

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
Grado en Ingeniería del Software

**Desarrollo de una aplicación en Android para la
Cámara de Comercio de Málaga**

**Development of an Android application for the
Malaga Chamber of Commerce**

Realizado por
María Victoria Díaz Redondo
Tutorizado por
Eduardo Guzmán de los Riscos
Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, febrero de 2015

Fecha defensa:
El Secretario del Tribunal

Resumen: El uso de dispositivos móviles que incorporan un gran número de funcionalidades que poseen ordenadores tradicionales, en la actualidad se ha extendido notablemente entre la población. Hoy en día, en nuestro país, la tasa de penetración supera el 65% y es el sistema operativo móvil de Google, Android, el que lidera el mercado y casi a nivel mundial superando a su rival más directo, el sistema iOS de Apple.

En este trabajo se pretende ofrecer un servicio al usuario de smartphone con sistema Android, donde pueda encontrar la información más relevante que la Cámara Oficial de Comercio de Málaga publica en su sitio web.

A lo largo de este documento se presentará una evolución de la telefonía móvil, una comparación entre los sistemas operativos más relevantes del mercado actual y a continuación el análisis, diseño e implementación de la solución propuesta. Esta constará de una aplicación cliente para instalar en dispositivos móviles, y otra que actuará de servidora de contenidos.

En la pantalla inicial se ofrece el acceso a las ofertas de servicios, y funcionalidad para contactar por correo electrónico o por teléfono. La información de los servicios ofertados se ofrece en formato listado-detalle, ofreciendo la solicitud de ampliación de información en un email formateado.

Con acceso desde la pantalla inicial, también se ofrecen servicios de geolocalización, mediante un sistema de rutas, en la que el usuario puede ver la localización de la Cámara de Comercio, calcular su ruta para llegar, y puede seguir esta ruta mediante una representación etapa por etapa, de manera gráfica, en el mapa, con indicaciones detalladas de camino, duración, etc.

Palabras claves: Android, JSON, SQLite, Google Maps API, ListView, Eclipse, Java, aplicación móvil, Cámara de Comercio de Málaga

Abstract: The use of mobile devices has currently spreaded among the population. Nowadays, many of these devices include features of traditional computers. The penetration rate exceeds 65% and the operating system for mobile Google Android leads the market and almost worldwide, surpassing its most direct rival, i.e. the Apple iOS system.

This work intends to offer a service to the Android mobile devices users, where they will find the most relevant information that the Chamber of Commerce of Malaga publishes on its web site.

Throughout this document an evolution of mobile telephony will be presented and a comparison between the most relevant operating systems on the current market. Next, the document presents the phases of development: analysis, design and implementation of the proposed solution. This solution will consist of a client application to be installed on Android mobile devices, and another that performs as a content service.

The main screen provides access to services and also it allows the user to contact by either email or phone with the Chamber of Commerce.

The information of the services offered by each area is displayed like a list view and a detail view. This detailed view provides the request for further information in a previously formatted email.

The app also includes geolocation services, through a system of routes, in which the user can see the location of the Chamber of Commerce, calculate your route to get, and follow this route through a representation on the map, step by step, with detailed directions like duration and distance.

Keywords: Android, JSON, SQLite, Google Maps API, ListView, Eclipse, Java, app, mobile application, Chamber of Commerce of Malaga.

Índice de Contenidos

CAPÍTULO 1: INTRODUCCIÓN	3
1.1 INTRODUCCIÓN AL TFG	3
1.2 OBJETIVOS	3
1.3 ETAPAS DE DESARROLLO	5
1.4 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	5
CAPÍTULO 2: ESTADO DEL ARTE	7
2.1 EVOLUCIÓN DE LA TELEFONÍA MÓVIL.....	7
2.2 PLATAFORMAS MÓVILES	10
2.2.1 <i>Android</i>	10
2.2.1 <i>iOS</i>	11
2.2.3 <i>Comparativa distintas plataformas</i>	12
2.3 CÁMARA DE COMERCIO DE MÁLAGA	13
2.3.1 <i>¿Qué es la Cámara y quiénes forman parte?</i>	13
2.3.2 <i>¿Cómo desarrolla su actuación la Cámara?</i>	13
2.3.3 <i>Servicios que ofrece la Cámara</i>	14
CAPÍTULO 3: SISTEMA DE DESARROLLO: ANDROID	15
3.1 SISTEMA ANDROID	15
3.1.1 <i>Características y especificaciones</i>	16
3.1.2 <i>Versiones de Android</i>	17
3.2 HERRAMIENTAS: ANDROID SDK Y ECLIPSE.....	18
3.3 COMPONENTES DE UNA APP ANDROID	18
CAPÍTULO 4: DESARROLLO DE LA APLICACIÓN	21
4.1 ANÁLISIS Y DISEÑO DE LA APLICACIÓN.....	21
4.1.1 <i>Requisitos funcionales</i>	21
4.1.2 <i>Metodología de diseño (cascada y prototipado)</i>	22
4.2 IMPLEMENTACIÓN	22
4.2.1 <i>Desarrollo web</i>	22
4.2.2 <i>Desarrollo de herramientas locales</i>	24
4.2.3 <i>Desarrollo de la app</i>	26
4.3 PRUEBAS CON EL EMULADOR Y DISPOSITIVO REAL.....	38
4.3.1 <i>Emulador</i>	38
4.3.2 <i>Dispositivo real</i>	39
CAPÍTULO 5: CONCLUSIONES Y TRABAJOS FUTUROS	41
5.1 CONCLUSIÓN DEL TFG.....	41
5.2 FUTURAS AMPLIACIONES	41
REFERENCIAS BIBLIOGRÁFICAS	43

Índice de Figuras

Figura 1 - Handie Talkie H12-16 [2].....	7
Figura 2 - Motorola DynaTac [2].....	7
Figura 3 - Teléfonos móviles 2G [2].....	8
Figura 4 - Teléfonos móviles 3G [2].....	9
Figura 5 - Teléfonos móviles 4G [4].....	9
Figura 6 - Cuota de mercado de Sistemas móviles en España [11]	12
Figura 7 - Arquitectura Sistema Android [9]	15
Figura 8 - Versiones del Sistema Android	17
Figura 9 - Reparto de versiones en el mercado actual [12].....	17
Figura 10 - Ciclo de vida de la clase Activity [13].....	19
Figura 11 - Extensión en Typo3	23
Figura 12 - Ejemplo de misión comercial.....	23
Figura 13 - Cadena JSON	24
Figura 14 - Conector ODBC - MySql.....	25
Figura 15 - Base de datos Comercio Exterior.....	25
Figura 16 - Paquetes de la aplicación.....	26
Figura 17 - Casos de uso.....	27
Figura 18 - Clases del modelo de negocio	28
Figura 19 - Clase DatabaseManagerHelper	28
Figura 20 - Clase JSON.....	29
Figura 21 - Diagrama de secuencia.....	29
Figura 22 - Carga JSON.....	30
Figura 23 - Parse JSON-Curso.....	31
Figura 24 - Clase Curso BaseColumns.....	32
Figura 25 - Guardado de datos en SQLite	32
Figura 26 - Cursor Adapter	33
Figura 27 - Descarga y guardado de imágenes.....	34
Figura 28 - Método crearMapa	35
Figura 29 - Método calcularRuta	36
Figura 30 - Método onPathSetListener	36

Capítulo 1: Introducción

1.1 Introducción al TFG

El proyecto tiene como finalidad el estudio de las posibilidades de negocio y promoción que ofrece el intercambio de datos y la sinergia entre sitio Web y aplicación móvil, generando contenido informativo actualizado en dispositivos móviles, y posibilitando de esta manera alcanzar a un mayor número de clientes con la información actualizada en tiempo real.

Se desarrollará una aplicación que permita, a los usuarios de dispositivos móviles con sistema Android, acceder en cualquier momento a la información actualizada de los servicios y ofertas que presta la Cámara de Comercio de Málaga.

El proyecto será desarrollado usando la plataforma de desarrollo recomendada por Google, Eclipse, y el gestor de base de datos *SQLite Database Browser*.

Paralelamente a este proyecto, el alumno D. Francisco Andrade García realizará el desarrollo de esta aplicación móvil para dispositivos con sistemas iOS en otro TFG.

1.2 Objetivos

Con el diseño, implementación e implantación de esta herramienta, se pretende:

Perfeccionar la técnica de **toma de requisitos** de usuario, mediante la identificación y entrevista de los responsables de los servicios de la corporación, con el fin de estimar cuáles son susceptibles de ofrecerse mediante una solución de movilidad.

Una vez realizadas las entrevistas, se procederá a especificar el **ámbito del software** a desarrollar, con las funcionalidades y requerimientos necesarios. De acuerdo con esto se podrá realizar una mejor estimación de las fases temporales e hitos de desarrollo del proyecto.

Mediante el uso de **diagramas de clase** y **casos de uso**, se realizará el diseño del software a desarrollar.

Sin necesidad de especificar los servicios de la Cámara de Comercio que se tratarán en la aplicación, el desarrollo requerirá del diseño de una **funcionalidad web**, que se comunicará al menos con dos servidores de la

corporación. Esta parte ha de obtener la información de dos bases de datos, y generar una salida de forma que pueda ser procesada y presentada por la aplicación móvil. Este servicio se desarrollará en PHP, y generará una salida JSON (*JavaScript Object Notation*).

La **aplicación móvil** procesará esta salida y almacenará los datos en una base de datos **SQLite** en el dispositivo móvil, con el fin de que esté disponible para una consulta sin conectividad (en cuyo caso se indicaría al usuario la fecha de actualización de la información).

A partir de la información almacenada en esta base de datos, la aplicación presentará a los usuarios la información categorizada de manera dinámica, a distintos niveles, según la información provista por el sitio Web. La aplicación estará desarrollada para adaptarse a los requerimientos de los distintos dispositivos, móviles y tabletas, en sus distintos tamaños de pantalla y resolución, y las últimas versiones del sistema Android.

Se utilizarán métodos de desarrollo en cascada y prototipado, en la fase de diseño y desarrollo inicial e implantación.

Como etapa final se hará una fase de implantación y pruebas entre los usuarios, en la que se podrá verificar el correcto funcionamiento de la aplicación.

Pedagógicamente, podemos considerar los siguientes objetivos del proyecto:

- 1) **Estudio de la Plataforma.** Se analizará el entorno Eclipse (**ADT**) y los distintos servicios que provee a la hora del desarrollo de aplicaciones móviles para sistemas Android, así como el lenguaje Java.
- 2) Estudio de mecanismos de **intercambio de datos**. Se estudiarán las tecnologías **PHP y JSON** para la actualización de contenidos desde el sitio web, en aplicaciones móviles.
- 3) Estudio de **Técnicas de Bases de datos**. Se diseñará y creará una base de datos **SQLite**, almacenada en el dispositivo móvil.
- 4) **Implementación de la aplicación móvil.** Aplicando técnicas y estándares estudiados anteriormente se desarrollará una aplicación móvil para mostrar la información de interés para el usuario en dispositivo móvil con sistema **Android**.

