

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

**ESCUELA DE INGENIERÍA DE
TELECOMUNICACIÓN Y ELECTRÓNICA**



PROYECTO FIN DE CARRERA

***Aplicación para móviles de acceso a información sobre situación
de transportes***

TITULACIÓN: Ingeniería Técnica de Telecomunicaciones especialidad en Sistemas Electrónicos.

TUTORES: Álvaro Suárez Sarmiento

Elsa María Macías López

AUTOR: Héctor Pérez Arteaga

FECHA: Julio 2014

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

ESCUELA DE INGENIERÍA DE

TELECOMUNICACIÓN Y ELECTRÓNICA



PROYECTO FIN DE CARRERA

***Aplicación para móviles de acceso a información sobre situación
de transportes***

Presidente:

Secretarios:

Vocal:

Tutores:

Autor

NOTA:.....

TITULACIÓN: Ingeniería Técnica de Telecomunicaciones especialidad en Sistemas Electrónicos.

TUTORES: Álvaro Suárez Sarmiento

Elsa María Macías López

AUTOR: Héctor Pérez Arteaga

FECHA: Julio 2014

Índice

1.	Introducción.....	9
1.1.	Ámbito del proyecto	10
1.2.	Transporte público en las ciudades.....	11
1.3.	Objetivos de este proyecto	12
1.4.	Estructura de la memoria.....	13
2.	Estudio de herramientas utilizadas.....	15
2.1.	Sistema operativo Android	16
2.1.1.	Características generales.....	16
2.1.2.	Arquitectura de Android.....	19
2.1.3.	Estructura de una aplicación Android	20
2.1.4.	Jerarquía visual de Android	21
2.2.	SDK de Android	23
	Dalvik Debug Monitor Service	23
	Plugin ADT para Eclipse	24
2.3.	Extensible Markup Language	25
2.4.	Simple Object Access Protocol.....	27
2.5.	Otras herramientas	28
3.	Análisis previo y funcional	33
3.1.	Análisis Previo	34
3.1.1.	Arquitectura final del proyecto	37
3.1.2.	Otras aplicaciones similares	37
3.2.	Análisis Funcional.....	38
3.2.1.	Módulo de simulación de datos	38
3.2.2.	Módulo Aplicación móvil Android.....	40
3.2.3.	Módulo de servicio web	43
4.	Análisis orgánico	45
4.1.	Módulo de servicio web Java	46
4.1.1.	Introducción	46
4.1.2.	Funcionamiento.....	47
4.1.3.	Estructura del servicio web	50
4.2.	Modulo Aplicación móvil Java de Android.....	54

4.2.1. Introducción	54
4.2.2. Archivo XML de localización de parada.....	56
4.2.3. Submódulo Activity del Menú principal.....	61
4.2.4. Submódulo Activity donde encontrar parada.....	66
4.2.5. Submódulo Activity de muestra de resultado.....	69
4.2.6. Estructura del proyecto Eclipse.....	72
4.2.7. Módulo de simulación.....	76
5. Descripción a nivel usuario	79
5.1. Preparando el terminal	80
5.2. Iniciar la aplicación.....	82
5.3. Menú principal	82
5.4. Opción Mostrar mapa	84
5.5. Opción Paradas cercanas a mí	86
5.6. Opción Paradas cercanas a una zona.....	88
5.7. Información de parada.....	89
6. Conclusiones y posibles ampliaciones	91
6.1. Conclusiones	92
6.2. Posibles ampliaciones	93
Pr. Pliego de condiciones y presupuesto	95
Pr.1 Pliego de condiciones	96
Pr.1.1 Descripción del proyecto	96
Pr.1.2 Requisitos de hardware	96
Pr.2 Presupuesto.....	98
Pr.2.1 Descripción de las fases del proyecto	98
Pr.2.2 Mano de obra.....	100
Pr.2.3 Coste del material	100
Pr.2.4 Coste total del proyecto.....	100
Anexo A: Datos de paradas	103
Referencias.....	121
Glosario.....	123

1. Introducción

En este capítulo se presenta el alcance del *Proyecto Final de Carrera (PFC)*, sus objetivos y la estructura de la memoria.

1.1. **Ámbito del proyecto**

En los últimos años, hemos asistido a un crecimiento espectacular de las posibilidades de comunicación que ofrecen los smartphones. Cada día aparecen nuevas funcionalidades en el mercado, que no dejan de apasionar a los compradores que se cuentan por miles de millones en todo el Mundo. Han dejado de ser teléfonos móviles para convertirse en ordenadores en miniatura. Pero añaden un conjunto de sensores unificados en un único dispositivo que aún no han dado todo lo que tienen que dar. Precisamente este conjunto de sensores es lo que les hace tener una capacidad de procesamiento mayor que el de los computadores portátiles al uso.

El hecho de la continua bajada de precios, unido a la existencia de plataformas software de libre acceso en las que desarrollar soluciones ha hecho que proliferen este tipo de dispositivos enormemente en los últimos años. Esto unido a los continuos avances y posibilidades de la computación moderna hace pensar que todavía están por verse muchas soluciones y usos para estos terminales. El ámbito de este PFC es justamente el de los teléfonos móviles y sus aplicaciones modernas. Para realizar este PFC hemos utilizado el lenguaje de programación Java y también introducirnos en el desarrollo de aplicaciones Android. La razón de ello es que actualmente es una de las plataformas más utilizadas en el Mundo para llevar a cabo el desarrollo de aplicaciones móviles. Además de que el índice de ventas de teléfonos móviles Android es el mayor comparado con el de otras plataformas actuales.

Por otro lado, el uso de sensores en los teléfonos móviles eficazmente proporcionaría a sus usuarios posibilidades apenas explotadas a día de hoy. Un ejemplo de estos sensores es el *Global Positioning System (GPS)*. Aunque hay muchas aplicaciones que lo usan, todavía se puede explotar más en la vida de a diario de los ciudadanos.

Usando el GPS, en concreto, nos acercamos a un problema en el que el teléfono móvil proporciona soluciones muy elegantes para la Vida que desarrolla un ciudadano de una gran ciudad genérica. Además de resolver un problema cotidiano que suele quitar mucho tiempo de a diario y por otro lado mejoraría las prestaciones de los servicios públicos en dichas ciudades.

1.2. Transporte público en las ciudades

La ciudad de Las Palmas de Gran Canaria cuenta con un servicio público de transporte. Actualmente sólo existe una compañía pública de servicio de guaguas urbanas. Estas son el medio de transporte diario único para cientos de ciudadanos. Su uso, como transporte público, presenta ciertos inconvenientes. El desconocimiento de los recorridos, la red o los horarios hace motivan la pérdida de usuarios al servicio de transporte. Especialmente aquellos usuarios que pueden elegir y optan por otras alternativas. Como puede ser el coche particular. A la hora de optimizar el tiempo y moverse, el viajero puede encontrar un gran problema en la incertidumbre que le supone no conocer con cierto rigor la hora de llegada del siguiente vehículo. Esto se agrava cuando nos encontramos en paradas o líneas no conocidas, o próximos a la finalización del servicio.

Muchas veces, simplemente no podemos asumir la posibilidad de que un autobús llegue con demora, otras nos vendrían bien pero no tenemos forma de saberlo. En relación a ello está la incomodidad, ya sea por la falta de espacio o por la obligación de permanecer en la parada un tiempo indeterminado. Quizás con lluvia o de noche, con los ojos puestos en la carretera. Y ¿por qué no decirlo? puede ser muy aburrido. Esta incomodidad podría ser fácilmente subsanada si tuviésemos conocimiento exacto del momento de la llegada.

Si el viajero tuviera oportunidad de optimizar el tiempo en caso de demora, saliendo con el tiempo justo del trabajo, de la casa, resguardarse en un local o simplemente permanecer en un parque cercano hasta el momento antes de la llegada del autobús, entonces podría hacer una estimación acertada del coste en tiempo que le supondría su trayecto en transporte público en cada momento. En definitiva tendría más control sobre el tiempo y probablemente la valoración Social de este medio de transporte mejoraría. Por el contrario, encontrarse en una parada sin información, pensando que podría haber una avería o incidencia, fomenta una experiencia negativa. Y constituye una precarización evitable del servicio. A este respecto, la empresa de transporte colocó hace unos años unos paneles informativos en algunas paradas. Mediante un servicio de localización GPS integrado en la flota, un servidor hace

estimaciones en tiempo real. Dicha información puede ser consultada en algunas paradas de la red. Ello supone un avance.

1.3. Objetivos de este proyecto

Nosotros encontramos varios puntos mejorables en la instalación de puntos de información en las marquesinas:

- El primero es la escasa implantación. Podemos encontrar estos paneles en contadas marquesinas.
- Suelen estar instalados en paradas de mucha afluencia y bien comunicadas con el resto de la red sin grandes problemas de frecuencia.
- Pueden pasar desapercibidos para aquellos usuarios estresados que no reparan en la información que les proporciona su entorno.

Nosotros entendemos que lo más eficiente es que el usuario conozca tales datos sin necesidad de encontrarse en ese lugar: no puedes ver el panel desde cierta distancia, y la estimación puede no ser precisa. Un aviso acústico o una vibración en el momento exacto podrían evitar mantener pendientes del reloj al viajero y facilitaría la vida a los disminuidos visuales.

El objetivo de este proyecto no es otro que el de ofrecer una aplicación móvil a los viajeros que facilite el uso de la guagua como transporte público.

Se pretende ser práctico y que nuestro software no sea una carga, que sea ligero y rápido para su uso en cualquier situación. Por ello el software debería facilitar la información de forma intuitiva, sin explicación previa. Y debería disponer de atajos que lo hagan viable para que se pueda usar sin mucha carga de tiempo adicional por parte de los viajeros en su vida diaria.

Por todo lo anterior hemos planteado la implantación de un prototipo de prueba y ensayo como demostración de las posibilidades que este servicio tendría. No se trata, por tanto, del diseño de una aplicación profesional: a día de hoy existen algunas soluciones profesionales de este tipo que aportan sus soluciones de manera

muy eficiente, lo que nos refuerza la idea inicial que tuvimos cuando se elaboró el anteproyecto de este PFC. Se plantea además este PFC como un trabajo inicial susceptible de demostrar las posibilidades que tiene este servicio tanto para el viajero como para la empresa de transporte.

1.4. Estructura de la memoria

La memoria de este PFC se ha redactado estructurándola en 6 capítulos:

- Introducción.
- Estudio de herramientas.
- Análisis previo y funcional.
- Análisis orgánico.
- Descripción de interfaz de usuario.
- Conclusiones.

En el capítulo 2 enumeramos las herramientas de las que nos hemos valido para elaborar la aplicación móvil y damos una breve descripción de cada una de ellas.

El análisis previo o de requisitos investiga cuales son los elementos que necesitamos si queremos implementar nuestra solución. Y cuáles son las necesidades que hemos de satisfacer, da paso al análisis funcional que se adentra un poquito más, dividiendo el proyecto en módulos de funcionalidad aunque sin entrar en la parte técnica. Estos análisis se presentan en el capítulo 3.

El análisis orgánico, del capítulo 4, parte del análisis funcional. En él describimos la parte técnica y explicamos ciertas decisiones que vamos tomando al atacar el problema.

La descripción a la que se refiere el capítulo 5 podría servir de guía de usuario. Hacemos en ella un recorrido por la parte visual de la aplicación, dando a conocer su funcionamiento visto por el usuario final.

El capítulo 6 trata sobre las conclusiones finales que nos quedan después de finalizar este proyecto. También proponemos las ampliaciones que aportarían más valor a la solución.

Terminamos con el pliego de condiciones y presupuesto donde presentamos los requisitos de equipamiento y dinero necesarios para hacer realidad este proyecto.

2. Estudio de herramientas utilizadas

En este capítulo presentamos las herramientas usadas para llevar a cabo este PFC. Hacemos un análisis y justificamos su uso en los casos necesarios.

2.1. Sistema operativo Android

Lo primero que diferencia a Android de otros sistemas operativos para dispositivos móviles, es el gran número de dispositivos en los que se puede instalar. Tiene cabida en dispositivos (no sólo *smartphones*) de gama alta, pero también en los de la gama más baja. El sistema proporciona por sí mismo un conjunto de programas que garantiza la plena funcionalidad. Pero se acompaña de diversas *tiendas* de aplicaciones en las el usuario encuentra un gran número de ellas.

También ofrece al usuario una serie de servicios de conectividad que le permiten navegar, manejar y programar aplicaciones de forma sencilla.

2.1.1. Características generales

Android [1] dispone de una máquina virtual de código Java con un conjunto de bibliotecas de bajo y medio nivel que garantizan el acceso al dispositivo. El usuario encuentra además en la capa más altas el conjunto de aplicaciones destinadas al manejo del dispositivo.

Android tiene un una versión modificada de Linux en su núcleo (*kernel*), ello implica que este es *software libre*. Además, la alianza *Open Handset Alliance* [1], un consorcio más de 50 compañías de hardware, software y telecomunicación comprometidas con la promoción de estándares abiertos para dispositivos móviles, apoya *Android*. Las compañías que la integran distribuyen dispositivos basados en esta plataforma con el objetivo final de estandarizar, mejorar y abaratar el uso y desarrollo de aplicaciones para móviles. Y parece ser, que lo están consiguiendo.

Para posibilitar el desarrollo en este sistema, Google creó un *Software Development Kit (SDK)* [1], el cual provee algunas herramientas y *Application Programming Interfaces (APIs)*. Su plugin nos permite desarrollar aplicaciones Android en el *Entorno de Desarrollo Integrado (IDE)* Eclipse [1]. Otra alternativa es el uso de la *Native Development Kit (NDK)* de Google para hacer el desarrollo en C. Con el *NDK* [1] se pueden crear aplicaciones más rápidas ya que se desarrolla directamente utilizando bibliotecas nativas y no a través de la *Máquina Virtual Dalvik (DVM)* [1].

Las principales características de *Android* son:

- Posee un *framework* de desarrollo de aplicaciones que permite la reutilización y el reemplazo de componentes, típico en los lenguajes de programación orientados a objeto como *Java*.
- *Navegador web integrado* basado en el motor de código abierto *WebKit* [2], aunque tenemos la posibilidad de elegir entre otros buenos navegadores.
- Maneja *gráficos optimizados*, con una biblioteca abierta de gráficos 2D y 3D basados en la especificación estándar *OpenGL ES 1.0 (Open Graphics Library)* [3], que define una *API* multilenguaje y multiplataforma para escribir aplicaciones que producen dichos gráficos.
- Coopera con un sistema gestor de base de datos denominado *SQLite* [1], para almacenamiento de datos estructurados.
- Da soporte para medios con formatos comunes de audio, vídeo e imágenes planas (*MPEG4, H.264, MP3, OGG, AAC, AMR, JPG, PNG, GIF*).
- Soporta todas las tecnologías de comunicaciones existentes en los teléfonos móviles actuales.
- Posee un ambiente rico de desarrollo incluyendo un emulador de dispositivo, herramientas para depurar, perfiles de memoria y rendimiento, y un *plugin* para Eclipse. Así como también su propio IDE, aun en fase beta, llamado *Android Studio* [4].
- El usuario tiene a su disposición diferentes tiendas de aplicaciones, obras y servicios. El más popular en España es *Google Play Store*.
- Hereda el sistema de privilegios de Linux. Para el cual cada aplicación es un usuario con acceso y posibilidades limitadas. Esto mejora la seguridad del sistema.

A continuación se enumeran las ventajas que se encuentran a *Android*:

- La seguridad al elegir aplicaciones a instalar basándonos en las opiniones, valoraciones y advertencias de la comunidad.
- La curva de aprendizaje es rápida debido a la excelente documentación disponible en la red.
- Es software de código abierto. Cualquier programador puede realizar modificaciones a las partes internas de un programa, a diferencia de productos cerrados que distribuyen el código binario bloqueado. Lo que supone una ventaja a la hora de adaptarlo a un hardware.
- La participación en cualquier desarrollo es fácil. Sus *APIs* permiten desarrollar cualquier tipo de aplicación, facilitando el trabajo y motivando a los desarrolladores.
- La filosofía de *Android* es que funcione sobre terminales potentes. Permite cambiar con cierta facilidad su sistema operativo, con lo que se tiene un control y acceso más fácil a sus configuraciones.

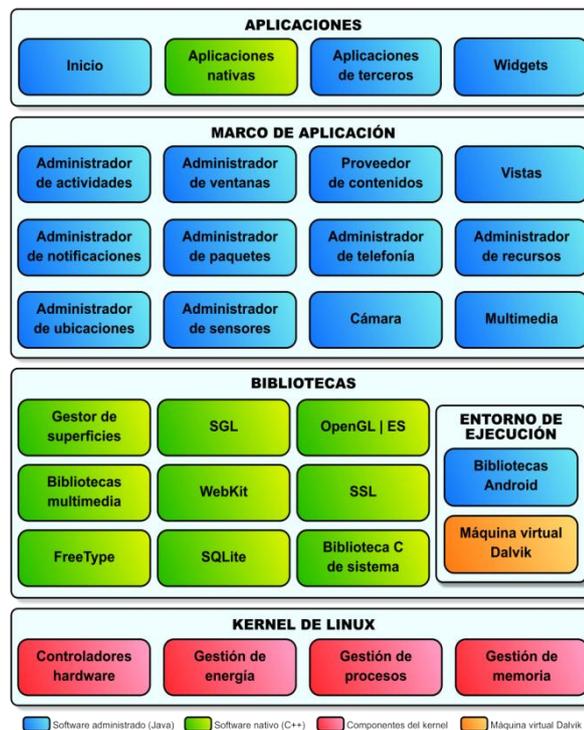


Figura 2.1. Arquitectura de un sistema operativo Android.

