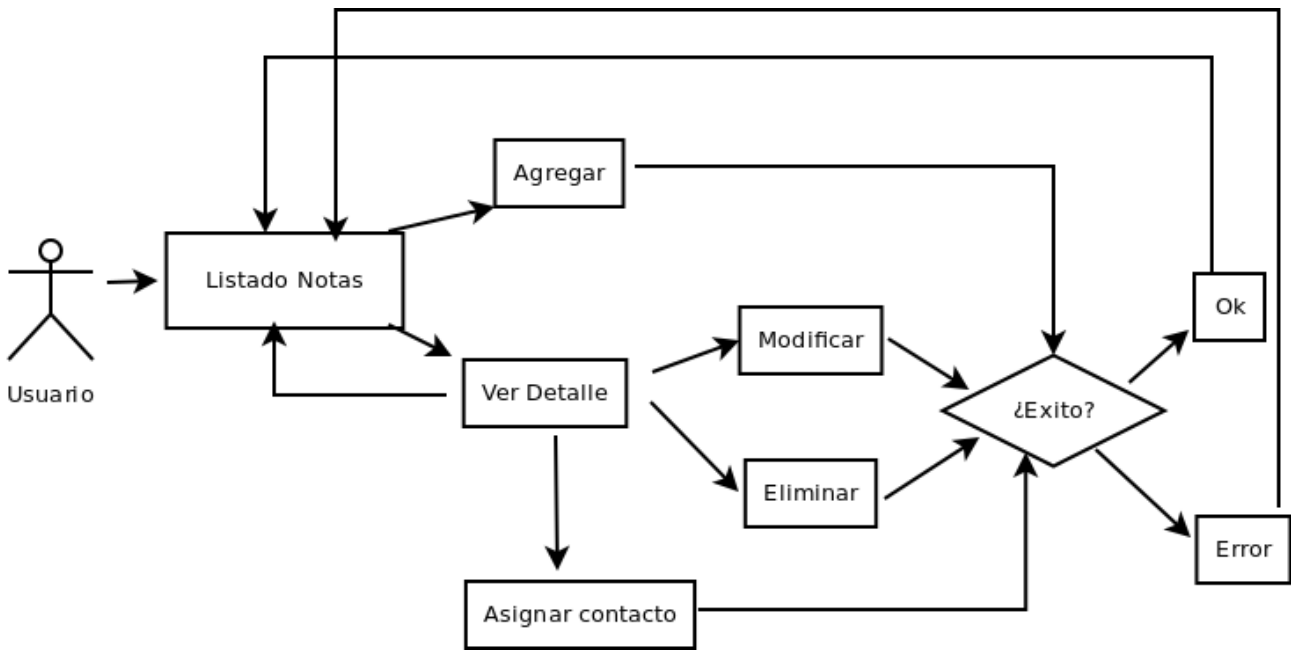
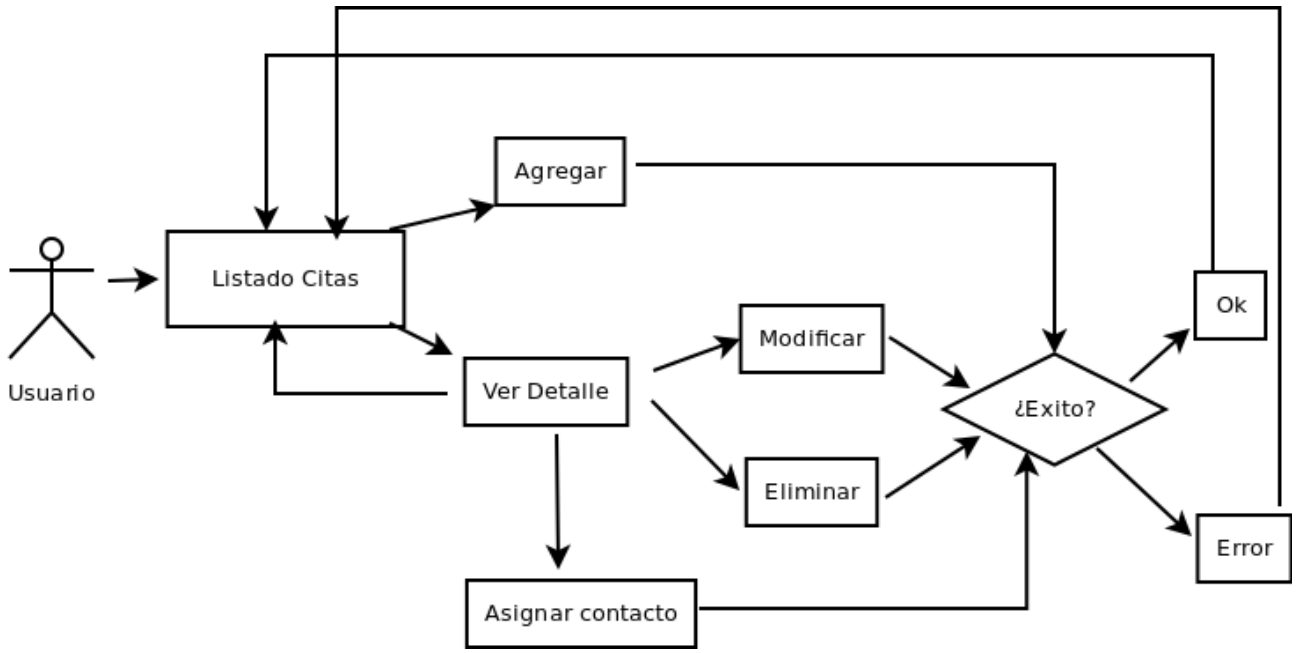
- Escenario para el uso de la funcionalidad de geolocalización