1.3 Etapas de desarrollo

El proyecto se realizará en 4 etapas:

- 1) Toma de requisitos: 10 horas
- 2) Análisis, diseño, e implementación de la aplicación web: 24 horas
- 3) Análisis, diseño, e implementación de las aplicaciones móviles para los dispositivos (teléfonos y tabletas): 223 horas.
- 4) Implantación en los dispositivos de la empresa, y pruebas: 40 horas
- 5) Subida a Play Store: 3 horas.

1.4 Descripción de la solución propuesta

Para el cumplimiento de los objetivos del Trabajo Fin de Grado, se presenta una aplicación móvil que presenta información relevante desde los sitios web camaramalaga.com y formacioncamara.com, mostrándola de forma adecuada al dispositivo, y de manera funcional, a usuarios de dispositivos con sistemas Android.

La aplicación constará de dos partes:

- La primera parte consiste en una aplicación web que permite la extracción de datos de los distintos portales de la corporación donde se aloja. Esta parte ha de obtener la información de dos bases de datos, y codificar los resultados obtenidos en las consultas en un objeto JSON cuya estructura se definirá en la fase de estudio del proyecto. De este objeto *JSON* obtendrá la aplicación móvil los datos necesarios para presentar la información a los usuarios
- La segunda parte del proyecto es el diseño y desarrollo de la aplicación móvil. Ésta almacenará los datos en una base de datos en el dispositivo, y la presentará al usuario, en una estructura que se definirá en el propio proyecto. En el caso de que el dispositivo se encuentre sin conexión, se mostrará la información de la base de datos descargada en la última visita con conexión.

Capítulo 2: Estado del arte

2.1 Evolución de la telefonía móvil

La historia de la telefonía móvil se divide en generaciones marcadas por ciertos hitos tecnológicos y comerciales. A continuación se describirán las características y principales dispositivos de cada una de ellas.

- **Generación 0:** comienza a finales del siglo XIX y se trata de telefonía basada en radio y estaba orientada a fines específicos como la navegación marítima o militares. Los estándares de comunicación en esta generación son: **IMTS** (*Improved Mobile Telephone System*) y **PTT** (siglas de *Push To Talk* - pulsar para hablar).



En los años de la Segunda Guerra Mundial, la compañía Motorola lanzó el *Handie Talkie H12-16*, el cual permitía comunicarse a distancia entre las tropas. [1]

Figura 1 - Handie Talkie H12-16 [2]

- **Generación 1:** aparece a finales de los años 1980 y es considerada realmente como de teléfonos móviles. Se trata de telefonía orientada al sector empresarial que funcionaba por medio de comunicación analógica, es decir, que la transmisión y recepción de datos se apoyaba sobre un conjunto de ondas de radio que cambiaban de modo continuo. La compañía japonesa NTT (*Nippon Telegraph and Telephone Corporation*) lidera la comercialización seguida por las compañías de los países nórdicos, las cuales implementaron la itinerancia internacional conocida como *Roaming* mediante el estándar NMT. Otras empresas formaron parte del sector como fueron ATT y Vodafone se incorporaron en estos años. Los estándares de comunicación en esta generación son: **NMT** (*Nordic Mobile Telephone*) y **AMPS** (*Advanced Mobile Phone System*).



Los dispositivos eran de gran tamaño y peso. Sólo podían ser utilizados para la transmisión de voz y la seguridad era muy baja, se podían escuchar conversaciones ajenas mediante un sintonizador de radio e incluso cargar el importe de la llamada a otra persona haciendo uso de las frecuencias. Motorola lanzó el terminal *DynaTac* en 1983.[2]

Figura 2 - Motorola DynaTac [2]

- **Generación 2:** aparece a finales de los años 1990 e incorpora la telefonía digital basada en conmutación de circuitos, es decir, se produce una comunicación de extremo a extremo. Se mejora el manejo de llamadas, se establecen más enlaces simultáneos en el mismo ancho de banda y se integran otros servicios adicionales al de la voz como el Servicio de Mensajes Cortos (*Short Message Service* o **SMS**). El estándar más utilizado es el **GSM** (*Global System for Mobile Communications*) que incorpora itinerancia internacional, una implementación eficiente de los SMS hasta 160 caracteres, mensajes multimedia MMS y el protocolo WAP para acceso a Internet.
- **Generación 2.5:** se refiere a teléfonos móviles 2G que incorporan algunas de las mejoras y tecnologías del estándar 3G como es el caso del estándar **GPRS** (*General Packet Radio Service* - Servicio General de Radio por Paquetes) que potencia el uso de los mensajes MMS, el protocolo WAP y la utilización de terminales multimedia y de servicios multicontenido.



Figura 3 - Teléfonos móviles 2G [2]

- **Generación 3:** surge a principios del año 2000 y se caracteriza por la adopción de comunicación basada en paquetes para los servicios de datos, ya que aparece la necesidad de aumentar la capacidad de transmisión de datos para poder ofrecer servicios como la conexión a Internet desde el móvil, la videoconferencia, la televisión y la descarga de archivos. Aparece el estándar **UMTS** (*Universal Mobile Telecommunications System*) que proporcionan un mayor ancho de banda y los primeros servicios de *Streaming* para contenidos de vídeo y radio. Supuso una mejora de la conectividad, un gran incremento en el consumo de datos y la creación de dispositivos con mayores funciones y capacidades de procesamiento más altas, los *smartphones*. Las pantallas fueron más grandes, a color y táctiles, incorporando

procesadores, con gran poder gráfico y conectividad totalmente cubierta, convirtiéndose en dispositivos multifuncionales. [2]

- **Generación 3.5:** se caracteriza por la adopción del estándar **HSDPA** (High Speed Downlink Packet Access) que proporciona una mayor velocidad de transmisión por lo que favorece un acceso a Internet, mejorando la significativamente la capacidad máxima de transferencia de información y ofreciendo una menor latencia.



Figura 4 - Teléfonos móviles 3G [2]

- **Generación 4:** es la generación actual y aparece desde finales de los años 2000, donde la comunicación está basada completamente en el protocolo IP, siendo un sistema y una red, que se alcanza gracias a la convergencia entre las redes de cables e inalámbricas. Proporciona unas velocidades mayores a las de 301 Mbit/s con un radio de 8 MHz. [3]



Figura 5 - Teléfonos móviles 4G [4]

- **Generación 5:** estas redes no estarán disponibles hasta el año 2020 pero las compañías más relevantes ya están llevando a cabo pruebas como por ejemplo *Ericsson*, que ha conseguido alcanzar velocidades de 5 Gbps reales de bajada en una prueba pre-estándar en Suecia. Esto significa aumentar 70 veces la velocidad que alcanzan en España las actuales redes 4G (75 Mbps aproximadamente alcanzada por el último servicio de cuarta generación 4G-LTE (*Long Term Evolution*)[5], aunque todavía no existe homogeneidad entre los estándares utilizados en las pruebas de las distintas compañías y no están aún definidos. Esta generación del futuro próximo será la contemporánea a las ciudades y coches inteligentes, las operaciones médicas a distancia o la comunicación entre máquinas mediante conexión de datos remota. [6]

2.2 Plataformas móviles

En los últimos años el crecimiento de las ventas de teléfonos móviles inteligentes o *smartphones* ha sido exponencial. Ya en 2011 logró superar las ventas de portátiles, PC's, tablets y netbooks llegando a los 500 millones de terminales. [7]

El uso de teléfonos inteligentes alcanza ya el 53,7% de la población española, según el informe de la ONTSI "La Sociedad en Red".[8]

Por ello, las aplicaciones para dispositivos móviles se han posicionado como un mercado emergente con grandes posibilidades.

El mercado actual de las aplicaciones móviles está liderado por Apple y Google, con sus sistemas iOS y Android y con App Store y Play Store como plataformas de venta, respectivamente, según el número de aplicaciones y negocio generado.

2.2.1 Android

Android es un sistema operativo basado en el kernel de Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, y también para relojes inteligentes, televisores y automóviles, inicialmente desarrollado por Android Inc., que Google respaldó económicamente y más tarde compró esta empresa en 2005. Android fue presentado en 2007 junto la fundación del Open Handset Alliance: un consorcio de compañías de hardware, software y telecomunicaciones para avanzar en los

estándares abiertos de los dispositivos móviles. El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008.[9]

Android se ha convertido en el sistema operativo móvil más utilizado en el mundo, con una ventaja considerable respecto a iOS y sobre todo, Windows Phone. Según las palabras de Sundar Pichai, jefe de Android en Google, en el evento Google I/O 2014 celebrado el pasado mes de junio, en estos momentos el sistema tiene más de 1.000 millones de usuarios activos cada mes. Teniendo en cuenta que en junio de 2013, el volumen de usuarios que usaban el sistema operativo de Google era 538 millones, estamos ante un incremento de 85% en tan solo 1 año. [10]

Una de las ventajas que tiene el sistema Android, aparte de que es un sistema de código abierto y no tiene restricciones en cuanto a dispositivos como pasa con su competidor más directo iOS, son los múltiples servicios que integra Google y ofrece de manera gratuita a los usuarios, de forma que con una simple cuenta de correo electrónico de Gmail podemos usarlos y combinar sus capacidades. Algunas de las herramientas son: Google Docs, Google Drive, Google Calendar, Google Maps, etc.

El entorno de desarrollo es independiente del sistema, otra ventaja también respecto a iOS, ya que la plataforma es Eclipse y el plugin de desarrollo Android SDK o Android Software Development Kit utilizando el lenguaje Java. También están disponibles otras herramientas de desarrollo como Android Studio que está siendo muy bien acogido por los desarrolladores y está desbancando a Eclipse como IDE de desarrollo.

2.2.1 iOS

iOS (iPhone OS), el sistema operativo desarrollado por Apple Inc. para iPhone, iPod Touch, iPad y Apple TV, es una evolución de Darwin BSD y por lo tanto un sistema operativo Unix. A diferencia de Android, iOS está desarrollado únicamente para unos pocos dispositivos diseñados por la propia Apple, constituyendo así un “ecosistema cerrado”. Sin embargo, de cara a los desarrolladores esta característica proporciona una ventaja, ya que las herramientas de diseño de aplicaciones y *frameworks* proporcionadas por Apple están optimizadas para dicho hardware, simplificando así las tareas de desarrollo y contribuyendo a garantizar la fiabilidad de las aplicaciones. Además, las aplicaciones para iOS se distribuyen por medio de la App Store, pasando estrictos controles de calidad antes de ser publicadas.[7]

2.2.3 Comparativa distintas plataformas

A continuación se muestran las ventas en España de dispositivos bajo los sistemas operativos más relevantes en el mercado móvil:

Spain	3 m/e Mar 2013	3 m/e Mar 2014	% pt. Change
Android	93.7	88.6	-5.1
BlackBerry	0.2	0.0	-0.2
iOS	3.1	7.6	4.5
Windows Phone	1.3	3.0	1.7
Other	1.8	0.8	-1.0



Figura 6 - Cuota de mercado de Sistemas móviles en España [11]

Como podemos apreciar en la Figura 6, el mercado en España es liderado por Android con un 88,6% de ventas y el sistema iOS se ha incrementado este último año en 4.5 puntos.

2.3 Cámara de Comercio de Málaga

Mi actividad profesional, desde que terminé mis estudios de Ingeniería Técnica en Informática de Gestión ha sido desarrollada en la Cámara Oficial de Comercio, Industria, Navegación y Servicios de la Provincia de Málaga. Mis tareas han estado relacionadas con la programación web principalmente.