2.1.2. Arquitectura de Android

En la figura 2.1 se presentan los principales componentes que existen dentro de la plataforma *Android*.

Hacemos un recorrido por ella a continuación analizando sus elementos más importantes:

- *Aplicaciones*: es la parte más cercana al usuario final. *Android* se suministra con un conjunto de aplicaciones base para los clientes, como correo electrónico, programa de *SMS*, calendario, mapas, navegador, contactos, y muchas más que se pueden obtener desde el sitio *Google Play*.
- *Framework de aplicaciones*: los desarrolladores tienen acceso completo a la misma *API* del *framework* que se usa para las aplicaciones base. La arquitectura de *Android* está diseñada para simplificar la reutilización de componentes ya que cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de dichas capacidades (sujeto a reglas de seguridad del *framework*). Este mismo mecanismo permite sustituir componentes que hayan podido quedar obsoletos o en desuso, de forma fácil y sin tener que modificar código ajeno.
- *Bibliotecas*: incluye un conjunto de bibliotecas *C/C++* usadas por varios componentes del sistema *Android*. Estas bibliotecas se exponen a los desarrolladores a través del *framework* de aplicaciones de *Android*. Entre ellas cabe destacar: *System C* (implementación biblioteca C standard), bibliotecas de medios, bibliotecas de gráficos, 3D, *SQLite*, entre otras.
- *Bibliotecas Java*: incluye un conjunto de bibliotecas que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje *Java*.
- *Máquina virtual*: cada aplicación *Android* ejecuta su propio proceso de manera independiente, con su propia instancia de la máquina virtual *Dalvik*. *Dalvik* ha sido escrito de forma que un dispositivo puede ejecutar múltiples máquinas virtuales de forma eficiente. *Dalvik* ejecuta archivos en el formato *Dalvik*

Executable (.dex), el cual está optimizado para utilizar la memoria mínima. El código *Java* se convierte al formato que entiende la máquina virtual *Dalvik*.

- *Núcleo de Linux*: depende de *Linux* versión 2.6 para los servicios base del sistema, como seguridad, gestión de memoria, gestión de procesos, stack de red, y modelo de controladores. El kernel también actúa como una capa de abstracción entre el hardware y el software.

2.1.3. Estructura de una aplicación Android

La estructura de una aplicación Android está definida por la interacción de distintos componentes, haciendo énfasis en la agrupación débil de distintas piezas. La aplicación hace uso de las distintas APIs expuestas por Android, de forma que los componentes encargados de realizar cada tarea puedan ser manipulados o reemplazados sin problemas, asegurando la máxima flexibilidad.

Citamos a continuación los principales componentes de una app Android:

- *Activity*: representa cada una de las principales tareas que el usuario puede llevar a cabo en la aplicación. Típica (aunque no necesariamente) corresponde a una pantalla específica de la aplicación y, también normalmente, una activity es el punto de entrada (pantalla inicial) de la aplicación. Desde ella se invocan las vistas, específicas o *layouts*, para la aplicación.
- *IntentReceiver*: permite a la aplicación declarar ciertos procedimientos que responden a cambios en el estado del terminal, por ejemplo, llamada o email recibido, cambio en la localización...
- *Service*: una tarea que funciona en segundo plano y que puede y debe ejecutarse sin interacción con el usuario. Una aplicación puede mandar los mensajes necesarios a un determinado servicio activo.
- *ContentProvider*: establece un nivel que permite a las distintas aplicaciones compartir datos. Con independencia del almacenamiento local que utilicen para sus propósitos, las aplicaciones necesitan declarar *ContentProviders* para poner a disposición de otros procesos los datos que consideran necesarios.

Estos son algunos de las principales, pero no las únicas piezas de construcción de la aplicación.

2.1.4. Jerarquía visual de Android

Como se ha comentado anteriormente, una aplicación *Android* está definida por la interacción de distintos componentes que se agrupan entre sí. A continuación, se comenta de manera técnica la jerarquía que siguen los componentes de *Android* en el desarrollo de una aplicación visual.

Todas las aplicaciones en *Android* poseen un archivo *AndroidManifest.xml*, que es el archivo de control que indica al sistema qué es lo que debe hacer con los componentes de más alto nivel que se han creado (específicamente *Activity*, *Service*, *IntentReceiver* y *ContentProvider* descritos con anterioridad). Debido a esto, este archivo es el encargado de encajar todas las partes de una aplicación indicando a qué *Intents* responde cada *Activity*. Una vez que se han seleccionado los componentes que usa la aplicación se deben declarar en este archivo. La principal clase de *Android* es *Activity*, un objeto de la clase *android.app.Activity*. Una actividad realiza una gran variedad de cosas, pero por ella misma no presenta nada en la pantalla. Para conseguir que aparezca algo en la pantalla es necesario diseñar la *Interfaz de Usuario (UI)*, con *Views* y *Viewgroups*, que son las clases que se usan para crear la interfaz entre el usuario y la plataforma *Android*.

Vista

Una Vista (*View*) es un objeto cuya clase es *android.view.View*. Es una estructura de datos cuyas propiedades contienen los datos de la capa y la información específica del área rectangular de la pantalla. Una *View* tiene: *layout*, *drawing*, *focus change*, *scrolling*... La clase *View* sirve como clase base para los *widgets*, que son unas subclases ya implementadas que dibujan los elementos en la pantalla. Los *widgets* se pueden usar para construir tu interfaz más rápidamente. La lista de *widgets* útiles incluyen: *Text*, *EditText*, *InputMethod*, *MovementMethod*, *Button*, *RadioButton*, *CheckBox*, y *ScrollView*.

Viewgroups

Un *Viewgroup* es un objeto de la clase *android.view.Viewgroup*, y como su propio nombre indica, es un objeto especial cuya función es contener y controlar la lista de *Views* y de otros *Viewgroups*. Los *Viewgroups* permiten añadir estructuras a la interfaz y acumular complejos elementos en la pantalla que son abarcados por una sola entidad. Esta clase es útil como clase base para los *Layouts*, que son subclases implementadas que proveen los tipos más comunes de los *Layouts* de pantalla. Los *Layouts* proporcionan una manera de construir una estructura para una lista de *Views*.

Árbol estructurado de la interfaz de usuario

En la plataforma *Android* se define una *Activity* del interfaz usando un árbol de nodos *View* y *Viewgroups*. El árbol puede ser tan simple o complejo como se necesite, y se puede desarrollar usando los *Widgets* y *Layouts* que *Android* proporciona o creando tus propias *Views*. Para añadir el árbol a la pantalla, el *Activity* llama al método *setContentView()* y pasa una referencia al objeto nodo principal. Una vez que el sistema *Android* ha referenciado el objeto nodo principal ya puede trabajar directamente con el nodo para medir y dibujar el árbol. El nodo principal entonces pide que sus nodos hijos se dibujen a sí mismos, a partir de ese momento cada nodo *Viewgroup* del árbol es responsable de pintar sus hijos directos.

Cada *Viewgroup* es el responsable de tomar medidas sobre el espacio que tienen, preparando a sus hijos y llamando a *Draw()* por cada hijo que se muestra a sí mismo. El hijo hace una petición sobre el tamaño y la localización del padre, pero el objeto padre toma la última decisión sobre el tamaño que cada hijo puede tener.

LayoutParams

Sirve para indicar al padre su posición y su tamaño. Todos los *Viewgroup* usan como clase anidada una extensión de *ViewGroup.LayoutParams*. Esta subclase contiene los tipos de propiedades que definen la posición y el tamaño de un hijo. En la Figura 2.3 se muestra su funcionamiento. Hay que reseñar que cada subclase *LayoutParams* tiene su propia sintaxis para cambiar los valores. Cada elemento hijo debe definir unos

LayoutParams que sean apropiados para su padre, aunque se pueden definir diferentes *LayoutParams* para sus hijos.

Todos los *Viewgroups* incluyen anchura y altura. Muchos también incluyen márgenes y bordes. Se puede especificar exactamente la altura y la anchura, aunque hay muchas veces que esto no sea necesario. Más corriente es indicar a la *View* que tenga las dimensiones del tamaño de su contenedor, o que llegue a ser tan grande como el contenedor le permita.

2.2. SDK de Android

Este entorno de desarrollo incluye una variedad de herramientas especialmente diseñadas para ayudar en el desarrollo de aplicaciones móviles sobre la plataforma *Android*. Las herramientas más importantes son: el *Android Virtual Device Manager* y las *Android Development Tools (ADT)* plugin para *Eclipse* [5]. Además de estas dos herramientas, el *SDK* también incluye un surtido de herramientas para depuración, empaquetado e instalación de aplicaciones en el dispositivo o en el emulador.

A continuación se explican algunas herramientas proporcionadas por este entorno de desarrollo usadas en la realización de este PFC.

Dalvik Debug Monitor Service

La *Dalvik Debug Monitor Service (DDMS)* [6] es una herramienta integrada con la DVM, y permite administrar los procesos que corren en una instancia de dispositivo además de asistir en la depuración de ellos. Esta herramienta se puede usar para terminar la ejecución de un proceso, seleccionar un determinado proceso para depurar, generar reportes a partir de información obtenida, realizar capturas de pantalla del dispositivo y mucho más.

Plugin ADT para Eclipse

El plugin *ADT* agrega poderosas funciones al entorno de *Eclipse* haciendo que la creación y depuración de las aplicaciones *Android* sea fácil y rápida. Estas funciones son accesibles desde los diferentes menús que proporciona *Eclipse*. A continuación se mencionan algunas de las características de este plugin:

- Provee acceso a otras herramientas de desarrollo de *Android* desde el entorno de *Eclipse*. Por ejemplo, *ADT* permite el acceso a muchas de las capacidades de la herramienta *DDMS*, tales como tomar fotografías de la pantalla, administrar el redireccionamiento de puertos, fijar puntos de ruptura y examinar la información y procesos directamente dentro de *Eclipse*.
- Provee un asistente para la creación de proyectos *Android*, el cual ayuda a crear rápidamente todos los directorios y archivos necesarios para crear una nueva aplicación *Android*.
- Automatiza y simplifica el proceso de construcción de una aplicación.
- Provee un editor de código *Android* que ayuda a escribir *XML* válido para el archivo *AndroidManifest.xml*.

Emulador Android

El *Emulador Android* (AVDM, del inglés *Android Virtual Device Manager*) es un dispositivo virtual que se ejecuta en el ordenador. El emulador tiene como finalidad ayudar a diseñar y depurar las aplicaciones en un ambiente similar al que existe en un dispositivo real. Existen distintas versiones del emulador tanto para *Windows* como para *Mac OS* o *Linux*. Para abrir una instancia de este emulador se ha utilizado las herramientas que proporciona *ADT* para *Eclipse*, aunque no es la única forma de hacerlo. También se puede abrir una instancia mediante consola de comandos.

Eclipse

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los *IDEs* pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Eclipse es una plataforma de desarrollo de código abierto basada en Java. Por sí misma, es simplemente un marco de trabajo y un conjunto de servicios para la construcción del entorno de desarrollo de los componentes de entrada. Afortunadamente, Eclipse tiene un conjunto de complementos, incluidas las *Herramientas de Desarrollo de Java (JDT)* [7].

Eclipse dispone de multitud de extensiones o *plugins* que nos permiten por ejemplo desarrollar aplicaciones Android (*Android ADT*).

2.3. Extensible Markup Language

El *eXtensible Markup Language (XML)* [8], es un lenguaje de marcas desarrollado por el *World Wide Web Consortium (W3C)* utilizado para almacenar datos en forma legible. Se ha convertido en un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable. Usamos este formato para almacenar los datos de cada parada.

El XML busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible. XML permite expresar de forma abstracta información estructura proporcionando mucha flexibilidad. Las ventajas de XML son:

- *Extensible*: después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.
- *El analizador es un componente estándar*: no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de

cualquiera de los analizadores disponibles. De esta manera se evitan errores y se acelera el desarrollo de aplicaciones.

Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones. Podemos comunicar aplicaciones de distintas plataformas, sin que importe el origen de los datos, es decir, podríamos tener una aplicación en Linux con una base de datos Postgres y comunicarla con otra aplicación en Windows y Base de Datos MS-SQL Server. Transformamos datos en información, pues se le añade un significado concreto y los asociamos a un contexto, con lo cual tenemos flexibilidad para estructurar documentos.

Una etiqueta consiste en una marca hecha en el documento, que señala una porción de éste como un elemento. Un pedazo de información con un sentido claro y definido. Las etiquetas tienen la forma `<nombre>`, donde *nombres* el nombre del elemento que se está señalando. En la figura 2.1 se muestra un ejemplo para entender la estructura de un documento XML.

```
<?xml version="1.0"?>
<!DOCTYPE MENSAJE SYSTEM "mensaje.dtd">

<mensaje>
  <remite>
    <nombre>Alfredo Reino</nombre>
    <email>alf@ibium.com</email>
  </remite>
  <destinatario>
    <nombre>Bill Clinton</nombre>
    <email>president@whitehouse.gov</email>
  </destinatario>
  <asunto>Hola Bill</asunto>
  <texto>
    <parrafo>¿Hola qué tal? Hace
      <enfasis>mucho</enfasis>
      que no escribes.
      A ver si llamas y quedamos para tomar algo
    </parrafo>
  </texto>
</mensaje>
```

Figura 2.1: Ejemplo de XML.

2.4. Simple Object Access Protocol

El protocolo simple de acceso a objetos (*SOAP*, del inglés *Simple Object Access Protocol*) [9], es un estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Clama que las aplicaciones deben ser independientes del lenguaje de desarrollo, por lo que las aplicaciones cliente y servidor pueden estar escritas con HTML, DHTML, Java, Visual Basic u otras herramientas y lenguajes disponibles. Lo importante es tener alguna implementación de SOAP (dependiendo de la herramienta de desarrollo elegida) y enlazar sus bibliotecas con la aplicación. Aunque esto no es estrictamente necesario, es preferible trabajar usando dichas bibliotecas, con el fin de no reescribir un código ya probado. SOAP es un marco extensible y descentralizado que permite trabajar sobre múltiples arquitecturas de protocolos de redes de comunicación. Los procedimientos de llamadas remotas pueden ser modelados en la forma de varios mensajes SOAP interactuando entre sí. Estos mensajes constan de 3 secciones: envelope, header y body (véase organización en figura 2.2):

- *envelope*: es el elemento raíz del mensaje para describir su contenido y la forma de procesarlo.
- *header*: es la información de identificación del contenido. Un grupo de reglas de codificación para expresar las instancias de tipos de datos definidos por la aplicación.
- *body*: es el contenido del mensaje. Una convención para representar las llamadas y las respuestas a procedimientos remotos.

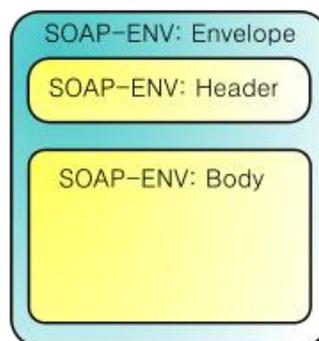


Figura 2.2: Estructura de una comunicación SOAP.

Modelo de intercambio de mensajes

SOAP usa transmisiones unidireccionales desde un emisor a un receptor. Pero se suelen combinar mensajes para implementar patrones, como petición/respuesta. Y así crear combinaciones más complejas. Un software que implante SOAP debe procesar un mensaje siguiendo el orden siguiente de tareas:

- Analizar el mensaje llevando a cabo la identificación de las del mensaje SOAP de las que hemos hablado anteriormente.
- Identificar las partes obligatorias, de no obtenerlas, descartamos el mensaje.
- Si no es el destinatario último del mensaje recibido, desencapsularlo eliminando las partes identificadas en el punto 1, antes de remitirlo.

2.5. Otras herramientas

En este apartado presentamos otras herramientas que hemos utilizado para llevar a cabo el PFC: XAMPP [10], Apache [11], Axis2 [12], Microsoft © Excel [13] y el servicio de Google Maps [14].

XAMPP

Es una distribución de Apache completa que contiene MySQL, PHP y Perl. Una herramienta que útil para desplegar los servidores usados en el PFC. Dispone de una consola de control (Figura 2.4) para activar o desactivar los servidores, facilitando mucho la tarea con ellos. Para este proyecto utilizamos *Tomcat* [15] y Apache [11].

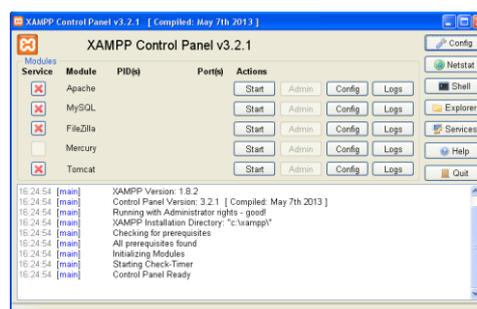


Figura 2.4: Consola de control de XAMPP.

Apache

Servidor web *HTTP* de código abierto, para plataformas *Unix (BSD, GNU/Linux...)*, *Microsoft Windows, Macintosh* y otras, que implementa el protocolo *HTTP/1.12* y la noción de sitio virtual. Presenta entre otras características: ser altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan *PHP* como módulo de *Apache*.