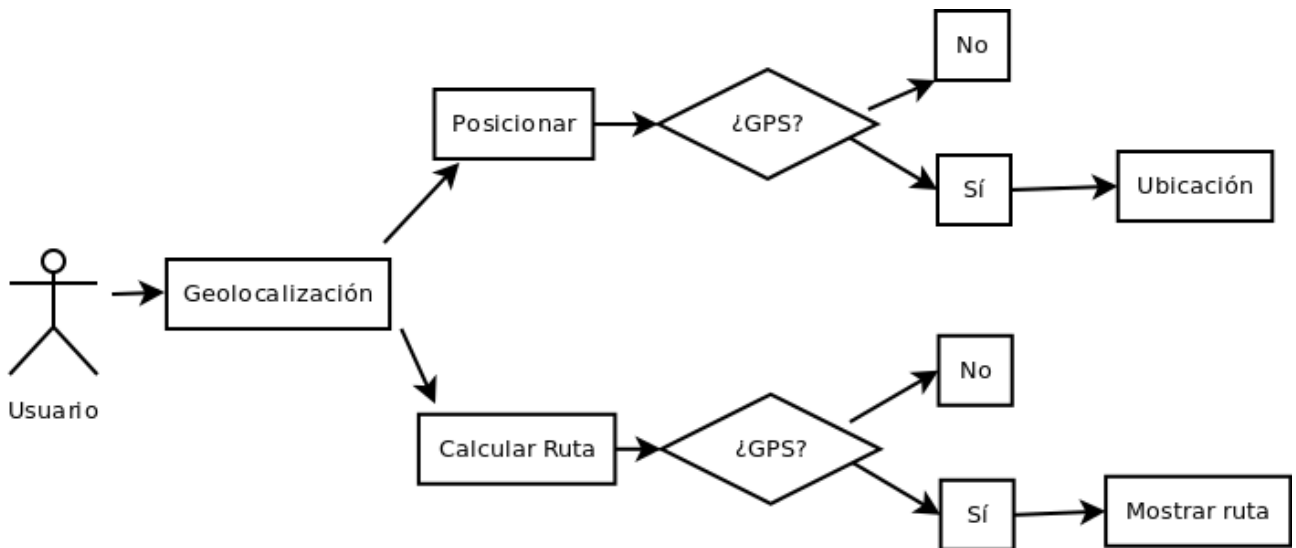
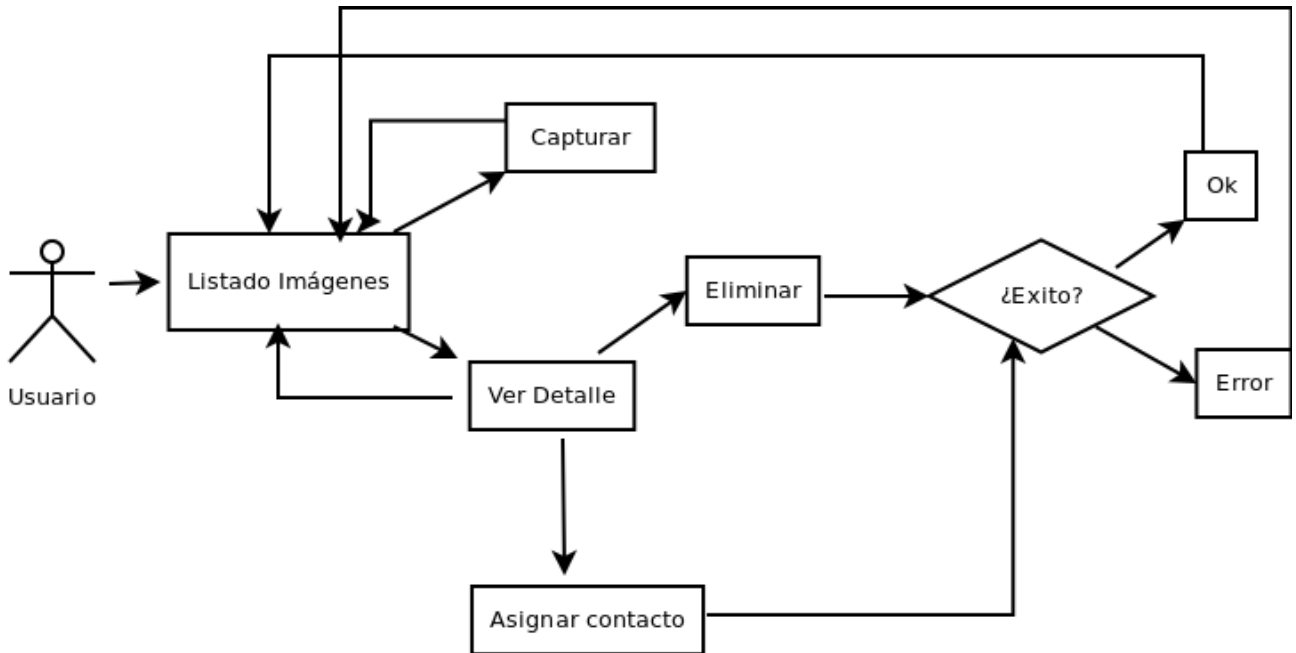
Daniel tiene una cita en el polígono industrial de una ciudad que no conoce. Necesita saber su ubicación, así que activa el GPS de su terminal y abre la aplicación. Pulsa la opción 'Geolocalización' y a continuación selecciona 'Posicionar'. Introduce los datos de la ubicación y pulsa 'Aceptar'. La aplicación muestra sobre el mapa su localización exacta.

Pero Daniel no sabe llegar desde el lugar donde se encuentra. Necesita calcular una ruta que le permita llegar al polígono industrial. Así que, en el menú principal, pulsa la opción 'Geolocalización' y luego selecciona 'Calcular Ruta'. Introduce la dirección de origen y la de destino y pulsa 'Calcular ruta'. La aplicación muestra sobre el mapa la ruta calculada.