Con el crecimiento de usuarios de dispositivos móviles y las necesidades que se van generando en la sociedad, se pensó en la creación de una app corporativa que mostrara los principales servicios y sirviera de contacto directo con la Cámara desde cualquier dispositivo móvil y en cualquier momento o lugar.

Como ya se ha descrito en la introducción, este es el objetivo del presente trabajo, por lo que se va a proceder a la descripción tanto de la corporación como de los servicios que presta para contextualizar la aplicación.

2.3.1 ¿Qué es la Cámara y quiénes forman parte?

La Cámara es una Corporación de Derecho Público que se configura legalmente como órgano consultivo y de colaboración con las Administraciones Públicas y que tiene como finalidad la representación, promoción y defensa de los intereses generales del comercio, la industria y la navegación, así como la prestación de servicios a las empresas.

Todas las personas físicas o jurídicas, nacionales o extranjeras, que ejerzan actividades comerciales, industriales o navieras y por tanto estén sujetos al Impuesto de Actividades Económicas.

2.3.2 ¿Cómo desarrolla su actuación la Cámara?

Las líneas maestras de actuación son:

- Consolidar su papel como prestadora de servicios, especialmente a las pymes.
- Promover vocaciones empresariales y la creación de empresas.
- Colaborar activamente en el proceso de simplificación administrativa en la creación de empresas.
- Incorporar a las pymes en la sociedad de la información.
- Promover la innovación entre las empresas.

- Potenciar y diversificar su actividad en la promoción del comercio exterior e internacionalización de las empresas.
- Recuperar e incentivar la Formación Profesional y Empresarial.
- Modernizar y apoyar la estructura y actividad comercial.

Para alcanzar estos objetivos, se trabaja sobre cuatro pilares: creación y consolidación de empresas, nuevas tecnologías, internacionalización, turismo, formación y empleo.

2.3.3 Servicios que ofrece la Cámara

Los servicios de la Cámara de Comercio de Málaga están orientados al crecimiento de las empresas y de los profesionales.

Los departamentos que la forman y sus funciones principales son:

- Formación, Empleo y Servicio a Pymes y Autónomos: en el que se encuentran programas de apoyo al demandante de empleo, agencia de colocación, cursos y másters especializados, asesoramiento a empresa, apoyo a emprendedores, vivero de empresas, programas de apoyo a la innovación, accesibilidad municipal, informes de mercado, listado de empresas, validación de certificados digitales *Camerfirma*, etc.
- Comercio Exterior y Turismo: consultoría y apoyo a la internacionalización, ayudas y asesoramiento en la promoción exterior, seminarios especializados del sector turismo, sede “Q” de calidad Turística en Málaga, informes y estudios, jornadas sobre internacionalización, etc.
- Secretaría General: licitaciones públicas, corte de arbitraje, servicio de mediación y asesoría jurídica.
- Relaciones Institucionales: comunicación, prensa y protocolo.

En este trabajo, se han tenido en cuenta cuatro grandes áreas de servicios, que incluyen los contenidos más relevantes al usuario de dispositivos móviles, los cuáles son:



Capítulo 3: Sistema de desarrollo: Android

3.1 Sistema Android

La arquitectura del sistema Android está organizada en capas de forma jerárquica, es decir, cada capa superior depende de la inferior para poder realizar su cometido, y a su vez ofrece sus servicios a las capas superiores. Es la que se muestra en la siguiente figura:

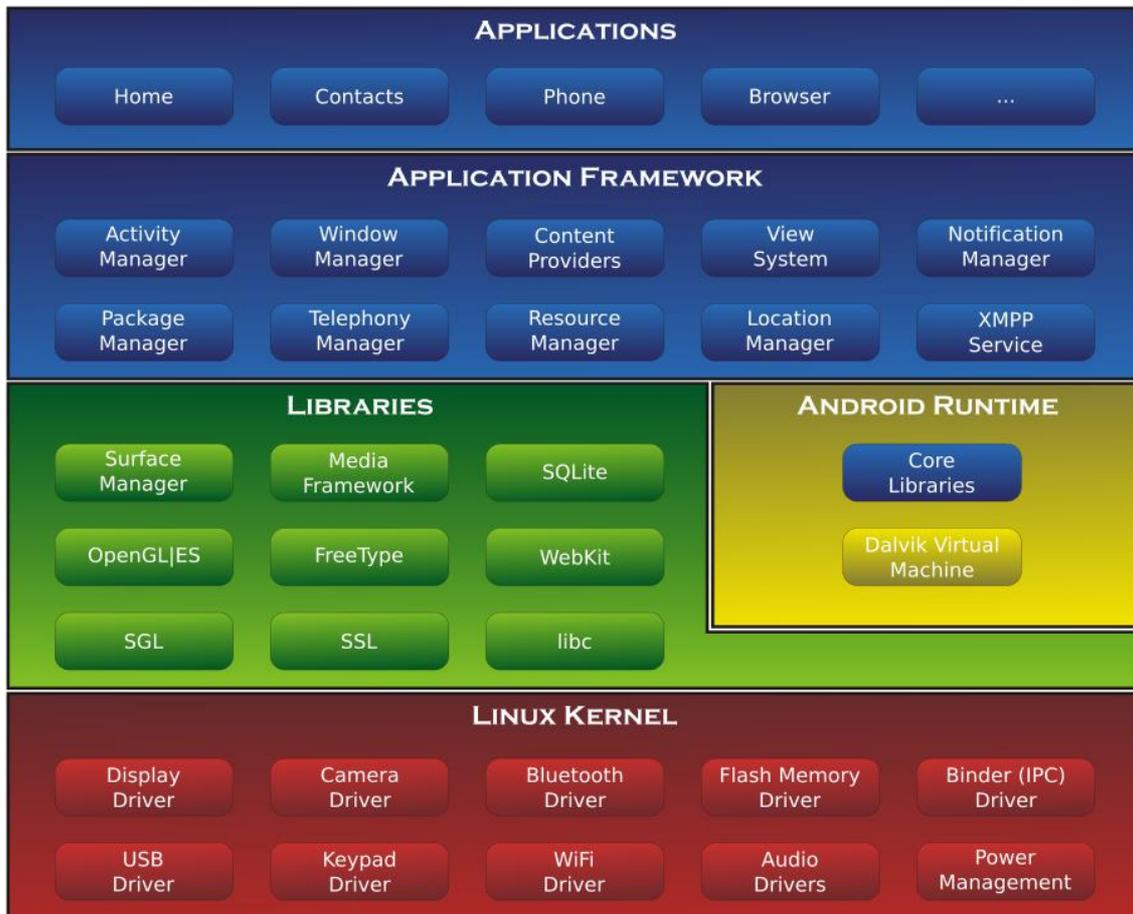


Figura 7 - Arquitectura Sistema Android [9]

Como se puede apreciar en la Figura 7, la arquitectura del sistema Android está compuesta por un kernel basado en Linux, con middleware, librerías y APIs en lenguaje C, y aplicaciones que se ejecutan en un framework que incluye librerías compatibles con Java.

Android usa una máquina virtual denominada Dalvik que optimiza los recursos ya que los dispositivos cuentan con prestaciones en cuanto a memoria, procesador y almacenamiento muy reducidas.

3.1.1 Características y especificaciones

A continuación veremos cuáles son las características del sistema Android [9] en la siguiente tabla:

Almacenamiento	La base de datos utilizada es SQLite
Conectividad	GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, HSDPA, HSPA+, NFC y WiMAX.GPRS,UMTS y HSDPA+.
Mensajería	SMS y MMS
Navegador web	El navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit, emparejado con el motor JavaScript V8 de Google Chrome.
Soporte de Java	El bytecode Java no es ejecutado, sino que primero se compila en un ejecutable Dalvik y corre en la Máquina Virtual Dalvik, que es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles.
Soporte multimedia	Formatos multimedia soportados: WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP.45
Soporte para streaming	Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player. Se planea el soporte de Microsoft Smooth Streaming con el port de Silverlight a Android. Adobe Flash HTTP Dynamic Streaming
Soporte para HW adicional	Cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D.
Play Store	Catálogo de aplicaciones gratuitas o de pago en el que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC.
Multitáctil	Android tiene soporte nativo para pantallas capacitivas con soporte multi-táctil.
Multitarea	Las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj.
Tethering	Android soporta tethering, que permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico.

3.1.2 Versiones de Android

El historial de versiones del sistema operativo Android se inició con el lanzamiento de Android beta en noviembre de 2007. La primera versión comercial (de prueba), Android 1.0, fue lanzada en septiembre de 2008.

Estas actualizaciones típicamente corrigen fallos de programa y agregan nuevas funcionalidades. Las distintas versiones de Android han sido desarrolladas bajo un nombre en clave y lanzamiento en orden alfabético:



Figura 8 - Versiones del Sistema Android

A continuación se muestra en la *Figura 9* la cuota de mercado de las distintas versiones del sistema Android en enero de 2014, por lo que se aprecia que la versión más extendida actualmente es la **Jelly Bean 4.1.x** con la **API 16**, con un 35.9% y es la **versión mínima** que requiere la aplicación desarrollada en este trabajo.

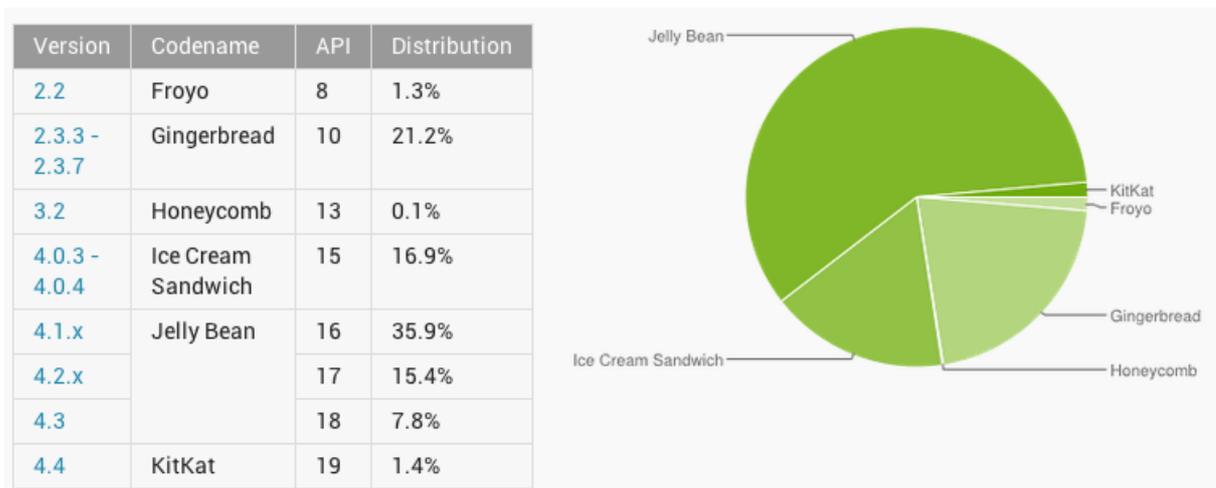


Figura 9 - Reparto de versiones en el mercado actual [12]

3.2 Herramientas: Android SDK y Eclipse

El entorno de desarrollo o IDE (*Integrated Development Environment*) utilizado para la implementación de la aplicación en este trabajo ha sido Eclipse junto con el Android SDK (Software Development Kit) que contiene las herramientas necesarias para el desarrollo de apps como son: depurador, emulador o simulador, documentación y ejemplos, y librerías.