Tomcat

Utiliza los conceptos: *applets* [16] y los *servlets* [17]. Un *applet* es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo en un navegador web. Debe ejecutarse en un contenedor, que le proporciona un programa anfitrión, mediante un plugin o en aplicaciones como teléfonos móviles que soportan el modelo de programación por *applets*. Ofrece información gráfica y a veces interactúa con el usuario, típicamente carece de sesión y tiene privilegios de seguridad restringidos. Normalmente lleva al cabo una función muy específica que carece de uso independiente. Ejemplos comunes de *applets* son las Java applets y las animaciones Flash. Un Java applet [18] es un código Java que carece de un método *main*, por eso se utiliza principalmente para el trabajo de páginas web, ya que es un pequeño programa que es utilizado en una página HTML y representado por una pequeña pantalla gráfica dentro de ésta. El *servlet* es una clase Java, utilizada para ampliar las capacidades de un servidor. Aunque pueden responder a cualquier tipo de solicitudes, éstos son utilizados comúnmente para extender las aplicaciones alojadas por servidores web, de tal manera que pueden ser vistos como applets, vistos en la sección anterior, de Java que se ejecutan en servidores en vez de navegadores web. El uso más común de los

servlets es generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.

Tomcat es un contenedor de servlets desarrollado bajo el proyecto *Jakarta* [19] en la *Apache Software Foundation*. Se usa como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad funcionando en cualquier sistema operativo que disponga de la *Máquina Virtual Java (JVM)*.

En el caso concreto de este proyecto, Tomcat ejecuta nuestro servicio web, con el motor Axis2 [12].

Axis 2

Apache Axis2 es un motor nuclear para servicios web derivada de SOAP *Apache Axis* [12]. Existen implementaciones de Axis2 en Java y en C. No solo provee la capacidad de agregar servicios web a las aplicaciones web, sino que además puede funcionar como servidor autónomo. Su uso en el proyecto se justifica por la utilización en el mismo del protocolo SOAP, anteriormente descrito y porque ofrece las siguientes características:

- *Velocidad*: usa su propio modelo de objetos y un análisis sintáctico basado en *Streaming API para XML (StAX)* para lograr una velocidad de proceso significativamente mayor que la de versiones anteriores de Apache Axis.
- *Uso reducido de memoria*.
- *Despliegue instantáneo*: equipado con la capacidad de desplegar servicios web y *handlers* con el sistema en pleno funcionamiento sin tener que detener la ejecución del servidor. Basta con copiar simplemente los archivos requeridos de servicios web al directorio de servicios en el repositorio, y el modelo de despliegue automáticamente despliega el servicio y lo pondría a disposición para su uso.
- *Servicios web asincrónicos*: soporta servicios web asincrónicos y la invocación asincrónica de servicios web por medio de clientes y transportes no bloqueantes.

- *Flexibilidad*: el desarrollador tiene completa libertad para insertar extensiones al motor para el procesamiento a la medida de encabezamientos (*headers*), administración del sistema, o cualquier otro aspecto imaginable.
- *Estabilidad*: define un conjunto de interfaces publicados que cambian con relativa lentitud, comparados con el resto de Axis.
- *Framework de transporte*: tiene una abstracción limpia y simple para la integración y el uso de diversos transportes (esto es, *senders* y *listeners* para SOAP por vía de diversos protocolos tales como SMTP, FTP, middleware orientada a mensajes...), siendo el núcleo del motor completamente independiente de los mecanismos de transporte.
- *Soporte de Web Services Description Language (WSDL) versiones 1.1 y 2.0*: facilita la construcción de *stubs* para el acceso a servicios remotos y la exportación de descripciones legibles.
- *Composición y extensibilidad*: empleando módulos y fases que soportan nuevas especificaciones WSDL de manera más simple y limpia. Sin embargo no permiten su despliegue instantáneo, ya que alteran el comportamiento global del sistema.

Microsoft © Excel y Visual Basic

La hoja de cálculo de la suite Microsoft Office ©, elabora tablas y formatos que incluyen cálculos matemáticos mediante. Permite utilizar *funciones* prefabricadas como: *Suma()*, *Promedio()*, *BuscarV()*... Y es útil para gestionar *Listas* o *Bases de Datos*; es decir agrupar, ordenar y filtrar la información.

Aunque Visual Basic es de propósito general, también provee facilidades para el desarrollo de aplicaciones de bases de datos usando *Data Access Objects*, *Remote Data Objects* o *ActiveX Data Objects* mediante un *IDE*.



Figura 2.5: Activación de la API.

Servicio de Google Maps

En este proyecto la utilizamos para visualizar los mapas, que nos proporciona acceso a toda la funcionalidad *Google Maps*. El API de *Google Maps Engine* proporciona acceso mediante programación a los activos contenidos en Google Maps proyectos Engine. Proporciona acceso de sólo lectura a todos los activos. Vemos la presentación del desarrollador en la figura 2.5.

Esta herramienta nos proporciona unas bibliotecas con clases que al instanciar nos muestran un mapa que podemos programar, con una perfecta integración en nuestro código Java.

Para ello, usando nuestra cuenta de usuario Google pedimos un código con el que podemos obtener el acceso a la API en modalidad de desarrollador. Vemos una captura de la activación de la API en la figura 2.6.

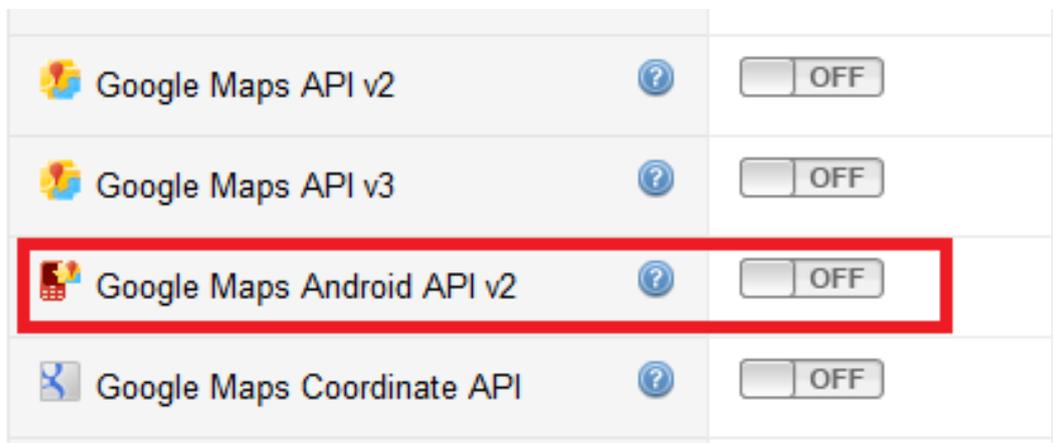


Figura 2.6: Activación de la API.

3. Análisis previo y funcional

En este capítulo se realiza un estudio previo acerca de la arquitectura del sistema y de las distintas alternativas posibles para el desarrollo de las aplicaciones. Además, se desglosa el proyecto en diferentes módulos, en los que se explica de manera general la funcionalidad de los mismos y su relación entre ellos.

3.1. Análisis Previo

Este proyecto tiene como objetivo la creación de una solución para ofrecer una información útil y veraz sobre el tiempo de llegada de las guaguas en cualquier lugar. Todo ello con el mínimo de pasos posibles y gastando cuanto menos recursos mejor.

Los usuarios potenciales de esta solución son de niveles muy variados, a cualquier persona podría interesarle. Por ello la aplicación debe ir al grano, mostrar solamente la información que es necesaria y ahorrando recursos.

Dada la gran variedad de dispositivos que podrían ejecutar la aplicación móvil, encontramos máquinas cuyas prestaciones están limitadas en algunos casos.

Se ha de atender las necesidades de las personas que visitan la ciudad por primera vez. Pero también las de la gente que tiene su rutina y repite la misma parada una y otra vez. Para estas debe suponer una mejora en su vida diaria. Y una ayuda en la optimización de su tiempo.

Para estimar debidamente la llegada de la próxima guagua, dadas las condiciones de la ciudad, no nos podemos ceñir al horario de la compañía. Dado que este está sujeto a imprevistos y condiciones puntuales del tráfico... Necesitamos una valoración en tiempo real a cada pocos minutos. El sistema debe ser algo parecido al de la figura 3.1.

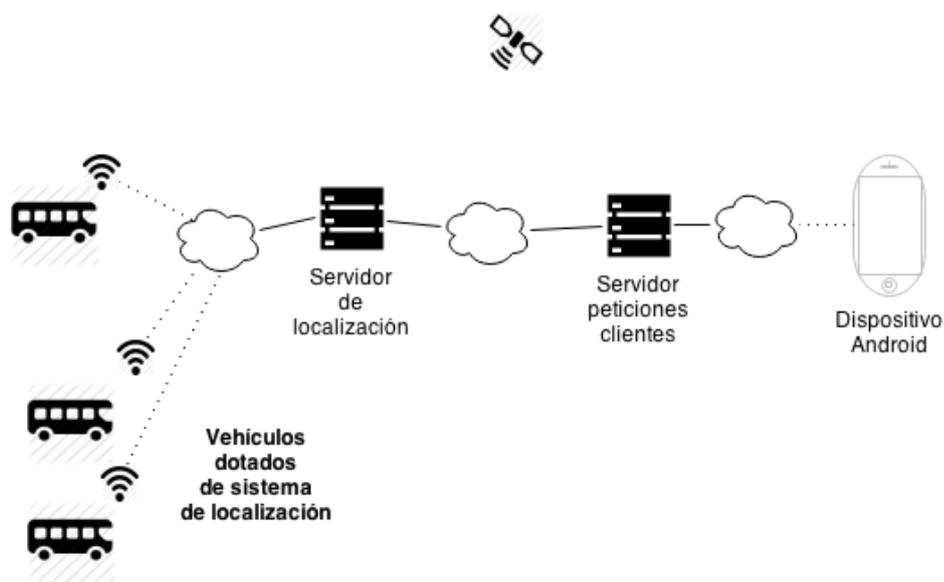


Figura 3.1: Topología inicial de la solución.

Hemos pensado a priori en una solución como la que mostramos en la figura 3.2. Dividimos el sistema en tres bloques:

- *Rojo*: Lleva a cabo el muestreo de la localización de los vehículos en ruta. Por ello se equipa a los transportes con dispositivos GPS, los cuales envían datos constantemente a un servidor. Con ello obtenemos una base de datos actualizadas con el punto exacto donde se encuentran nuestras guaguas.
- *Verde*: gestiona la información anterior. Y constaría de un servidor privado, encargado de recibir las actualizaciones que constantemente le llegan de los GPS y por otro lado de estimar, basado en estadísticas, los tiempos de llegada. Este servidor sería atacado por otro. Y este último servidor atendería las peticiones de información de los dispositivos de los usuarios. Con una IP pública a la que estos últimos tendrían acceso.
- *Azul*: la aplicación móvil.

Al atacar este proyecto encontramos el actual sistema que ha implementado la compañía de guaguas que ya posee una infraestructura de localización funcionando y realiza las estimaciones de tiempo. De conversaciones con la compañía sabemos que tienen capacidad para estimar la llegada de todas las guaguas. Por tanto, nuestro proyecto se centraría en la mejora del sistema vigente. Haciendo accesible de todas partes la información que vemos en los paneles informativos.

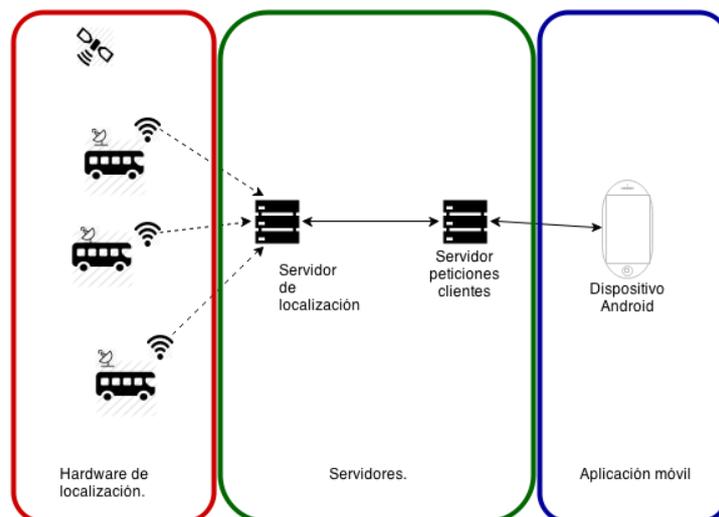


Figura 3.2: Planteamiento teórico de la solución.

La idea es pues, atacando a un servidor, presentar la información a los usuarios. La primera pregunta con la que nos encontramos es si elaborar una página web adaptada a dispositivos móviles o una aplicación. Aquí hemos valorado la simplicidad, la rapidez y la integración con el sistema Android para decantarnos por la aplicación móvil. En primer lugar, el acceso a la información con un programa cliente o una aplicación es más rápido, pues ya cuenta con cierta información y no tiene que cargar una página web. Por otro lado, una aplicación móvil permite ahorrar recursos en el servidor ya que el proceso lo realiza el dispositivo del usuario y éste solo se encargaría de facilitar ciertos datos. Una aplicación permite lanzar notificaciones, accesos directos y guardar cierta información con los gustos del usuario con la que brindar un servicio más práctico.

Decidida la creación de la aplicación, visitamos la sede de la compañía para obtener los datos y la manera de atacar a su servidor. Pero encontramos alguna dificultad administrativa y cierta falta de fluidez en las conversaciones. Por lo que decidimos finalmente desarrollar la solución programando unos servicios propios que simule tener los datos y atacarlos con la aplicación móvil. Lo cual, dado el caso es fácilmente reversible con unos pequeños cambios en el código de la aplicación del móvil.

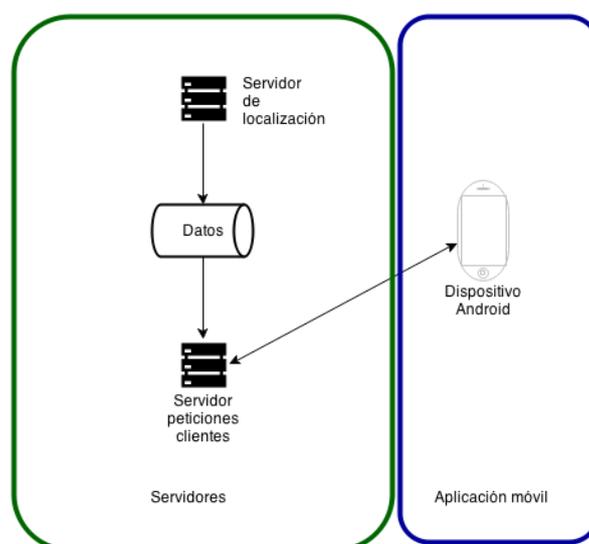


Figura3.3: Planteamiento teórico del proyecto.

Por tanto ahora nos encontramos ante una tarea adicional, la creación de dos servidores, conectados a un mismo almacenaje de datos. El primer servidor actualiza los datos y el segundo los consulta a petición y los facilita a los usuarios. En la figura 3.3 se muestra un diagrama que representa dicha estructura.

En el siguiente apartado dividimos y analizamos los bloques del proyecto.

3.1.1. Arquitectura final del proyecto

Tras analizar los requisitos que el proyecto necesita, resumimos en la figura 3.4 los tres bloques principales de que consta la aplicación que implementamos. Un primer bloque se encarga de simular la obtención de datos. Dicha información es consultada por el siguiente bloque, un servicio web que atiende peticiones. Y por último el tercer bloque, la aplicación móvil. En los siguientes apartados buscamos una solución para cada bloque funcional.

3.1.2. Otras aplicaciones similares

Aunque en el momento de aceptar el proyecto no existían. En el mercado ya podemos encontrar otros proyectos parecidos al nuestro. En Google Play Store hemos encontrado, varias, aquí van unos ejemplos:

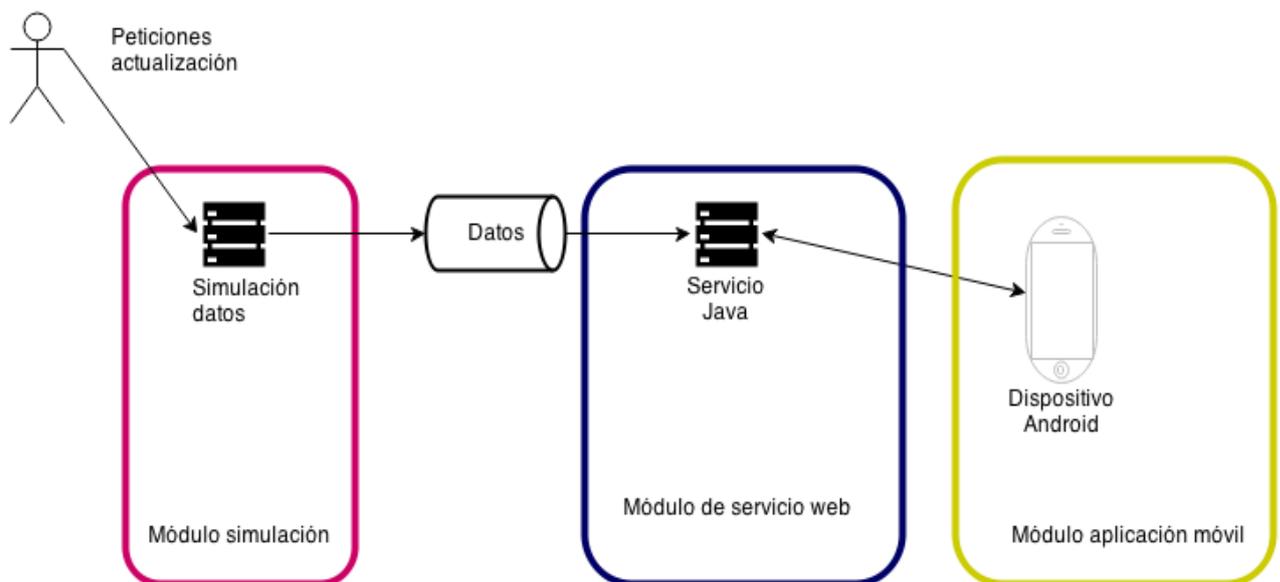


Figura 3.4: Arquitectura general del proyecto.