4.2.2 Flujos de interacción

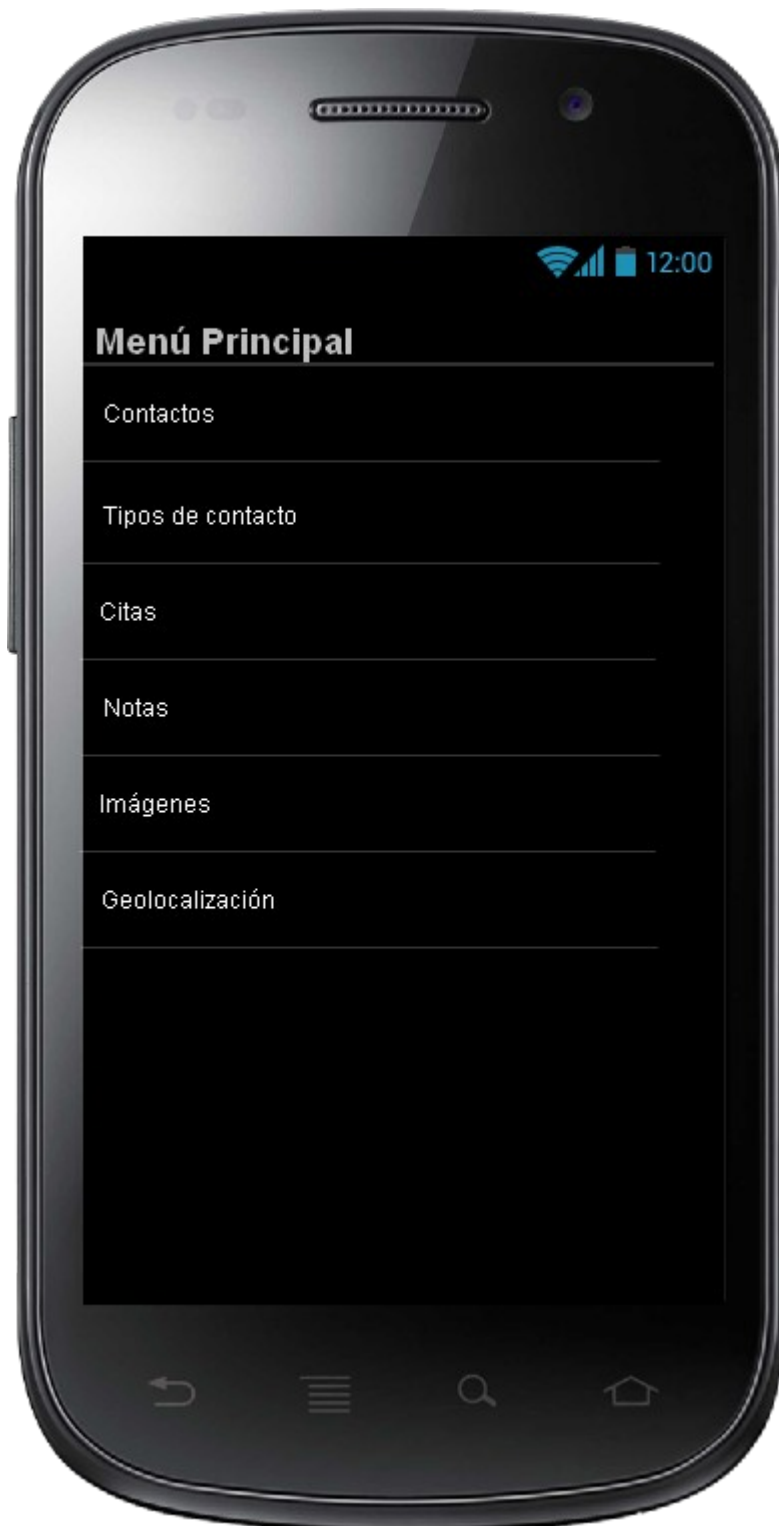




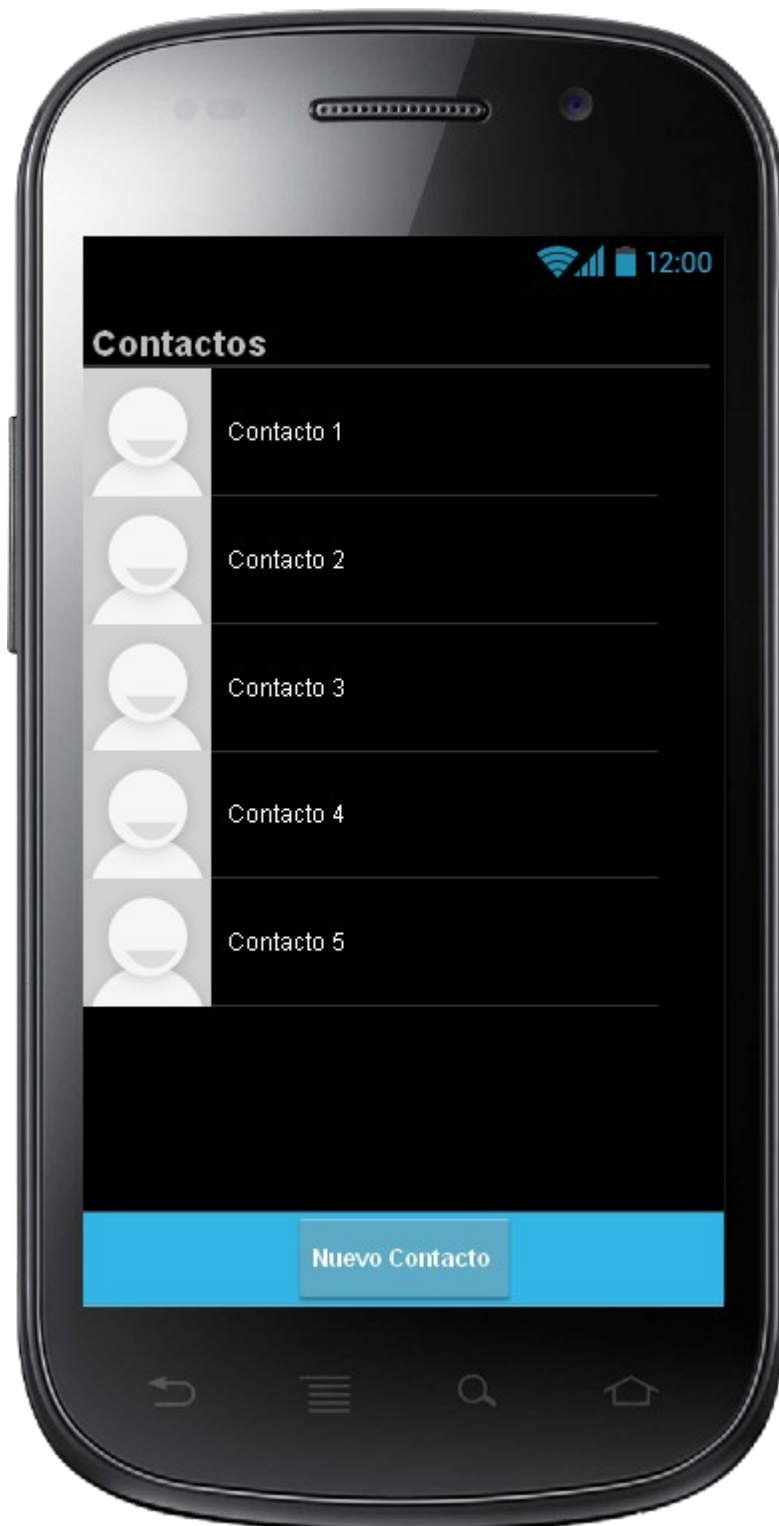


4.3 Prototipado

Interfaz Menú Principal



Interfaz Listado Contactos



Interfaz Nuevo Contacto



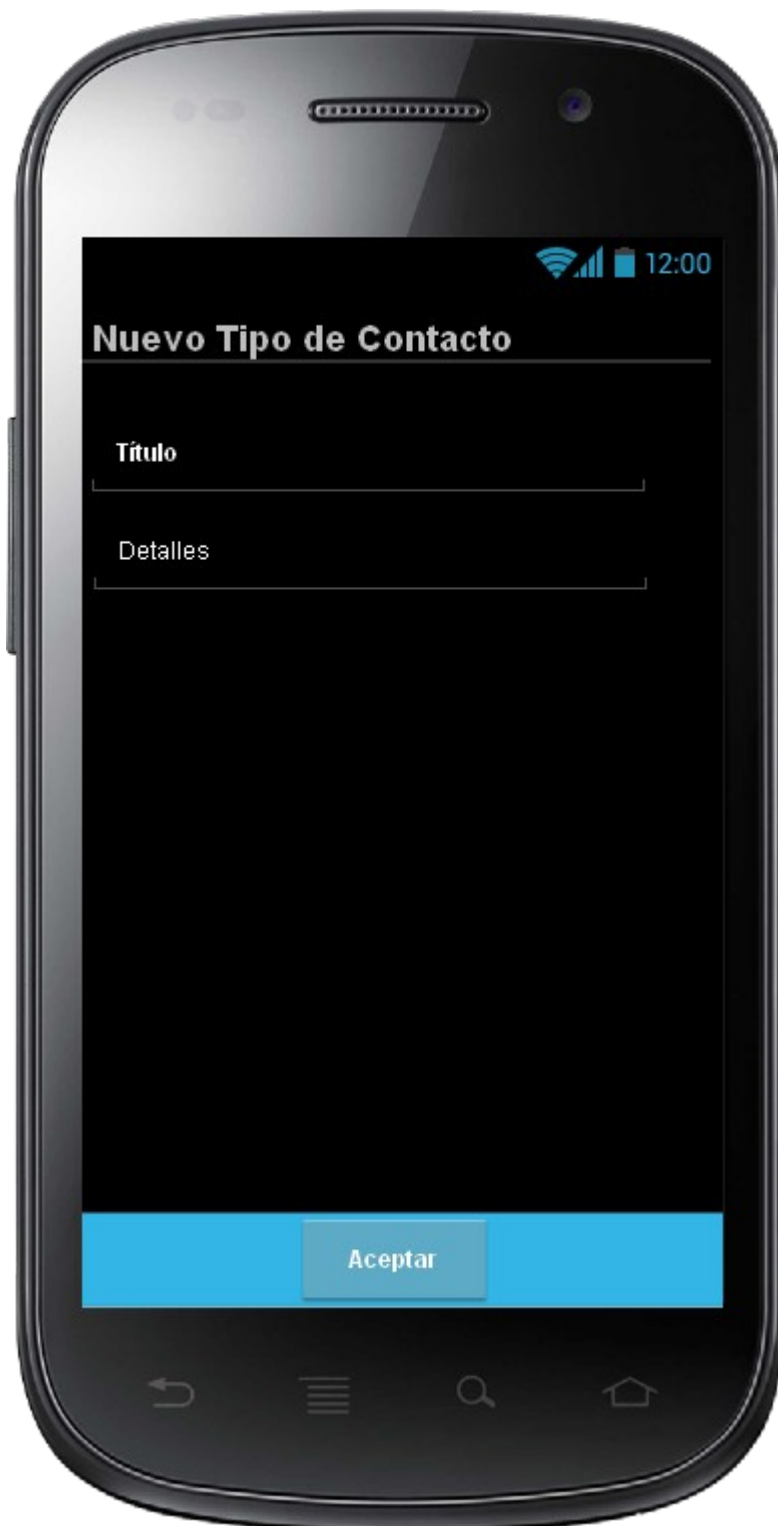
Interfaz Ver Detalle de contacto



Interfaz Listado Tipos de Contacto



Interfaz Nuevo Tipo de Contacto



Interfaz Ver Detalle de Tipo de Contacto



Interfaz Listado Citas



Interfaz Nueva Cita



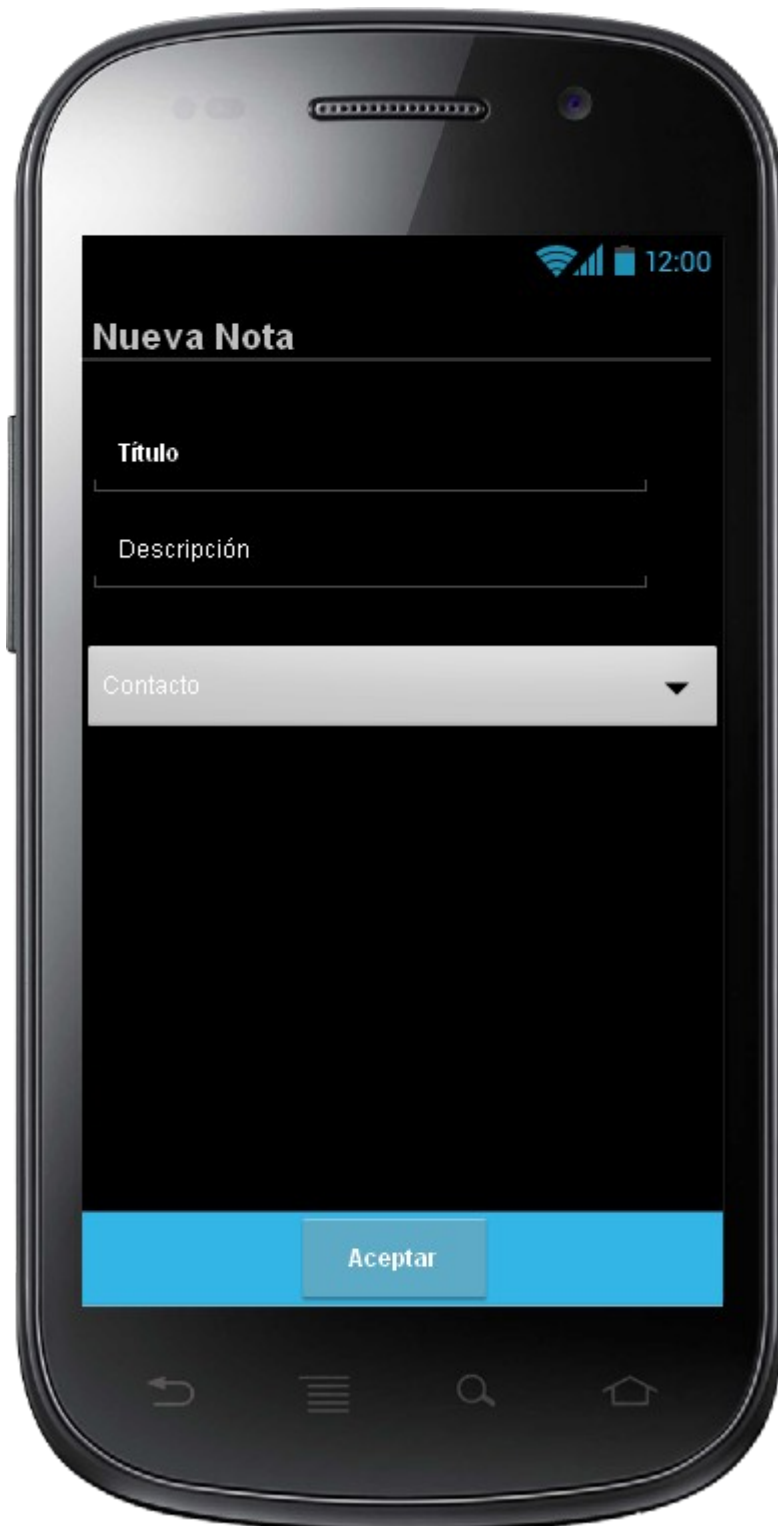
Interfaz Ver Detalle de Cita



Interfaz Listado Notas



Interfaz Nueva Nota



Interfaz Ver Detalla de Nota



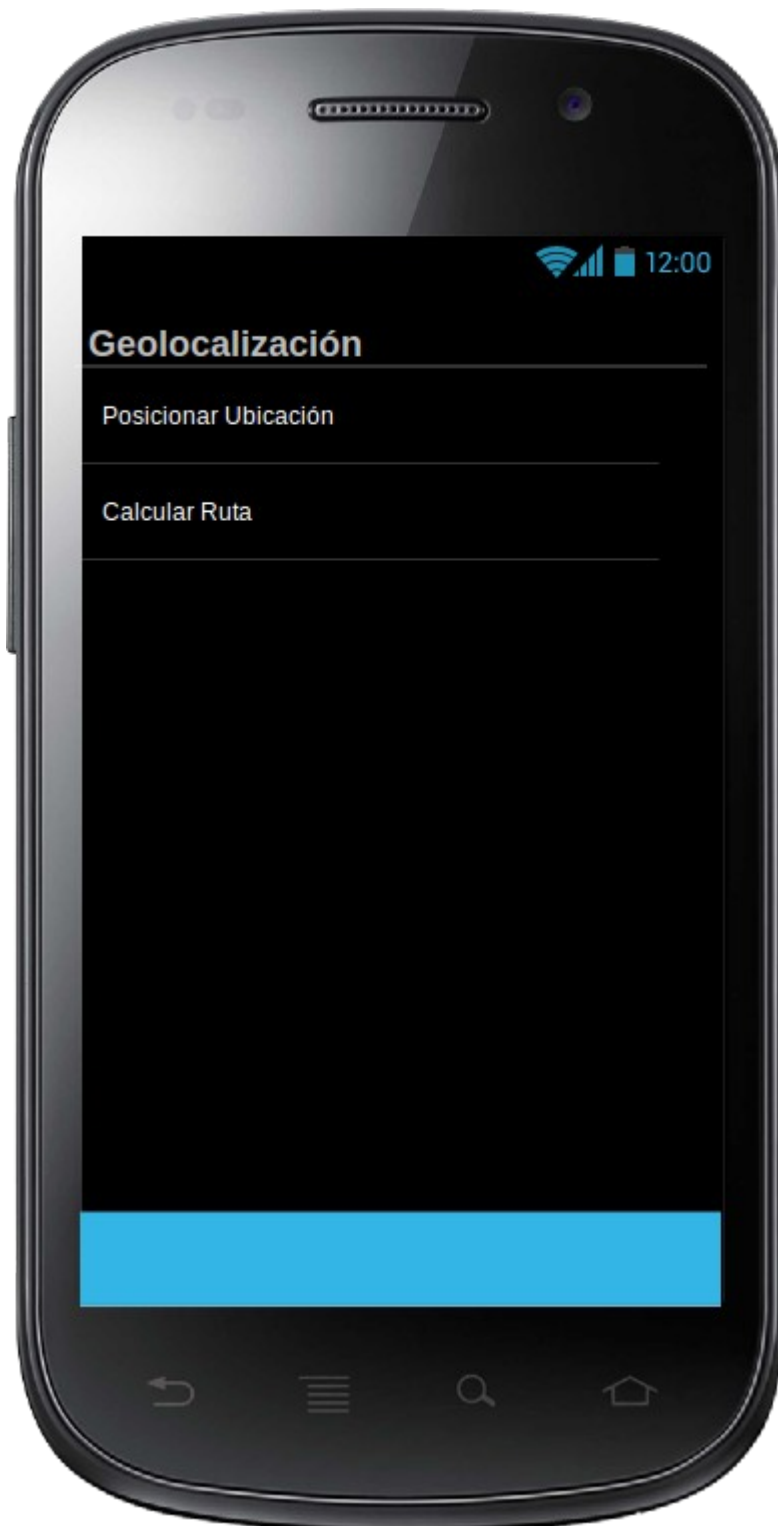
Interfaz Listado Imágenes



Interfaz Ver Detalle de Imagen



Interfaz Menú de Geolocalización



Interfaz Posicionar Ubicación



Interfaz Calcular Ruta



4.4 Evaluación final

Como último paso en el proceso de Diseño Centrado en el Usuario (DCU) es importante realizar una evaluación del prototipo obtenido de forma que podamos corregir los errores detectados. Para llevarla a cabo nos ayudaremos de la técnica de 'test de usuario' en la que pediremos a un número determinado de usuarios finales que lleven a cabo una serie de tareas concretas en la aplicación. Después les pediremos que respondan a una batería de preguntas con el fin de poder evaluar su experiencia de usuario, detectar posibles errores en el prototipo y, si fuera necesario, corregirlos.