La descarga es gratuita y se puede acceder desde <http://developer.android.com>. *Android Developer Tools* contiene las herramientas de desarrollo y depuración de cada versión de Android dependientes de cada plataforma y están actualizadas para la nueva versión que haya aparecido.

3.3 Componentes de una app Android

Las aplicaciones en Android están compuestas por los siguientes elementos:

- **Activity:** las actividades son el bloque encargado de la interfaz de la aplicación, presentando los elementos visuales e interaccionar con el usuario.

Esta clase tiene un ciclo de vida, como se muestra en la figura 10, que se puede gestionar a partir de los siguientes métodos:

- *onCreate()*: se ejecuta cuando la actividad es creada por primera vez y se utiliza sobre todo para inicializar las vistas, leer los parámetros de entrada y enlazarlos con los componentes, etc.
- *onRestart()*: se lanza cuando una actividad se ha parado y vuelve a ser iniciada.
- *onStart()*: se ejecuta cuando una actividad va a ser visible al usuario.
- *onResume()*: se ejecuta cuando el usuario empieza a interactuar con la actividad.
- *onPause ()*: se lanza cuando una actividad pasa a segundo plano pero aún no se ha destruido.
- *onStop()*: se ejecuta cuando la actividad ya no es visible al usuario.
- *onDestroy()*: es lo último que se ejecuta cuando una actividad se ha destruido.

- **Service:** implementan acciones que se ejecutan en segundo plano y no dependen de la actividad que realice el usuario con la aplicación.
- **BroadcastReceiver:** este componente responde a las notificaciones del sistema (batería, pantalla, descargas, etc.) y puede ser utilizado para notificar al usuario de los cambios sucedidos.
- **ContentProvider:** define los accesos a los datos almacenados por la aplicación para que otras aplicaciones puedan acceder o incluso modificarlos si el proveedor de contenidos lo permite.
- **Intent:** sirve para invocar componentes, como por ejemplo lanzar una nueva actividad.
- **Layout:** escrito en XML representa la visualización y estilo de los datos en cada actividad.
- **View:** son los elementos que componen la interfaz de usuario, como los botones o las entradas de texto.

Capítulo 4: Desarrollo de la aplicación

4.1 Análisis y Diseño de la aplicación

El análisis y diseño de la aplicación se ha realizado de forma conjunta para los dos TFG (la versión para el sistema iOS ha sido desarrollada por Francisco Andrade García) ya que las aplicaciones deben ser coherentes y similares en funcionalidad y aspecto, además de respetar la imagen corporativa de la Cámara de Comercio.

4.1.1 Requisitos funcionales

Para esta aplicación se han tomado una serie de requisitos iniciales necesarios para otorgar una funcionalidad que al usuario le resulte relevante y englobe los principales servicios o actividades de la corporación.

Se ha optado por una metodología en cascada y basada en prototipos para que la aplicación vaya evolucionando a lo largo del tiempo, utilizando la base y siguiendo las directrices de este primer desarrollo. El prototipo servirá de representación y permitirá a la Corporación supervisar y guiar el diseño y funciones principales, además de permitir hacerse una idea concreta de cómo va a resultar la aplicación final.

Como requisitos principales se definen los siguientes:

Requisitos

- R1 Mostrar listado de cursos**
- R2 Ampliar información de curso seleccionado**
- R3 Mostrar listado de misiones comerciales**
- R4 Ampliar información de misión comercial seleccionada**
- R5 Mostrar listado de programas y ayudas**
- R6 Ampliar información programa/ayuda seleccionado**
- R7 Mostrar listado de licitaciones**
- R8 Ampliar información sobre la licitación seleccionada**
- R9 Solicitar más información por teléfono y correo electrónico**
- R10 Mostrar ubicación de la Cámara**
- R11 Calcular ruta de llegada a la Cámara**

4.1.2 Metodología de diseño (cascada y prototipado)

La metodología que se ha determinado es espiral y basada en prototipos. Esta opción proporcionará la capacidad evolutiva a la aplicación y la prueba por parte de la Cámara de Comercio, con el fin de asegurar su correcta funcionalidad y cumplir con las directrices corporativas.

Este primer desarrollo seguirá evolucionando según las necesidades y sugerencias de los usuarios y de la Corporación.

El prototipo será probado y evaluado, y una vez que cumpla con los requisitos iniciales establecidos y descritos en este trabajo, se concluirá la app resultante, que será publicada en Play Store, y servirá de representación y que a su vez seguirá siendo evaluada por la corporación para poder establecer nuevas funcionalidades, para el próximo ciclo, además de permitir hacerse una idea concreta de cómo va a resultar la aplicación final.

4.2 Implementación

La implementación completa de la aplicación conlleva la interacción de tecnologías Web, herramientas de escritorio para los usuarios, y la aplicación móvil nativa.

4.2.1 Desarrollo web

Los datos requeridos en la aplicación se encuentran en dos servidores, bajo los dominios www.camaramalaga.com y www.formacioncamara.com, diseñados con los gestores de contenido **Typo3** y **Joomla**, respectivamente.

La mayoría del contenido se ha gestionado de manera estática en *HTML* a través de los gestores de contenido. Para crear un contenido dinámico que pueda actualizar los datos de la aplicación móvil a través de la cadena *JSON*, se han desarrollado en *PHP* las distintas aplicaciones web de las áreas servidoras de contenido. Estas aplicaciones cuentan con una base de datos en *MySQL* y páginas para el acceso y gestión de los datos.

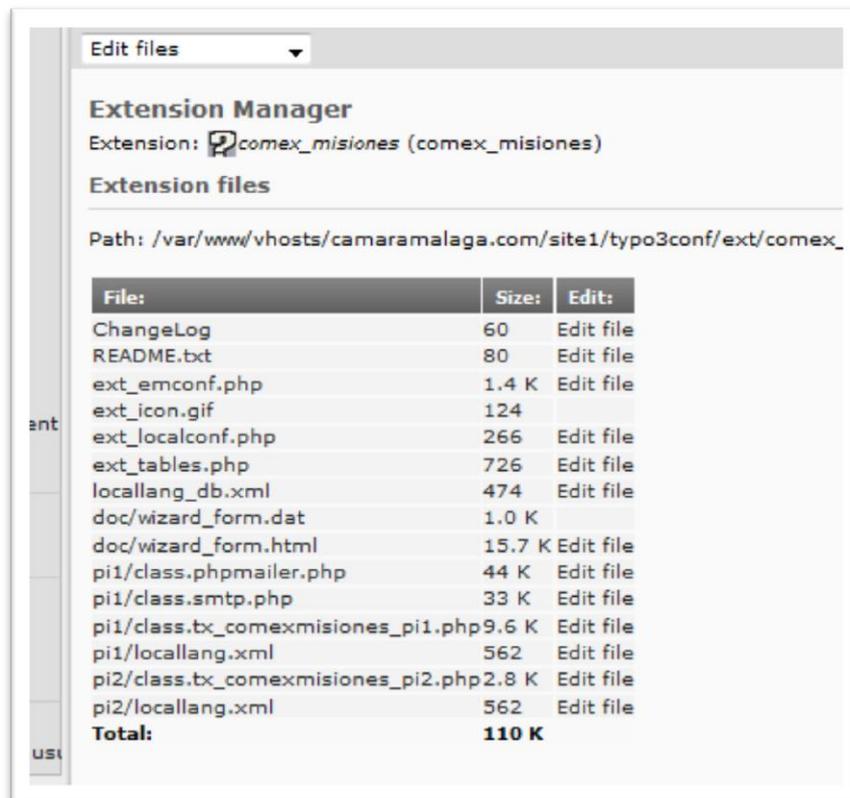


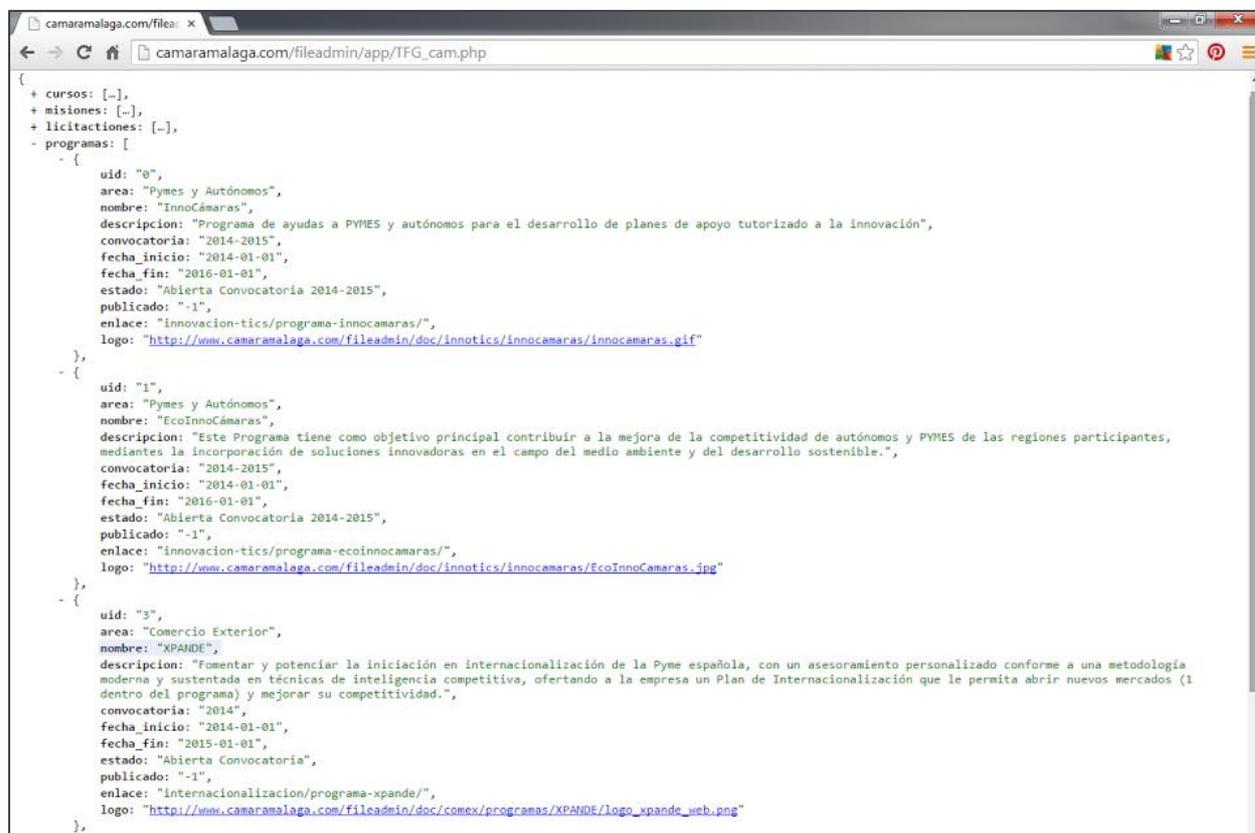
Figura 11 - Extensión en Typo3

La Figura 11 muestra la estructura de una extensión de gestión de misiones comerciales, para el gestor de contenidos *Typo3* y se encarga de almacenar los datos en la base de datos del *CMS* y mostrarlos en la web, de manera que los contenidos son actualizados y formateados dinámicamente mediante las consultas implementadas en los archivos *php*. El resultado en la web es el que muestra la Figura 12.