- *GuaguApp (guaguas Tenerife)*: de uso cotidiano con información a tiempo real de los trayectos y pasos por paradas de las guaguas. Funciona con los servidores de la compañía *Titsa*. Ofrece una vista general de Tenerife y sus distintos nodos y paradas y es capaz de calcular rutas y tiempos según horarios. Incorpora muchas funcionalidades extra a parte de la estimación de los tiempos de llegada. Llama la atención la incorporación de un alimentador de noticias de la propia compañía de guaguas donde se avisa de sucesos e incidencias en el servicio y que la app transforma en una notificación. Es fácilmente extrapolable a otras zonas y regiones y se encuentra en este momento intentando expandirse.
- *Moovit Bus, Metro, Tren Info*: funciona en varias ciudades españolas pero también en Latinoamérica. Como elemento diferenciador, incorpora alarmas. Y en ellas empezamos a ver elementos de red social, ya que puede asistir en las quedadas compartiendo las rutas. Sus datos están basados en las capturas que hacen los terminales de los propios usuarios. Mejora por tanto en escenarios donde su uso sea mayor.

3.2. Análisis Funcional

Nos disponemos a razonar la solución planteada, y a tomar las decisiones sobre las herramientas y el diseño que queremos alcanzar. Partiendo de los bloques en que dividimos el proyecto anteriormente ahora describimos módulos funcionales y su funcionalidad.

3.2.1. Módulo de simulación de datos

Como vimos en el anterior apartado, el primer bloque funcional se encarga de la simulación de datos. En este análisis asignamos un módulo a dicho bloque funcional. Al simular la obtención de los minutos restantes, este dato debe cambiar continuamente. Para controlar la forma y la velocidad a la que cambian, nos decantamos por desplegar un pequeño código en un servidor para lanzarle peticiones de actualización.

Este módulo tiene dos funcionalidades básicas:

- *Inicialización*: crea los archivos de texto plano a la vez que escribe en ellos un dato aleatorio. Tal y como muestra la figura 3.5, estos datos, son puestos a disposición del módulo siguiente.
- *Actualización*: sería la funcionalidad a repetir indefinidamente, con el intervalo deseado. Su función consiste en restar un minuto al tiempo de llegada de cada vehículo, y en caso de encontrarse algún cero, escribir un nuevo dato.

El primero, de inicialización, se ejecuta una vez y crea los archivos de texto plano a la vez que escriben en ellos un dato aleatorio.

El segundo código se encarga de restar a cada dato de cada archivo, el entero 1. Así simulamos el paso del tiempo. A la velocidad que queramos.

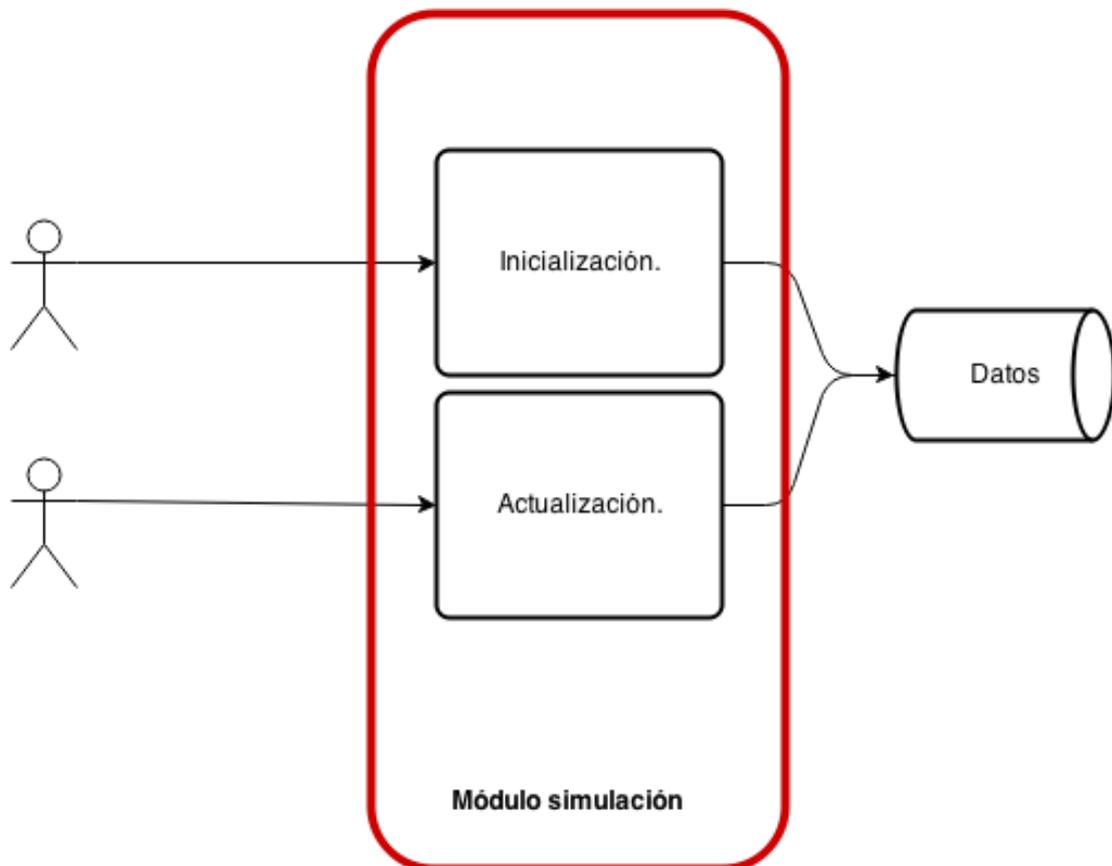


Figura 3.5: Funciones del simulador.

3.2.2. Módulo Aplicación móvil Android

La parte principal del proyecto consiste en el diseño de esta aplicación. Es a través de ella por donde el usuario interactúa con el sistema. Debe facilitar el acceso a la información de forma sencilla y para ello hemos establecido algunas condiciones.

El usuario no debe escribir o manejarse por un menú buscando su parada. Esta aplicación puede ser usada mientras caminamos, quizás con prisas y con una sola mano. La aplicación debe darnos lo que queremos con unos pocos clics, cuanto menos mejor.

La búsqueda debe ser intuitiva, por esto hemos pensado en un mapa, que puede centrarse en la posición actual en principio o desplazarse a la zona que queremos ver.

La aplicación no debe consumir muchos recursos ni ser visualmente pesada, cosa que tomamos en consideración más adelante.

Funcionalidades

El diseño de la aplicación gira en torno tres *opciones* que interaccionaran con el usuario ofreciéndole tres funcionalidades.

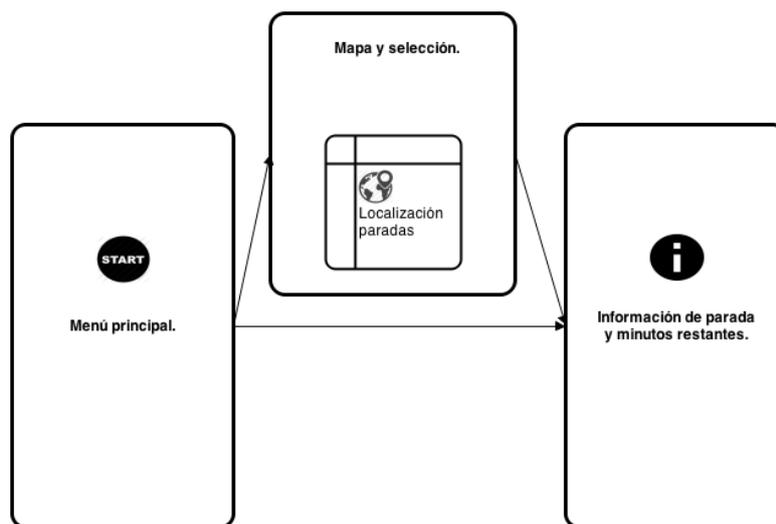


Figura 3.6: Esquema funcional de la aplicación Android.

Nuestra aplicación móvil llega a un usuario que quiere conocer cuando llega su guagua, ahora. No cualquier transporte, ni el horario o frecuencia de la línea por la que espera. Solo le interesa el siguiente vehículo que pasaría por una parada en concreto.

Probablemente la parada está relativamente cerca de su lugar de estudios, trabajo o residencia. Es probable que no sepa el nombre de la calle donde la parada se encuentra, por supuesto no sabe el número de parada si le asignamos una numeración a las marquesinas. El usuario sabe dónde está y a donde quiere ir. Nada más.

Hemos de enfocar la aplicación por ahí si queremos entrar en la rutina de las personas. Y ser de utilidad.

Por tanto la aplicación móvil permite elegir la parada objetivo entre las más cercanas. Aunque también ofrece la opción de establecer nuestro radio de visión en otro punto con simplemente clicar en otro lugar. Aunque vemos esto más detenidamente con posterioridad.

La figura 3.6 muestra los tres bloques funcionales en que dividimos nuestra aplicación móvil:

- Menú principal.
- Mapa y selección de parada.
- Panel de información.

Un menú principal muestra al usuario las opciones disponibles, todas ellas con el objeto de buscar una parada o visualizar la información de alguna preestablecida como favorita.

El siguiente módulo, un mapa, debe mostrar las paradas dentro del área de interés del usuario de una forma rápida. Mostrando solamente las ubicadas en un radio concreto para no cargar demasiado el terminal. La posición de las paradas la ubicamos dentro del propio terminal. Dado que es una información relativamente fija y ahorramos ancho de banda y conexión.

El usuario debe navegar en el mapa hasta encontrar la parada que busca. Momento en el cual pasamos al siguiente bloque.

El siguiente módulo es el que presenta la información relevante. Y se comporta de la forma que muestra la figura 3.7, aunque analizamos esto con más detenimiento en el análisis orgánico.

Esta funcionalidad debe seguir activa aún cuando el usuario no esté atendiendo el móvil o se encuentre usando otra aplicación. El móvil avisa con un aviso acústico minutos antes de la llegada de la guagua.

Está pensado para que el usuario pueda permitirse acudir a la parada de guaguas en el momento en que el autobús va a llegar.

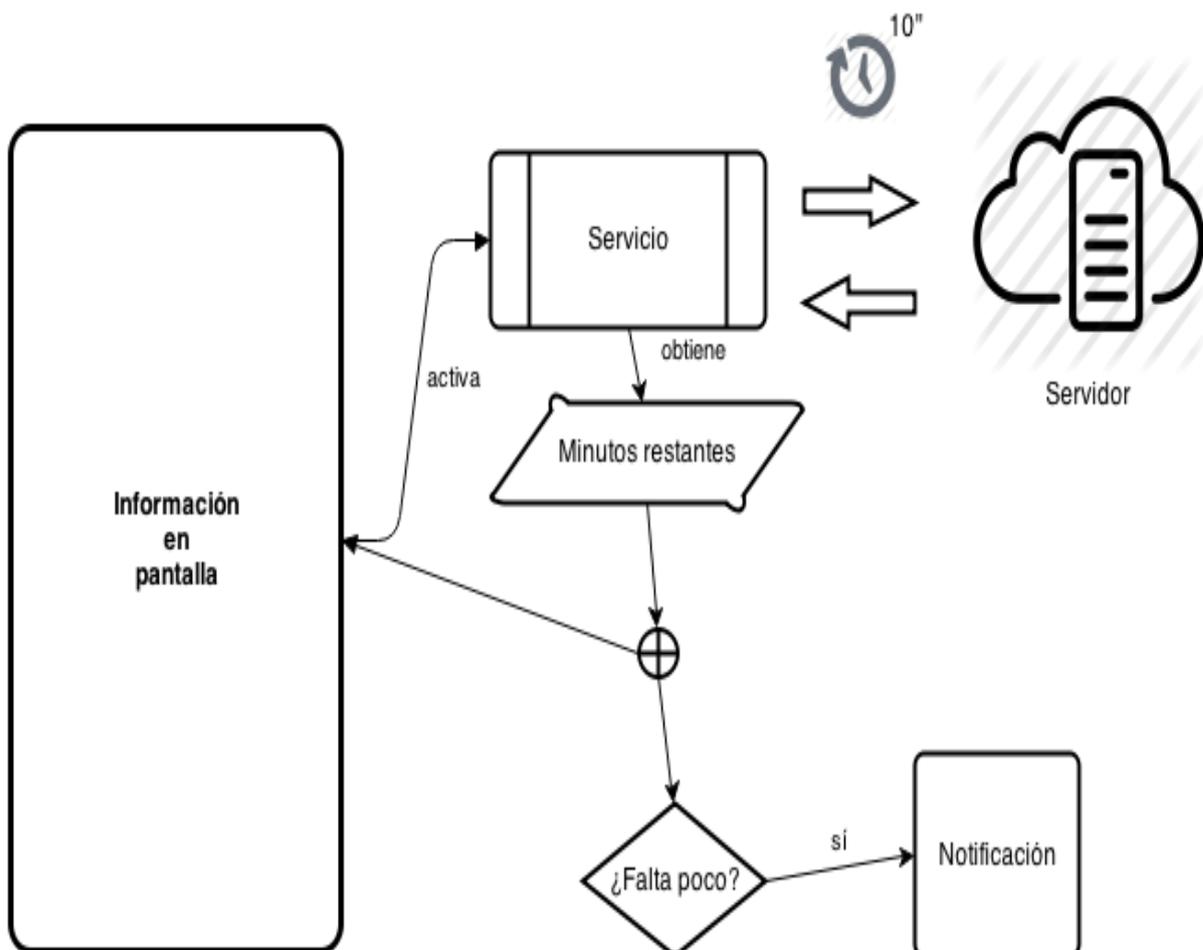


Figura 3.7: Funcionamiento del bloque informativo.

3.2.3. Módulo de servicio web

Como hemos mostrado, este módulo es el encargado de atender las peticiones que envían los terminales móviles. Su funcionalidad es sencilla, tal y como muestra la figura 3.8.

Interesa que sea rápido, sencillo y ejecute un código robusto.

El servicio debe tener los datos a mano, el refresco de los mismos no es parte de su funcionalidad. Aunque si lo es la consulta en base a las peticiones que reciba.

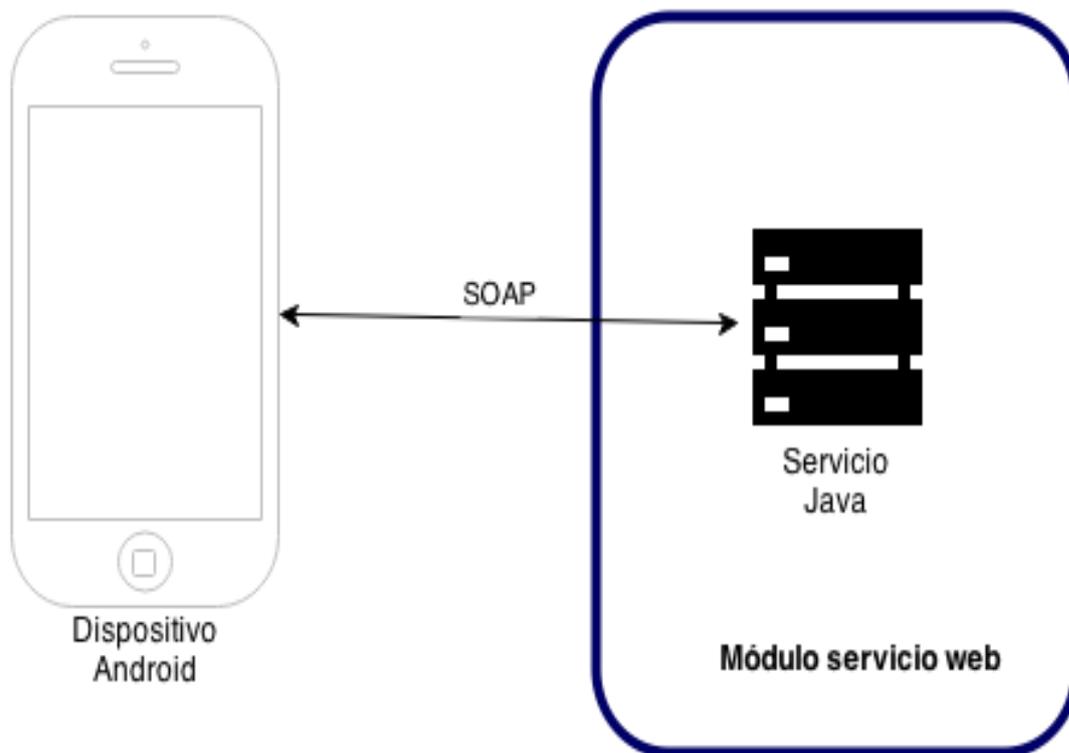


Figura 3.8: Esquema funcional del módulo de servicio web.