En el caso concreto de nuestra aplicación hemos detectado que hay tareas que se repiten para diferentes funcionalidades, por lo que las interfaces de usuario son muy similares. Por ejemplo, añadir contacto, añadir cita, añadir tipo de contacto... son casi iguales y solo cambian los campos propios de los formularios. Igual para tareas como ver el listado de citas o los listados de notas, contactos, etc... Por lo tanto, para evaluar el prototipo consideramos que no será necesario que el usuario realice todas y cada una de las tareas de las que dispone la aplicación. Seleccionaremos unas cuantas tareas que sean representativas y que nos permita evaluar todas las funcionalidades, pero en el caso de interfaces idénticas pediremos al usuario realizar una sola tarea y su evaluación sobre ella la aplicaremos al resto.

A continuación se detallan las tareas elegidas que deberá realizar el usuario sobre la aplicación.

- Listado Citas (listado contactos)
- Añadir Nueva Cita
- Modificar Cita
- Eliminar Cita
- Asignar Cita a un Contacto
- Capturar Imagen
- Ver Detalle de Imagen
- Asignar Imagen a un Contacto
- Mostrar Ubicación
- Planificar Ruta

Junto a la realización de las tareas, los usuarios deben responder una serie de preguntas relacionadas con el manejo de la interfaz, y también con su perfil de usuario. Es decir, dividiremos las preguntas en dos grupos: aquellas que nos proporcionen información acerca del del usuario que realiza el test y aquellas que evalúen el prototipo como tal. Las preguntas sobre el usuario se deberán responder sólo una vez, ya que su respuesta no cambia a lo largo de la evaluación. Sin embargo, las referentes a la evaluación del prototipo, se deberán responder de forma individual para cada una de las tareas que lleve a cabo el usuario.

Preguntas para obtener información acerca del usuario

- Edad
- Sexo
- Nivel de estudios
- Empleo actual
- ¿Ha trabajado alguna vez como agente comercial? En caso afirmativo, años de experiencia.
- Nivel de conocimientos tecnológicos
- ¿Dispone de smartphone o tablet? En caso afirmativo, ¿cual es su uso principal? (ocio, trabajo, multimedia...)
- ¿Tiene experiencia en el manejo de otros dispositivos táctiles? ¿Cuales?
- ¿Dispone de una cámara de fotos digital? Experiencia en su uso
- ¿Dispone de un dispositivo GPS ? Experiencia en su uso

Preguntas para evaluar el prototipo

- Finalidad de la tarea realizada
- ¿Ha completado la tarea propuesta con éxito?
- Tiempo empleado para realizar la tarea
- Indique el nivel de dificultad que ha encontrado al realizar la tarea.

Muy fácil	Fácil	Normal	Difícil	Muy difícil
-----------	-------	--------	---------	-------------

- Indique el nivel de dificultad para encontrar las características solicitadas en los menús

Muy fácil	Fácil	Normal	Difícil	Muy difícil
-----------	-------	--------	---------	-------------

- Indique el nivel de dificultad para entender los mensajes mostrados por la aplicación

Muy fácil	Fácil	Normal	Difícil	Muy difícil
-----------	-------	--------	---------	-------------

- Indique los obstáculos principales que ha encontrado para realizar la tarea solicitada.
- Siendo 1 muy mala y 10 muy buena, puntúe su experiencia como usuario realizando la tarea solicitada.
- Indique cualquier propuesta de mejora que tenga sobre el proceso.
- Indique cualquier otro comentario que quiera realizar.

5 Implementación

5.1 Introducción

Como ya hemos mencionado anteriormente, para la implementación de la aplicación se ha utilizado la herramienta Phonegap. Su uso permite realizar la codificación en HTML5, CSS y Javascript. Para la creación de la interfaz gráfica se ha utilizado la librería JQuery Mobile, que es un framework Javascript que nos permite integrar componentes visuales con aspecto propio de las aplicaciones móviles, además de proporcionarnos toda la potencia de JQuery para manejar los objetos DOM del documento HTML.

5.2 Creación del proyecto

Para utilizar Phonegap, es necesario generar un proyecto de forma que lo pueda interpretar. Esto supone generar una jerarquía de carpetas y configurar ciertos ficheros. Es una tarea que la misma herramienta realiza desde un ejecutable que se distribuye junto a la librería. Por lo tanto, una vez descargado el archivo desde su página oficial y descomprimido en el directorio que se desee, habrá que ejecutar el comando

```
./create <ruta_carpeta_proyecto> <nombre_paquete> <nombre_proyecto>
```

El proyecto queda creado en <ruta_carpeta_proyecto>. Previamente tendremos que tener instalado el SDK de Android y de forma opcional, Eclipse o cualquier otro IDE o editor de texto. En este proyecto se ha utilizado Sublime Text 2, sin tener que realizar ninguna otra configuración adicional..

5.3 Estructura del proyecto

Dentro de la carpeta de proyecto, el código HTML de la aplicación se encuentra en la ruta

ruta_carpeta_proyecto/assets/www/

Para mayor claridad y comodidad a la hora de desarrollar, se han organizado los distintos componentes del proyecto en tres carpetas

- **/css** que contiene las hojas de estilo del proyecto. Es en este directorio donde se encuentra el css de jquery mobile que nos permite darle un aspecto visual propio de una aplicación móvil nativa.
- **/js** con todos los ficheros javascript que han sido necesarios, tanto los de desarrollo propio como los propios de jquery
- **/html** donde está todo el código HTML

Queda en la raíz un único fichero index.html que es la página principal de la aplicación y la que se ejecuta por defecto al iniciarse.

5.4 Componentes

5.4.1 Menú principal

Muestra un listado con todos los componentes de la aplicación. Este componente está compuesto por una vista situada en el raíz de `www` y que es la primera en ejecutarse por defecto al iniciar la aplicación

- **`ruta_del_proyecto/assets/www/index.html`**

La vista hace uso de la función `testBD()` al cargar el cuerpo del documento. Esta función se encuentra en

- **`ruta_del_proyecto/assets/www/js/init.js`**

`testBD()` se encarga de comprobar si existen las tablas SQLite necesarias para el funcionamiento de la aplicación, y si no existen las crea. Es decir, que contiene todos los scripts necesarios para generar la estructura de la base de datos. Una vez termina este proceso, se procede a contar cuantos registros tienen los componentes para mostrarlos en el listado de la vista. Si las tablas se acaban de generar, mostrará un 0 en todos ellos. Si ya existían y tenían registros, mostrará el total de los mismos.

5.4.2 Componente Contactos

Es el encargado de gestionar todos los contactos de la aplicación. Es decir, permite añadir un nuevo contacto, modificar uno existente, ver en detalle sus datos y/o eliminarlo de la aplicación.

