

---

# **Ranking Tapas**

**Aplicación Móvil para la votación y el conocimiento de las tapas leonesas basado en tecnología Android**

**Victoria Alicia Marcos Sbarbaro**

Ingeniería Técnica en Informática de Sistemas

**Consultor: Roberto Ramirez Vique**

**Septiembre 2013**

*A todas las personas excelentes que se encuentran a mi lado, en especial a mi padre que se fue hace poco y del cuál estoy muy orgullosa.*

## Resumen

Esta documentación corresponde a la memoria de Proyecto Fin de Carrera basado en el desarrollo de una aplicación para un dispositivo móvil para la plataforma Android. Se tratará de la realización de una aplicación basada en un contexto social y divertido. La aplicación se llamará "Ranking Tapas" y será un programa ameno y sencillo cuyo objetivo será el de compartir información entre usuarios donde quiera que se encuentren y la interacción entre los mismos mediante votación.

Para ello será necesario estudiar tanto el mercado actual para dispositivos y aplicaciones, así como esta plataforma para conocer sus principales características y el funcionamiento de su arquitectura y, comprender, cómo se debe desarrollar en ella. Se usará la plataforma Topoos, basada en Android, que ofrece servicios de geolocalización y dependencia del contexto de manera rápida y sencilla, dos de las características importantes de la aplicación.

También se realizará un análisis de los requisitos necesarios para realizar el diseño y desarrollo de la aplicación. En esta documentación se recogen todos los pasos seguidos para completar el proyecto.

## Índice general

Resumen .....	3
Índice general .....	4
1. Introducción y objetivos .....	6
1.1 Introducción.....	6
1.2 Objetivos.....	6
1.3 Medios empleados .....	7
1.4 Estructura de la memoria.....	7
2 Estudio de Viabilidad .....	8
2.1 Situación actual .....	8
2.2 Potencial del mercado. Alcance del Proyecto .....	8
2.3 Evaluación inicial de la tecnología disponible .....	9
2.4 Evaluación inicial de coste-beneficio .....	10
3 Análisis de Riesgos.....	11
3.1 Planificación temporal.....	11
3.2 Número de usuarios .....	11
3.3 Visualización y funcionamiento en distintos dispositivos.....	12
4 Calendario del Proyecto.....	12
5 Requisitos .....	13
5.1 Contenido .....	13
5.2 Presentación.....	13
5.3 Usuarios a considerar.....	14
5.4 Requisitos funcionales.....	14
5.4.1 Compartir.....	14
5.4.2 ListarTapasEventos .....	14
5.4.3 ListarMisTapas .....	14
5.4.4 Visualizar Mapa.....	14
5.4.5 Ver .....	14
5.4.6 Votar .....	14
5.4.7 Publicar.....	14
5.5 Requisitos no funcionales.....	15
5.5.1 Requisitos de Interfaz .....	15
5.5.2 Requisitos de información .....	15
6 Análisis del Sistema .....	15
6.1 Diagramas de casos de uso .....	15
6.2 Descripción textual de casos de uso .....	18
6.2.1 CU01 – compartir .....	18
6.2.2 CU02 – listarTapasEventos .....	18
6.2.3 CU03 – listarMisTapas .....	19
6.2.4 CU04 – visualizarMapa .....	20
6.2.5 GT01 – compartirTapa .....	20
6.2.6 GT02 – verTapa.....	21
6.2.7 GT03 – hacerFoto .....	22
6.2.8 GT04 – introducirComentario .....	23
6.2.9 GT05 – votar.....	23
6.2.10 GT06 – publicar.....	24
6.2.11 GE01 – compartirEvento .....	25

6.2.12 GE02 – verEvento .....	25
6.2.13 GE03 – introducirComentario .....	26
6.2.14 GE04 – publicar .....	27
7 Diseño .....	27
7.1 Consideraciones previas .....	27
7.1.1 Visualización en distintos dispositivos .....	27
7.2 Diseño de la Arquitectura del Sistema .....	29
7.3 Diseño de la Interfaz de usuario y Navegación de la aplicación .....	31
7.3.1 Esquema general .....	31
7.3.2 Elementos de Navegación .....	32
7.3.3 Pantalla inicial .....	33
7.3.4 Pantalla compartir tapa y/o evento .....	33
7.3.5 Pantalla listar todas las tapas y eventos del sistema .....	35
7.3.6 Pantalla visualizar Mapa .....	38
7.3.7 Pantalla listar sólo tapas y eventos del usuario del móvil .....	38
8 Implementación del Sistema .....	39
8.1 Entorno de desarrollo .....	40
8.2 Google Maps V2 .....	40
8.2.1 Descargar y configurar el SDK de Google Play servicios .....	41
8.2.2 Obtener una clave de Android para la API de Google Maps .....	42
8.2.3 Especificar permisos y ajustes en el Manifest de la aplicación .....	45
8.2.4 Añadir el control del mapa de un proyecto Android nuevo o existente .....	48
8.3 Configuración Topoos .....	48
8.3.1 Registro como desarrollador en la plataforma de Topoos .....	49
8.3.2 Crear nueva aplicación en Topoos .....	49
8.3.3 Instalación de Topoos SDK .....	49
8.4 Estructura de la aplicación y manifiesto .....	50
8.5 Base de Datos .....	51
8.6 AndroidManifest .....	52
8.7 Plataforma Google Analytics .....	54
8.8 Actividad Principal .....	56
8.9 Menú de dispositivo móvil .....	59
8.10 Código de pantallas .....	59
8.10.1 Pantalla Inicial .....	60
8.10.2 Pantalla 1 .....	61
8.10.3 Pantalla 2 .....	62
8.10.4 Pantalla 3 .....	64
8.10.5 Pantalla 4 .....	64
8.10.6 Votaciones .....	65
8.11 Pruebas e Instalación .....	67
8.11.1 Pruebas en emulador .....	67
8.11.2 Instalación .....	68
8.11.3 Acceso a servicios de TOPOOS .....	68
9 Simulación .....	69
10 Conclusiones .....	69
11 Referencias bibliográficas .....	71
12 Fuentes de Información .....	71

# 1. Introducción y objetivos

## 1.1 Introducción

El presente documento recoge toda la información del desarrollo llevado a cabo para el proyecto final de carrera. Este proyecto surge a partir de la motivación de desarrollar una aplicación para una plataforma móvil que permita al usuario compartir información referente a tapas de bares y eventos en los mismos de forma geolocalizada, es decir, poder acceder a la localización de los mismos.

Podrá hacer uso de las redes sociales, tan demandadas hoy en día, teniendo la opción de publicar dicha información en ellas.

Con este objetivo global, se ha utilizado para el desarrollo la plataforma Android al ser una de las plataformas con mayor ratio de crecimiento del mercado y por permitir, mediante interfaces, crear todas las funcionalidades que sean necesarias para el desarrollo de esta aplicación como se podrá ver a lo largo del documento.

También se ha utilizado la plataforma Topoos, basada en Android, que recoge las características de geolocalización y dependencia del contexto de la aplicación, a la vez que aporta almacenaje de PDI (puntos de interés), otro tipo de almacenaje, reglas y eventos.

## 1.2 Objetivos

El objetivo principal de este proyecto es desarrollar una aplicación para un dispositivo móvil que facilite al usuario el acceso y distribución de la información de las tapas de los bares de su ciudad mediante fotos de las mismas y comentarios sobre ellas, permitiéndole puntuarlas y así, de forma divertida, conectar con la opinión de todos los usuarios que tengan dicha aplicación instalada en su móvil.

A su vez, podrá intercambiar información de posibles eventos en los locales de la ciudad.

El tiempo de ocio es importante y en una ciudad como León, que presume de tener ricas tapas en sus bares, esta aplicación puede ayudar a los habitantes de la misma y a los extranjeros a disponer de esta información.

Serán los propios usuarios los que alimenten los datos del programa.

Quien dice León dice otra ciudad pues, por la característica de geolocalización, haciendo uso de Google Maps, el usuario, al agregar la tapa y/o evento, almacenará automáticamente estos puntos de interés en el mapa de la ciudad cuando haga los registros de las tapas y eventos en la aplicación.

Se mencionó en la anterior entrega el registro del usuario en la aplicación como una de las características de la misma, pero haciendo el análisis de los requisitos y el diseño, se ha llegado a la conclusión de que este registro no es necesario. No importa quién introduce la información en la aplicación sino el dato en sí, por lo tanto tendrá carácter anónimo.

También se hizo mención de una posible configuración por parte del usuario de los puntos necesarios para llegar tanto al nivel de mejor tapa, denominado ahora TAPA

ESTRELLA, como al nivel de peor tapa, TAPA CALAVERA. Viendo y mejorando las características de la aplicación se decide suprimir esta configuración por parte del usuario, dejándola de forma predeterminada por parte del desarrollador consiguiendo así el objetivo de la sencillez del uso del programa.

Además, mientras no tenga ninguno de esos niveles (dependiente de los puntos de los votos) obtendrá el nivel de TAPA NORMALITA

Por lo tanto los principales objetivos del proyecto son:

- Crear una aplicación informática en la que el usuario podrá almacenar información geográfica de las tapas de su interés y eventos que ocurran en los bares de la ciudad, y que a su vez obtendrá esta información de otros usuarios de la aplicación. Asimismo podrá votar las tapas pudiendo llegar a obtener tres niveles, nivel TAPA ESTRELLA, nivel TAPA CALAVERA o NIVEL TAPA NORMALITA para darle un carácter divertido a la aplicación. Todo esto gracias a Topoos
- Dicha aplicación debe estar disponible para terminales móviles con tecnología Android
- Montar el entorno de desarrollo para realizar esta aplicación con los elementos necesarios
- Utilizar recursos y metodologías propias del *Software Libre*, de tal forma que el software desarrollado y el conocimiento generado sea de utilidad para la comunidad.

### 1.3 Medios empleados

Para llevar a cabo el desarrollo de este proyecto será necesario el uso de los siguientes elementos de software:

- Un sistema operativo: Windows Vista de 64 bits
- Herramienta de desarrollo: IDE Eclipse Helios 3.6.2 con las herramientas de Android en el plug-in ADT version 22.2..1.v2
- Librerías de desarrollo: JAVA SDK 6, Android SDK rev, 18.0.1, SDK Topoos v1.0.0.12, Google Maps Android API v2, SDK de AdMob
- Herramienta online de diseño de interfaz de la aplicación: FluidUI
- Microsoft Office 2003
- ArgoUML para realizar los diagramas de caso de uso

### 1.4 Estructura de la memoria

Esta es la estructura de la memoria en donde podemos ver los capítulos que hay y el contenido que tienen:

**RESUMEN:** aquí disponemos de un pequeño resumen del proyecto, donde se expone de forma breve en qué consiste y las tareas que se han realizado.

**INTRODUCCIÓN:** en este capítulo podemos ver los objetivos principales del proyecto, los medios empleados y la estructura de esta memoria.

**ESTUDIO DE VIABILIDAD:** en este capítulo hay un estudio donde se tratan los siguientes temas:

Se describe la situación actual para este tipo de aplicaciones.

Se estudian los **usuarios potenciales** que pueden tener la aplicación y el plan inicial para lograr llegar al mayor número de usuarios.

Evaluación de la **tecnología y documentación disponibles** para realizar el proyecto.

Estudio de los **beneficios** que se podrían obtener con el proyecto.

**ANÁLISIS DE RIESGOS:** en este capítulo se estudian los posibles riesgos que tiene el proyecto, tanto en su etapa de desarrollo como en su distribución y su instalación en

distintos tipos de dispositivos. Se analiza la forma de evitar que sucedan esos riesgos y las posibles acciones correctoras.

**CALENDARIO DEL PROYECTO:** en este capítulo se describe el tiempo empleado en el desarrollo de los puntos principales del proyecto. Los puntos que se han obviado quedan recogidos en la temporalización de los principales.

**REQUISITOS:** en este capítulo se describen los requisitos iniciales de la aplicación. Se define el contenido que debe tener, los usuarios a considerar y los requisitos tanto funcionales como no funcionales.

**ANÁLISIS DEL SISTEMA:** en este capítulo se detallan los diagramas de caso de uso y la descripción textual para los mismos.

**DISEÑO:** en este capítulo se realiza el diseño de la aplicación para que pueda implementarse después. Se incluyen los siguientes apartados:

**Consideraciones previas:** en este apartado se estudian distintas características de Android que será necesario tener en cuenta para la elaboración del proyecto, como la visualización de las aplicaciones en distintos tipos de dispositivos.

**Diseño de la arquitectura:** en este apartado se presenta un diseño de la aplicación a alto nivel y se detallan los módulos correspondientes.

**Diseño Interfaz de usuario y navegación:** en este apartado se define el esquema principal de navegación junto con sus elementos (botones e iconos), así como el diseño de las pantallas de la aplicación.

**CONCLUSIONES:** en este apartado se aportan los diferentes puntos de vista sobre la realización del proyecto.

**REFERENCIAS BIBLIOGRÁFICAS:** en este capítulo se incluyen las referencias bibliográficas utilizadas para el desarrollo del proyecto.

**FUENTES DE INFORMACION:** en este capítulo se exponen las diferentes fuentes de información consultadas durante la realización de este proyecto.

Queda por incluir en este documento los capítulos referentes a la **IMPLEMENTACIÓN** y el **CODIGO FUENTE**

## 2 Estudio de Viabilidad

### 2.1 Situación actual

Hoy en día la gente opta por instalar, cada vez más, aplicaciones en sus móviles que les permita acceder y compartir información de todo tipo en cualquier momento y lugar. Si esta información está enfocada al ocio y se puede compartir de forma sencilla, divertida y rápida, puede generar gran interés por los demandantes.

Esta aplicación pretende intercambiar la opinión de los usuarios sobre las tapas de los bares de una ciudad e informar sobre los posibles eventos festivos de la misma de forma divertida e interactiva, valiéndose de las últimas tecnologías disponibles en teléfonos inteligentes formado por una mezcla de texto, imágenes, notificaciones y puntos de interés.

### 2.2 Potencial del mercado. Alcance del Proyecto

Al desarrollarse una aplicación libre para móviles que incorporen el sistema operativo Android, estamos hablando de una cantidad potencial de usuarios muy grande.

Si en 2013 hay un sistema que esté ganando ventaja frente a la competencia es Android. Sigue siendo el número uno en telefonía móvil y hay rumores de que ha destronado a Apple en el terreno de las tabletas.



Las estadísticas del grupo Gartner dejan entrever que el sistema de Google sigue aumentando su presencia: el 79% de los teléfonos inteligentes que se vendieron entre abril y junio lo tienen instalado (hace un año, la cifra era del 64,2 %), frente al 14,2% de iOS (de Apple), el 3,3% de Windows Phone (de Microsoft) y el 2,7% de Blackberry.

Suma y sigue, porque la empresa de investigación Canalys asegura que las tabletas con el sistema Android ocuparon el 53% del mercado en el segundo trimestre (se comercializaron unos 18 millones), aupado por las ventas de la surcoreana Samsung, la estadounidense Amazon, la china Lenovo y la taiwanesa Acer.<sup>1</sup>

Según App Date al año se descargan casi cuatro veces más aplicaciones de Android que de iOS. En el año 2013 se descargaron un total de 87 millones de aplicaciones de Android en España, que superan con creces los 24 millones que se realizan en iOS en el mismo periodo. Aún así, son menos aplicaciones proporcionalmente.

Entre otros datos han analizado que uno de cada tres españoles va con el smartphone al baño, que un 80% no deja de prestarle atención y que muy pocos lo dejan en casa durante las vacaciones, siendo solamente 1.5% de los españoles quienes consiguen desenganchar. Esto puede ser precisamente a que el smartphone nos da información relativa al viaje, sin tener que ser por motivos laborales.<sup>2</sup>

El resultado que hay que obtener con la aplicación es llegar al mayor número posible de usuarios y que estos la usen el mayor tiempo posible.

Otro posible objetivo será el de contribuir al aumento de facturación de los bares de una ciudad debido a la popularidad de sus tapas y eventos realizados en ellos, pudiendo mejorar la economía en este tiempo tan marcado con la crisis.

### ***2.3 Evaluación inicial de la tecnología disponible***

Es muy fácil tener acceso a la tecnología para el desarrollo de aplicaciones para móviles con sistema operativo Android. Las herramientas de desarrollo necesarias son:

- El SDK de Android, que proporciona las librerías API necesarias.
- Entorno de desarrollo Eclipse, para el cual disponemos de un plugin llamado Android Development Tools (ADT), diseñado para proporcionar un entorno potente e integrado donde desarrollar aplicaciones. Este plugin aumenta las capacidades de Eclipse para poder añadir nuevos proyectos Android de forma rápida, y de esa forma poder crear, compilar y depurar aplicaciones, así como exportarlas para su distribución.
- Dispositivo virtual de Android, si se quiere desarrollar sin disponer de un terminal físico o si se quieren realizar las pruebas sobre un dispositivo virtual. Todas estas herramientas se pueden descargar de forma gratuita para los sistemas operativos más utilizados, incluyendo Linux, Windows y Mac OS.
- SDK de Topoos que proporciona los servicios necesarios de geolocalización, almacenamiento de PDIs y notificaciones necesarias en la aplicación, para ello será necesario registrarnos en Topoos y obtener unos Tokens.
- API Google Maps v2 necesaria para obtener los certificados necesarios tanto para desarrollar y depurar como para publicar y distribuir una aplicación Google.
- SDK de AdMob de Google para Android versión 6.2.1 para poder incluir publicidad en la aplicación

Hay una gran cantidad de documentación disponible sobre estas herramientas:

- Documentación oficial, donde se explican todos los componentes disponibles así como la arquitectura, incluyendo gran cantidad de código de ejemplo.
- Foros online, donde distintos desarrolladores proponen soluciones para distintos problemas planteados por otros desarrolladores.

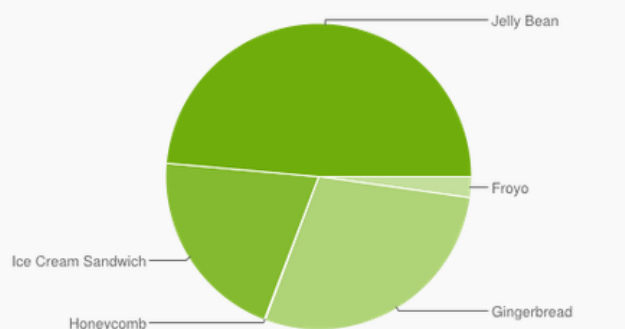
Hay distintas versiones de Android, cada versión incorpora nuevas funcionalidades. Las aplicaciones se desarrollan con una determinada versión y sólo pueden instalarse en dispositivos que soporten esa versión o posteriores.

Estos datos reflejan los dispositivos con la última aplicación Google Play Store, que es compatible con Android 2.2 y superior. Cada instantánea de datos representa todos los dispositivos que visitaron la tienda Play Google durante un período de 7 días que terminó el 2 de octubre de 2013.

No se muestran Cualquiera de las versiones con menos de 0,1% de distribución.

**Nota:** A partir de septiembre de 2013, los dispositivos que ejecutan versiones anteriores a Android 2.2 no aparecen en esta información porque esos dispositivos no son compatibles con el nuevo Google Play Store aplicación. Sólo la nueva aplicación es capaz de medir la cantidad de dispositivos que visitan activamente Google Play Store y creemos que esta medida refleja mejor su potencial base de usuarios.

Version	Codename	API	Distribution
2.2	Froyo	8	2.2%
2.3.3 - 2.3.7	Gingerbread	10	28.5%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	20.6%
4.1.x	Jelly Bean	16	36.5%
4.2.x		17	10.6%
4.3		18	1.5%



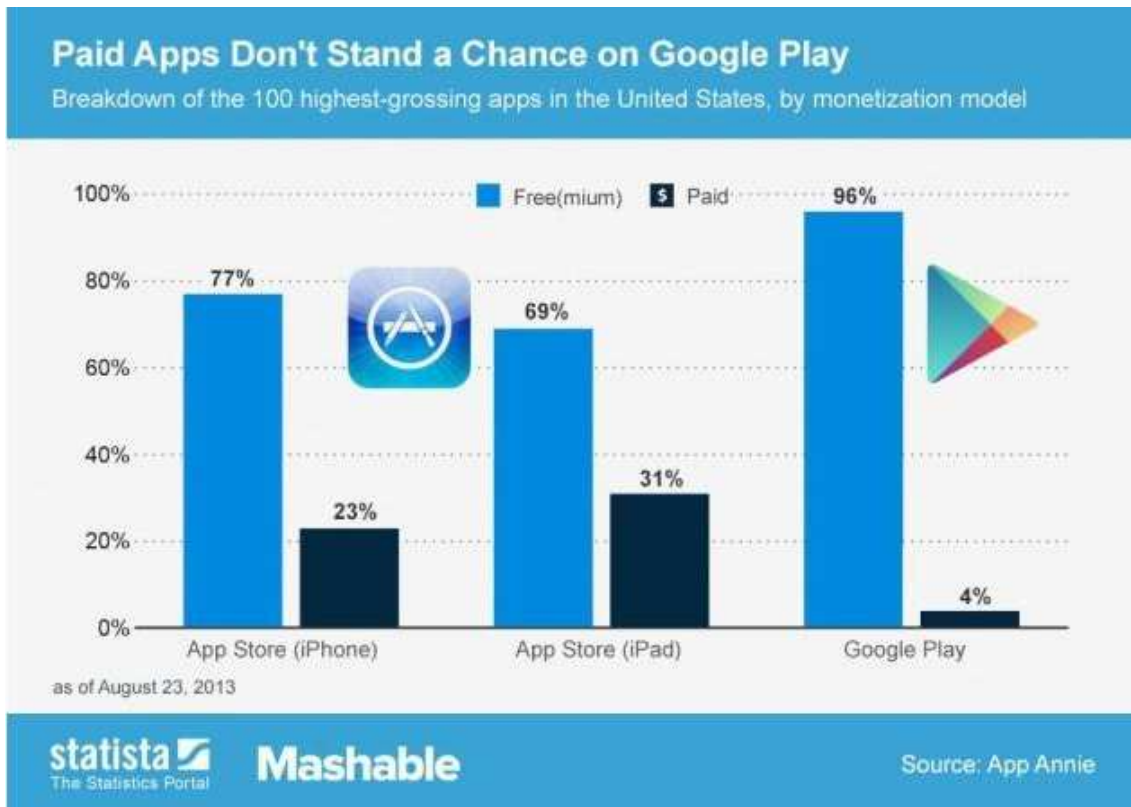
La aplicación está enfocada para usuarios de smartphones y, dado que aún se usan móviles con versión Android 2.2, el mío en particular tiene una versión 2.3.5, se realizará la aplicación para la versión 2.2 y posteriores.

