



Trabajo final de carrera

MEMORIA

Ramón Manzanares Rodríguez

A mi padre, que tras mucho esfuerzo no pudo  
ver como acababa todo esto...

A Maru, por su paciencia, cuidados y apoyo que me  
ha ofrecido durante todo este tiempo...

A mi familia en general, que siempre han estado ahí...

A la UOC por permitirme con su forma de  
educación finalizar mis estudios...

...MUCHISIMAS GRACIAS A TODOS ELLOS!!!

## Indice

1. Definición general del proyecto	5
1.1. Introducción	5
1.2. Definición del proyecto	6
1.3. Referencias	4
1.4. Características técnicas	7
1.4.1. De software	7
1.4.2. De hardware	7
2. Definición funcional	8
2.1. Características funcionales	8
2.2. Propiedades funcionales	9
2.3. Diseño interfaz gráfica	9
3. Planificación del proyecto	10
3.1. Lista de tareas	10
3.2. Diagrama de Gantt	11
4. Análisis y diseño del sistema	12
4.1. Usuarios y contextos de uso	12
4.1.1. Investigación	12
4.1.2. Perfiles de usuario	13
4.1.3. Contextos de uso	14
4.1.4. Análisis de tareas	14
4.2. Diseño conceptual	16
4.2.1. Escenarios de uso	16
4.2.2. Diagramas de casos de uso	19
4.2.3. Flujos de interacción	22
4.3. Prototipado	24
4.3.1. Sketches	24
4.3.2. Prototipo	24
5. Implementación	27
5.1 Organización de las clases implementadas	27
5.2 Gestión del posicionamiento	34
5.3 Gestión del almacenamiento	37

5.4 Gestión de datos entre actividades	39
6. Evaluación	40
6.1. Usuarios que realizarán el test	41
6.2. Preguntas sobre los usuarios	41
6.3. Tareas que realizar en el test	41
6.4. Preguntas sobre las tareas	41
7. Variación de la idea inicial	42
7.1 Futuras mejoras	43
8. Conclusiones	43
8.1 Valoración personal	44
9. Glosario	45
10. Bibliografía	49

## 1. Definición general del proyecto

### 1.1. Introducción

Este proyecto comprende el desarrollo de una aplicación para dispositivos móviles, concretamente para teléfonos móviles con SO android. He seleccionado la opción de realizar la aplicación con gestor de mapas y GPS, junto con la utilización/comunicación con una BBDD externa (se podría usar mobile backend as service (MBaaS)).

Tratará de la ubicación en mapas de sitios de interés del usuario, tanto de forma publica como privada.

La forma publica sera compartida con el resto de usuarios de la aplicación para crear una BBDD común para todos.

La forma privada sera almacenada en el propio dispositivo, y solo accesible por el usuario que añadió esta ubicación.

Esta aplicación va a ser bastante sencilla, tanto para el uso del cliente, como para el desarrollador de la misma. Se intentará la iniciación a la programación en android y las tecnologías móviles, como en el diseño de interfaces, programación en java (android), utilización de herramientas de los dispositivos como posicionamiento, consulta de datos en local y en remoto (conexión con servidor).

Se ha elegido este tipo de aplicación ya que se utilizan varios conceptos interesantes como la geolocalización, controles de hardware y la comunicación por red con BBDD.

### 1.2. Definición del proyecto

El nombre que he pensado, de forma provisional es “Whereis”.

La idea inicial se basa en la geolocalización, según el interés del propio usuario (de forma privada o publica) de ubicaciones de vehículos (donde aparcaste), domicilios de amigos, lugares de interés como restaurantes, tiendas, etc...

La diferencia entre la parte privada o publica es que la primera, los datos de ubicación se almacenaran en el propio dispositivo mientras que la opción publica se almacenara en una BBDD accesible por todos los usuarios de la aplicación.

Indicar que en la parte publica se añadirán sitios de interés publico como restaurantes, café, vida nocturna, zonas de compras, zonas deportivas, donde se podrá realizar una valoración del lugar. Esta valoración servirá para que si buscas en una zona determinada, la aplicación te dará información de lo lugares de interés cercanos, mostrándote primero

lo que posean mayor puntuación por parte de los usuarios.

Estos lugares se mostraran como símbolos en el mapa, diferenciados en la opción publica como en la privada, por ejemplo lugares de interés de color rojo serán de la opción privada, mientras que los símbolos de color azul serán los públicos.

### 1.3. Referencias

Una vez indicadas las principales características de la aplicación a desarrollar, buscamos aplicaciones del google play que tengan alguna similitud con lo indicado anteriormente.

Hay varias aplicaciones para saber donde aparcaste el coche (My parking, FindYourParking, ¿Donde aparqué?, Where have i parked?, etc...

Otras aplicaciones como 11870.com donde puedes localizar cualquier tipo de actividad como salud y belleza, servicios para el hogar, cultura, restaurantes, etc...La similitud es que podrías buscar cosas cercanas, y puedes ubicarlo en el mapa y ver opiniones de los usuarios de la aplicación.

A continuación realizo un análisis de algunas de las anteriores aplicaciones:

Análisis de aplicaciones existentes en el mercado				
Tipo	Datos	Funcionalidades	Interesa	No interesa
Localización de la ubicación de tu vehículo	Nombre: My parking	<ul style="list-style-type: none"> <li>- Muestra en el mapa tu ubicación actual.</li> <li>- Puedes marcar esta ubicación para indicar el aparcamiento de tu vehículo.</li> <li>- Crea rutas para llegar a tu coche.</li> <li>- Añade notas y datos mas concretos de la ubicación.</li> </ul>	<ul style="list-style-type: none"> <li>- Ubicación actual en el mapa.</li> <li>- Marcar esta ubicación.</li> <li>- Añadir notas y datos.</li> </ul>	<ul style="list-style-type: none"> <li>- Crear ruta para llegar a la ubicación marcada, podría ser motivo de actualización.</li> </ul>
	Desarrollador: Parinno			
	Valoración: 4/5 estrellas (4,1 promedio)			
	Web del fabricante: www.parinno.com			
	Precio: Gratis			
	Idioma: Español			

Descubre y comparte sitios, alrededor de donde te encuentres, con opiniones fotos y vídeos de miles de usuarios	<b>Nombre:</b> 11870.com	- Solicitud de usuario y contraseña (no es obligatoria para usar la aplicación).		
	<b>Desarrollador:</b> 11870.com	- Seleccionar las posibilidades de búsqueda de forma clasificada.		
	<b>Valoración:</b> 4/5 estrellas (3,8 promedio)	- Muestra resultados de búsqueda ordenado por cercanía.		
	<b>Web del fabricante:</b> 11870.com	- De cada resultado, muestra dirección, ubicación en mapa, como también poder subir una foto u opinar.		
	<b>Precio:</b> Gratis	- Contiene una BBDD amplia, con todos esos lugares.		
	<b>Idioma:</b> Español			

## 1.4. Características técnicas

### 1.4.1. De software

- El entorno de desarrollo a usar es Eclipse (mac y windows).
- SDK de Android (ADT bundle de mac y de windows)
- Sistema operativo Android 3.1 o superior

### 1.4.2. De hardware

- Para el desarrollo se utilizara Mac Os 10.5.8 o Windows xp
- Para las BBDD existe la duda de usar BBDD externas o mobile backend as service (MbaaS).
- Para las pruebas se dispone de varios móviles con distintos SO:
  - HTC desire HD, con SO Android Honeycomb 3.1
  - Samsung galaxy S3, con SO Android Jelly Bean 4.1.2

## 2. Definición funcional

### 2.1. Características funcionales

A continuación detallo una serie de requisitos funcionales:

Modulo a desarrollar	Funcionalidad requerida	Funcionalidad objetiva
Mapa basado en el API de google	Localización de lugares	Localizar puntos de interés privados o públicos
	Mostrar ubicaciones	Consultar ubicaciones almacenadas en la BBDD
	Configurar tipo de ubicación	Ubicaciones privadas o publicas
	Mostrar información sobre ubicaciones	Proporcionar información básica sobre los puntos de ubicación
	Aplicar acciones a ubicaciones publicas	Ampliación de información como nombre, dirección, puntuación
	Zoom	Aplicar zoom al mapa
Gestión del GPS	Conversiones coordenadas de geolocalización y direcciones (API de google)	Facilitar la ubicación de los sitios de interés y su interpretación por parte de los usuarios
Gestión de la BBDD	Gestión de usuarios	Almacenamiento de usuarios y contraseñas en la BBDD para el acceso a datos públicos
	Gestión de la cache local	Almacén de datos para el correcto funcionamiento de la aplicación en la parte privada
	Gestión de la BBDD externa (o en su lugar MbaaS)	Almacén de datos para el correcto funcionamiento de la aplicación en la parte publica



## 2.2. Propiedades funcionales

A continuación detallo el comportamiento de la aplicación en funcionamiento y sus necesidades operativas:

- Si se usar la BBDD en un servidor, este deberá estar accesible con sus servicios activos las 24h. Si se usara el mobile backend as service debería tener acceso a internet, en este caso a la nube.
- El uso de la aplicación es esporádico, cuando el usuario necesite usarla.
- El consumo de batería requerido por la aplicación es mínimo por el uso.
- Las actualizaciones de los datos corren a cargo de los propios usuarios.
- El diseño de la aplicación sera lo mas sencilla de manejo para el usuario.
- La aplicación controlara todo los posibles errores provocados por el usuario, y conseguir ser lo mas estable y robusta.
- Permisos para ejecutar la aplicación:
  - Acceso a la ubicación
  - Recepción de mapas (google maps)
  - Comunicación por red (Internet)
  - Controles de hardware
  - Almacenamiento en SD

## 2.3. Diseño interfaz gráfica

Lo primero que aparece al ejecutar la aplicación es el login o Registro en el sistema. Se podría usar la aplicación para uso local o privado si no se accede al sistema remoto.

Después del login, aparece un mapa con la situación actual localizado en el centro, y 3 botones, uno para “marcar” o añadir un nuevo lugar al sistema (publico o privado), otro botón “Mi ubicación” que realiza un zoom desde un mapa muy alejado hasta un mapa en tu ubicación, donde ya se pueden apreciar nombres de calles y un tercer botón para “buscar” lugares al rededor de la posición actual del usuario.

Si el usuario pulsa sobre el botón marcar, saldría una actividad que es la que añade datos a la memoria externa o a la BBDD. Aparece como predeterminado que la información va a ser publica, aunque el usuario puede ponerla como privada. Aparece también un menú desplegable o lista de opciones, donde puedes indicar que tipo de ubicación se trata, aparcamiento de coche, casa de amigos, restaurantes, cafetería, zona deportiva, etc... El usuario debe rellenar la información como nombre, dirección, notas u opinión y puntuación.

Indicar también la política de votos, que visualmente son 5 estrella, y puedes realizar añadir votos que con puntuación de 0 a 5. El ultimo botón seria el de guardar y almacenar los datos en local o remotamente.

Si en la actividad principal pulsas sobre el botón “Mi ubicación”, como se indica anteriormente se realiza un zoom que va desde un mapa mundial, a un mapa de calles.

Si en la actividad principal (donde el mapa) pulsaras el botón “BUSCAR” ejecutaríamos la siguiente ventana que muestro a continuación:

Primero aparecerían el tipo de filtro, que serían todas las posibles clasificaciones de las ubicaciones almacenadas, restaurantes, zonas de ocio, zonas deportivas, etc... Después de seleccionar el filtro, aparecerían de forma ordenada por votos y cercanía todas las ubicaciones almacenadas en la ubicación actual del usuario.

Podrías pulsar sobre cualquiera de ellas y entrar en otra ventana con información detallada sobre la misma, como nombre, dirección, geolocalización de la ubicación actual, este usuario podría puntuar esta ubicación, incluir unas notas u opinión sobre el lugar y ver otras notas u opiniones de otros usuarios.