4. Análisis orgánico

El objetivo del análisis orgánico es explicar de forma clara, la lógica de cada módulo. Para una mejor comprensión. Se realiza una breve descripción del comportamiento de cada módulo y en los casos necesarios, de los submódulos en los que se divide para llevar a cabo su función. Habíamos dividido nuestro proyecto en partes funcionales. Ahora explicamos en funcionamiento de cada módulo.

4.1. Módulo de servicio web Java

Como ya observamos en el análisis funcional, el sistema de aviso de llegadas debe contar con un servidor que maneje la información en las mismas. Atendiendo las peticiones que se sucedan. Véase figura 4.1.

4.1.1. Introducción

Una vez tenemos los datos debidamente actualizados. Buscamos la manera de hacerlos accesibles a nuestro teléfono móvil. Aprovechamos el manejo que hemos obtenido del lenguaje de programación Java y programamos un *web service* en Java. También aprovechamos la herramienta XAMPP para instalar en nuestra máquina un servidor Tomcat 7, compatible con Java y con la IDE de desarrollo Eclipse, en la que programar el código mediante la técnica *Bottom Up*, que consiste en escribir el código *Plain Old Java Object (POJO)* [20] para que el IDE implemente el resto.

A la hora de implementar servicios Web es necesario disponer de un medio para el intercambio de datos XML. Se han barajado dos opciones: REST Se ha elegido SOAP por varios motivos. Aunque a simple vista SOAP parece más complejo que REST, éste es únicamente una idea o filosofía extendida débilmente y apoyada por unos pocos. SOAP es un estándar aprobado por el W3C y se dispone de amplia documentación y apoyo por parte de la industria que ya lo utiliza para sus desarrollos.

Por otro lado, aunque REST se presenta como una buena alternativa, aún le queda mucho por madurar, siendo su aspecto más negativo la falta de referencias y documentación para los programadores comunes. SOAP es una tecnología consolidada, con suficiente material bibliográfico, ejemplos... y es la más utilizada para el desarrollo de servicios Web, independientemente del lenguaje de programación y plataforma utilizada. Es más formal, deliberado, y proporciona soluciones fiables para el desarrollo de servicios Web. La estructura y las partes internas de los mensajes SOAP están bien organizadas y son relativamente sencillas. Al igual que REST, SOAP presenta un marco de mensajes basados en XML pero está respaldado por estándares formales de un organismo reconocido como es el W3C. SOAP permite además

mecanismos de expansión que lo hacen una excelente opción para sentar las bases de los servicios Web sobre las cuales se pueden construir capacidades más avanzadas.

Valiéndose del framework Axis2, recibe y envía peticiones SOAP, consultando los datos. Almacenados en archivos *.txt*.

4.1.2. Funcionamiento

El servicio es ubicado en un servidor Tomcat v7.0 escuchando los paquetes que vienen al server por el puerto *6085*. Las peticiones se realizan bajo el protocolo SOAP, y para atenderlas nos valemos del *framework Apache Axis 2*.

Tal y como puede verse en la figura 4.2, la petición se recibe acompañada de una cadena de texto que se corresponde con la información de la parada.

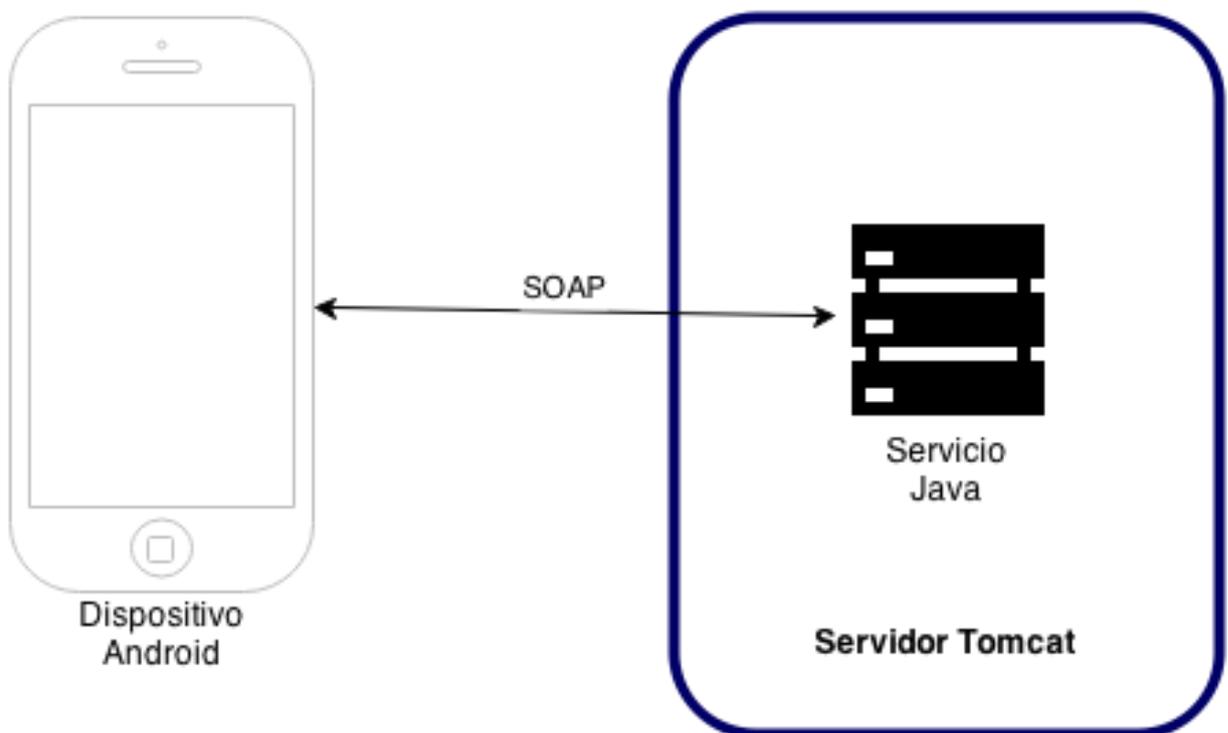


Figura 4.1: Esquema de funcionamiento del servicio web.

```

<v:Envelope
  xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:d="http://www.w3.org/2001/XMLSchema"
  xmlns:c="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:v="http://schemas.xmlsoap.org/soap/envelope/" >
  <v:Header/>
  <v:Body>
    <n0:StartMe
      xmlns:n0="http://server.guaguas" >
      <nombreDeArchivo
        i:type="d:string">
        parada48002.txt
      </nombreDeArchivo>
    </n0:StartMe>
  </v:Body>
</v:Envelope>

```

Figura 4.2: Captura de una petición al servicio sobre una parada concreta.

A continuación el servicio envía la respuesta al dispositivo. Un entero que representa la estimación en minutos del tiempo de llegada de la próxima guagua.

Tras abrir el archivo y procesarlo, el servicio devuelve la información solicitada al terminal, tal y como se muestra en la figura 4.3. En este caso, quedarían 33 minutos para la llegada del siguiente vehículo, que puede leerse dentro del campo *return*. Véase el esquema de la figura 4.4, representa su diagrama de flujo.

```

<?xml
  <soapenv:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" >
    <soapenv:Body>
      <ns:StartMeResponse
        xmlns:ns="http://server.guaguas" >
        <ns:return>
          33
        </ns:return>
      </ns:StartMeResponse>
    </soapenv:Body>
  </soapenv:Envelope>

```

Figura 4.3: Captura de una respuesta de nuestro servicio a una petición.

Su diagrama de clases es sencillo, tal y como vemos en la figura 4.5. Solo es necesaria una clase con 3 métodos.

La clase *MyService* publica tres métodos. Lanzamos peticiones al método *StartMe(String)*. El cual es el que se encarga de devolver el resultado en forma de String.

El primer método, *obtenerInfo(String)*, extrae del archivo la información mediante el método *extraer*. Para luego adaptarlo al formato solicitado mediante un tercer método llamado *quitarAlmohadillas(String)*.

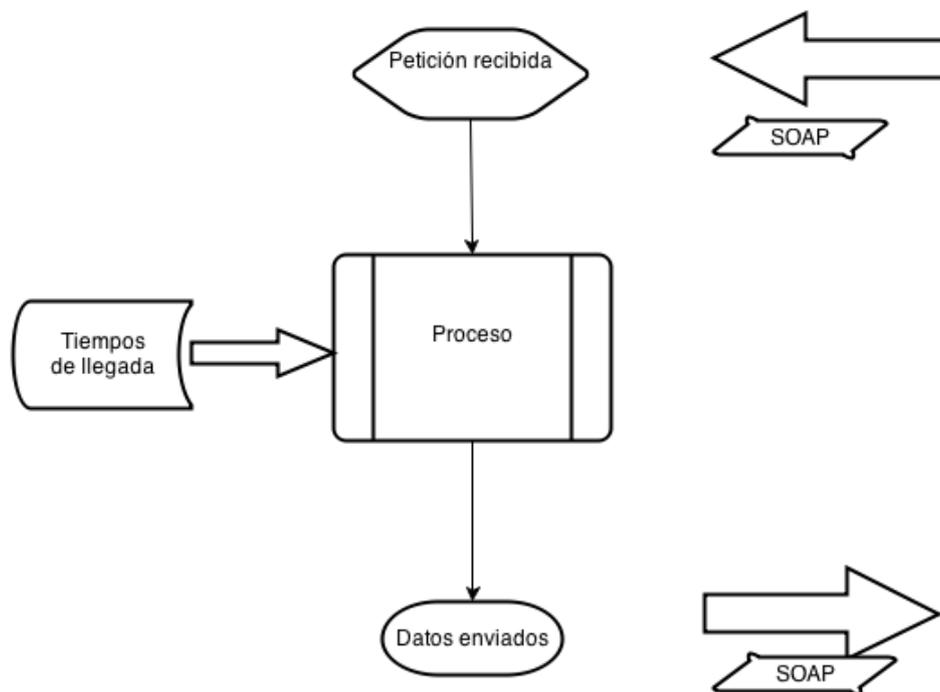


Figura 4.4: Esquema básico de funcionamiento del Java Web Service.



Figura 4.5: Diagrama de clases del web Service.

4.1.3. Estructura del servicio web

La estructura del código de este servicio web es muy simple tal como se muestra en la figura 4.6.

Dentro de la carpeta *Java Resources* encontramos el paquete que alberga la clase Java que gobierna el servicio. Es la encargada de recibir la orden y generar una respuesta. También en la observamos las que el código se sustenta en las siguientes bibliotecas externas:

- Java Runtime Enviroment, 1.7.0.
- Apache Tomcat 7.0.
- Biblioteca de aplicaciones web.

La máquina virtual Java está previamente instalada en el PC donde desarrollamos la aplicación. Es de ahí de donde la toma Eclipse. Una vez hemos especificado su ubicación en nuestro sistema operativo.

También mostramos las dependencias (figura 4.7) que tiene la aplicación dentro de las bibliotecas aportadas por la máquina virtual.

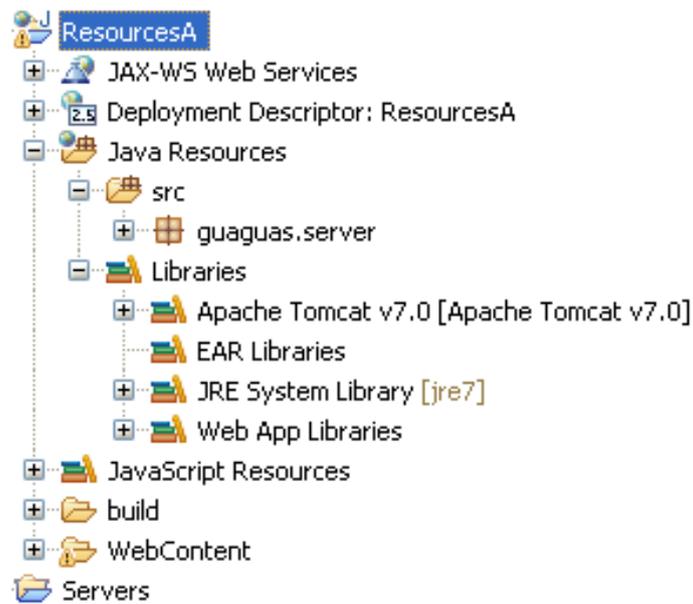


Figura 4.6: Estructura del código del servicio web.



Figura 4.7: Dependencias del JRE.

Al estar el servicio diseñado para funcionar dentro de un servidor *Apache Tomcat 7* genera dependencias en las bibliotecas de dicho servidor tal como se muestra en la figura 4.8.

Dentro de las dependencias del servidor de aplicaciones encontramos las bibliotecas de Axis2, necesarias para trabajar con el framework siguiendo SOAP. También encontramos analizadores sintácticos XML y bibliotecas HTTP necesarias para nuestra solución.

Paquete POJO

MyService.java (figura 4.2) es la clase que procesa a nivel de programación Java las peticiones, recupera los datos y envía la información requerida. Sus métodos se muestran en la figura 4.11. Esta clase se con invocando el método *StartMe(String)*, el cual se valdría de *extraer(String)*, para abrir el archivo correspondiente y de los otros dos para acomodar la información. Al formato que espera nuestra aplicación móvil.

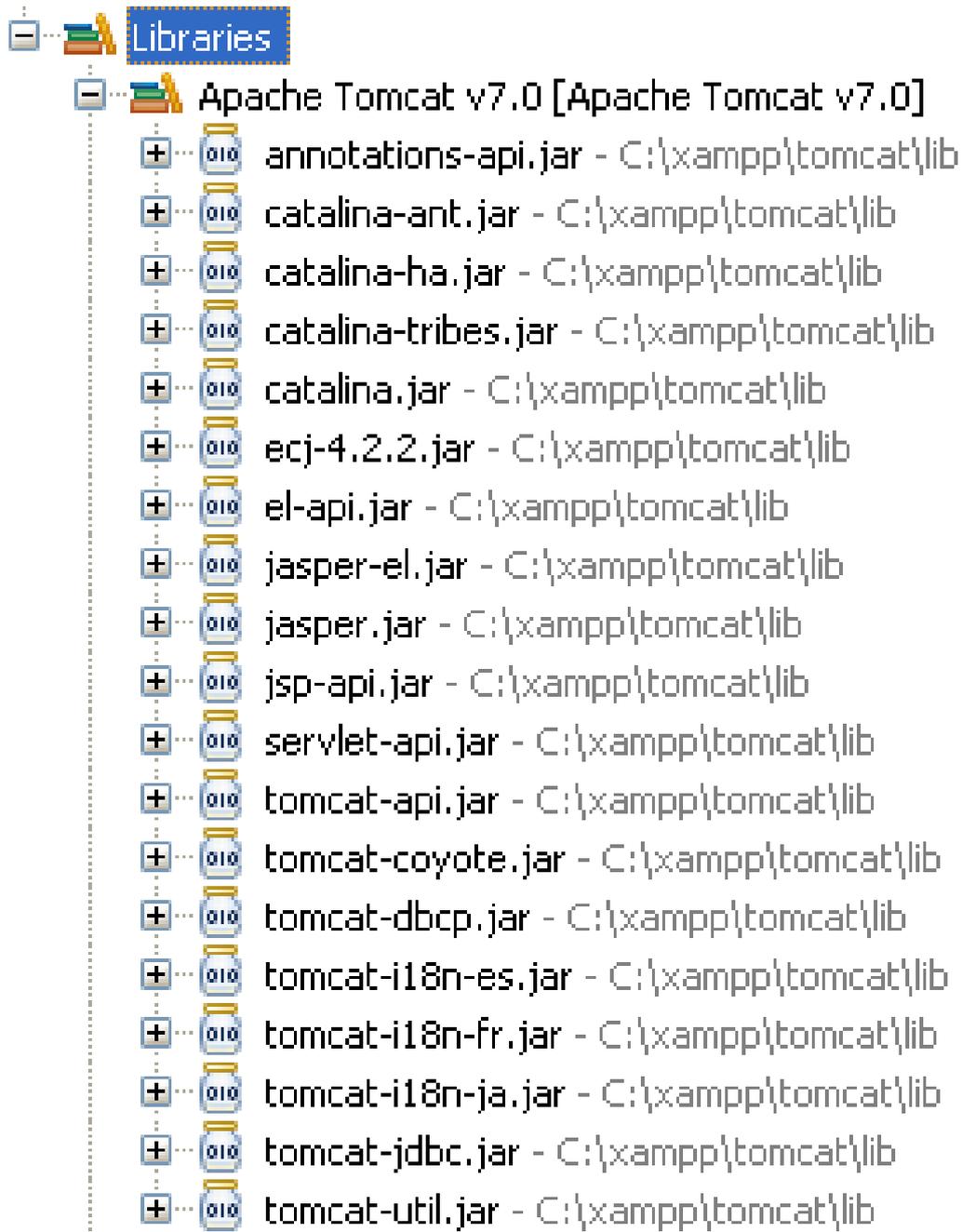


Figura 4.8: Dependencias del servidor Tomcat.