## 2.4 Evaluación inicial de coste-beneficio

Al tratarse de una aplicación de software libre es necesario estudiar los distintos modelos de negocio que puede ofrecer este tipo de software.

Las aplicaciones Android que más beneficios consiguen son las “gratuitas”. App Annie ha realizado una gráfica en la que muestra como estadísticamente el 96% de las aplicaciones del Top 100 – según descargas – de Play Store son gratuitas frente al 77% de iPhone o bien un 69% de iPad.

Bien sean aplicaciones que ofrecen compras dentro de la aplicación o bien que estén bien plagadas de publicidad, queda patente que destacan sobre el resto en cuanto a popularidad y descargas.<sup>3</sup>



Por lo tanto el modelo popular a usar sería el freemium<sup>4</sup> o bien financiando con publicidad.

En nuestro caso incluiremos publicidad gracias al SDK de AdMob para Android, librería proporcionada por Google.

Esto supondría que para obtener beneficios sería necesario que mucha gente se descargara la aplicación.

## 3 Análisis de Riesgos

### 3.1 Planificación temporal

Este riesgo consiste en que el proyecto lleve más del tiempo establecido o que no se puedan hacer las entregas en el plazo fijado provocando la modificación de las funcionalidades de la aplicación:

**Capacidades que es necesario adquirir:** para realizar el proyecto será necesario adquirir algunas capacidades propias del desarrollo de aplicaciones para Android en un tiempo breve, que podría no ser suficiente teniendo en cuenta la planificación del proyecto. Debido a que se trata de un proyecto nuevo en un entorno a conocer, podría ser necesario modificar alguna funcionalidad para conseguir la misma de una forma más sencilla.

Para cubrir este riesgo se cuenta con la experiencia del programador y la gran cantidad de documentación, tanto oficial de Android y Topoos, así como una gran cantidad de foros de ayuda de programadores.

### 3.2 Número de usuarios

Este riesgo consiste en que no alcancemos el número de usuarios esperado. La posible causa para que se de este riesgo es:

**Presentación inadecuada de la aplicación:** una vez que los usuarios han visto que existe la aplicación necesitamos que se la descarguen, para ello tendremos que presentarles las ventajas de la aplicación y las suficientes imágenes para que les resulte atractiva.

### 3.3 Visualización y funcionamiento en distintos dispositivos

Existe el riesgo de que la aplicación se vea diferente o funcione de forma diferente de unos dispositivos a otros. Hay una gran cantidad de modelos diferentes de dispositivos móviles con sistema operativo Android en los que podría instalarse esta aplicación, además de pertenecer también dichos modelos a muchas marcas distintas. Cada dispositivo tiene sus propias características técnicas.

Para intentar evitar esto, se podrá probar la aplicación en dispositivos virtuales proporcionados por el SDK de Android pero aún así, podría fallar en dispositivos reales.

Como se va a hacer uso de Google Maps, podría no cargar bien el mapa debido a los certificados necesarios para ello.

A la hora de implementar la aplicación y probarla en diferentes dispositivos móviles y emuladores, se ha detectado que es necesario que los móviles tengan actualizado Google Maps como mínimo a la versión 2 y que, a su vez, dispongan de las aplicaciones de Google Play Store y Google Play Services.

Se seguirán las recomendaciones oficiales de Android a la hora de emplear componentes de desarrollo y se contará con la documentación técnica oficial.

## 4 Calendario del Proyecto

El siguiente calendario muestra la temporalización inicial de las tareas relevantes que componen el proyecto.

<b>Análisis previo y planificación</b>	12 días	19/09/2013	30/09/2013
Selección y propuesta del proyecto	4 días	19/09/2013	24/09/2013
Preparación del proyecto	4 días	25/09/2013	28/09/2013
Planificación del proyecto	1 día	28/09/2013	28/09/2013
Creación del documento	3 días	28/09/2013	30/09/2013
<b>Entrega PEC1 (1/10/2013)</b>	Hito	30/09/2013	
<b>Cambios en documento para completar PEC1</b> Viabilidad, Riesgos y otros puntos	4 días	10/10/2013	14/10/3013
<b>Análisis de Requisitos</b>	5 días	14/10/2013	18/10/2013
Especificación requisitos funcionales	4 días	14/10/2013	17/10/2013
Especificación requisitos no funcionales	1 día	17/10/2013	18/10/2013
<b>Análisis de Sistema</b>	2 días	18/10/2013	19/10/2013
Diagramas de casos de uso	1 día	18/10/2013	18/10/2013
<b>Diseño</b>	14 días	19/10/2013	29/10/2013

Diseño de Arquitectura	2 días	19/10/2013	21/10/2013
Diseño de interfaces	5 días	20/10/2013	25/10/2013
Preparación del documento	7 días	23/10/2013	29/10/2013
<b>Entrega PEC2 (29/10/2013)</b>	Hito	29/10/2013	
Montaje del entorno	2 días	3/10/2013	5/10/2013
<b>Implementación</b>	43 días	30/10/2013	7/12/2013
<b>Pruebas y correcciones</b>	11 días	29/11/2013	9/12/2013
Preparación del documento y entregables	5 días	5/12/2013	10/12/2013
<b>Entrega PEC3 (10/12/2013)</b>	Hito	10/12/2013	
Pruebas y correcciones del código y documento	4 días	20/12/2013	24/12/2013
Instalación y pruebas de Camtasia 8	2 días	26/12/2013	28/12/2013
<b>Preparación del vídeo y montaje</b>	2 días	29/12/2013	2/1/2013
Preparación del código, ficheros y entregables	3 días	4/12/2013	7/12/2013
<b>Entrega Final (8/01/2014)</b>	Hito	8/01/2014	

## 5 Requisitos

### 5.1 Contenido

La aplicación debe ser capaz de almacenar y visualizar la información introducida por el usuario mediante fotos de las tapas y comentarios de la misma.

Es necesario que el usuario aporte foto de la tapa (esta foto la puede realizar en el momento antes de publicar o disponer de ella en la galería de imágenes de su móvil, ambas opciones estarán disponibles, pero sin foto no podrá publicar esta información). También es necesario que el usuario aporte comentario o breve descripción de la tapa para que pueda ser publicada.

Dicha información será introducida automáticamente en el mapa en forma de punto de interés, por lo tanto es necesario acceder a ciertos permisos como la ubicación del usuario pudiendo ser esta detallada, mediante GPS, o basada en red, es decir no detallada, mediante la comunicación por red. Por lo tanto será necesario que el usuario pueda acceder a datos de red, sino se producirá un error de problema de conexión en la ejecución de la aplicación.

De la misma forma, al publicar un evento, también se almacenará esta información de forma geolocalizada y será necesario aportar una breve descripción del evento para que este pueda ser publicado. No es necesario aportar fotografía del evento, de hecho no habrá opción de proporcionar fotografía por parte del usuario.

### 5.2 Presentación

La aplicación será presentada de forma divertida y atractiva en cuatro sencillas pantallas.

El uso de la aplicación debe ser sencillo e intuitivo y será de forma táctil.

El acceso a cada una de las pantallas se hará a través de la pulsación de pestañas que formarán el menú de la aplicación.

Se podrá publicar la información compartida a través de programas de carácter social si estuvieran instalados en el móvil tales como el whatsapp y el facebook, de la misma

forma se podrá distribuir esta información mediante correo electrónico si existe previamente en el dispositivo móvil. El envío de la información se hará a través de botones configurados en la aplicación. El diseño de la interfaz así como la navegación tanto del menú y opciones disponibles se verá en el apartado de diseño de dicho documento.

### **5.3 Usuarios a considerar**

El usuario que usará la aplicación será un usuario anónimo y tendrá acceso a las funcionalidades de la misma.

### **5.4 Requisitos funcionales**

Las siguientes funcionalidades estarán presentes en la aplicación.

#### **5.4.1 Compartir**

Permitirá al usuario anónimo agregar información de una tapa y o evento a la aplicación aportando comentarios sobre los mismos y la foto de la tapa.

*Nota: la foto y los comentarios para la tapa son necesarios. Los comentarios para el evento son necesarios*

#### **5.4.2 ListarTapasEventos**

El usuario podrá obtener un listado de todas las tapas y todos los eventos compartidos por todos los usuarios de la aplicación.

#### **5.4.3 ListarMisTapas**

El usuario podrá obtener un listado de todas sus tapas y todos sus eventos compartidos en la aplicación. Mediante esta opción también podrá ver qué nivel de tapa tiene según las votaciones obtenidas por todos los usuarios pudiendo ser estos TAPA ESTRELLA, TAPA NORMALITA o TAPA CALAVERA

#### **5.4.4 Visualizar Mapa**

Gracias a esta funcionalidad, el usuario podrá visualizar un mapa con los lugares donde se encuentran las tapas y eventos agregados por todos los usuarios más cercanos a él

#### **5.4.5 Ver**

Con esta funcionalidad el usuario podrá visualizar una tapa y/o evento en particular

#### **5.4.6 Votar**

Gracias a esta funcionalidad, el usuario podrá votar positiva o negativamente una tapa previamente visualizada

#### **5.4.7 Publicar**

Con esta funcionalidad el usuario podrá publicar la tapa o el evento a compartir en la aplicación entre sus amigos mediante correo electrónico, whatsapp, facebook o cualquier aplicación de carácter social que tenga previamente instalada en el móvil

## 5.5 Requisitos no funcionales

### 5.5.1 Requisitos de Interfaz

El sistema contará con un interfaz diseñado para funcionar en dispositivos móviles Android con acceso a las funcionalidades principales de la aplicación.

### 5.5.2 Requisitos de información

La información de las tapas y los eventos que deberá considerarse es:

- Fotografía de la tapa
- Comentarios necesarios para la publicación de la tapa
- Comentarios necesarios para la publicación del evento
- Situación geográfica de la tapa
- Situación geográfica del evento
- Nivel de votación de la tapa

## 6 Análisis del Sistema

Esta sección recoge el resultado de la fase de análisis del sistema. Para ello, partiendo de la vista general aportada por los diagramas de casos de uso, se detallará cada uno de ellos y se formalizarán y concretarán los requisitos no funcionales expresados en el apartado anterior.

### 6.1 Diagramas de casos de uso

El siguiente diagrama recoge una vista global de los actores y casos de uso que describen los requisitos funcionales de la aplicación:

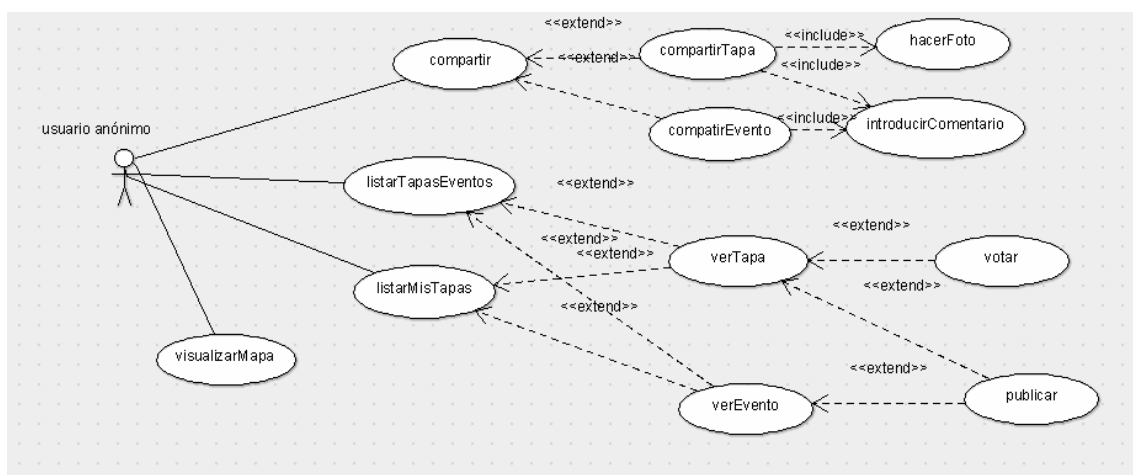
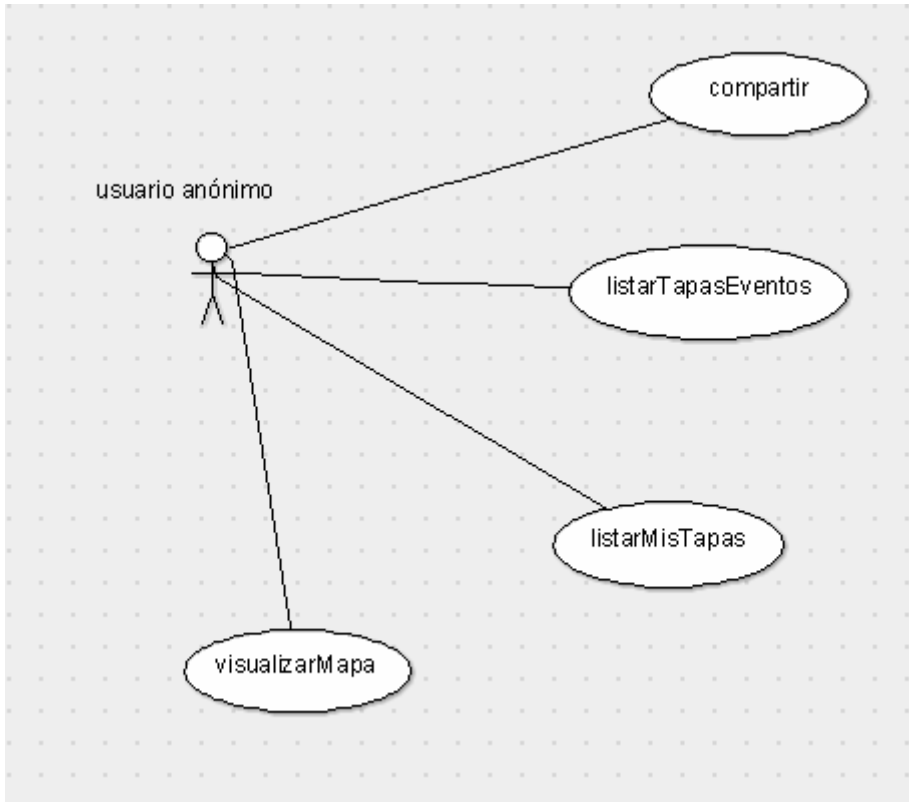


Diagrama general

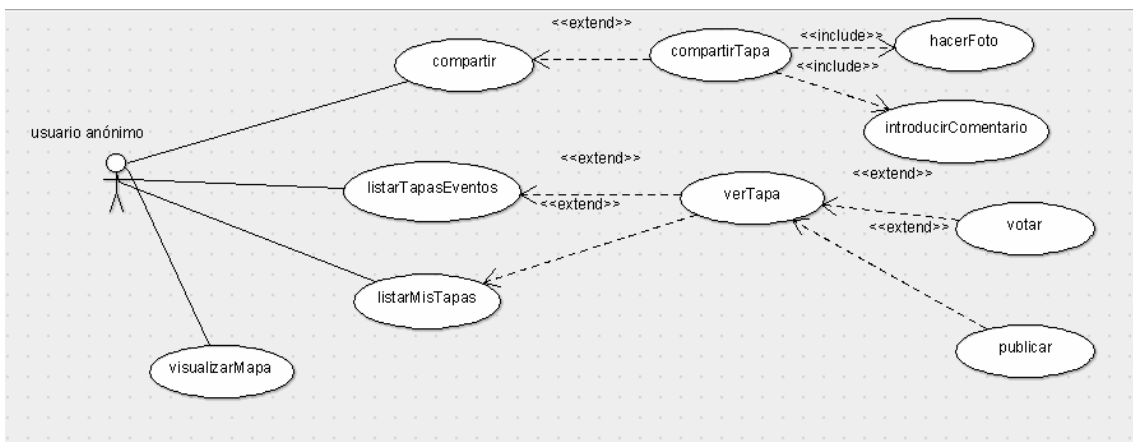
Como se observa, en el sistema sólo actúa un actor y éste es el usuario anónimo de la aplicación, es decir, el dueño del móvil que tiene la aplicación instalada pero tiene carácter anónimo pues no se identifica de ninguna forma.

Internamente, durante la implementación, posiblemente sí que se inserte la información de la tapa y evento a compartir mediante un identificador, pero esto es transparente para el usuario de la aplicación.

Para facilitar la lectura, he agrupado en varios paquetes las funcionalidades principales del sistema, las que afectan a las tapas y las que afectan a los eventos.

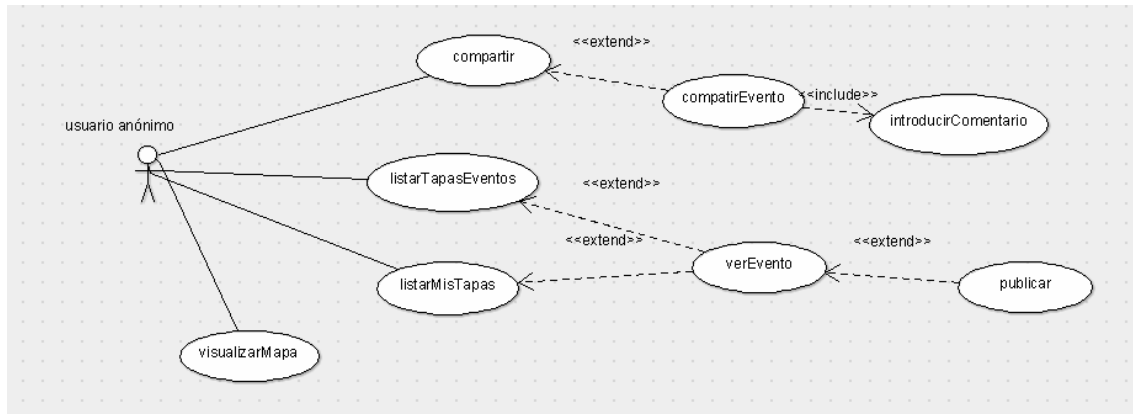


*Paquete principal de casos de uso*



*Paquete Gestión de Tapas*





Paquete Gestión de Eventos

La siguiente tabla resume los casos de uso contemplados, que serán desarrollados en la sección posterior.