### 3. Planificación del proyecto

#### 3.1. Lista de tareas

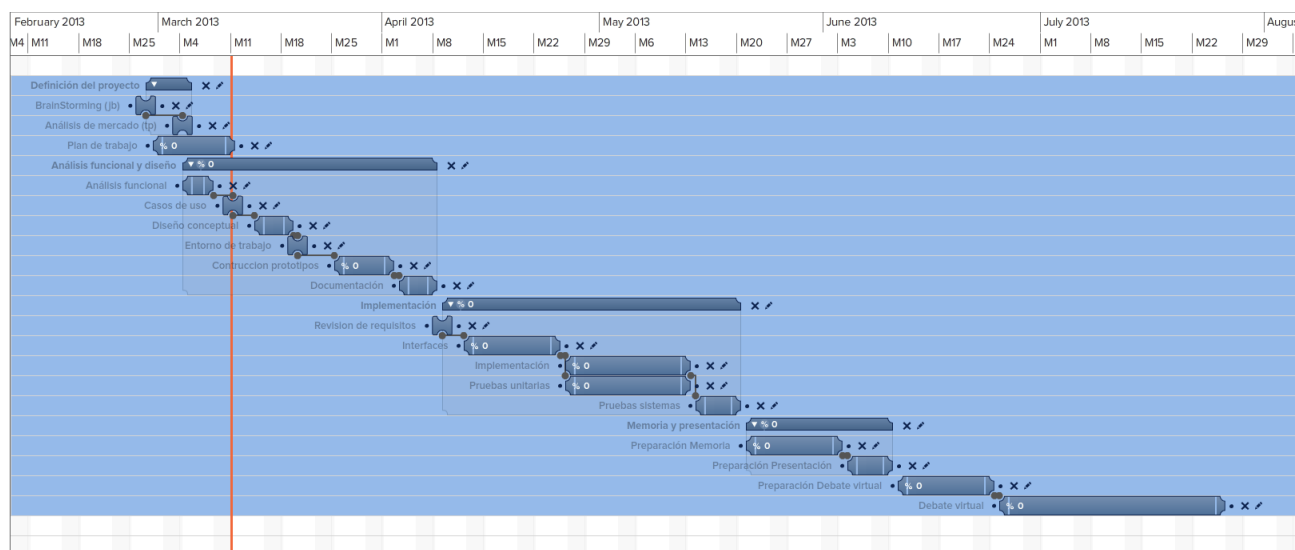
Planifico la lista de tareas a realizar para desarrollar el proyecto en la siguiente tabla:

	Tarea	Duración	Inicio	Fin
1	TFC “Whereis”	108 días	27/02/13	27/07/13
2	Definición del proyecto	5 días	27/02/13	05/03/13
3	Brainstorming	3 días	27/02/13	01/03/13
4	Análisis de mercado	2 días	04/03/13	05/03/13
5	Plan de trabajo (PAC1)	10 días	28/02/13	11/03/13
6	Elaboración	10 días	28/02/13	11/03/13
7	Entrega	0 días	11/03/13	11/03/13
8	Análisis funcional y diseño técnico (PAC 2)	27 días	04/03/13	08/04/13
9	Análisis funcional	5 días	04/03/13	08/03/13
10	Casos de uso	3 días	11/03/13	13/03/13
11	Diseño conceptual	4 días	14/03/13	19/03/13
12	Preparación entorno de trabajo	3 días	20/03/13	22/03/13
13	Construcción de prototipos	7 días	25/03/13	02/04/13
14	Elaboración de documentación	4 días	03/04/13	08/04/13
15	Entrega	0 días	08/04/13	08/04/13
16	Implementación (PAC 3)	30 días	09/04/13	20/05/13

17	Revisión de requisitos	3 días	09/04/13	11/04/13
18	Construcción de interfaces	10 días	12/04/13	25/04/13
19	Implementación del código	12 días	26/04/13	13/05/13
20	Pruebas unitarias	12 días	26/03/13	13/05/13
21	Pruebas de sistemas y revisión	5 días	14/05/13	20/05/13
22	Entrega	0 días	20/05/13	20/05/13
23	<b>Memoria y presentación del TFC</b>	<b>15 días</b>	<b>21/05/13</b>	<b>10/06/13</b>
24	Elaboración de la memoria	10 días	21/05/13	03/06/13
25	Elaboración de la presentación	5 días	04/06/13	10/06/13
26	Entrega final	0 días	10/06/13	10/06/13
27	Preparación del debate virtual	10 días	11/06/13	24/06/13
28	<b>Debate virtual</b>	<b>25 días</b>	<b>24/06/13</b>	<b>27/07/13</b>

### 3.2. Diagrama de Gantt

En la siguiente imagen se puede observar el diagrama de gantt en el que están reflejados las tareas a realizar:



\*Tuve que usar un editor de diagramas de gantt online ya que usando el mac la aplicación GanttProject 2.6.1.app no logre ejecutarla y actualmente tengo problemas con mac os 10.5.8 para conseguir aplicaciones de este tipo. Disculpen y si necesitan ver con mas detalle el diagrama, se los envié sin ningún inconveniente.

## 4. Análisis y diseño del sistema

El análisis y diseño del sistema se ha realizado acorde a la metodología DCU aprendida con la documentación facilitada por los profesores de la UOC. A continuación se detalla como se ha realizado esta fase.

### 4.1. Usuarios y contexto de uso

En esta primera fase de investigación, he utilizado la técnica de **investigación contextual**, ya que me permite observar al usuario mientras interactúa con un sistema parecido al que voy a diseñar.

#### 4.1.1. Investigación

Las aplicaciones elegidas para realizar la investigación han sido “My parking”, “11870.com” y “Guachapp”, todas obtenidas del google play de Android.

El dispositivo utilizado han sido un Samsung galaxy S3, con SO Android 4.1.2

A los usuarios (Maru Trujillo con 30 años, María del Carmen Rodríguez con 60 años, Alexis Hernández con 27 años) se les solicita realizar las siguientes tareas:

- TEST 1: Localizar su ubicación actual (Guachapp)
- TEST 2: Marcar estacionamiento actual de tu vehículo (My parking)
- TEST 3: Consultar centros de salud y belleza cercanos (11870.com)

Resultados de los TEST:

Usuario	Maru Trujillo
Perfil	Publicista, uso de dispositivos con Android desde hace ya años
Test 1	Exitosa, sin problemas
Test 2	Exitosa, sin problemas
Test 3	Duda dentro de la opción “salud y belleza” que pulsar para obtener resultados correctos del test
Conclusiones	Guachapp, crea dudas en localizar su ubicación

Usuario	María del Carmen Rodríguez
Perfil	Ama de casa, uso de dispositivos con Android desde hace menos de un año

Test 1	Tarda unos minutos en acertar con el dato que le solicito en el test
Test 2	Exitosa, sin problemas
Test 3	Exitosa, sin problemas
Conclusiones	Dentro de las dudas que le crea usar aplicaciones desconocidas, bastante rápido el uso de las 3 aplicaciones, exceptuando el test 1

Usuario	Alexis Hernández
Perfil	Técnico informático, desarrollador de aplicaciones multiplataforma
Test 1	Exitosa, sin problemas
Test 2	Exitosa, sin problemas
Test 3	Exitosa, sin problemas
Conclusiones	No conocía ninguna de las aplicaciones, indica que Guachapp es bastante lenta, pero rápida ejecución de las pruebas

#### 4.1.2. Perfiles de usuario

Después de realizar la investigación puedo afirmar que la experiencia con el uso de aplicaciones móviles no puede ser motivo para la diferenciación de perfiles de usuario. Esta claro que se necesita al menos una experiencia mínima con el uso de la tecnología a desarrollar, pero este no marca diferencia entre usuarios. Por diferencia demográfica de los usuarios no puedo realizar distinción alguna ya que, en principio, la aplicación va a ser de uso nacional. La única forma de distinguir entre usuarios sería por el interés que genera el uso de la aplicación.

##### Perfiles de usuario:

###### Usuario activo:

Usuario que habitualmente hace uso de la aplicación, compartiendo constantemente ubicaciones, puntuandolas y escribiendo notas sobre estas y otros lugares registrados en la aplicación.

###### Usuario pasivo:

Usuario que hace uso de la aplicación ocasionalmente, visitando lugares compartidos por otros usuarios, aunque no puntúen ni comenten nada sobre ellos. No comparten ninguna o pocas ubicaciones.

### 4.1.3. Contextos de uso

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

Se puede hacer uso de la aplicación desde cualquier ubicación, disponiendo de un smartphone con SO Android. Para poder usar las funcionalidades de la aplicación, el dispositivo debe de tener GPS y conexión a internet activa.

Un requisito que debe cumplir el usuario es tener conocimientos básicos de uso de la tecnología del dispositivo.

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

### 4.1.4. Análisis de tareas

Las principales tareas del sistema son:

- Tarea 1: Login en el sistema o crear usuario
- Tarea 2: Añadir una nueva ubicación (pública o privada)
- Tarea 3: Consultar ubicaciones cercanas
- Tarea 4: Añadir información a una ubicación existente
- Tarea 5: Obtener ruta desde la posición actual del usuario hasta una ubicación existente

Muestro información de las condiciones y pasos a seguir para cada una de las tareas.

<b>Tarea 1: Login en el sistema o crear usuario</b>	
Objetivo	Login en el sistema o crear un usuario
Pre condición	
Post condición	El usuario puede hacer uso normal de la aplicación
Pasos a seguir:	<ol style="list-style-type: none"> <li>1 Abrir la aplicación</li> <li>2 Si el usuario ya esta registrado, introducir usuario y contraseña, pulsar botón login</li> <li>3 Si el usuario no esta registrado, pulsar botón registro.</li> <li>4 Rellenar el formulario, pulsar aceptar y se registra el usuario</li> <li>5 Repetir paso 2 para los nuevos usuarios ya registrados</li> </ol>

**Tarea 2: Añadir una nueva ubicación**

Objetivo	Crear un nuevo lugar o ubicación en el sistema, sea compartido o no
Pre condición	Iniciar la aplicación correctamente (Login de usuario)
Post condición	La ubicación queda añadida al sistema, de la forma que elija el usuario
Pasos a seguir:	<ol style="list-style-type: none"> <li>1 Login en el sistema, la aplicación te ubica automáticamente</li> <li>2 Pulsar el botón “marcar”</li> <li>3 Seleccionar si quieres que sea ubicación privada o publica</li> <li>4 Añadir nombre, dirección, notas y puntuación (parte publica)</li> <li>5 Pulsar el botón “guardar”</li> </ol>

**Tarea 3: Consultar ubicaciones cercanas**

Objetivo	Buscar ubicaciones de distintas categorías, que este ubicadas cerca
Pre condición	Iniciar la aplicación correctamente (Login de usuario)
Post condición	
Pasos a seguir:	<ol style="list-style-type: none"> <li>1 Login en el sistema, la aplicación te ubica automáticamente</li> <li>2 Pulsar el botón “buscar”</li> <li>3 Selecciona la categoría que va a filtrar la búsqueda</li> <li>4 Mostrara un listado de ubicaciones almacenadas en el sistema, según el filtro que haya elegido el usuario</li> <li>5 Pulsa sobre la ubicación que te interese</li> <li>6 Muestra toda la información sobre esta ubicación, nombre, dirección, mapa, puntuación, notas, notas de otros usuarios...</li> </ol>

**Tarea 4: Añadir información a una ubicación existente**

Objetivo	Añadir puntuación y notas a un punto existente en el sistema
Pre condición	<ul style="list-style-type: none"> <li>· Iniciar la aplicación correctamente (Login de usuario)</li> <li>· El punto de ubicación exista en el sistema</li> </ul>
Post condición	La información sobre la ubicación queda modificada
Pasos a seguir:	<ol style="list-style-type: none"> <li>1 Login en el sistema, la aplicación te ubica automáticamente</li> <li>2 Pulsar el botón “buscar”</li> <li>3 Selecciona la categoría que va a filtrar la búsqueda</li> <li>4 Mostrara un listado de ubicaciones almacenadas en el sistema, según el filtro que haya elegido el usuario</li> </ol>

- |   |  |
|---|--|
| 5 | Pulsa sobre la ubicación que te interese                             |
| 6 | Modifica la información sobre esta ubicación puntuación y añade nota |

### Tarea 5: Obtener ruta desde la posición actual del usuario hasta una ubicación existente

Objetivo	Saber como llegar a uno de los puntos almacenados en el sistema
Pre condición	<ul style="list-style-type: none"> <li>· Iniciar la aplicación correctamente (Login de usuario)</li> <li>· El punto de ubicación exista en el sistema</li> </ul>
Post condición	
Pasos a seguir:	<ol style="list-style-type: none"> <li>1 Login en el sistema, la aplicación te ubica automáticamente</li> <li>2 Pulsar el botón “buscar”</li> <li>3 Selecciona la categoría que va a filtrar la búsqueda</li> <li>4 Mostrara un listado de ubicaciones almacenadas en el sistema, según el filtro que haya elegido el usuario</li> <li>5 Pulsa sobre la ubicación que te interese</li> <li>6 Pulsa sobre el mata de esta ubicación, el sistema genera la ruta automáticamente</li> </ol>

## 4.2. Diseño conceptual

### 4.2.1. Escenarios de uso

Lo siguiente que realizo en esta fase es definir los escenarios de uso, que simplemente son hipótesis de la situación en la que los usuarios se pueden encontrar al realizar tareas descritas en el apartado anterior, usando los contextos de uso y los perfiles de usuario ya definidos.