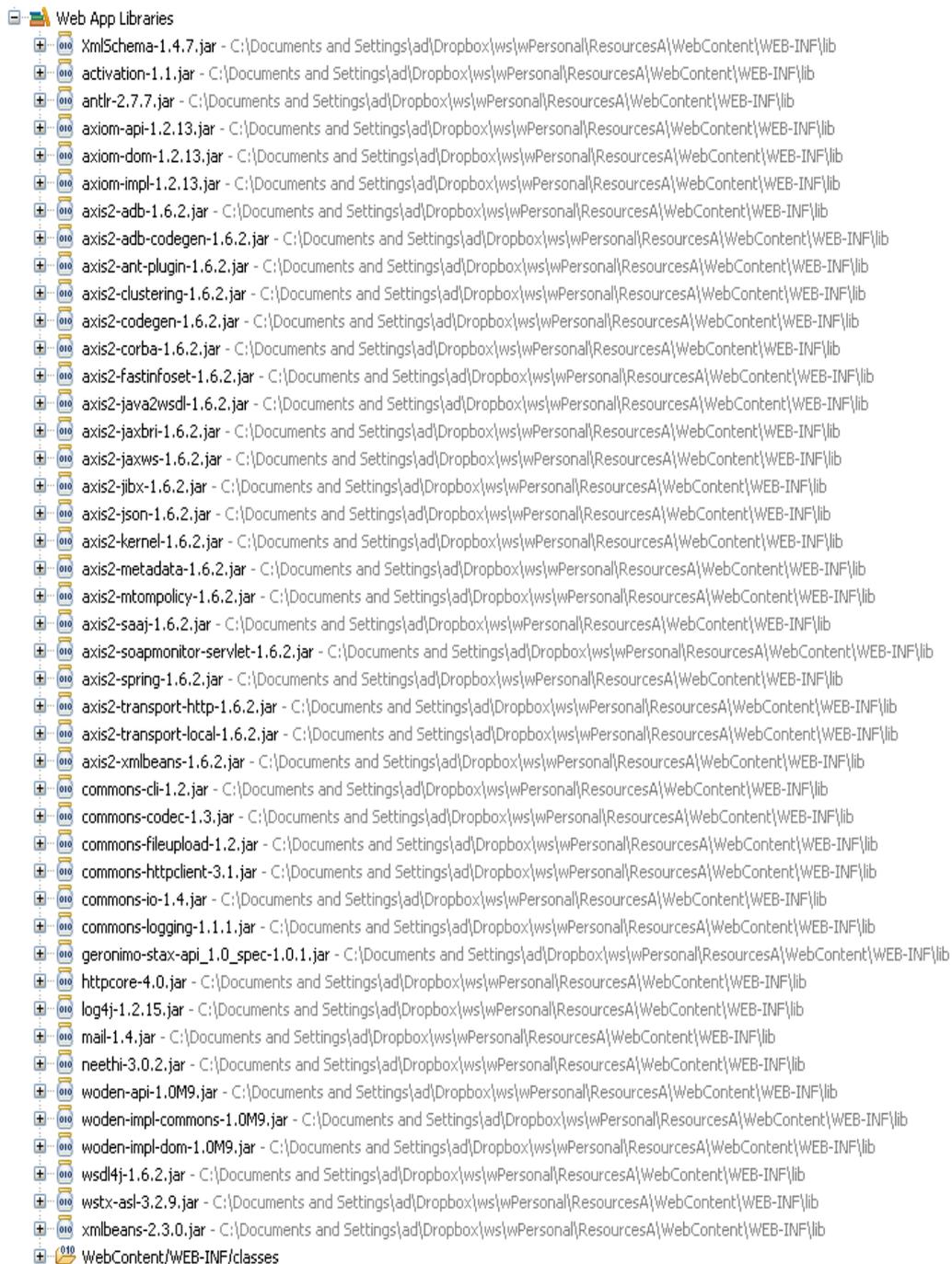


Figura 4.9: Bibliotecas Web App.



Figura 4.10: Ubicación de clase *MyService*.

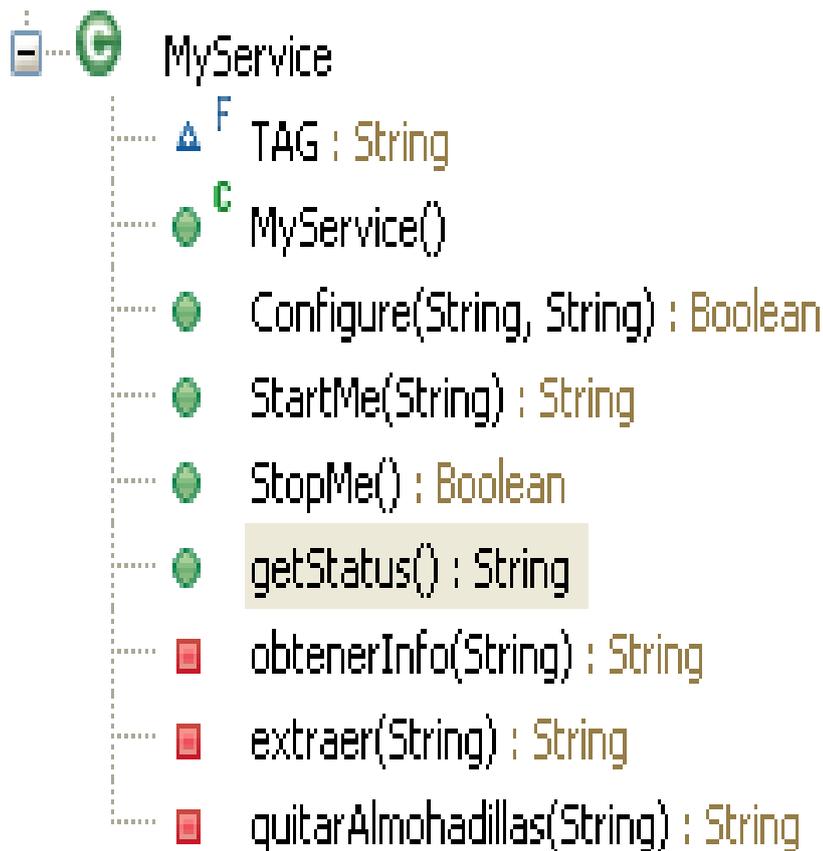


Figura 4.11: Estructura y métodos de la clase *MyService*.

4.2. Modulo Aplicación móvil Java de Android

El objeto de este PFC es esta aplicación. Nos disponemos a analizar su funcionamiento orgánico. Explicamos cómo trabaja el código para ofrecer de una manera sencilla la información al usuario. Evitando gastar muchos recursos de memoria y ancho de banda al dispositivo.

4.2.1. Introducción

La aplicación móvil se diseñó de forma nativa para funcionar en sistema operativos Android a partir de su versión 4.1 (*Jelly Bean*).

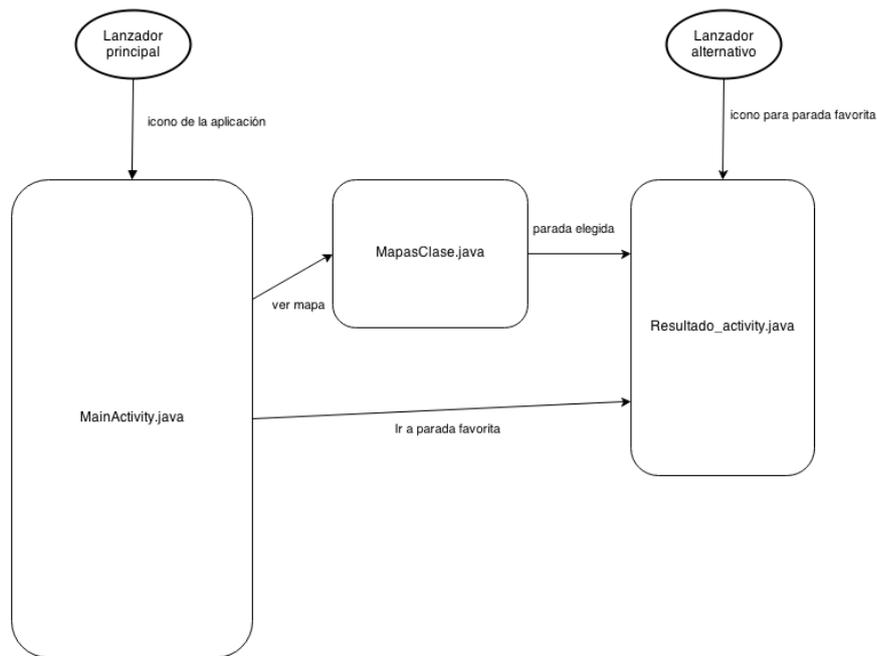


Figura 4.12: Secuencia en la que las *Activity* se muestran al usuario.

La figura 4.12 muestra el recorrido que hace el usuario por la interfaz visual de la aplicación. En una aplicación nativa de Android, normalmente lo que funciona son clases que extienden de la clase *Activity*.

Cada una constituye una *pantalla* o *ventana* de nuestra aplicación. Normalmente, cargan esta interfaz gráfica de un archivo XML guardado en la carpeta **/res/layout**, tal como se muestra en la figura 4.13.

Como ya mostramos en la figura 4.12, el usuario tiene dos vías de entrada en nuestra aplicación móvil, o lo que es lo mismo, la *app* tiene dos lanzadores (*launchers*). Existe un *launcher* alternativo que nos lleva a nuestra parada favorita y nos muestra la información de la misma. Ahorrándonos con ello todo el proceso de búsqueda.

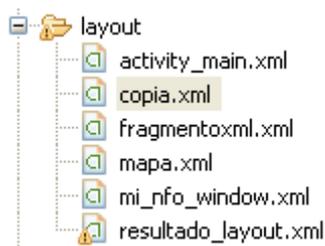


Figura 4.13: Detalle carpeta layout del proyecto en Eclipse.

4.2.2. Archivo XML de localización de parada

Un archivo XML contiene los datos de cada parada de las líneas. La aplicación móvil considera una parada diferente para cada línea cuando dos líneas coinciden. Se encuentra ubicado entre las carpetas de recursos de la aplicación *res/raw*.

Cada ítem constituye una parada, constituida por los siguientes *tags*:

- *Id*: número entero único que diferencia a esa parada del resto y que se utiliza posteriormente en búsquedas.
- *lineaNum*: entero cuya información es el número de línea.
- *Sentido*: 0 cuando el sentido es de ida y un 1 cuando el sentido es de vuelta.
- *codLinea*: concatenación de las dos variables anteriores.
- *Líneas*: cadena de caracteres que representa el origen y destino final de la línea.
- *Title*: cadena de caracteres, La localización o el nombre de la parada, para identificarla.
- *LatLng*: localización, latitud y longitud de una parada separadas por coma.
- *Snippet*: texto opcional aclaratorio. Aporta datos extra sobre la localización o un nombre alternativo.



Figura 4.14: Detalle de parada seleccionada en el mapa de la aplicación móvil.

```

1374 <item>
1375 <id>7002</id>
1376 <lineaNum>7</lineaNum>
1377 <sentido>0</sentido>
1378 <numParada>2</numParada>
1379 <codLinea>70</codLinea>
1380 <lineas>TEATRO - CAMPUS UNIVERSITARIO</lineas>
1381 <title>Lentini (Monopol)</title>
1382 <LatLng>28.102325,-15.415268</LatLng>
1383 <snippet> </snippet>
1384 </item>
1385
1386 <item>
1387 <id>7003</id>
1388 <lineaNum>7</lineaNum>
1389 <sentido>0</sentido>
1390 <numParada>3</numParada>
1391 <codLinea>70</codLinea>
1392 <lineas>TEATRO - CAMPUS UNIVERSITARIO</lineas>
1393 <title>Ctra. del Centro (Antigua Cárcel)</title>
1394 <LatLng>28.09343,-15.425415</LatLng>
1395 <snippet> </snippet>
1396 </item>

```

Figura 4.15: Contenido del archivo XML que aloja la información de las paradas.

La aplicación móvil sustrae posteriormente la información mediante un analizador sintáctico de *XML*. Cada parada constituye un objeto de la clase *Parada2*, que comentamos más adelante. Estos objetos forman un *array* con el que la aplicación trabaja. Véase un ejemplo de uso de esta información, esta vez durante la visualización del mapa, en la figura 4.14.

Elaboración

La información contenida en el XML puede verse en las tablas del anexo A. Quisimos crear las tablas en un entorno manejable y potente y optamos por *Microsoft Excel*. Cuya hoja de cálculo adjuntamos en el citado anexo. Ello nos ha dado la oportunidad de automatizar la escritura de nuestro fichero de paradas. A continuación vemos el script que hemos empleado.

Option Explicit

```
*****
'*
'* Rutina para generar las líneas de código de XML, desde la
'* hoja 'Tabla de Paradas' de Excel con VBA,
'*
'* La 'Código de la línea' (ID) se construye concatenando:
'*     el número de línea (LL),
'*     el sentido de la línea(S),
'*     el número de la parada (PP).
'*     Código: LLSPP
'*
'* Versión: "<?xml version="1.0" encoding="utf-8"?>"
'*
*****
```

Sub Crear_Rutina_XML()

```
Dim Linea As Integer      ' Número de líneas.
Dim Texto As String      '
Dim strItem0 As String   ' Inicio del 'Item'.
Dim strItem1 As String   ' Identificador de Línea.
Dim strItem2 As String   ' Número de línea.
Dim strItem3 As String   ' Sentido de la línea (0/1).
Dim strItem4 As String   ' Número de la parada.
Dim strItem5 As String   ' Cód. línea: Núm.línea + sentido.
Dim strItem6 As String   ' Nombre de la Línea.
Dim strItem7 As String   ' Denominación de la parada.
Dim strItem8 As String   ' Coordenadas (Latitud, longitud).
Dim strItem9 As String   ' Dirección de la parada.
Dim strItem10 As String  ' Final del 'Item

Dim Linea2
Dim Fila As Integer
Dim Parada_id            ' Código parada
Dim Parada_LineaNum 'As Integer ' Número de línea.
Dim Parada_Sentido 'As Integer ' Sentido de la línea (0/1).
Dim Parada_Numero 'As Integer ' Número de parada por sentido
Dim Parada_CodLinea 'As Integer ' Código Línea + Sentido.
Dim Parada_Lineas As String ' Denominación de la línea.
Dim Parada_Title As String ' Denominación de la parada.
Dim Parada_LatLng      ' Coordenadas de las paradas.
```

```

Dim Parada_Snippet As String      ' Dirección de la parada (opcional)

Dim LastRow                      ' Última fila con datos

Texto = "<resources>"
Worksheets("Código XML").Range("A1").Value = Texto
Worksheets("Código XML").Range("A2").Value = ""

' Inicializamos en contador de líneas.
Linea = 1

' Activamos la hoja 'Tabla de datos'
Worksheets("Tabla de datos").Activate

' Buscamos la última fila con datos.
LastRow = ActiveSheet.Range("A1").End(xlDown).Row

' Se inicia la rutina:

' Seleccionamos la primera celda, de la hoja activa.
Range("A1").Select

'Range("A1").Value = Texto

' Se inicia un bucle para leer la 'Tabla de datos',
' desde la primera línea hasta la última (LastRow).
' Pasamos los valores a unas variables intermedias,
' y finalmente la pasamos a la Hoja 'Código XML'.

For Linea = 1 To LastRow

    Worksheets("Tabla de datos").Select
    With Sheets("Tabla de datos")
        Parada_id = ActiveCell.Offset(Linea, 0).Value
        Parada_LineaNúm = ActiveCell.Offset(Linea, 1).Value
        Parada_Sentido = ActiveCell.Offset(Linea, 2).Value
        Parada_Numero = ActiveCell.Offset(Linea, 3).Value
        Parada_CodLinea = ActiveCell.Offset(Linea, 4).Value
        Parada_Lineas = ActiveCell.Offset(Linea, 5).Value
        Parada_Title = ActiveCell.Offset(Linea, 6).Value
        Parada_LatLng = ActiveCell.Offset(Linea, 7).Value
        Parada_Snippet = ActiveCell.Offset(Linea, 8).Value
    End With

```

```

' Pasamos los datos a la variable intermedia.

strItem0 = "<item>"          ' Inicio del 'Item'.
strItem1 = "<id>" & Parada_id & "</id>"
strItem2 = "<lineaNum>" & Parada_LineaNúm & "</lineaNum>"
strItem3 = "<sentido>" & Parada_Sentido & "</sentido>"
strItem4 = "<numParada>" & Parada_Numero & "</numParada>"
strItem5 = "<codLinea>" & Parada_CodLinea & "</codLinea>"
strItem6 = "<lineas>" & Parada_Lineas & "</lineas>"
strItem7 = "<title>" & Parada_Title & "</title>"
strItem8 = "<LatLng>" & Parada_LatLng & "</LatLng>"
strItem9 = "<snippet>" & Parada_Snippet & "</snippet>"
strItem10 = "</item>"

' Seleccionamos la hoja donde vamos escribir los datos.
Worksheets("Código XML").Select

With Sheets("Código XML")
    ActiveCell.Offset(1, 0).Value = strItem0
    ActiveCell.Offset(2, 0).Value = strItem1
    ActiveCell.Offset(3, 0).Value = strItem2
    ActiveCell.Offset(4, 0).Value = strItem3
    ActiveCell.Offset(5, 0).Value = strItem4
    ActiveCell.Offset(6, 0).Value = strItem5
    ActiveCell.Offset(7, 0).Value = strItem6
    ActiveCell.Offset(8, 0).Value = strItem7
    ActiveCell.Offset(9, 0).Value = strItem8
    ActiveCell.Offset(10, 0).Value = strItem9
    ActiveCell.Offset(11, 0).Value = strItem10
    ActiveCell.Offset(12, 0).Select
End With

Next Linea ' Nueva línea

Texto = "</resources>"
ActiveCell.Offset(1, 0).Value = Texto

End Sub

</resources>

```

4.2.3. Submódulo Activity del Menú principal

El menú principal de la aplicación alberga las opciones que ella ofrece. Da paso pues, al mapa de selección de parada en sus tres variantes o a la información de la parada favorita del usuario.

