



BCN Bicing Info – Una aplicació Android que se nutre de Open Data para presentar, sirviéndose de Google Maps, información en tiempo real de utilidad para el usuario de Bicing, el servicio público de bicicletas compartidas de la ciudad de Barcelona

TRABAJO FINAL DE CARRERA

MEMORIA

Área de TFC: Aplicaciones Multimedia de Nueva Generación

Alumno: **José Miguel Gimeno García**

Plan de estudios: Ingeniero Técnico de Telecomunicación – Especialidad Telemática

Consultor: **Manel Llopart Vidal**

Estudios de Informática, Multimedia y Telecomunicación

Universitat Oberta de Catalunya – UOC

Junio 2015



Esta obra está sujeta a una licencia de Reconocimiento-
NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>BCN Bicing Info – Una aplicación Android que se nutre de Open Data y se sirve de Google Maps para mostrar información en tiempo real de utilidad para el usuario de Bicing, el servicio de bicicletas compartidas de la ciudad de Barcelona</i>
Nombre del autor:	José Miguel Gimeno García
Nombre del consultor:	Manel Llopart Vidal
Fecha de entrega (mm/aaaa):	06/2015
Área del Trabajo Final:	Aplicaciones Multimedia de Nueva Generación
Titulación:	<i>Ingeniero Técnico de Telecomunicación, especialidad Telemática</i>
Resumen del Trabajo (máximo 250 palabras):	
<p>Gracias principalmente a la evolución tecnológica que han experimentado los dispositivos móviles y, paralelamente, a la irrupción de potentes sistemas operativos, se ha experimentado una verdadera revolución en el sector de las comunicaciones móviles en la última década.</p> <p>Entre los sistemas operativos para dispositivos móviles, dos plataformas de filosofías contrapuestas destacan sobre el resto: <i>iOS</i> de <i>Apple</i> como SO propietario de uso exclusivo en terminales de la propia marca, frente a <i>Android</i> de <i>Google</i> como SO de código abierto y gratuito.</p> <p>Por otra parte, la filosofía <i>Open Data</i> constituye un mecanismo para la distribución de información de origen gubernamental de modo que sea accesible y reutilizable.</p> <p><i>Bicing</i> es un sistema de transporte urbano basado en el uso compartido de bicicletas que pueden ser recogidas y devueltas en más de 400 estaciones diferentes repartidas por el núcleo urbano de la ciudad de Barcelona. Nace para dar un servicio práctico, saludable y respetuoso con el medio ambiente, como un modo alternativo de desplazarse de un punto a otro de la ciudad.</p> <p>En el presente TFC se ha realizado una <i>app Android</i> que se nutre de los catálogos de datos en abierto para presentar información de utilidad al usuario de <i>Bicing</i>. Un componente esencial de la aplicación es la visualización de un mapa con la ubicación del usuario y de las estaciones de bicicletas <i>Bicing</i> que se genera con la API <i>Google Maps Android</i>, permitiendo p.ej. buscar la estación más cercana trazando sobre el mapa la ruta hasta la misma.</p>	

Abstract (in English, 250 words or less):

In the last decade there has been a revolution in the mobile communications sector, thanks mostly to technological development that mobile devices have undergone as well as due to the emergence of powerful operating systems.

Among the operating systems for mobile devices, two platforms of rival interests stand above the rest. On the one hand, Apple's iOS is a proprietary OS to be used only with Apple terminals; on the other hand, Google's Android is an open-source free OS.

Furthermore, Open Data philosophy is a tool devised for distributing information from governmental sources so that it is accessible and reusable.

Bicing is a public bike share scheme than consists in using bicycles as a means of urban transport; they can be collected and returned in more than 400 different stations throughout the city center of Barcelona. It has been conceived to be a useful, healthy and environmentally-friendly service, as an alternative way to move from one point to another of the city.

In this FDP it has been made an Android app that is fed by open data catalogs to present useful information in order to help users of *Bicing* service. Displaying a map showing the location of the user and the *Bicing* bike stations generated with Google Maps Android API is an essential component of the app that e.g. allows finding the nearest station tracing the route to it on the map.

Palabras clave (entre 4 y 8):

Android – app – Open Data – Google Maps – Bicing

Resumen

R.1.- Auge de los dispositivos móviles

En la última década se ha experimentado una verdadera revolución en el sector de las comunicaciones móviles. En la actualidad, el parque de terminales de telefonía móvil es ya superior al de ordenadores. Es más, la conexión a Internet vía dispositivo móvil es ya más frecuente que por medio de ordenadores, y la tendencia indica que esta diferencia será todavía mayor.

De una parte el **hardware**: la evolución tecnológica que han alcanzado estos dispositivos móviles, y las infraestructuras de red de los operadores que les prestan servicio (actualmente con redes de hasta quinta generación – 5G). De otra parte el **software**: los nuevos SO móviles que han aparecido paralelamente y las herramientas accesibles gratuitamente a cualquiera, que permiten el desarrollo de aplicaciones (*apps*) y su publicación en *app stores* para su libre descarga. Todo ello, conjuntamente con la explosión de las redes sociales y la denominada Web 2.0., ha derivado en un escenario actual que ha supuesto un drástico y profundo cambio en el comportamiento social humano, una suerte de nuevo *homo sapiens mobilis*.

R.2.- Distintos SO alternativos para dispositivos móviles. *Android*

Dentro de los SO para dispositivos móviles destacan, en la actualidad, dos plataformas que, conjuntamente, acaparan la mayor parte de la cuota de mercado, *iOS* de Apple y *Android* de Google. Se trata de dos filosofías de SO diametralmente opuestas. La del *iOS* es la de un SO propietario (no abierto) y de uso exclusivo de los terminales fabricados por Apple. A diferencia de la de *Android*, que es la de un SO de código abierto (licencia *Apache*), basado en *Linux* y disponible de manera gratuita, con lo que se encuentra en terminales de diferentes fabricantes – más aún, si cabe, teniendo en cuenta el hecho de que estos fabricantes forman parte, a su vez, del consorcio de empresas *Open Handset Alliance* (OHA) promovido y auspiciado por Google para el

desarrollo de *Android*, lo que representa una posición estratégica determinante que le permite superar a *iOS* en penetración de mercado.

R.3.- IDE Eclipse

Eclipse es actualmente uno de los IDEs más populares para desarrollar aplicaciones en Java (no obstante más del 90% de su código fuente está escrito en este lenguaje), en C/C++, y naturalmente también en *Android*. Instalando los complementos necesarios (en concreto *Android Developer Tools* o abreviadamente *ADT Plugin*), se configura este IDE, por medio del cual se facilitan tareas básicas como la creación de los menús y *layouts* y se dispone de un emulador para depurar las aplicaciones sin necesidad de instalarlas en un terminal físico. Desde la página oficial de *Android* para desarrolladores ([Android Developers](#)) puede descargarse el paquete *ADT Bundle* que incluye todo lo necesario para desarrollar aplicaciones *Android*. Incluye Eclipse junto con el complemento *ADT Plugin*, las herramientas *Android SDK Tools*, las herramientas de plataforma *Android Platform Tools*, la plataforma *Android* más reciente, y la última imagen de sistema *Android* disponible para ser utilizada en el emulador.

Por medio del gestor *Android SDK Manager* se pueden actualizar en cualquier momento estas herramientas así como las nuevas versiones de *Android* en cuanto son lanzadas. Se utilizará esta distribución *ADT Bundle* para la realización del proyecto, que ha sido la única que ha tenido soporte en el sitio oficial de *Android Developers* hasta diciembre de 2014. A partir de esta fecha, con el lanzamiento de la versión 1.0 de *Android Studio*, el IDE desarrollado durante los últimos años por *Google*, éste ha pasado a ser el que tiene soporte oficial. Sin embargo, como los desarrolladores han estado empleando *Eclipse* desde los primeros momentos de *Android*, resulta más sencillo encontrar documentación y referencias, por lo que se trabajará con este entorno, al menos en principio y hasta conocer un poco más esta nueva herramienta y sus ventajas. La situación general, no obstante, es de transición de un IDE a otro, por lo que a muchos desarrolladores les ocurre lo mismo y *Google* sigue prestando soporte para ayudar a este cambio.

R.4.- Filosofía de datos en abierto u *Open Data*

Mediante la que se distribuye abiertamente información de diverso contenido, en especial de origen gubernamental y de las administraciones públicas, de modo que sea accesible y reutilizable, sin limitaciones significativas.

Dentro de los catálogos de datos en abierto, destacan los que mantienen las grandes ciudades, por su cantidad y calidad, y que facilitan el aprovechamiento para el desarrollo de aplicaciones que se nutran de sus datos. Para el presente TFC se utilizará fundamentalmente el catálogo de datos abiertos del Ayuntamiento de Barcelona (*OpenData BCN*) y de la Generalitat de Catalunya (*dades obertes gencat*).

R.5.- *Google Maps Android API*

Mediante esta interfaz se consigue incrustar mapas de *Google Maps* en aplicaciones *Android*. El popular servidor *Google Maps* proporciona aplicaciones de mapas de manera gratuita con características como permitir zoom, añadir marcadores, etc. Junto con el sensor de localización mediante GPS del dispositivo móvil servirá para mostrar la ubicación del usuario entre otros aspectos.

R.6.- Aplicación que proporciona información útil y en tiempo real.

Se pretende que esta información suponga un valor añadido para el usuario de *Bicing*. Como se explicará más adelante (capítulos sobre el Estado del arte), cuando se opta por este servicio, es de gran ayuda para el usuario conocer cuál es la estación más cercana a su ubicación actual. Utilizando el catálogo de datos en abierto de la ciudad de Barcelona la aplicación debe mostrar al usuario la estación de mayor disponibilidad entre las más cercanas a su ubicación bien para recoger (evitando acudir a las estaciones vacías), bien para devolver la bici (para no sobrepasar el límite de 30 minutos gratuitos mientras se busca una estación con un hueco libre).

Además de esta funcionalidad básica se valorará si implementar o no otras que se definirán más adelante – en el apartado sobre alcance de la aplicación – como el análisis de tendencias para aumentar las posibilidades de recogida o devolución según los datos que se recogerán periódicamente en una base de datos mantenida al efecto.

R.7.- Servicio *Bicing* de la ciudad de Barcelona

Bicing es un sistema de transporte urbano basado en el uso compartido de bicicletas que pueden ser recogidas y devueltas en distintas estaciones, más de 400, repartidas por el núcleo urbano de la ciudad de Barcelona. Nace para dar un servicio práctico, saludable y respetuoso con el medio ambiente. Como un modo alternativo y sostenible de desplazarse de un punto a otro de la ciudad combinándolo o no con el transporte público. En ningún caso supone una oferta turística o recreativa ni una competencia a las empresas de alquiler de bicicletas, su *leitmotiv* es ser un sustituto o complemento del resto de transportes urbanos.

Dedicatoria

*A mi esposa Cristina.
Gracias por tener la paciencia
para aguantarme durante
estos meses.*

Índice

Ficha del trabajo final	iii
Resumen breve.....	iii
<i>Abstract</i>	iv
Palabras clave	iv
Resumen largo	v
R.1.- Auge de los dispositivos móviles	v
R.2.- Distintos SO alternativos para dispositivos móviles. <i>Android</i>	v
R.3.- IDE Eclipse	vi
R.4.- Filosofía de datos en abierto u <i>Open Data</i>	vii
R.5. - Google Maps Android API.....	vii
R.6.- Aplicación que proporciona información útil y en tiempo real	vii
R.7.- Servicio <i>Bicing</i> de la ciudad de Barcelona	viii
Dedicatoria.....	ix
Índice	x
Índice de figuras	xiv
Índice de tablas	xvi
Lista alfabética de acrónimos	xvii
Capítulo 1.- Introducción.....	1
1.1.- Contexto y justificación del trabajo.....	1
1.1.1.- Beneficios	1
1.1.2.- Motivación.....	1
1.2.- Objetivos del trabajo	2
1.3.- Enfoque y método seguido	4
1.4.- Planificación del trabajo	5
1.5.- Breve resumen de productos obtenidos	5

1.6.- Breve descripción del resto de capítulos de la memoria	6
Capítulo 2.- E. del arte I: Ciclismo urb. y sist. de bicicletas comp. <i>Bicing</i>	8
2.1.- Ciclismo urbano	8
2.1.1.- Definición, orígenes y motivación.....	8
2.1.2.- Datos estadísticos. La bicicleta como transp. público	8
2.1.3.- Ventajas, inconvenientes y su posible solución.....	12
2.2.- Sistemas de bicicletas compartidas	15
2.2.1.- Definición. Breve historia.....	15
2.2.2.- Situación actual.....	17
2.3.- <i>Bicing</i> . Descripción del servicio y relación con el TFC	19
Capítulo 3.- Estado del arte II: <i>Open Data</i>	21
3.1.- Definición y principios de los datos abiertos	21
3.2.- Excepciones. Qué no se considera <i>Open Data</i>	21
3.3.- Ejemplos. Que sí se considera <i>Open Data</i>	22
3.4.- Características	22
3.5.- Leitmotiv, puesta en valor, beneficios y tipos de datos abiertos	23
3.6.- Catálogos de datos abiertos. Catálogos utilizables para el TFC.....	25
Capítulo 4.- E. del arte III: Sist. operativos y <i>apps</i> para disp. móviles. <i>Android</i>	28
4.1.- <i>Android</i>	28
4.1.1.- Qué es <i>Android</i>	28
4. 1.2.- Origen. Breve aproximación histórica.....	28
4. 1.3.- Características diferenciadoras de <i>Android</i>	29
4. 1.4.- Arquitectura de <i>Android</i> . Breve descripción	32
4. 1.5.- Situación actual.....	34
4. 1.6.- Cuota de mercado	35
4. 1.7.- Idoneidad para el proyecto	36
4.2.- Herramientas y técnicas disponibles	38

4.2.1.- SDK oficial de <i>Android</i>	38
4.2.2.- IDE Eclipse. <i>Google Maps</i>	39
4.2.3.- Otras herramientas y aspectos a tener en cuenta	39
4.3.- Aplicaciones similares. <i>App</i> oficial	41
Capítulo 5.- Diseño.....	43
5.1.- Estrategia de diseño	43
5.2.- Estructura de un proyecto <i>Android</i>	44
5.2.1.- Información inicial para crear un proyecto.....	44
5.2.2.- La versión del API.....	45
5.2.3.- Estructura de directorios	46
5.2.4.- Ciclo de vida de una actividad en <i>Android</i>	49
5.3.- Diseño del menú principal.....	50
5.3.1.- Icono de la aplicación	51
5.4.- Diseño del mapa	52
5.5.- Limitaciones del emulador incorporado. <i>Genymotion</i>	52
5.6.- Inspección de los datos abiertos. Diseño de la base de datos	54
5.6.1.- Inspección del recurso de las estaciones de <i>Bicing</i>	54
5.6.2.- Inspección del recurso de previsión meteorológica.....	58
Capítulo 6.- Implementación	61
6.1.- Clase y actividad principal. Menú inicial	61
6.2.- Clase y actividad lista de estaciones.....	65
6.2.1.- Tarea <code>DownloadTask</code>	67
6.2.2.- Tarea <code>WriteOnDataBaseTask</code>	67
6.2.3.- Tarea <code>ListViewStationsTask</code>	68
6.2.4.- Adaptador para mostrar los datos	68
6.3.- Clase y actividad mapa.....	70
6.4.- Resto de clases y actividades.....	72

6.4.1.- Clase y actividad de predicción meteorológica	72
6.4.2.- Clase y actividad de ayuda	73
6.4.3.- Clase y actividad curiosidades estadísticas	73
6.4.4.- Clase y actividad <i>About</i>	74
6.5.-Menús de preferencias	74
Capítulo 7.- Conclusiones y líneas futuras	75
7.1.- Objetivos alcanzados y funciones implementadas y descartadas.....	75
7.2.- Líneas de trabajo futuras	77
7.3.- Puntos fuertes del proyecto y aspectos que se cambiarían	78
Bibliografía.....	80
Referencias	84
Anexo.- Mapa conceptual.....	90

Índice de figuras

Diagrama de Gantt.....	5
Figura 1.- Producción anual de bicicletas y automóviles	10
Figura 2.- Ranking <i>The Copenhagenize Index 2013</i>	18
Figura 3.- Tipos de datos abiertos	24
Figura 4. - Ranking <i>Open Knowledge Foundation</i> por país.....	25
Figura 5.- Arquitectura de <i>Android</i>	33
Figura 6.- Distribución de las versiones de <i>Android</i>	40
Figura 7.- Muestra de la <i>app</i> oficial <i>Bicing</i>	41
Figura 8.- Asistente de Eclipse para crear una <i>app Android</i>	46
Figura 9.- Árbol de directorios de un proyecto <i>Android</i>	48
Figura 10.- Propiedades de un proyecto <i>Android</i>	49
Figura 11.- Ciclo de vida de una actividad.....	51
Figura 12.- Menú principal.....	52
Figura 13.- Control <code>ListView</code> genérico	53
Figura 14.- Pantalla principal con el icono de la <i>app</i> instalada.....	53
Figura 15.- <i>Clusterización</i> o agrupamiento de marcadores en el mapa	53
Figura 16.- Trazado de la ruta hasta la estación de destino sobre el mapa.....	54
Figura 17.- Descripción del recurso de estaciones de <i>Bicing</i>	56
Figura 18.- Cap. de pantalla del fichero XML de estaciones	57
Figura 19.- Cap. de pantalla con la descripción del recurso meteorológico.....	60
Figura 20.- Cap. de pantalla del fichero XML, códigos de comarca.....	61
Figura 21.- Cap. de pantalla del fichero XML, códigos de descripción.....	61
Figura 22.- Cap. de pantalla del fichero XML, predicción por comarca.....	62
Figura 23.- Cap. de pantalla con el menú principal de la aplicación.....	66
Figura 24.- Cap. de pantalla con la actividad lista de estaciones	70

Figura 25.- Cap. de pantalla con el mensaje emergente de lista estación.....	71
Figura 26.- Mapa centrado en ubicación del usuario y marcadores agrupados.....	73
Figura 27.- Mapa después de hacer zoom desagrupando <i>clústers</i>	73
Figura 28.- Mapa con <code>Toast</code> al hacer clic en un <i>clúster</i>	73
Figura 29.- Mapa tras pulsar botón de búsqueda estación más cercana	73
Figura 30.- Mapa después de hacer zoom para ver los detalles de la ruta.....	73
Figura 31.- Mapa haciendo clic en marcador estación para ver detalles	73
Figura 32.- Detalle de la actividad de pronóstico meteorológico.....	74
Figura 33.- Cap. de pantalla de la actividad de ayuda al usuario	75
Figura 34.- Detalle actividad curiosidades (1).....	76
Figura 35.- Detalle actividad curiosidades (2).....	76
Figura 36.- Detalle actividad curiosidades (y 3).....	76
Figura 37.- Detalle de la actividad <i>About</i>	76

Índice de tablas

Tabla 1.- Número de bicicletas en el mundo	9
Tabla 2.- Sistemas de bicicletas compartidas en el mundo	17
Tabla 3.- Estadísticas de <i>Bicing</i>	20
Tabla 4.- Cuota de mercado de los principales SO móviles	35
Tabla 5.- Distribución de las versiones de <i>Android</i>	47

Lista alfabética de acrónimos

- 5G – *Fifth Generation*, en inglés Quinta Generación
- ADT – *Android Developer Tools*, en inglés Herramientas del Desarrollador en *Android*
- API – *Application Programming Interface*, en inglés Interfaz de Programación de Aplicaciones
- BCN – Barcelona
- EE.UU. – Estados Unidos
- FDP – *Final Degree Project*, en inglés Trabajo Final de Carrera
- GPS – *Global Positioning System*, en inglés Sistema de Posicionamiento Global
- IDE – *Integrated Development Environment*, en inglés Entorno de Desarrollo Integrado
- IT – *Information Technology*, en inglés Tecnología de la Información
- NDK – *Native Development Kit*, en inglés Kit de Desarrollo Nativo
- OHA – *Open Handset Alliance*, en inglés Alianza para Terminales en Abierto
- SDK – *Software Development Kit*, en inglés Kit de Desarrollo de Software
- SO – Sistema Operativo
- TFC – Trabajo Final de Carrera
- TIC – Tecnologías de la Información y la Comunicación
- UAB – Universitat Autònoma de Barcelona
- UOC – Universitat Oberta de Catalunya
- URL – *Uniform Resource Locator*, en inglés Localizador de Recursos Uniforme
- URL – Universitat Ramon Llull
- URV – Universitat Rovira i Virgili

Capítulo 1.- Introducción

1.1.- Contexto y justificación del Trabajo

En el presente Trabajo Final de Carrera (TFC) se pretende diseñar e implementar una aplicación (*app*) para dispositivos móviles con sistema operativo *Android*, sirviéndose del IDE Eclipse. Esta *app*, mediante reutilización de datos ya existentes y accesibles bajo filosofía *Open Data*, con ayuda de *Google Maps Android API*, facilitará información de utilidad y en tiempo real al usuario del servicio *Bicing* de la ciudad de Barcelona.

La elección de este proyecto se ajusta al área escogida para la realización del TFC: Aplicaciones Multimedia de Nueva Generación. Se trata de una aplicación, es multimedia y es de nueva generación.

1.1.1.- Beneficios

Los beneficios que se pretenden con la realización de este TFC comprenden:

- Ser de utilidad a los usuarios del servicio *Bicing*.
- Contribuir a la difusión de un modelo de transporte urbano saludable y respetuoso con el medio ambiente.
- Contribuir a la filosofía de datos abiertos.

1.1.2.- Motivación

Después de varios semestres de duro esfuerzo, conciliando el estudio con la vida familiar y laboral, con la elaboración de este TFC se pone fin a los estudios cursados en la UOC permitiendo obtener el ansiado título de Ingeniero Técnico de Telecomunicación. Los propósitos que han motivado la elección de este TFC son los siguientes:

- Introducirse y profundizar en el desarrollo de aplicaciones para dispositivos móviles *Android*, y en la programación con la interfaz *Google Maps Android API* para incrustación de mapas.
- Introducirse en la reutilización de recursos *Open Data* para la construcción de aplicaciones.

- Aplicar de manera práctica el conocimiento adquirido durante los estudios en materias como Programación Orientada a Objetos, Ingeniería del Software o Bases de Datos.
- Profundizar en el dominio de Eclipse, IDE que ya se ha empleado en otras asignaturas de la titulación.
- Adquirir experiencia en la redacción de textos científico-técnicos, cuidando los aspectos formales e introduciéndose en el empleo de LaTeX, un sistema de composición de textos especialmente orientado a la edición de artículos, tesis y documentos con contenido científico y técnico.
- Servir de síntesis a los estudios de Ingeniería Técnica de Telecomunicación que se finalizan.
- Servir de preparación para la futura dedicación profesional como ingeniero. Poniendo en práctica conocimientos y habilidades adquiridos durante los estudios para enfrentarse a problemas del mundo real.
- Representar un paso previo a los estudios de Máster, Postgrado y Doctorado, como una preparación para futuras actividades de especialización e investigación. En particular para los estudios de Máster Interuniversitario de Ingeniería de Telecomunicación (UOC/URL) y los de Máster Interuniversitario de Seguridad de las TIC (UOC/UAB/URV) que se tienen intención de cursar una vez titulado.

1.2.- Objetivos del Trabajo

El objetivo fundamental es la realización de una aplicación *Android* que, nutriéndose de *Open Data*, presente información en tiempo real sobre la disponibilidad de las estaciones más próximas al usuario de *Bicing*.

Dentro de las funcionalidades básicas mínimas de la aplicación se implementarán las siguientes.

Información sobre la disponibilidad de bicicletas en las estaciones más cercanas a la ubicación del usuario. En concreto deberá hacerlo mostrando un mapa por pantalla mediante *Google Maps* en el que se visualice la localización actual del usuario y la de las estaciones más cercanas de las que interesa conocer la distancia a la que se encuentran y el número de bicis disponibles y huecos libres.