Está compuesto por tres vistas que corresponden a los ficheros HTML

- **`ruta_del_proyecto/assets/www/html/contactosList.html`** que proporciona un listado con todos los contactos de la aplicación. Al cargar el cuerpo del documento realiza una llamada a la función `cargaListadoContactos()`
- **`ruta_del_proyecto/assets/www/html/contactosForm.html`** desde donde se permite añadir un nuevo registro. Esta vista llama a la función `desplegableTiposContacto()` al cargar el documento, y a la función `guardarContacto()` al pulsar el botón de guardar
- **`ruta_del_proyecto/assets/www/html/contactosDetalle.html`** que muestra todos los datos almacenados del contacto y permite modificarlos o eliminarlo por completo del sistema. Al cargar el cuerpo del documento llama a dos funciones: `desplegableTiposContacto()` y `cargaDetallesContacto()`. Asociada al botón de modificar está la llamada a la función `modificarContacto()`
Al pulsar el botón de eliminar se carga una ventana emergente con un mensaje de confirmación sobre la acción. En caso de pulsar Cancelar se llama a la función `cancelarContacto()` que hace una redirección al listado de de contactos sin eliminar el registro. En caso de aceptar se llama a la función `eliminarContacto()`

Todas las funciones que se utilizan en las vistas se encuentran en dos ficheros javascript. Uno es el propio del componente y otro es el del componente Tipos de Contacto

- **`ruta_del_proyecto/assets/www/js/tiposContacto.js`** contiene la función `desplegableTiposContacto()` que devuelve un componente HTML con un listado de todos los tipos de contacto que están dados de alta en la aplicación.
- **`ruta_del_proyecto/assets/www/js/contactos.js`** contiene los atributos del componente y el resto de funciones que se han mencionado en las vistas. Sus nombres describen sus funcionalidades básicas y todas ellas trabajan con la tabla `Contactos` de la base de datos, añadiendo, modificando o

eliminando registros. La excepción es la función *validaDatos()* que comprueba si los datos que se están intentando manejar son válidos o no. Si son correctos, devuelve un resultado vacío para que continúe la ejecución de la función que la ha llamado. Si no lo son, devuelve un mensaje de error para que sea mostrado por pantalla.

5.4.3 Componente Tipos de Contacto

Es el encargado de gestionar todos los tipos de contacto de la aplicación. Es decir, permite añadir un nuevo tipo de contacto, modificar uno existente, ver en detalle sus datos y/o eliminarlo de la aplicación. La principal funcionalidad de los tipos de contacto es permitir al usuario organizar sus contactos por categorías que el mismo defina, como por ejemplo contactos personales, agentes comerciales, administración, proveedores...

Está compuesto por tres vistas que corresponden a los ficheros HTML

- ***ruta_del_proyecto/assets/www/html/tiposContactoList.html*** que proporciona un listado con todos los tipos de contacto de la aplicación. Al cargar el cuerpo del documento realiza una llamada a la función *cargaListadoTiposContacto()* que a su vez llama a la función *listadoTiposContacto()* que es la encargada de interactuar con la BD y devolver los registros disponibles.
- ***ruta_del_proyecto/assets/www/html/tiposContactoForm.html*** desde donde se permite añadir un nuevo registro. Esta vista solamente tiene asociada la función *guardarTipoContacto()* al pulsar el botón de guardar.
- ***ruta_del_proyecto/assets/www/html/tiposContactoDetalle.html*** que muestra todos los datos almacenados del tipo de contacto y permite modificarlos o eliminarlo por completo del sistema. Al cargar el cuerpo del documento llama a la función *cargaDetallesTipoContacto()*. Asociada al botón de modificar está la llamada a la función *modificarTipoContacto()*. Al pulsar el botón de eliminar se carga una ventana emergente con un mensaje de confirmación sobre la acción. En caso de pulsar Cancelar se llama a la función *cancelarTipoContacto()* que hace una redirección al listado de tipos de contactos sin eliminar el registro. En caso de aceptar se llama a la función *eliminarTipoContacto()*

Todas las funciones que se utilizan en las vistas se encuentran en dos ficheros javascript. Uno es el propio del componente y otro es un contenedor de funciones de utilidad que pueden ser utilizadas por cualquier componente.

- ***ruta_del_proyecto/assets/www/js/utills.js*** contiene la función *gup()* que permite enviar y recibir parámetros por GET desde javascript. Lo utilizamos para enviar la id del componente seleccionado.
- ***ruta_del_proyecto/assets/www/js/tiposContacto.js*** contiene los atributos del componente y el resto de funciones que se han mencionado en las vistas. Sus nombres describen sus funcionalidades básicas y todas ellas trabajan con la tabla *Tipocontacto* de la base de datos, añadiendo, modificando o eliminando registros. La excepción es la función *validaDatos()* que comprueba si los datos que se están intentando manejar son válidos o no. Si son correctos, devuelve un resultado vacío para que continúe la ejecución de la función que la ha llamado. Si no lo son, devuelve un mensaje de error para que sea mostrado por pantalla.

5.4.4 Componente Citas

Es el encargado de gestionar todas las citas de la aplicación. Es decir, permite añadir una nueva cita, modificar una existente, ver en detalle sus datos y/o eliminarla de la aplicación. Las citas se pueden asociar a un contacto de la aplicación.

Está compuesto por tres vistas que corresponden a los ficheros HTML

- **ruta_del_proyecto/assets/www/html/citasList.html** que proporciona un listado con todas las citas de la aplicación. Al cargar el cuerpo del documento realiza una llamada a la función *cargaListadoCitas()* que a su vez llama a la función *listadoCitas()* que es la encargada de interactuar con la BD y devolver los registros disponibles.
- **ruta_del_proyecto/assets/www/html/citasForm.html** desde donde se permite añadir un nuevo registro. Al cargar el cuerpo del documento se produce una llamada a la función *desplegableContactos()* que devuelve un desplegable con todos los contactos existentes en la aplicación. Además tiene asociada la función *guardarCita()* al pulsar el botón de guardar.
- **ruta_del_proyecto/assets/www/html/citaDetalle.html** que muestra todos los datos almacenados de la cita y permite modificarlos o eliminarla por completo del sistema. Al cargar el cuerpo del documento llama a dos funciones *desplegableContactos()* y *cargaDetallesCita()*. Asociada al botón de modificar está la llamada a la función *modificarCita()*. Al pulsar el botón de eliminar se carga una ventana emergente con un mensaje de confirmación sobre la acción. En caso de pulsar Cancelar se llama a la función *cancelarCita()* que hace una redirección al listado de citas sin eliminar el registro. En caso de aceptar se llama a la función *eliminarCita()*

Todas las funciones que se utilizan en las vistas se encuentran en tres ficheros javascript. Uno es el propio del componente, otro el de utilidades mencionado en el componente Tipos de Contacto y por último el del componente Contactos para cargar el listado con todos los contactos disponibles en la BD.

- **ruta_del_proyecto/assets/www/js/utills.js** contiene la función *gup()* que permite enviar y recibir parámetros por GET desde javascript. Lo utilizamos para enviar la id del componente seleccionado.
- **ruta_del_proyecto/assets/www/js/contactos.js** que contiene la función *desplegableContactos()*
- **ruta_del_proyecto/assets/www/js/citas.js** contiene los atributos del componente y el resto de funciones que se han mencionado en las vistas. Sus nombres describen sus funcionalidades básicas y todas ellas trabajan con la tabla *Cita* de la base de datos, añadiendo, modificando o eliminando registros. La excepción es la función *validaDatosCitas()* que comprueba si los datos que se están intentando manejar son válidos o no. Si son correctos, devuelve un resultado vacío para que continúe la ejecución de la función que la ha llamado. Si no lo son, devuelve un mensaje de error para que sea mostrado por pantalla.

5.4.5 Componente Notas

Es el encargado de gestionar todas las notas de la aplicación. Es decir, permite añadir una nueva nota, modificar una existente, ver en detalle sus datos y/o eliminarla de la aplicación. Las notas se pueden asociar a un contacto de la aplicación.

Está compuesto por tres vistas que corresponden a los ficheros HTML

- **ruta_del_proyecto/assets/www/html/notasList.html** que proporciona un listado con todas las notas de la aplicación. Al cargar el cuerpo del documento realiza una llamada a la función *cargaListadoNotas()* que a su vez llama a la función *listadoNotas()* que es la encargada de interactuar con la BD y devolver los registros disponibles.
- **ruta_del_proyecto/assets/www/html/notasForm.html** desde donde se permite añadir un nuevo registro. Al cargar el cuerpo del documento se produce una llamada a la función *desplegableContactos()* que devuelve un desplegable con todos los contactos existentes en la aplicación. Además tiene asociada la función *guardarNota()* al pulsar el botón de guardar.