A continuación se detallan los posibles escenarios de uso de la aplicación, utilizando la técnica de personajes, relatando situaciones ficticias con usuarios pertenecientes a perfiles identificados y realizando tareas ya descritas.

#### Escenario 1, para el uso de la gestión general de ubicaciones

Al señor Moisés, estando en su domicilio después de almorzar, decide ver su correo electrónico al cual le ha llegado información sobre “Whereis”. Moisés es una persona bastante curiosa y decide probar la aplicación, crea una cuenta e inspecciona todas las posibilidades de dicha aplicación, añadiendo algún punto como donde tiene aparcado su vehículo y algunas farmacias cercanos a su domicilio.



Perfil de usuario	Usuario pasivo
Contexto	Vida cotidiana, en el hogar
Objetivo	Consultar y añadir ubicaciones cerca del hogar
Tareas	Añadir/Consultar una ubicación existente (Tarea 1, 2 y 3)
Necesidad de información	Localización actual
Funcionalidades necesarias	Localizar la posición del usuario, añadir o mostrar puntos cercanos, y como llegar a ellos
Desarrollo de las tareas	Abrir la aplicación, para añadir ubicación, pulsar sobre “marcar”, seleccionar privado y rellenar campos. Para la consulta pulsar el botón buscar, y seleccionar “farmacia”

## Escenario 2, para el uso de la gestión de tipos de ubicaciones

El señor Moisés, dos días después de haber probado “Whereis”, junto con sus amigos una noche de verano, deciden ir a cenar a un sitio que solo 2 de todos los compañeros conocen. Es un lugar recóndito, de difícil acceso, situado a las faldas del Teide, en La Orotava. Cuando acaba de cenar, decide sacar su móvil, ejecutar la aplicación y almacenar la ubicación dicho lugar, añadiendo una nota comentando “lo bueno que le ha parecido este lugar” y dándole la máxima puntuación.

Perfil de usuario	Usuario activo
Contexto	Vida cotidiana, fuera del hogar
Objetivo	Añadir un nuevo restaurante a la BBDD del sistema
Tareas	Añadir una nueva ubicación (Tarea 1 y 2)
Necesidad de información	Localización actual
Funcionalidades necesarias	Localizar la posición del usuario y añadir ubicación
Desarrollo de las tareas	Abrir la aplicación, para añadir ubicación, pulsar sobre “marcar”, seleccionar publica, y rellenar los campos de nombre, notas y puntuación

### Escenario 3, para el uso de la gestión de ubicaciones ya existentes

Eduardo, amigo del señor Moisés, fue la persona que le envió a este el correo informándole de la aplicación “Whereis”. Un sábado va de ruta con su moto y se para en un pequeño pueblo. En la misma carretera general, deciden quedarse a descansar ya que se les ha echado la noche encima, en un lugar llamado “El refugio”. Le parece un buen sitio, y justamente antes de salir, por la mañana, le comenta a sus amigos moteros que va a añadir este lugar en una aplicación que lleva usando unos meses. Al intentar añadir este lugar la aplicación le indica que ya existe, y decide añadir notas y puntuarla, y mientras, lee notas dejadas por otros usuarios de la aplicación.

Perfil de usuario	Usuario activo
Contexto	Vida cotidiana, fuera del hogar
Objetivo	Añadir notas y puntuación a una ubicación
Tareas	Añadir una nueva ubicación (Tarea 1 y 4)
Necesidad de información	
Funcionalidades necesarias	Mostrar datos de la BBDD sobre “El refugio”, añadir notas y puntuar
Desarrollo de las tareas	Abrir la aplicación, para añadir ubicación, pulsar sobre “marcar”, seleccionar publica, y rellenar los campos de nombre, notas y puntuación. La aplicación le informara que ya existe esta ubicación, y le mostrara la información de la misma. El usuario podrá añadir nota y puntuar.

### Escenario 4, para el uso de la gestión de ubicaciones cercanas

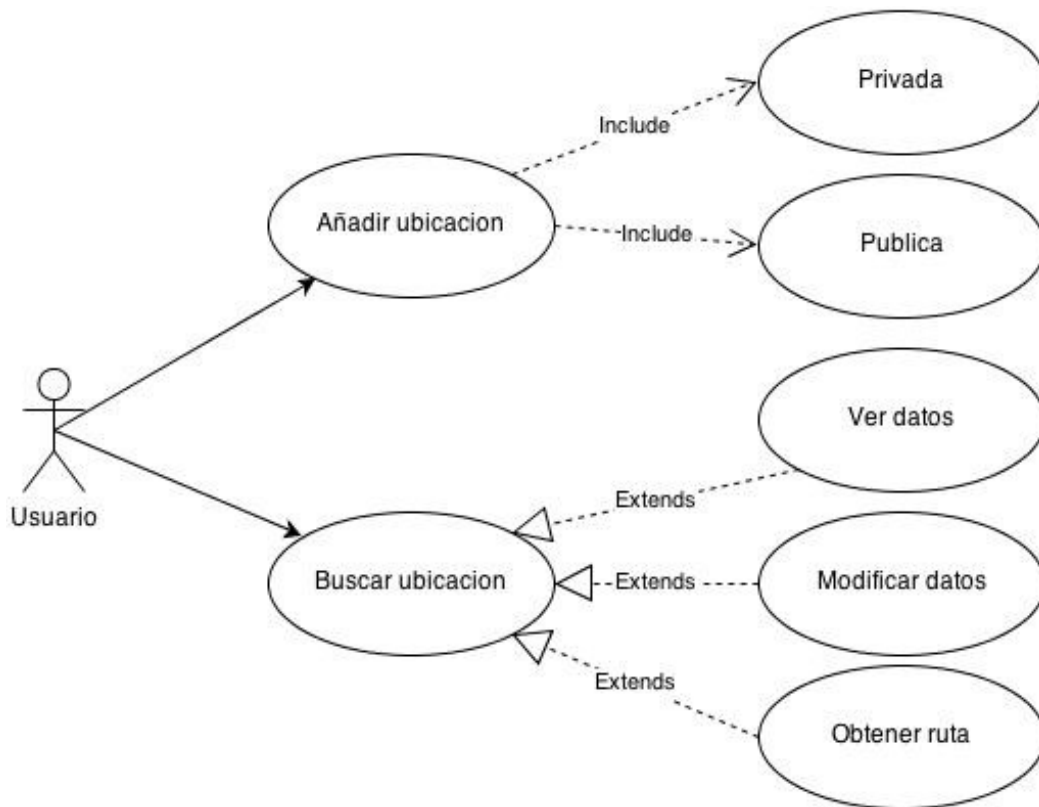
Maru, una usuaria que a diario utiliza la aplicación “Whereis” ya que nunca consigue aparcar el coche, para dirigirse a su trabajo, cerca de este, y lo ubica como privado en la aplicación todas las mañanas y así no tener problema de encontrarlo al salir de su trabajo. Esta habituada a usar la aplicación y un día en El Puerto de la Cruz, después de cenar, decide buscar en la aplicación zonas de ocio nocturno cerca de donde se encuentra, y leer comentarios de otros usuarios y así seleccionar el sitio que mas le apetece para tomarse unas copas y bailar con sus amigas. También le gustaría que le mostrara como llegar al lugar elegido porque no conoce muy bien esta zona.

Perfil de usuario	Usuario pasivo
Contexto	Vida no cotidiana, fuera del hogar
Objetivo	Utilizar la información de la aplicación un día diferente

<b>Tareas</b>	Consultar datos de ubicaciones cercanas (Tarea 1, 3 y 5)
<b>Necesidad de información</b>	Localización actual
<b>Funcionalidades necesarias</b>	Localizar la posición del usuario y mostrar puntos cercanos en el mapa, y como llegar a ellos
<b>Desarrollo de las tareas</b>	Abrir la aplicación, para la consulta ubicaciones cercanas pulsar el botón buscar, y seleccionar “zonas de ocio nocturnas”

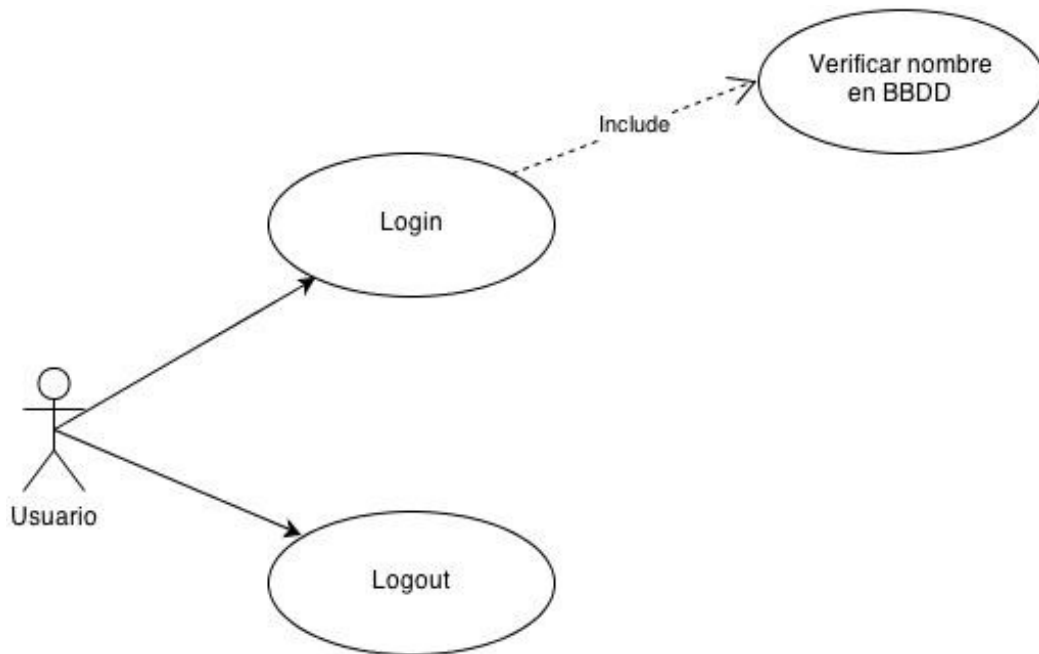
### 4.2.2. Diagramas de casos de uso

El siguiente diagrama muestra los casos de uso de las tareas de la aplicación en general.