Trazar y mostrar la ruta hasta la estación más cercana a la ubicación del usuario en función de si se desea buscar bicis disponibles o huecos libres. Para el cálculo de la ruta óptima, además del origen y el destino, se tendría en cuenta la disponibilidad de estaciones, el uso en la medida de lo posible de carriles bici o vías similares, además y opcionalmente se valorará proporcionar correspondencia con paradas/estaciones de transporte público y las incidencias del tráfico concretas que afectan a su ruta.

Pronóstico del tiempo para las próximas horas. Para advertir al usuario de la conveniencia o no del uso de *Bicing* si p.ej. se espera una tormenta.

La aplicación deberá facilitar al usuario la interacción intuitiva con la misma por medio de menús, botones, listas, mapas, etc. que proporciona la plataforma *Android*.

Como funcionalidades adicionales opcionales se valorará la implementación de alguna de las siguientes, dependiendo de la dificultad que se presente en su realización y de que las funcionalidades básicas se hayan implementado o no con éxito, ello se concretará a medida que se desarrolle el proyecto. En todo caso, por el momento, no forman parte el alcance de este TFC:

- a) Recoger comentarios de usuarios sobre incidencias en el servicio y presentarlos según ubicación en el plano de la ciudad.
- b) Análisis de tendencias de uso de las distintas estaciones mediante un monitoreo periódico de las disponibilidades de las mismas. El objeto que se persigue es que el usuario que se dispone a recoger una bici (o a devolverla) pueda decidir a qué estación acudir de entre las cercanas en función de la tendencia de disponibilidad de cada una, asegurándose una mayor probabilidad de encontrar bicis disponibles (o huecos libres) según su necesidad.
- c) Publicar la *app* en *Google Play* para su descarga por cualquier usuario.
- d) Realizar una versión de la *app* para un *smartwatch*, utilizando la última versión de *Android* con extensiones para dispositivos *wearables*.

1.3.- Enfoque y método seguido

El enfoque adoptado ha sido crear una aplicación desde cero. De esta manera se persiguen dos objetivos. Por un lado, familiarizarse con el desarrollo de aplicaciones en la plataforma *Android* y, por otro lado, diferenciar la aplicación del resto de aplicaciones existentes que pueden descargarse en *Google Play* sobre *Bicing*.

Se ha optado por seguir una estrategia por fases, en la que se parte de la aplicación *Android* más sencilla posible (mostrar el mensaje “*hola, mundo*” por pantalla) e ir añadiendo en las sucesivas fases las funcionalidades deseadas. De esta manera se han planificado las fases como se expone a continuación.

- Fase 0.- Verificación del entorno del trabajo y construcción de una aplicación básica de prueba tipo “*hola, mundo*”.
- Fase 1.- Estructura de un menú principal y separación de las funcionalidades en *Activities* o pantallas.
- Fase 2.- Mostrar por pantalla un mapa con la ubicación del usuario. Resolución de dificultades con el emulador y *Google Maps*.
- Fase 3a.- Conexión de la aplicación a Internet. Conexión y lectura de catálogos de *open data*.
- Fase 3b.- Creación, escritura y lectura de una base de datos que mantenga la información obtenida de las estaciones *Bicing*.
- Fase 4.- *Activity* lista de estaciones.
- Fase 5.- *Activity* mapa – mostrar estaciones en el mapa.
- Fase 6.- *Activity* mapa – búsqueda de estación más cercana.
- Fase 7 – *Activity* mapa – búsqueda con criterios y trazado de ruta.
- Fase 8.- Resto de *activities*.
- Fase 9.- Refinamiento de las *activities*.

- Fase 10.- Pruebas, redacción de menú ayuda y traducción inglés.

1.4.- Planificación del Trabajo

En cuanto a los recursos necesarios, se ha invertido un tiempo considerable en configurar el entorno de trabajo y familiarizarse con él. Como se explicará más adelante, se han descargado e instalado los complementos necesarios para poder emplear el *IDE Eclipse* en el desarrollo de la app. Además ha sido necesario crear una cuenta de desarrollador para poder emplear los mapas de la *Google Maps Android API* e instalar el emulador de *Genymotion*. En el siguiente diagrama de Gantt, se muestra la planificación temporal del TFC:

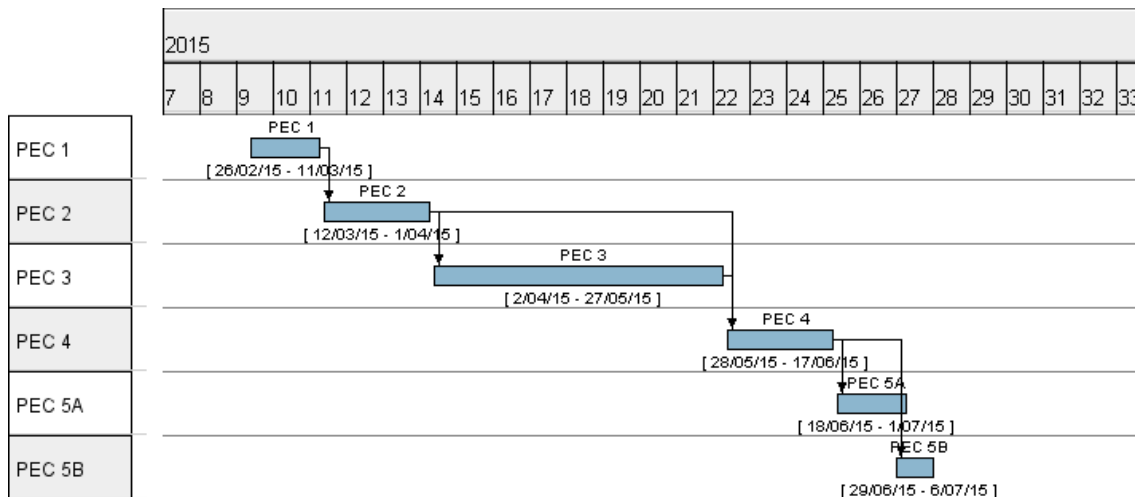


Diagrama de Gantt

1.5.- Breve resumen de productos obtenidos

Los documentos que forman parte de la entrega final del TFC son:

- El código fuente de la aplicación, con comentarios y con la estructura de directorios que utilizan los proyectos en *Eclipse*.
- La memoria del TFC. Además se redactará una versión mediante el programa LaTeX.
- Un mapa conceptual del TFC.
- Una presentación en vídeo (disponible en Vimeo).

1.6.- Breve descripción de los otros capítulos de la memoria

En los capítulos sobre el Estado del Arte se pretende proporcionar una visión global de la cultura del ciclismo urbano y de los sistemas de bicicletas compartidas así como de la filosofía de datos abiertos u *Open Data*, poniendo en contexto la situación actual en el ámbito del desarrollo de aplicaciones móviles, y en concreto en la plataforma *Android*, que servirá para implementar la aplicación objeto de este TFC: *BCN Bicing Info*, una aplicación *Android* que se nutre de *Open Data* para presentar, sirviéndose de *Google Maps*, información en tiempo real de utilidad para el usuario de *Bicing*, el servicio público de bicicletas compartidas de la ciudad de Barcelona.

En el segundo capítulo se introduce el tema del ciclismo urbano y de los sistemas de bicicletas compartidas que han experimentado en las últimas dos décadas un notable éxito, entre ellos el de *Bicing* en Barcelona. A continuación, en el tercer capítulo, se expone en qué consiste la filosofía de datos abiertos u *Open Data*, y la utilización que se hace de ella en este proyecto. En el cuarto capítulo se repasan los sistemas operativos para dispositivos móviles y el auge en el desarrollo de aplicaciones, haciendo hincapié en el caso de *Android*. Analizar las distintas herramientas y técnicas de desarrollo de aplicaciones (*apps*) en *Android* es el objeto del quinto capítulo, las cuales servirán para la realización de la *app* de este TFC. Finalmente en el sexto capítulo se describen y comparan otras *apps* relacionadas con este proyecto que pueden encontrarse en la actualidad.

En esta los capítulos de Diseño e implementación se presentará cómo se ha llevado a cabo el desarrollo de la aplicación de este TFC.

En primer lugar, partiendo del Estado del arte, se concretarán en el capítulo 7 los objetivos definidos en esta introducción, para adecuarlos a la realidad y a las características que se han mostrado en dicho Estado del arte y para servir de punto de partida para el diseño de la aplicación.

En segundo lugar, en el capítulo 8 partiendo de estas especificaciones, se planifica una hoja de ruta para desarrollar la aplicación. Se apuesta por la máxima de “divide y vencerás”. La estrategia que se sigue es construir la aplicación paso a paso, comenzando por la aplicación más sencilla posible, “*hola, mundo*” e ir añadiendo las

funcionalidades progresivamente, una vez que se ha asegurado de que las anteriores funcionan correctamente. Esta manera de proceder no siempre es adecuada y depende de la naturaleza de cada proyecto, sin embargo, en este TFC esta planificación se adapta perfectamente ya que la propia forma de construcción de las aplicaciones en *Android* separa las distintas funcionalidades en *Activities* (lo que llamamos pantallas en nuestro móvil o tableta) con lo que lo más sencillo es abordar el problema como actividades por separado que es posible ir desarrollando una a continuación de otra sin necesidad de afrontar las distintas funcionalidades simultáneamente.

Una vez establecidas estas bases, en el capítulo 9 se trata la implementación propiamente dicha de la aplicación. Para ello se utilizan las herramientas que hemos descrito en el Estado del arte y otras que se han tenido que emplear para solucionar problemas que han surgido y que no se habían previsto. Son estas dificultades que surgen inesperadamente y que deben resolverse las que proporcionan la experiencia práctica al desarrollador y complementan el aprendizaje teórico adquirido en la etapa universitaria. En definitiva, las que siguen formando al ingeniero a medida que avanza en su etapa profesional.

Finalmente en el capítulo 10 se exponen las conclusiones sobre el TFC y las posibles líneas futuras que tratar.

N.B.- Este documento en formato PDF ha sido generado con marcadores para facilitar la navegación por el mismo cuando se emplea una aplicación cualquiera entre las muchas que existen para abrir este tipo de archivos, como p.ej. *Adobe Reader*. Por ello se recomienda la utilización de uno de estos programas para la lectura del mismo en vez de la impresión en papel, la cual proporciona una experiencia menos interactiva y menos respetuosa con el medio ambiente. Asimismo, las notas a las referencias bibliográficas que figuran al final del documento y que se indican entre corchetes a lo largo del texto, incluyen un vínculo para acceder de una manera más cómoda a las mismas.

Capítulo 2.- Estado del arte I: Ciclismo urbano y sistemas de bicicletas compartidas. El servicio *Bicing*

2.1.- Ciclismo urbano

2.1.1.- Definición. Orígenes y motivación

El ciclismo urbano consiste en el uso de la bicicleta como medio de transporte, más que como una actividad lúdica o deportiva, para cubrir distancias modestas de unos pocos kilómetros habitualmente en un entorno urbano. Bien sea compartiendo la vía con los demás vehículos a motor o no. Actualmente es considerada como la modalidad de ciclismo más practicada en el mundo [[HER04](#)].

Históricamente, la proliferación del automóvil en los países desarrollados a partir de la segunda mitad del siglo XX ha traído como consecuencia una progresiva congestión del tráfico y una pérdida de espacio público en detrimento de los peatones y de la habitabilidad urbana. La propuesta del ciclismo urbano pretende recuperar parte de este espacio público con las ventajas que supone y que se expondrán más adelante. En los países en desarrollo, en cambio, sencillamente es la alternativa más asequible para el transporte urbano.

Después del transporte a pie, la bicicleta es el medio de locomoción urbano más popular.

2.1.2.- Datos estadísticos y promoción de la bicicleta como medio de transporte

A continuación se exponen algunos datos de interés sobre el número de bicicletas que hay en el mundo y las unidades que se fabrican en comparación con las de la producción de automóviles.

De acuerdo con la organización *Worldometers Real Time World Statistics*, se estima que actualmente hay más de mil millones de bicicletas en el mundo [[WOM14](#)], de las cuales aproximadamente la mitad se encuentran en China, como refleja la Tabla 1 que aparece en su página *web*:

Country	Quantity	Year
China	450,000,000	1992
USA	100,000,000	1995
Japan	72,540,000	1996
Germany	62,000,000	1996
India	30,800,000	1990
Indonesia	22,300,000	1982
Italy	23,000,000	1995
UK	20,000,000	1995
France	20,000,000	1995
Brazil	40,000,000	1996
Netherlands	16,500,000	2000
Canada	10,150,000	1992
Spain	6,950,000	1995
Sweden	6,000,000	1995
South Korea	6,500,000	1985
Mexico	6,000,000	1986
Belgium	5,200,000	1995
Rumania	5,000,000	1995
Denmark	4,500,000	1995
Switzerland	3,800,000	1996
Hungary	3,500,000	1995
Australia	3,300,000	1995
Finland	3,250,000	1995
Norway	3,000,000	1995

Tabla 1.- Número de bicicletas en el mundo.

Fuente: [[WOM14](#)] *Worldometers*.

Hasta finales de la década de los 60, la producción anual de automóviles era esencialmente la misma, sin embargo a partir de entonces la producción de bicicletas se ha disparado en todo el mundo hasta llegar a ser entre dos veces y dos veces y media la de automóviles, según datos del *WorldWatch Institute* que reproducimos a continuación en la Gráfica 1 [[WWI14](#)].

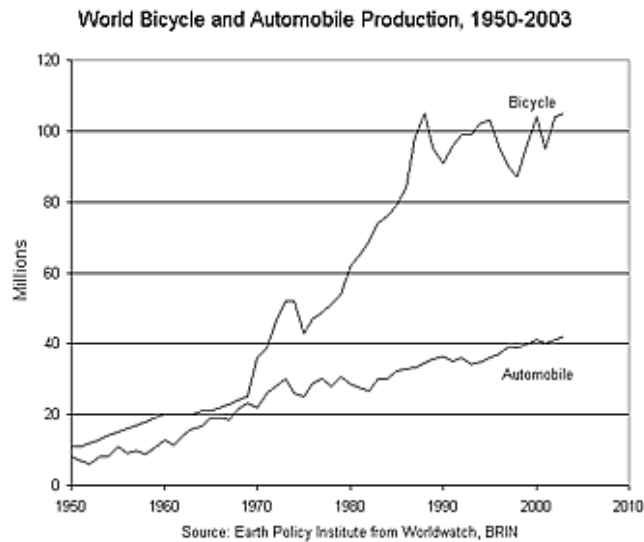


Figura 1.- Producción anual de bicicletas y automóviles en el período 1950-2003. Fuente: *WorldWatch Institute* [WW14].

Estos datos certifican el crecimiento en la popularidad del uso de la bicicleta que se ha experimentado en el último medio siglo.

En cuanto a la distribución geográfica de la producción, de cada tres bicicletas fabricadas, dos se hacen en China, según estadísticas consultadas en la web de la ONG *International Bicycle Fund* [IBF14]. Destaca el hecho de que cuatro países asiáticos se encuentren entre los cinco mayores productores: China, India, Taiwán y Japón. En Vietnam, por ejemplo, se registró en 2002 la mayor tasa de crecimiento en la producción, con un incremento del 250%, fabricándose más de dos millones [GAR04].

Como puede observarse, el uso de la bicicleta está especialmente extendido entre la población asiática. Es muy popular igualmente en países que tienen o han tenido economías no capitalistas, como en los mencionados China y Vietnam, o como en Cuba y Corea del Norte, donde los gobiernos han impulsado su uso, más bien debido a su deficiente desarrollo industrial más que por conciencia medioambiental.

La Unión Europea, por su parte, es el tercer productor mundial. Con países donde el uso de la bicicleta es generalizado, como Dinamarca y especialmente, Países Bajos, donde p.ej. en Ámsterdam hay más bicicletas que habitantes [GRS13].

En el polo opuesto está una potencia económica como Estados Unidos, donde la bicicleta se encuentra más marginada, debido a la cultura dominante del automóvil, con un 95% aparcamientos gratuitos, el precio del combustible más barato, una fiscalidad más ventajosa y un 50% más de propietarios de automóviles que en Europa occidental. Además, debido a las pobres infraestructuras para ciclistas, se tienen doce veces más posibilidades de fallecer por accidente que las personas que viajan en automóvil. Los ciclistas estadounidenses son dos veces más propensos que los alemanes a fallecer por accidente en carretera, y hasta tres veces más que los holandeses [\[GAR04\]](#).

No se les da, por tanto, la misma importancia a los ciclistas urbanos en todos los países, ni se toman en cuenta de la misma manera a la hora de estudiar la planificación y el desarrollo de las infraestructuras viales. De hacerlo, seguramente se subsanarían muchas de las deficiencias y problemas de tráfico y contaminación que asolan actualmente a muchas ciudades del mundo, lo que constituye todo un desafío para los responsables políticos. En palabras del ex Presidente de Singapur, Lee Kuan Yew, y que da nombre al prestigioso premio *World City Prize* que se otorga anualmente a la mejor ciudad, “no importa cuánto gastemos en construir vías rápidas, siempre serán insuficientes para alojar la creciente cantidad de vehículos que entran a la ciudad” [\[LKY\]](#).

Según un estudio suizo, que cita el Comité Pro-bici de Sevilla [\[HIN\]](#) “un coche que viaja a 40 km/h, requiere tres veces más espacio que uno que viaja a 10 km/h; mientras que en un carril reservado a automóviles particulares utilizado al máximo (con cuatro ocupantes por automóvil) no transporta más de tres mil personas por hora, sin embargo en este mismo espacio, reservado para bicicletas pueden transitar más de treinta mil ciclistas”.

En opinión de este mismo Comité Pro-bici: “Somos conscientes de que en toda ciudad en crecimiento son necesarias las grandes avenidas, sin embargo la creación de espacios para medios de transporte alternativos, como la bicicleta, la correcta planeación y distribución de estos espacios, nos ayuda a prevenir problemas futuros, a desahogar congestionamientos, y lo más importante, a brindarle seguridad a los millones de personas que ya utilizan este medio de transporte”.

2.1.3.- Ventajas. Inconvenientes y su posible solución

Entre las ventajas de la práctica del ciclismo urbano, se destacan las siguientes:

- Es saludable para quien lo practica. Al tratarse de un ejercicio cardiovascular, si se practica de manera moderada y periódicamente, redundará en la prevención de la obesidad y de las enfermedades del corazón, que afectan especialmente a los países ricos. Mejorar la forma física repercute en una mayor autoestima. Por añadidura, cuando la práctica es generalizada, puede suponer un ahorro para los sistemas nacionales de salud, de la misma manera que se ha demostrado en otras actuaciones como la restricción en el consumo del tabaco, por ejemplo.
- Es respetuoso con el medio ambiente. Al contrario que en los automóviles, no produce ningún tipo de emisión de gases contaminantes durante su uso, incluso en los modelos de bicicleta eléctrica. La contaminación acústica es igualmente imperceptible cuando la comparamos con la de un automóvil y muy similar a la de un peatón. Para la fabricación se consumen significativamente menos recursos que para la producción de automóviles y los procesos productivos son mucho menos contaminantes.
- Es económicamente asequible para el usuario. Los modelos más básicos no suponen un desembolso elevado y siempre están las alternativas del alquiler y de los sistemas de bicicletas compartidas. No hay excusa para no utilizarlo.
- Para las administraciones públicas, también es económicamente asequible. Las infraestructuras viales (carriles-bici, carriles compartidos, aparcas-bicis, etc...) no son de elevado coste o de coste mucho menor que para el tráfico de automóviles.
- Es sociable. A diferencia de la conducción de un automóvil, no provoca estrés ni agresividad entre los usuarios, promueve el buen comportamiento cívico, y la interacción con otros usuarios que comparten la vía como p.ej. peatones, corredores o patinadores.
- Es divertido. Para la mayoría de la población las sensaciones que produce un paseo en bicicleta se consideran agradables.

- En determinadas ciudades especialmente congestionadas, se trata del medio de transporte más rápido para distancias cortas (inferiores a 5 km.) según un estudio de la Unión Europea [[DEK99](#)].

A pesar de que la lista de ventajas es amplia e importante, se han propuesto varias objeciones al ciclismo urbano, que se exponen a continuación junto con las posibles soluciones:

- La distancia es un factor limitador. No todo el mundo se siente capacitado o tiene tiempo para pedalear distancias largas. Una posible solución consiste en integrar el ciclismo urbano con la red de transporte público. En la actualidad se promueve la colocación de aparca-bicis en las proximidades de estaciones y paradas de metro, autobús y tren, o cerca de aparcamientos disuasorios. De la misma manera, hay un auge en la utilización de bicicletas plegables que permiten llevarla consigo.
- Otro factor importante es la seguridad. El ciclismo se percibe como peligroso al compartir la vía con el resto de vehículos a motor, frente a los que está en inferioridad de condiciones en caso de accidente. Sin embargo, estadísticamente el porcentaje de ciclistas fallecidos suele ser inferior al de los vehículos a motor [[DEK99](#)]. En este sentido es clave planificar infraestructuras viales que tengan en cuenta a los ciclistas, para integrarse con el resto de tráfico rodado y con los peatones, así como invertir en educación vial. El uso del casco es recomendable y en muchas poblaciones obligatorio.
- Robos y vandalismo. Es más frecuente el hurto de las bicicletas o de partes de las mismas y el causar daños vandálicos que en el caso de los automóviles. Para intentar superar este inconveniente existen iniciativas muy interesantes como la matriculación o el registro de bicicletas que ya funcionan en algunas ciudades. De la misma forma que existen aparcamientos de automóviles vigilados también se podrían facilitar zonas de aparca-bicis con vídeo-vigilancia.
- No permite llevar equipaje o pasajeros. Si bien esto es una limitación, también debemos pensar que durante la mayoría de los desplazamientos

urbanos apenas se lleva una carga importante. Actualmente existen sillas que se acoplan perfectamente a las bicicletas para llevar menores que no tienen edad suficiente para andar en bicicleta.

- La climatología incide en su práctica. Aunque cada vez existan mejores y más completos equipamientos, pedalear con lluvia o con frío excesivo no deja de ser incómodo y peligroso. En el otro extremo, una temperatura muy alta favorece la sudoración y el cansancio, lo que puede hacer desistir de su utilización para acudir al lugar de trabajo o a la universidad, por ejemplo.

Para hacerse una idea de los beneficios de las ventajas que se han expuesto, se cita un estudio realizado por la empresa *Grass Roots* en febrero de 2014, según el cual “un aumento del 20% del uso de la bicicleta supondría una inyección económica para España de 500 millones de euros” [[GRA](#)]. Por ejemplo, en lo que respecta a la reducción de emisiones contaminantes, en este informe se estima que si el 15% de los madrileños utilizara la bicicleta para ir al trabajo se ahorraría la emisión de CO₂ equivalente a la producida por una ciudad de 100.000 habitantes. En cuestiones de salud, se afirma que la reducción del estrés y la ansiedad, derivada del ejercicio físico regular, supondría una disminución de hasta un 25% en el absentismo laboral. En Reino Unido, donde se han recogido datos durante los últimos ocho años, de más de 34.000 empresas, con 430.000 bicicletas participantes, cifran en mil millones de libras el ahorro que ha supuesto para el sistema sanitario. Las empresas además han notado que los trabajadores que acuden en bicicleta rinden más y mejor.

2.2.- Sistemas de bicicletas compartidas.

2.2.1.- Definición. Breve historia.

Un sistema de bicicletas compartidas (SBC) consiste en un servicio, promovido normalmente por los ayuntamientos, mediante el cual se pone a disposición del usuario un conjunto de bicicletas para ser utilizadas como medio de transporte urbano, las cuales se recogen y se devuelven en determinados puntos denominados estaciones, siendo una característica propia de estos sistemas el que se pueda recoger y devolver la bicicleta en puntos distintos.

Las bicicletas son empleadas por los usuarios, que ya no necesitan tener una bicicleta en propiedad, para realizar trayectos de corto recorrido como parte de sus desplazamientos por la ciudad, sin que deba entenderse que el servicio tenga fines lúdicos, deportivos o turísticos (para los que ya existen las agencias privadas de alquiler, por ejemplo). Para evitar esto último y asegurar que el servicio se ciñe a su verdadero propósito, suelen imponerse límites de tiempo a partir de los cuales se considera que la utilización del servicio no cumple los fines deseados, en cuyo caso, como medida disuasoria, se imponen recargos a los usuarios. Cosa que no ocurre en un uso normal, en el que el coste es considerablemente más económico.