Misión comercial	
	Misión comercial ESTADOS UNIDOS DE AMÉRICA
Fecha Inicio:	20 Septiembre
Fecha Fin:	28 Septiembre
Sector:	TICs y Agroalimentario
Ciudad:	Miami y San Francisco

Figura 12 - Ejemplo de misión comercial

Para que la aplicación pueda leer los datos se ha creado un servicio web que realiza las consultas sobre los datos de los sitios web y los transcribe a una cadena *JSON*, el resultado de esto es lo que se puede apreciar en la Figura 13.



```
{
  + cursos: [-],
  + misiones: [-],
  + licitaciones: [-],
  - programas: [
    - {
      uid: "0",
      area: "Pymes y Autónomos",
      nombre: "InnoCámaras",
      descripción: "Programa de ayudas a PYMES y autónomos para el desarrollo de planes de apoyo tutorizado a la innovación",
      convocatoria: "2014-2015",
      fecha_inicio: "2014-01-01",
      fecha_fin: "2016-01-01",
      estado: "Abierta Convocatoria 2014-2015",
      publicado: "-1",
      enlace: "innovacion-tics/programa-innocamaras/",
      logo: "http://www.camaramalaga.com/fileadmin/doc/innotics/innocamaras/innocamaras.gif"
    },
    - {
      uid: "1",
      area: "Pymes y Autónomos",
      nombre: "EcoInnoCámaras",
      descripción: "Este Programa tiene como objetivo principal contribuir a la mejora de la competitividad de autónomos y PYMES de las regiones participantes, mediante la incorporación de soluciones innovadoras en el campo del medio ambiente y del desarrollo sostenible.",
      convocatoria: "2014-2015",
      fecha_inicio: "2014-01-01",
      fecha_fin: "2016-01-01",
      estado: "Abierta Convocatoria 2014-2015",
      publicado: "-1",
      enlace: "innovacion-tics/programa-ecoinnocamaras/",
      logo: "http://www.camaramalaga.com/fileadmin/doc/innotics/innocamaras/EcoInnoCamaras.jpg"
    },
    - {
      uid: "3",
      area: "Comercio Exterior",
      nombre: "XPANDE",
      descripción: "Fomentar y potenciar la iniciación en internacionalización de la Pyme española, con un asesoramiento personalizado conforme a una metodología moderna y sustentada en técnicas de inteligencia competitiva, ofertando a la empresa un Plan de Internacionalización que le permita abrir nuevos mercados (1 dentro del programa) y mejorar su competitividad.",
      convocatoria: "2014",
      fecha_inicio: "2014-01-01",
      fecha_fin: "2015-01-01",
      estado: "Abierta Convocatoria",
      publicado: "-1",
      enlace: "internacionalizacion/programa-xpande/",
      logo: "http://www.camaramalaga.com/fileadmin/doc/comex/programas/XPANDE/logo_xpande_wcb.png"
    }
  ]
}
```

Figura 13 - Cadena JSON

4.2.2 Desarrollo de herramientas locales

Aunque no es necesario para la aplicación, se han desarrollado algunas herramientas locales, a través del gestor de bases de datos Microsoft Access, para que los usuarios pudieran gestionar los datos.

Para ello, se creó una vinculación de datos externa mediante el conector *ODBC* de *Sun* para *MySQL*, ya que es un driver estándar que conecta con un servidor desde plataformas *Windows*, *Linux* y *Mac OS X*.

En la Figura 14 se muestra la configuración del conector para el servidor de www.camaramalaga.com.

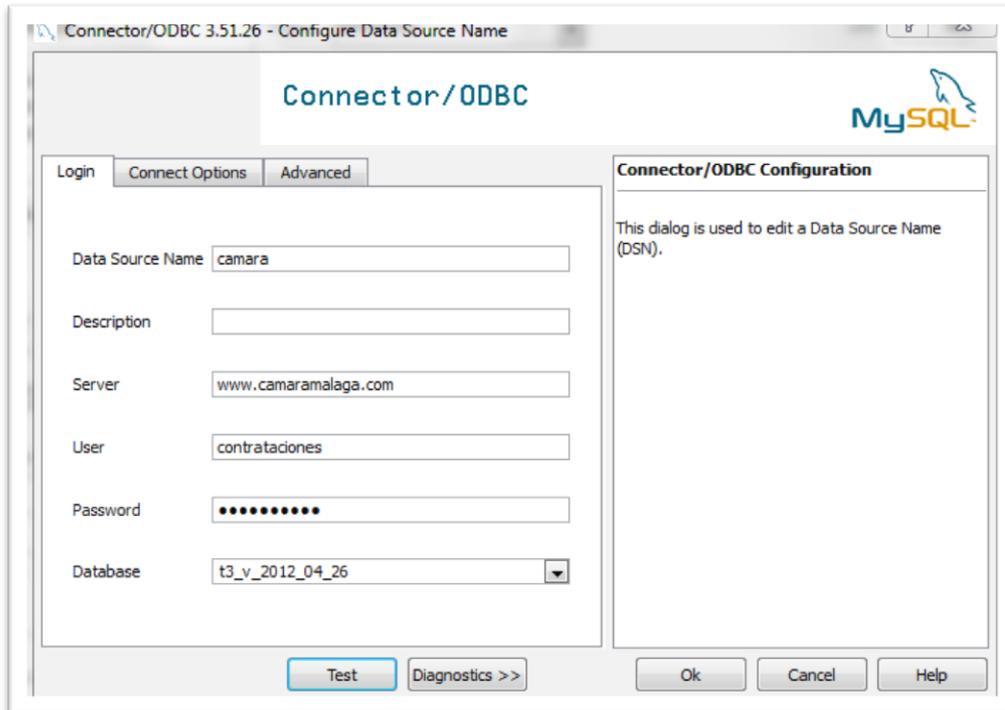


Figura 14 - Conector ODBC - MySql

En la Figura 15 podemos ver la herramienta desarrollada en Microsoft Access para gestionar los datos de una acción promocional perteneciente al área de Internacionalización:

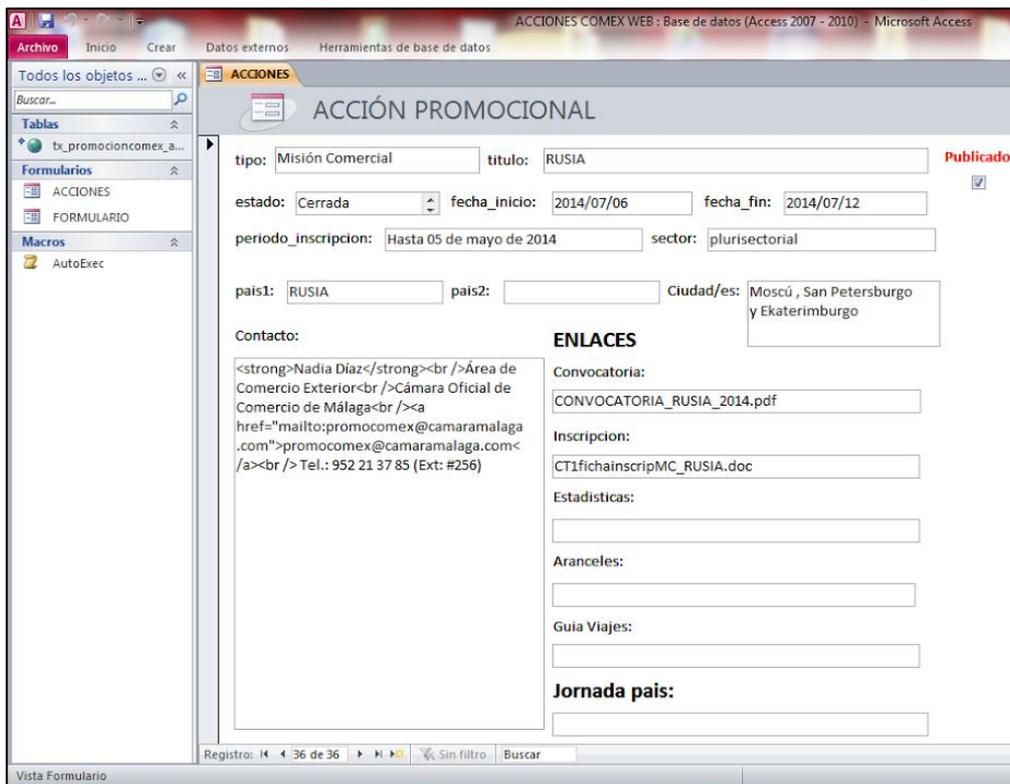


Figura 15 - Base de datos Comercio Exterior

4.2.3 Desarrollo de la app

La aplicación se ha desarrollado en Eclipse ADT utilizando *Java* como lenguaje de programación y *SQLite* para la base de datos.

Se ha establecido una estructura en paquetes siguiendo el patrón de diseño MVC (Modelo-Vista-Controlador) según las funciones de las clases que contiene y son los siguientes:

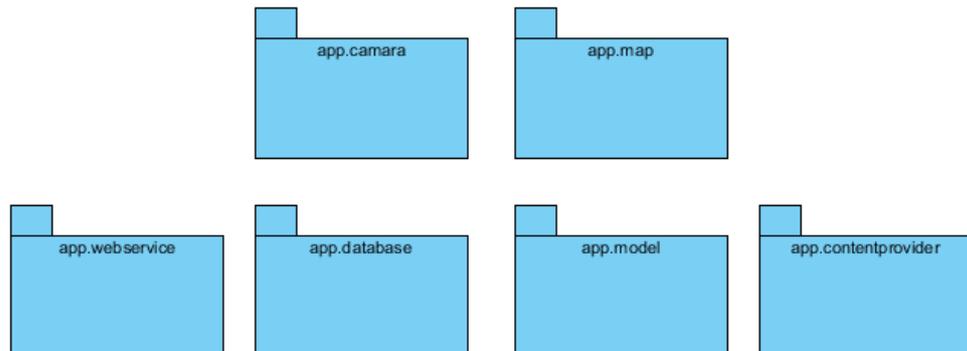


Figura 16 - Paquetes de la aplicación

- **app.camara:** este paquete contiene las actividades de la aplicación.
- **app.map:** contiene las clases para gestionar la navegación y rutas de la API de Google Maps para Android.
- **app.webservice:** contiene las clases que se conectan al servidor, descargan la cadena JSON y la parsean creando los objetos del modelo.
- **app.database:** contiene clases que gestionan las conexiones y accesos a la base de datos en SQLite, su ubicación, creación y borrado.
- **app.model:** contiene las clases del modelo de negocio.
- **app.contentprovider:** será el contenedor de las clases para la gestión de datos y acceso a ellos desde otras aplicaciones.

4.2.3.1 Casos de uso y clases

Los casos de uso se han realizado en base a los requisitos funcionales de la aplicación y se agrupan en tres áreas: mostrar información, mostrar mapa y contactar.