Código	Nombre	Descripción	Actor
CU01	compartir	Compartir una tapa y/o evento	Usuario anónimo
CU02	listarTapasEventos	Listar todas las tapas y los eventos publicados en la aplicación por todos los usuarios de la misma	Usuario anónimo
CU03	listarMisTapas	Listar todas las tapas y eventos publicadas por el propio usuario del móvil	Usuario anónimo
CU04	visualizarMapa	Visualizar el mapa para ver los lugares donde se encuentran las tapas y eventos más cercanos al usuario	Usuario anónimo
GT01	compartirTapa	Compartir una tapa	Usuario anónimo
GT02	verTapa	Ver una tapa en particular	Usuario anónimo
GT03	hacerFoto	Hacer una foto de la tapa o elegir una foto de la tapa desde la galería de imágenes del móvil del usuario	Usuario anónimo
GT04	IntroducirComentario	Introducir breve descripción de la tapa a compartir	Usuario anónimo

GT05	votar	Votar a favor o en contra de la tapa	Usuario anónimo
GT06	publicar	Publicar la tapa por correo, red social, etc...	Usuario anónimo
GE01	compartirEvento	Compartir un evento	Usuario anónimo
GE02	verEvento	Ver un evento en particular	Usuario anónimo
GE03	introducirComentario	Introducir breve descripción del evento a compartir	Usuario anónimo
GE04	publicar	Publicar el evento por correo, red social, etc...	Usuario anónimo

## 6.2 Descripción textual de casos de uso

### 6.2.1 CU01 – compartir

<b>Identificador</b>	CU01
<b>Nombre</b>	compartir
<b>Resumen</b>	Este caso de uso sirve para compartir una tapa y/o evento en el sistema
<b>Actor</b>	Usuario anónimo
<b>Precondiciones</b>	Aplicación Android instalada, Conexión a Internet. El usuario ha entrado en la aplicación
<b>Postcondiciones</b>	Entra en la pantalla de compartir tanto foto como evento
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El caso se inicia cuando el usuario selecciona la pestaña compartir del menú de la misma</li> <li>2. el usuario selecciona compartir foto y/o compartir evento</li> </ol>
<b>Inclusiones</b>	Ninguna
<b>Extensiones</b>	GT01(compartirTapa) y GE01(compartirEvento)

### 6.2.2 CU02 – listarTapasEventos

<b>Identificador</b>	CU02
<b>Nombre</b>	listarTapasEventos
<b>Resumen</b>	Este caso sirve para listar todas las tapas

	y todos los eventos que hay insertados en el sistema
<b>Actor</b>	Usuario anónimo
<b>Precondiciones</b>	Aplicación Android instalada, Conexión a Internet. El usuario ha entrado en la aplicación y ha seleccionado la opción de listar todas las tapas y todos los eventos desde el menú de la misma
<b>Postcondiciones</b>	El usuario ve la lista de todas las tapas y todos los eventos existentes en el sistema
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. el usuario selecciona la pestaña de ver todas las tapas y todos los eventos</li> <li>2. aparece una pantalla con la lista de todas las tapas y todos los eventos que se han introducido previamente en el sistema.</li> </ol>
<b>Inclusiones</b>	Ninguna
<b>Extensiones</b>	GT02(verTapa) y GE02(verEvento)

### 6.2.3 CU03 – listarMisTapas

<b>Identificador</b>	CU03
<b>Nombre</b>	listarMisTapas
<b>Resumen</b>	Este caso sirve para listar todas las tapas y eventos introducidos por el propio usuario del móvil
<b>Actor</b>	Usuario anónimo
<b>Precondiciones</b>	Aplicación Android instalada, Conexión a Internet. El usuario ha entrado en la aplicación y ha seleccionado la opción de listar todas mis tapas desde el menú de la misma
<b>Postcondiciones</b>	El usuario ve la lista de todas sus tapas y todos sus eventos que ha introducido previamente. El usuario ve un recuadro con el nivel de su tapa pudiendo ser TAPA ESTRELLA (nivel más alto) y TAPA CALAVERA (nivel más bajo). Si no ha alcanzado ninguno de esos niveles verá TAPA NORMALITA
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. el usuario selecciona del menú la opción de listarMisTapas</li> <li>2. aparece una lista con todas las tapas y/o eventos que ha introducido previamente el mismo</li> </ol>

	usuario. El usuario ve también en la misma pantalla el recuadro del nivel de su tapa
<b>Inclusiones</b>	Ninguna
<b>Extensiones</b>	GT02(verTapa) y GE02(verEvento)

#### 6.2.4 CU04 – visualizarMapa

<b>Identificador</b>	CU04
<b>Nombre</b>	visualizarMapa
<b>Resumen</b>	Este caso sirve para visualizar el mapa de todas las tapas y eventos que existen en el sistema y que se encuentran cerca de donde está situado el usuario
<b>Actor</b>	Usuario anónimo
<b>Precondiciones</b>	Aplicación Android instalada, Conexión a Internet. Google maps instalado en el móvil. El usuario ha entrado en la aplicación y ha seleccionado la opción de visualizar mapa desde el menú de la misma
<b>Postcondiciones</b>	El usuario ve la pantalla con un mapa donde se encuentran situados las tapas y los eventos más cercanos. Las tapas tendrán un icono diferente al de los eventos en el mapa. El usuario podrá pinchar en cada una de las tapas y eventos del mapa para verlo en grande
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. el caso se inicia cuando el usuario selecciona del menú de la aplicación la opción de visualizar el mapa</li> <li>2. aparece el mapa con todas las tapas y todos los eventos compartidos cercanos a él</li> </ol>
<b>Inclusiones</b>	Ninguna
<b>Extensiones</b>	Ninguna

#### 6.2.5 GT01 – compartirTapa

<b>Identificador</b>	GT01
----------------------	------

<b>Nombre</b>	compartirTapa
<b>Resumen</b>	Este caso sirve para compartir una tapa en el sistema
<b>Actor</b>	Usuario anónimo
<b>Precondiciones</b>	El usuario ha seleccionado la opción de compartir del menú de la aplicación. El usuario ha introducido foto ya sea hecha por su cámara o seleccionada de la galería de imágenes (ambas opciones se las da el sistema). El usuario ha introducido una breve descripción de la tapa
<b>Postcondiciones</b>	El sistema muestra un cuadro de información indicando que se está compartiendo la información y le da opción al usuario de publicar dicha información mediante correo, red social, etc.. La tapa queda introducida en el sistema
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. el usuario pulsa icono de la cámara para hacer o seleccionar foto de la tapa</li> <li>2. el usuario introduce breve descripción de la tapa</li> <li>3. el usuario pulsa el botón del icono de la copa correspondiente a la acción de compartir tapa</li> <li>4. el sistema muestra cuadro de procesando información y cuadro de publicar información</li> <li>5. el usuario elige publicar o no. Si decide publicar aparece cuadro de opciones disponibles dependiendo de las aplicaciones instaladas en el móvil</li> <li>6. la tapa queda introducida en el sistema</li> </ol>
<b>Inclusiones</b>	GT03 (hacerFoto) y GT04 (introducirComentario)
<b>Extensiones</b>	Extiende al caso de uso CU01 (compartir)

### 6.2.6 GT02 – verTapa

<b>Identificador</b>	GT02
<b>Nombre</b>	verTapa
<b>Resumen</b>	Este caso sirve para ver una tapa en particular de la lista de tapas y eventos
<b>Actor</b>	Usuario anónimo

<b>Precondiciones</b>	El usuario ha seleccionado listarTapasEventos o listarMisTapas y por lo menos hay una tapa en la lista
<b>Postcondiciones</b>	El usuario ve una pantalla con la foto de la tapa, la descripción de la misma, dos botones de votar a favor o en contra con la puntuación actual de la tapa en cada uno de ellos y un botón de publicar. En la pantalla también aparece un icono de PDI para poder visualizar la tapa en el mapa
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. el usuario pincha encima de una tapa de la lista de tapas y eventos</li> <li>2. aparece una pantalla con la foto de la tapa, la descripción de la misma y los botones de votar y publicar. También aparece el icono de PDI</li> </ol>
<b>Inclusiones</b>	Ninguna
<b>Extensiones</b>	GT05 (votar) y GT06 (publicar) Extiende a los casos de uso CU02 (listarTapasEventos) y CU03 (listarMisTapas)

### 6.2.7 GT03 – hacerFoto

<b>Identificador</b>	GT03
<b>Nombre</b>	HacerFoto
<b>Resumen</b>	Este caso sirve para que el usuario introduzca una foto de la tapa en el sistema. Esta puede ser hecha en el momento o seleccionada de la galería de imágenes de su dispositivo móvil
<b>Actor</b>	Usuario anónimo
<b>Precondiciones</b>	El usuario ha seleccionado la opción de compartir en el menú de la aplicación y ha pulsado en el icono de la cámara que le aparece en la pantalla
<b>Postcondiciones</b>	El usuario elige si hacer foto o seleccionar imagen de la galería de imágenes. Si hace la foto tiene la opción de descartarla o guardarla
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. el usuario pulsa en el icono de la cámara de la pantalla</li> <li>2. el usuario selecciona cámara o galería para hacer la foto o seleccionarla de la galería de imágenes</li> </ol>

	<ol style="list-style-type: none"> <li>3. si hace la foto, el usuario selecciona guardar o descartar</li> <li>4. la foto aparece en la pantalla de compartir</li> </ol>
<b>Inclusiones</b>	Ninguna
<b>Extensiones</b>	Ninguna

### 6.2.8 GT04 – introducirComentario

<b>Identificador</b>	GT04
<b>Nombre</b>	introducirComentario
<b>Resumen</b>	Este caso sirve para que el usuario pueda introducir una breve descripción de la tapa y así poder compartirla. Habrá un límite de caracteres que podrá introducir
<b>Actor</b>	Usuario anónimo
<b>Precondiciones</b>	El usuario ha seleccionado la opción compartir del menú de la aplicación
<b>Postcondiciones</b>	El usuario introduce un breve comentario de la tapa
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. el caso se inicia cuando, en la pantalla de compartir de la aplicación, el usuario pincha encima del recuadro de descripción de la tapa</li> <li>2. el usuario introduce por teclado, ya sea físico o virtual del dispositivo móvil, una breve descripción de la tapa hasta un límite de caracteres</li> </ol>
<b>Inclusiones</b>	Ninguna
<b>Extensiones</b>	Ninguna

### 6.2.9 GT05 – votar

<b>Identificador</b>	GT05
<b>Nombre</b>	votar
<b>Resumen</b>	Este caso sirve para que el usuario pueda votar positiva o negativamente una tapa del sistema
<b>Actor</b>	Usuario anónimo
<b>Precondiciones</b>	El usuario ha seleccionado ver tapa de la

	lista de la tapas y eventos o de la lista de sus tapas y sus eventos
<b>Postcondiciones</b>	El usuario pulsa o bien el botón de votar positivamente o bien el botón de voto negativo diferenciados por colores, verde para positivo y rojo para negativo. Cada vez que pulsa el botón la puntuación de la tapa crece o decrece en un punto.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. el usuario pulsa el botón de voto positivo o negativo de la pantalla de visualización de la tapa</li> <li>2. los puntos de la tapa crecen o decrecen un punto cada vez que pulsa el botón correspondiente</li> </ol>
<b>Inclusiones</b>	Ninguna
<b>Extensiones</b>	Extiende al caso de uso GT02 (verTapa)

### 6.2.10 GT06 – publicar

<b>Identificador</b>	GT06
<b>Nombre</b>	publicar
<b>Resumen</b>	Este caso sirve para que el usuario pueda publicar la información de la tapa a través del correo, red social, etc ... dependiendo de lo que tenga instalado en su móvil
<b>Actor</b>	Usuario anónimo
<b>Precondiciones</b>	El usuario ha seleccionado una tapa de la lista de tapas y eventos del sistema o de la lista de sus tapas y sus eventos
<b>Postcondiciones</b>	El usuario pulsa el botón de publicar de la pantalla de vota la tapa y aparece un recuadro con las opciones disponibles
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. el usuario pulsa el botón de publicar correspondiente a un botón azul</li> <li>2. aparece un cuadro de sistema con diferentes opciones</li> <li>3. el usuario selecciona una de las opciones disponibles y sigue las instrucciones</li> </ol>
<b>Inclusiones</b>	Ninguna
<b>Extensiones</b>	Extiende al caso de uso GT02 (verTapa)



### 6.2.11 GE01 – compartirEvento

<b>Identificador</b>	GE01
<b>Nombre</b>	compartirEvento
<b>Resumen</b>	Este caso sirve para compartir un evento en el sistema
<b>Actor</b>	Usuario anónimo
<b>Precondiciones</b>	El usuario ha seleccionado la opción de compartir del menú de la aplicación. El usuario ha introducido una breve descripción del evento
<b>Postcondiciones</b>	El sistema muestra un cuadro de información indicando que se está compartiendo la información y le da opción al usuario de publicar dicha información mediante correo, red social, etc... El evento queda introducido en el sistema
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. el usuario pulsa el recuadro de descripción del evento e introduce breve descripción del mismo</li> <li>2. el usuario pulsa el botón del icono del hombre correspondiente a la acción de compartir evento</li> <li>3. el sistema muestra cuadro de procesando información y cuadro de publicar información</li> <li>4. el usuario elige publicar o no. Si decide publicar aparece cuadro de opciones disponibles dependiendo de las aplicaciones instaladas en el móvil</li> <li>5. el evento queda introducido en el sistema</li> </ol>
<b>Inclusiones</b>	GE03 (introducirComentario)
<b>Extensiones</b>	Extiende al caso de uso CU01 (compartir)

### 6.2.12 GE02 – verEvento

<b>Identificador</b>	GE02
<b>Nombre</b>	verEvento
<b>Resumen</b>	Este caso sirve para ver un evento en particular de la lista de tapas y eventos
<b>Actor</b>	Usuario anónimo
<b>Precondiciones</b>	El usuario ha seleccionado listarTapasEventos o listarMisTapas y por

	lo menos hay un evento en la lista
<b>Postcondiciones</b>	El usuario ve una pantalla con la descripción del evento y un botón de publicar. En la pantalla también aparece un icono de PDI para poder visualizar el evento en el mapa
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. el usuario pincha encima de un evento de la lista de tapas y eventos</li> <li>2. aparece una pantalla con descripción del evento y el botón de publicar. También aparece el icono de PDI</li> </ol>
<b>Inclusiones</b>	Ninguna
<b>Extensiones</b>	GE04 (publicar) Extiende a los casos de uso CU02 (listarTapasEventos) y CU03 (listarMisTapas)

### 6.2.13 GE03 – introducirComentario

<b>Identificador</b>	GE03
<b>Nombre</b>	introducirComentario
<b>Resumen</b>	Este caso sirve para que el usuario pueda introducir una breve descripción del evento y así poder compartirlo en el sistema. Habrá un límite de caracteres que podrá introducir
<b>Actor</b>	Usuario anónimo
<b>Precondiciones</b>	El usuario ha seleccionado la opción compartir del menú de la aplicación
<b>Postcondiciones</b>	El usuario introduce un breve comentario del evento
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. el caso se inicia cuando, en la pantalla de compartir de la aplicación, el usuario pincha encima del recuadro de descripción del evento</li> <li>2. el usuario introduce por teclado, ya sea físico o virtual del dispositivo móvil, una breve descripción del evento hasta un límite de caracteres</li> </ol>
<b>Inclusiones</b>	Ninguna
<b>Extensiones</b>	Ninguna

### 6.2.14 GE04 – publicar

<b>Identificador</b>	GE04
<b>Nombre</b>	publicar
<b>Resumen</b>	Este caso sirve para que el usuario pueda publicar la información del evento a través del correo, red social, etc... dependiendo de lo que tenga instalado en su móvil
<b>Actor</b>	Usuario anónimo
<b>Precondiciones</b>	El usuario ha seleccionado un evento de la lista de tapas y eventos del sistema o de la lista de sus tapas y sus eventos
<b>Postcondiciones</b>	El usuario pulsa el botón de publicar de la pantalla de visualización del evento y aparece un recuadro con las opciones disponibles
<b>Flujo normal</b>	<ol style="list-style-type: none"><li>1. el usuario pulsa el botón de publicar correspondiente a un botón azul</li><li>2. aparece un cuadro de sistema con diferentes opciones</li><li>3. el usuario selecciona una de las opciones disponibles y sigue las instrucciones</li></ol>
<b>Inclusiones</b>	Ninguna
<b>Extensiones</b>	Extiende al caso de uso GE02 (verEvento)

## 7 Diseño

Este apartado incluye el diseño de toda la aplicación que se va a ir completando con cada iteración del proyecto.

### 7.1 Consideraciones previas

#### 7.1.1 Visualización en distintos dispositivos

Android se ejecuta en una variedad de dispositivos que ofrecen diferentes tamaños de pantalla y densidades. Para las aplicaciones, el sistema Android ofrece un entorno de desarrollo coherente entre dispositivos y se ocupa de la mayor parte del trabajo para ajustar la interfaz de usuario de cada aplicación a la pantalla en la que se muestra. Al mismo tiempo, el sistema proporciona una API que le permite controlar la interfaz de usuario de la aplicación para los tamaños de pantalla específicos y densidades, con el fin de optimizar el diseño de su interfaz de usuario para diferentes configuraciones de pantalla.

A partir de Android 1.6 (API Nivel 4), Android ofrece soporte para múltiples tamaños de pantalla y densidades, lo que refleja las múltiples configuraciones de pantalla

diferentes que un dispositivo pueda tener. Se pueden utilizar las funciones del sistema Android para optimizar la interfaz de usuario de la aplicación para cada configuración de pantalla y así, asegurarse, de que la aplicación lo hace correctamente.<sup>5</sup>

## Términos y conceptos

- **Tamaño de la pantalla:** tamaño físico real, medido en la diagonal de la pantalla. Por simplificar Android divide los tamaños en cuatro grupos: pequeño, normal, grande, extra grande.
- **Densidad de la pantalla:** cantidad de píxeles en un área física de la pantalla (dpi, dots per inch). Por simplificar Android divide las densidades en cuatro grupos: pequeña, normal, grande, extra grande (ldpi, mdpi, hdpi, xhdpi).
- **Píxel independiente de la densidad (dp):** es una unidad virtual que debe usarse para definir el diseño visual de la aplicación, para definir las dimensiones y la posición de una forma independiente de la densidad de la pantalla.

Para optimizar la aplicación para diferentes tamaños y densidades de pantalla se pueden proporcionar distintos recursos para los distintos grupos generalizados. En tiempo de ejecución el sistema utilizará los recursos apropiados para la aplicación entre los disponibles según las características de la pantalla del dispositivo donde se está ejecutando.

No es necesario proporcionar recursos alternativos para cada combinación de tamaño de la pantalla y la densidad. El sistema ofrece funciones de compatibilidad robustos que pueden manejar la mayor parte de la labor de la presentación de su aplicación en cualquier pantalla del dispositivo, siempre y cuando se haya implementado la interfaz de usuario utilizando técnicas que le permiten cambiar el tamaño de gracia como la Independencia de la densidad de la pantalla.

## Independencia de la densidad de la pantalla

Una aplicación puede considerarse independiente de la densidad cuando mantiene el tamaño de los elementos que forman cada pantalla, desde el punto de vista del usuario, al ejecutarse en dispositivos con distintas densidades.

El sistema escala las unidades dp y usa los recursos adecuados entre los disponibles para conseguir la independencia de la densidad de pantalla.

En la mayoría de los casos, se puede conseguir simplemente especificando todas las dimensiones con valores en unidades dp, o utilizando "wrap\_content" (tamaño necesario para mostrar el contenido que se incluye en la vista). El sistema escala los recursos y los elementos para mostrarlos en el tamaño adecuado, utilizando el factor necesario según la densidad de pantalla del dispositivo donde se está ejecutando.

Para que las imágenes no pierdan calidad al escalarse conviene proporcionar imágenes con distintas resoluciones para las distintas densidades de pantalla. Estas imágenes deben guardar esta proporción para ser más óptimo: 3:4:6:8. Por ejemplo para la imagen del icono de la aplicación podríamos tener cuatro imágenes con estos tamaños en píxeles:

- 36x36 para densidad pequeña
- 48x48 para densidad normal
- 72x72 para densidad grande
- 96x96 para densidad extra grande

## **Servicios de imágenes de Topoos**

La plataforma Topoos dispone de servicios de alojamiento de imágenes PIC, el cuál realizará automáticamente todas las transformaciones necesarias para poder mostrar las imágenes en todo tipo de dispositivos de manera apropiada y eficiente.

### **Buenas Prácticas**

Para asegurarse de que la aplicación se va a ver bien en dispositivos con pantallas de distintos tamaños se deben seguir estos consejos:

- Especificar las dimensiones de los elementos de las vistas de forma independiente de la densidad de la pantalla como utilizando unidades 'dp', o especificaciones como 'wrap\_content' o 'fill\_parent'. Para definir tamaños de texto se puede utilizar la unidad 'sp' (scale-independent pixel), donde el factor de escala depende de las especificaciones del usuario.
- No utilizar valores en píxeles para especificar esas medidas.
- No usar AbsoluteLayout sino RelativeLayout, que usa un posicionamiento relativo para posicionar las vistas hijas en la vista madre. Se podría por ejemplo especificar que un botón apareciera 'a la derecha' de un texto, siendo así su posición independiente de la pantalla donde se visualice.
- Proporcionar imágenes alternativas de distintos tamaños.

A la hora de desarrollar aplicaciones se pueden crear dispositivos virtuales con pantallas de distintos tamaños y densidades, para así estar seguros de que el diseño se va a ver como esperamos en los distintos dispositivos.

## **7.2 Diseño de la Arquitectura del Sistema**

Para comenzar se presentará un diagrama de componentes del sistema de la aplicación a alto nivel para tener una imagen global del proyecto que se va a realizar



La figura anterior muestra un diagrama de los componentes de la aplicación:

- **Terminal Android:** es el dispositivo móvil donde instalaremos la aplicación "Ranking Tapas". Debe ser un Terminal con acceso a datos de red
- **SDK Topoos:** la aplicación usa el SDK de Topoos que proporcionará los servicios necesarios de almacenamiento en base de datos de entidades y operaciones haciendo uso de SQLite, tracking de usuarios para consultar la ubicación de la tapas y eventos en tiempo real, almacenamiento y consulta de puntos de interés (PDIs) mediante procesos automáticos de enriquecimiento de información geográfica del PDI durante el registro y herramientas de representación y visualización de PDIs en mapas interactivos, servicio de redes sociales, servicios de notificaciones cuando se producen ciertos eventos, alojamiento de imágenes PIC ya que topoos realiza automáticamente todas las transformaciones necesarias para mostrar las imágenes en todo tipo de dispositivos de manera apropiada y eficiente.  
Será necesario registrarse en Topoos y adquirir unas claves para la implementación de la aplicación
- **Servicios Google API:** la aplicación usa el servicio Google Maps Android v2. Serán necesarios unos certificados para usar este servicio.
- **SDK AdMob:** plataforma publicitaria diseñada por Google para proporcionar obtención de ingresos mediante publicidad en la aplicación a través de la inserción y configuración de banners en la misma.  
Será necesario registrarse en AdMob para poder hacer uso de dicha plataforma.