El primer programa comunitario de intercambio de bicicletas tuvo lugar en 1965 en Ámsterdam. Dentro del contexto reivindicativo y de protesta que caracterizó la década de los 60, el grupo radical holandés *Provo* puso a disposición de todos los ciudadanos y de manera gratuita un conjunto de aproximadamente 50 bicicletas, todas ellas pintadas de blanco (por lo que el sistema se conoció como *Witte Fietsen*, en holandés “bicicletas blancas”). La única condición que se imponía era que al finalizar cada uso se dejara la bicicleta libre (sin candado) para que pudiera ser utilizada por otra persona. Sin embargo la iniciativa fracasó inmediatamente puesto que las bicicletas terminaron siendo estropeadas o robadas por los usuarios. No obstante, el concepto de las bicicletas compartidas caló en la sociedad [[SHA11](#)].

Aunque fuera en la década de los 70 la ciudad francesa de La Rochelle la primera en poner en marcha un sistema permanente, la primera generación de SBC fue liderada por Copenhague, pero estuvo repleta de problemas que condujeron al lanzamiento de una segunda generación de SBC en 1995 [[DEM09](#)]. El nuevo sistema,

Bycyklen, reunía ya una serie de características propias de los SBC actuales como un diseño fácilmente distinguible para las bicicletas, estaciones diseñadas para que las bicicletas puedan cogerse prestadas o devolverse y el pago para desbloquear su uso (todavía mediante máquinas con monedas).

La última generación de SBC, se sirve de la experiencia reunida con las dos anteriores generaciones e incorpora a la gestión las ventajas de las TIC. Complementándose en los últimos años con la utilización de los dispositivos móviles para realizar el pago. En 1998, en la localidad francesa de Rennes se puso en marcha *Vélo à la Carte* (actualmente *Le Vélo STAR*), el SBC pionero en utilizar las TIC [[SHA11](#)]. Actualmente el sistema más conocido de estas características quizás sea el parisino *Vélib'* que opera más de 20.000 bicicletas con más de 1.800 estaciones abiertas 24/7 y distribuidas a lo largo del centro de la capital francesa aproximadamente cada 300 metros [[VEL](#)].

Recientemente, en 2012 el sentimiento de decepción de los ciudadanos de la capital danesa al clausurarse de manera definitiva su SBC, propició el crecimiento del movimiento ya existente denominado *Copenhagenize*, que propugna un modelo sostenible no sólo de transporte sino de urbanismo en su conjunto y que se ha extendido al resto del mundo [[COP](#)]. Mikael Colville-Andersen acuñó el término *copenhaguenizar* para la describir la acción de hacer más respetuosa una ciudad con la bicicleta [[COP2](#)].

2.2.2.- Situación actual.

En la Tabla 2 se muestra la distribución por países del número de SBC, bicicletas y estaciones disponibles en 2010.

Worldwide Bikesharing Programs

COUNTRY	PROGRAMS	BICYCLES	STATIONS
Argentina	1	560	15
Australia	2	2,600	200
Austria	3	1,500	82
Belgium	1	2,500	180
Brazil	2	452	43
Canada	1	6,100	490
Chile	1	150	15
China	19	123,172	4,422
Czech Republic	1	30	16
Denmark	3	2,650	187
France	29	36,830	3,141
Germany	5	13,330	811
Ireland	1	550	44
Italy	19	3,763	362
Japan	1	150	15
Luxembourg	2	400	64
Mexico	1	1,200	90
Monaco	1	10	2
Norway	1	1,660	154
Poland	1	155	13
Romania	1	300	3
Slovenia	1	300	31
South Korea	2	2,031	185
Spain	25	14,048	1,142
Sweden	2	1,500	110
Switzerland	1	600	45
Taiwan	2	5,000	61
United Kingdom	2	12,091	820
United States	4	3,122	313
Total	136	236,754	13,056

Tabla 2.- Sistemas de bicicletas compartidas en el mundo. Fuente: [\[SHA11\]](#).

Como se observa, China aparece en un lugar destacado en cuanto a número de bicicletas y estaciones. Sin embargo, si se compara con el tamaño de su población, es sobre todo en países europeos como Francia o España donde están más implantados. De acuerdo con la compañía alemana de I+D, *Choice GmbH*, es en Europa donde mayor crecimiento se ha producido en los últimos 10 años, poniéndose en marcha más de 400 nuevos SBC [\[GAR11\]](#).

A continuación se relacionan algunos ejemplos destacados de SBC en pleno funcionamiento:

- En España, por su tamaño destacan cuatro ciudades sobre el resto: en Barcelona, *Bicing* con aprox. 6.000 bicis y 420 estaciones [[BIC](#)]; en Sevilla, *SEVICI*, con aprox. 2.500 bicis y 250 estaciones [[SEV](#)]; en Valencia, *Valenbisi*, con aprox. 2.750 bicis y 275 estaciones [[VAL](#)]; y en Zaragoza, *Bizi* con aprox. 1.300 bicis y 130 estaciones [[BIZ](#)].
- En el resto del mundo: los mayores en número son los de Wuhan y Hangzhou, ambos en China con 90.000 y 65.000 bicis, respectivamente. También destacan por número los de París y Shanghai con 20.000 bicis cada una, aunque la capital por excelencia de los SBC es Ámsterdam, donde existe una auténtica cultura de la bicicleta.

Para finalizar, existe un índice que mide el nivel de respeto con la bicicleta que tienen las ciudades, denominado *The Copenhagen Index*, donde Ámsterdam y Copenhague (ver fig. 2) ocupan las dos primeras posiciones del ranking de 2013 y las españolas Sevilla y Barcelona las posiciones cuarta y trece, respectivamente [[TCI](#)].

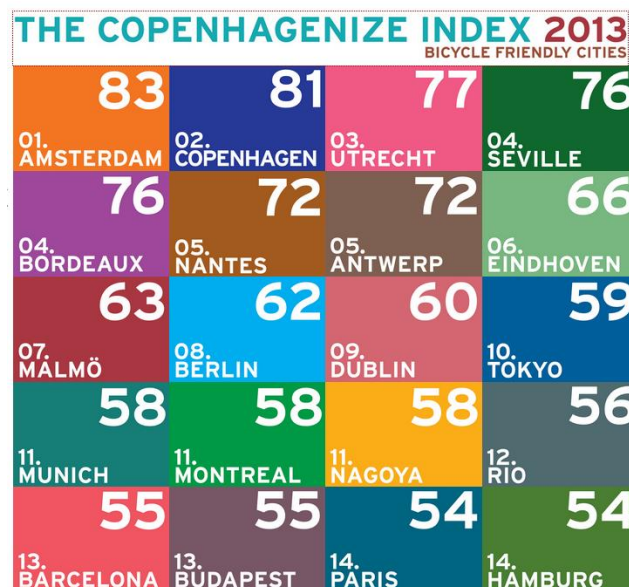


Figura 2.- Ranking *The Copenhagenize Index 2013*. Fuente: [[TCI](#)].

2.3.- *Bicing*. Descripción del servicio y relación con las necesidades de información que puede demandar el usuario de la *app BCN Bicing Info*.

Se trata de un SBC promovido por el Ayuntamiento de Barcelona y en funcionamiento desde marzo de 2007.

Se describen brevemente a continuación las características del servicio para poner en contexto la situación en la que se va a encontrar el usuario al que va dirigida la *app* que se desarrolla en este TFC, y la información que necesitará que se le proporcione.

Utilizar el servicio *Bicing* es muy sencillo. Previamente hay que abonarse (actualmente el coste anual es de 47,16 euros e incluye un seguro), recibiendo a los pocos días la tarjeta de abonado en el domicilio. A partir de ese momento con la tarjeta puede retirarse una bici de cualquier estación y devolverla en la que resulte más conveniente según el trayecto que se desee seguir. Si la bici retirada está en mal estado, se puede devolver en la misma estación dentro de los tres primeros minutos y el sistema asigna otra bici distinta.

Para asegurarse de que el servicio se usa como un medio de transporte y no para un propósito de ocio o turístico los primeros 30 minutos son gratuitos. A partir de ese tiempo, ya no está incluido en el abono anual y hay un recargo de 0.74 euros por cada tramo de 30 minutos hasta un máximo de 2 horas. Superadas estas dos horas la penalización es todavía mayor: 4.49 euros por hora. También hay una normativa de sanciones en caso de no devolverse una bici dentro de las 24 horas siguientes a la retirada de una estación (150 euros). En casos graves se reserva el derecho de excluir temporalmente al abonado para la utilización del servicio, p.ej. cuando se han recibido tres avisos por haber sobrepasado el límite de dos horas. No obstante, están previstas situaciones especiales, como que al llegar a la estación de destino, ésta se encuentre llena y no se pueda devolver la bici, en ese caso al acercar la tarjeta de abonado al lector el sistema amplía en diez minutos el límite inicial de 30 minutos e informa de las estaciones más cercanas con huecos libres.

En cuanto al horario, es bastante flexible. Las únicas horas en donde no funciona la recogida de 02:00 a 05:00 y los viernes de 03:00 a 05:00. Sábados, domingos y festivos está 24 horas. El servicio de devolución está también disponible cualquier día las 24 horas (incluso cuando no está habilitada la retirada).

En la Tabla 3 se pueden ver algunos datos estadísticos extraídos de la *web* de *Bicing* (actualizados a septiembre de 2014).

En el lado negativo, el sistema ha recibido numerosas críticas por el mal estado de algunas bicis (pinchazos, frenos defectuosos, sin cambios de marchas, luces no funcionan, etc.) y con frecuencia las estaciones han sufrido daños por actos vandálicos. Además, con relativa frecuencia el anclaje de la bici en el momento de devolverla no funciona correctamente y el sistema penaliza al abonado al entender que superaba el límite de 30 minutos).

En el lado positivo, se espera que para 2017, se amplíe la flota de bicicletas con unidades eléctricas.

Es de importancia, por tanto para el usuario de *Bicing* que esté empleando la *app* de este TFC, que se le muestre información para ayudarle a localizar tanto la estación más cercana a su ubicación con disponibilidad de bicis cuando necesite recoger una, como encontrar la estación más próxima con huecos libres para devolverla a tiempo y evitar posibles recargos por rebasar el periodo de tiempo gratuito. Esta será la principal utilidad de la *app*.

Información del sistema	
Septiembre 2014	
Servicio Bicing	
96.846	Número de abonados
1.117.632	Número de usos mensuales
10.113.500	Número de usos acumulados año 2014
14,08	Tiempo medio de viaje (minutos)
Bicicletas	
6.000	Número de bicicletas
6,2	Media de usos por bicicleta y día
518,94	Km. recorridos por bici al mes
Estaciones	
420	Número de estaciones
96,89	Porcentaje estaciones sin avería (%)
520	Media de bicicletas reparadas al día
Internet	
71.339	Número de visitas mensuales a la web
5.286	Número de seguidores en Twitter
17.590	Número de fans en Facebook
728	Número de seguidores en Instagram
Top 3 estaciones más usadas del mes	
568 usos/día	42 - C/ Ciutat de Granada 168 / Av. Diagonal
528 usos/día	55 - La Rambla, 80
517 usos/día	79 - Pl. Universidad
Fecha de actualización: 07/10/2014	

Tabla 3.- Estadísticas de *Bicing*. Fuente: [\[BIC\]](#)

Capítulo 3.- Estado del arte II: Open Data – Situación actual y catálogos de datos en abierto utilizables en el proyecto

3.1.- Definición y principios de los datos abiertos.

Los datos abiertos u *Open Data*, en inglés, son cualquier conjunto de conocimientos cuyo acceso, uso y reutilización son libres y sin más restricciones que el reconocimiento de la autoría y el ser compartidos tal y como aparecen, siendo distribuidos abiertamente conforme a una serie características.

Los principios que definen este movimiento son, primero, el libre acceso, uso y reutilización, segundo, sin restricciones, tercero, reconocimiento de los autores, y por último el ser compartidos de la misma forma en que aparecen. Estos principios se detallarán en las características que se relacionan en el [apdo. 2.4](#) de este capítulo.

3.2.- Excepciones. Qué no se considera *Open Data*.

Por haber sido ya tratado en otros movimientos semejantes, no se consideran incluidos dentro de esta filosofía los dos siguientes conceptos:

- El software. Por existir ya las iniciativas de software libre y código abierto, auspiciadas por las organizaciones *Open Source Initiative* y *Free Software Foundation*, respectivamente, y motivadas por el proyecto GNU (acrónimo recursivo de *GNU is Not Unix*) y con un sistema propio de licencias (GPL, BSD, etc.) [[GNU](#)].
- Las publicaciones - Por existir ya *Open Access* o acceso abierto, un movimiento que promueve el libre acceso a contenidos y publicaciones digitales, como material educativo (p.ej. el movimiento *OpenCourseWare* creado por el MIT – *Massachusetts Institute of Technology* [[OCW](#)]), artículos científicos y académicos, así como libros y revistas. Sobre este punto no hay común acuerdo y hay entidades como la fundación *Open Definition* que sí consideran estos datos como *open data* [[ODE](#)].

3.3.- Ejemplos. Qué sí se considera *Open Data*.

Se consideran datos abiertos especialmente la información geográfica, de transporte, financiera, medioambiental, meteorológica, histórica, académica y científica. Conocimientos que tradicionalmente han estado en manos de organismos e instituciones públicas y por tanto financiados por los ciudadanos o bien conocimientos que aunque hayan sido recopilados o investigados por entidades privadas sean objeto de un bien común como, por ejemplo, el genoma humano.

3.4.- Características.

De acuerdo con la fundación *Open Definition* [[ODE2](#)] para que un conocimiento se considere abierto su distribución debe cumplir una serie de características que citamos a continuación:

- **Acceso** – Debe estar disponible de manera íntegra y su consulta ha de ser gratuita o a un coste razonable, preferiblemente descargable por Internet. El formato será el más conveniente para facilitar su tratamiento o modificación.
- **Redistribución** – La licencia de distribución no ha de restringir la venta o redistribución posterior, ya sea de manera individual o formando parte de un conjunto, ni exigir ningún tipo de contraprestación económica por ello.
- **Reutilización** – La licencia no debe prohibir hacer modificaciones o versiones derivadas, ni impedir que éstas sean distribuidas en las mismas condiciones que el original. En cualquier caso se permite que haya algún tipo de obligación en relación al reconocimiento de la autoría o a la integridad del original.
- **Sin restricciones tecnológicas ni económicas** – Se debe facilitar la distribución de manera que no existan obstáculos tecnológicos para su consulta, tratamiento y redistribución. Se tendrá especial cuidado en el formato usado, de manera que sea abierto, es decir que esté disponible gratuitamente y de forma pública.
- **Reconocimiento** – La licencia puede requerir para la redistribución o reutilización el reconocimiento de sus autores y contribuyentes, sin que esta exigencia vaya acompañada de contraprestación alguna. En este caso se debe facilitar una relación de los reconocimientos que deben incluirse.

- **Integridad** – La licencia puede exigir que las modificaciones que se realicen no utilicen el mismo nombre que el original o que contengan un número de versión diferente.
- **Sin discriminación** – La licencia no ha de discriminar el acceso a ninguna persona o grupo de personas.
- **Sin restricción de campo de trabajo o finalidad** – La licencia no debe restringir la utilización a un campo o ámbito determinados de trabajo o una finalidad concreta.
- **Distribución de la licencia** – Los derechos incluidos en la distribución se aplican a cualquier redistribución sin que sea necesario incluirlos de nuevo.
- **Licencia conjunta no específica** – Cuando se distribuye formando parte de un conjunto, la licencia no debe depender del paquete concreto en el que está incluida ya que en cualquier momento se puede extraer de este conjunto y ser tratada de manera individual conservando los derechos de la licencia del paquete original.
- **No restricción en distribuciones conjuntas** – La licencia no ha de prohibir o imponer restricciones a las otras obras que formen parte del paquete, sean o no estas otras obras *open data*.

3.5.- Leitmotiv, puesta en valor, beneficios y tipos de datos abiertos.

Especialmente significativos son los datos abiertos de origen gubernamental (en inglés, *open government data*). En aras de unos principios popularmente demandados de una mayor transparencia y control democrático de los gobiernos, los datos abiertos permiten, por ejemplo, conocer en qué se gastan los impuestos de los contribuyentes, controlando su buen fin, denunciando comportamientos y desembolsos para fines no deseados (como p.ej. excesiva emisión de gases contaminantes, actuaciones inmobiliarias ilegales o deducciones de impuestos fraudulentas).

Esto también redundará en beneficio de los propios organismos públicos puesto que las críticas que reciben a modo de *feedback* les ayudan a mejorar su eficacia y eficiencia.

Asimismo se favorece la participación del individuo en la mejora de la sociedad, asumiendo un papel más activo y comprometido frente a su rol de pasividad actual.

En la parcela no gubernamental, promueve el emprendimiento y la innovación, permitiendo a empresas y autónomos que utilicen los datos abiertos para una idea de negocio que aporte valor y beneficio.

Facilita el desarrollo de la investigación dentro de la comunidad científica y académica, permitiendo el acceso y tratamiento de un volumen de datos ingente compartido de manera global.

De igual modo aporta una mayor calidad de vida, haciendo más fácil el día a día de los ciudadanos, puede pensarse, p. ej., en los servicios de tráfico, poniendo a disposición de los conductores las incidencias en las carreteras. Por no mencionar el impacto que tiene especialmente en los colectivos más desfavorecidos, aquí puede imaginarse, p.ej., en la utilidad que tiene para un discapacitado el tener acceso a la hora de salir a la calle a una relación con la ubicación de las actuaciones llevadas a cabo en su ciudad que no supongan barreras en su desplazamiento (señalando los pasos de peatones sin bordillo, ascensores para sillas de ruedas, aparcamientos para discapacitados, etc.).

A continuación se enumeran algunos tipos de datos abiertos que destacan por su relevancia (ver fig. 3):

- Información geográfica
- Científicos y académicos
- Estadísticos
- Meteorológicos
- De transporte y tráfico
- Medioambientales
- Financieros



Figura 3.- Fuente diagrama [\[ODE\]](#)

3.6.- Catálogos de datos abiertos. Censo actual de catálogos *Open Data*. Situación en España y catálogos utilizables para el proyecto.

La página *Open Data Census* mantenida por la *Open Knowledge Foundation* [OKF] proporciona un censo de los conjuntos de datos que los países y gobiernos distribuyen como *Open Data*. En el mes de octubre de 2014 muestra un total de 250 países y más de 850 conjuntos de datos aunque apenas superan el centenar los considerados como datos abiertos por esta entidad (por considerar que no cumplen con todos los criterios para considerarse datos abiertos).

Dentro de este recurso, existe una clasificación por países en función de la puntuación que se les asigna según criterios determinados por esta fundación [OKF2]. Se desglosan datos por cada país y por cada área temática, detallando para cada

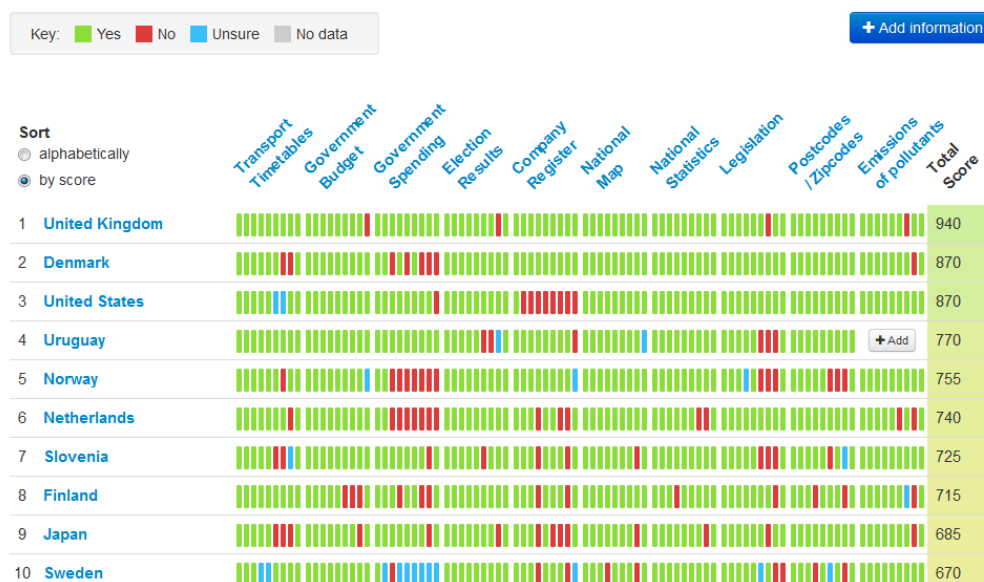


Figura 4.- Ranking *Open Knowledge Foundation* por país. Fuente: [OKF2]

conjunto de datos las características de la definición de datos abiertos que se cumplen (en verde) y las que no (en rojo) así como las que tienen dudas (en azul). En el ranking, cuyos 10 primeros puestos se pueden ver en la fig. 4, aparece en cabeza Reino Unido (1), Dinamarca (2) y Estados Unidos (3). En estas primeras posiciones se encuentran fundamentalmente el resto de países de influencia británica (Nueva Zelanda (12), Australia (11), Canadá (15)) y resto de países nórdicos (Noruega (5), Finlandia (8), Suecia (10)), por delante de los países de Europa occidental (excepción de Países Bajos (6) que también ocupa primeras posiciones). España ocupa la posición 32. Se observa

que hay una evidente relación entre el nivel de desarrollo de los países y su posición en este ranking aunque, por otra parte, llaman la atención las posiciones tan adelantadas de Eslovenia (7) y Uruguay (4), síntoma de que sus gobiernos están haciendo un especial esfuerzo en destacar en este aspecto.

Por otro lado, también resulta muy útil la herramienta de búsqueda que mantiene la fundación CTIC [[CTI](#)], con un mapa donde aparecen localizados los catálogos disponibles, que se pueden filtrar por diferentes criterios (región, país, formato, entidad gestora). En el caso de España aparecen 41 catálogos. La fundación CTIC otorga estrellas en función de las características de los catálogos, los que obtienen las mayores puntuaciones son los de *Opendata Cáceres* y *Zaragoza Public Data Catalogue* que obtienen cinco estrellas, entre los merecedores de cuatro estrellas se encuentran el *Catalonia Open Data Catalog* y el *OpenData BCN* de Barcelona. A nivel del Gobierno de España la entidad que se encarga de gestionar los datos es el Ministerio de Industria, Energía y Turismo promoviendo el movimiento *Open Data* a través del proyecto *Aporta* [[GOB](#)].

Para la realización del presente proyecto se utilizará fundamentalmente el catálogo de datos abiertos del Ayuntamiento de Barcelona, *OpenData BCN* [[BCN](#)], y secundariamente el catálogo de la Generalitat de Catalunya [[CAT](#)].

Así, en el catálogo *OpenData BCN*, dentro del recurso “Estaciones de *Bicing*”, de actualización constante y en formato XML, están disponibles los siguientes datos para cada estación de bicicletas que son fundamentales para la *app* de este TFC: coordenadas de la estación (longitud y latitud), dirección (calle y número), listado de estaciones más próximas, estado de la estación, número de aparcamientos libres (huecos) y número de bicis disponibles [[BCN2](#)].

En el catálogo de la Generalitat de Catalunya, tenemos los recursos:

- “Incidencias del transporte público en el área de Barcelona”, de actualización diaria y en formato RSS [[CAT2](#)];
- “Información aerobiológica de Catalunya: predicción de los niveles de polen y esporas alergógenos”, de actualización semanal y en formato PDF [[CAT3](#)];

- “Predicción metereológica de Catalunya”, actualizado tres veces al día y en formato JSON [[CAT4](#)].