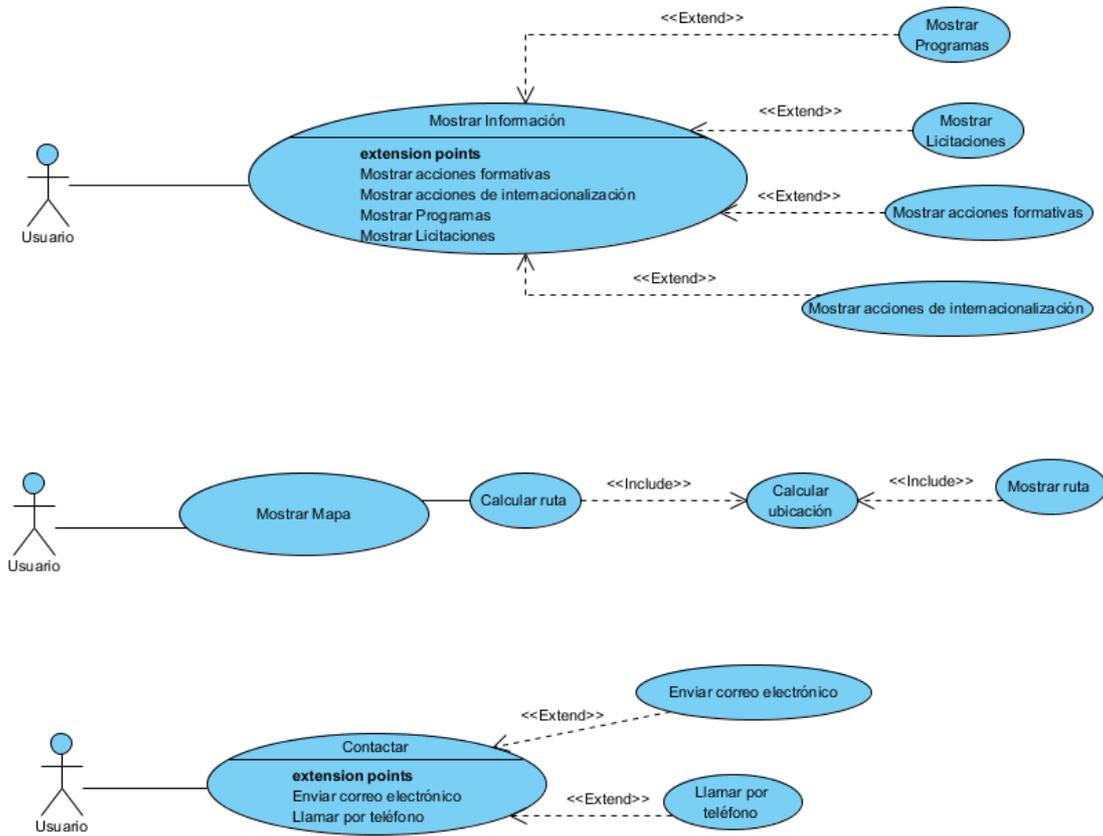


Figura 17 - Casos de uso

Las clases del modelo de negocio, que se encuentran en el paquete **app.model**, representan a los cuatro nodos de información que muestra la aplicación (Curso, Licitación, Programa y Acción) y proporcionan los métodos *getter* and *setter* para el acceso a sus propiedades.

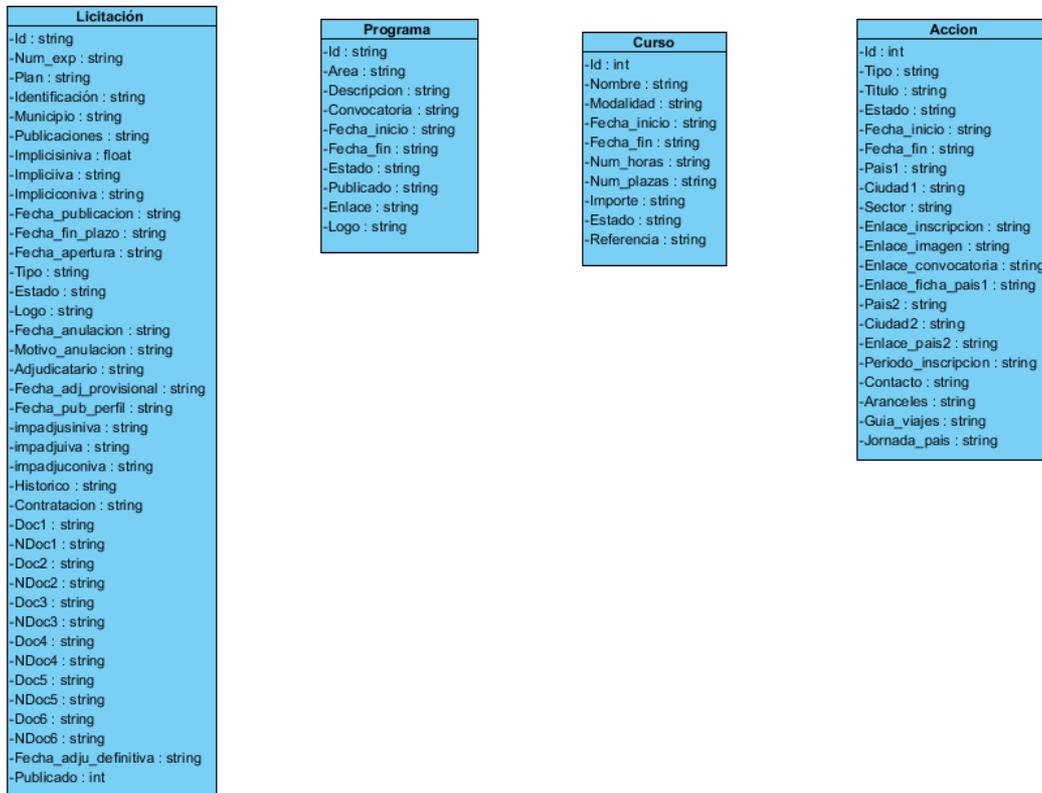


Figura 18 - Clases del modelo de negocio

Como representación del paquete **app.database** la clase más relevante es la encargada de la gestión de la base de datos y su especificación es la siguiente:

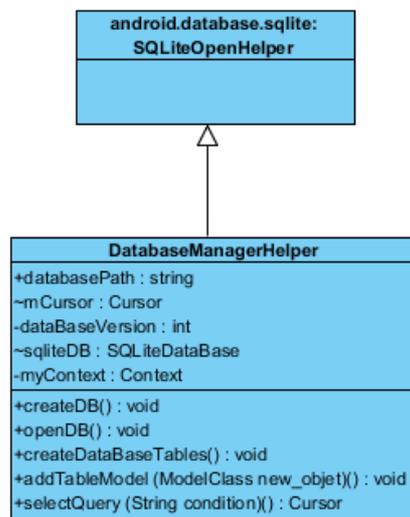


Figura 19 - Clase DatabaseManagerHelper

La clase encargada del tratamiento de la cadena *JSON* y el proceso de transformación a clase del modelo y guardado en la base de datos se encuentra en el paquete **app.webservice** y esta es su especificación:

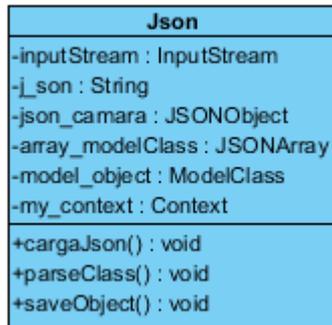


Figura 20 - Clase JSON

El funcionamiento de la aplicación es similar en todas las operaciones ya que se ha seguido el mismo patrón **listado-detalle** en todas las actividades, por lo que el siguiente diagrama de secuencia se toma como referencia, y simula el proceso de la petición del usuario sobre la información de acciones formativas de la Cámara de Comercio de Málaga en la aplicación, pulsando el botón "Formación":

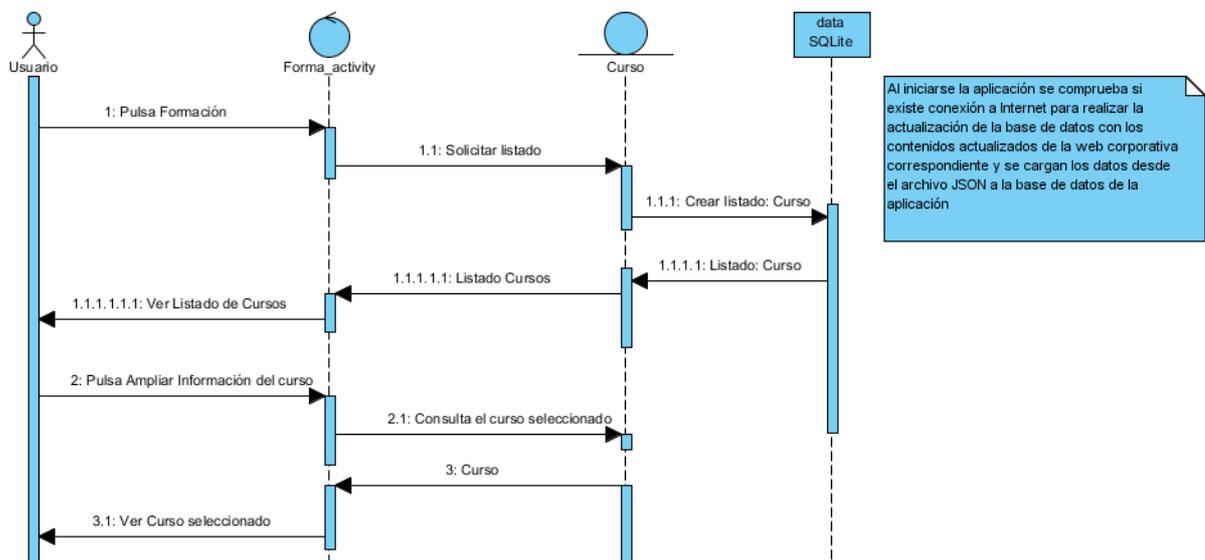


Figura 21 - Diagrama de secuencia

4.2.3.2 Inicio de la aplicación y precarga de datos

La app se inicia con una *Splash Activity*, es decir, una pantalla inicial que muestra el logo de la corporación y que permite iniciar el proceso de procesamiento de la cadena Json del servidor web.