## 7.3 Diseño de la Interfaz de usuario y Navegación de la aplicación

En esta sección se abordan las diferentes interfaces propuestas para la aplicación y su funcionamiento en cuanto a navegación, de tal manera, que sirva como pequeño manual de usuario de la aplicación.

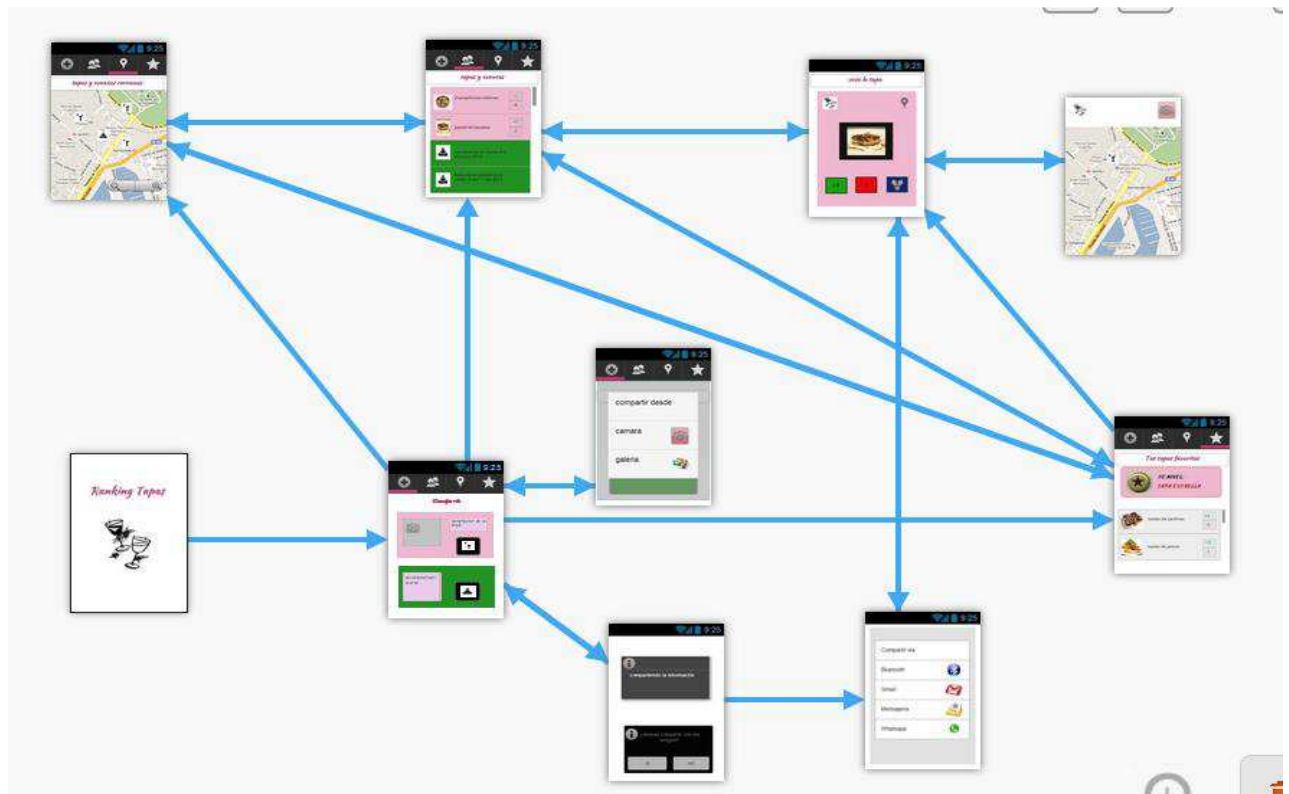
Para ello se ha realizado el diseño de las mismas mediante la herramienta online FluidUI y se puede ver el mismo de forma interactiva en el siguiente enlace

[https://www.fluidui.com/editor/live/preview/p\\_GKORNMQIFserH3PpQ531oTb9z03TlvQh.1382958735112](https://www.fluidui.com/editor/live/preview/p_GKORNMQIFserH3PpQ531oTb9z03TlvQh.1382958735112)

**Nota:** la versión gratuita solamente proporciona la realización de un solo proyecto con 10 pantallas como máximo. He realizado el diseño de las interfaces y su navegación en este número máximo de pantallas que se podrá ver en el enlace anterior. Como es necesaria alguna pantalla más, realizaré las pertinentes de modo particular para insertarlas en el apartado correspondiente del diseño pero no aparecerán en el diseño interactivo del anterior enlace

### 7.3.1 Esquema general

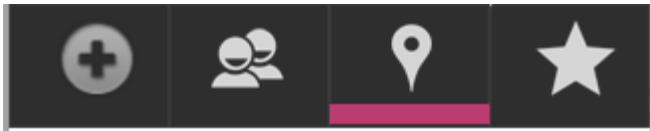
En la siguiente figura aparece el diseño general de las pantallas de la aplicación "Ranking Tapas"



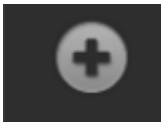
## 7.3.2 Elementos de Navegación

### MENU DE LA APLICACIÓN

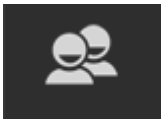
La aplicación dispone del siguiente menú de navegación que aparecerá en todas las pantallas principales de la misma



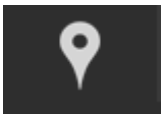
Las diferentes opciones del mismo se activarán mediante la pulsación



→ compartir



→ listar todas las tapas y eventos



→ visualizar mapa



→ listar tapas y eventos del usuario del móvil

Para saber cuál está activo en cada momento, aparecerá en el mismo una barra de color fucsia



### BOTONES e ICONOS



→ icono logo de la aplicación



→ opción de volver a la pantalla de una tapa en particular

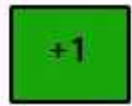


→ publicar



→ visualizar localización de la tapa o el evento





→ vota positivamente. El número que aparece es la puntuación actual positiva de la tapa



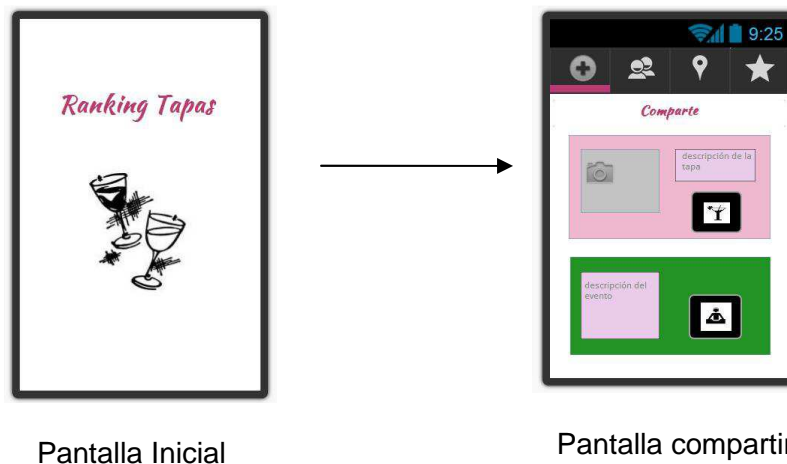
→ vota negativamente. El número que aparece es la puntuación actual negativa de la tapa

### 7.3.3 Pantalla inicial

La aplicación comienza con la siguiente pantalla donde se muestra el nombre de la misma y su logo.

Esta pantalla se visualizará por un espacio muy corto de tiempo y pasará automáticamente a la siguiente pantalla.

En la figura se muestra la pantalla y su flujo de navegación a la siguiente.



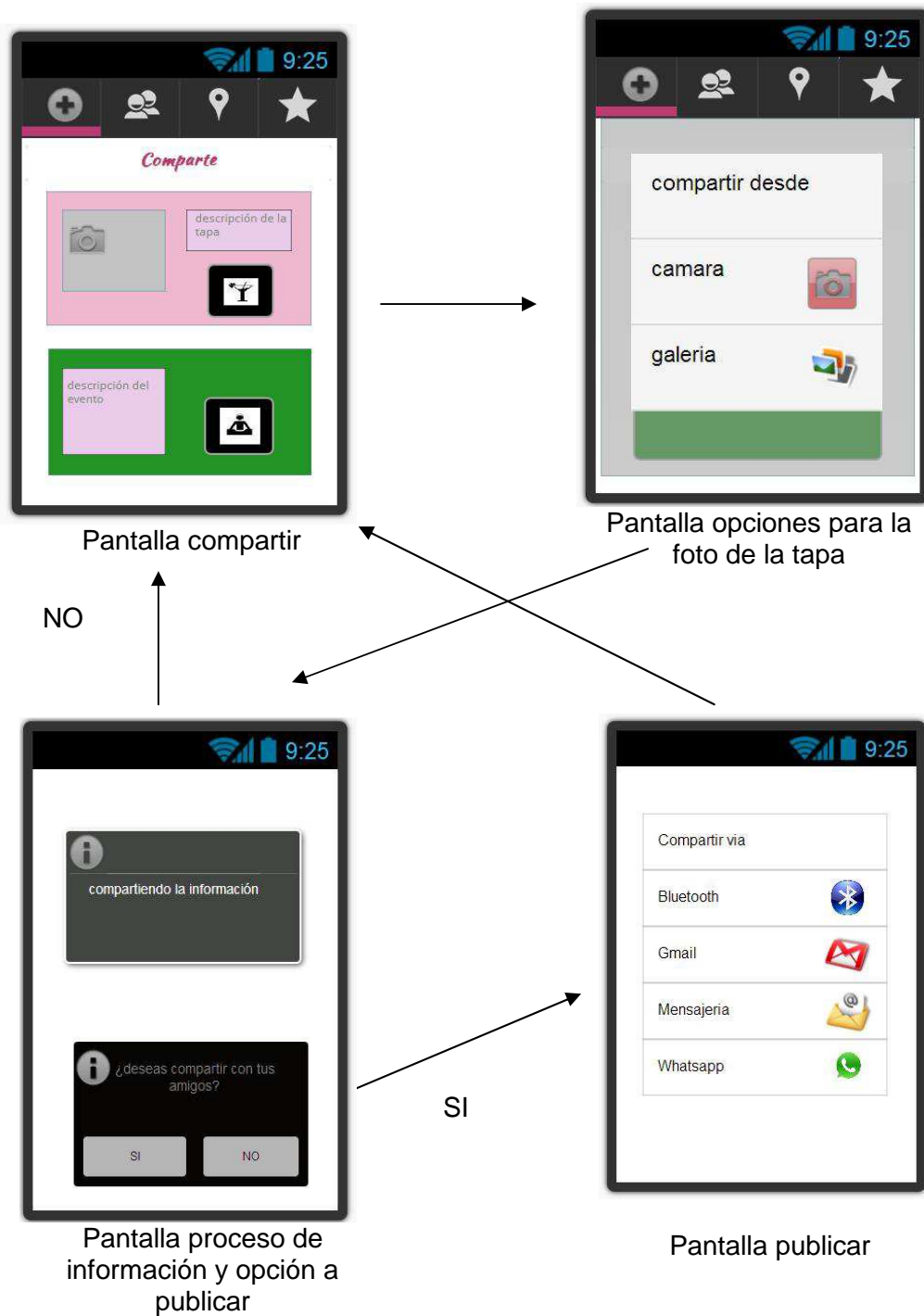
### 7.3.4 Pantalla compartir tapa y/o evento

En esta pantalla, aparte del menú de la aplicación, aparece la siguiente información que debe proporcionar el usuario

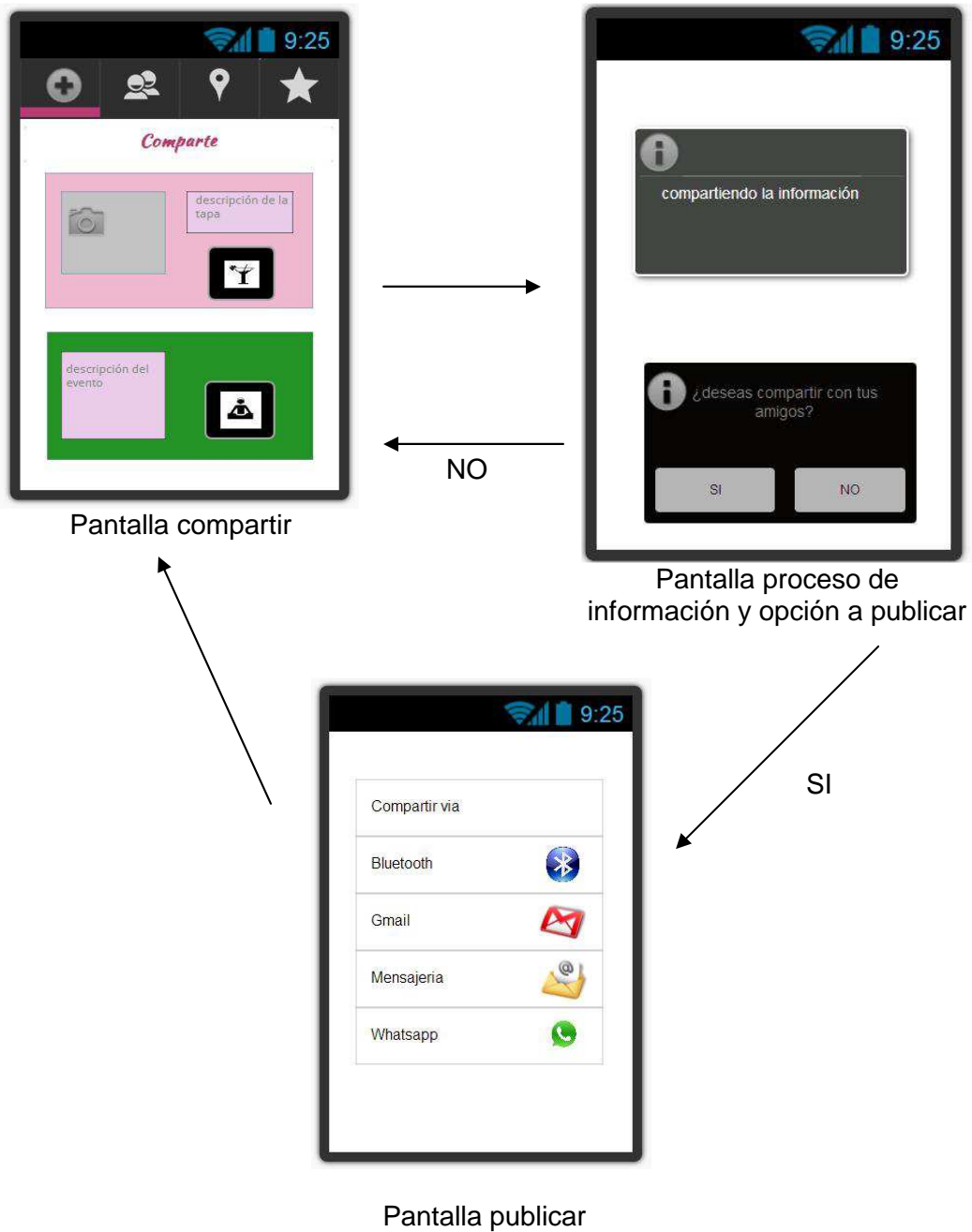
- en el caso de la tapa:
  - fotografía: pulsando en el icono de la cámara
  - descripción: pulsando en el recuadro de texto
- en el caso del evento:
  - descripción: pulsando en el recuadro de texto correspondiente

En la figura se muestra dicha pantalla y el flujo de navegación dependiendo de una opción u otra

## FLUJO PARA COMPARTIR TAPA



## FLUJO PARA COMPARTIR EVENTO



### 7.3.5 Pantalla listar todas las tapas y eventos del sistema

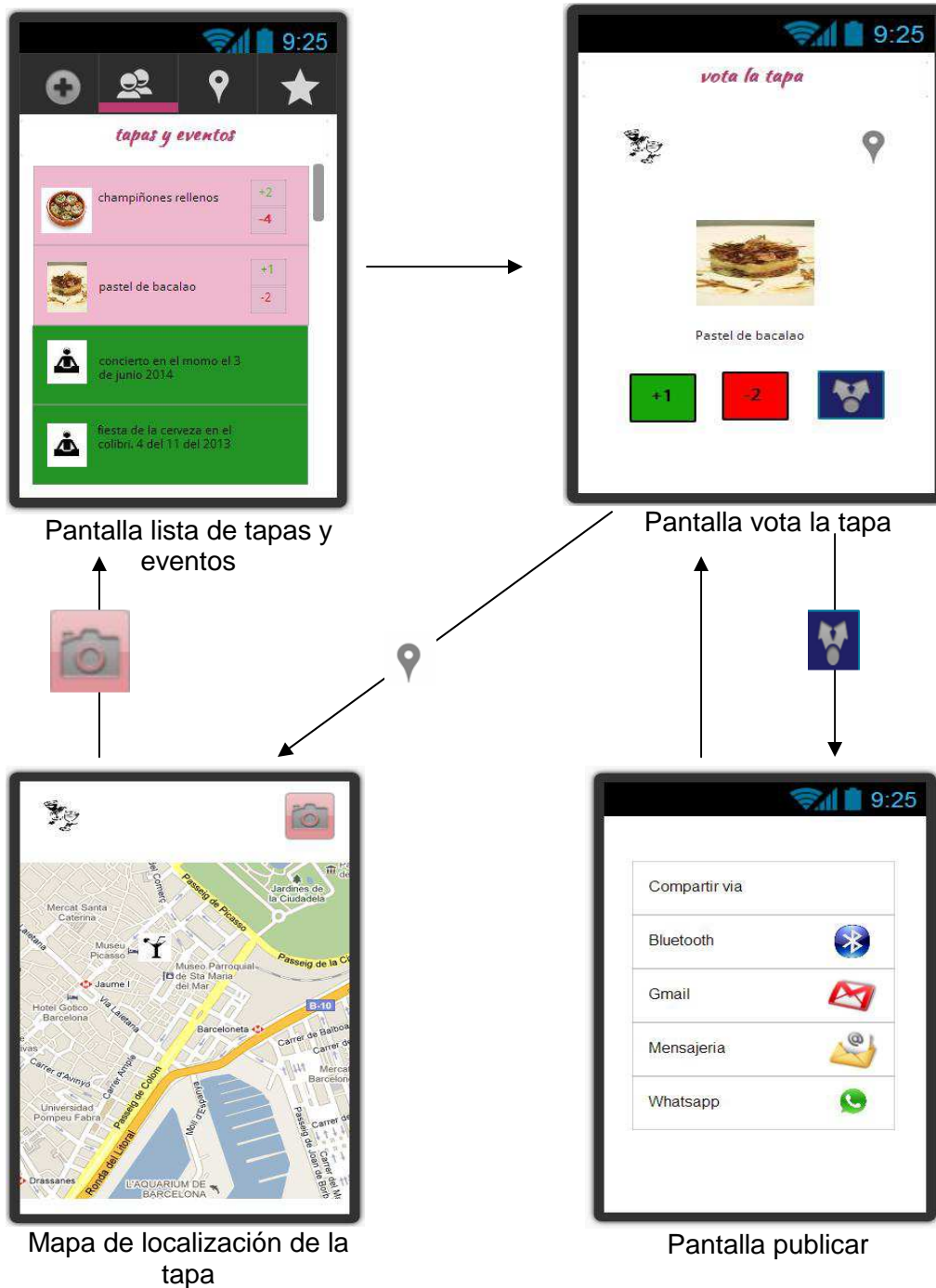
En la siguiente pantalla, que el usuario seleccionará pulsando el icono correspondiente a la opción de listar todas las tapas y eventos, aparece la lista con todas las tapas y todos los eventos que hay agregados en el sistema en ese momento.

La lista diferencia tapas y eventos según la imagen de cada elemento. Para las tapas, la imagen será la fotografía introducida por el usuario. Para el evento la imagen se corresponderá a un icono diseñado para tal acción.

Se podrá seleccionar cada elemento de la lista, ya sea tapa o evento, pulsando encima de él.

En la figura se muestra dicha pantalla y el flujo de navegación dependiendo de una opción u otra.

### FLUJO PARA VER UNA TAPA DE LA LISTA



### FLUJO PARA VER UN EVENTO DE LA LISTA

Este es uno de los flujos que no aparecen en el enlace proporcionado anteriormente. La pantalla es similar a la de vota la tapa pero sin los botones de votar disponiendo, solamente, el botón de publicar.

No aparece, por tanto, la fotografía del evento sino el icono correspondiente a un evento y su descripción.