Estos recursos pueden servir para que la *app*, muestre al usuario de *Bicing* información el fin de que pueda tomar decisiones al respecto, como p.ej. decidir no ir una mañana en coche al trabajo si se le advierte de que hay una incidencia de tráfico importante y prefiere acudir en bici; o si es alérgico y quiere que se le avise si los niveles de polen son altos y le dificultarán el trayecto; o bien si desea estar alertado de si se esperan tormentas por la tarde y opta por no usar *Bicing* (o sí lo hace, pero está prevenido para protegerse de la lluvia).

Capítulo 4.- Estado del arte III: Sistemas operativos y aplicaciones para dispositivos móviles. Plataforma *Android*

4.1.- *Android*

4.1.1.- Qué es *Android*.

Android es un sistema operativo (SO) para dispositivos móviles basado en Linux. El concepto comprende más que el hecho de ser un SO, *Android* es además un entorno, *framework*, de desarrollo de aplicaciones.

Actualmente *Android* es el SO más popular del mundo, es capaz de ejecutarse en todo tipo de dispositivos, desde *smartphones* y *tablets* hasta *smartwatches* y *Google Glasses* pasando por lectores de *ebook*, vídeo-consolas o televisores UHD.

4. 1.2.- Origen. Breve aproximación histórica.

El principal promotor de este SO es sin duda Google, que en 2005 adquiere la empresa Android Inc. [AND], operación de compra que apenas tuvo relevancia mediática en aquel entonces, y que constituía un movimiento estratégico de gran calado en la multinacional californiana, previendo en ese momento el inminente despliegue de un futuro escenario de *smartphones* y aplicaciones móviles. Como así ha sido. La apuesta que hizo la compañía con sede en Mountain View, con esa visionaria decisión, ha sido recompensada ampliamente.

No obstante, si bien se trata de un sistema desarrollado inicialmente por Google, desde bien temprano su desarrollo está gestionado por la fundación *Open Handset Alliance* [OHA].

Este grupo, OHA, es un consorcio de aprox. un centenar de empresas del sector de la informática y telecomunicaciones, promotoras de estándares abiertos para dispositivos móviles. Entre estas empresas se encuentran por ejemplo operadores móviles de primera fila como Vodafone, T-Mobile, China Mobile o Telefónica, fabricantes de semiconductores como Intel, NVIDIA o Texas Instruments, fabricantes de dispositivos móviles como Asus, HTC, LG, Motorola, Samsung, Sony-Ericsson o Toshiba, y empresas de servicios de software como la propia Google o eBay.

Según reza la página oficial de la OHA, los objetivos de este grupo de empresas son, por un lado, proporcionar un mayor ritmo de innovación en los dispositivos móviles de telecomunicación y, por otro lado, ofrecer a los usuarios una alternativa gratuita y mejor.

Esta estrategia recuerda, salvando las distancias, el conocido caso ocurrido a principios de la pasada década de los ochenta con Sony y su sistema de video doméstico Betamax, que se estudia hoy día en las escuelas de negocio. Y es que, aunque el de Sony era un sistema de vídeo superior al de su competidor VHS, fue éste y no aquél el gran triunfador del mercado, en el que no hubo sitio para dos formatos. La siguiente vez, Sony ya había aprendido la lección y no hace mucho hemos sido testigos de cómo su formato para DVD de alta capacidad *Blu-ray* barrió al de su competidor. Esto fue posible gracias a una excelente estrategia de mercado con la que llegó a acuerdos con los principales estudios de cine y empresas de distribución de películas. Mediante esta comparación se trata de explicar la política de alianzas que ha seguido Google y que le ha valido para aupar a *Android* en una posición, como se verá a continuación, de dominio de mercado.

4. 1.3.- Características diferenciadoras de *Android*.

Fundamentalmente se trata de un sistema operativo de código abierto (*open source*). De manera que el ritmo de innovación en el desarrollo de aplicaciones es de semanas y no de meses o incluso años como ocurre con los sistemas que no son de código abierto. Sencillamente cuando alguien tiene una nueva idea no tiene más que publicarla y todo el mundo puede contribuir a su desarrollo, por lo que, como es evidente, la velocidad de corrección de errores y de innovación es mucho mayor.

Se trata de un sistema operativo totalmente preparado para funcionar con los más modernos *smartphones* de última generación, los cuales incorporan un sinfín de nuevas características como cámaras, GPS, acelerómetros, brújulas y otros sensores. Además es relativamente sencillo desarrollar aplicaciones para *Android*, y sobre todo, es gratis.

Esto último cobra una gran importancia teniendo en cuenta el hecho de que cuando los fabricantes de dispositivos móviles lanzan un nuevo terminal al mercado, están soportando también el coste del desarrollo del sistema operativo, que antes de la

aparición de *Android* llegaba a alcanzar hasta el 25% del coste total del terminal. Se suele denominar sistemas operativos comerciales o propietarios a los desarrollados por los fabricantes, que son no *open-source*, para diferenciarlos de los *open-source* o de código abierto. De esta manera se entiende uno de los fines perseguidos por la citada alianza, cuando este grupo de empresas se decidió a apoyar un código abierto para dispositivos móviles buscaba un significativo ahorro de costes. Conviene recordar que además de una innovación tecnológica también supone un negocio.

Como novedad, *Android* fue lanzado como un sistema operativo para dispositivos móviles totalmente nuevo, el único que ha sido desarrollado completamente desde cero basado en Linux 2.6 y que es de código abierto u *open-source*, es decir gratuito. La inmensa mayoría del código fuente de este sistema operativo ha sido publicado bajo licencia de software Apache, libre y abierta.

Para intentar responder a la cuestión de porqué Google decidió desarrollar entonces un sistema operativo para dispositivos móviles en vez de para ordenadores personales, se puede afirmar que muy inteligentemente detectaron el inicio de un cambio de tendencia en los hábitos de los usuarios que les llevó a apostar por la tecnología móvil.

Estos cambios en el comportamiento de los usuarios han tenido lugar en apenas media década y se ponen de manifiesto en algunos datos que se exponen a continuación extraídos del último estudio de la Fundación Telefónica, el informe *Sociedad de la Información en España 2013* [SIE]. Ya existen en el mundo 6.800 millones de teléfonos móviles, casi tantos como personas, 7.100 millones. España es además el país europeo con mayor número de *smartphones*, 26 millones. Las *tablets* tampoco se quedan cortas y sus ventas a nivel mundial ya superan a las de los ordenadores portátiles. Los usuarios están conectados permanentemente, los españoles miran el móvil una media de 150 veces al día. Los usuarios emplean los móviles para realizar búsquedas antes de efectuar compras, ocupan sus tiempos muertos mientras viajan en el transporte urbano, o consultan alguna duda que les surge incluso teniendo al alcance de la mano un ordenador con conexión. Entre los motivos que comentan los usuarios para emplear teléfonos inteligentes está el de dar una imagen moderna. ¿Qué será lo próximo? Ya se da por hecho que 2014 es el año en el que el “Internet de las cosas” se incorporará a la vida cotidiana, por medio de los dispositivos *wearables*, como los relojes y las gafas

inteligentes. ¿Qué se puede esperar en el futuro? Algunas ideas, que rozan la ciencia ficción, como los implantes subcutáneos, se comentan ya abiertamente en foros como el Congreso TIC de La Rioja, celebrado en abril de 2014 [[TIC](#)].

Android ha sido concebido como una plataforma de desarrollo de software para estos dispositivos móviles y permite a los desarrolladores implementar sus programas en Java (lenguaje de programación de *Android* por antonomasia) y controlar estos dispositivos por medio de bibliotecas desarrolladas o adaptadas por Google. Aunque, en principio es posible el empleo de otros lenguajes distintos a Java, como p.ej. C, su uso no está soportado oficialmente por Google.

Se ha mencionado que una de las principales características de *Android* es que está estructurado como soporte *framework* de aplicaciones. Es decir como un marco de trabajo para el desarrollo de software, constituyendo una estructura de soporte definida, mediante la cual un proyecto de software puede ser organizado y desarrollado incluyendo soporte de programas, bibliotecas y un lenguaje implementado para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Hasta la última versión 4.4.4, *Android* utiliza una máquina virtual *Dalvik* para dispositivos móviles. De acuerdo con la propia página de *Android* [[AND2](#)], *Dalvik* es la máquina virtual que utiliza esta plataforma, ha sido diseñada por Dan Bornstein con contribuciones de otros ingenieros de Google. Esta máquina virtual está optimizada para requerir poca memoria y está diseñada para permitir ejecutar varias instancias de la máquina virtual simultáneamente, delegando en el sistema operativo subyacente otras tareas. A menudo *Dalvik* es nombrada como una máquina virtual Java, pero esto no es estrictamente correcto, ya que el *bytecode* con el que opera no es Java *bytecode*. Sin embargo, la herramienta *dx* incluida en el SDK de *Android* permite transformar los archivos Class de Java compilados por un compilador Java al formato de archivos Dex, que son los que usa *Dalvik*.

También lleva integrado un navegador de Internet basado en el motor de código abierto WebKit, es el mismo *framework* para aplicaciones que funciona como base p.ej. para el navegador web Safari (el mismo que el iOS de Apple), Midori (de Xfce) y Google Chrome.

Android utiliza un sistema de gestión de bases de datos relacional o SQL (*Structure Query Language*), y al igual que en el caso del navegador, se ha optado también por un sistema de código abierto llamado SQLite para almacenamiento de los datos estructurados. Pero, a diferencia de los demás sistemas de gestión de base de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina *host*. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

Android es un sistema operativo del siglo XXI y soporta la mayoría de formatos imágenes, audio y vídeo con los que funcionan la última generación de terminales de telefonía móvil. Además soporta telefonía móvil GSM, EDGE, 3G, 4G, y toda clase de conectividad, Bluetooth, WiFi y NFC. Está preparado para explotar toda la capacidad de los últimos *smartphones* del mercado, que incluyen cámara digital con fotografía y vídeo en Full HD, incluso en 3D, pantalla multitáctil, GPS, acelerómetro y giroscopio, brújula y otros sensores como de proximidad y de iluminación ambiental.

A semejanza de Apple y su *App Store*, *Android* posee su propio sitio de descargas y desarrollo de aplicaciones denominado Google Play [[GPY](#)], de manera que los desarrolladores pueden publicar sus aplicaciones gratuitas y de pago, con herramientas para el seguimiento de su publicación, y los usuarios pueden descargarlas y realizar comentarios y recomendaciones.

4. 1.4.- Arquitectura de *Android*. Breve descripción.

Android está basado en Linux 2.6 para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red, y modelo de *drivers*. Como podemos ver en la fig. 5, el núcleo o *kernel* de Linux (en color rojo), con sus diferentes controladores o *drivers* para p.ej. el teclado, la cámara o un puerto USB, actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

Sobre este *kernel* o núcleo de Linux, hay diferentes librerías (en color verde) como las del navegador Webkit, o del sistema de gestión de bases de datos SQLite, o de los gráficos OpenGL|ES, entre otras. Al lado de estas últimas se encuentra el *Android runtime* o entorno de ejecución. Una vez que se ha escrito un programa en lenguaje Java, se genera un archivo que se ejecuta en este entorno, el cual está optimizado para sistemas que requieren un bajo consumo de energía, como es el caso de los teléfonos móviles.

Finalmente, encima del todo están las librerías de aplicaciones y los sistemas de servicios como el del teléfono que gestiona las llamadas entrantes.

Todas las aplicaciones de *Android* se ejecutan con sus propios procesos, esto quiere decir que cuando una aplicación falla, el teléfono no se queda “colgado” sino que sigue funcionando a pesar de ello. Además permite que una tarea que se esté ejecutando en ese momento se quede “dormida” al comenzar a ejecutarse otra de repente. Esto

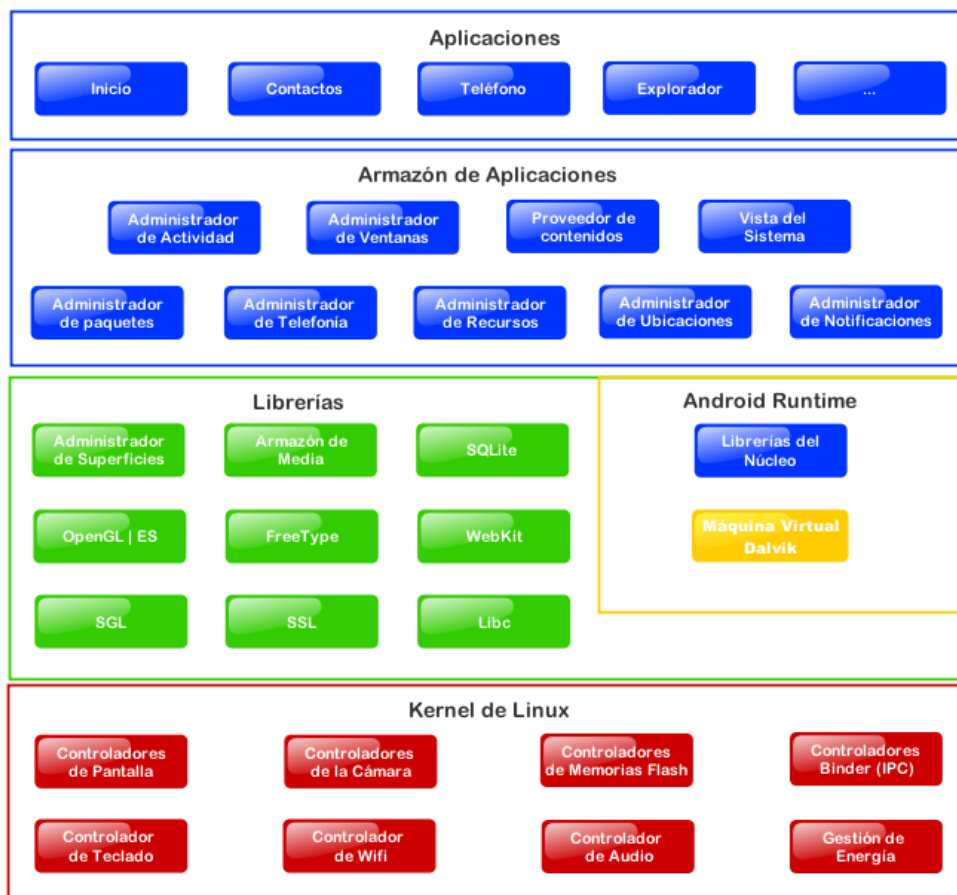


Figura 5.- Arquitectura de *Android*. Fuente [[AND2](#)]

ocurre cuando, por ejemplo, el usuario está jugando y recibe una llamada entrante, atendería la llamada y una vez terminada volvería al juego exactamente en el mismo punto donde lo había dejado en el momento de recibirse la llamada.

Al tratarse de una arquitectura abierta que utiliza el lenguaje de programación Java, esto permite diseñar aplicaciones todo lo complejas que se quieran y adecuar mayor o menor complejidad en función de las propias prestaciones del terminal, así se pueden diseñar aplicaciones más sencillas para terminales básicos y aplicaciones más avanzadas para los últimos *smartphones* que se lancen al mercado.

Una característica importante de *Android* es que el sistema se encarga en cada momento de elegir qué componente usar para cada tarea ya que no tiene por qué haber componentes únicos sino que pueden ser intercambiables. Por ejemplo, no tiene por qué utilizarse siempre el mismo programa para abrir una fotografía, sino que puede haber varios programas que realicen esta tarea, y podemos intercambiarlos.

4. 1.5.- Situación actual.

Justo cuando se están redactando estas líneas, se están produciendo una serie de acontecimientos de gran trascendencia en el desarrollo del SO *Android*, y que suponen un cambio importante en algunos de los puntos que se han descrito en el apartado anterior.

El pasado diciembre, Google realizó el lanzamiento de la última versión de *Android*, que han decidido denominar *Android 5.0 Lollipop*. El hecho de que se emplee un número nuevo de versión (5.0) ya nos da idea de que los cambios con respecto a la versión anterior son de gran calado.

A primera vista destacan dos aspectos. Por una parte, que por primera vez en la historia de *Android* se cambia la máquina virtual por defecto, que hasta ahora siempre había sido Dalvik, por una nueva denominada ART, que promete una experiencia de usuario más fluida y ágil. De acuerdo con la página oficial de *Android*, la diferencia con respecto a la máquina virtual anterior es que, mientras con Dalvik el código se ejecutaba a medida que se necesitaba (*JIT Compiler*), en la nueva ART se hace antes de que el usuario lo necesite (*AOT Compiler*). Por otra parte, esta nueva versión da soporte a los

nuevos procesadores de 64 bits que acaban de irrumpir en el mercado de *smartphones* y cuyo lanzamiento va a coincidir con el de esta nueva versión de Android.

4. 1.6.- Cuota de mercado.

De acuerdo con los últimos datos publicados por *International Data Corporation*, IDC, correspondientes al segundo trimestre de 2014, el reparto de la cuota de mercado actual de sistemas operativos (SO) para teléfonos móviles inteligentes, también conocidos como *smartphones*, es la que se muestra en la Tabla 3. Como se observa, la posición de *Android* sigue siendo aún más dominante que a finales de 2013 y el único SO que crece después de que *Windows Phone*, que parecía empezar a recuperar terreno, se estancara a comienzos de 2014 [IDC].

Period	Android	iOS	Windows Phone	BlackBerry OS	Others
Q2 2014	84.7%	11.7%	2.5%	0.5%	0.7%
Q2 2013	79.6%	13.0%	3.4%	2.8%	1.2%
Q2 2012	69.3%	16.6%	3.1%	4.9%	6.1%
Q2 2011	36.1%	18.3%	1.2%	13.6%	30.8%

Source: IDC, 2014 Q2

Tabla 4.- Cuota de mercado de los principales SO móviles, 2º trimestre 2014. Fuente: *IDC Worldwide Quarterly Mobile Phone Tracker* [IDC].

En los últimos años se ha vivido un verdadero *boom* en la aparición de aplicaciones móviles. En las tiendas oficiales de *apps* como *App Store* de *Apple*, *Google Play* de *Android* (anteriormente llamada *Android Market*), y las tiendas de *Windows Phones* y otros hay disponibles más de 2.5 millones de aplicaciones. Además existen 10 millones de aplicaciones fuera de estas *stores* oficiales, de las que se estima que el 15% contienen *malware*, según indica la empresa *Kaspersky Lab* [KAS].

Esta inusitada proliferación no siempre es sinónimo de calidad ni de éxito. Conforme a un estudio de esta empresa, más de la mitad de las aplicaciones que se pueden descargar actualmente se consideran muertas, es decir, no tienen actualizaciones, ni registran comentarios, ni descargas. Se cifra en 700.000 el número de aplicaciones de este tipo entre el total ofertado por todas las plataformas en las tiendas oficiales. La que

mayor índice de abandono tiene es Windows (69%), seguida de *iOS* (65%) y *Android* (41%) [ELP].

El rápido crecimiento en cuota de mercado de los dos principales competidores, *iOS* y *Android*, supuso en un primer momento una bipolarización de facto en el ámbito de las aplicaciones móviles, relegando a terceras plataformas a un papel menor como el caso de *BlackBerry* (muy tocada desde el último colapso accidental de su red de mensajería), o a una caída en picado desde posiciones anteriormente de liderazgo sólido en el sector, como el caso de Nokia (líder durante años en ventas de móviles) y su acuerdo con Microsoft (líder en ventas de SO para PC). La gran beneficiada sin duda ha sido *Android* que ocupa una posición de dominio, aunque la previsión es que tanto *iOS* como Windows recuperen cuota en el medio plazo (según la previsión de IDC), siendo la gran perdedora *BlackBerry*.

En el caso de *Android*, sus principales puntos fuertes son que se trata por un lado de un SO de licencia libre (licencia Apache) y, por otro lado, que tiene detrás un consorcio de empresas llamado *Open Handset Alliance* muy importante que está liderado por Google y que le confiere una situación de mercado inmejorable (con socios como Motorola, Sony-Ericsson, Samsung, HTC, etc.).

En el caso de *Apple*, su fortaleza es su innovador diseño y sus sofisticadas prestaciones, y también la fidelidad del cliente que ha ido ganando a lo largo de muchos años y que ahora mismo se ve plasmada en tres productos móviles que han sido revolucionarios cada uno de ellos en el momento de su aparición: *iPod Touch*, *iPhone* y *iPad*.

4. 1.7.- Idoneidad para el proyecto.

Este TFC se ceñirá a la plataforma *Android*, intentando ser también un básico tutorial introductorio a la programación en este entorno. Se ha preferido al de *Apple* por su filosofía de código abierto y por el rápido crecimiento experimentado en los últimos años lo que lo ha catapultado como el SO móvil más popular, así como su disponibilidad en terminales de diferentes fabricantes de móviles y operadoras de telefonía, que puede hacerlo más accesible, sin precisar realizar, p.ej. un importante desembolso económico para adquirir un ordenador *Mac* (única manera de desarrollar aplicaciones para *iOS*).

Pero sobre todo por el verdadero factor diferencial de *Android* frente a sus competidores, que es el control absoluto por parte del usuario del software que se ejecuta en su móvil.

4.2.- Herramientas y técnicas de desarrollo disponibles

Una posible clasificación de las aplicaciones para dispositivos móviles sería la siguiente [[PRI](#)]:

- Aplicaciones básicas. Son aplicaciones muy sencillas en las que la interacción con el dispositivo consiste en el mero intercambio de información.
- *Webs* móviles. Son adaptaciones de *webs* que ya existen actualmente para que puedan ser visualizadas en dispositivos móviles.
- Aplicaciones webs sobre móviles. Son aplicaciones que no precisan estar instaladas en el dispositivo para ejecutarse. Están basadas en tecnologías HTML 5, CSS3 y JavaScript que corren en un navegador.
- Aplicaciones *web* móviles nativas. Un término medio entre las aplicaciones *webs* y las aplicaciones nativas.
- Aplicaciones nativas. Son propias de cada plataforma y aprovechan al máximo el potencial de la misma y de los dispositivos proporcionando la máxima experiencia al usuario.

Encuadraríamos la *app* de este TFC dentro de la última clase, aplicaciones nativas. Naturalmente se podría discutir si no es la única posibilidad, pero se entiende que sí es la más sencilla y es la que se suele emplear cuando se comienza a desarrollar un proyecto por facilitar la realización de pruebas.

4.2.1.- SDK oficial de *Android*.

Para desarrollar aplicaciones se emplea el **SDK** (*Software Development Kit*) oficial de *Android* [[SDK](#)]. Las herramientas más útiles que comprende este kit son:

- Sqlite3: para gestionar bases de datos mediante SQL;
- Hprof-conv y dmtracedump: para detectar posibles problemas de memoria o CPU en la ejecución, mejorando el rendimiento de la aplicación (perfilado o *profiling*);
- Layoutopt y Draw 9-patch: para mejorar y analizar el rendimiento de la interfaz gráfica;
- Monkey y monkeyrunner: para realizar tests de estrés de la aplicación;
- Zipalign, Proguard: para mejorar los ficheros de la aplicación,

- Emulador: un emulador que nos sirve de banco de pruebas para testear las aplicaciones,
- Logcat: para visualizar los archivos .log de sistema;
- Android: gestor de dispositivos virtuales;
- Abd: para visualizar el estado del emulador;

4.2.2.- IDE Eclipse, entorno de desarrollo utilizado. *Google Maps*.

Todas estas herramientas se agrupan en un *plugin* denominado **ADT Bundle** (*Android Developer Tools*) que tiene soporte oficial para el IDE Eclipse, de uso generalizado en desarrollo de aplicaciones en multitud de lenguajes de programación y plataformas y empleado de modo profesional en desarrollo de proyectos informáticos y que será el que se utilice para implementar la *app* de este TFC [[ADT](#)].

Para incrustar mapas en la *app* se emplea *Google Maps Android API v2* [[MAP](#)].

En principio, se utilizará *Apache* como servidor para la BD que se usa para realizar el análisis de tendencias.

4.2.3.- Otras herramientas y aspectos a tener en cuenta.

Para la edición y composición de la memoria del TFC se empleará el sistema *LaTeX* con su herramienta para la bibliografía *BibTEX*. También, el gestor de bibliografía *JabRef*. Para el mapa conceptual el programa *FreeMind*. Todos ellos de software libre.