Para ello se ha implementado una clase llamada **JSON** dentro del paquete *app.webservices* que se encarga de este cometido.

```
private void cargaJSON(){
    try {
        // defaultHttpClient
        DefaultHttpClient httpClient = new DefaultHttpClient();
        HttpPost httpPost = new HttpPost(WEB_SERVICE);

        HttpResponse httpResponse = httpClient.execute(httpPost);

        HttpEntity httpEntity = httpResponse.getEntity();

        is = httpEntity.getContent();
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

    try {

        BufferedReader reader = new BufferedReader(new InputStreamReader(is, "UTF-8"), 8);
        StringBuilder sb = new StringBuilder();
        String line = null;

        while ((line = reader.readLine()) != null) {

            sb.append(line + "\n");

        }

        is.close();
        j_son = sb.toString();
    } catch (Exception e) {

        Log.e("Buffer Error", "Error converting result " + e.toString());
    }

    try {

        json_camara = new JSONObject (j_son);
    } catch (JSONException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    // fin: se ha guardado el array de arrays en la variable de texto j_son

    this.parseJSON();
}
}
```

Figura 22 - Carga JSON

La cadena se parsea y se crean objetos del modelo que son insertados en la base de datos de la aplicación.

```
private void parseCurso(){
    try {

        array_curso = json_camara.getJSONArray("cursos");

        if (array_curso != null) tabla = 0;

        for (int j = 0; j < array_curso.length(); j++) {
            JSONObject jso = array_curso.getJSONObject(j);

            curso_js.setEstado(jso.getString(TAG_ESTADO_CURSO));
            curso_js.setFecha_fin(jso.getString(TAG_FECHA_FIN_CURSO));
            curso_js.setFecha_inicio(jso.getString(TAG_FECHA_INICIO_CURSO));
            curso_js.setIdcurso(jso.getInt(TAG_ID_CURSO));
            curso_js.setImporte(jso.getString(TAG_IMPORTE_CURSO));
            curso_js.setTipo(jso.getString(TAG_TIPO_CURSO));
            curso_js.setModalidad(jso.getString(TAG_MODALIDAD_CURSO));
            curso_js.setNombre(jso.getString(TAG_NOMBRE_CURSO));
            curso_js.setNum_horas(jso.getString(TAG_NUM_HORAS_CURSO));
            curso_js.setNum_plazas(jso.getString(TAG_NUM_PLAZAS_CURSO));
            curso_js.setReferencia(jso.getString(TAG_REFERENCIA_CURSO));

            this.guardaCurso();
        }

    } catch (JSONException e) {
        e.printStackTrace();
    }
}
```

Figura 23 - Parse JSON-Curso

Para ello, la clase **DBManageHelper**, dentro del paquete *app.database*, proporciona los métodos necesarios para creación, actualización y borrado de datos. La clase **DBConstant** de este mismo paquete se encarga de proveer los nombres de los campos mediante clases que contienen los valores reales que se encontraran en la cadena *JSON*, de esta forma, se mantiene la integridad y coherencia en todo el código de la aplicación, ya que se utilizan esas constantes en lugar de texto o variables locales ante cualquier modificación externa. Estas clases implementan el interface de Android *BaseColumns*, este es el ejemplo para la clase **Curso**:

```

public static final class Curso implements BaseColumns {

    private Curso() {}

    public static final String CURSO_TABLA_NOMBRE = "curso";
    public static final String _ID = "_id";
    public static final String ID_CURSO= "id";
    public static final String NOMBRE_CURSO = "nombre";
    public static final String TIPO_CURSO = "tipo";
    public static final String MODALIDAD_CURSO = "modalidad";
    public static final String FECHA_INICIO_CURSO = "fecha_inicio";
    public static final String FECHA_FIN_CURSO = "fecha_fin";
    public static final String NUM_HORAS_CURSO = "num_horas";
    public static final String NUM_PLAZAS_CURSO = "num_plazas";
    public static final String IMPORTE_CURSO = "importe";
    public static final String ESTADO_CURSO = "estado";
    public static final String REFERENCIA_CURSO = "referencia";

}

```

Figura 24 - Clase Curso BaseColumns

El guardado de los datos se realiza de la siguiente forma desde la clase *DBManagerHelper*:

```

public void addCurso(Curso nuevo_curso){

    ContentValues registro = new ContentValues();

    registro.put(DBConstant.Curso.ID_CURSO, nuevo_curso.getIdcurso());
    registro.put(DBConstant.Curso.NOMBRE_CURSO, nuevo_curso.getNombre());
    registro.put(DBConstant.Curso.TIPO_CURSO, nuevo_curso.getTipo());
    registro.put(DBConstant.Curso.MODALIDAD_CURSO, nuevo_curso.getModalidad());
    registro.put(DBConstant.Curso.FECHA_INICIO_CURSO, nuevo_curso.getFecha_inicio());
    registro.put(DBConstant.Curso.FECHA_FIN_CURSO, nuevo_curso.getFecha_fin());
    registro.put(DBConstant.Curso.NUM_HORAS_CURSO, nuevo_curso.getNum_horas());
    registro.put(DBConstant.Curso.NUM_PLAZAS_CURSO, nuevo_curso.getNum_plazas());
    registro.put(DBConstant.Curso.IMPORTE_CURSO, nuevo_curso.getImporte());
    registro.put(DBConstant.Curso.ESTADO_CURSO, nuevo_curso.getEstado());
    registro.put(DBConstant.Curso.REFERENCIA_CURSO, nuevo_curso.getReferencia());

    db_curso_id = sqliteDB.insertOrThrow(DBConstant.Curso.CURSO_TABLA_NOMBRE, null, registro);
    Log.i(DEBUG_TAG, "Añadido Curso: " + nuevo_curso.getReferencia() + "(ID = " + db_curso_id + ")");
    close();
}

```

Figura 25 - Guardado de datos en SQLite

4.2.3.3 Estructura de la app

La estructura que sigue la aplicación en cada apartado es común a todas las partes que la componen (Formación [Curso], Programas y ayudas [Programa], Comercio Exterior y Turismo [Acción], Licitaciones [Licitación]).

ListView y Adapter

Todas las Activities principales de estos apartados se componen de un elemento *ListView* que se rellena a través del *Adapter* correspondiente en cada caso.

Al pulsar el botón de “Consultar”, se abre una nueva *Activity* de detalle, que muestra una información más amplia del objeto que corresponde al ítem seleccionado en ese momento.

El proceso de relleno del *ListView* se realiza a través de la clase *Adapter* y los métodos *bindView* y *newView*, que hereda de la clase *CursorAdapter*, para cada clase del modelo.

El método *bindView* se encarga de asignar los valores a los componentes de cada ítem de la lista y en método *newView* asigna el layout correspondiente a cada ítem.

```
@Override
public void bindView(View view, final Context context, Cursor cursor) {
    // TODO Auto-generated method stub

    this.tvNombreCurso = (TextView) view.findViewById(R.id.tvNombreCurso);
    this.tvModalidadCurso = (TextView) view.findViewById(R.id.tvModalidadCurso);
    this.tvTipoCurso = (TextView) view.findViewById(R.id.tvTipoCurso);
    this.btConsultar = (Button) view.findViewById(R.id.btDetalleCurso);

    forma_curso = new Curso();
    btConsultar.setTag(forma_curso);
    forma_curso.setNombre(cursor.getString(cursor.getColumnIndex(DBConstant.Curso.NOMBRE_CURSO)));

    this.tvNombreCurso.setText(cursor.getString(cursor.getColumnIndex(DBConstant.Curso.NOMBRE_CURSO)));
    this.tvModalidadCurso.setText(cursor.getString(cursor.getColumnIndex(DBConstant.Curso.MODALIDAD_CURSO)));
    this.tvTipoCurso.setText(cursor.getString(cursor.getColumnIndex(DBConstant.Curso.TIPO_CURSO)));
    //final String ref = cursor.getString(cursor.getColumnIndex(DBConstant.Curso.REFERENCIA_CURSO));
    //btConsultar.setTag(forma_curso);
    this.nombreCurso = tvNombreCurso.getText().toString();
    btConsultar.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            Curso ncurso = new Curso();
            ncurso = (Curso) v.getTag();

            Bundle bundle = new Bundle();
            btConsultar.setTag(forma_curso);
            bundle.putString("Nombre", ncurso.getNombre());
            cierraBaseDatos();

            Intent intent = new Intent(context, CursoActivity.class);
            intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            intent.putExtras(bundle);

            context.startActivity(intent);
        }
    });

    this.cierraBaseDatos();
}
```

Figura 26 - Cursor Adapter

Almacenamiento y carga de imágenes

Las imágenes son descargadas y almacenadas en una carpeta de recursos de la aplicación para evitar esperas molestas y no agradables visualmente al usuario ya que cada elemento del *ListView* se genera en el momento de visualización en la pantalla.

El proceso se ha llevado a cabo dentro de la clase `app/webservice/Json.java` de la siguiente forma:

```
private void guardarImagenAccion(String img) {

    String url_img = this.URL_IMAGEN_ACCION + img;
    Bitmap imagen = null;

    try {

        URL imageUrl = new URL(url_img);
        HttpURLConnection conn = (HttpURLConnection) imageUrl.openConnection();
        conn.connect();
        imagen = BitmapFactory.decodeStream(conn.getInputStream());

    }catch(IOException ex){

        ex.printStackTrace();
    }

    ContextWrapper cw = new ContextWrapper(my_context);
    File dirImages = cw.getDir("imagenes", my_context.MODE_PRIVATE);
    File myPath = new File(dirImages, img+".PNG");

    FileOutputStream fos = null;

    try {

        fos = new FileOutputStream(myPath);
        imagen.compress(Bitmap.CompressFormat.PNG, 10, fos);
        fos.flush();

    }catch (FileNotFoundException ex){
        ex.printStackTrace();
    }catch (IOException ex){
        ex.printStackTrace();
    }

    path_img_accion = myPath.getAbsolutePath();

}
```

Figura 27 - Descarga y guardado de imágenes

Google Maps API y Navegación

Para la implementación del mapa que posiciona la localización de la Cámara de Comercio de Málaga y calcula la ruta desde la posición del usuario, se ha contado con una librería que contiene las clases necesarias para la navegación en el mapa. Las clases proporcionadas son:

- ❖ Navigator.java
- ❖ Directions.java
- ❖ Route.java
- ❖ Legs.java
- ❖ Steps.java

El primer paso es registrarse en Google y obtener una clave para la API de *Google Maps*, en este caso se ha utilizado la versión 2. Una vez obtenida, se incluye en el manifiesto de la aplicación.

El mapa se incluye en el *layout* **mapa_activity.xml** de la siguiente manera:

```
<fragment
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginRight="0dp"
    android:layout_marginTop="110dp"
    class="com.google.android.gms.maps.SupportMapFragment" />
```

Y en la clase **Mapa_Activity.java**, que hereda de *FragmentActivity*, se realiza la carga del contenido posicionando la ubicación de la Cámara.

```
private void crearMapa(){

    FragmentManager fmanager = getSupportFragmentManager();
    Fragment fragment = fmanager.findFragmentById(R.id.map);
    SupportMapFragment supportmapfragment = (SupportMapFragment)fragment;

    mapa = supportmapfragment.getMap();
    mapa.addMarker(new MarkerOptions().position(LAT_LNG_CAMARA).title("Camara"));
    mapa.animateCamera(CameraUpdateFactory.newLatLngZoom(LAT_LNG_CAMARA, 17));
}
```

Figura 28 - Método crearMapa

El uso de la librería realiza el trazado de la ruta en el mapa, indicando un origen y un destino y proporciona un método que permite almacenar los pasos, que posteriormente se usarán para indicarle al usuario la ruta por etapas:

```
public void calcularRuta(View v){  
  
    LatLng position = this.getCurrentLocation();  
  
    nav = new Navigator(mapa,position,LAT_LNG_CAMARA);  
    nav.setOnPathSetListener(this);  
    nav.findDirections(false);  
  
}
```