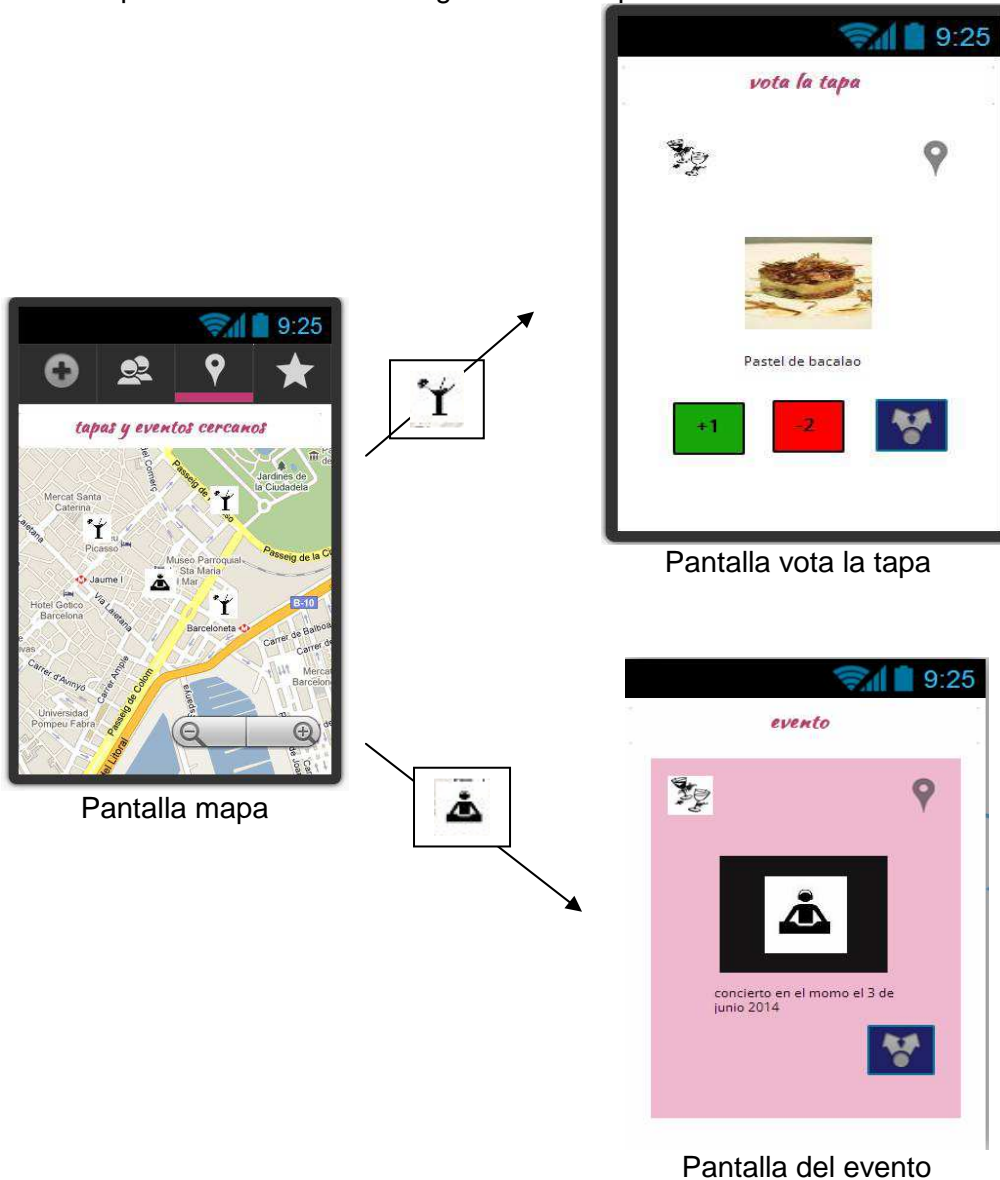
La opción de visualizar en el mapa sí está disponible  
Sería algo así



### 7.3.6 Pantalla visualizar Mapa

En la siguiente pantalla, que el usuario seleccionará pulsando el icono correspondiente a la opción de visualizar mapa, aparece el mapa en cuestión con los iconos correspondientes a la localización de las tapas y de los eventos introducidos en el sistema.

Pulsando en cada uno de ellos podrá ver la visualización en particular de la tapa y/o del evento (mirar pantallas anteriores de vota la tapa y evento)  
También aparece el menú de navegación de la aplicación.



### 7.3.7 Pantalla listar sólo tapas y eventos del usuario del móvil

En la siguiente pantalla, que el usuario seleccionará pulsando el icono correspondiente a la opción de listarMisTapas o sus tapas favoritas, aparece la lista con todas las tapas y eventos compartidos por él mismo, junto con el menú de navegación y un recuadro

indicando el nivel de tapa alcanzado: TAPA ESTRELLA, TAPA NORMALITA O TAPA CALAVERA

Podrá seleccionar cada tapa y/o evento en particular apareciendo posteriormente la pantalla de vota la tapa o del evento correspondiente



Tapas favoritas nivel  
TAPA ESTRELLA



Tapas favoritas nivel  
TAPA CALAVERA



Tapas favoritas nivel  
TAPA NORMALITA

**NOTA:** Solamente aparece en el flujo del enlace interactivo la pantalla correspondiente al nivel TAPA ESTRELLA. Las otras dos se verán en el funcionamiento del programa

## 8 Implementación del Sistema

En este capítulo se va a tratar la implementación de la aplicación RankingTapas. Debido a diversos contratiempos con esta tecnología, se ha tenido que suprimir el apartado de publicidad del código, quedando postergado a una futura implementación. También se ha agregado la nueva funcionalidad de votación para los eventos compartidos por los usuarios, pues se ha considerado inteligente en cuanto a implementación y funcionamiento, además se ha aprovechado código ya implementado para dicho funcionamiento

A su vez se ha incluido la plataforma de Google Analytics para el seguimiento de ciertos aspectos de la aplicación.

Aparte se ha rediseñado la visualización de algunas pantallas, como las de los mapas, debido al funcionamiento con la API de Google maps v2 y considerado, de manera personal, que el resultado es más óptimo visualmente que en el documentado del apartado de diseño. A pesar de estos cambios, no se ha perdido la funcionalidad inicial y propuesta para la aplicación.

Se adjunta con este documento el código del proyecto RankingTapas, el código de Google-play-services\_lib y el fichero de instalación apk. Todo ello incluido en un fichero con formato rar.

Se deberá incluir como librería del proyecto el paquete de google-play-services\_lib que se adjunta.

## 8.1 Entorno de desarrollo

En el apartado de Medios empleados de este documento se habló de las herramientas que han sido necesarias para realizar dicha implementación.

Los requisitos necesarios del sistema para la instalación del entorno son:

- Windows XP (32-bit), Vista (32- or 64-bit), or Windows 7 (32- or 64-bit)
- Eclipse 3.6.2 (Helios) o superior<sup>7</sup>
- Eclipse JDT plugin<sup>8</sup> (incluido en la mayoría de los paquetes Eclipse IDE)
- JDK 6 (JRE solo no es suficiente). En particular he usado la versión 1.6.0.14
- Android Development Tools plugin<sup>9</sup> (recomendado y usado en el proyecto)

La instalación de los mismos no queda explicada en este documento debido a que la persona que esté relacionada con este entorno será capaz de realizar la óptima configuración del mismo con los detalles proporcionados.

No obstante se detallan a continuación las configuraciones necesarias para la inclusión de la API de Google Maps V2, la inclusión de la plataforma Topoos y la inclusión de la plataforma Analytics dentro de la aplicación.

No es necesario realizar estos pasos para el funcionamiento correcto del código pues la configuración ya está incluida en la implementación del código y en el ejecutable.

Aún así, es aconsejable usar datos propios tales como ADDMIN TOKEN y CLAVE de API de Google Maps V2 generadas por el propio usuario y la inclusión de las mismas en el código. Para ello habrá que adaptar el fichero AndroidManifest.xml y la clase AccessInterface.

## 8.2 Google Maps V2

Desde diciembre de 2012 la API de Google Maps v1 queda oficialmente obsoleta. Por ello este proyecto Android trabaja con la API de Google Maps v2.

Para que funcione correctamente en los móviles, estos tendrán que tener actualizada esta versión y tener instalado tanto Google Play Services como Google Play Store.

Este requisito queda actualizado en dicho documento en el apartado correspondiente de Visualización y funcionamiento en distintos dispositivos.



Para la realización del código, también ha sido necesario configurar e instalar estas aplicaciones en un emulador. Ha sido un arduo trabajo de varias horas y días hasta que se ha conseguido el objetivo. Para más información ver apartado de pruebas y emulador.

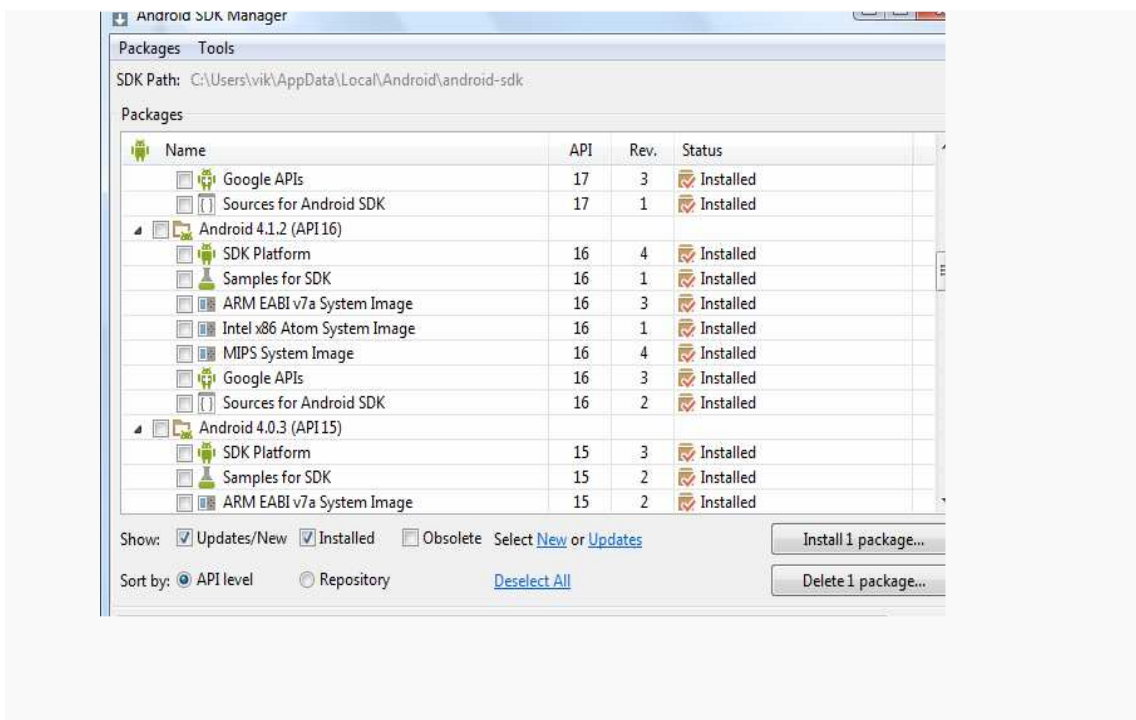
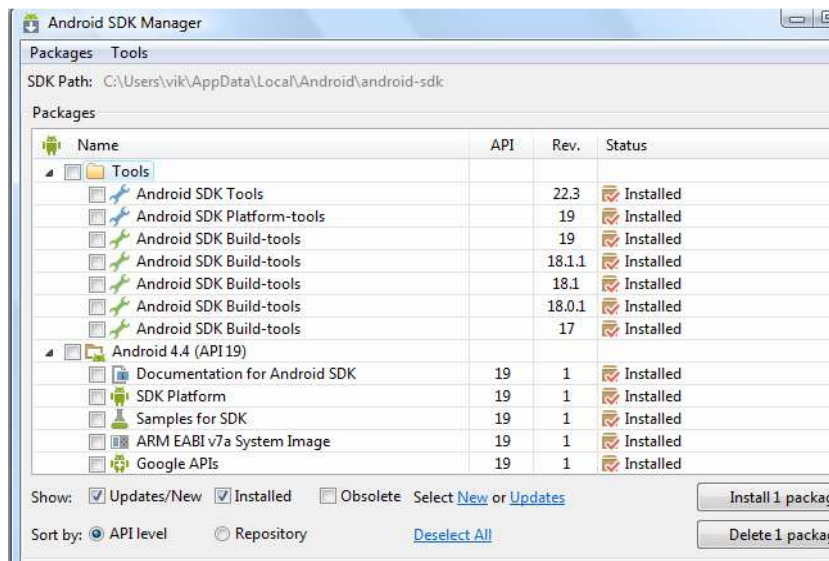
El proceso general para incluir un mapa en nuestra aplicación Android es la siguiente:



## 8.2.1 Descargar y configurar el SDK de Google Play servicios

Antes con la API v1, la librería que usábamos para incluir los mapas era una librería externa, ahora con esta API v2 se incluye como parte de Google Play services SDK. Para ello es necesario descargarla desde Android SDK Manager e instalar Google Play services SDK y Android SDK

Desde el eclipse habrá que acceder al menú de SDK e instalar los paquetes necesarios. En la aplicación se usa Android 4.1.2 (API 16)



## 8.2.2 Obtener una clave de Android para la API de Google Maps

La clave de la API de Google Maps v1 ya no funcionara con la API de Google Maps v2. Ahora google tiene otro sistema de asignar claves, y lo hace mediante Google APIs Console, en esta consola tendremos que facilitarle 2 cosas para que nos de una clave, por un lado La firma digital con la que firmamos nuestras aplicaciones, y por otro el nombre del paquete de la aplicación en la que usaremos la clave.

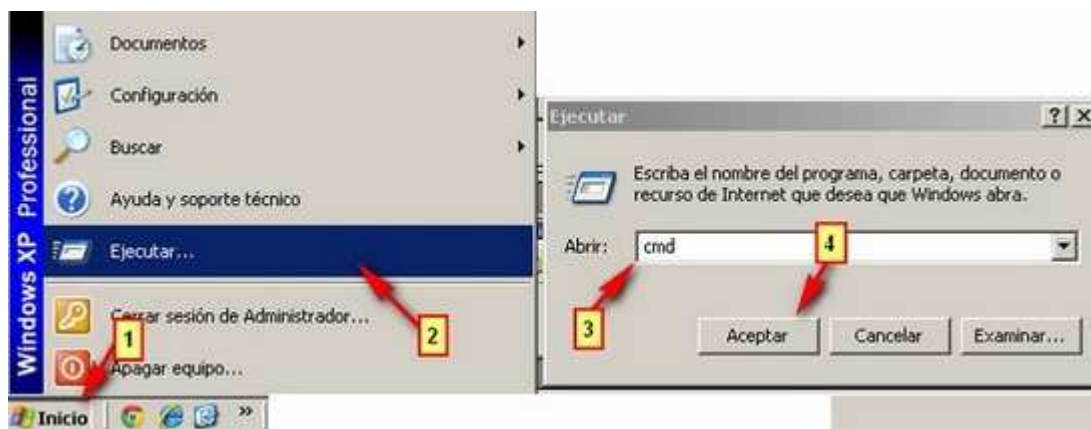
Estas claves para la API están relacionadas específicamente con un certificado/paquete. Realmente sólo necesitamos una clave para cada certificado. Siempre que firmes tú las aplicaciones con el mismo certificado, te valdrá esta clave API. Sin embargo, Google recomienda firmar cada una de sus aplicaciones con un certificado diferente y obtener una clave diferente para cada uno. En este proyecto se han usado dos claves, una para probar la aplicación en el emulador y otra para el funcionamiento de la misma en un dispositivo real. La segunda se ha generado con la firma proporcionada a la hora de generar un keystore para realizar el ejecutable .apk de la aplicación. Es importante saber dónde se encuentran los certificados para obtener las firmas y posteriormente las claves API- La forma de obtenerlas es la misma y los pasos para conseguir una API key de Google Maps v2 se observa en los siguientes puntos

**Nota:** Es muy importante no equivocarse de API key con el certificado que se use al firmar la aplicación (desarrollo o para publicar), pues esto provocaría que el mapa no cargara correctamente en la aplicación.

### **RECUPERAR INFORMACIÓN SOBRE EL CERTIFICADO SHA-1 CON EL QUE SE FIRMA LA APLICACIÓN**

En este caso hacemos uso del certificado de depuración. Si queremos obtenerla a partir de la firma de la aplicación, a la hora de generar el paquete de instalación lo haremos con modalidad de firma y para ello crearemos una KEY nueva y un alias nuevo y obtendremos la huella digital SHA-1 a partir de las mismas

Se ejecuta lo siguiente desde el CMD



```
keytool -list -v -keystore "C:\Documents and Settings\Administrador\.android\debug.keystore"  
-alias androiddebugkey -storepass android -keypass android
```

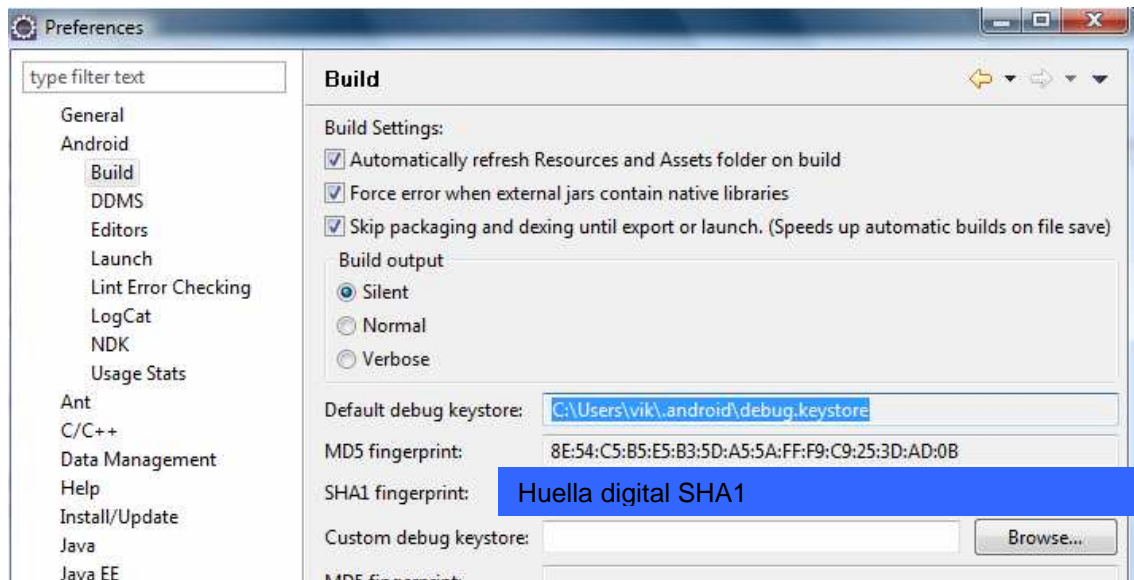


```
C:\WINDOWS\system32\cmd.exe

C:\>keytool -list -v -keystore "C:\Documents and Settings\Administrador\.android\debug.keystore" -alias androiddebugkey -storepass android -keypass android
Nombre de alias: androiddebugkey
Fecha de creaci3n: 22-dic-2011
Tipo de entrada: PrivateKeyEntry
Longitud de la cadena de certificado: 1
Certificado[1]:
Propietario: CN=Android Debug, O=Android, C=US
Emisor: CN=Android Debug, O=Android, C=US
N.nero de serie: 4ef26b42
Vlido desde: Thu Dec 22 00:26:58 CET 2011 hasta: Sat Dec 14 00:26:58 CET 2041
Huellas digitales del certificado:
MD5:
SHA1: Huella digital SHA1
Nombre del algoritmo de firma: SHA1withRSA
Versi3n: 3

C:\>_
```

Tambi3n se puede obtener esta informaci3n desde el men3 de Eclipse, Preferences→Android→Build

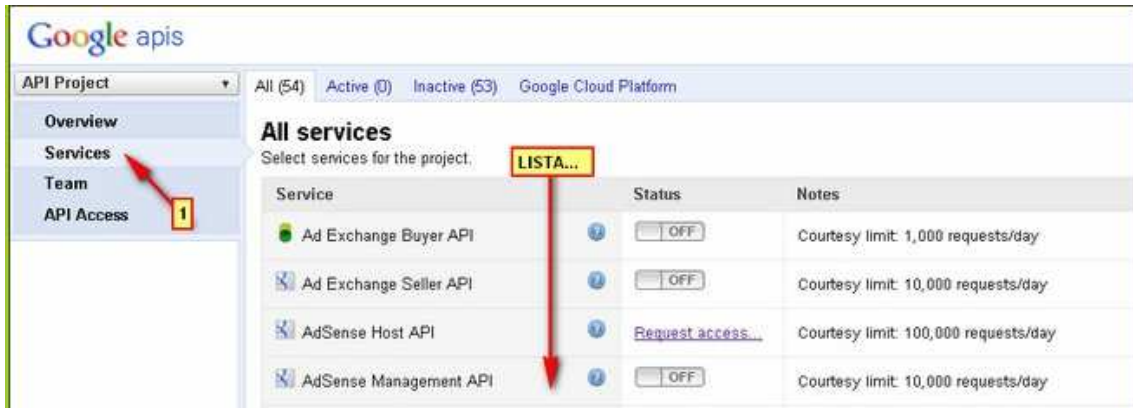


Copiamos el valor de la huella digital SHA-1 y accedemos a la consola de Google API, <https://code.google.com/apis/console/>

### **ACCESO A CONSOLA GOOGLE API Y CREACI3N DE NUEVA CLAVE DE ANDROID**

Debemos tener una cuenta de google para acceder a esta web

Una vez dentro, en la parte de la izquierda tenemos un men3, selecciona Services, y en el listado busca Google Maps Android API v2.