Entre otros aspectos a tener en cuenta, está el de la versión del SO. La última versión de *Android* disponible al comenzar el proyecto era la 4.4.4 *KitKat* (como curiosidad las diferentes versiones del SO reciben nombres de postres o dulces siguiendo un orden alfabético). Apenas llega al 25% el porcentaje de los terminales *Android* que tienen instalada la versión *KitKat* 4.4 o superior. La mayoría de los usuarios tiene instaladas versiones 4.1 y 4.2 (un 45% aprox.), además alrededor del 12% de los usuarios tiene versiones iguales o anteriores a la 2.3, como puede verse en la fig. 6, y es que, muchas veces, el ritmo de actualización del SO supera al ciclo de vida de los terminales, en especial en tiempos de crisis, cuando se prolonga más su uso. Por este motivo, es importante tener en cuenta este hecho a la hora de diseñar una aplicación y hacerla compatible con versiones antiguas [[DIS](#)].

No obstante, aunque este es un tema crítico a la hora de desarrollar una aplicación profesional, no se le dará excesiva importancia en este TFC, puesto que para el desarrollo del proyecto se utilizará fundamentalmente el emulador en el que se pueda hacer correr la versión que deseemos del SO. Este mismo criterio se sigue por ejemplo a la hora de tener en cuenta la resolución de pantalla para la que se trabaja y otros detalles similares.

En cualquier caso, aunque la herramienta para mostrar el funcionamiento de la *app* en la presentación y defensa del proyecto sea el emulador, se tratará de instalar en un terminal real, en este caso el terminal elegido es un Samsung Galaxy S3 con 4.3 *Jelly Bean*.

En cuanto a la publicación en Google Play, para su permitir su descarga, a priori es una idea que no se baraja aunque se tratará en la memoria o en un anexo de explicar los pasos que debe dar un desarrollador para publicar y mantener sus *apps*.

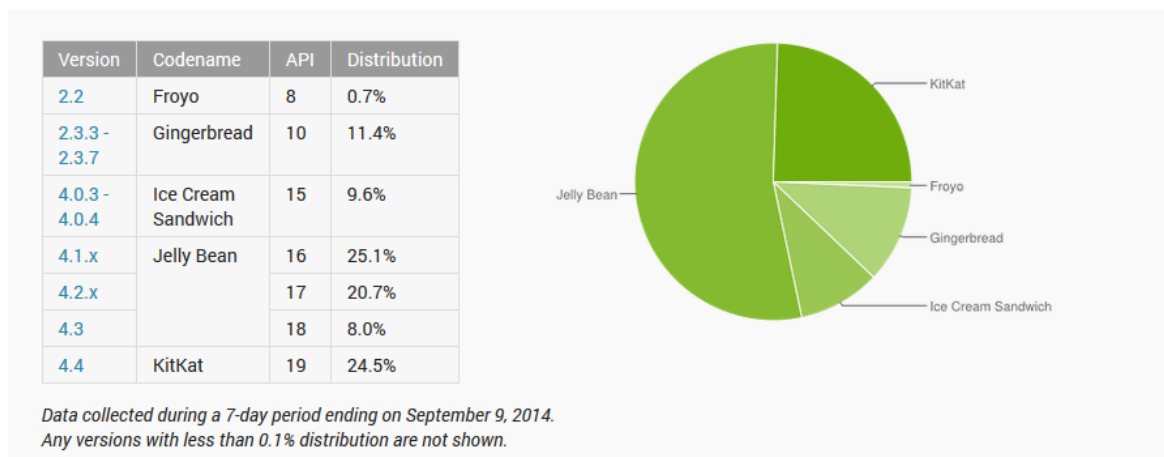


Figura 6.- Distribución de las versiones de *Android*, Actualizado a fecha 9 septiembre 2014.

4.3.- Aplicaciones similares. *BCN Bicing Info* vs. *App Bicing* oficial

Además de la *app* oficial de *Bicing*, hay numerosas aplicaciones que podemos encontrar en Google Play de utilidad sobre este mismo tema [OFI].

La oficial, con cerca de 100.000 descargas, y actualmente en su versión 2.1 (actualizada en marzo 2014 por última vez, aunque únicamente para incluir publicidad de una operadora de telecomunicaciones), destaca por ser muy completa y disponer además de versiones para las plataformas *Android*, *iOS*, *Windows Phone* y *BlackBerry*. Los requisitos mínimos para *Android* son que éste sea de la versión 2.2 en adelante. Las funcionalidades básicas que implementa son que permite saber la disponibilidad de estaciones y bicicletas de *Bicing* en tiempo real. También calcula la ruta más segura para desplazarse de una estación a otra. La interfaz de usuario está muy cuidada (ver fig. 7).

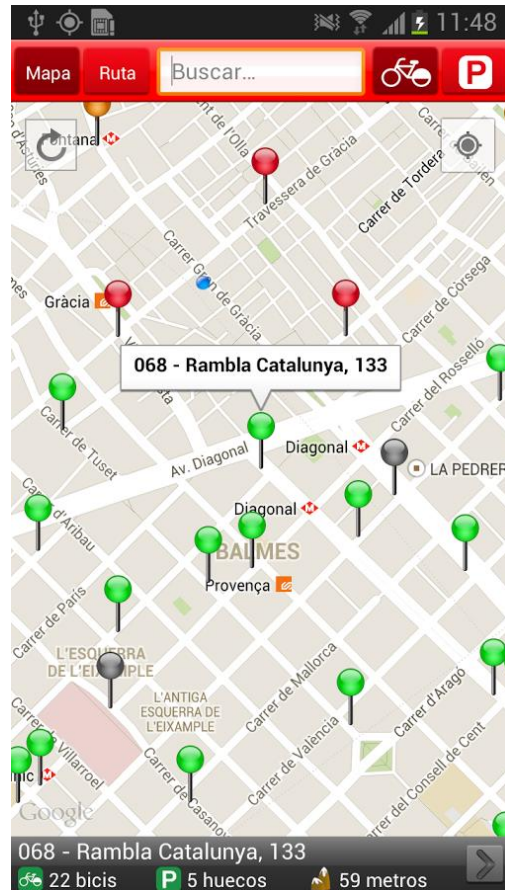


Figura 7.- Muestra de la *app* oficial *Bicing*. Fuente: *Google Play*

Entre el resto, aunque hay alguna *app* específica para *Bicing* (como *Bicing Barcelona* [BIB] muy sencilla y con escasa popularidad), destacan las que proporcionan información para varias ciudades dentro de la misma aplicación, como *Cycle Hire*, que abarca Barcelona, Valencia, Santander, Sevilla, Zaragoza y otras 35 ciudades en todo el mundo. Ha sido calificada como la “Mejor aplicación de alquiler de bicicletas” por CNET y The Londonist [CYC]. Otra *app* de estas características es *CityBikes* [CBK] con información de más de 170 ciudades y casi 100.000 descargas. Lo que más llama la atención de esta última (además del número de ciudades soportadas) es la forma de identificar de manera visual en el mapa (por colores: rojo, ámbar o verde) la disponibilidad de las estaciones.

No se pretende realizar con este TFC un proyecto de estas características (sobre todo en la calidad de la interfaz de usuario) ya que se trata de *apps* desarrolladas y mantenidas por empresas dedicadas al desarrollo de *apps* con equipos de hasta 20 profesionales a tiempo completo, pero dentro de las posibilidades se van a incluir funcionalidades que no se han visto en ninguna de las encontradas en *Google Play*, como el análisis de tendencias. Una vez concluida la implementación de la *app* se comparará en la Memoria el resultado obtenido con estas otras *apps*.

Capítulo 5.- Diseño

5.1.- Estrategia de diseño

Se ha optado por seguir una estrategia por fases, en la que se parte de la aplicación *Android* más sencilla posible (mostrar el mensaje “*hola, mundo*” por pantalla) e ir añadiendo en las sucesivas fases las funcionalidades deseadas. De esta manera se han planificado las fases como se expone a continuación.

- Fase 0.- Verificación del entorno del trabajo y construcción de una aplicación básica de prueba tipo “*hola, mundo*”.
- Fase 1.- Estructura de un menú principal y separación de las funcionalidades en *Activities* o pantallas.
- Fase 2.- Mostrar por pantalla un mapa con la ubicación del usuario. Resolución de dificultades con el emulador y *Google Maps*.
- Fase 3a.- Conexión de la aplicación a Internet. Conexión y lectura de catálogos de *open data*.
- Fase 3b.- Creación, escritura y lectura de una base de datos que mantenga la información obtenida de las estaciones *Bicing*.
- Fase 4.- *Activity* lista de estaciones.
- Fase 5.- *Activity* mapa – mostrar estaciones en el mapa.
- Fase 6.- *Activity* mapa – búsqueda de estación más cercana.
- Fase 7 – *Activity* mapa – búsqueda con criterios y trazado de ruta.
- Fase 8.- Resto de *activities*.
- Fase 9.- Refinamiento de las *activities*.
- Fase 10.- Pruebas, redacción de menú ayuda y traducción inglés.

5.2.- Estructura de un proyecto *Android*

5.2.1.- Información inicial para crear un proyecto

Cuando se crea un proyecto *Android* por primera vez necesitamos proporcionar la siguiente información (ver figura 8):

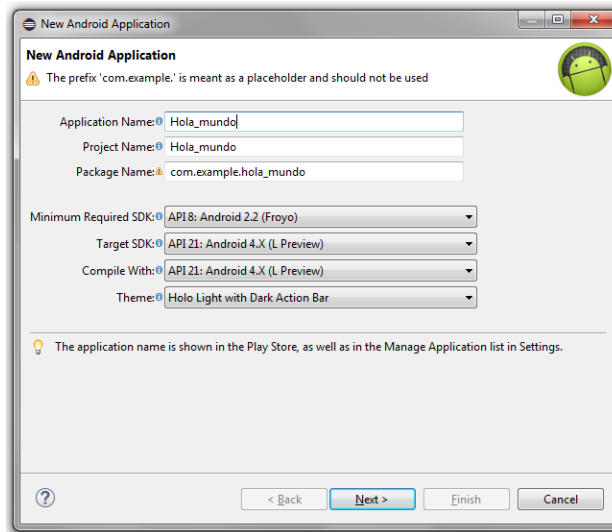


Figura 8.- Asistente de Eclipse para crear una nueva aplicación *Android*.

- Nombre de la aplicación (*Application Name*). Es el nombre de la aplicación, que aparecerá por defecto cuando se instale en un dispositivo.
- Nombre del proyecto (*Project Name*). Su ámbito es únicamente dentro del IDE Eclipse.
- Nombre del paquete (*Package Name*). En nomenclatura DNS.
- Versión mínima del SDK (*Minimum Required SDK*). Se refiere a la API mínima que requiere la aplicación para ser instalada en el dispositivo. Cuanto menor sea, mayor será el número de dispositivos en el que se podrá instalar, y viceversa.
- Versión objetivo del SDK (*Target SDK*). Se refiere a la API máxima en la que puede funcionar la aplicación. Normalmente se utiliza la última versión que haya disponible. Si se lanza una nueva versión de *Android*, habría que

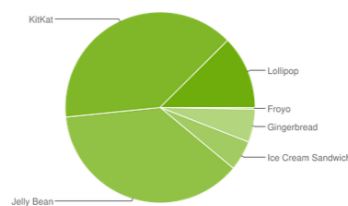
verificar que la aplicación funciona sin problemas en la nueva API y actualizar este valor.

- Versión para compilar (*Compile With*). Indica la API con la que se compilará la aplicación. Por defecto, Eclipse utiliza la última versión de que dispone. Es aconsejable usar la última versión, ya que aunque la aplicación no emplee las nuevas características de la versión más reciente, ésta corrige fallos de las anteriores.

5.2.2.- La versión del API

La cuestión de la versión del API es muy importante. Por una parte, una API mínima muy alta hará que nuestra aplicación no pueda ser utilizada por los usuarios con terminales más antiguos; aunque como se ha visto en el Estado del arte, el parque de dispositivos *Android* se ha actualizado significativamente en los últimos meses. Se puede ver en la tabla 5 la distribución actual a fecha 1 de junio 2015. Por otra parte, es indispensable estar siempre al corriente del lanzamiento de las últimas versiones. Así, mientras se realizaba este proyecto la última versión disponible era la API 21 (*Android Lollipop 5.0.1*), sin embargo cuando se escriben estas líneas la versión más actual es la API 22 (*Android Lollipop 5.1.1*) y además ya hay una versión de prueba de la próxima que provisionalmente se denomina *Android M*.

Version	Codename	API	Distribution
2.2	Froyo	8	0.3%
2.3.3 - 2.3.7	Gingerbread	10	5.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	5.1%
4.1.x	Jelly Bean	16	14.7%
4.2.x		17	17.5%
4.3		18	5.2%
4.4	KitKat	19	39.2%
5.0	Lollipop	21	11.6%
5.1		22	0.8%



Data collected during a 7-day period ending on June 1, 2015.
Any versions with less than 0.1% distribution are not shown.

Tabla 5.- Distribución de las versiones de *Android* en junio 2015. Fuente [[ADB](#)]

- Primera actividad y su *layout*. Por defecto el asistente crea una *activity* inicial con el nombre de `MainActivity` y una UI (*layout*) en blanco con el nombre de `activity_main`. Se trata respectivamente de un fichero fuente con una clase Java `MainActivity.java` y de un archivo XML `activity_main.xml` con la definición de los controles de la interfaz (botones, cuadros de texto, etc.).
- También se puede configurar si se desea añadir un icono para lanzar la aplicación (*launcher icon*). Alternativamente el asistente incorpora otras posibilidades distintas a actividad en blanco (*Blank Activity*) para crear la actividad principal.

Una *activity* o actividad en *Android* es una pantalla que contiene la interfaz de usuario (UI – *user interface*) de la aplicación. En general, una *app* puede necesitar una o varias pantallas, pero también puede darse el caso de no precisar ninguna, pensemos p.ej. en un servicio (también es una aplicación en *Android*) que se esté ejecutando en segundo plano y por lo tanto no muestre ninguna UI.

5.2.3.- Estructura de directorios

Examinemos las carpetas de un proyecto básico, una aplicación que muestra por pantalla el mensaje “*Hola, mundo*”. Como puede verse en la siguiente captura de pantalla (ver figura 9) del explorador de Eclipse (*Package Explorer*), todo proyecto en *Android* se estructura de forma jerárquica. Vamos a describir brevemente las carpetas que contiene.

En primer lugar se observan dos proyectos abiertos. El primero es una librería de compatibilidad `appcompat_v7` que es necesario añadir a nuestro proyecto para que soporte versiones anteriores (se genera automáticamente con el primer proyecto y posteriormente hay que asegurarse de que los proyectos la incorporan en la configuración de compilación). El segundo,

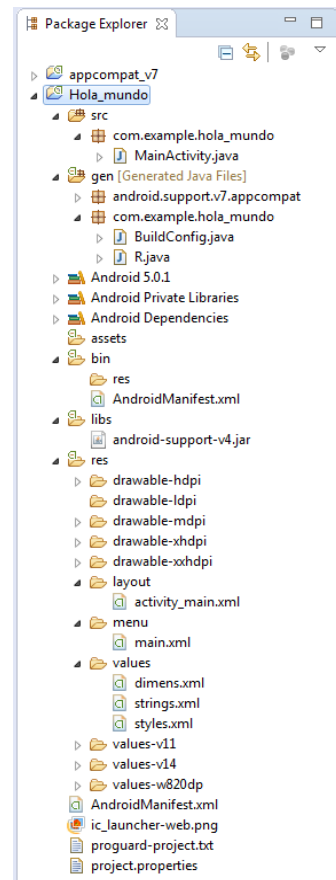


Figura 9.- Árbol de directorios de un proyecto Android.

Hola_mundo, es el proyecto propiamente dicho. Como muestra la figura 10 en la que se ven las propiedades del proyecto (*Properties for Hola_mundo*), la librería de compatibilidad está referenciada.

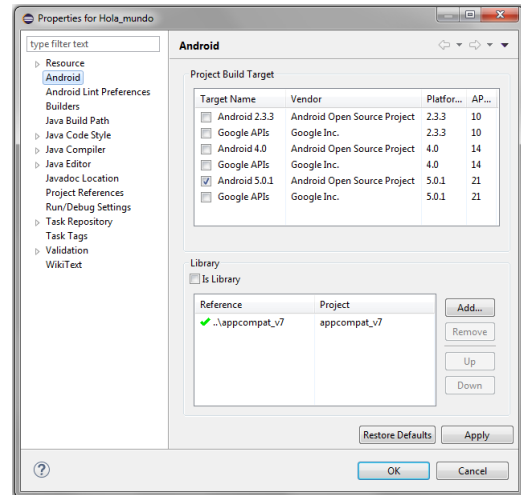


Figura 10.- Propiedades de un proyecto *Android*.

- En `/src` se encuentra el código fuente (*source*) de la aplicación (en este caso un único archivo Java `MainActivity.java` ya que este ejemplo sólo tiene una actividad).
- En `/gen` se van situando los códigos generados (*generated*) automáticamente. Fundamentalmente contiene el archivo `R.java` que hace referencia a los recursos del proyecto (*resources*). No debe modificarse.
- *Android 5.0.1* (u otra versión dependiendo de cómo se configure el proyecto – ver fig. 10 *Properties for*) contiene un fichero con las bibliotecas de clases necesarias para crear una aplicación *Android*.
- *Android Private Libraries* contiene bibliotecas necesarias para que la aplicación se pueda ejecutar, como las de compatibilidad p.ej.
- *Android Dependencies* son otras bibliotecas adicionales.
- La carpeta `/assets` contiene ficheros que se empaquetan con la aplicación cuando se instalan en un dispositivo. En este caso está vacía ya que se trata de un proyecto muy sencillo.
- En `/bin` (*binary*) se localiza la aplicación compilada (será el nombre de la aplicación con la extensión correspondiente `Hola_mundo.apk`). En el subdirectorio `/bin/classes` se encuentran las clases Java compiladas.
- La carpeta `/libs` (*libraries*) contiene bibliotecas externas que utiliza la aplicación, básicamente serán bibliotecas desarrolladas por terceros.

- En `/res` se almacenan los recursos (*resources*) que utiliza la aplicación. Contiene varias subcarpetas:
 - `/res/drawable` para los gráficos (se puede distinguir entre distintas resoluciones mediante sufijos como `-ldpi`, `-mdpi`, `-hdpi`, `-xhdpi`, etc).
 - `/res/layout` para la configuración y diseño de los controles de las *activities*. En este caso aparece `activity_main.xml` con la disposición de la actividad principal.
 - `/res/menu` para los menús de cada pantalla o *activity* (en este caso aparece un menú por defecto `main.xml` para la actividad principal).
 - `/res/values` se usa para almacenar recursos adicionales como cadenas de texto (`strings.xml`), dimensiones usadas como medida (`dimens.xml`) o archivos de estilo (`styles.xml`). Es muy interesante la posibilidad de poder usar también sufijos para separar estos recursos por versiones (como p.ej. `values-v11` o `values-v14`), por idiomas (`values-es` para castellano, `values-fr` para francés, etc.), o por anchura de pantalla (como el que aparece `values-w820dp` que sería para dispositivos con anchura, *w* de *wide*, superior a 820 puntos). De esta manera podemos tratar la disposición de la pantalla y otros recursos distinguiendo el idioma que está seleccionado, el tipo de terminal, la versión *Android* instalada, las dimensiones de la pantalla o su orientación. Es por este motivo que la misma aplicación no se ve de la misma forma en distintos terminales).
- El archivo `AndroidManifest.xml` es de gran importancia puesto que en él se especifican los permisos de la aplicación, se declaran las actividades de que se compone, la versión mínima SDK que soporta, etc.

Hemos visto *grosso modo* la disposición del proyecto. Inmediatamente nos damos cuenta de que por un lado tenemos la lógica de la aplicación (lo que tiene que

hacer) en ficheros de código fuente Java y por otro lado tenemos la disposición de la UI (cómo se ve en pantalla) que se especifica en archivos XML. Aunque la aplicación podría construirse casi íntegramente con código Java, sin emplear ficheros XML con los recursos, resulta evidente la ventaja de emplear esta separación.

Esta forma de trabajar condiciona la estrategia de diseño de la aplicación. Por una parte se trabaja en el aspecto visual y el diseño de la UI mediante los correspondientes archivos XML (no es necesario tampoco conocer muy a fondo este lenguaje puesto que los asistentes se encargan de generar casi todo el código). Por otra parte se trabaja en la programación propiamente dicha. Es indudable que es una gran ventaja y facilita el desarrollo de la aplicación, especialmente en proyectos de cierta envergadura en los que participan equipos de varios desarrolladores. Así puede haber programadores trabajando en el código de *activities* distintas, diseñadores encargándose del aspecto visual elaborando los archivos de recursos, etc. Luego simplemente hay que añadir la referencia a cada recurso cuando se precise. P.ej. cuando aparece un botón en la pantalla, su ubicación en pantalla, el texto que aparece en el propio botón y las dimensiones y características del mismo son un tema de diseño y pueden cambiarse o hacer diferentes versiones (por idioma, por orientación pantalla, etc.) sin influir en la lógica que hay detrás de hacer clic en ese botón.

5.2.4.- Ciclo de vida de una actividad en Android

Otro aspecto a tener en cuenta a la hora del diseño es el ciclo de vida de una actividad, es decir el conjunto de estados por el que va pasando la *activity* desde que se invoca hasta que se presenta por pantalla o finalmente se destruye. De acuerdo con la figura 11 los eventos principales con los que se va a trabajar son:

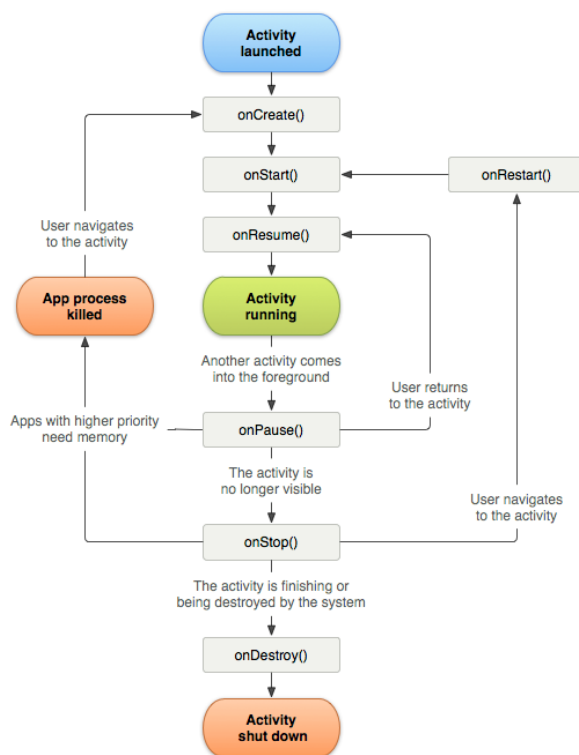


Figura 11.- Ciclo de vida de una actividad. Fuente [ACY]

- `onCreate()` este método se llama cuando se ejecuta la actividad por primera vez.
- `onStart()` se invoca cuando se muestra la actividad al usuario.
- `onResume()` se llama cuando el usuario ya puede interactuar con la aplicación.
- `onPause()` se invoca cuando se detiene la actividad actual para continuar con la anterior.
- `onStop()` se llama cuando la aplicación ya no es visible para el usuario.
- `onDestroy()` se invoca justo antes de que el SO destruya la actividad.
- `onRestart()` se llama cuando después de haber estado pausada la actividad vuelve a ejecutarse.

5.3.- Diseño del menú principal y disposición de las funciones de la app

Se pretenden disponer las distintas funcionalidades que se han especificado en un menú principal de tipo lista de manera que sea posible acceder haciendo clic en la opción que escoja el usuario y se pueda volver al mismo en cualquier momento haciendo clic en el botón *volver*.