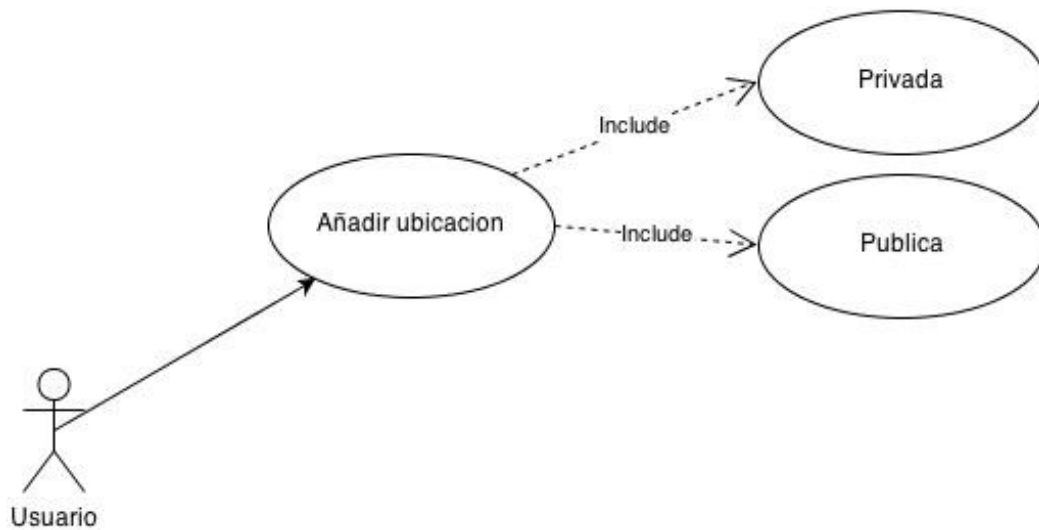


Los siguientes diagramas representan los casos de uso de cada una de las tareas.

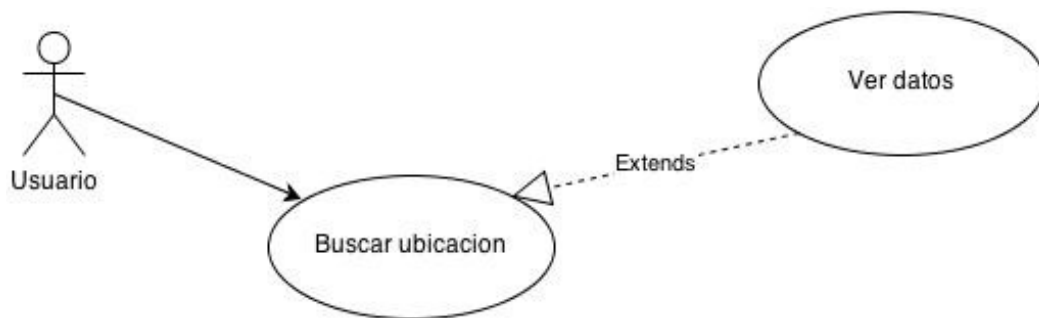
**Tarea 1: Login en el sistema o crear usuario**



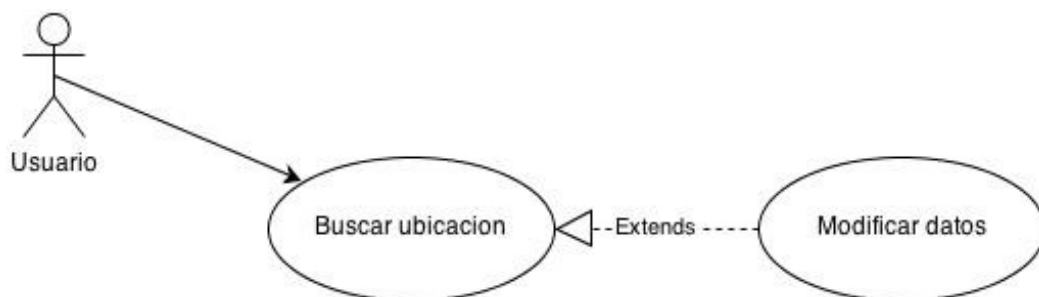
**Tarea 2: Añadir una nueva ubicación (publica o privada)**



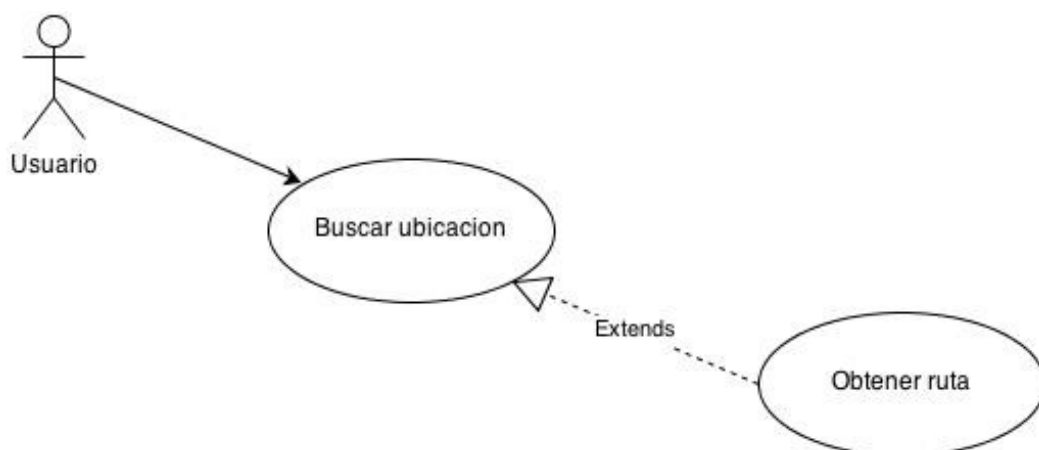
**Tarea 3: Consultar ubicaciones cercanas**



**Tarea 4: Añadir información a una ubicación existente**



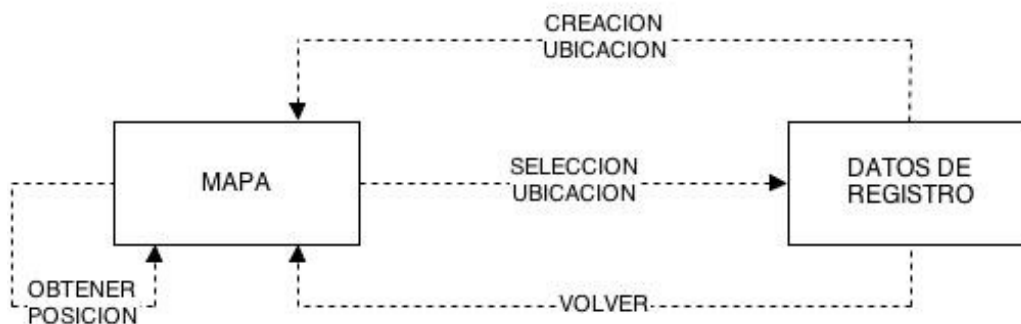
**Tarea 5: Obtener ruta desde la posición actual del usuario hasta una ubicación existente**



### 4.2.3. Flujos de interacción

A continuación muestro los flujos de interacción de las principales tareas (la Tarea 1, ya que es el login del usuario, no creo el diagrama de flujo).

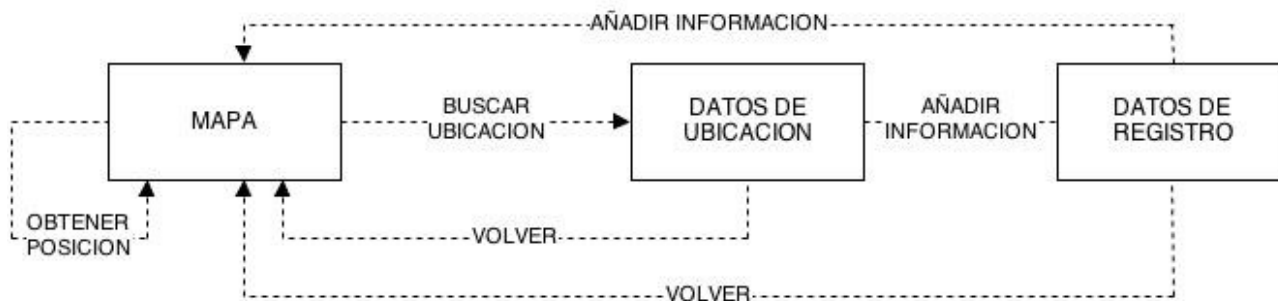
#### Tarea 2: Añadir una nueva ubicación (pública o privada)



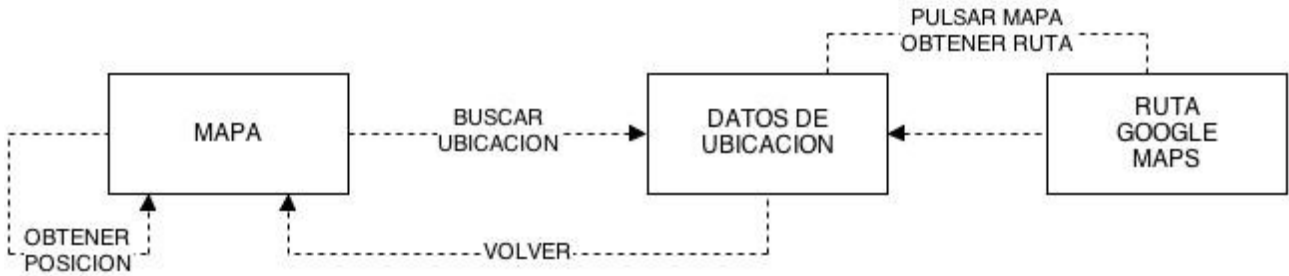
#### Tarea 3: Consultar ubicaciones cercanas



#### Tarea 4: Añadir información a una ubicación existente



**Tarea 5: Obtener ruta desde la posición actual del usuario hasta una ubicación existente**



## 4.3. Prototipado

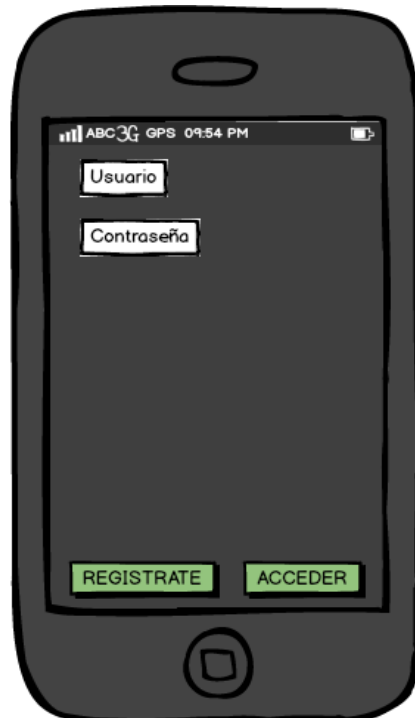
### 4.3.1. Sketches

No realizo sketches, ni dibujos simulando lo que se pretende desarrollar. Directamente genero los prototipos.

### 4.3.2. Prototipo

El prototipo esta vez lo genero con la aplicación Balsamiq Mockups, distinta a la utilizada anteriormente, y aunque esta aplicación esta mas orientada a IOS, no he podido utilizar otra aplicación similar ya que estoy teniendo problemas con los sistemas operativos con los que trabajo en mi casa, Windows XP y Mac OS X 10.5.8, ya que no localizo una aplicación gratuita que sea soportada por dichos sistemas. Este es el prototipo final:

Interfaz de login:





Interfaz principal:



Interfaz de almacenamiento de ubicación:



Interfaz que muestra una búsqueda:



Interfaz que muestra la información de la ubicación seleccionada:



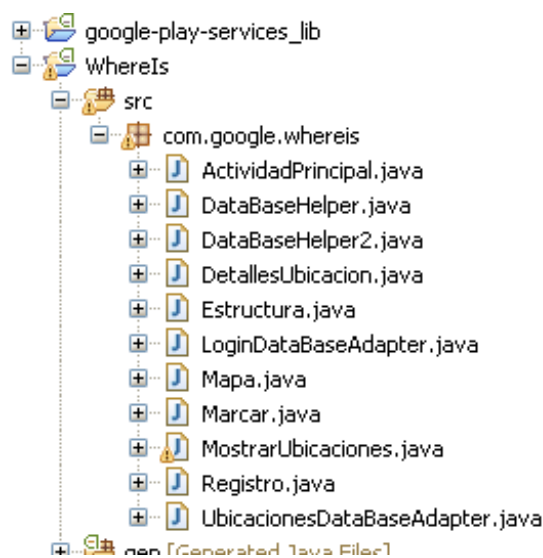
## 5. Implementación

A continuación paso a comentar algunas características de las clases desarrolladas y decisiones tomadas durante la fase de implementación. Durante el desarrollo intento respetar algunos aspectos para el buen funcionamiento del sistema, intentando que sea lo mas ágil posible, utilizando de forma responsable los recursos del móvil.