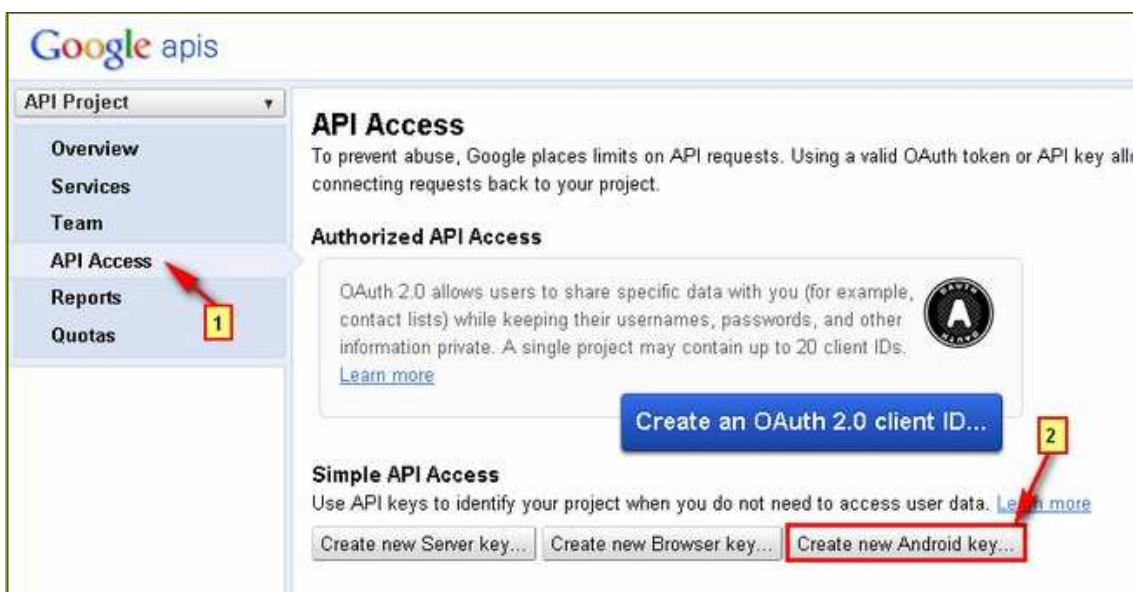
Debemos activar la de Android, para ello habrá que pinchar en el switch para ponerlo en ON.



Quedando así



Seleccionamos API Access en el menú de la parte Izquierda y pulsamos la opción de Create new Android Key



Y aquí tenemos que pegar el código SHA1 seguido de; y el nombre del paquete que queremos poner, en este caso es es.example.rankingtapas



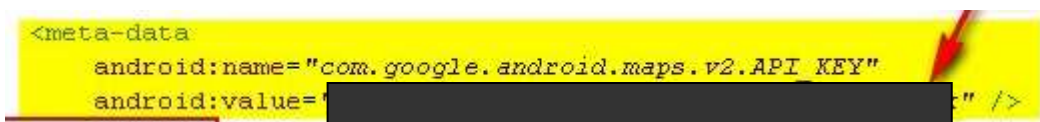
Al darle a crear, saldrá la clave que es la que incluiremos en el AndroidManifest del proyecto



### 8.2.3 Especificar permisos y ajustes en el Manifest de la aplicación

En el fichero AndroidManifest.xml habrá que incluir la siguiente configuración

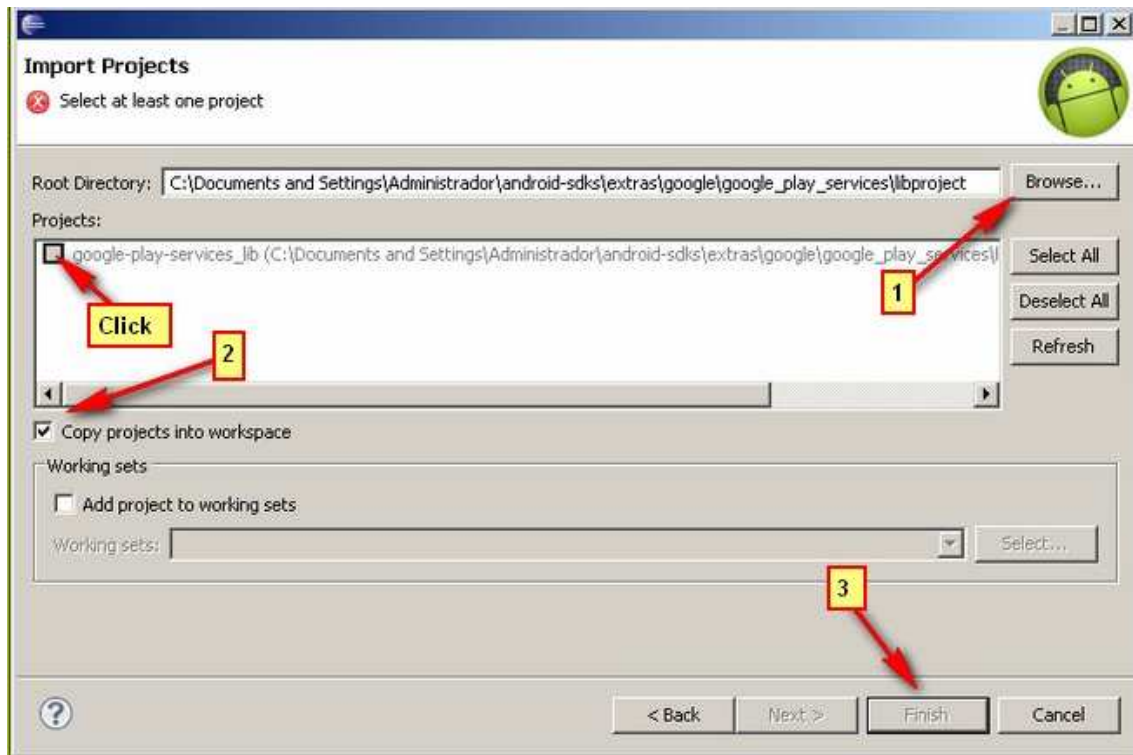
- Añadir API KEY obtenida de la consola de Google como meta-data



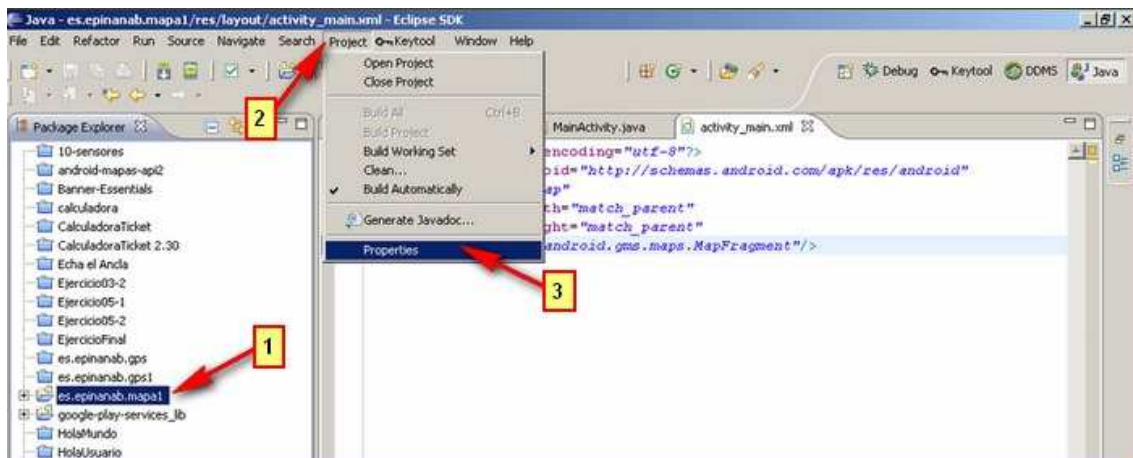
- Asignar Permisos necesarios para el API de Google Maps V2
  - **android.permission.INTERNET**. Este permiso da acceso a internet, y es usado por el API para descargar el mapa desde los servidores de Google Maps.
  - **com.google.android.providers.gsf.permission.READ\_GSERVICES**. Permite el acceso al API hacia los servicios web de Google.
  - **android.permission.WRITE\_EXTERNAL\_STORAGE**. Permite que el API guarde una caché en la memoria externa del teléfono
  - **android.permission.ACCESS\_COARSE\_LOCATION**. Permite al API usar la WiFi o Datos del teléfono para determinar tu localización.
  - **android.permission.ACCESS\_FINE\_LOCATION**. Permite al API usar el sistema de posicionamiento Global (GPS) para determinar la ubicación del dispositivo de una forma mas precisa.
- Requerir OpenGL ES versión 2. Esto notifica a los servicios exteriores la exigencia de requerir OpenGL. En concreto y traduciendo en nuestro idioma, es hacer que GooglePlay sepa que nuestra aplicación necesita OpenGL, impidiendo que nuestra aplicación sea mostrada a dispositivos (Móviles o Tabletas) que no soporten esta función para evitar errores en el tiempo de ejecución.

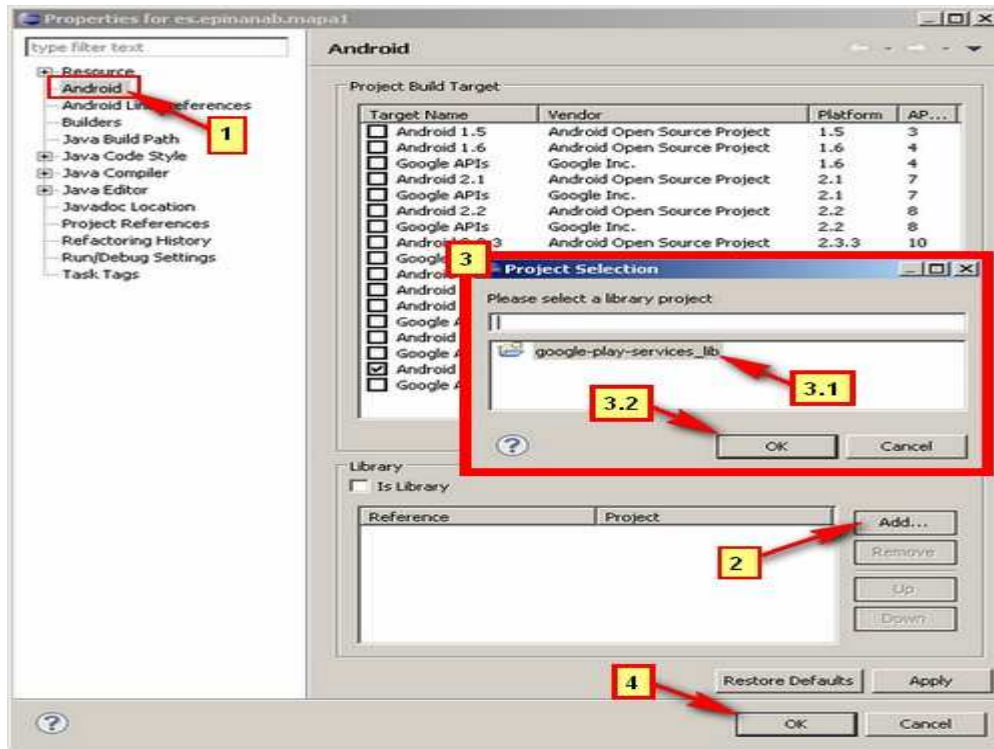
```
<uses-feature  
    android:glEsVersion="0x00020000"  
    android:required="true" />
```

- Añadir la librería de Google Play Services al proyecto. Esto sólo hay que hacerlo una vez. Al principio se añadió la librería que viene con el sdk de Android, *android-sdk/extras/google/google\_play\_services/libproject/google-play-services\_lib*, pero posteriormente y para pruebas en emulador, tuve que añadir otra librería más reciente para la correcta visualización de los mapas en el mismo (ver apartado de pruebas y emulador).  
Para añadir la librería en cuestión habrá que hacer lo siguiente
  - Importar la librería como proyecto desde Eclipse seleccionando en el menú la opción de File→import→Existing Android Code Into Workspace. Copiar en el workspace y finalizar



- Añadir esta librería al proyecto Android de la aplicación





**Nota:** ver apartado de AndroidManifest para obtener el resultado final del mismo

### 8.2.4 Añadir el control del mapa de un proyecto Android nuevo o existente

A la hora de implementar los mapas necesarios en la aplicación, estos se añaden como SupportMapFragment en los layouts correspondientes y se accede a ellos a través del código java necesario para ello (en el código proporcionado de la aplicación de puede observar la implementación de los mapas)

```
<fragment  
    android:id="@+id/mapview"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    class="com.google.android.gms.maps.SupportMapFragment"/>
```

```
SupportMapFragment fm = (SupportMapFragment)  
getSupportFragmentManager().findFragmentById(R.id.mapview);
```

## 8.3 Configuración Topoos

El proyecto hace uso de la plataforma proporcionada por Topoos y para ello es necesario añadir cierta configuración de acceso a la misma



### 8.3.1 Registro como desarrollador en la plataforma de Topoos

Para ello, a través del siguiente enlace, <https://login.topoos.com/?oauth=True&authreq=49e2a4c3-e30e-4ad6-8ba6-6a60e394093c>, creamos una cuenta de acceso a la plataforma

### 8.3.2 Crear nueva aplicación en Topoos

Entra en el panel de desarrollador y haga clic en el botón "Mis aplicaciones" y luego en "Crear nueva aplicación"

La plataforma le redirige a una página donde habrá seleccionar el tipo de aplicación que desea crear, en este caso es "aplicaciones móviles"

Acepte los términos de uso y la política de privacidad y haga clic en el botón "Guardar". La plataforma proporciona los tokens necesarios que la aplicación debe utilizar para conseguir la autorización al realizar llamadas a la API (ID CLIENTE, SECRETO DEL CLIENTE, APPTOKEN ADMIN, APPTOKEN USER).



El token que se usa en la aplicación es el APPTOKEN ADMIN.  
La clase del código que se encarga del acceso a Topoos, es la clase AccessInterface.  
Consultar apartado Implementación

### 8.3.3 Instalación de Topoos SDK

Incluir en la librería de la aplicación el SDK de Topoos. Se puede descargar dicha librería desde el siguiente enlace <http://docs.topoos.com/tools/sdks/android/download-and-changes-log/>

## 8.4 Estructura de la aplicación y manifiesto

En la siguiente tabla se muestran los principales componentes de la aplicación

Directorio	Fichero	Propósito
/	<b>AndroidManifest.xml</b>	Información de la aplicación que será mostrada al sistema Android, necesitamos acceso a Internet y permisos para el uso de la cámara: <a href="#">android.permission.WRITE_EXTERNAL_STORAGE</a> <a href="#">android.permission.INTERNET</a> <a href="#">android.permission.ACCESS_NETWORK_STATE</a> También necesitamos permisos para poder acceder a la plataforma de Google Maps V2. Ver Apartado de Google Maps V2 y apartado de AndroidManifest
/res/layout		Contiene la definición del interfaz gráfico
/res/values	<b>Strings.xml</b>	Contiene los valores de las cadenas que serán utilizados en la aplicación
	<b>Color.xml</b>	Contiene los valores de los colores que serán utilizados en la aplicación
	<b>Styles.xml</b>	Contiene los valores de los estilos que serán utilizados en la aplicación
	<b>Analytics.xml</b>	Contiene la configuración de los parámetros para la implementación de la plataforma Analytics en la aplicación
/res/drawable		Contiene todas las imágenes a utilizar por el interfaz gráfico
/res/menu		Contiene la configuración del menú. Tendrá dos opciones: detalles y salir de la aplicación. Se activa con la pulsación del botón menú del teléfono
/res/anim		Contiene diferentes elementos de animación para los layouts que lo requieren
/gen	<b>R.java</b>	Contiene los identificadores de los elementos contenidos /res. Es un fichero que no debe ser modificado por el usuario, se modifica automáticamente en función de los cambios en los ficheros de arriba.
/libs		Contiene las librerías necesarias para el proyecto

El código java se divide en cinco paquetes

- **es.example.rankingtapas**: contiene las actividades principales de la aplicación
- **es.example.rankingtapas.internal**: contiene diversas clases que se ocupan de la creación de la base de datos, tablas, consultas y clases de utilidades para la aplicación. Además la clase Tapento está incluida en este directorio y es la clase que contiene los datos propios de un elemento, ya sea tapa o evento con los atributos propios de los mismos

- **es.example.rankingtapas.topos:** contiene las clases que actúan contra la plataforma Topoos, ya sea la de acceso como la de las categorías propias generadas para la aplicación (SYSTEM, USER, TAPA y EVENTO).
- **es.example.rankingtapas.widget:** contiene las clases necesarias para la visualización de las imágenes en las listas de los layouts
- **es.example.rankingtapas.analytics:** contiene las actividades que insertan la plataforma Analytics de las que extienden las principales actividades de la aplicación

## 8.5 Base de Datos

La aplicación RankingTapas crea y usa una base de datos llamada **DBRANK**. Todo el código referente a base de datos del sistema se encuentra en las clases *es.example.rankingtapas.internal.PersistenceSQL.java* y *es.example.rankingtapas.internal.PersistenceSQLiteHelper*.

Se crean dos tablas cada vez que se instala por primera vez la aplicación. Mientras no se instale nuevamente la misma, las tablas permanecerán en el sistema. Una vez que se instala por primera vez la aplicación o para siguientes instalaciones, estas tablas se borran y se crean de nuevo vacías. Por lo tanto si el usuario ha registrado anteriormente alguna tapa/evento en el sistema, no aparecerá en el listado de sus tapas/eventos sino en el listado de todas las tapas/eventos registradas en el sistema.

- **Poi:** se usa para insertar una nueva tapa y/o evento del usuario a través de un objeto creado exclusivamente para la aplicación llamado TAPENTO. El código que realiza esta inserción se encuentra en *es.example.rankingtapas.internal.PersistenceSQL.java*

```
public static void insertTapento(Context context, Tapento d)
```

El objeto Tapento contendrá esta información de cada tapa/evento que se inserte en el sistema

nombre	valor
id	Identificador de la tapa/evento. Será un POIid
type	categoría: TAPA, EVENTO
desc	Descripción de la tapa/evento
date	Fecha de inserción
urlFoto	url de la imagen en el caso de la tapa. En el caso de ser un evento se registra con el valor "evento"
votomas	Votos positivos de la tapa/evento
votomenos	Votos negativos de la tapa/evento
latitud	Latitud de la localización donde se inserta tapa/evento
Longitud	Longitud de la localización donde se inserta tapa/evento

**Nota:** inicialmente se diseñó y creó el objeto TAPENTO para operar con las tapas y eventos propios del usuario. Aunque existan los datos de los votos dentro de TAPENTO, el sistema de votos de la aplicación se gestionará únicamente con las operaciones de POI-Warnings, ver apartado de Votaciones del documento. Únicamente se usa esta tabla para controlar las tapas/eventos que registra el usuario. Mediante el id, que se obtiene a través de TOPOOS a la hora de registrar la tapa/evento se obtienen todas las tapas/eventos registrados por todos los usuarios que hagan uso de la aplicación RankingTapas

- **TABLE\_VOTO:** se usa para insertar el id de la tapa/evento que se vote en cada momento y poder controlar así, si dicha tapa/evento ha sido votado. El usuario solamente podrá votar una vez, tanto positiva como negativamente una tapa/evento. El código que realiza esta inserción se encuentra en *es.example.rankingtapas.internal.PersistenceSQL.java*

```
public static void addVoteTV(int poild, Context context)
```

Como se observa en el código anterior, se pasa por parámetro el identificador del objeto POI

A su vez, para controlar cada tapa/evento que se inserta en el sistema, se hace uso de la plataforma TOPOOS de manera que cuando ocurre esta inserción, se hace un registro de un nuevo objeto POI proporcionando datos como nombre, descripción, dirección. Una vez registrado, la plataforma proporciona un identificador del mismo que es el que usaremos para insertar TAPENTO e identificar la tapa/evento en cuestión.

En el siguiente ejemplo obtenemos la información de un objeto POI registrado en TOPOOS a través de la aplicación RankingTapas.

Para obtener esta información se ha invocado al método GET definido para las operaciones disponibles para los objetos POI en TOPOOS.

Ver documentación en el apartado de bibliografía y consultar apartado de pruebas y emulador

```
{"id":12774,"name":"evento","description":"macootoo","latitude":42.6138278,"longitude":-5.5546997,"elevation":0,"accuracy":0,"vaccuracy":0,"registertime":"2013-12-07T13:32:20.757+00:00","categories":[{"Id":396,"Description":"EVENTO","is_system_category":false},{"Id":398,"Description":"USER","is_system_category":false}],address:"Calle José Echegaray 2,24195 Villaquilambre","cross_street":"Calle José Echegaray 2","city":"Villaquilambre","country":null,"postal_code":"24195","phone":null,"twitter":"#evento","last_update":"2013-12-07T13:32:20.757+00:00","warnings":{"closed":1,"duplicated":0,"wrong_indicator":0,"wrong_info":0}}
```

Esta información es la que se gestiona a través de la aplicación.