- **ruta_del_proyecto/assets/www/html/notaDetalle.html** que muestra todos los datos almacenados de la nota y permite modificarlos o eliminarla por completo del sistema. Al cargar el cuerpo del documento llama a dos funciones *desplegableContactos()* y *cargaDetallesNota()*. Asociada al botón de modificar está la llamada a la función *modificarNota()*. Al pulsar el botón de eliminar se carga una ventana emergente con un mensaje de confirmación sobre la acción. En caso de pulsar Cancelar se llama a la función *cancelarNota()* que hace una redirección al listado de notas sin eliminar el registro. En caso de aceptar se llama a la función *eliminarNota()*

Todas las funciones que se utilizan en las vistas se encuentran en tres ficheros javascript. Uno es el propio del componente, otro el de utilidades mencionado en el componente Tipos de Contacto y por último el del componente Contactos para cargar el listado con todos los contactos disponibles en la BD.

- **ruta_del_proyecto/assets/www/js/utills.js** contiene la función *gup()* que permite enviar y recibir parámetros por GET desde javascript. Lo utilizamos para enviar la id del componente seleccionado.
- **ruta_del_proyecto/assets/www/js/contactos.js** que contiene la función *desplegableContactos()*
- **ruta_del_proyecto/assets/www/js/notas.js** contiene los atributos del componente y el resto de funciones que se han mencionado en las vistas. Sus nombres describen sus funcionalidades básicas y todas ellas trabajan con la tabla *Nota* de la base de datos, añadiendo, modificando o eliminando registros. La excepción es la función *validaDatosNotas()* que comprueba si los datos que se están intentando manejar son válidos o no. Si son correctos, devuelve un resultado vacío para que continúe la ejecución de la función que la ha llamado. Si no lo son, devuelve un mensaje de error para que sea mostrado por pantalla.

5.4.6 Componente Imágenes

Es el encargado de gestionar todas las imágenes de la aplicación. Es decir, permite añadir una nueva imagen, modificar una existente, ver en detalle sus datos y/o eliminarla de la aplicación. Las imágenes se pueden asociar a un contacto de la aplicación.

Está compuesto por tres vistas que corresponden a los ficheros HTML

- **ruta_del_proyecto/assets/www/html/imagenesList.html** que proporciona un listado con todas las imágenes de la aplicación. Al cargar el cuerpo del documento realiza una llamada a la función *cargaListadoNotas()* que a su vez llama a la función *listadoImagenes()* que es la encargada de interactuar con la BD y devolver los registros disponibles.
- **ruta_del_proyecto/assets/www/html/imagenesForm.html** desde donde se permite añadir un nuevo registro. Al cargar el cuerpo del documento se produce una llamada a la función *desplegableContactos()* que devuelve un desplegable con todos los contactos existentes en la aplicación. En esta vista tenemos dos opciones para añadir una imagen. Al pulsar el botón Capturar Imagen se hace una llamada a la función *capturePhoto()* y esta accede a la cámara del dispositivo para hacer una nueva foto. Asociada al botón Desde Galería está la función *getPhoto()* a la que le pasamos el parámetro *pictureSource.SAVEDPHOTOALBUM* que es un método propio de la API de Phonegap para acceder a las galerías del dispositivo y poder seleccionar una de las imágenes ya existentes. Además tiene asociada la función *guardarNota()* al pulsar el botón de guardar.
- **ruta_del_proyecto/assets/www/html/imagenDetalle.html** que muestra todos los datos almacenados de la imagen y permite modificarlos o eliminarla por completo del sistema. Al cargar el cuerpo del documento llama a dos funciones *desplegableContactos()* y *cargaDetallesImagen()*. Asociada al botón de modificar está la llamada a la función *modificarImagen()*. Al pulsar el botón de eliminar se carga una ventana emergente con un mensaje de confirmación

sobre la acción. En caso de pulsar Cancelar se llama a la función *cancelarImagen()* que hace una redirección al listado de notas sin eliminar el registro. En caso de aceptar se llama a la función *eliminarImagen()*

Todas las funciones que se utilizan en las vistas se encuentran en tres ficheros javascript. Uno es el propio del componente, otro el de utilidades mencionado en el componente Tipos de Contacto y por último el del componente Contactos para cargar el listado con todos los contactos disponibles en la BD.

- ***ruta_del_proyecto/assets/www/js/utils.js*** contiene la función *gup()* que permite enviar y recibir parámetros por GET desde javascript. Lo utilizamos para enviar la id del componente seleccionado.
- ***ruta_del_proyecto/assets/www/js/contactos.js*** que contiene la función *desplegableContactos()*
- ***ruta_del_proyecto/assets/www/js/camara.js*** contiene los atributos del componente y el resto de funciones que se han mencionado en las vistas. Sus nombres describen sus funcionalidades básicas. En este fichero tenemos algunas que se encargan de interactuar con la tabla Imagen de la base de datos, añadiendo, modificando o eliminando registros, y otras que acceden a la API de Phonegap para trabajar con componentes nativos del dispositivo como la cámara de fotos o las galerías de imágenes del sistema. *onDeviceReady()*, *onPhotoDataSuccess()*, *onPhotoURISuccess()*, *capturePhoto()* y *getPhoto()* son las funciones que acceden a la API de Phonegap.

5.4.7 Componente Geolocalización

Es el encargado de ofrecer las funcionalidades de geolocalización haciendo uso del receptor GPS del dispositivo y una conexión a internet (wifi, 3G, etc...)

Está compuesto por seis vistas que corresponden a los ficheros HTML

- ***ruta_del_proyecto/assets/www/html/geolocalizacion.html*** que proporciona un listado con todas las funcionalidades del componente Geolocalización. Este listado sirve como menu principal del componente y es la primera vista que se cargará siempre que se acceda a él.
- ***ruta_del_proyecto/assets/www/html/geolocalizacionActual.html*** muestra la ubicación actual del usuario en un mapa utilizando la API de Google Maps. En la vista se hace uso además de funciones de geolocalización de la API de Phonegap para conocer la posición actual y obtener las coordenadas de la misma.
Al cargar el cuerpo del documento se llama a la función *inicio()* que comprueba si se tiene acceso a la ubicación actual del usuario utilizando la función de Phonegap *geolocation.getCurrentPosition()*. Si se tiene acceso se continua la ejecución, de lo contrario se muestra un mensaje de error.
- ***ruta_del_proyecto/assets/www/html/geolocalizacionBuscar.html*** muestra una caja de texto donde el usuario introduce la ubicación que desea posicionar en el mapa y el botón Buscar que tiene asociada una llamada a la función *buscarDireccion()*. La vista muestra también un mapa con una ubicación por defecto. Este mapa es el que mostrará la ubicación introducida por el usuario una vez se pulse Buscar.
- ***ruta_del_proyecto/assets/www/html/geolocalizacionBuscarMapa.html*** no es una vista como tal, sino más bien un fichero html auxiliar de *geolocalizacionBuscar.html* y que será el encargado de hacer todas las llamadas a las API de Phonegap y Google Maps para mostrar las ubicaciones en el mapa. Al cargar el documento se hace una llamada a la función *codeAddress()* que inicializa el mapa con una dirección por defecto y lo pinta por pantalla. Si a esta función se le pasa un valor en el parametro *direccion* recibido por GET en lugar de una cadena vacía, la ubicación se realiza sobre la dirección que ha introducido el usuario y se muestra esta en lugar de la de por defecto.