Para ello intento utilizar lo menos posible el teclado, ya que no es del todo cómodo para el usuario, la utilización del GPS no abusiva para el consumo de energía de los móviles y el acceso a internet, que únicamente se utiliza para la descarga de mapas.

### 5.1 Organización de las clases implementadas

Las clases implementadas están contenidas todas en el paquete ***com.google.whereis***.

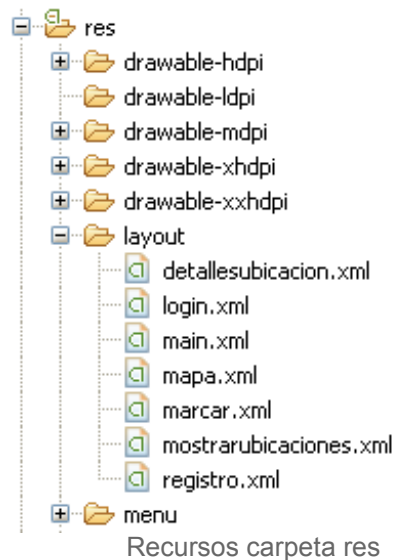


Todas las clases desarrolladas en el paquete ***com.google.whereis***

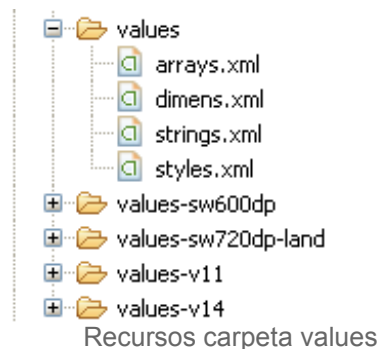
Además de las clases, se han tenido que añadir una serie de dependencias como ***google-play-services\_lib.jar*** y ***android-support-v4.jar*** que sirve para poder añadir funciones y métodos para implementar mapas de google, como fragments. En relación las APIs utilizadas, se añaden la API de Google, que contiene las librerías de clases ***android.jar***, ***effects.jar***, ***usb.jar*** y ***maps.jar***, que son utilizadas también para la implementación de alguna de las clases desarrolladas en la aplicación.



La carpeta **res** contiene los recursos que la aplicación utilizará, como los recursos gráficos en las carpetas drawable, los diferentes diseños de todas las ventanas en la carpeta layout, relacionadas con clases que controla cada uno de estos elementos.



La carpeta values que contiene estilos y constantes usadas en las clases desarrolladas. Uno de esos estilos es MiHolo que simplemente hereda del tema Holo. Respecto a las constantes, se han definido serie de strings y un array para poder mostrar datos de forma cómoda asociados a elementos de las ventanas como TextView o Spinners.



En el archivo AndroidManifest.xml se definen versiones de uso, permisos que debe conocer el sistema operativo y los usuarios que se descarguen la aplicación del google play, todas las actividades incluidas en la aplicación y el valor de la key para el uso de los mapas de google maps.

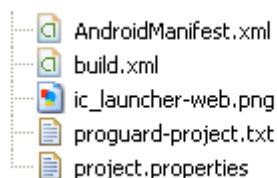
En **ic\_launcher-web.png** se ha diseñado el icono de la aplicación.

En relación a los permisos, han sido necesarios para la activación del GPS, almacenamiento en la memoria externa, acceso a internet y la recepción de mapas.

```
<uses-permission android:name="com.google.whereis.permission.MAPS_RECEIVE" />
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.RECEIVED_BOOT_COMPLETED" />
```



Icono de la aplicación



Archivo AndroidManifest.xml

En relación con las clases implementadas, voy a mencionar características principales y un resumen de la funcionalidades de cada una de ellas.

### Clase ActividadPrincipal

Como su propio nombre indica, esta clase es la principal de la aplicación. Esta clase corresponde con la una actividad, la que se ejecuta inicialmente. Desde ella se puede acceder al resto de actividades. El usuario podrá realizar el acceso o registro del user de la aplicación. Se activara un Dialog para el login de la aplicación, o accederá a la actividad registro para realizar el registro del user en la misma.

### Clase Registro

Esta clase también corresponde con otra actividad de la aplicación. Desde ella se puede registrar un user, indicando su nombre, contraseña y debiendo repetir esta ultima

para asegurarnos que es la que el usuario quiere introducir. Esta clase hace uso de la clase LoginDataBaseAdapter, para poder realizar un registro en la base de datos de usuarios.

### Clase DataBaseHelper

La clase DataBaseHelper extiende de la clase SQLiteOpenHelper que crea, conecta y actualiza con una base de datos SQLite. Esta clase tiene un constructor y dos métodos abstractos onCreate() y onUpdate() que sirven para lo anteriormente indicado con la base de datos de usuarios.

### Clase LoginDataBaseAdapter

Esta clase se encarga de interactuar con la base de datos de usuarios. Aquí se implementan las funciones que permiten a otras clases insertar, eliminar, actualizar y consultar registros de la base de datos de usuarios. Falta afinar la implementación, como por ejemplo eliminar usuarios ya creados, pero básicamente esta desarrollada.

### Clase Mapa

La clase mapa corresponde con otra actividad que muestra un mapa mundial donde será ubicado el usuario, utilizando el GPS del dispositivo. Extiende de un FragmentActivity y utiliza los mapas de google maps para realizar dichas funcionalidades. Desde ella se puede acceder al almacenamiento de ubicaciones mediante el botón “Marcar” (implementado en la función addMarker (View) ), a la búsqueda de ubicaciones cercanas a la ubicación actual del usuario por medio del botón “Buscar” (implementado en la función uSearch (View) ) y a la realización de un zoom para poder ver claramente las calles cercanas donde se encuentra el usuario, a través del botón “Mi ubicación” (implementado en la función animatedCamera (View) ). También esta desarrollada una función que, cuando el gps no ubica correctamente al usuario, este puede pulsar sobre el mapa y esa ubicación será la que se almacene en la base de datos.

### Clase Marcar

Esta clase, actividad, es la que se encarga del almacenamiento de las ubicaciones. Dependiendo de la elección del usuario en un RadioGroup, pudiendo ser pública o privada, la clase almacena la información en una Base de datos o en la memoria externa del móvil. Por medio de un Spinner, muestra las posibles categorías de almacenamiento (Lugares de interés, comida, café, alojamiento, arte, compras, zonas deportivas y vida nocturna). Además el usuario debe introducir el nombre de la ubicación, dirección, notas y por medio de un RatingBar puntuar de 0 a 5 dicho lugar. Hay un botón “guardar” que desde su evento setOnClickListener() se ejecutará la mayor parte del código de esta clase.

### Clase DataBaseHelper2

La clase DataBaseHelper2 extiende de la clase SQLiteOpenHelper es similar a DataBaseHelper, pero sirve para gestionar la creación, conexión y actualización de la base de datos de ubicaciones.

### Clase UbicacionesDataBaseAdapter

Esta clase se encarga de interactuar con la base de datos de ubicaciones. Similar a la clase LoginDataBaseAdapter. Falta afinar la implementación pero básicamente esta desarrollada.

### Clase Estructura

Esta clase gestiona los datos a recuperar de las bases de datos de ubicaciones. Contiene variables que corresponden a las columnas de la base datos (id, usuario, categoría, nombre, dirección, nota, puntuación, latitud y longitud) y una serie de funciones y métodos como getX() y setX() (la X se sustituye por cada uno de los nombres de las columnas, como por ejemplo getCategory() o setCategoria() ) que devuelve o asignan datos a dichas variables.

### Clase MostrarUbicaciones

Esta clase, también actividad, muestra por medio de un ListView los datos de ubicaciones cercanas a la geolocalización actual del usuario. Seleccionando por medio de un spinner de categorías que filtra las ubicaciones almacenadas en la base de datos, si se trata de ubicaciones publicas o almacenadas en la memoria externa, si se trata de ubicaciones privadas, muestra estas por proximidad y mejor puntuación. A parte de los métodos de las clases ya implementadas en android como onItemSelected() de la clase spinner, desarrollo una función cargaLista() que simplemente carga el ListView con un adaptador de Strings.

### Clase DetallesUbicacion

Esta clase incluida en las actividades de la aplicación muestra los datos, en detalle de cada ubicación, obtenidos de la base de datos de ubicaciones. Los datos que muestra son el nombre, dirección, geolocalización y notas de otros usuarios. Se puede añadir una nueva nota y puntuar por parte del usuario que no ha creado dicha ubicación. Finalmente no se ha podido ni usar ni desarrollar por falta de tiempo.

\* Para realizar las pruebas con las bases de datos generadas (DBLogin.db y DBUbicaciones.db) he utilizado la utilidad que ofrece el SDK llamada **ADB (Android Debug Bridge)**.y una aplicación llamada **SQLite DataBase Browser**.

Primero he utilizado la aplicación WhereIs para almacenar ubicaciones de forma aleatoria, he utilizado adb de la siguiente forma:

```
adb pull /data/data/com.google.whereis/databases/DBLogin.db c:/BBDD
adb pull /data/data/com.google.whereis/databases/DBUbicaciones.db c:/BBDD
```

y las he almacenado en la carpeta BBDD de la raíz del disco C: de mi equipo. Luego, con la ayuda de SQLite DataBase Browser he consultado el estado de la bbdd:

SQLite Database Browser - C:/DBUbicaciones.db

File Edit View Help

Database Structure Browse Data Execute SQL

Table: Ubicaciones

New Record Delete Record

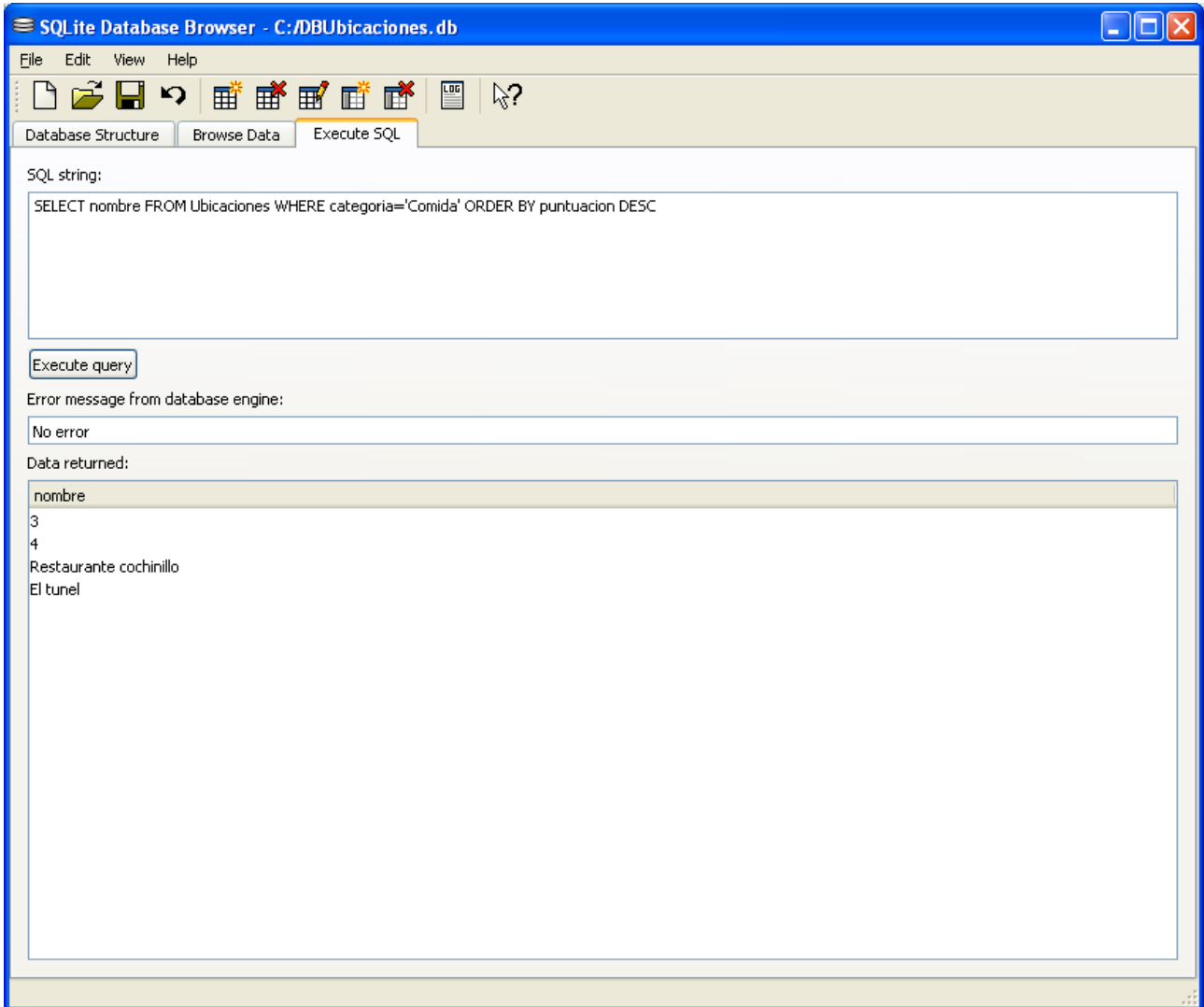
id	usuario	categoria	nombre	direccion	nota	puntuacion	latitud
1	1	Comida	Restaurante cochini	Calle la cruz, 4			28.5049616
2	2	Cafe	Bar Unamuno	Calle Villalba Herbas			28.4679033
3	3	Alojamiento	Hotel Barlovento	Calle la palma, 1			28.4463769
4	4	Arte	Grupo de pintores	Av. Mendez Nuñez,			28.4750054
5	5	Compras	Centro comercial Ca	Av. Santa Cruz, sn			28.0821192
6	6	Zonas deportivas	El muro	La cuesta, sn			28.4750784
7	7	Vida nocturna	Dreams	Av. maritima, 1			28.4551018
8	8	Comida	El tunel	Av. San Andres			28.5044535
9	9	Cafe	Genesis	Calle Alcalde mandill			28.4615398
10	10	Alojamiento	Hotel Botanico	Calle puerto, 35			28.4116482
11	11	Arte	Escultores y artistas	Calle de la x, 2			28.4670157
12	12	Compras	Maya	Plaza de España, sn			28.4664729
13	13	Zonas deportivas	Club deportivo Saluc	Barrio de la salud			28.4717056
14	14	Vida nocturna	Papagayo	Av. maritima, 2			28.0664651
15	15 a0	Lugares de interes c	lugar0	Calle 0	cuesta 0€	2.5	28.4
16	16 a1	Lugares de interes c	lugar1	Calle 1	cuesta 1€	3.5	29.4
17	17 a2	Lugares de interes c	lugar2	Calle 2	cuesta 2€	4.5	30.4
18	18 a3	Lugares de interes c	lugar3	Calle 3	cuesta 3€	5.5	31.4
19	19 a4	Lugares de interes c	lugar4	Calle 4	cuesta 4€	6.5	32.4
20	20 a	Lugares de interes c		1	1	1	1.0
21	21 a	Lugares de interes c		2	2	2	2.0
22	22 a	Comida		3	3	3	3.0
23	23 a	Comida		4	4	4	2.5

1 - 34 of 34

Go to: 0



También he simulado una consulta como la que se realiza en la clase UbicacionesDataBaseAdapter.class:



## 5.2 Gestión del posicionamiento

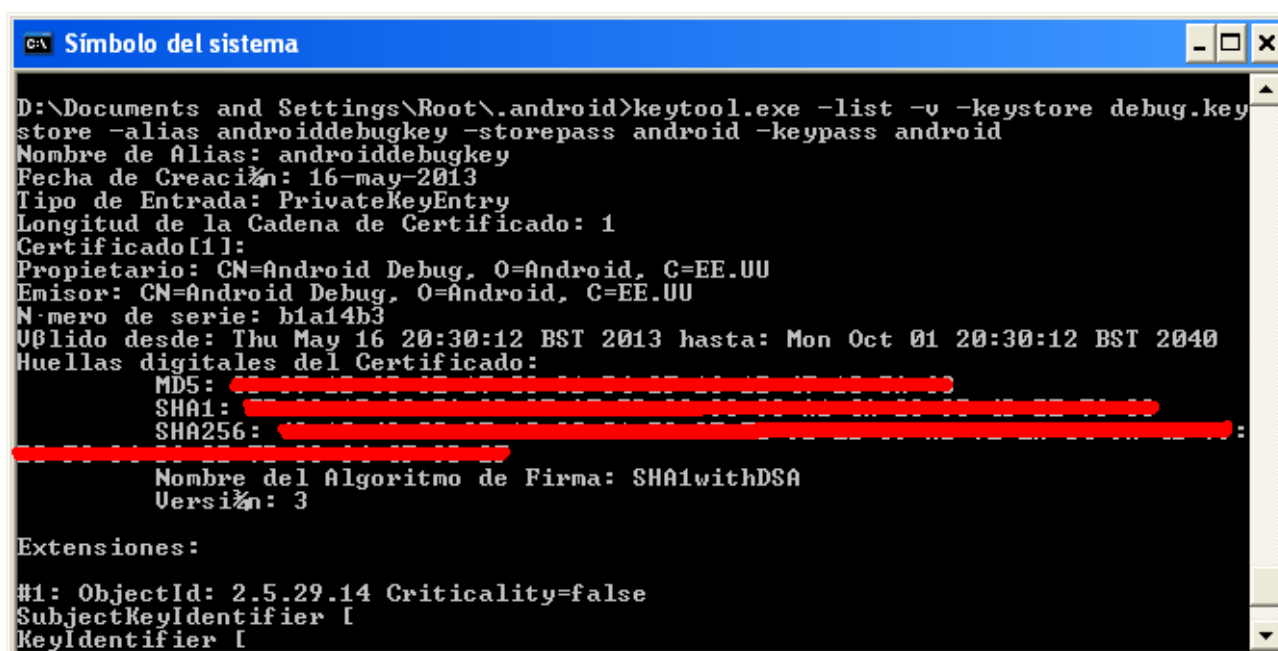
Para que la aplicación pueda usar los servicios de posicionamiento, he tenido que realizar unas tareas que describo a continuación:

- Clave (key) de acceso a la geolocalización: Desde google indican que para tener acceso a su API se requiere configurar la aplicación con una clave (<https://developers.google.com/console/help/#UsingKeys>) y un identificador de proyecto.

En la parte de **Android client IDs** Indican que se necesita realizar una serie de pasos como añadir el servicio “Google Maps Android API V2” y generar una “Key for Android apps (with certificates)” en la consola de google.

Para realizar este tarea, se ejecuta de en la línea de comando de Windows Xp:

```
keytool.exe -list -v -keystore debug.keystore -alias androiddebugkey -storepass android -keypass android
```



```

ca Símbolo del sistema
D:\Documents and Settings\Root\.android>keytool.exe -list -v -keystore debug.keystore -alias androiddebugkey -storepass android -keypass android
Nombre de Alias: androiddebugkey
Fecha de Creación: 16-may-2013
Tipo de Entrada: PrivateKeyEntry
Longitud de la Cadena de Certificado: 1
Certificado[1]:
Propietario: CN=Android Debug, O=Android, C=EE.UU
Emisor: CN=Android Debug, O=Android, C=EE.UU
Número de serie: b1a14b3
Válido desde: Thu May 16 20:30:12 BST 2013 hasta: Mon Oct 01 20:30:12 BST 2040
Huellas digitales del Certificado:
MD5: [REDACTED]
SHA1: [REDACTED]
SHA256: [REDACTED]
Nombre del Algoritmo de Firma: SHA1withDSA
Versión: 3

Extensiones:
#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [

```

Una vez obtenida la clave SHA1 de mi almacén de claves, me dirijo a la consola de google (con una cuenta de google creada para la ocasión developer.ramon.manzanares) y añado un acceso simple a la API de google en “Create new Android key...”, donde me solicitan tanto la clave SHA1 de mi archivo keystore.debug (en este caso uso el almacén predeterminado debug) como el nombre del paquete de la aplicación com.google.whereis. Quedaría de la siguiente forma:

Búsqueda Imágenes Maps Play YouTube Noticias Gmail Drive Más ▼ developer.ramon.manzanares@gmail.com ▼ | Configuración ▼ | Ayuda | Cerrar sesión

Google apis

API\_Key\_Google\_Maps ▼

- Overview
- Services
- Team
- API Access
- Reports
- Quotas

### API Access

To prevent abuse, Google places limits on API requests. Using a valid OAuth token or API key allows you to exceed anonymous limits by connecting requests back to your project.

#### Authorized API Access

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private. A single project may contain up to 20 client IDs. [Learn more](#)

**Create an OAuth 2.0 client ID...**

#### Simple API Access

Use API keys to identify your project when you do not need to access user data. [Learn more](#)

Key for Android apps (with certificates)	Actions
API key: <del>AKIA...</del> Android apps: <del>com.google.android.maps.v2</del> ; com.google.whereis Activated on: Jun 6, 2013 4:00 AM Activated by: developer.ramon.manzanares@gmail.com – you Obsolete key: <del>AKIA...</del> Status: Inactive	<a href="#">Generate new key...</a> <a href="#">Revert to obsolete key...</a> <a href="#">Edit allowed Android apps...</a> <a href="#">Delete key...</a>
Key for browser apps (with referers)	Actions
API key: <del>AKIA...</del> Referers: Any referer allowed Activated on: May 16, 2013 2:52 PM Activated by: developer.ramon.manzanares@gmail.com – you	<a href="#">Generate new key...</a> <a href="#">Edit allowed referers...</a> <a href="#">Delete key...</a>

[Create new Server key...](#)
[Create new Browser key...](#)
[Create new Android key...](#)
[Create new iOS key...](#)

#### Notification Endpoints

Use notification endpoints to identify domains that may receive webhook notifications from your API. [Learn more](#)

Allowed Domains:	No domains allowed	<a href="#">Edit...</a>
------------------	--------------------	-------------------------

Ahora hay que añadir permisos en el archivo AndroidManifest.xml de mi proyecto para poder acceder a los mapas, la librería y la key que me da la consola de google:

```
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

```
<uses-library android:name="com.google.android.maps" />

<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="AKIA..." />
</application>
```

AndroidManifest.xml librería y key de google maps

· Después de realizar lo anteriormente indicado, se puede hacer uso en las actividades de los mapas de google. En esta aplicación solo hará falta en Mapa.class usar esta librería.

Estos son los casos en los que utilizo dichas librerías:

```
public class Mapa extends FragmentActivity /* implements OnMapClickListener */{

    private GoogleMap mapa;
    private double latitud;
    private double longitud;
```

Declaración del mapa.