El identificador del objeto registrado, proporcionado por TOPOOS y marcado en amarillo en el código anterior, lo usamos para insertar en la base de datos de la aplicación un objeto TAPENTO, que es la clase que se ha definido para contener la información de una tapa o evento del sistema

## 8.6 AndroidManifest

Una aplicación Android está definida por el contenido de su manifiesto. Cada aplicación Android declara todas sus actividades, puntos de entrada, interfaces de comunicación, permisos e *intents* a través del fichero **AndroidManifest.xml**

Nuestro fichero contiene todos esos permisos más los necesarios para poder acceder y mostrar los mapas proporcionados por la plataforma de Google Maps V2

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
package="es.example.rankingtapas" android:installLocation="auto"
```

```
android:versionCode="1"
android:versionName="1.0" >

<permission android:name="es.example.rankingtapas.permission.MAPS_RECEIVE"
    android:protectionLevel="signature"/>

<uses-permission
    android:name="es.example.rankingtapas.permission.MAPS_RECEIVE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
    android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"
/>

<uses-feature
    android:glEsVersion="0x00020000"
    android:required="true" />
<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="16" />

<application
    android:allowBackup="true"
    android:icon="@drawable/icono_ranking"
    android:label="@string/app_name"
    android:theme="@style/AppTheme">

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

    <activity
        android:name="es.example.rankingtapas.RankingTapasActivity"
        android:screenOrientation="portrait"
        android:label="@string/app_name">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name="es.example.rankingtapas.Pestana1"
        android:screenOrientation="portrait"
        >
    </activity>
    <activity
        android:name="es.example.rankingtapas.Pestana2"
        android:screenOrientation="portrait">
    </activity>
</application>
```

```
        android:name="es.example.rankingtapas.Pestana3"
        android:screenOrientation="portrait" >
    </activity>
    <activity
        android:name="es.example.rankingtapas.Pestana4"
        android:screenOrientation="portrait" >
    </activity>
    <activity
        android:name="es.example.rankingtapas.DetailTapentoActivity"
        android:theme="@android:style/Theme.Dialog"
        android:screenOrientation="portrait" >
    </activity>
    <activity
        android:name="es.example.rankingtapas.DetailMapTapaActivity"
        android:theme="@android:style/Theme.Dialog"
        android:screenOrientation="portrait" >
    </activity>

    <activity
        android:name="es.example.rankingtapas.MainActivity"
        android:screenOrientation="portrait" >
    </activity>
    <activity
        android:name="es.example.rankingtapas.AboutActivity"
        android:configChanges="orientation"
        android:screenOrientation="portrait" >
    </activity>
</application>
</manifest>
```

Las líneas marcadas se corresponden con la configuración necesaria que tiene que tener el fichero de manifiesto para la configuración de Google Maps V2. Consultar el apartado de Google Maps V2

## 8.7 Plataforma Google Analytics

Analytics es una plataforma disponible de Google que se utiliza para hacer un seguimiento de la interacción que tienen los usuarios con la aplicación a través de varios dispositivos y entornos.

La usaremos para poder obtener información referente a usuarios activos de la aplicación y desde dónde la están utilizando.

A su vez, también podremos acceder a posibles excepciones y mal funcionamiento de la misma

Requisitos para poder añadir esta plataforma a la aplicación:

- Android Developer SDK<sup>10</sup>
- Google Analytics SDK para Android v3<sup>11</sup>
- La propia aplicación Android para implementar Analytics
- Una cuenta en Analytics<sup>12</sup>

Configuración

1. Actualizar el fichero *AndroidManifest.xml* añadiendo los permisos siguientes

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

En nuestro caso ya disponíamos de estos permisos en el fichero

2. Añadir los métodos de EasyTracker a cada una de las Actividades de esta forma

```
package com.example.app;

import android.app.Activity;

import com.google.analytics.tracking.android.EasyTracker;

/**
 * An example Activity using Google Analytics and EasyTracker.
 */
public class myTrackedActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    public void onStart() {
        super.onStart();
        ... // The rest of your onStart() code.
        EasyTracker.getInstance(this).activityStart(this); // Add this method.
    }

    @Override
    public void onStop() {
        super.onStop();
        ... // The rest of your onStop() code.
        EasyTracker.getInstance(this).activityStop(this); // Add this method.
    }
}
```

3. Crear el fichero *analytics.xml* con los datos necesarios y añadirlo al directorio */res/values* del proyecto. El fichero en cuestión será el siguiente:

```
<?xml version="1.0" encoding="utf-8" ?>

<resources xmlns:tools="http://schemas.android.com/tools"
tools:ignore="TypographyDashes">

    <!--Replace placeholder ID with your tracking ID-->
    <string name="ga_trackingId">UA-XXXX-Y</string>

    <!--Enable automatic activity tracking-->
    <bool name="ga_autoActivityTracking">true</bool>

    <!--Enable automatic exception tracking-->
```

```
<bool name="ga_reportUncaughtExceptions">true</bool>
</resources>
```

Se pueden configurar más parámetros en el fichero, para más información consultar en el apartado de Fuentes de información en la sección de Analytics.

## 8.8 Actividad Principal

La clase *es.example.rankingtapas.MainActivity* es la que contiene la actividad principal de la aplicación. Extiende de la clase *android.app.TabActivity* pues se encarga de cargar las pestañas (tabs) del menú de la aplicación. Está incluido el código necesario para el uso de la plataforma Analytics (ver apartado Analytics)

```
package es.example.rankingtapas;

import com.google.analytics.tracking.android.EasyTracker;
import android.app.TabActivity;
import android.content.Intent;
import android.content.res.Resources;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.Window;
import android.widget.TabHost;
import android.widget.TabHost.OnTabChangeListener;
import es.example.rankingtapas.R;

@SuppressWarnings("deprecation")
public class MainActivity extends TabActivity {

    private static final int MENU_ACERCA_DE = Menu.FIRST;
    private static final int MENU_QUIT = Menu.FIRST + 1;

    @Override
    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.activity_main);

        // Tablas de pestañas

        final TabHost tabHost = getTabHost();// Creamos el tabhost de la actividad
        TabHost.TabSpec spec;// Creamos un recurso para las propiedades de la
        // pestaña

        tabHost.setOnTabChangedListener(new OnTabChangeListener() {

            @Override
            public void onTabChanged(String tabId) {

            }

        });

        Intent intent;// Intent que se utiliza para abrir cada pestaña
        Resources res = getResources();
```



```
// Pestaña1
intent = new Intent().setClass(this, Pestana1.class); // Se crea el

// intent para

// abrir la

// actividad(clases)

spec = tabHost
    .newTabSpec("Pestana1")
    .setIndicator("", res.getDrawable(R.drawable.pestana1_style))
    .setContent(intent);

tabHost.addTab(spec);

// Pestaña2
intent = new Intent().setClass(this,
Pestana2.class).addFlags(intent.FLAG_ACTIVITY_CLEAR_TOP); // Se crea el

// intent para

// abrir la

// actividad(clases)

spec = tabHost
    .newTabSpec("Pestana2")
    .setIndicator("", res.getDrawable(R.drawable.pestana2_style))
    .setContent(intent);

tabHost.addTab(spec);

// Pestaña3
intent = new Intent().setClass(this,
Pestana3.class).addFlags(intent.FLAG_ACTIVITY_CLEAR_TOP); // Se crea el

// intent para

// abrir la

// actividad(clases)

spec = tabHost
    .newTabSpec("Pestana3")
    .setIndicator("", res.getDrawable(R.drawable.pestana3_style))
    .setContent(intent);
tabHost.addTab(spec);

// Pestaña4
intent = new Intent().setClass(this,
Pestana4.class).addFlags(intent.FLAG_ACTIVITY_CLEAR_TOP); // Se crea el

// intent para

// abrir la

// actividad(clases)

spec = tabHost
```

```
        .newTabSpec("Pestana4")
        .setIndicator("",
            res.getDrawable(R.drawable.pestana4_style))
        .setContent(intent);
    tabHost.addTab(spec);

    //background color tabHost

    getTabHost().getTabWidget().getChildAt(0).setBackgroundResource(R.drawable.pestana1_drawable_shape); //Pestaña1

    getTabHost().getTabWidget().getChildAt(1).setBackgroundResource(R.drawable.pestana2_drawable_shape); //Pestaña2

    getTabHost().getTabWidget().getChildAt(2).setBackgroundResource(R.drawable.pestana3_drawable_shape); //Pestaña3

    getTabHost().getTabWidget().getChildAt(3).setBackgroundResource(R.drawable.pestana4_drawable_shape); //Pestaña4
    }
    //Metodo quit del menu contextual->salir de la aplicacion
    public void quit(){

        setResult(RESULT_OK);
        finish();

    }
    //Metodo AcercaDe del menu contextual->Mostrar pantalla Acerca de
    public void AcercaDe(){
        Intent intent = new Intent(this, AboutActivity.class);
        this.startActivity(intent);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        menu.add(0, MENU_ACERCA_DE, 0, getString(R.string.about));
        menu.add(0, MENU_QUIT, 0, getString(R.string.exit));
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case MENU_ACERCA_DE:
                AcercaDe();
                return true;
            case MENU_QUIT:
                quit();
                return true;
        }
        return false;
    }

    @Override
    public void onStart() {
        super.onStart();
        EasyTracker.getInstance(this).activityStart(this);
    }

    @Override
    public void onStop() {
        super.onStop();
        EasyTracker.getInstance(this).activityStop(this);
    }
}
```

```
}  
}
```

El layout correspondiente es `res/layout/activity_main.xml`

```
<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    android:background="#000"  
    android:orientation="vertical" >  
    <TabHost  
        android:id="@android:id/tabhost"  
        android:layout_width="fill_parent"  
        android:layout_height="fill_parent"  
        android:layout_alignParentTop="true"  
        android:background="@drawable/fondo_general_ranking" >  
        <LinearLayout  
            android:layout_width="fill_parent"  
            android:layout_height="fill_parent"  
            android:orientation="vertical" >  
            <TabWidget  
                android:id="@android:id/tabs"  
                android:layout_width="fill_parent"  
                android:layout_height="wrap_content" />  
            <FrameLayout  
                android:id="@android:id/tabcontent"  
                android:layout_width="fill_parent"  
                android:layout_height="fill_parent" />  
            </LinearLayout>  
        </TabHost>  
    </RelativeLayout>
```

## 8.9 Menú de dispositivo móvil

Aparte del menú de navegación de la aplicación comentada en el apartado de diseño, se ha incluido un menú a la hora de pulsar el botón de menú propio del teléfono.

Al pulsar este botón, el usuario dispone de dos opciones **Acerca de** y **Salir**.

La opción de **Acerca de** muestra la pantalla inicial de carga de la aplicación y con la opción **Salir** el usuario sale de la aplicación.

En la pantalla de visualizar todas las tapas y eventos cercanos en el mapa, se ha incluido una ventana de *Legal Notices* como recomendación de Google

*"If you use the Google Maps Android API in your application, you must include the Google Play Services attribution text as part of a "Legal Notices" section in your application. Including legal notices as an independent menu item, or as part of an "About" menu item, is recommended."*

## 8.10 Código de pantallas

### 8.10.1 Pantalla Inicial

Al entrar en la aplicación se cargará durante unos instantes la pantalla de inicio de la aplicación. La actividad que realiza esto es *es.example rankingtapasRankingTapasActivity* y el layout correspondiente es *splash.xml*.

En el fichero *AndroidManifest.xml* se observa la carga de esta actividad

```
<activity
  android:name="es.example rankingtapas.RankingTapasActivity"
  android:screenOrientation="portrait"
  android:label="@string/app_name">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

Y el resultado es el siguiente



En la aplicación RankingTapas hay cuatro pantallas principales correspondientes a las cuatro pestañas que forman el menú de navegación de la aplicación, como ya se comentó en el apartado de diseño. El usuario se puede mover por estas distintas opciones pulsando cada una de las pestañas.

La clase comentada anteriormente, *MainActivity* carga estas pestañas y las pantallas finales obtenidas a través del emulador son las que se incluyen a continuación.

**Nota:** dado que el código java de todas las actividades y sus layouts correspondientes es bastante amplio y que se adjunta el código junto con el documento, no se detallará dicho código en el mismo debido a su extensión

### 8.10.2 Pantalla 1

Se corresponde con la pantalla de compartir una tapa y/o un evento. La clase se llama *Pestana1* y extiende de *CustomActivity* que es la clase que inserta la plataforma Analytics y configura el acceso a Topoos a través de *AccessInterface*. El layout correspondiente es *pestana1.xml*



**Nota:** los botones de publicación se han modificado por motivos de diseño. Inicialmente aparecen inactivos (color gris) y se activarán (cambiando a color verde) cuando se haya cargado la foto de la tapa junto con su descripción a la hora de publicarla. En el caso de publicar un evento, su botón correspondiente se activará cuando el usuario haya insertado una descripción del mismo en el campo de texto. Una vez publicada la tapa y/o el evento, el botón correspondiente volverá a su estado inactivo inicial

Cuando el usuario inserta una nueva tapa y/o evento se realizan diferentes acciones en el código. Se chequea la activación de los botones correspondientes, ya sea Publicar Tapa o Publicar evento mediante el método *checkRegisterButtonStatus()* pues controla que haya una imagen cargada y una descripción en el campo de texto en el caso de la tapa, y que haya descripción en el caso del evento.

Se inserta la nueva tapa y/o evento mediante el código siguiente. (En este caso se inserta una tapa)

- 1) se hace el registro de un nuevo *mRegisteredPOI*, que es el objeto POI que usaremos proporcionado por la plataforma TOPOOS. Este registro se hace a través de la clase *AccessInterface*

```
mRegisteredPOI = AccessInterface.RegisterTapento(Pestana1.this, POICategories.TAPA,
image.getFilename_unique(), t.getText().toString(), m_DeviceLocation);
```

- 2) Se crea un Nuevo objeto tipo TAPENTO, diseñado exclusivamente para contener toda la información de la tapa/evento del usuario a introducir en la aplicación y se inserta el mismo en la tabla correspondiente de la base de datos de la aplicación

```
Tapento newTapento = new Tapento(mRegisteredPOI.getId(), POICategories.TAPA,
t.getText().toString(), new Date(), nombreFoto, 0,0,
m_DeviceLocation.getLatitude(),m_DeviceLocation.getLongitude() );
```

```
PersistenceSQL.insertTapento(Pestana1.this, newTapento);
```



### 8.10.3 Pantalla 2

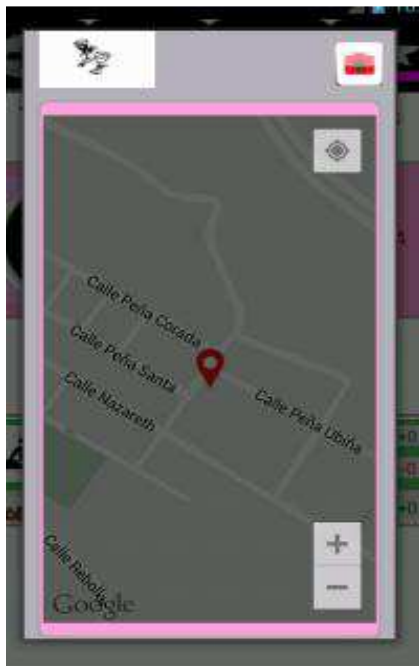
Se corresponde con la pantalla de visualizar las tapas y eventos introducidos por los usuarios de la aplicación. La clase se llama *Pestana2* y extiende de *AnalyticsListActivity* que inserta plataforma Analytics y configura acceso a Topoos a través de *AccessInterface*. *AnalyticsListActivity* extiende de *ListActivity* pues la finalidad es cargar una lista de elementos en la pantalla. El layout correspondiente es *pestanda2.xml*



Pantalla lista



Pantalla detalle evento



Pantalla mapa

**Nota:** el usuario selecciona un elemento de la lista, en este caso el evento. Aparece la pantalla de detalle evento con los botones de votar positivo, negativo y compartir. Si el usuario pulsa el icono la parte superior derecha de la pantalla, aparece un mapa con la localización de la tapa en el mapa. Para volver a la pantalla de detalle, el usuario deberá pulsar el icono de la parte superior derecha que se corresponde con la imagen de una cámara. La pantalla de detalle de la tapa es igual que la de detalle del evento pero con la imagen de la tapa. Se ha modificado la pantalla de detalle del evento permitiendo votar el mismo mediante la inclusión de los botones de votación

Cuando el usuario selecciona tapa/evento de la lista, se genera una llamada a la actividad *DetailtapentoActivity* y se le pasa la información del POI para poder visualizar la misma en la pantalla de detalle

```
Intent intent= new Intent(Pestana2.this,DetailTapentoActivity.class);
intent.putExtra("poi", m_pois.get(fixedpos));
startActivity(intent);
```

El layout correspondiente al detalle de la tapa/evento es *detail\_tapento\_layout.xml*

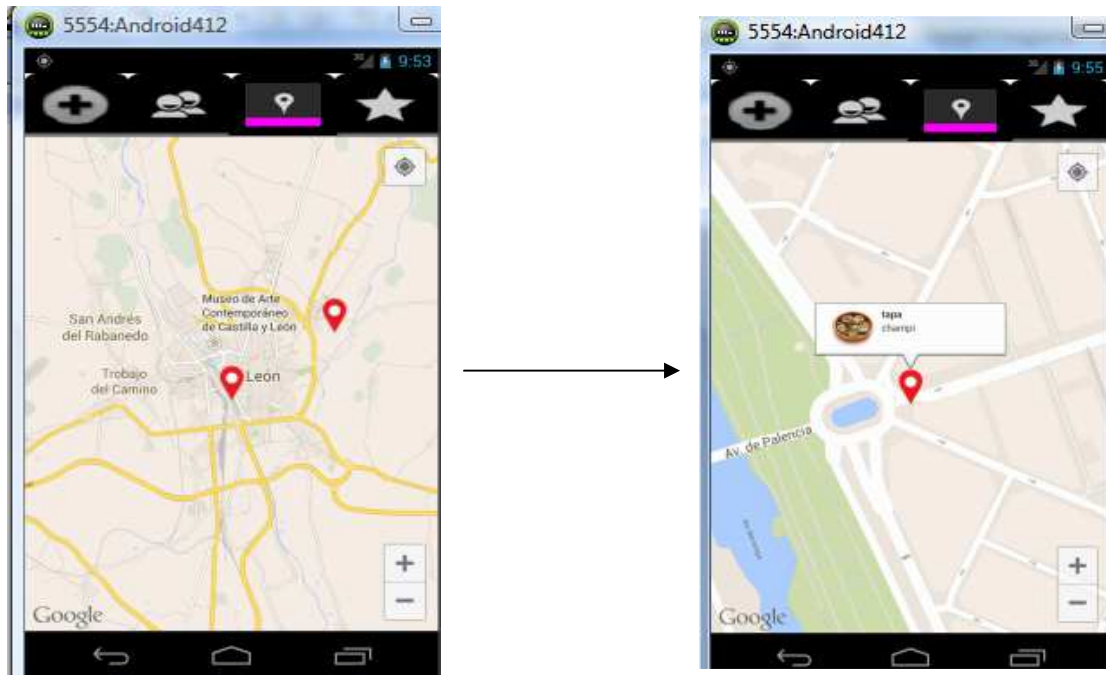
Si el usuario pulsa en el icono de visualizar mapa que se encuentra en la esquina superior derecha de la pantalla de detalle, se abrirá una nueva pantalla correspondiente a un mapa con una marca que será el punto de localización donde se registró dicha tapa/evento. El código que lanza esta nueva actividad, *DetailMapTapaActivity*, se encuentra en *DetailTapentoActivity* y a su vez se pasa la información del POI. El layout correspondiente a la pantalla de mapa del detalle de la tapa/evento es *detail\_maptapento\_layout.xml*

```
Intent intent = new Intent(getApplicationContext(),DetailMapTapaActivity.class);
intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP|
Intent.FLAG_ACTIVITY_SINGLE_TOP);
intent.putExtra("poi", mPOI);
startActivity(intent);
```

En la pantalla del detalle de la tapa/evento también existen tres botones correspondientes a las votaciones y a la acción de compartir. Se ha creado un apartado de votaciones en este documento. En cuanto a la acción de compartir, en el caso de la tapa se comparte la imagen de la misma y en el caso del evento se comparte su descripción. Esta funcionalidad solo funciona correctamente en un dispositivo real. Ver apartado de pruebas.

### 8.10.4 Pantalla 3

Se corresponde con la pantalla de visualizar en el mapa las tapas y eventos cercanos publicados por los usuarios con un máximo de carga de 30 tapas y/o eventos. La clase se llama *Pestana3* y extiende de *CustomMapActivity* que es la que inserta Analytics y configura acceso a Topoos a través de *AccessInterface*. *CustomMapActivity* extiende de *FragmentActivity* para poder realizar la carga del mapa. El layout correspondiente es *pestanda3.xml*



**Nota:** Se ha modificado el diseño de los iconos por motivos de estética. Cuando el usuario pulsa el icono, el mapa se centra en ese punto con un zoom y aparece una ventana de información con la imagen de la tapa y la descripción de la misma. En el caso de tratarse de un evento, la imagen correspondiente de la ventana de información será el icono de evento de la aplicación. A veces tarda en cargar la imagen y es necesario pulsar de nuevo en el icono.

El componente que se usa para la inclusión de mapas con la API de Google Maps V2 en el layout correspondiente puede ser de tipo *MapFragment* o *SupportMapFragment*. Esto nos permitirá entre otras cosas añadir uno [o varios, novedad de la API] mapas a cualquier actividad, sea del tipo que sea, y contando por supuesto con todas las ventajas del uso de fragments.

Dado que el nuevo control de mapas se basa en estos fragments, si queremos mantener la compatibilidad con versiones de Android anteriores a la 3, tendremos que utilizar la librería de soporte *android-support*.

En nuestra aplicación se ha usado el tipo *com.google.android.gms.maps*.