Como puede verse en la captura de pantalla de la figura 12 se establece un título para cada opción del menú y una breve descripción. Esta será la pantalla principal, la que por defecto aparecerá cuando se ejecute la aplicación por primera vez. Siguen un orden de acuerdo con la importancia que se estima que tiene para el usuario. Así la opción de mapa se supone que es la que usará varias

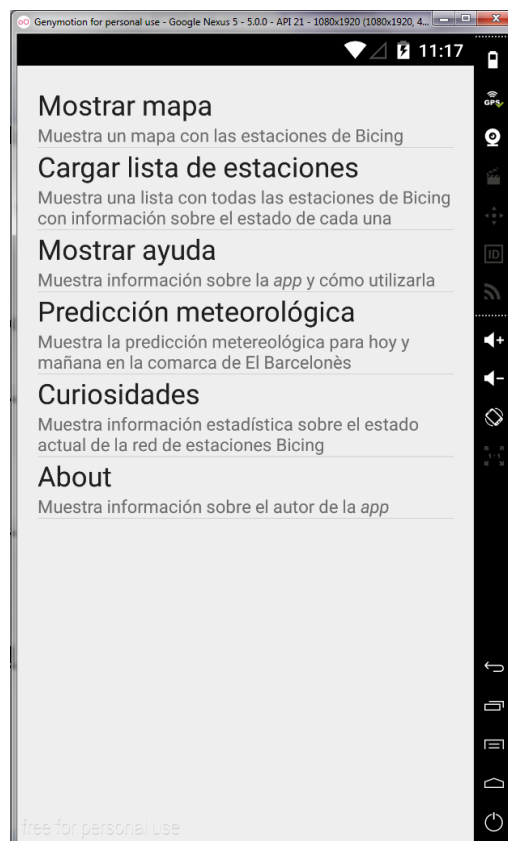


Figura 12.- Menú principal.

veces cada vez que inicie la aplicación, mientras que la de *about* probablemente la consulte como mucho una única vez.

La disposición que se adapta mejor a este diseño es una `ListView` y será la que veamos en la parte de implementación. Como puede verse en la figura 13 este tipo de control permite incluir un título y una descripción para cada opción del menú. Además, en el caso de que el tamaño de pantalla no permitiera mostrar todas las opciones, el usuario podría hacer *scroll* (desplazar la lista) hacia abajo para acceder a las que no se visualicen en ese momento, mostrando una barra de desplazamiento a la derecha.

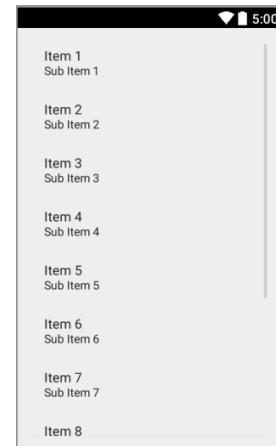


Figura 13.- Control `ListView` genérico.

5.3.1.- Icono de la aplicación

Para acceder a la aplicación se personalizará un sencillo icono que el usuario podrá arrastrar y soltar para colocar en cualquier lugar de su dispositivo como muestra la captura de pantalla de la fig. 14 en la que el icono de la *app* aparece en la pantalla principal del terminal (en este caso un emulador de un *Nexus 5* con *Android 5.0.0*).

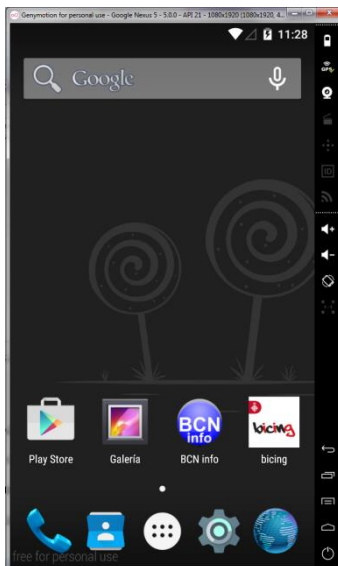


Figura 14.- Pantalla principal de un terminal con el icono de la *app* instalada. Como puede verse, a la derecha aparece el icono de la aplicación oficial de *Bicing*, que no aprovecha las mayores capacidades gráficas de las últimas versiones de *Android* para mostrar un icono algo más elaborado.



Figura 15.- Clusterización o agrupamiento de marcadores en el mapa.

5.4.- Diseño del mapa con Google Maps Android API v2

Para la actividad del mapa, la principal dentro de la aplicación, utilizamos esta herramienta que nos proporciona Google. La idea es mostrar en la mayor parte de la pantalla un mapa con la ubicación del usuario y marcadores para indicar la localización de las estaciones de *Bicing*. Además usaremos colores para distinguir el estado de las estaciones. En la parte de implementación se explicará cómo se ha buscado una solución (*clusterización* o agrupación de marcadores – ver fig. 15) a la sobrecarga visual que supone el mostrar demasiados marcadores en un mapa.

Además un conjunto de botones (control `Button`) permitirán la interacción del usuario con la aplicación, de manera que pueda buscar la estación más cercana a su ubicación según distintos criterios. También se utilizarán `Toasts` (mensajes emergentes) para mostrar información. La aplicación dibujará sobre el mapa la ruta hasta la estación encontrada (ver fig. 16). Para ello se adaptará una aplicación de ejemplo que se encuentra en la página oficial de *Google Maps Android*, que básicamente convierte en una petición de ruta los datos de ubicación del usuario y de la estación de destino y posteriormente trata esta información para trazar la ruta en el mapa. Finalmente una serie de cuadros de texto (control `TextView`) mostrarán información al usuario sobre la estación encontrada.

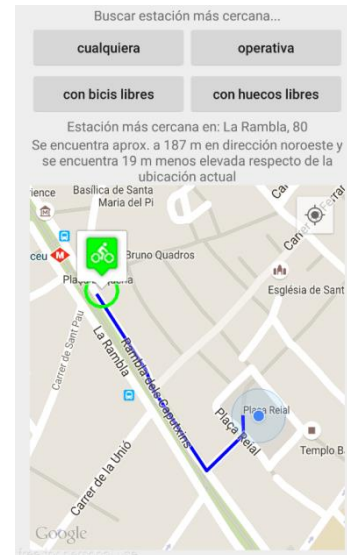


Figura 16.- Trazado de la ruta hasta la estación de destino sobre el mapa.

5.5.- Limitaciones del emulador incorporado. Utilización de la herramienta Genymotion

El emulador que acompaña el SDK de *Android* tiene algunas limitaciones a la hora de crear dispositivos virtuales y es especialmente lento a la hora de hacer pruebas. Además, ejecutar aplicaciones en un dispositivo virtual con las últimas versiones de *Android* es inasumible por la ralentización que se experimenta. En cualquier caso el motivo principal de utilizar un emulador distinto es la necesidad de mostrar mapas en el

emulador lo cual no es soportado actualmente (al menos en el momento de desarrollar la *app*).

Las ventajas de utilizar la versión gratuita del emulador de *Genymotion* (hay otra versión de pago con más características pero para esta *app* no tiene relevancia) sobre el emulador que proporciona *Google* son fundamentalmente dos: rapidez de ejecución (incluso más rápido que en un terminal físico) y posibilidad de mostrar mapas.

No obstante, también es cierto que para poder hacer esto último hay que preparar previamente el emulador. Cargando una imagen del SO que se pueden encontrar en comunidades de desarrolladores para poder instalar *Google Services*, de esta forma nos podemos identificar mediante una cuenta de *Google* e instalar aplicaciones desde *Google Play* y ya poder mostrar los mapas.

Se ha utilizado la versión 2.3.1 de este software, no obstante cuando se escriben estas líneas ya se ha lanzado la versión 2.4. También es necesario tener instalado *VirtualBox* en el equipo junto con una serie de complementos de red, todo ello, aunque costoso en tiempo es muy sencillo de realizar siguiendo las instrucciones de instalación.

5.6.- Inspección de los recursos *open data*. Diseño de la base de datos

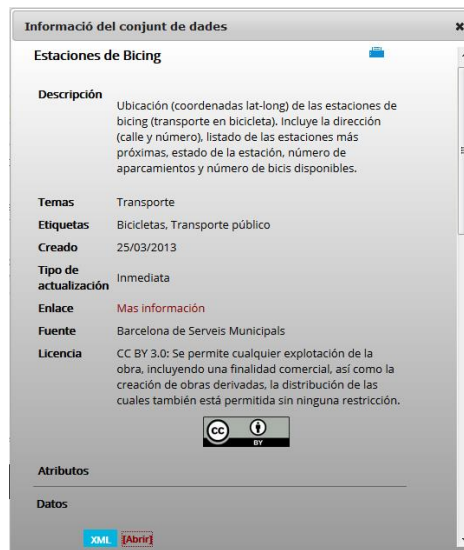


Figura 17.- Descripción del recurso de las estaciones de *Bicing*. Fuente [\[CCBY\]](#)

5.6.1. Inspección del recurso de las estaciones de *Bicing*

Como ya se ha expuesto en el Estado del arte, la información de las estaciones de *Bicing* se encuentra disponible en abierto el catálogo de datos en abierto del *Ajuntament de Barcelona* [\[CODAB\]](#). Se distribuye bajo licencia CC BY 3.0, es decir, una licencia *Creative Commons* (CC) que permite cualquier tipo de explotación, incluso comercial, y la creación de obras derivadas sin ninguna restricción. La única condición es el reconocimiento (*attribution*) de la autoría (BY) [\[CCBY\]](#).

En la figura 17 puede verse una breve descripción que se facilita del recurso (junto con el enlace al recurso en sí, al final). Se trata de un archivo en formato XML que se actualiza periódicamente. No se dan más detalles sobre el recurso por lo que no queda más remedio que analizar el contenido y deducir la manera en que se disponen los datos.

Esta inspección previa del fichero servirá para derivar la estructura de la base de datos que emplearemos para leer, almacenar y tratar la información relativa a las estaciones de *Bicing* y hacer que la *app* se nutra de ella. En la figura 18 puede verse una captura de pantalla que muestra parte del fichero XML.

```
-<bicing_stations>
  <updateTime>143422251</updateTime>
  -<station>
    <id>1</id>
    <type>BIKE</type>
    <lat>41.397952</lat>
    <long>2.180042</long>
    <street>Gran Via Corts Catalanes</street>
    <height>21</height>
    <streetNumber>760</streetNumber>
    <nearbyStationList>24, 369, 387, 426</nearbyStationList>
    <status>OPN</status>
    <slots>16</slots>
    <bikes>8</bikes>
  </station>
  -<station>
    <id>2</id>
    <type>BIKE</type>
    <lat>41.39553</lat>
    <long>2.17706</long>
    <street>Roger de Flor/ Gran V&iacut;a</street>
    <height>21</height>
    <streetNumber>126</streetNumber>
    <nearbyStationList>360, 368, 387, 414</nearbyStationList>
    <status>OPN</status>
    <slots>25</slots>
    <bikes>1</bikes>
  </station>
  -<station>
    <id>3</id>
    <type>BIKE</type>
    <lat>41.393699</lat>
    <long>2.181137</long>
    <street>Ali Bei</street>
    <height>21</height>
    <streetNumber>44</streetNumber>
    <nearbyStationList>4, 6, 119, 419</nearbyStationList>
    <status>OPN</status>
    <slots>11</slots>
    <bikes>15</bikes>
  </station>
  -<station>
    <id>4</id>
```

Figura 18.- Captura de pantalla del fichero XML. Fuente [[XMLB](#)]

Como se observa, en archivo contiene una única etiqueta `<bicing_stations>` dentro de la que se incluye una única etiqueta `<updateTime>` que contiene la hora de actualización del archivo y a continuación una serie de etiquetas `<station>` que contienen los campos con los datos de las estaciones. Veámoslo más en detalle.

- `<id>` contiene el identificador de la estación. Este valor nunca está vacío. Se trata de un identificador único para cada estación, es decir que no se repite para otras estaciones ni cambia cada vez que se actualiza. Por lo tanto nos servirá como clave primaria (*primary key*) de nuestra base de datos.
- `<type>` contiene una cadena de caracteres (*string*). Este valor nunca está vacío. Un vistazo exhaustivo a todo el archivo no nos saca de dudas sobre qué utilidad puede tener este campo, puesto que el valor es el mismo para todas las estaciones (BIKE). Suponemos que se trata de estaciones con bicis

convencionales, puesto que se ha previsto incorporar bicicletas eléctricas al sistema de bicicletas compartidas pero no se ha observado que aparezcan otros valores y esto que, como se ha expuesto en el Estado del arte, supuestamente ya están operativas este otro tipo de bicicletas. Todo parece indicar que, o bien no se está reflejando esta distinción en este campo (el administrador no lo está gestionando), o bien todavía no están disponibles. En este sentido, se ha dirigido una consulta al servicio de atención aunque de momento no se ha obtenido respuesta. En caso de que se reciba contestación, se expondrá su resolución en la memoria.

- `<lat>` contiene la latitud geográfica en grados de la ubicación de la estación. Este valor nunca está vacío. Lo trataremos como un número real, ya que tiene decimales. Valores positivos (entre 0 y 90) hacen referencia al hemisferio norte, mientras que valores negativos (entre 0 y -90) hacen referencia al hemisferio sur. En el caso de Barcelona siempre tendremos valores positivos por lo que no debemos preocuparnos por el signo.
- `<long>` ídem que la anterior pero en este caso se trata de la longitud geográfica. Nunca está vacío. Valores positivos (entre 0 y 180) hacen referencia al hemisferio oriental, mientras que valores negativos (entre 0 y -180) hacen referencia al hemisferio occidental. En el caso de Barcelona siempre tendremos valores positivos por lo que no debemos preocuparnos por el signo.
- `<street>` contiene una cadena de caracteres (*string*) con el nombre de la calle (u otro tipo de vía, como plaza, avenida, etc...). Este valor nunca está vacío. Además, se observa que ciertos caracteres como las vocales que llevan acentos, la ce de cedilla y otros no aparecen correctamente por lo que habrá que tratarlos antes de mostrarlos al usuario como se verá en la parte de implementación.
- `<streetNumber>` contiene un entero (*integer*) con el número de la vía en el que se encuentra la estación. Este valor puede estar vacío por lo que es importante tratarlo correctamente para no provocar errores.

- `<height>` contiene un entero (*integer*) que deducimos que se trata de la altura en metros sobre el nivel del mar. Este valor no está nunca vacío y puede valer cero.
- `<nearbyStationList>` contiene una lista de enteros (*integer*) separados por comas, haciendo referencia a los identificadores de las estaciones más próximas. A priori parece un mecanismo que ayudará a la hora de implementar la funcionalidad de búsqueda de la estación más cercana pero como se verá se ha realizado de otra manera distinta. Este valor nunca está vacío.
- `<status>` contiene una cadena de caracteres (*string*) con el estado en que se encuentra la estación, distinguiendo entre `OPN` (*opened* u operativa) y `CLS` (*closed* o fuera de servicio). Este valor nunca está vacío.
- `<slots>` contiene un número entero (*integer*) positivo con el número de huecos libres en la estación. Este valor nunca está vacío, en el caso de que la estación esté fuera de servicio aparece un 0.
- `<bikes>` contiene un número entero (*integer*) positivo con el número de bicis disponibles en la estación. Este valor nunca está vacío, en el caso de que la estación esté fuera de servicio aparece un 0.

5.6.2. Inspección del recurso de previsión meteorológica

En cuanto a la información de previsión meteorológica, como ya se había adelantado en el Estado del arte, ésta se encuentra disponible en el catálogo de datos en abierto de la *Generalitat de Catalunya* [CAT]. Se trata de un archivo XML mantenido por el Servicio Meteorológico de Cataluña y que se actualiza diariamente a las 15:00 horas y muestra la predicción para el día actual y el siguiente. En la fig. 19 se puede ver la descripción del recurso que se encuentra.

Predicción comarcal

Dataset en formato RDF | Imprimir | Comparte

Acceso a los datos

Ficheros

Formato	Idioma	Especificaciones técnicas
XML	Català	Predicción del tiempo comarca a comarca. Atención, ha cambiado la url del fichero. En vez de http://www.meteo.cat/servmet/opendata/ctermini_comarcal.xml , la dirección correcta es: http://static-m.meteo.cat/content/opendata/ctermini_comarcal.xml (09/12/2014).

Descripción

El Servicio Meteorológico de Cataluña actualiza cada día a las 15 horas la predicción comarca a comarca. La información contiene el símbolo del tiempo para la mañana y para la tarde, así como la temperatura máxima y mínima para los dos días siguientes.

Categorías: Medio ambiente Meteorología

Ámbito geográfico: Catalunya

Periodicidad

Fecha de creación: 09/12/2014
Frecuencia de actualización: Diaria

Fuente de los datos

Generalitat de Catalunya. Departament de Territori i Sostenibilitat. Servei Meteorològic de Catalunya.

Licencia

Autorización de acuerdo ley 37/2007:
Sin licencia asociada: sin ningún tipo de restricción más allá de las condiciones básicas establecidas en el artículo 8 de la Ley 37/2007 sobre reutilización de la información del sector público (citación de la fuente, no alteración ni desnaturalización de la información y especificación de la fecha de última actualización).

Aviso legal | Accesibilidad | Sobre gencat | © Generalitat de Catalunya

Figura 19.- Captura de pantalla con la descripción del recurso. Fuente [CAT]

De igual modo que se ha explicado en el apartado anterior, una inspección previa a la estructura del recurso proporciona la estrategia a seguir para el diseño de la actividad que se nutrirá de sus datos.

En la fig. 20 aparece una captura de pantalla donde puede apreciarse las primeras líneas del archivo XML, donde después de la hora de actualización se relacionan los códigos de las distintas comarcas. Para nuestra aplicación sólo nos interesarán los datos de la comarca con el código 13, que corresponde a El Barcelonès. En el siguiente *screenshot* (fig. 21) se puede distinguir los código de los símbolos utilizados para la descripción (en catalán) del tiempo meteorológico.

```

-<smc datacreacio="17-06-2015 15:00:00">
<comarca id="1" nomCOMARCA="L'Alt Camp" nomCAPITALCO="Valls"/>
<comarca id="2" nomCOMARCA="L'Alt Empordà" nomCAPITALCO="Figueres"/>
<comarca id="3" nomCOMARCA="L'Alt Penedès" nomCAPITALCO="Vilafranca del Penedès"/>
<comarca id="4" nomCOMARCA="L'Alt Urgell" nomCAPITALCO="La Seu d'Urgell"/>
<comarca id="5" nomCOMARCA="L'Alta Ribagorça" nomCAPITALCO="El Pont de Suert"/>
<comarca id="6" nomCOMARCA="L'Anoia" nomCAPITALCO="Igualada"/>
<comarca id="7" nomCOMARCA="El Bages" nomCAPITALCO="Manresa"/>
<comarca id="8" nomCOMARCA="El Baix Camp" nomCAPITALCO="Reus"/>
<comarca id="9" nomCOMARCA="El Baix Ebre" nomCAPITALCO="Tortosa"/>
<comarca id="10" nomCOMARCA="El Baix Empordà" nomCAPITALCO="La Bisbal d'Empordà"/>
<comarca id="11" nomCOMARCA="El Baix Llobregat" nomCAPITALCO="Sant Feliu de Llobregat"/>
<comarca id="12" nomCOMARCA="El Baix Penedès" nomCAPITALCO="El Vendrell"/>
<comarca id="13" nomCOMARCA="El Barcelonès" nomCAPITALCO="Barcelona"/>
<comarca id="14" nomCOMARCA="El Berguedà" nomCAPITALCO="Berga"/>
<comarca id="15" nomCOMARCA="La Cerdanya" nomCAPITALCO="Puigcerdà"/>
<comarca id="16" nomCOMARCA="La Conca de Barberà" nomCAPITALCO="Montblanc"/>
<comarca id="17" nomCOMARCA="El Garraf" nomCAPITALCO="Vilanova i la Geltrú"/>
<comarca id="18" nomCOMARCA="Les Garrigues" nomCAPITALCO="Les Borges Blanques"/>
<comarca id="19" nomCOMARCA="La Garrotxa" nomCAPITALCO="Olot"/>
<comarca id="20" nomCOMARCA="El Gironès" nomCAPITALCO="Girona"/>
<comarca id="21" nomCOMARCA="El Maresme" nomCAPITALCO="Mataró"/>
<comarca id="22" nomCOMARCA="El Montsià" nomCAPITALCO="Amposta"/>
<comarca id="23" nomCOMARCA="La Noguera" nomCAPITALCO="Balaguer"/>
<comarca id="24" nomCOMARCA="Osona" nomCAPITALCO="Vic"/>
<comarca id="25" nomCOMARCA="El Pallars Jussà" nomCAPITALCO="Tremp"/>
<comarca id="26" nomCOMARCA="El Pallars Sobirà" nomCAPITALCO="Sort"/>
<comarca id="27" nomCOMARCA="El Pla de l'Estany" nomCAPITALCO="Banyoles"/>
<comarca id="28" nomCOMARCA="El Pla d'Urgell" nomCAPITALCO="Mollerussa"/>
<comarca id="29" nomCOMARCA="El Priorat" nomCAPITALCO="Falset"/>
<comarca id="30" nomCOMARCA="La Ribera d'Ebre" nomCAPITALCO="Móra d'Ebre"/>
<comarca id="31" nomCOMARCA="El Ripollès" nomCAPITALCO="Ripoll"/>
<comarca id="32" nomCOMARCA="La Segarra" nomCAPITALCO="Cervera"/>
<comarca id="33" nomCOMARCA="El Segrià" nomCAPITALCO="Lleida"/>
<comarca id="34" nomCOMARCA="La Selva" nomCAPITALCO="Santa Coloma de Farners"/>
<comarca id="35" nomCOMARCA="El Solsonès" nomCAPITALCO="Solsona"/>
<comarca id="36" nomCOMARCA="El Tarragonès" nomCAPITALCO="Tarragona"/>
<comarca id="37" nomCOMARCA="La Terra Alta" nomCAPITALCO="Gandesa"/>
<comarca id="38" nomCOMARCA="L'Urgell" nomCAPITALCO="Tàrraga"/>
<comarca id="39" nomCOMARCA="La Val d'Aran" nomCAPITALCO="Vielha e Mijaran"/>
<comarca id="40" nomCOMARCA="El Vallès Occidental" nomCAPITALCO="Sabadell/Terrassa"/>
<comarca id="41" nomCOMARCA="El Vallès Oriental" nomCAPITALCO="Granollers"/>

```

Figura 20.- Captura de pantalla del contenido del archivo XML, donde se ven los códigos de comarca. Fuente [[XMLM](#)]

```

<simbol id="1" nomsimbol="sol"/>
<simbol id="2" nomsimbol="sol i núvols alts"/>
<simbol id="3" nomsimbol="entre poc i mig ennuvolat"/>
<simbol id="4" nomsimbol="molt cobert"/>
<simbol id="5" nomsimbol="plugim"/>
<simbol id="6" nomsimbol="pluja"/>
<simbol id="7" nomsimbol="xàfec"/>
<simbol id="8" nomsimbol="tempesta"/>
<simbol id="9" nomsimbol="tempesta calamarsa"/>
<simbol id="10" nomsimbol="neu"/>
<simbol id="11" nomsimbol="boira"/>
<simbol id="12" nomsimbol="boirina"/>
<simbol id="13" nomsimbol="xàfec neu"/>
<simbol id="20" nomsimbol="entre mig i molt ennuvolat"/>
<simbol id="21" nomsimbol="cobert"/>
<simbol id="22" nomsimbol="calitja"/>
<simbol id="23" nomsimbol="ruixat"/>
<simbol id="24" nomsimbol="xàfec amb tempesta"/>
<simbol id="25" nomsimbol="xàfec"/>
<simbol id="26" nomsimbol="ruixat"/>
<simbol id="27" nomsimbol="neu feble"/>
<simbol id="28" nomsimbol="tempesta neu"/>
<simbol id="29" nomsimbol="xàfec"/>
<simbol id="30" nomsimbol="aiguaneu"/>
<simbol id="31" nomsimbol="ruixat"/>
<simbol id="32" nomsimbol="plugim"/>
<tempesta id="1" nomprobtempmati="Baix" nomprobtemptarda="Baix"/>
<tempesta id="2" nomprobtempmati="Moderat" nomprobtemptarda="Moderat"/>
<tempesta id="3" nomprobtempmati="Alt" nomprobtemptarda="Alt"/>
<tempesta id="4" nomprobtempmati="Molt alt" nomprobtemptarda="Molt alt"/>
<calamarsa id="1" nomprobcalamati="Baix" nomprobcalatarda="Baix"/>
<calamarsa id="2" nomprobcalamati="Moderat" nomprobcalatarda="Moderat"/>
<calamarsa id="3" nomprobcalamati="Alt" nomprobcalatarda="Alt"/>
<calamarsa id="4" nomprobcalamati="Molt alt" nomprobcalatarda="Molt alt"/>

```

Figura 21.- Captura de pantalla del contenido del archivo XML, donde se ven los códigos de descripción meteorológica.