```
// Gestion del Mapa
mapa = ((SupportMapFragment) getSupportFragmentManager()
        .findFragmentById(R.id.map)).getMap();
mapa.setIndoorEnabled(true);
mapa.setMapType(GoogleMap.MAP_TYPE_HYBRID);
mapa.setMyLocationEnabled(true);
mapa.getUiSettings().setZoomControlsEnabled(false);
mapa.getUiSettings().setCompassEnabled(true);
mapa.getUiSettings().setMyLocationButtonEnabled(false);
// mapa.setOnMapClickListener(this);
```

Configuración del mapa para la visualización en la actividad

```
// Manejador del boton "Mi ubicacion"
public void animateCamera(View view) {
    if (mapa.getMyLocation() != null)
        mapa.animateCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(
            mapa.getMyLocation().getLatitude(), mapa.getMyLocation()
                .getLongitude()), 18));
```

Realización del zoom a la geolocalización del usuario en el momento actual

```
// Manejador del boton "Marcar"
public void addMarker(View view) {

    latitud = mapa.getMyLocation().getLatitude();
    longitud = mapa.getMyLocation().getLongitude();
```

Actualizar latitud y longitud con los datos actuales

## 5.3 Gestión del almacenamiento

Para que la aplicación pueda usar los servicios de almacenamiento se ha utilizado dos vertientes, las bases de datos en SQLite y el acceso a ficheros en la memoria externa. En la clase Marcar.class es donde se decide que tipo de almacenamiento.

· El uso de las bases de datos SQLite lo gestiono con 4 clases, DataBaseHelper, DataBaseHelper2, LoginDabaBaseAdapter y UbicacionesDabaBaseAdapter, todas ya explicadas anteriormente.

Muestro algunas implementaciones de dichas tareas:

```
public class DataBaseHelper extends SQLiteOpenHelper {
    public DataBaseHelper(Context context, String name, CursorFactory factory,
        int version) {
        super(context, name, factory, version);
    }

    // Se llama cuando no existe ninguna base de datos en disco y necesita la
    // clase de ayuda para crear una nueva
    @Override
    public void onCreate(SQLiteDatabase _db) {
        _db.execSQL(LoginDataBaseAdapter.DATABASE_CREATE);
    }

    // Se llama cuando hay una versión de la base de datos no coincide.
    // Significa que la versión
    // de la base de datos en el disco necesita ser actualizado a la versión
    // actual
    @Override
    public void onUpgrade(SQLiteDatabase _db, int _oldVersion, int _newVersion) {
        // Log de la version actualizada
        Log.w("TaskDBAdapter", "Upgrading from version " + _oldVersion + " to "
            + _newVersion + ", which will destroy all old data");

        // Actualizar la base de datos existente para adaptarse a la nueva
        // versión. Múltiple
        // versiones anteriores se pueden manejar mediante la comparación de los valores _oldVersion y _newVersion
        // El caso más sencillo es eliminar la tabla antigua y crear una nueva.
        _db.execSQL("DROP TABLE IF EXISTS " + "TEMPLATE");
        //Crear una nueva
        onCreate(_db);
    }
}
```

Constructor y métodos crear/actualizar de la clase DataBaseHelper

```

import android.content.ContentValues;

public class LoginDataBaseAdapter
{
    static final String DATABASE_NAME = "DBLogin.db";
    static final int DATABASE_VERSION = 1;
    public static final int NAME_COLUMN = 1;
    //Crear campo público para cada columna de la tabla.
    //Declaración de SQL para crear una nueva base de datos
    static final String DATABASE_CREATE = "create table " + "LOGIN" + "( "
        + "ID" + " integer primary key autoincrement,"
        + "USERNAME text,PASSWORD text); ";
    //Variable que contiene la instancia de base de datos
    public SQLiteDatabase db;
    //Contexto de la aplicación utilizando la base de datos
    private final Context context;
    //Abrir o actualizar DataBaseHelper
    private DataBaseHelper dbHelper;
    public LoginDataBaseAdapter(Context _context)
    {
        context = _context;
        dbHelper = new DataBaseHelper(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
}

```

Constructor y constantes de la clase LoginDataBaseAdapter

```

//Eliminar entrada de la base de datos
public int deleteEntry(String UserName)
{
    //Cadena id=String.valueOf(ID);
    int numberOfEntriesDeleted = db.delete("LOGIN", "USERNAME=?", new String[]{UserName});
    return numberOfEntriesDeleted;
}
//Obtener entrada de la base de datos
public String getSingleEntry(String userName)
{
    Cursor cursor=db.query("LOGIN", null, " USERNAME=?", new String[]{userName}, null, null, null);
    if(cursor.getCount()<1) //UserName no existe
    {
        cursor.close();
        return "NO EXISTE";
    }
    cursor.moveToFirst();
    String password= cursor.getString(cursor.getColumnIndex("PASSWORD"));
    cursor.close();
    return password;
}
//Actualizar entrada en la base de datos
public void updateEntry(String userName,String password)
{
    //Define la actualizacion de la fila contenida
    ContentValues updatedValues = new ContentValues();
    //Asigna valores para cada fila
    updatedValues.put("USERNAME", userName);
    updatedValues.put("PASSWORD",password);

    String where="USERNAME = ?";
    db.update("LOGIN",updatedValues, where, new String[]{userName});
}

```

Métodos insertar, eliminar y actualizar de la clase LoginDataBaseAdaptor

· El uso de la memoria externa del móvil se gestiona en la clase Marcar.class. Depende de la selección del usuario en el RadioGroup que se muestra en esta actividad. Si se selecciona la ubicación como privada, el código de la clase derivara en la verificación del estado de la tarjeta SD (desmontada o montada, y si es de solo lectura) y genera un fichero ubicaciones\_sd.txt que es donde se irán guardando los datos privados.

```
// Variable para el estado de la memoria externa
String estado = Environment.getExternalStorageState();

if (estado.equals(Environment.MEDIA_MOUNTED)) {
    try {
        File ruta_sd = Environment.getExternalStorageDirectory();
        File fichero = new File(ruta_sd.getAbsolutePath(), "ubicaciones_sd.txt");
        OutputStreamWriter fichero_salida = new OutputStreamWriter(
            new FileOutputStream(fichero));
        fichero_salida.write(user + "," + categoria + ","
            + nombreUbicacion + ","
            + direccionUbicacion + "," + notas + ","
            + puntuacion + "," + latitud + ","
            + longitud);
        fichero_salida.close();
    } catch (Exception ex) {
        Log.e("Ficheros", "Error al escribir fichero a tarjeta SD");
    }

    Toast.makeText(getApplicationContext(),
        "Datos almacenados en memoria interna",
        Toast.LENGTH_LONG).show();
} else if (estado.equals(Environment.MEDIA_MOUNTED_READ_ONLY)) {
    Toast.makeText(getApplicationContext(), "Tarjeta SD Montada. Solo lectura", Toast.LENGTH_LONG).show();
    return;
} else {
    Toast.makeText(getApplicationContext(), "Tarjeta SD DESMONTADA!!!", Toast.LENGTH_LONG)
        .show();
    return;
}
```

Forma de almacenar datos en la memoria externa (tarjeta SD)

Se añade permisos en el AndroidManifest.xml para poder acceder a la memoria externa del terminal:

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

## 5.4 Gestión de datos entre actividades

Para la comunicación entre actividades, se utiliza variables de tipo Bundle. En la aplicación hay dos ocasiones donde tengo que realizar el paso de parámetros antes de la llamada de la siguiente actividad, y son en la clase ActividadPrincipal.class y en Mapa.class.

Para la clase ActividadPrincipal.class lo que implemento es el paso de la variable userName que va a ser almacenada por la clase Marcar.class, pasando por medio de la clase Mapa.class:

```
Toast.makeText(ActividadPrincipal.this, "Genial, acceso permitido", Toast.LENGTH_LONG).show();
//Crear Intent para el acceso y lanzar la actividad Mapa
Intent intent2 = new Intent (getApplicationContext(), Mapa.class);
intent2.putExtra("usuario", userName);
startActivity(intent2);
```

La variable userName que contiene el nombre de usuario es enviada a Mapa.class

También, por el mismo método, desde Mapa.class paso los parámetros usuario (que viene desde ActividadPrincipal.class), latitud y longitud a Marcar.class:

```
private Bundle extras;
private String user;

...

Intent intent1 = new Intent(getApplicationContext(), Marcar.class);
extras = getIntent().getExtras();
user = extras.getString("usuario");
intent1.putExtra("usuario", user);
intent1.putExtra("lat", latitud);
intent1.putExtra("long", longitud);
startActivity(intent1);
```

Paso de parámetros user, latitud y longitud de la clase Mapa.class a Marcar.class

```
private Bundle extras;
private String user;

...