*SupportMapFragment*, y nos hemos asegurado de que la librería anteriormente mencionada, **android-support-v4.jar**, esté incluida en el proyecto y así la aplicación pueda ser ejecutada desde versiones antiguas de Android, concretamente desde la versión 2.2

En el apartado de Google Maps V2 de este documento se encuentra más información referente a este tema.

### 8.10.5 Pantalla 4

Se corresponde con la pantalla de visualizar la lista de las tapas y los eventos publicados por el propio usuario del móvil, con un máximo de carga de 30 tapas y/o



eventos. La clase se llama *Pestana4* y extiende de *AnalyticsListActivity* que inserta plataforma Analytics y configura acceso a Topoos a través de *AccessInterface*. *AnalyticsListActivity* extiende de *ListActivity* pues visualmente se trata de una lista de elementos en la pantalla. El layout correspondiente es *pestanda4.xml*



**Nota:** visualiza la lista de tapas y eventos publicados por el usuario. En la parte superior de la pantalla aparece un recuadro indicando el nivel que adquiere el usuario calculado en la misma clase *Pestana4*, según las votaciones obtenidas en sus tapas y eventos publicados. El usuario podrá acceder a un elemento en cuestión pulsando encima de él y aparecerá la pantalla de detalle de la tapa y evento, según el elemento seleccionado. Esta pantalla se corresponde con la pantalla vista en el apartado *Pantalla 2, detalle de pantalla*.

### 8.10.6 Votaciones

Inicialmente se propuso en el diseño de la funcionalidad del sistema, la votación únicamente para las tapas del mismo.

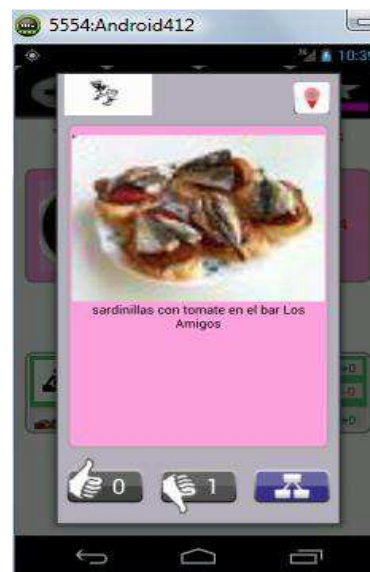
Se ha modificado dicha funcionalidad permitiendo al usuario votar también los eventos registrados.

Un usuario podrá votar solamente una vez la tapa y el evento que elija, ya sea el suyo como el de los demás usuarios publicados. La aplicación detecta si dicha tapa o evento ha sido votado alguna vez y activa o desactiva los botones de los votos según corresponda.

Cuando el usuario vota positivamente (pulsando el botón verde), el flujo de la actividad y la visualización de las pantallas es el siguiente.



Cuando el usuario vota negativamente (pulsando el botón rojo), el flujo de la actividad y la visualización de las pantallas es el siguiente.



Para poder gestionar los votos de cada tapa/evento, se ha hecho uso de la característica de POI-Warnings proporcionada por la plataforma TOPOOS<sup>13</sup>. Lo que hacemos es, al registrar un voto positivo, registrar un warning de duplicado en el POI y en el caso de voto negativo, registramos un warning de cerrado en el POI. Lo hacemos de esta manera:

#### VOTO POSITIVO

```
topoos.POIWarning.Operations.AddDuplicated(ctx,id);
```

## VOTO NEGATIVO

```
topoos.POIWarning.Operations.AddClosed(ctx,id);
```

Los Warnings en realidad no han sido diseñados para este propósito pero en la aplicación le hemos dado esta utilidad, así cuando queremos obtener el contador global de votos positivos y/o negativos de una tapa/evento, bastará con pedir la información del punto de interés a TOPOOS y leer el valor de estos warnings.

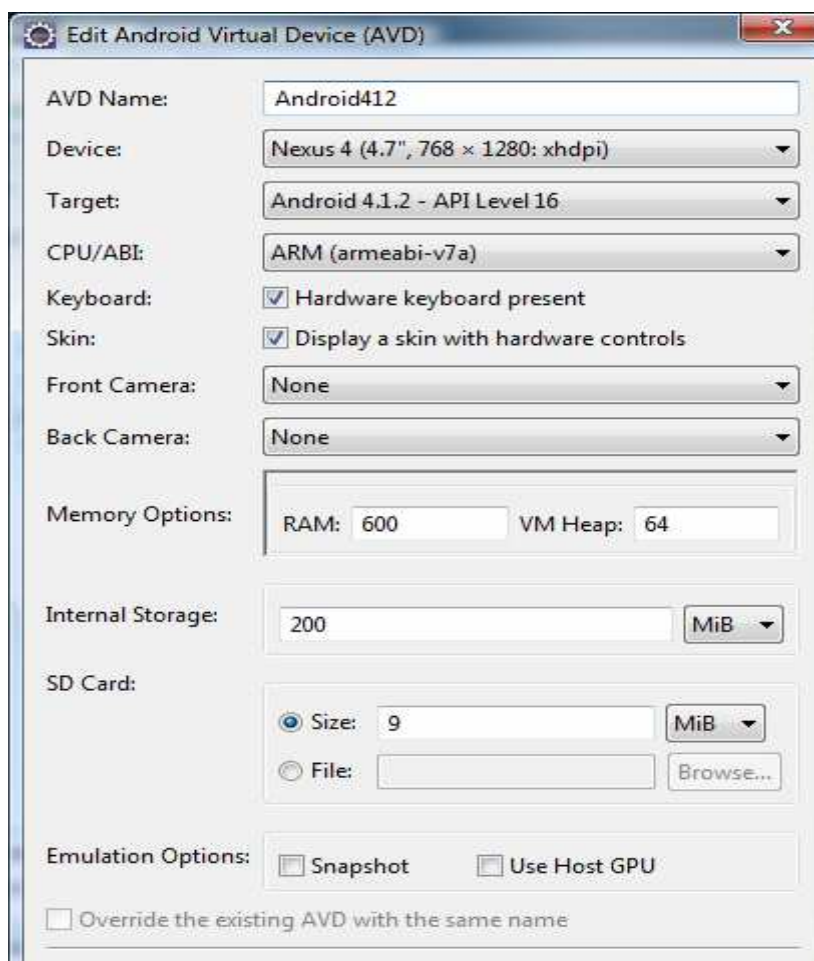
Cada vez que se registra un voto, se añade en la tabla correspondiente TABLE\_VOTO, el identificador del objeto POI. El objetivo es controlar qué tapa/evento ha sido votado para no volverlo a votar.

```
PersistenceSQL.addVoteTV(mPOI.getId(), DetailTapentoActivity.this);
```

## 8.11 Pruebas e Instalación

### 8.11.1 Pruebas en emulador

Para realizar la implementación del código y sus pruebas durante el mismo, se ha configurado finalmente un emulador con las siguientes características



Para que los mapas fueran visibles en dicho emulador, se ha instalado en él, después de arrancar el mismo los siguientes paquetes:

- com.google.android.gms-1.apk: instala la aplicación Google Play Services
- com.android.vending-1.apk: instala la aplicación Google Play Store

Una vez instalados, con el emulador arrancado, se ha configurado el mismo para añadir de forma manual y mediante el visor DDMS (desde eclipse) un punto geográfico y así poder realizar las acciones propias del LocationManager.

Es necesario que el emulador tenga acceso a Internet y que tenga su configuración WIFI activada, sino no es posible realizar las acciones de la aplicación. En un dispositivo real sin acceso a datos no funciona la aplicación y esta da error.

También se han añadido ciertas imágenes en la sdcard configurada en el emulador, a través de DDMS, para la realización de las pruebas correspondientes a la adición de la tapa.

Las pruebas sobre compartir mediante redes sociales y hacer fotografía a la hora de compartir una tapa, sólo funcionan en un dispositivo real.

### 8.11.2 Instalación

Para la instalación de la aplicación en un dispositivo real es necesario crear un ejecutable con extensión .apk que se hace a través de eclipse.

Para ello, previamente, se accede al menú de *Android tools* mediante el botón derecho sobre el proyecto y se selecciona la opción de *Exported signed Application Package*, se siguen las instrucciones generando una nueva clave y un alias.

Cada vez que se genera un .apk mediante el menú de *Export* → *Export Android Application*, será necesario incluir la contraseña de dicha clave junto con la contraseña de su alias correspondiente.

Para trasladar el fichero ejecutable al dispositivo real, se conecta el mismo al ordenador, a través del cable USB. Previamente habrá que haber activado la opción de *Fuentes desconocidas* en el menú de configuración de aplicaciones en el móvil.

Se copia el fichero, si hubiera una instalación anterior de la aplicación en el móvil se borran sus datos y se desinstala, se desconecta correctamente el dispositivo y desde el mismo se instala el nuevo ejecutable de la aplicación insertado en él.

### 8.11.3 Acceso a servicios de TOPOOS

En la web oficial de Topoos se pueden consultar posibles servicios para visualizar POIs generados, categorías, etc... Cada vez que se añade una tapa y/o un evento, se genera un nuevo POI cuyo identificador es el que vamos a usar para realizar la mayoría de las operaciones dentro de la aplicación.

Por ejemplo, para obtener los POIs cercanos y posteriormente visualizarlos en la lista de la pestaña 2, usamos el método *getUserTapentosNear* implementado en la clase *AccessInterface*, dentro del mismo se llamará al método de la API de Topoos correspondiente.

Para obtener ciertos POIs mediante un filtro, como el que usamos para visualizar los POIs propios del usuario en la pestaña4, lo hacemos a través del método *getUserTapentosWhere* implementado en la clase *AccessInterface*, dentro del mismo se llamará al método de la API de Topoos correspondiente.

Para la realización de las pruebas ha sido necesario contar con los servicios de Topoos relacionados con POI, tales como

- **GET** para obtener los datos de un determinado POI  
[https://api.topoos.com/1/pois/get\\_where.json?oauth\\_token=b9955415-82d3-4481-a736-330d962df9ea&categories=21&pois=6595](https://api.topoos.com/1/pois/get_where.json?oauth_token=b9955415-82d3-4481-a736-330d962df9ea&categories=21&pois=6595)
- **UPDATE** para actualizar los datos tales como latitud y longitud de un POI concreto  
[https://api.topoos.com/1/pois/update.json?oauth\\_token=62f5280d-c954-4ee7-a0a8-68ff9c2810d8&poi=6902&lat=40.953107&lng=-4.123233](https://api.topoos.com/1/pois/update.json?oauth_token=62f5280d-c954-4ee7-a0a8-68ff9c2810d8&poi=6902&lat=40.953107&lng=-4.123233)
- **DELETE** para borrar un POI específico  
[https://api.topoos.com/1/pois/delete.json?oauth\\_token=b9955415-82d3-4481-a736-330d962df9ea&poi=6547](https://api.topoos.com/1/pois/delete.json?oauth_token=b9955415-82d3-4481-a736-330d962df9ea&poi=6547)

TOPOOS proporciona estos servicios y muchos más que se pueden consultar en el apartado Reference de su web

***Nota:** Para el uso de los servicios GET, UPDATE y DELETE, los POIs tienen que estar previamente creados e insertados en la plataforma. El Token debe ser el creado por el usuario cuando configura su cuenta en TOPOOS.*

## 9 Simulación

En este capítulo se va a tratar la presentación de una simulación mediante la grabación de un vídeo con el programa Camtasia 8.

El vídeo en cuestión, vmarcoss\_video.zip, se sube al directorio correspondiente mediante FTP.

En la simulación se hace una breve introducción de lo que consiste la aplicación RankingTapas y se muestra el funcionamiento de la misma.

Para realizar el vídeo, se ha configurado un emulador virtual, mirar apartado de Pruebas e Instalación del presente documento, en el que no se han podido ejecutar ciertas acciones como realizar fotografía con la cámara a la hora de publicar una tapa o compartir la información mediante aplicaciones sociales.

Tampoco se puede acceder al menú contextual a través del emulador.

No obstante dichas acciones se ejecutan correctamente en un dispositivo real.

La simulación es muy lenta pues el funcionamiento del emulador no es tan rápido como un dispositivo móvil, sumando además procesos lanzados del propio eclipse y el programa de realización de vídeo. Esto ha hecho que el vídeo tarde un poco más de lo esperado. Se ha intentado ajustarlo lo más posible a 15 minutos, quedando representada la funcionalidad de la aplicación.

## 10 Conclusiones

La elección de realizar este proyecto enfocado a Android fue una elección personal debido al auge que tiene hoy en día este tipo de aplicaciones.

Desde un punto de vista profesional, es importante estar actualizado en este tipo de tecnologías, más en el ámbito informático, dada la demanda laboral actual. Desde un

punto de vista personal, supone un reto para mí y es un aliciente más para crear una aplicación de este tipo partiendo de documentación sin la realización, por ahora, de ningún curso enfocado a este sistema.

La realización de la implementación del diseño propuesto y las posteriores pruebas tanto en emuladores, como en un dispositivo real, no ha sido fácil debido a que se ha tenido que superar una serie de retos y tomar diferentes decisiones para completar su desarrollo como la modificación en algunos puntos del diseño visual y la adición de funcionalidad de votos para los eventos, no propuesta inicialmente en el apartado de diseño.

Estos problemas han hecho dedicar muchas horas en la implantación y codificación provocando la alteración de ciertas consideraciones comentadas en el apartado de diseño como la inclusión de publicidad dentro de la aplicación.

El mayor esfuerzo ha consistido en entender e integrar correctamente la API de Google Maps V2, así como entender el funcionamiento de la plataforma TOPOOS y la inclusión de la misma en la aplicación.

Para realizar el código de este proyecto, se ha reutilizado código ya existente de aplicaciones parecidas y se ha modificado para el pertinente uso de la misma.

Se han consultado bastantes fuentes de información, foros de usuarios y peleado con otras plataformas necesarias para la aplicación, considerando estas muy útiles dentro de esta tecnología.

Finalmente se han respetado y conseguido las principales funcionalidades propuestas en el proyecto.

Se ha realizado una simulación con el programa Camtasia 8 de la aplicación RankingTapas, teniendo en cuenta que ciertas ejecuciones no se han podido realizar en el emulador virtual previamente configurado. Sin embargo estas mismas acciones sí se pueden ejecutar correctamente en un dispositivo móvil real.

Desde un punto de vista personal, ver como con los conocimientos adquiridos durante la carrera en las diversas asignaturas ha permitido desarrollar una aplicación con una verdadera utilidad ha supuesto una alta satisfacción propia. Si a esto sumamos los nuevos conocimientos adquiridos en la etapa de autoaprendizaje a lo largo del desarrollo de este proyecto, se ve cómo se cumplen con creces los objetivos propuestos y se amplían conocimientos a la vez que se plasman los que ya se poseían.

Desde el punto de vista del desarrollo, a lo largo de este proyecto se ha visto como el mercado de aplicaciones para dispositivos móviles está en pleno auge. Cada día que pasa se suman más desarrolladores a este campo y es fácil quedarse obsoleto para cualquiera que se dedique al desarrollo en este entorno. Lo que está claro es que este sector es aún joven y que puede dar mucho de sí.

## 11 Referencias bibliográficas

- 1 [http://noticias.lainformacion.com/economia-negocios-y-finanzas/la-nueva-era-tecnologica-empieza-en-2013\\_MXLK3lou8w1sonq1yFkrR4/](http://noticias.lainformacion.com/economia-negocios-y-finanzas/la-nueva-era-tecnologica-empieza-en-2013_MXLK3lou8w1sonq1yFkrR4/)
- 2 <http://www.xatakandroid.com/mercado/en-espana-se-descargan-87-millones-de-aplicaciones-de-android-al-ano-mas-que-ios>
- 3 <http://www.xatakandroid.com/aplicaciones-android/en-android-los-usuarios-no-pagan-por-aplicaciones-solo-un-4-y-de-forma-particular>
- 4 <http://es.wikipedia.org/wiki/Freemium>
- 5 [http://developer.android.com/guide/practices/screens\\_support.html](http://developer.android.com/guide/practices/screens_support.html)
- 6 <https://www.fluidui.com/editor/live/>
- 7 <http://www.eclipse.org/downloads/index-helios.php>
- 8 <http://www.eclipse.org/jdt/>
- 9 <http://developer.android.com/tools/sdk/eclipse-adt.html>
- 10 <http://developer.android.com/sdk/index.html>
- 11 <https://developers.google.com/analytics/devguides/collection/android/resources>
- 12 <https://support.google.com/analytics/answer/2614741?hl=en>
- 13 <http://docs.topoos.com/reference/poi-warnings/>

## 12 Fuentes de Información

Para la realización del presente proyecto se han utilizado las fuentes de información que se muestran a continuación.

### Documentación online Android oficial y no oficial

<http://developer.android.com>

<http://androideity.com/2011/08/05/patrones-de-diseno-en-android/>

<http://www.aprendeandroid.com/l6/keyMaps.htm>

<http://www.limecreativelabs.com/anadir-archivos-tarjeta-dispositivo-virtual-android/>

### Documentación online Topoos

<http://docs.topoos.com/tools/sdks/android/>

<https://developers.topoos.com/>

### Diversos artículos

<http://findemor.porexpertos.es/topoos-api-desarrollo-aplicaciones-moviles-dependientes-contexto/>

<http://blog.piensaenpixels.com/consejos-para-disenar-aplicaciones-o-paginas-web-para-moviles-ii/263>

<http://blog.findemor.es/2012/12/ejemplo-desarrollo-aplicacion-android-completa/>

<http://jagonzalez.org/ejecutar-aplicaciones-android-en-el-emulador-del-sdk/>

[http://www.programacion.com/articulo/introduccion\\_a\\_los\\_layouts\\_para\\_android\\_400](http://www.programacion.com/articulo/introduccion_a_los_layouts_para_android_400)

<http://www.xatakandroid.com/tutoriales/borrando-dalvik-cache-para-ganar-espacio>

## Prototipados de aplicaciones móviles

<http://blog.aplicacionesmovil.com/aplicaciones-celular/herramientas-de-prototipado-de-aplicaciones-moviles/>

<http://blog.mp3.es/como-crear-plantillas-web-con-pencil-project/>  
<https://www.fluidui.com/editor/live/>

## Consola Google

<https://code.google.com/intl/es/>

## Google Maps v2

[https://developers.google.com/maps/documentation/android/start#getting\\_the\\_google\\_maps\\_android\\_api\\_v2](https://developers.google.com/maps/documentation/android/start#getting_the_google_maps_android_api_v2)

<http://android-er.blogspot.com.es/2012/12/a-simple-example-using-google-maps.html>

<http://maurizziof.blogspot.com.es/2013/08/como-usar-googlemaps-api-v2-en.html>

<http://proyectosbeta.net/2012/12/configurar-y-usar-google-maps-api-v2-para-android/>

<https://developers.google.com/maps/documentation/android/intro>

<http://www.sgoliver.net/blog/?p=3244>

<http://mobiforge.com/design-development/developing-with-google-maps-v2-android>

[http://www.vogella.com/articles/AndroidGoogleMaps/article.html#maps\\_mapsview](http://www.vogella.com/articles/AndroidGoogleMaps/article.html#maps_mapsview)

<http://www.androidcurso.com/index.php/recursos-didacticos/tutoriales-android/41-unidad-7-seguridad-y-posicionamiento/223-google-maps-api-v2>

<http://www.aprendeandroid.com/l6/keyMaps.htm>

<http://adrianlatorre.tumblr.com/post/63258189655/de-sarrollando-en-android-2-google-maps-api>

<https://github.com/sgolivernet/curso-android-src/tree/master/android-mapas-api2-p3/res>

<http://androideity.com/2013/05/05/google-maps-android-api-v2/>

<http://stackoverflow.com/questions/14040185/running-google-maps-v2-on-android-emulator>

<http://maurizziof.blogspot.com.es/2013/08/como-usar-googlemaps-api-v2-en.html>

## Herramientas

<https://www.fluidui.com/editor/live/>

<http://html-color-codes.info/codigos-de-colores-hexadecimales/>

[http://www.agenciacreativa.net/coordenadas\\_google\\_maps.php](http://www.agenciacreativa.net/coordenadas_google_maps.php)



<http://www.techsmith.com/camtasia.html>

## **Analytics**

<https://developers.google.com/analytics/devguides/collection/android/resources>

<https://developers.google.com/analytics/devguides/collection/android/parameters>

<http://www.google.es/intl/es/analytics/>

## **Otros enlaces**

<http://stackoverflow.com/questions/3769762/android-color-xml-resource-file/3770183#3770183>

<http://www.mkyong.com/android/android-imageview-example/>

<http://stackoverflow.com/questions/15037779/how-do-you-display-google-maps-via-new-google-maps-api-v2-on-android>

<http://stackoverflow.com/questions/13957462/replacement-for-itemizedoverlay-in-google-maps-v2-for-android>

<http://www.androidexperto.com/trucos-consejos-android/instalar-aplicaciones-manualmente/>

<http://www.genbetadev.com/expertos/respuestas/probar-aplicaciones-android-en-mi-telefono>

<http://www.limecreativelabs.com/anadir-archivos-tarjeta-dispositivo-virtual-android/>

<http://stackoverflow.com/questions/14040185/running-google-maps-v2-on-android-emulator>

<http://maurizziof.blogspot.com.es/2013/08/como-usar-googlemaps-api-v2-en.html>

<http://ingjoseberna.blogspot.com.es/2013/01/geolocalizacion-y-utilizacion-de-mapas.html>