Su funcionamiento es muy simple. Se presentan principalmente dos opciones de funcionamiento:

- *Paradas cercanas a mí*: muestra las paradas que están cerca del usuario.
- *Paradas en una zona*: muestra un mapa completo de Las Palmas de Gran Canaria para que el usuario seleccione la zona de la que desea obtener las paradas.

Las anteriores opciones son suficientes para la funcionalidad de la aplicación, pero también, como alternativa se ofrecen estas dos opciones:

- *Mostrar mapa*: muestra todas las paradas de almacenadas.
- *Parada favorita*: atajo a nuestra parada favorita, para saltarnos el modo mapa e ir más rápido.

Para dar paso a las actividades siguientes, la clase *MainActivity* recopila ciertos datos que empaqueta en un *Intent*.

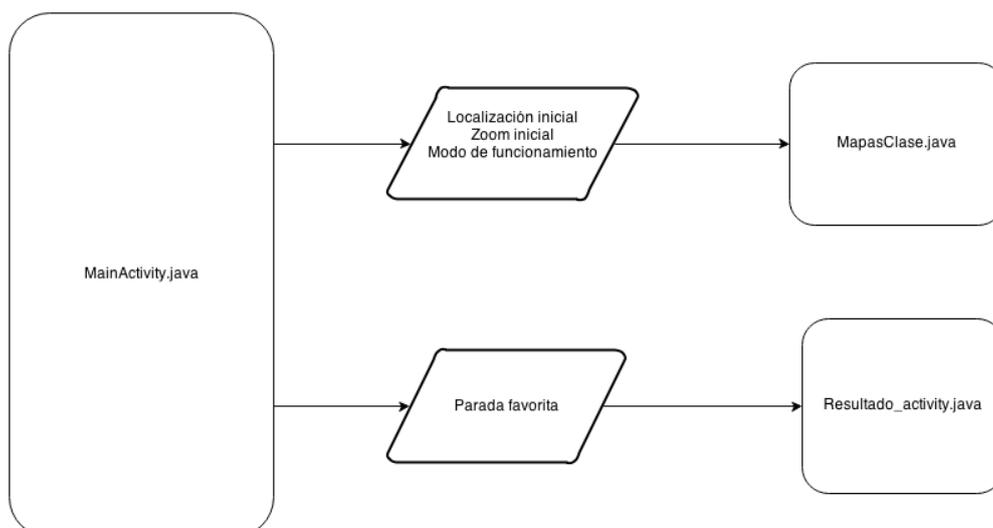


Figura 4.16: Vínculos de la clase MainActivity con el resto de módulos.

```

public void lanzaMapa(View v){
//Lanza el mapa con todas las paradas y una posición por defecto.
Intent intent = new Intent(this ,MapasClase.class);
intent.putExtra(TAG_ACCION, ACCION_MOSTRAR_TODO);
intent.putExtra(LATITUD_DOUBLE_TAG, LPGC.latitude);
intent.putExtra(LONGITUD_DOUBLE_TAG, LPGC.longitude);
intent.putExtra(ZOOM_TAG,
ZOOM_VISTA_GENERAL_POR_DEFECTO);

startActivity(intent);
}

```

Figura 4.17: Método que lanza el mapa mostrando todas las paradas.

Esta *Activity* es la encargada de lanzar un *intent* invocando al mapa donde el usuario puede localizar las paradas. En ese *intent* debe estar contenida la información de cómo debe ejecutarse la *Activity* que contiene el mapa.

En la figura 4.16 describimos las formas de invocar al mapa según el modo de funcionamiento elegido por el usuario.

Mostrar todo

Tal y como se muestra en la Figura 4.17 el *intent* que lanza se provee de un objeto identificativo *String* que junto con un valor que ambas clases conocen, sirve a la siguiente *Activity* para conocer el modo deseado.

La clase *MainActivity* debe dar valores de zoom, localización, modo de funcionamiento. Se da una posición y zoom que abarca toda la ciudad. Con estos valores va jugando la actividad para controlar y dar paso al mapa.

```

public void cercanasSinGPS(View v){
//Lanza el mapa de LPGC sin paradas. El usuario elige en que zona hacer zoom y verlas
Intent intent = new Intent(this ,MapasClase.class);
intent.putExtra(TAG_ACCION, ACCION_CERCANAS_SIN_GPS);
intent.putExtra(LATITUD_DOUBLE_TAG, LPGC.latitude);
intent.putExtra(LONGITUD_DOUBLE_TAG, LPGC.longitude);
intent.putExtra(ZOOM_TAG, ZOOM_VISTA_GENERAL_POR_DEFECTO);
startActivity(intent);
}

```

Figura 4.18: Método que lanza el mapa mostrando las paradas de una zona.

```

private void localizar() {
// Lanza una petición de localización
    peticiónDeLocalización = LocationRequest.create();
    peticiónDeLocalización.setPriority(100);
    clienteDeServicioLocalización = new
    LocationClient(getApplicationContext(), this, this);
    clienteDeServicioLocalización.connect();
}

```

Figura 4.19: Método para petición de localización.

Mostrar paradas de una zona

Tal y como se muestra en la figura 4.18, en este modo entregamos los mismo valores, solo que cambiamos la *acción*. De este modo el comportamiento es distinto, el cual vemos más adelante.

Mostrar paradas cercanas a mí

Este modo solo se ofrece al usuario en caso de haber obtenido una localización con anterioridad lo suficientemente reciente y precisa. Ella se intenta obtener de forma asíncrona utilizando *listeners*. Es decir, implementando un *LocationListener* en la propia clase (figura 4.19).

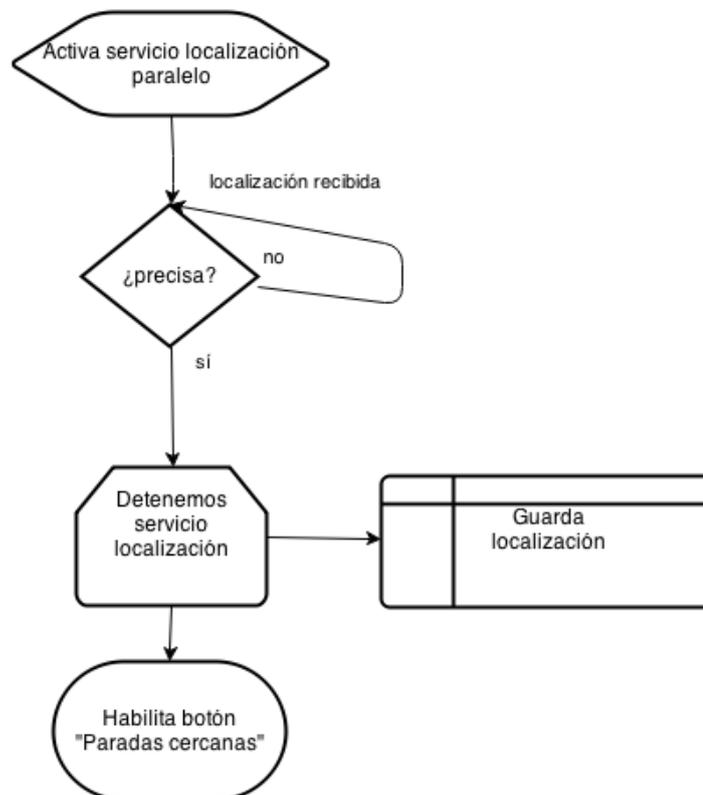


Figura 4.20: Control sobre la opción *Paradas cercanas a mí*.

Se instancia un *LocationClient*, estableciendo como *listener* nuestra propia clase. Así conseguimos trabajar de forma asíncrona. Cuando nos llega un dato se evalúa su idoneidad y al obtener el primer dato válido matamos el servicio.

Al obtenerlo se habilita el botón. De esta forma conseguimos instanciar el mapa conociendo previamente la posición deseada. Ahorrando recursos y tiempo a los terminales.

Dicho proceso está activo durante el tiempo de vida de la *MainActivity.java* o cuando se satisfaga el requerimiento de obtener una localización precisa. En cuyo caso quedaría la opción *Paradas cercanas a mí* habilitada.

Si finalmente el usuario opta por esta acción, se carga un *intent* ejecutando el código de la figura 4.21. Damos en él nuestra localización para que el mapa que visualizar se sitúe sobre ese punto.

Parada favorita

A diferencia de las otras opciones. Aquí la *MainActivity* da paso a la actividad que muestra los minutos que quedan. En el *intent*, especificamos que debe mostrar directamente la parada que se estableció previamente como favorita. Únicamente con este valor.

```
public void lanzaParadasCercanas(View v){
// Mostrar las paradas cercanas al usuario.
    adaptarRequest();
    Intent intent = new Intent(this ,MapasClase.class);
    intent.putExtra(TAG_ACCION,ACCION_PARADAS_CERCANAS);
    intent.putExtra(RADIO_PARADAS_TAG, RADIO_PARADA_METROS);

    double latitud = ultimaLocalizacion.getLatitude();
    double longitud = ultimaLocalizacion.getLongitude();
    float zoom = ZOOM_PARADAS_CERCANAS_POR_DEFECTO;

    intent.putExtra(LATITUD_DOUBLE_TAG, latitud);
    intent.putExtra(LONGITUD_DOUBLE_TAG, longitud);
    intent.putExtra(ZOOM_TAG, zoom);

    startActivity(intent);
}
```

Figura 4.21: Método para lanzar el mapa, centrándonos en las paradas más cercanas.

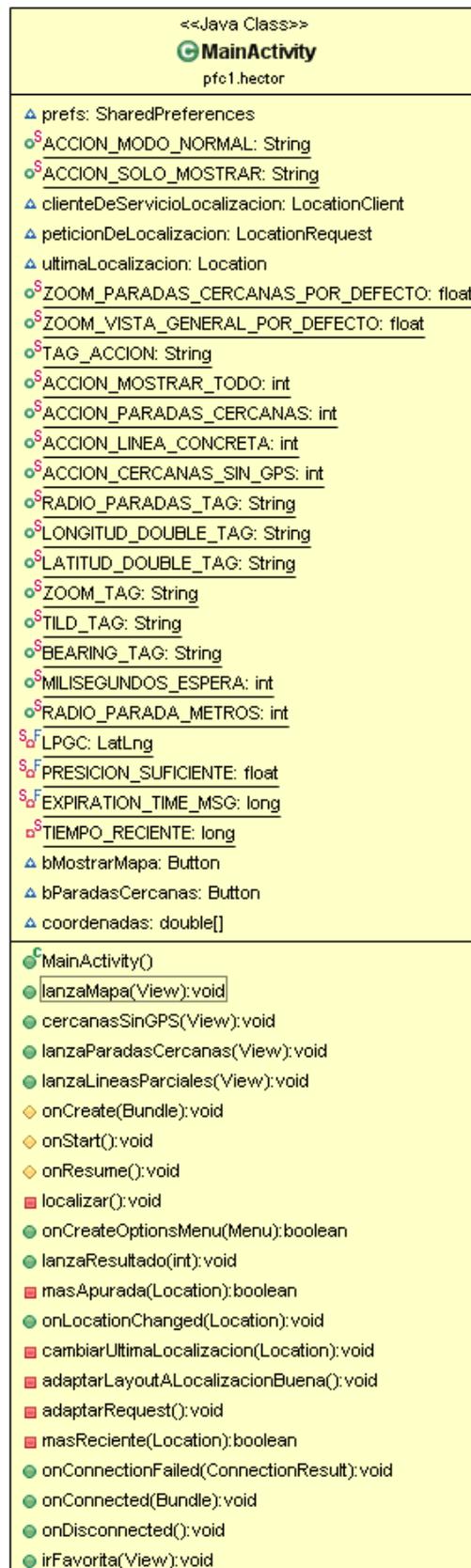


Figura 4.22: Diagrama de clases de *MainActivity*.

4.2.4. Submódulo Activity donde encontrar parada

La funcionalidad de selección de parada gira, como podemos observar en la figura 4.14, en torno a la clase *MapasClase*, que extiende de la clase padre *Activity*.

Hemos creado para la aplicación la clase *Parada2* (figura 4.23). Cada objeto representa los datos de una parada, tales como coordenadas, número identificativo, información... Pero también este objeto es capaz de asociarse con un objeto *Marker*.

En su mayoría, las variables de esta parada son las mismas que las que existen en los campos del fichero *paradas.xml*. Siguiendo las recomendaciones de programación Java, hemos dotado a la clase de métodos *getter & setters*, que pueden ser invocados desde otras clases para obtener o modificar los datos que alberga.

Para el visionado del mapa hemos recurrido a la API *Google Maps*, dentro de los servicios de *Google*. Nos hemos decantado por este sistema al darnos cuenta de las funcionalidades que ofrece. Tanto de zoom, inclinación y animaciones como la posibilidad de dibujar formas básicas o personalizar los iconos. Cosas que como hemos explicado nos han sido de mucha utilidad.

En el diagrama de la figura 4.24 se observa que el funcionamiento de la clase principal de este submódulo corresponde a un bucle. En el cual hay una parte de inicialización, donde se analiza en *Intent* entrante. En él, analizamos las variables que nos llegan desde el submódulo anterior, para instanciar con ellos el un *GoogleMap*.

La carga de la lista de paradas se realiza atendiendo a los parámetros llegados del *Intent*. Y se tiene en cuenta si debemos poner un radio o no. Si no nos llegan coordenadas iniciales en el *Intent* aplicamos coordenadas por defecto que tenemos. Y mostramos el mapa.

Una vez cargado el mapa, el programa filtra las paradas que se dibujan. En este proceso aplicamos un criterio de cercanía. Averiguando la distancia desde nuestro punto de foco a la parada y comparándola con el máximo permitido. Así obtenemos un *array* de objetos *Parada2*.



Figura 4.23: Métodos y variables de la clase Parada2.

Una vez tenemos el *array* con los objetos *Parada2* que dibujar comenzamos a colocarlos en el mapa. En primer lugar obtenemos de él su localización geográfica exacta por medio de los valores de latitud y longitud que alberga el objeto. En segundo lugar vemos de que línea se trata para dibujar el logo que corresponda a la ruta.

Cada *Marker* a su vez, lleva asociado un objeto de la clase *InfoWindowAdapter*, que mostramos en el diagrama de clases. Este objeto despliega una pequeña ventana emergente cada vez que pulsemos sobre un *Marker*. Ello se realiza mediante un *listener* para tal efecto del que nuestra clase principal es interfaz.

Mientras, seguimos teniendo plena libertad para navegar por el mapa. Podemos alejarlo, torcerlo, inclinarlo... Tanto así que si tocamos en cualquier parte del mapa que no tenga un icono de parada. El mapa lanza una animación en la que centra el mapa en el lugar elegido. También dibuja las paradas cercanas a ese punto y un círculo centrado en el mismo. En ello consiste el bucle de nuestro mapa.

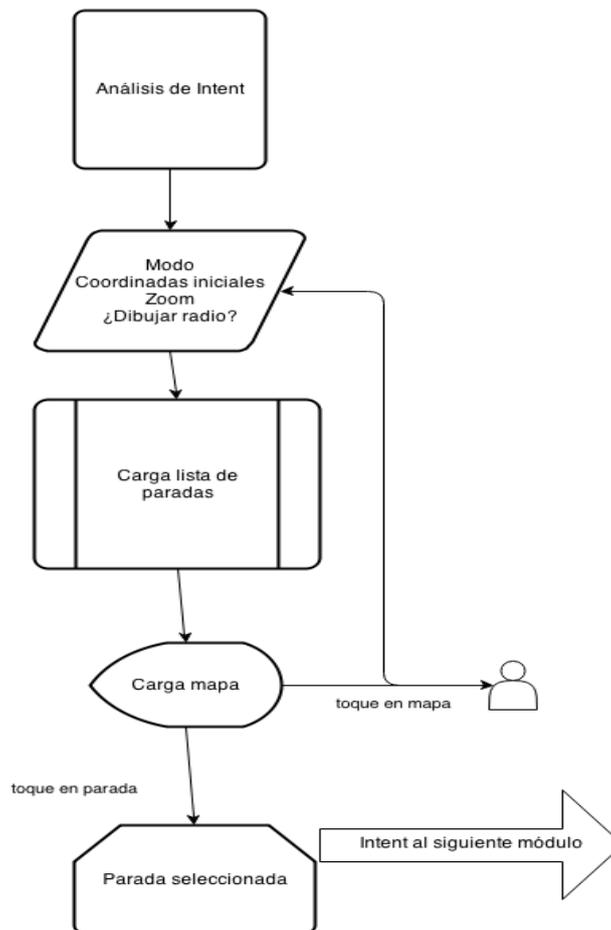


Figura 4.24: Diagrama de funcionamiento de la Activity MapasClase.

Interrumpimos el bucle cuando habiendo seleccionado una parada, presionemos en su menú emergente previo. Con ello lanzamos un *Intent* en el cual albergamos el dato del número de parada elegida, dado que la clase *Parada2* no puede estar contenida en un *Intent*.

Este *Intent* invoca el siguiente submódulo, mediante el código que se muestra en la figura 4.25.

```

private void lanzarActividadResultado(Parada2 paradaEncontrada) {
// Auto-generated method stub
    Intent intent = new Intent(this, Resultado_activity.class);
    intent.putExtra(Resultado_activity.TAG_PARADA_ID,
        paradaEncontrada.getId());
    intent.setAction(MainActivity.ACCION_MODO_NORMAL);

    startActivity(intent);
}
  
```

Figura 4.25: Método de para lanzar siguiente submódulo.