Figura 29 - Método calcularRuta

Al calcularse la ruta a través de la clase *Navigator.java*, el método *onPathSetListener*, permite la obtención de los valores de los pasos, distancia y duración de la misma. El parámetro *directions* contiene los datos parseados a través de la cadena *JSON* que proporciona la API de Google.

```
@Override  
public void onPathSetListener(Directions directions) {  
  
    rutas = new ArrayList<Route>();  
    rutas = directions.getRoutes();  
  
    btEtapas.setVisibility(View.VISIBLE);  
    tvDistancia.setText(rutas.get(0).getLegs().get(0).getLegDistance());  
    tvTiempo.setText(rutas.get(0).getLegs().get(0).getLegDuration());  
  
    pasos = rutas.get(0).getLegs().get(0).getSteps();  
  
    btEtapas.setOnClickListener(new View.OnClickListener() {  
  
        @Override  
        public void onClick(View v) {  
  
            Intent intent = new Intent(c, EtapasActivity.class);  
            intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);  
  
            c.startActivity(intent);  
  
        }  
    });  
}
```

Figura 30 - Método onPathSetListener

Android Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="app.camara"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="16"
        android:targetSdkVersion="21" />

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

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.CALL_PHONE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
    <!--
    The following two permissions are not required to use
    Google Maps Android API v2, but are recommended.
    -->
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="app.camara.permission.MAPS_RECEIVE" />

    <permission
        android:name="app.camara.permission.MAPS_RECEIVE"
        android:protectionLevel="signature" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppBaseTheme" >
        <meta-data
            android:name="com.google.android.maps.v2.API_KEY"
            android:value="AIzaSyAZBv_jVkBMK3EsymMhw-jidM2mAeXi6D4" />
        <meta-data
            android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />

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

        <activity
            android:name=".CamaraActivity"
            android:background="@android:color/white"
            android:label="@string/app_name"
            android:theme="@android:style/Theme.NoTitleBar.Fullscreen" >
        </activity>
        <activity
            android:name=".FormaActivity"
            android:label="@string/volver"
            android:parentActivityName=".CamaraActivity" >
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value="app.camara.CamaraActivity" />
        </activity>
        <activity
            android:name=".EtapasItemActivity"
            android:label="@string/title_activity_etapas_item" >
        </activity>
        <activity
            android:name=".EtapasActivity"
            android:label="@string/volver"
            android:parentActivityName=".EtapasActivity" >
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value="app.camara.EtapasActivity" />
        </activity>
    </application>
</manifest>
```

4.3 Pruebas con el emulador y dispositivo real

4.3.1 Emulador



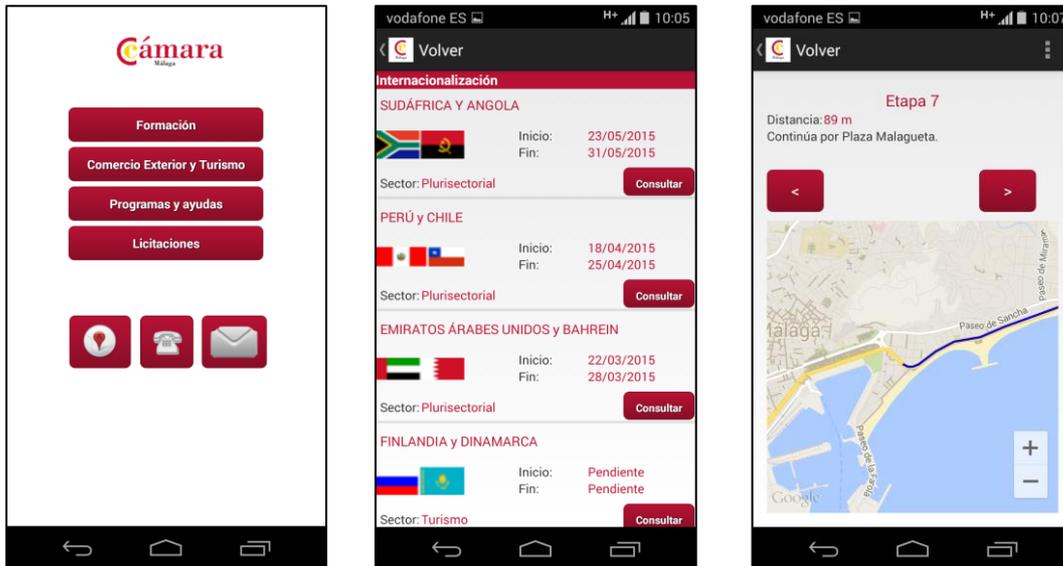
Device: Galaxy Nexus (4.65", 720x1280: xhdpi)
Target: Google APIs (Google Inc.) – API Level 17
CPU/ABI: Google APIs Intel Atom (x86)
RAM: 512 MB - **VM Heap:** 32 MB
Internal Storage: 200 MB
SD Card: 100 MB



4.3.2 Dispositivo real

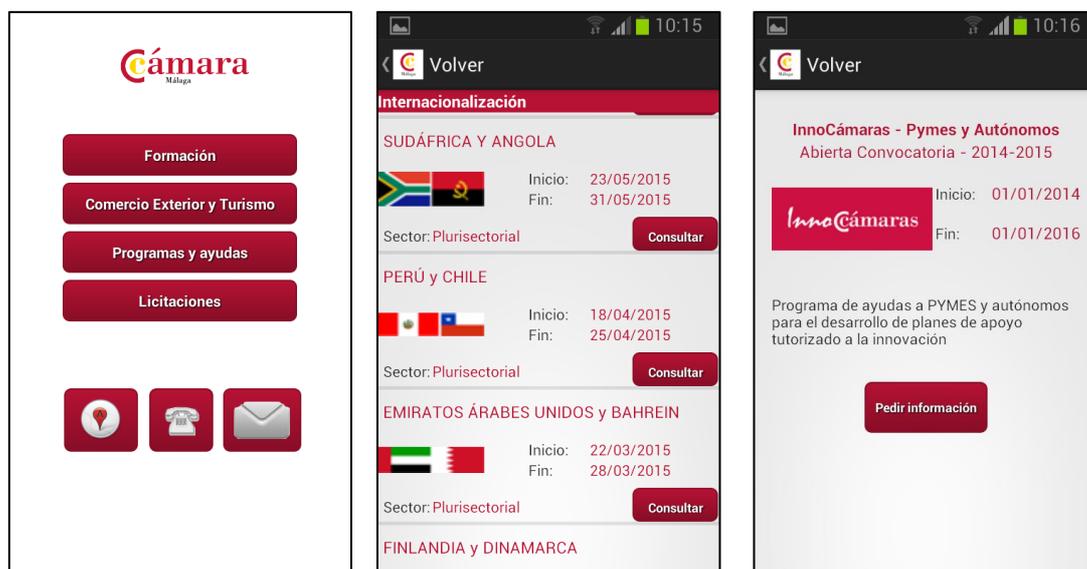
Para las pruebas en dispositivos reales, se ha utilizado distintos tamaños y sistemas para poder comprobar el funcionamiento de manera más fehaciente.

Dispositivo 1: Motorola G 4.5”, Sistema Android 4.4.4, API Level: 19



En el dispositivo 1 se ha comprobado que los contenidos se muestran de manera correcta y bien ajustados al tamaño.

Dispositivo 2: Samsung ACE 2 3.8" Android 4.1.2 API Level: 16



En el dispositivo 2 se ha detectado que la versión del sistema no corresponde a la mínima que exige la aplicación. Aún así, se ha comprobado que el usuario puede usar de manera correcta todas las funcionalidades.

Conclusiones de las pruebas realizadas

Al realizar las distintas pruebas se ha determinado que el permiso para que Google determine nuestra ubicación, (opción de Servicios de ubicación, que aparece en los ajustes del sistema), mejora de manera sustancial la funcionalidad de determinación de la ubicación, al realizar el cálculo de ruta. Esta mejora supone un incremento de la velocidad de geoposicionamiento del dispositivo.

Capítulo 5: Conclusiones y trabajos futuros

5.1 Conclusión del TFG

El principal reto para mí en este TFG ha sido el sistema elegido y la plataforma de desarrollo. Mi experiencia en desarrollo de aplicaciones móviles se reducía a un curso de desarrollo para iOS6, por lo que no ha sido fácil adaptarme a la estructura de las aplicaciones en Android y a su línea de desarrollo.

En cuanto al lenguaje *Java*, sí que puedo decir, que me ha resultado más fácil que el *ObjectiveC*, puesto que ya lo había estudiado en mi titulación.

Pienso que el desarrollo de aplicaciones móviles es actualmente un área en auge de expectativas profesionales en nuestro sector y, sobre todo, de crecimiento del autoempleo, ya que son productos que cuentan con importante valoración en el mercado y cada vez más empresas se dan cuenta del beneficio que pueden obtener con la implementación de una app para su negocio.

5.2 Futuras ampliaciones

La aplicación desarrollada en este trabajo muestra alguna de la información que ofrece la web de la Cámara de Comercio. En esta primera versión, el usuario no puede interactuar con la aplicación por lo que sería interesante, para un futuro, añadir una gestión de cuenta de usuario.

De esta forma, el usuario podría inscribirse en las jornadas, acciones promocionales internacionales, cursos, etc a través del móvil mediante su registro e identificación correspondiente. Además, la inclusión de un calendario o agenda, permitiría la visualización de los eventos a los que está inscrito, recibir alertas, cancelar asistencias, etc.

REFERENCIAS BIBLIOGRÁFICAS

Evolución de la telefonía móvil y Plataformas Móviles:

- [1] Evolución de los dispositivos móviles
<http://evoluciondedispositivosmoviles.blogspot.com.es/>
- [2] Historia y evolución de los dispositivos móviles
<http://jcecembmob.blogspot.com.es/2012/02/mde-lab-entrada-1.html>
- [3] Telefonía móvil 4G
http://es.wikipedia.org/wiki/Telefonía_móvil_4G
- [4] MovilZona
<http://www.movilzona.es>
- [5] Samsung espera tener listo el servicio móvil de 5G en 2020
<http://www.vanguardia.com/actualidad/tecnologia/207978-samsung-espera-tener-listo-el-servicio-movil-de-quinta-generacion-5g-en>
- [6] El futuro es la velocidad supersónica del 5G
http://www.tecnoplora.com/empresas/futuro-velocidad-supersonica_2014072300141.html
- [7] El éxito de las plataformas de aplicaciones móviles
<http://www.gradiant.org/actualidad/noticias/606-el-exito-de-las-plataformas-de-aplicaciones-moviles.html>
- [8] Observatorio Nacional de las Telecomunicaciones y la SI
<http://www.ontsi.red.es/>

Sistema Operativo Android

- [9] Android
<http://es.wikipedia.org/wiki/Android>
- [10] Android ya tiene más de 1.000 millones de usuarios activos cada mes
<http://www.genbeta.com/movil/android-tiene-ya-mas-de-1-000-millones-de-usuarios-activos-cada-mes>
- [11] Cuota de mercado de sistemas móviles en España
<http://www.xatakamovil.com/mercado/ios-y-windows-phone-le-quitan-cuota-de-mercado-a-android-en-espana-con-mucha-timidez>

[12] Android Distribution Numbers Update for January 2014

<http://www.droid-life.com/2014/01/11/android-distribution-numbers-update-for-january-2014-jelly-bean-closing-in-on-60/>

[13] Android developer.

<http://developer.android.com>

Implementación

<http://stackoverflow.com/>

<https://developers.google.com>

<https://github.com/tyczj/MapNavigator>