extras = getIntent().getExtras();
user = extras.getString("usuario");
latitud = extras.getDouble("lat");
longitud = extras.getDouble("long");
```

Recepción de parámetros desde la clase Marcar.class

## 6. Evaluación

En el proceso de Diseño Centrado en el Usuario (DCU) es importante realizar una evaluación de la aplicación, de forma que podamos corregir los errores detectados. Podríamos usar la técnica del test de usuario en la que pediremos a un número determinado de usuarios finales que lleven a cabo una serie de tareas en la aplicación. Luego les pediremos que respondan a una serie de preguntas con el fin de poder evaluar su experiencia de uso. De esta forma podríamos detectar posibles errores en la aplicación y, si fuera necesario, corregirlos.



## 6.1. Usuarios que realizarán el test

El perfil de usuario que realizarían el test intentaría que fueran profesionales o gente con bastante nivel de uso de tecnología. La edad de estos usuarios se encontraría entre los 20 hasta los 45 años. En cuanto al idioma, ahora solo estaría desarrollado en castellano, aunque posiblemente se expanda a idiomas como el catalán, inglés, francés y alemán, con lo que actualmente necesitaríamos usuarios de habla hispana, por el idioma de desarrollo de la aplicación.

## 6.2. Preguntas sobre los usuarios

- ¿Que edad tienes?
- ¿Hombre o mujer?
- ¿Trabajas o estudias? ¿Nivel de estudios?
- ¿Que nivel de conocimientos tecnológicos tienes?
- ¿Dispones de dispositivo táctil? Si es afirmativa su respuesta, ¿Cual es su uso principal? (trabajo, ocio, multimedia, uso cotidiano...)
- ¿Tiene algún otro dispositivo de características similares a un smartphone? (gps de vehículo, cámara de fotos, videojuegos de ultima generación como psp, wii, etc...). Si es afirmativa su respuesta, indíquenos la experiencia de uso.

## 6.3. Tareas que realizar en el test

A continuación muestro las tareas elegidas para que los usuarios testen la aplicación:

- Registrarse en la aplicación
- Mostrar ubicación actual
- Añadir aparcamiento de tu vehículo
- Añadir un lugar publico (compartido)
- Buscar ese mismo lugar publico añadido con anterioridad y ver la información
- Buscar las zonas de compras cerca de la ubicación actual
- Ver/escribir notas en la información de la zona deportiva mas cercana.

## 6.4. Preguntas sobre las tareas

A continuación muestro las preguntas elegidas para que los usuarios respondan, junto con las tareas anteriores:

- ¿Que Finalidad buscaba la tarea realizada?
- ¿Ha completado las tareas propuesta con éxito?
- ¿Qué dificultades ha encontrado para completarlas?

- ¿Que Tiempo ha empleado para realizar la tarea?
- ¿De qué elementos carece la aplicación?
- ¿Qué defectos ha detectado en el uso de la aplicación?
- Indicando entre 0 (muy mala) y 10 (muy buena) una puntuación sobre su experiencia como usuario realizando las tareas indicadas anteriormente.
- ¿Cómo se podría mejorar la herramienta?
- Indique cualquier otro comentario sobre las tareas realizadas.

## 7. Variación de la idea inicial

Se han cumplido la mayoría de requisitos que desde un principio se habían elegido para la finalidad de la aplicación.

La variación mas importante de la idea principal es el uso de las bases de datos de forma remota. La visualización de cada una de las ubicaciones en detalle, por problemas de diseño y tiempo, no se ha podido desarrollar esta parte del planteamiento inicial, importante ya que la ampliación de datos, de una forma publica, no se esta llevando a cabo.

Por los mismos motivos, tampoco se ha podido desarrollar una mayor accesibilidad de la aplicación en cuestión de idiomas. Se pretendía realizar la aplicación en al menos 4 idiomas, y finalmente se ha quedado solo en el idioma del desarrollador.

Por otro lado, se ha conseguido desarrollar una funcionalidad que no estaba contemplada desde un principio. Se trata del zoom en el Mapa, cuando un usuario esta geolocalizandose. Esta función es cómoda para el usuario, ampliando la situación del usuario hasta un tamaño en que puede diferenciar en el mapa el lugar donde se encuentra.

El aspecto gráfico de la aplicación no es del todo el idóneo, no se aprecia muy profesional, pero casi esta mejor que lo que se había pensado en un principio.

Finalmente, a nivel del aprendizaje, aunque la evolución ha sido bastante mas lenta de lo que se pensaba en un principio, y sabiendo que nunca había utilizado ninguna de las tecnologías que se ha usado en este proyecto, sabiendo que tenia un tiempo bastante limitado para desarrollarlo, he aprendido muchísimo con esta experiencia. Se que aun queda un largo camino por recorrer, pero ha sido tan gratificar

## 7.1 Futuras mejoras

Aunque el tiempo que hemos tenido para desarrollar el proyecto me ha parecido insuficiente, voy a seguir implementando la aplicación con algunas mejoras.

- Desarrollar la parte del acceso a bases de datos externas, en un servidor, para así poder estar disponible para todos los usuarios el mayor número posible de ubicaciones accesibles desde la aplicación. Se debería investigar sobre la compresión y encriptación de los datos, para el envío y recepción desde los servidores.

- Incluir módulos para que la aplicación pueda acceder a las distintas redes sociales, y poder publicar, por ejemplo, en facebook y twitter.

- Mejora visual del interface de usuario, hasta que parezca un producto profesional.

- Incluir fotografías de las ubicaciones y almacenarlas en las bases de datos de la aplicación.

- Añadir Menús, por ejemplo en la parte de muestra de mapa, para que sea el usuario quien decida de que forma visualizarlo.

- Mejorar la accesibilidad de la aplicación añadiendo idiomas, como se pensaba desde un principio.

- Control de versiones de la aplicación, para controlar la evolución del desarrollo de la aplicación, y así poder estudiar y observar como sería otro caso, no el de este proyecto, en el que habrían al menos dos desarrolladores trabajando en distintos módulos del proyecto, y poder gestionar la unión del código fuente de forma organizada, con repositorios y las versiones del mismo, usando una arquitectura cliente-servidor.

- Crear una versión para IOS, ya que actualmente, prácticamente el 70% de las aplicaciones que se desarrollan para android, también se desarrollan en la otra plataforma.

## 8. Conclusiones

Probablemente, la parte más positiva del desarrollo e implementación de este proyecto sea el aprendizaje que he logrado, gracias a toda la información que existe en Internet sobre el desarrollo en Android, en conocimientos de programación, de la plataforma de desarrollo y de la gestión y organización de proyectos de software. Aunque la línea de aprendizaje no ha sido lo rápida que habría esperado, ya que el desarrollo de software no es mi única ocupación, he utilizado conocimiento adquirido durante los estudios de ingeniería en la UOC y en otras universidades, lograr adquirir todos los conocimientos que en un principio no esperaba llegar a alcanzar, ha sido de un grado satisfactorio muy elevado.

El proyecto se ha logrado entregar, no del todo finalizado y por ello me queda un especial sabor agri dulce. Por ello voy a continuar hasta finalizar con el punto anterior de esta memoria, las futuras mejoras de la aplicación.

## 8.1 Valoración personal

La importancia que tiene hoy en día el sistema Android en la vida de muchísimas personas ha sido la principal basa por la que decidí decantarme en desarrollar este proyecto en esta plataforma. Después de la finalización del mismo, dura tarea que sobre todo ha ocupado mucho tiempo, ya que mi experiencia primero con el lenguaje java y segundo con las plataformas móviles es nula, puedo asegurar que ha sido totalmente enriquecedora.

Respecto a este proyecto, indicar que no acabo muy contento con el resultado, sobre todo por el esfuerzo que tuve que hacer a veces con parte del código, y que finalmente no ha quedado del todo bien. Las dudas que he ido encontrando durante el camino, las he resuelto con ayuda de paginas web donde muchos desarrolladores ponen información sin animo de lucro a compartir con todo aquel que quiera seguir sus lineas de implementación. Básicamente me he apoyado en las paginas que indico en la bibliografía.

Una grata sorpresa ha sido la utilización del entorno de desarrollo Eclipse con el plugin ADT. Aunque ya había escuchado criticas muy positivas hacia el, la facilidad de uso, comodidad y todo el potencial que tiene, realmente me han llegado a gustar mucho. Antes de intentar publicar esta aplicación en el google play me gustaría finalizar una parte que se queda sin desarrollar, el acceso a bases de datos externas, para el almacenamiento tanto de los usuarios, como de las ubicaciones, para que todos los usuarios que la utilicen puedan ver y compartir lugares de sus zonas. También cambiar un poco el aspecto visual de la misma, para que sea mas atractivo, y añadir otros lenguajes para que mas gente pueda usarla.

Mi intención, después de como he quedado de satisfecho es seguir en la linea de desarrollo de software para android y otras plataformas móviles, por supuesto mejorando ya que aun me veo muy novato, es continuar esta linea profesional y ampliar conocimiento en un futuro no muy lejano.

## 9. Glosario

### Android:

Sistema operativo libre y multiplataforma, basado en Linux. Desde su compra por parte de Google ha ampliado su mercado vertiginosamente hasta tal punto que hoy se considera un serio competidor del sistema IOs.

### Activity:

Componente de una aplicación Android que permite mostrar elementos gráficos en una pantalla y que permite la interacción con el usuario. Se define con el lenguaje XML.

### ADT:

Android Development Tools, es un plugin para Eclipse que está diseñado para darle un ambiente potente, integrado en el que la construcción de aplicaciones de Android.

### API:

Application Programming Interface (Interfaz de programación de aplicaciones) es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas.

### Array (vector):

es una zona de almacenamiento continuo, que contiene una serie de elementos del mismo tipo

### BBDD:

Base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

### Brainstorming:

Palabra inglesa (lluvia de ideas o tormenta de ideas) es una herramienta de trabajo grupal que facilita el surgimiento de nuevas ideas sobre un tema o problema determinado. La lluvia de ideas es una técnica de grupo para generar ideas originales en un ambiente relajado.

### Bundle:

Clase de Android que sirve para contener tipos primitivos y objetos de otras clases. Con esta clase puedes pasar datos entre distintas actividades.

### Clases:

En la programación orientada a objetos, una clase es una construcción que se utiliza como un modelo (o plantilla) para crear objetos de ese tipo.

### DCU:

Diseño Centrado en el Usuario es una filosofía de diseño que tiene por objetivo la creación de productos que resuelvan necesidades concretas de sus usuarios finales, consiguiendo la mayor satisfacción y mejor experiencia de uso posible con el mínimo

esfuerzo de su parte.

#### Dialog:

Son interfaces gráficas que solicitan al usuario las decisiones o información adicional requerida por la aplicación para continuar con una tarea.

#### Drawable:

Es una clase de android, una abstracción que representa “algo que puede ser dibujado”. Esta clase se extiende para definir gran variedad de objetos gráficos más específicos. Muchos de ellos pueden ser definidos como recursos usando ficheros XML.

#### Eclipse:

Plataforma de programación y compilación que soporta diferentes lenguajes. Soporta elementos gráficos y contiene, entre otros, un editor de código, un compilador y un depurador de código.

#### Geolocalización:

Se refiere al posicionamiento con el que se define la localización de un objeto espacial (representado mediante punto, vector, área, volumen) en un sistema de coordenadas y datos determinado. Este proceso es utilizado frecuentemente en los Sistemas de Información Geográfica

#### GPS:

Global Positioning System (sistema de posicionamiento global) es un sistema global de navegación por satélite que permite determinar en todo el mundo la posición de un objeto, una persona o un vehículo con una precisión hasta de centímetros aunque lo habitual son unos pocos metros de precisión.

#### Google Play (antes Android Market):

es una tienda de software en línea desarrollada por Google para los dispositivos con sistema operativo Android.

#### Implementar:

es la realización de una aplicación, instalación o la ejecución de un plan, idea, modelo científico, diseño, especificación, estándar, algoritmo o política.

#### Interfaz de usuario (IU):

es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo.

#### IOS (iPhone OS):

es un sistema operativo móvil de la empresa Apple originalmente desarrollado para el iPhone siendo después usado en dispositivos como el iPod Touch, iPad y el Apple TV. Apple no permite la instalación de iOS en hardware de terceros.

#### Key (clave):

Palabra inglesa. Es una forma de autenticación que utiliza información secreta para

controlar el acceso hacia algún recurso.

**Layout:**

Palabra inglesa que sirve para definir la distribución de los diferentes elementos que un programador incorpora al diseño de una pantalla.

**ListView:**

Vista que muestra los elementos de una lista de desplazamiento vertical.

**Login:**

En el ámbito de seguridad informática, login o logon (en español ingresar o entrar) es el proceso mediante el cual se controla el acceso individual a un sistema informático mediante la identificación del usuario utilizando credenciales provistas por el usuario.

**Multipataforma:**

es un atributo conferido a los programas informáticos o los métodos de cálculo y los conceptos que se ejecutan e interoperar en múltiples plataformas informáticas.

**Play Store:**

Repositorio de Google del software desarrollado con el sistema operativo Android.

**Prototipo:**

cualquier tipo de objeto en pruebas, diseñado para una demostración de cualquier tipo.

**RadioGroup:**

clase de android que se utiliza para crear un ámbito múltiple exclusión de un conjunto de RadioButtons (es un botón de dos estados que se pueden activar o desactivar)

**RatingBar:**

clase de android con la que se puede obtener una calificación, según se configure.

**Repositorio:**

es un sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos.

**SD:**

de Secure Digital es un formato de tarjeta de memoria inventado por Panasonic.

**SQLite:**

Sistema gestor de base de datos relacionales que se integra con el programa que lo utiliza para ser usados con dispositivos que requieran una gestión eficiente del sistema de almacenamiento. Es el sistema gestor de bases de datos que usa Android.

**Sketches:**

es una escena pintada a mano.

**Smartphone:**

es un teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de almacenar datos y realizar actividades semejantes a una mini computadora y conectividad que un teléfono móvil convencional.

**SO (OS):**

siglas de Sistema Operativo Operating System.

**Spinner:**

una clase de android que permite al usuario elegir entre varios objetos.

**String:**

En programación, una cadena de caracteres, palabra, ristra de caracteres o frase es una secuencia ordenada de longitud arbitraria (aunque finita) de elementos

**View:**

clase de android que representa el elemento básico para los componentes de interfaz de usuario, la vista.

**XML:**

Xtension Markup Language es un lenguaje de marcas desarrollado que permite estructurar documentos y compartir información eficientemente entre varias plataformas. Android lo usa para estructurar sus Layouts.



## 10. Bibliografía

La bibliografía básicamente ha sido la utilización de recursos online. No he utilizado ningún libro para la realización de este proyecto.

### FORMACION

#### **Android: Programación para aplicaciones móviles**

<http://www.androidcurso.com/>

Página de la Universidad Politécnica de Valencia que incluye en recursos didácticos una serie de tutoriales y videotutoriales con una información general sobre la plataforma Android.

#### **Curso online de programación en Android, impartido por Salvador Gómez**

[http://www.sgoliver.net/blog/?page\\_id=2935](http://www.sgoliver.net/blog/?page_id=2935)

Blog de Salvador Gómez donde puedes encontrar un curso online con todos los conceptos de la programación en Android

### CONSULTAS WEB

#### **Android-spa**

<http://www.android-spa.com/>

Página de una comunidad que contiene noticias, foros, manuales, tutoriales (español)

#### **Android developers**

<http://developer.android.com/index.html>

Página oficial de Android que contiene servicios de google, herramientas, guías de todas las API del sistema, referencias de las API, ejemplos (ingles)

#### **El Androide libre**

<http://www.elandroidelibre.com/>

Blog de actualidad de Android, donde se pueden obtener tutoriales muy prácticos (español)

#### **Stackoverflow**

<http://stackoverflow.com/>

Página de programadores con foros para resolver dudas de programación de muchísimos lenguajes