Por último, en la fig. 22, se observa el formato para la descripción de cada comarca, nuestra aplicación debe leer los datos de la comarca nº 13. Si analizamos más en detalle este formato distinguimos las siguientes partes:

```

- <prediccion idcomarca="13">
  <variable data="18-06-2015" dia="1" simbolmati="1.png" simboltarda="1.png" tempmax="28" tempmin="19" probcalamati="1" probcalatarda="1"/>
  <variable data="19-06-2015" dia="2" simbolmati="1.png" simboltarda="3.png" tempmax="27" tempmin="20" probcalamati="1" probcalatarda="1"/>
</prediccion>
- <prediccion idcomarca="14">
  <variable data="18-06-2015" dia="1" simbolmati="12.png" simboltarda="3.png" tempmax="25" tempmin="12" probcalamati="1" probcalatarda="1"/>
  <variable data="19-06-2015" dia="2" simbolmati="1.png" simboltarda="20.png" tempmax="26" tempmin="13" probcalamati="1" probcalatarda="1"/>
</prediccion>
- <prediccion idcomarca="15">
  <variable data="18-06-2015" dia="1" simbolmati="1.png" simboltarda="3.png" tempmax="22" tempmin="9" probcalamati="1" probcalatarda="1"/>
  <variable data="19-06-2015" dia="2" simbolmati="1.png" simboltarda="3.png" tempmax="23" tempmin="10" probcalamati="1" probcalatarda="1"/>
</prediccion>

```

Figura 22.- Captura de pantalla del contenido del archivo XML, con los datos que describen la predicción de cada comarca. Fuente [[XMLM](#)]

- data corresponde a la fecha de predicción en formato "dd-mm-aaaa"
- dia – un 1 corresponde al día actual y un 2 al día siguiente
- simbolmati – el código descriptivo de la predicción meteorológica para la jornada matinal
- simboltarda – el código descriptivo de la predicción meteorológica para la jornada vespertina
- tempmax – la temperatura máxima predicha en grados centígrados
- tempmin – la temperatura mínima predicha en grados centígrados
- probcalamati – el código descriptivo para la probabilidad de lluvia para la jornada matinal
- probcalatarda – el código descriptivo para la probabilidad de lluvia para la jornada vespertina

Capítulo 6.- Implementación

Una vez inspeccionados los archivos nos encontramos en disposición de realizar la implementación de la aplicación.

6.1.- Clase y actividad principal. Menú inicial.

Toda actividad en *Android* se implementa como una clase Java que hereda de la clase `Activity` (o de alguna de sus subclases). Para el menú principal se emplea una clase que llamamos `MainActivity` y que hereda de `ListActivity`:

```
public class MainActivity extends ListActivity {  
...  
}
```

Por otra parte en `activity_main.xml`, el fichero XML de recursos asociado, es donde se configuran los controles de la UI. Para conseguir que este fichero se cargue cuando se inicia la actividad y se muestre por pantalla al usuario, es necesario sobrescribir el método `onCreate` que se ejecuta cuando se crea la actividad:

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
...  
}
```

Para poder mostrar los elementos en la lista del menú principal de manera personalizada nos ayudamos de un adaptador de lista, de manera que será en una segunda clase `ActivityDetailsList` donde implementaremos los detalles de cada elemento de esta lista.

```
ListAdapter adapter = new ActivityListAdapter(this,  
ActivityDetailsList.ACTIVITIES);  
setListAdapter(adapter);
```

Cada actividad de la aplicación debe incluirse en el archivo AndroidManifest.xml entre las etiquetas <activity> </activity>, es la manera de decirle a Android que la clase que implementamos corresponde a una actividad. En el caso de la actividad principal le añadimos la acción MAIN para que sea la que se ejecute al inicio.

```
<activity
    android:name=".MainActivity"
    android:label="@string/app_name"
    android:icon="@drawable/ic_launcher_bcn_info_3" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity
    android:name=".MapActivity"
    android:label="@string/title_activity_map" >
</activity>
<activity
    android:name=".StationsListActivity"
    android:label="@string/title_activity_stations_list" >
</activity>
<activity
    android:name=".MainActivitySettings"
    android:label="@string/title_activity_main_activity_settings" >
</activity>
<activity
    android:name=".MapActivitySettings"
    android:label="@string/title_activity_map_activity_settings" >
</activity>
<activity
    android:name=".HelpActivity"
    android:label="@string/title_activity_help" >
</activity>
<activity
    android:name=".WeatherForecastActivity"
    android:label="@string/title_activity_weather_forecast" >
</activity>
<activity
    android:name=".CuriositiesActivity"
    android:label="@string/title_activity_curiosities" >
</activity>
<activity
    android:name=".WelcomeActivity"
    android:label="@string/title_activity_welcome" >
</activity>
<activity
    android:name=".AboutActivity"
    android:label="@string/title_activity_about" >
</activity>
```

Para cada elemento de la lista definimos una sencilla clase `ActivityDetails` con los tres atributos principales de cada uno: su título, su descripción y la actividad con la que se corresponden:

```
public class ActivityDetails {
    public final int titleId;

    public final int descriptionId;

    public final Class<? extends FragmentActivity> activityClass;

    public ActivityDetails(
        int titleId, int descriptionId, Class<? extends
FragmentActivity> activityClass) {
        this.titleId = titleId;
        this.descriptionId = descriptionId;
        this.activityClass = activityClass;
    }
}
```

Finalmente en la clase `ActivityDetailsList` relacionamos la actividad correspondiente a cada elemento de la lista:

```
public final class ActivityDetailsList {

    private ActivityDetailsList() {}

    public static final ActivityDetails[] ACTIVITIES = {
        new ActivityDetails(R.string.map_activity_title,
            R.string.map_activity_description,
            MapActivity.class),
        new ActivityDetails(R.string.stations_list_activity_title,
            R.string.stations_list_activity_description,
            StationsListActivity.class),
        new ActivityDetails(R.string.help_activity_title,
            R.string.help_activity_description,
            HelpActivity.class),
        new ActivityDetails(R.string.weather_activity_title,
            R.string.weather_activity_description,
            WeatherForecastActivity.class),
        new ActivityDetails(R.string.curiosities_activity_title,
            R.string.curiosities_activity_description,
            CuriositiesActivity.class),
        new ActivityDetails(R.string.about_activity_title,
            R.string.about_activity_description,
            AboutActivity.class),
    };
}
```


Este es el resultado que se obtiene por pantalla utilizando el emulador (fig. 23):

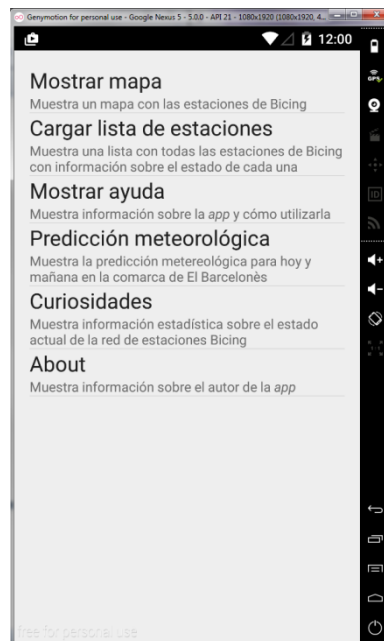


Figura 23.- Captura de pantalla que muestra el menú principal de la aplicación.

Obviamente si se hace clic en cualquiera de las opciones del menú no se tiene ningún resultado pero ahora se puede implementar cada actividad de manera independiente.

En esta actividad se han añadido posteriormente dos funciones. Por una parte el control de la conexión de red y por otra la actualización de la BD de las estaciones de Bicing cada vez que se pase por la pantalla de menú principal. De esta manera se obliga a que el contenido de la BD se refresque (si está permitido de acuerdo con la configuración de red escogida por el usuario). La función de control de la conexión de red se explicará más adelante y la de la BD se detallará en la actividad de la lista de estaciones.

Además se ha añadido un mensaje de bienvenida (que se ha implementado como una actividad aparte `WelcomeActivity`) para que se muestre por defecto la primera vez que se ejecute la aplicación (el usuario puede desactivar este mensaje desde el menú de preferencias). Lo comentaremos más adelante.

La referencia oficial de Android proporciona abundante documentación para trabajar con actividades [ACT], listas basadas en `ListView` [LSTV] y ejemplos de implementación de menús [SMPL].

6.2.- Clase y actividad lista de estaciones.

El siguiente paso consiste en crear la actividad que muestra un listado de las estaciones con información sobre su estado. En esta actividad también se implementa la lectura del archivo XML de datos en abierto y la creación y mantenimiento de la BD (aunque también se añade esta función para que se refresque la BD cada vez que se pasa por el menú principal como se ha adelantado anteriormente).

Esta actividad `StationsListActivity` también se estructura como una lista para ser mostrada por pantalla pero para poder personalizar todavía más el contenido de cada elemento necesitamos que herede de `ActionBarActivity`:

```
public class StationsListActivity extends ActionBarActivity {  
    ...  
}
```

Para gestionar la base de datos utilizamos un objeto de tipo `SQLiteDatabase` donde se crean las tablas de la BD y se almacenan los datos. También nos servimos de una clase `BCNInfoDBOpenHelper` que hereda de `SQLiteOpenHelper` para poder obtener los métodos que vamos a necesitar para operar con BD.

```
public class BCNInfoDBOpenHelper extends SQLiteOpenHelper {  
  
    private static final int DATABASE_VERSION = 1;  
    private static final String DATABASE_NAME = "BCNInfoDB";  
    private static final String STATIONS_TABLE_NAME = "stations";  
    private long updateTime = 0;  
    private static final String STATIONS_TABLE_DEFINITION =  
        "id INTEGER PRIMARY KEY, "  
        + "type TEXT, "  
        + "latitude REAL, "  
        + "longitude REAL, "  
        + "street TEXT, "  
        + "height INTEGER, "  
        + "streetNumber TEXT, "  
        + "nearbyStationList TEXT, "  
        + "status TEXT, "  
        + "slots INTEGER, "  
        + "bikes INTEGER";  
}
```

```

private static final String STATIONS_TABLE_CREATE = "CREATE
TABLE " + STATIONS_TABLE_NAME + " ( " + STATIONS_TABLE_DEFINITION +
)";

public BCNInfoDBOpenHelper(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
    // TODO Auto-generated constructor stub
}

@Override
public void onCreate(SQLiteDatabase db) {
    // TODO Auto-generated method stub
    db.execSQL(STATIONS_TABLE_CREATE);
}
}

```

Así, de acuerdo con el código anterior, vemos que cuando se crea esta actividad se ejecuta una instrucción SQL con el siguiente formato:

```

CREATE TABLE BCNInfoDB (id INTEGER PRIMARY KEY, type TEXT,
latitude REAL, longitude REAL, street TEXT, height INTEGER,
streetNumber TEXT, nearbyStationsList TEXT, status TEXT, slots
INTEGER, bikes INTEGER);

```

De acuerdo con el análisis previo que hemos realizado en la parte de diseño. Es decir usando como clave primaria el identificador de la estación.

Dentro de la actividad de la lista de estaciones hemos separado en tres tareas asíncronas, que en *Android* se denominan `AsyncTask`. En primer lugar se realiza la conexión para descargar el archivo XML, es la tarea `DownloadXMLTask`. En segundo lugar se escriben los datos descargados en la base de datos, es la tarea `WriteOnDataBaseTask`. Por último, se muestran los datos en la lista por pantalla, es la tarea `ListViewStationsTask`.

El motivo de realizar estas tareas de modo asíncrono es porque puede llevar un tiempo de ejecución que haría inviable que se realizaran síncronamente. P.ej. la conexión por Internet puede llevar unos segundos hasta que se realiza, si no lo hacemos de manera asíncrona el usuario vería una pantalla congelada y con la que no podría interactuar.

6.2.1.- Tarea `DownloadXMLTask`

Esta tarea intenta ejecutar el método `loadXMLFromNetwork` y devuelve el resultado como una lista para que pueda ser utilizado por la siguiente tarea. Este método `loadXMLFromNetwork` utiliza primero un método `downloadURL` para cargar la dirección donde se encuentra el recurso y a continuación se ayuda de una clase parser para realizar la lectura del archivo.

Es la clase `BicingStationsXMLParser`. En esta clase pueden verse los métodos empleados para ir saltando las etiquetas del archivo XML e ir guardando los datos de cada estación. También se han implementado unos métodos para corregir los acentos y caracteres especiales que no aparecerían correctamente en los nombres de las calles si no los corregimos previamente antes de mostrarlos al usuario.

La técnica seguida no es más que una adaptación de las prácticas recomendadas en la página de soporte de *Android Developers*, por un lado en las guías de conexión de red *Connecting to the Network* [[NWKCNT](#)] y *Managing Network Usage* [[MNGNWK](#)], y, por otro lado, en la guía de parseo de datos XML *Parsing XML Data* [[XMLPRS](#)].

Además, en la guía *Managing Network Usage* [[MNGNWK](#)] se explica cómo gestionar la conexión de red del dispositivo. De este modo, se ha implementado la aplicación para que el usuario configure si desea que se descarguen datos siempre que haya conexión de red, o sólo si se dispone de conexión WIFI para no consumir datos (dependiendo del operador puede suponer gastos).

El parser `XmlPullParser` utilizado es el recomendado en la guía *Parsing XML Data* [[XMLPRS](#)].

6.2.2.- Tarea `WriteOnDataBaseTask`

Esta tarea es la que se encarga propiamente de escribir en la BD los datos obtenidos en la tarea anterior. Básicamente ejecuta una instrucción SQL de tipo `INSERT OR REPLACE INTO` para cada fila de la tabla (para cada estación). Se toma la precaución de cerrar la BD antes de terminar la tarea para evitar errores de ejecución (igual que cuando se abre un archivo).

6.2.3.- Tarea ListViewStationsTask

Se encarga de mostrar mediante un adaptador los datos de la BD (con lo que se hace un acceso de lectura a la base de datos) y se tratan algunos datos (p.ej. el de la distancia de cada estación con respecto a la ubicación actual del usuario) para luego poder mostrarlos de una manera visualmente atractiva al usuario (ya en el adaptador `AdaptadorFilas` dentro del hilo de ejecución principal).

6.2.4.- Adaptador para mostrar los datos

Con el objeto de mostrar los datos de una manera visualmente atractiva empleamos un adaptador para personalizar cada subelemento de la lista de estaciones. Esto nos permite cambiar la apariencia de cada dato dependiendo incluso de su valor. Como vemos en la fig. 24 los colores cambian según el valor de cada elemento.

Verde para cuando está operativa la estación y dispone tanto de bicis como de huecos; amarillo cuando estando operativa no tiene huecos libres; rojo cuando estando operativa no dispone de bicis; magenta cuando se encuentra fuera de servicio. Además la distancia se expresa en dos formatos distintos dependiendo de si se encuentra a más de 1 km de distancia o no.

Cargar lista de estaciones Bicing		
44	Av Meridiana, 66	2,22 km NE
OPN	12 bicis	6 huecos
45	Marina, 65	1,62 km NE
OPN	17 bicis	0 huecos
46	Ramon trias Fargas, 19	1,68 km E
CLS	0 bicis	0 huecos
47	Ramon Trias Fargas	1,63 km E
OPN	0 bicis	26 huecos
48	Meridiana, 40	1,61 km NE
OPN	12 bicis	5 huecos
49	Rosa Sensat en front, 20	2,04 km E
OPN	3 bicis	17 huecos

Figura 24.- Captura de pantalla con la actividad lista de estaciones

Manteniendo pulsado (no haciendo clic y soltando) un elemento de la lista hace que se muestre un mensaje emergente (`Toast`) con un resumen del estado de la estación



Figura 25.- Captura de pantalla con el mensaje emergente al mantener pulsada una estación.

seleccionada (ver fig. 25). Esto se consigue implementando el método `onItemLongClickListener`:

```
@Override
public boolean onItemLongClick(AdapterView<?> a, View v,
    int position, long id) {
    Toast.makeText(getApplicationContext(), "Estación de Bicing nº"
+ listaFilas[position].getId() + " , ubicada en "
+ listaFilas[position].getDireccion() + ". Se encuentra a "
+ listaFilas[position].getEstado() + "\nDispone de "
+ listaFilas[position].getBicis() + " bicis libres y de "
+ listaFilas[position].getHuecos() + " huecos vacíos.",
    Toast.LENGTH_LONG).show();
    return false;
}
```

Por último, si se hace un clic normal, se enlaza con la actividad mapa centrando la vista en la estación seleccionada. La comunicación entre estas dos actividades se realiza pasando los parámetros por medio de un `bundle` (como si fuera una caja donde ponemos los valores que queremos pasar) y el lanzamiento de la actividad se realiza por medio de un `Intent` (desde actividad origen a actividad destino). Este lanzamiento ocurre en el momento en que se invoca el `startActivity` de acuerdo con los parámetros establecidos en las líneas previas.

```
@Override
public void onItemClick(AdapterView<?> a, View v, int position,
    long id) {
    Intent intent = new Intent(getApplicationContext(),
    MapActivity.class);
```

```

Bundle bundle = new Bundle();
bundle.putString("updateTime", updateTime);
bundle.putString("stationsTableName", stationsTableName);
bundle.putString("url", url);
bundle.putString("mapTypePref", "normal");
bundle.putBoolean("trafficLayerPref", false);
bundle.putBoolean("buildings3DPref", false);
bundle.putInt("id",
Integer.valueOf(listaFilas[position].getId()));
intent.putExtras(bundle);
startActivity(intent);
}

```

En cuanto al botón de cargar la lista para refrescar los datos se implementa con el método `OnClickListener`:

```

btn_cargar_estaciones = (Button) findViewById(R.id.btn_cargar_Lista);
btn_cargar_estaciones.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {

    Log.d(DEBUG_TAG_STATIONS_LIST_ACTIVITY, "Pulsado el botón
'Cargar lista de estaciones de Bicing'");

    updateList();
}

});

```

6.3.- Clase y actividad mapa

Previamente a usar mapas es necesario registrar la aplicación en la consola de *Google Play Services* para Desarrolladores y generar una clave para incluir en el archivo `AndroidManifest.xml`:

```

<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="AIzaSyCYA7cDYIjF_FOKsKFnsEB04xuzv1szCLA" />
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />

```

Este proceso es muy sencillo y simplemente es necesario registrarse y seguir los pasos descritos en la guía *Getting Started Google Maps Android API* [[GMA](#)]. Junto con las indicaciones de cómo realizar la configuración para poder utilizar los mapas (en

especial cómo obtener la clave API y configurar las bibliotecas de utilidades), también se encuentran unos ejemplos muy útiles de los que se ha partido para adaptarlos a la actividad mapa [[MSMPL](#)] y para emplear en la agrupación o *clusterización* [[CLUS](#)] de marcadores, adaptando la clase `MultiDrawable`.

En este caso la actividad se ha planteado como un `Fragment` para poder mostrar

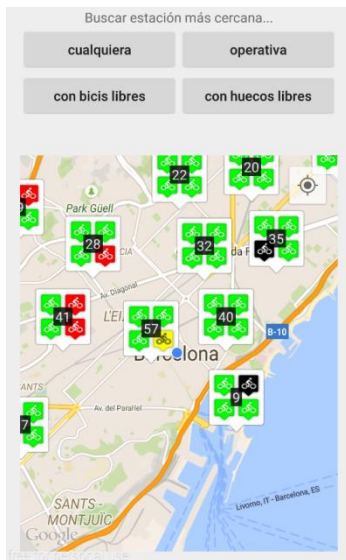


Figura 26.- El mapa centrado en la ubicación del usuario y con marcadores agrupados.



Figura 27.- Después de hacer zoom hacia adentro los clúster se van desagrupando.



Figura 28.- Si hacemos clic en un clúster nos aparece Toast mostrando detalle.



Figura 29.- Después de pulsar el botón 'cualquiera' la vista se centra en la estación más cercana (sin condiciones) y se muestra Toast con detalle.



Figura 30.- Haciendo zoom vemos los detalles de la ruta.



Figura 31.- Haciendo clic en el icono de la estación aparecen detalles sobre su estado y situación.

en una parte (fragmento) de la pantalla el mapa y dejar el espacio restante para los botones de búsqueda de la estación más cercana y para los cuadros de texto que muestran la información al usuario.

En cuanto a la tarea `ShowBicingRouteTask` se ha planteado como una tarea asíncrona ya que necesita realizar una conexión y una petición GET a la URL de *Google* que calcula rutas entre dos coordenadas. Asimismo se ha separado en la clase `BicingRoute` la problemática de enviar esta petición y tratar la respuesta para dibujar posteriormente la ruta en el mapa como una polilínea (`Polyline`). Para el tratamiento de la respuesta se ha adaptado de `GMapV2Direction` [[GMAPV2](#)]. En las fig. 26 a 31 vemos el resultado de la implementación en el emulador. Como se observa, se han empleado diferentes colores para los iconos de los marcadores y de los clústers de marcadores con objeto de distinguir el estado de las estaciones a simple vista. Negro, fuera de servicio; verde, operativa; amarilla, operativa sin huecos libres; roja, operativa sin bicis libres.

En cuanto a la ruta, es posible configurar desde el menú de preferencias (veremos los menús de ajustes más adelante) si se desea la ruta a pie, en bici (si hay carriles bici en el trayecto), u otras, de la misma manera que se puede hacer en la *web* de *Google Maps*.

6.4.- Resto de clases y actividades

6.4.1.- Clase y actividad de predicción meteorológica

Para esta actividad `WeatherForecastActivity`, también se ha empleado una tarea asíncrona `DownloadWeatherXMLTask` para la descarga de los datos desde el catálogo de *open data*. En este caso no se considera necesario mantener ninguna base de datos, una vez leídos los datos se presentan al usuario en la interfaz. Para que resulten más atractivos visualmente se han ubicado una serie de iconos en las carpetas de recursos para mostrar también de manera gráfica la descripción del pronóstico (icono de nubes para pronóstico nublado, etc.), de modo que dependiendo del valor descargado se muestra un icono u otro como puede verse en la fig. 32.



Figura 32.- Detalle de la actividad de pronóstico meteorológico.

Esta tarea asíncrona llama al método `loadWeatherXMLFromNetwork` que es el encargado de establecer la conexión y parsear el fichero XML. Para ello, del mismo modo que para parsear los datos de las estaciones, se emplea una clase `BarcelonaWeatherForecastXMLParser` que es la que lee específicamente los datos saltando las etiquetas del XML. Esta clase es asimismo la que contiene los métodos para traducir los códigos descriptivos a cadenas de texto o iconos de las carpetas de recursos para presentar la información al usuario.

6.4.2.- Clase y actividad de ayuda

La clase `HelpActivity` es puramente gráfica y no tiene ningún tipo de interacción con el usuario ni conexión con *open data*. Puede verse una captura de pantalla en la fig. 33. Mediante un *scrollview* el usuario hace deslizar el dedo para desplazarse hacia abajo (o hacia arriba) de la ayuda con el mismo efecto que mover una barra de desplazamiento.



Figura 33.- Captura de pantalla de la actividad de ayuda al usuario

6.4.3.- Clase y actividad curiosidades estadísticas

Aprovechando que tenemos almacenada la base de datos con la información actualizada de las estaciones, se ha implementado esta sencilla actividad en la clase `CuriositiesActivity`. Lo interesante de esta actividad es el empleo de cursores para recorrer la tabla de la BD y buscar valores. En las figs. 34 a 36 se aprecia lo que visualiza el usuario en la pantalla.

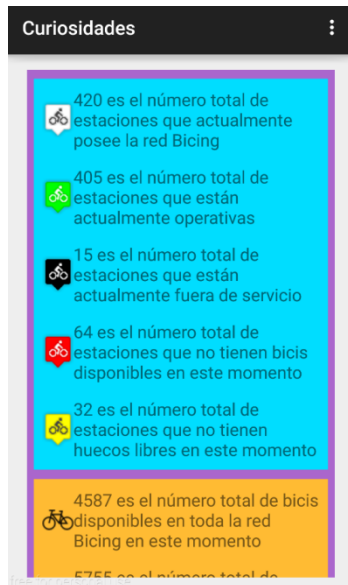


Figura 34.- Detalle actividad curiosidades (1).

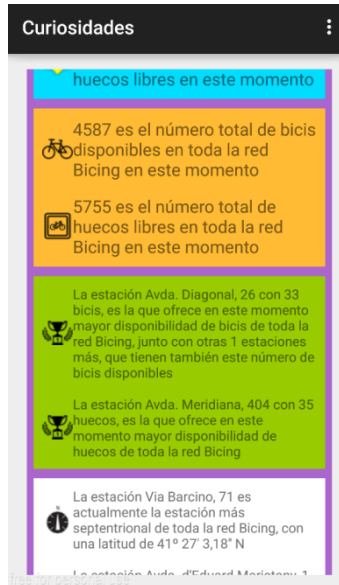


Figura 35.- Detalle actividad curiosidades (2).

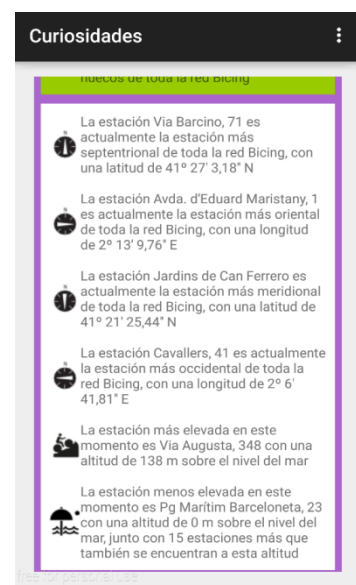


Figura 36.- Detalle actividad curiosidades (y 3).

6.4.4.- Clase y actividad About...

Una sencilla clase `AboutActivity` muestra información sobre el autor de la aplicación (ver captura de pantalla en la fig. 37).

6.5.- Menús de preferencias

Para implementar los menús de preferencias (que aparecen cuando se pulsa la tecla menú del dispositivo), *Android* proporciona el objeto `SharedPreferences`.



Figura 37.- Detalle de la actividad About.

Capítulo 7.- Conclusiones y líneas futuras

A lo largo de esta memoria se ha expuesto el trabajo realizado en este proyecto que pone fin a los estudios de Ingeniería Técnica de Telecomunicación, especialidad en Telemática.

Se ha introducido la problemática del ciclismo urbano como medio de transporte urbano y se han discutido las ventajas e inconvenientes de los sistemas de bicicletas compartidas. A continuación, se ha presentado la filosofía *Open Data*, y la importancia que está adquiriendo en nuestra sociedad, estableciendo posteriormente la relación de los datos en abierto con la aplicación objeto de este TFC. Asimismo se ha descrito la arquitectura y filosofía del SO *Android* y se han puesto de relieve sus principales características, explicando las razones por las que se ha elegido para realizar esta *app*.

7.1.- Objetivos alcanzados y funcionalidades implementadas y descartadas

Se ha partido de una aplicación básica para diseñar e implementar la aplicación, construyendo paso a paso las diferentes funcionalidades que se habían propuesto en un principio, cumpliendo con los objetivos principales que se recapitulan a continuación:

- Mostrar en un mapa las estaciones de *Bicing* permitiendo la interacción con el usuario y la consulta de información sobre las estaciones.
- Mantenimiento de una BD con la información de las estaciones obtenida de los catálogos de *open data*.
- Búsqueda de la estación más cercana a la ubicación del usuario, incluyendo los siguientes criterios:
 - Cualquier estación
 - Estación operativa más cercana (descarta las que se encuentran fuera de servicio)
 - Estación más cercana con bicis disponibles (descarta estaciones que no tienen bicis disponibles)

- Estación más cercana con huecos libres (descarta estaciones que no tienen huecos libres)
- Mostrar la predicción meteorológica del tiempo que va a hacer en la zona en las siguientes horas.

Por otra parte, ha habido una serie de funcionalidades que se pretendían implementar y que ha sido necesario desestimarlas durante el desarrollo del proyecto. Se relacionan a continuación:

- Análisis de tendencias.- En un principio se había planteado la posibilidad de incorporar una función de análisis de tendencias que implementara un servidor que se encargara de mantener una base de datos con el estado de cada estación *Bicing* y realizar un análisis de tendencia de ocupación/disponibilidad de huecos/bicis de acuerdo con los cambios producidos durante un periodo de tiempo (p.ej. durante la última hora). Por una parte, se trata de un ejercicio interesante que consiste en, uno, implementar el servidor con la BD y, dos, establecer la comunicación entre la *app* y dicho servidor. Sin embargo, por otra parte, no tiene mucho sentido desde un punto de vista práctico. Es decir, si la *app* tiene que comunicarse con un servidor para acceder al valor de estas tendencias, entonces por qué no implementar toda la BD y los accesos a los catálogos de *open data* en el lado del servidor y diseñar por tanto la *app* como un mero cliente que lanza consultas al servidor. Desde un punto de vista de considerar el TFC como un ejercicio práctico, en el que la finalidad sea la adquisición de habilidades y conocimientos más que la utilidad real de la aplicación, entonces tendría cabida este planteamiento. No obstante se toma la decisión de no implementar esta funcionalidad puesto que la *app*, por la propia definición del proyecto, interesa que gestione su propia BD.
- Mostrar información sobre los niveles de polen.- Lamentablemente no es posible utilizar los datos en abierto sobre el nivel del polen a causa de la forma en la que se encuentran en el catálogo en abierto (incumpliendo uno de los principios de la filosofía *open data* expuestos en el Estado del arte). Como la información no se encuentra siempre en el mismo formato y éste no es predecible no se puede implementar esta funcionalidad. Al menos desde

este catálogo. Se han buscado catálogos alternativos pero sólo existe esta fuente de información.

En tercer lugar, ha habido funciones que se han implementado y que, sin embargo, no se habían definido en un principio:

- Durante el desarrollo de la aplicación ha surgido de manera natural la necesidad de añadir una pantalla que muestre al usuario una ayuda para utilizar la aplicación y otra pantalla tipo *About...* que muestre información sobre el autor de la aplicación.
- Además, también aparece de manera intuitiva la necesidad de incorporar una pantalla que muestre al usuario informaciones estadísticas sobre el servicio *Bicing*, aprovechando las posibilidades de la BD que se implementa y el conocimiento sobre el desarrollo de la interfaz de usuario que se va adquiriendo paralelamente a la implementación de la *app*.

7.2.- Líneas de trabajo futuras

Sin duda, ha sido un reto el desarrollo de la aplicación, buena parte del tiempo se ha destinado a aprender a programar *apps* en *Android*, un SO que sufre continuas actualizaciones y lanzamiento de nuevas versiones. Durante la realización del TFC además de aparecer nuevas versiones del API de este SO también ha cambiado el IDE con soporte oficial que había venido siendo Eclipse y que ahora ha pasado a ser *Android Studio*, un IDE específico para *Android* que ha sido desarrollado por Google. La primera tarea si se desea seguir evolucionando esta aplicación sería migrar el proyecto al nuevo entorno de desarrollo.

En cuanto a los objetivos opcionales, no han podido llevarse a cabo, por no haber dispuesto de tiempo pero sin duda también representan las líneas futuras sobre las que trabajar:

- En efecto, el mundo *wearable* ha hecho su irrupción en estos dos últimos años, por lo que una posible evolución de la *app* sería adaptarla a un *smartwatch*.

- Otra posibilidad es introducirse en el desarrollo con iOS, la otra gran plataforma dentro del mundo de los dispositivos móviles e implementar la aplicación para este SO.

7.3.- Puntos fuertes del proyecto y aspectos que se cambiarían una vez concluido

Como punto fuerte de la *app* se destaca el hecho de que implementa varias soluciones para lograr el propósito de mostrar información valiosa al usuario de *Bicing*, incluyendo los siguientes aspectos:

- Es capaz de gestionar la conexión en función de las preferencias de red que ha fijado el usuario.
- Divide las tareas para que se ejecuten de manera asíncrona para que el usuario no pierda interacción con la *app* mientras se procesan (p.ej. tareas como conexión y descarga de datos, o escritura en base de datos que pueden consumir más de un segundo de tiempo).
- Ventajas con respecto a la *app* oficial de *Bicing* como p.ej.: el usuario puede ajustar las preferencias de conexión comentadas, o el tipo de mapa, los marcadores se presentan agrupados para no sobrecargar la pantalla con decenas de marcadores, tipo de medio de transporte para calcular y trazar la ruta (a pie, en coche, etc.), utilización de colores para distinguir a primera vista el estado de las estaciones sin necesidad de hacer clic para ver el detalle.

En cuanto a lo que se cambiaría del proyecto una vez concluido, se destacan un par de aspectos:

- El aspecto visual. En este sentido cuando se definió el proyecto se tenía en mente una *app* que explotará la cantidad de recursos de que dispone la plataforma *Android* (al comenzar el TFC la versión 4.4 *KitKat* y ahora mismo la 5.1 *Lollipop* y ya se anuncia el próximo lanzamiento de la siguiente versión – cuyo nombre empezará por M) para diseñar una interfaz gráfica más atractiva. Un modelo de buen diseño gráfico que se pretendía imitar es la de la *app Yahoo Tiempo*, que está reconocida por ser una de las mejor diseñadas del mercado. Sin embargo, no se ha podido disponer del tiempo necesario para aprender las técnicas a emplear.

- La implementación de alguna funcionalidad adicional. P.ej. búsqueda de estaciones por nombre o análisis de tendencias de disponibilidad de las estaciones.

Bibliografía

[1] **Cinar, Onur** (2012). *Android Apps with Eclipse* (1ª ed.) Nueva York: Apress.

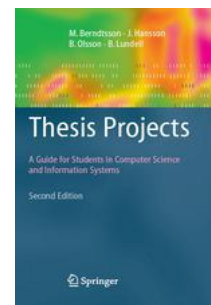
Comentarios: Este libro está dirigido a desarrolladores con nivel principiante o intermedio que deseen programar aplicaciones *Android* con Eclipse, independientemente de la versión de este IDE con la que trabajen. La que se ha escogido es la Java IDE por ser la más popular. En los primeros capítulos se describe *Android* y su arquitectura y los fundamentos del desarrollo de aplicaciones en esta plataforma. A continuación se hace una introducción histórica



a Eclipse y se detalla cómo realizar una instalación y cómo configurar el entorno para el desarrollo en *Android*. En los capítulos centrales se explica la manera de utilizar y dominar esta herramienta IDE. Finalmente, los últimos capítulos tratan de conceptos más avanzados como la incorporación de código nativo en C/C++ mediante el kit NDK y el depurado y solución de problemas.

[2] **Berndtsson, Mikael; Hansson, Jörgen; Olsson, Björn; Lundell, Björn** (2008). *Thesis Projects. A Guide for Students in Computer Science and Information Systems*. (2ª ed.) Londres: Springer-Verlag.

Comentarios: Los autores, profesores de la *University of Skövde* en Suecia y del *Software Engineering Institute* de la *Carnegie Mellon University* en Pittsburgh, (Pensilvania, EE.UU.), describen en el libro el proceso de conducción de proyectos como tesis y trabajos de fin de carrera de las disciplinas de Informática (*Computer Science and Information Systems*) que siguen en sus universidades. Consideran que todo proyecto de fin de carrera puede tener diferentes propósitos:



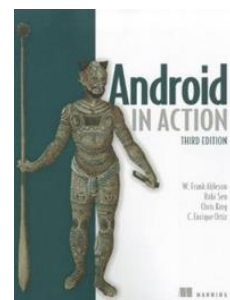
- ahondar en el aprendizaje (como una oportunidad de estudiar una materia en profundidad);
- ser un trampolín para la búsqueda de trabajo (como una preparación para la vida laboral, poniendo en práctica en problemas del mundo real los conocimientos y habilidades adquiridos durante los estudios);

- ser un peldaño hacia los estudios de postgrado y doctorado (como una preparación para la actividad de investigación).

Está concebido como una guía para los tres actores de cualquier proyecto: el estudiante, el supervisor y el examinador. Se estructura en tres partes. La primera parte del libro (caps. 1-3) proporciona una introducción general a los proyectos de esta área. La segunda parte (caps. 4-12) describe en detalle el proceso de llevar a cabo el proyecto. Por último, la tercera parte (caps. 13-15) contiene información complementaria con consejos para el estudiante sobre cómo buscar información y cómo escribir una memoria así como recomendaciones para los examinadores y supervisores sobre cómo evaluar los proyectos de los alumnos.

[3] **Ableson, Frank; Sen, Robi; King, Chris; Ortiz, Enrique** (2011). *Android in Action*. (3ª ed.) Greenwich, CT (Connecticut): Manning Publications Co.

Comentarios: Última edición de otro manual de los autores de una de las más representativas guías para desarrolladores, *Unlocking Android*, muy popular en el mundo hispanohablante por tratarse de una de las primeras obras sobre el tema que se tradujeron al castellano (publicada por Anaya Multimedia con el título de *Android – Guía para desarrolladores*). El libro constituye una guía práctica y a la vez exhaustiva de la programación de aplicaciones en este SO. Abarcan temas asequibles para el principiante como la introducción a *Android*, la descripción de la arquitectura y la instalación y configuración del entorno de desarrollo, pasando por capítulos de nivel intermedio y más avanzado que se han ido completando, ampliando y corrigiendo durante las sucesivas ediciones. Abundantes casos prácticos y código fuente de ejemplo ayudan a asimilar los conceptos aprendidos.



[4] **Android Open Source Project by Google Inc.** (2014). *Android Developers Web Site* [Última fecha de consulta 29 de septiembre 2014] URL: <http://developer.android.com/develop/index.html>

Comentarios: La *web Android Developers* es el sitio oficial donde buscar referencia y soporte para desarrollar aplicaciones en *Android*. El contenido está publicado bajo licencia *Creative Commons Attribution 2.5*. En un tutorial denominado *Training* se enseñan los primeros pasos para comenzar a desarrollar *apps*, con una serie de lecciones describiendo cómo llevar a cabo distintas tareas específicas, conteniendo códigos de ejemplo que pueden reutilizarse en otras aplicaciones. De manera progresiva se aprende, primero a instalar el kit *Android SDK* y configurar el entorno de desarrollo, y posteriormente se crea y ejecuta una primera *app* de prueba. Todo el proceso está permanentemente guiado y es la base para pasar al resto de tutoriales donde se tratan en profundidad otros temas. Entre otros recursos, la página ofrece una completa colección de ejemplos y una exhaustiva referencia de todas las bibliotecas y clases del SDK. Desde este sitio se encuentra también el enlace a la referencia y soporte del *Google Maps Android API* que se empleará para la realización del TFC.

[5] **Mittelbach, Frank; Goossens, Michel** (2004). *The LaTeX Companion*. (2^a ed.) Boston: Addison-Wesley.

Comentarios: La popularidad de LaTeX en el mundo académico ha propiciado la existencia de una inabarcable colección de libros y recursos en la Red. Se ha escogido esta publicación por tratarse de un reputado manual de referencia en la materia. En el mismo se introduce históricamente LaTeX exponiendo las ventajas y diferencias con respecto a otros procesadores de texto. A continuación se detalla la estructura del documento y los comandos básicos. Más adelante se profundiza en el diseño y la composición de cada página del documento. Los capítulos siguientes se especializan en aspectos concretos como el entorno matemático, las tablas, tipos de fuentes, matemáticas avanzadas, escritura en distintos idiomas y manipulación de imágenes y gráficos. Por último, se trata el tema de la automatización de tareas, como la generación de índices, la gestión de citas y bibliografía. En los anexos se detalla cómo configurar la instalación del sistema y actualizarla con multitud paquetes para diferentes aplicaciones.



Referencias

N.B.- Se ha preferido evitar la inclusión de las direcciones URL completas a la hora de citar los recursos electrónicos porque, dada su excesiva extensión, además de resultar poco estético, dificultarían la lectura del documento al ocupar en muchos casos varios renglones. En su lugar, se ha utilizado un simple vínculo con el enlace a la dirección correspondiente para ser consultado cuando se utiliza un lector cualquiera de archivos PDF entre los muchos que hay accesibles gratuitamente. Basta situar el cursor encima del mismo para, si se desea, visualizar la URL completa sin necesidad de acceder a la misma. A pesar del hecho de que, efectivamente, la versión impresa en papel no contiene las direcciones a estos recursos, se entiende que la práctica generalizada para consultarlos consiste en hacer clic en el enlace o bien en copiar y pegar la dirección en el navegador, por todo ello no tiene mucho sentido incluirlas para ser impresas en papel.

[HER04] **Herlihy, D.** (2004) *Bicycle: the History*. New Haven (Connecticut): Yale University Press. Comentarios: algunas páginas pueden consultarse en línea con el recurso Google Books en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[WOM14] **Worldometers Real Time Statistics.** (2014) *Bicycles produced this year*. [Estadística en tiempo real en línea en el siguiente [enlace](#)]. Comentarios: *Worldometers* es un grupo internacional de desarrolladores, investigadores y voluntarios con el propósito de hacer accesible a un amplio espectro de público de todo el mundo una serie de estadísticas en tiempo real que inducen a la reflexión sobre el escenario global actual. No tiene ningún tipo de afiliación política, gubernamental, ni corporativa. Ha sido elegida como una de las tres mejores *webs* de referencia por la *American Library Association* y proporciona información a organizaciones como Naciones Unidas en la Conferencia de Desarrollo Sostenible Río+20 o a los informativos de la BBC. Más información en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[WW14] **WorldWatch Institute.** (2014) *Bicycle Production Reaches 130 Million Units*. [Artículo en línea en el siguiente [enlace](#)]. Comentarios: WorldWatch Institute es un instituto independiente de investigación fundado en 1974 dedicado a la

concienciación global sobre el medio ambiente y el desarrollo sostenible. Más información en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[IBF14] **International Bicycle Fund.** (2014) *Bicycle Statistics: Usage, Production, Sales, Import, Export*. [Artículo en línea en el siguiente [enlace](#)]. Comentarios: *International Bicycle Fund* es una ONG que proporciona información y recursos para promover el transporte sostenible y su concienciación internacional. Más información en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[GAR04] **Gardner, G.** (2004) “Bicycles, A Transport Revolution”. *Goodstuff Guide, A Behind the Scenes Guide to the Things We Buy – from The WorldWatch Intitute*. [Artículo en línea en el siguiente [enlace](#)]. Comentarios: En el artículo, incluido dentro de una guía publicada por el *WorldWatch Institute*, se comentan una serie de estadísticas de interés sobre el empleo de la bicicleta para el transporte urbano. [Última consulta 1 abril 2015].

[GRS13] **G.R.S.** (2013, 13 de junio) “Ámsterdam sigue prefiriendo las bicicletas – aunque colapsen la ciudad”. *lainformacion.com* [Artículo en línea en el siguiente [enlace](#)]. [Última consulta 2 abril 2014].

[LKY] Más detalles sobre la trayectoria de este dirigente y sobre el premio Lee Kuan Yew World City Prize en el siguiente [enlace](#). [Última consulta 1 abril 2015]

[HIN] **Hinojosa, J.** *El uso de la bicicleta como medio de transporte urbano*. [Artículo en línea en el siguiente [enlace](#)] Comentarios: Se trata de una serie de cápsulas informativas sobre el tema presentadas por el Presidente del Comité Pro-bici de Sevilla. [Última consulta 1 abril 2015].

[DEK99] **Dekoster, J.; Schollaert, U.** (1999) *Cycling: the way ahead for towns and cities*. [Publicación en línea en el siguiente [enlace](#)]. Comentarios: Se ha editado una traducción al castellano: *En bici, hacia ciudades sin malos humos*, que puede consultarse en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[GRA] Consultado en nota de prensa en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[SHA11] **Shaheen, S.; Guzman, S.** (2011) “Worldwide Bikesharing”. *Access, the Magazine of the University of California Transportation Center* (Number 39, Fall 2011, pp. 22-27). [Artículo en línea en el siguiente [enlace](#)]. [Última consulta 1 abril 2015].

[DEM09] **DeMario, P.** (2009) “Bike-sharing: Its History, Models of Provision, and Future”. *Conferencia Velo-city 2009 celebrada en San Jose, California*. [Artículo en línea en el siguiente [enlace](#)]. [Última consulta 1 abril 2015].

[VEL] Más información en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[COP] Más información en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[COP2] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[GAR11] **Garthwaite, J.** (2011, 7 de junio) “Bike-Share Schemes Shift Into High Gear”. *National Geographic Daily News*. [Artículo en línea en el siguiente [enlace](#)]. [Última consulta 1 abril 2015].

[BIC] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[SEV] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[VAL] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[BIZ] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[TCI] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[GNU] Más información en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[OCW] Más información en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[ODE] Más información en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[ODE2] Más información en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[OKF] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[OKF2] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[CTI] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[GOB] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[BCN] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[CAT] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[BCN2] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[CAT2] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[CAT3] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[CAT4] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[AND] **Elgin, B.** (2005, 16 de agosto) “Google Buys Android for Its Mobile Arsenal”. *Bloomberg Businessweek*. [Artículo en línea en el siguiente [enlace](#)]. [Última consulta 1 abril 2015].

[OHA] Más información en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[SIE] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[TIC] I Congreso @dministración: Las TIC al Servicio del Ciudadano. [Vídeo completo de todas las ponencias disponibles en el siguiente [enlace](#)]. [**Se asistió en persona al congreso el 1 abril 2014**].

[AND2] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[GPY] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[IDC] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[KAS] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[ELP] (2013, 14 de octubre) “La mitad de las aplicaciones están muertas”. *El País*. [Artículo en línea en el siguiente [enlace](#)]. [Última consulta 1 abril 2015].

[PRI] **Prieto Blázquez, J.; Ramírez Vique, R.; Morillo Pozo, J.; Domingo Prieto, M.** (2011). *Tecnología y desarrollo en dispositivos móviles*. (1ª ed.) Barcelona: FUOC.

[SDK] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[ADT] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[MAP] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[DIS] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[OFI] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[BIB] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[CYC] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[CBK] Consultado en el siguiente [enlace](#). [Última consulta 1 abril 2015].

[ADB] Consultado en el siguiente [enlace](#). [Última consulta junio 2015].

[ACY] Consultado en el siguiente [enlace](#). [Última consulta junio 2015].

[CODAB] Consultado en el siguiente [enlace](#). [Última consulta junio 2015].

[CCBY] Consultado en el siguiente [enlace](#). [Última consulta junio 2015].

[XMLB] Consultado en el siguiente [enlace](#). [Última consulta junio 2015].

[CAT] Consultado en el siguiente [enlace](#). [Última consulta junio 2015].

[XMLM] Consultado en el siguiente [enlace](#). [Última consulta junio 2015].

[ACT] Consultado en el siguiente [enlace](#). [Última consulta junio 2015].

[LSTV] Consultado en el siguiente [enlace](#). [Última consulta junio 2015].

[SMPL] Consultado en el siguiente [enlace](#). [Última consulta junio 2015].

[NWKCNT] Consultado en el siguiente [enlace](#). [Última consulta junio 2015].

[MNGNWK] Consultado en el siguiente [enlace](#). [Última consulta junio 2015].

[XMLPRS] Consultado en el siguiente [enlace](#). [Última consulta junio 2015].

[GMA] Consultado en el siguiente [enlace](#). [Última consulta junio 2015].

[MSMPL] Consultado en el siguiente [enlace](#). [Última consulta junio 2015].

[CLUS] Consultado en el siguiente [enlace](#). [Última consulta junio 2015].

[GMAPV2] Consultado en el siguiente [enlace](#). [Última consulta junio 2015].

Anexo.- Mapa conceptual