- ***ruta_del_proyecto/assets/www/html/geolocalizacionRuta.html*** muestra dos cajas de texto donde el usuario introduce las ubicaciones de origen y destino sobre las que desea obtener la ruta. El botón Calcular tiene asociada una llamada a la función *calcularRuta()*. La vista muestra también un mapa con una ubicación por defecto. Este mapa es el que mostrará la ruta una vez calculada.
- ***ruta_del_proyecto/assets/www/html/geolocalizacionRutaMapa.html*** igual que sucede con *geolocalizacionBuscarMapa.html* no es una vista como tal, sino un fichero html auxiliar de *geolocalizacionRuta.html* desde el que se hacen todas las llamadas a las API de Phoneygap y Google Maps para calcular y mostrar las rutas en el mapa. Al cargar el documento se hace una llamada a la función *calcRoute()* que inicializa el mapa con una ruta por defecto y lo pinta por pantalla. Si a esta función se le pasa dos valores en los parámetros origen y destino, recibidos por GET, la ruta se calcula sobre las ubicaciones introducidas por el usuario y se muestra esta en lugar de la de por defecto.

Las vistas solamente utilizan llamadas a un fichero Javascript. Esto es así porque el resto de llamadas a funciones que utilizan las APIs de Phoneygap y GoogleMaps se hacen desde los propios ficheros HTML. Este componente dio problemas al hacer las llamadas a las dos apis y tener que combinar parámetros y resultados devueltos por los métodos, por lo que al intentar agruparlo en un fichero js propio del componente como en el resto daba errores de funcionamiento. Por falta de tiempo fue imposible depurarlos y corregir estos comportamientos, por lo que se utilizan las vistas 'auxiliares' desde las que hacer las llamadas directamente. El único fichero js utilizado es

- ***ruta_del_proyecto/assets/www/js/utills.js*** contiene la función *gup()* que permite enviar y recibir parámetros por GET desde javascript. Lo utilizamos para enviar la dirección introducida por el usuario.

6 Posibles mejoras

A lo largo de la realización del proyecto, en sus distintas fases, se han ido observando posibles mejoras que se podrían añadir a la aplicación final y que no se han realizado por falta de tiempo y/o recursos. Quedan anotadas para posibles futuras versiones de la misma y se detallan a continuación.

- Integrar la agenda de contactos del dispositivo móvil con la agenda de contactos de la aplicación, de forma que el usuario no tenga que añadir de nuevo los contactos de los que ya dispone en la memoria o en la SIM del teléfono uno a uno a la aplicación.
- Añadir botón para realizar llamadas directamente desde los contactos de la aplicación, sin tener que salir de la misma y efectuarla manualmente.
- Añadir una imagen en el perfil del contacto con posibilidad de efectuar la foto desde la propia aplicación en la pantalla de modificar o añadir contacto.
- Mostrar en el listado de citas y notas, la imagen del contacto asociado (si lo tiene) para una organización de las mismas más clara, sencilla y visual. Esta mejora debería ser posterior a la anteriormente propuesta .
- Enviar un email desde la aplicación a cualquier contacto, por ejemplo añadiendon un botón al lado de la dirección de correo en la pantalla de 'Detalles del contacto'.
- Poder asignar una imagen, una nota o una cita a más de un contacto. Esta funcionalidad sí se contempló desde un principio pero finalmente no se pudo implementar por falta de tiempo. Sería una mejora prioritaria para una futura nueva versión.

7 Conclusiones

En este proyecto se ha desarrollado por completo una aplicación móvil a medida, cumpliendo con las funcionalidades solicitadas en la fase de toma de requisitos. Para llevarlo a cabo se han intentado seguir las metodologías aprendidas en las asignaturas cursadas a lo largo de la carrera, detallando y documentando las fases propias de cualquier proyecto de desarrollo de software: toma de requisitos funcionales, planificación, análisis, diseño, prototipado, evaluación final, etc...

El poder llevar a cabo todo el proceso de principio a fin sobre una aplicación final real y funcional me ha permitido tomar conciencia de la importancia de todas y cada una de las etapas, algo que en las asignaturas cursadas ya se suponía pero en las que era difícil verlo globalmente por lo específico de sus PAC y/o Prácticas.

La elección del área del TFC tenía un objetivo personal, y era el poder familiarizarme con una tecnología que se muestra al alza, con un mercado en expansión y en la que cada vez se demandan más profesionales cualificados. El tener que desarrollar la aplicación para cualquier dispositivo móvil (aunque finalmente ha sido compilado para Android) me ha permitido documentarme más en concreto sobre estas ideas previas y confirmar que el desarrollo de aplicaciones para dispositivos móviles no sólo presenta un gran futuro, sino que ya es un prometedor presente.

En concreto, existe una gran demanda de desarrolladores para iOS (Apple) y Android, las dos tecnologías más potentes hoy por hoy, dejando muy atrás a otras plataformas como Symbian, Blackberry o Windows Mobile, que actualmente siguen muy por detrás en el mercado de teléfonos móviles.

¿Por qué entonces decidí desarrollar la aplicación en HTML5 estando tan claro que existen dos líderes destacados a nivel mundial? Para comprobar si realmente un lenguaje tan prometedor como es el nuevo estándar de HTML podía ser una alternativa al desarrollo nativo de cualquier plataforma. Y como impresión general creo que sí puede llegar a serlo en un futuro, cuando se finalice de definir el estándar de HTML5, los navegadores lo interpreten correctamente y los dispositivos en general estén optimizados para este lenguaje.

Actualmente, tras haber realizado la implementación del proyecto en HTML5 junto a CSS3, jQuery Mobile y un framework como Phonegap, pienso que es una muy buena herramienta con la que evitar tener que reescribir el mismo código para cada una de las plataformas para las que queremos lanzar la aplicación. Esto supone un gran ahorro en cuanto a tiempo y dinero. La aplicación del proyecto se entrega compilada para Android, pero el mismo código se podría haber compilado sin ningún tipo de problema para ejecutarse en iPhone, sin cambiar una sola línea de código y sin tener que pagar licencias o que tenga que ser necesario un Mac para su desarrollo.

Junto al argumento del desarrollo multiplataforma, otro elemento que me hace pensar que este tipo de desarrollos es una buena opción, es la aparición de herramientas como Phonegap, que dan acceso a gran parte de los componentes físicos del dispositivo como pueden ser el GPS, la cámara de fotos o el acelerómetro. Este pudiera ser el gran hueco que se encuentra HTML5 para el desarrollo de aplicaciones móviles, ya que, al ser interpretado en el navegador, no tiene acceso a este tipo de recursos. Pero Phonegap incorpora una capa intermedia entre el navegador y el dispositivo, accediendo al mismo y ofreciendo al desarrollador una API propia para poder hacer uso de él. Como contra, sí es destacable la ralentización de la ejecución del programa.

Para terminar me gustaría mencionar que, a nivel personal, la realización del proyecto ha sido una experiencia muy positiva. Me ha permitido investigar y aprender multitud de nociones y conceptos sobre este área que desconocía totalmente y que sin duda seguiré con atención de ahora en adelante. De echo, uno de los objetivos que me planteo a corto plazo, es aplicar las mejoras mencionadas en el anterior apartado a la aplicación final y hacerla pública para su descarga en Google Play.

8 Bibliografía

Algunas de las fuentes consultadas para elaborar el presente proyecto han sido:

- **Página oficial de Phonegap**
<http://phonegap.com/>
- **Página oficial de JQuery Mobile**
<http://jquerymobile.com/>
- **Página oficial de Eclipse**
<http://www.eclipse.org>
- **Android Developers**
<http://developer.android.com/sdk/index.html>
- **Google Maps API**
<https://developers.google.com/maps/?hl=es>
- **Blog de Jorge Iván Meza**
Primeros pasos con Phonegap para Android
<http://blog.jorgeivanmeza.com/2011/05/primeros-pasos-con-phonegap-para-android/>
- **Diario de un androide**
Tutorial de phonegap
<http://androide.hijodeblog.com/2012/06/19/tutorial-de-phonegap-parte-1-preparando-el-entorno/>