4.2.5. Submódulo Activity de muestra de resultado

Este es el módulo encargado de mostrar finalmente la información al usuario. Se compone de una ventana *Activity*, que es presentada en el diagrama de la figura 4.26 como un cuadro informativo y un servicio autónomo.

En primer lugar, la *Activity* extrae del *Intent* entrante que la invoca el número de parada seleccionada por el usuario. Con ese, este número busca la información sobre la parada para cargarla en pantalla. Puede verse esto en la descripción de la interfaz de usuario.

Posteriormente, se inicia el servicio. Usa un servicio y obtiene los datos de manera indirecta nos da la oportunidad de obtenerlo aunque el usuario no se encuentre atendiendo el móvil en ese momento o esté realizando otras tareas con el mismo.

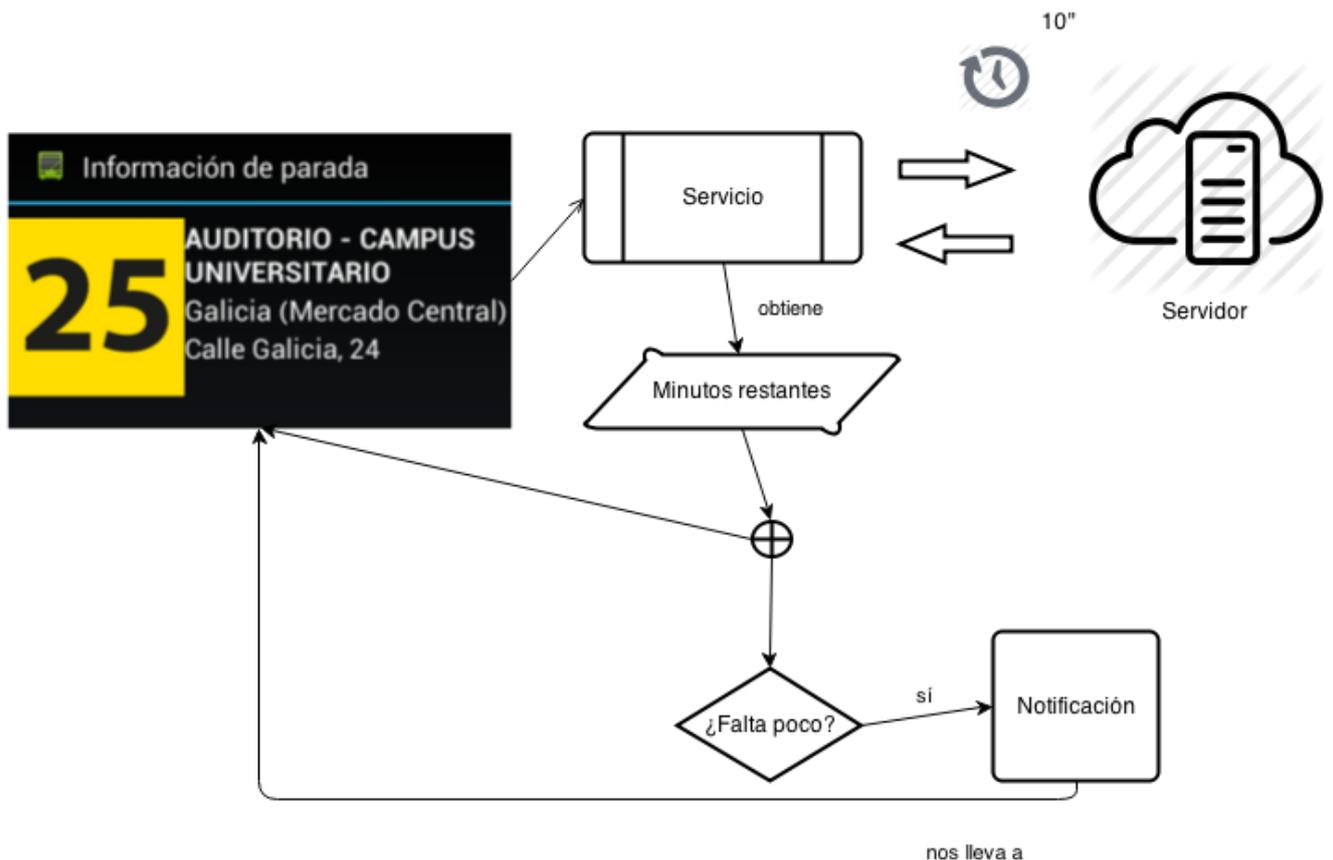


Figura 4.26: Análisis orgánico del submódulo de resultado.



Figura 4.27: Diagrama de la función de visualización/selección de parada.

```

public void conectarServicio(View v){
// Conecta con el servicio con un Intent
    Intent intent = new Intent(this, ComprobarTiempoService.class);

// startService(intent);
    Log.d("conectarServicio - result","Ejecutando bindService.");
    bindService(intent, mConexion, Context.BIND_ALLOW_OOM_MANAGEMENT);
}

```

Figura 4.28: Fragmento de código de la Activity.

En la figura 4.28 se muestra el código de invocación del servicio, en el que establecemos el *flag* `Context.BIND_ALLOW_OOM_MANAGEMENT` declaramos así que el servicio es independiente a la vida de de la Activity.

En la figura 4.29 mostramos como usando *Messenger*, la Activity pide al servicio los minutos restantes.

```

private void comunicarAlServicio(int id) {
// Envía la id de la parada al servicio para que la busque.

    Log.d("comunicarAlServicio - Res","Preguntamos minutos al servicio.");
    try {

        Message msg = Message.obtain();
        msg.what = ComprobarTiempoService.MSG_PREGUNTAR_MINUTOS_QUE_QUEDAN;
        msg.arg1 = id;
        msg.replyTo = mMessenger;

        mServicio.send(msg);
    } catch (RemoteException e) {

// El servicio ha fallado antes de que podamos hacer nada.
        Log.e("Comunicar al servicio","El servicio ha fallado ");
    } catch (NullPointerException e) {

// Handle exception
        Log.e("Comunicar al servicio","Capturada excepción al tratar de
enviar el mensaje");
    }
}

```

Figura 4.29: Fragmento de código de conexión con el servicio.

El servicio continua muestreando los minutos, y cuando quedan 5 minutos, lanza una notificación que conlleva un aviso acústico. Por tanto el usuario puede despreocuparse y realizar otra tarea. Ya el servicio nos avisa.

Responde a las peticiones de la *Activity* en caso de que se produzcan más peticiones de actualización, como vemos en este código.

Una vez hecho esto el servicio muere para no consumir recursos. El diagrama de clases se muestra en la figura 4.31.

4.2.6. Estructura del proyecto Eclipse

La figura 4.32 muestra la estructura del proyecto Eclipse.

En ella se ven representados distintos elementos que necesitamos para el desarrollo y funcionamiento de la aplicación. Empezando por arriba, vemos Una biblioteca de *Google* de la cual podemos tomar submódulos según nuestras necesidades. En este caso hemos cogido la API de *Google Maps*.

Entrando en el proyecto primeramente vemos los POJOs, es decir, el código escrito por nosotros que gobierna esta aplicación. Dentro de la carpeta *src*.

```
Message mensajeConMinutosQueQuedan = Message.obtain();
mensajeConMinutosQueQuedan.what = ComprobarTiempoService.MSG_MINUTOS_QUE_QUEDAN;
mensajeConMinutosQueQuedan.arg1 = minutos;
mensajeConMinutosQueQuedan.arg2 = 34;
mMessengerDelCliente.send(mensajeConMinutosQueQuedan);
catch (RemoteException e) {
    // handle exception
    Log.e("enviarMinutos", "Se ha desconectado el cliente");
    isConnected = false;
}
}
```

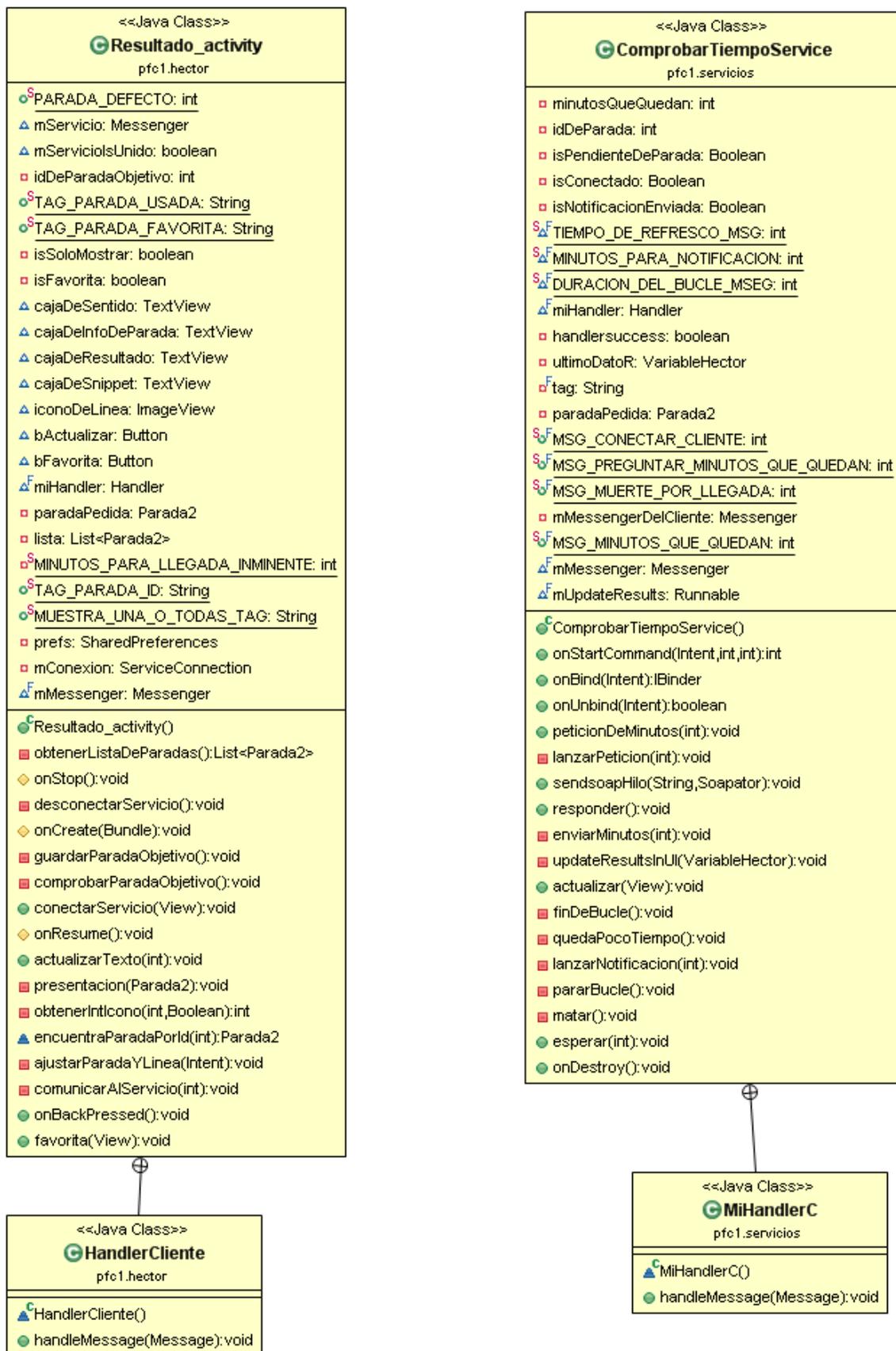


Figura 4.31: Diagrama de clases de la Activity y el servicio.

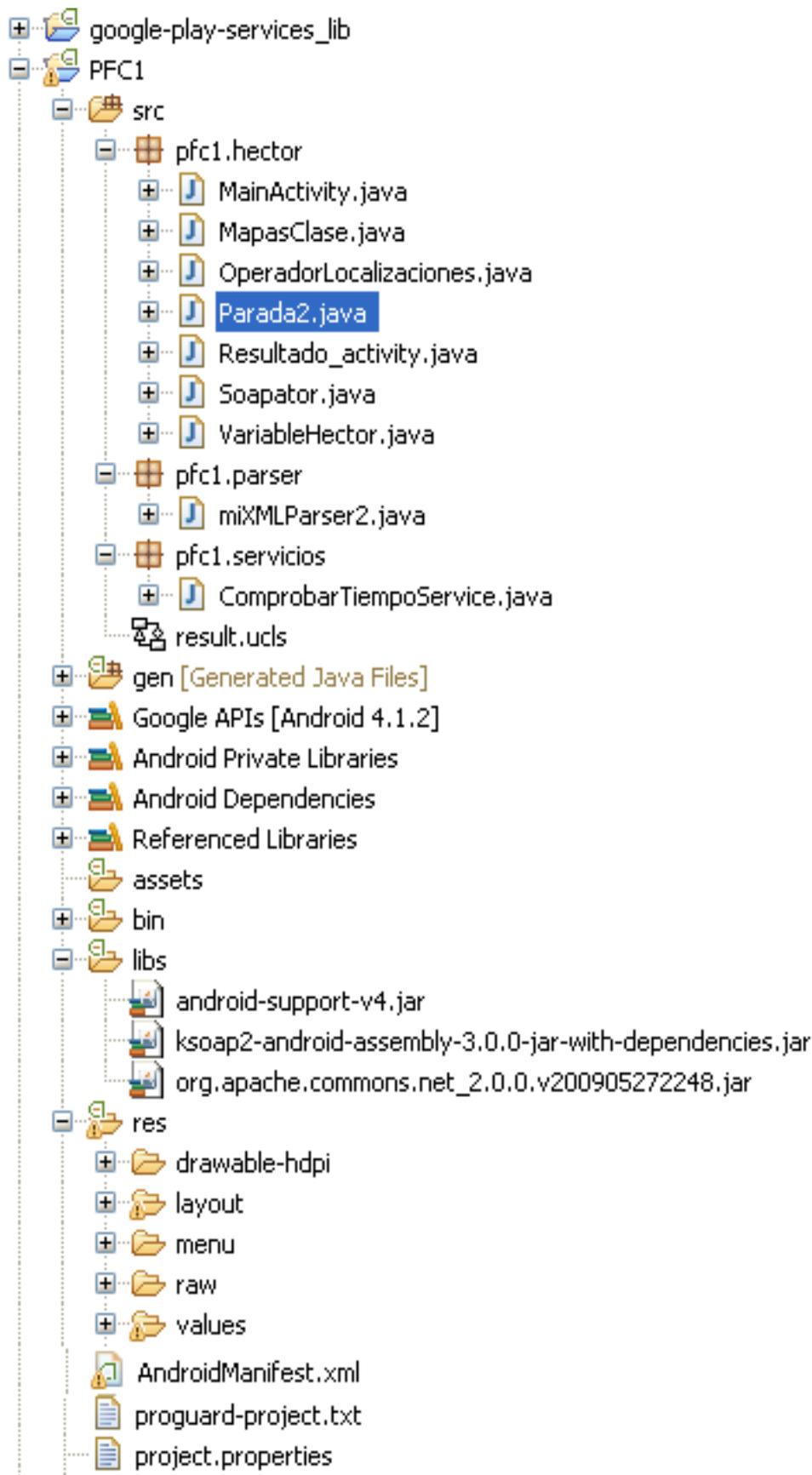


Figura 4.32: Estructura del proyecto en Eclipse.

La carpeta *gen*, que viene a continuación se genera automáticamente por el compilador y contiene referencias a variables, archivos... Es la que conecta los recursos en XML de la carpeta *res*, con el código Java.

Le siguen bibliotecas, de entre la que destacamos la Google APIs, necesaria para utilizar los servicios de *Google* de que nos valemos en este módulo. Las demás son comunes a cualquier proyecto *Android*, exceptuando las contenidas en la carpeta *libs* de más abajo. De entre ellas son importantes la *ksoap2* y la *apache.commons*, de las que obtenemos las rutinas necesarias para comunicarnos por el protocolo *SOAP*.

POJOs Java

El código Java está organizado en tres paquetes, en los cuales hemos separado el analizador sintáctico de XML y el servicio del resto de clases:

- *pfc1.hector*: paquete principal, en el albergamos las *Activity* con las clases que las complementan.
- *pfc1.parser*: hemos dispuesto este paquete para el parseador XML, que la clase encargada de recorrer el fichero XML que alberga las paradas para construir un *array* de *Parada2*.
- *pfc1.servicios*: el servicio que muestrea el tiempo en el servidor, y encargado de las comunicaciones con este.

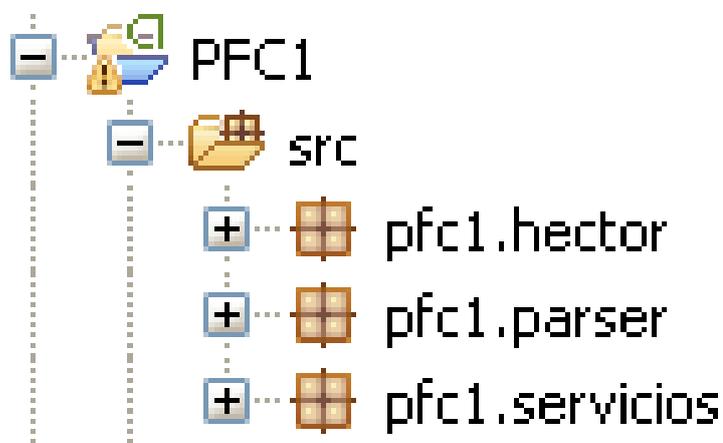


Figura 4.33: Paquetes en los que se encuentra el código de la aplicación.

4.2.7. Módulo de simulación

El módulo de simulación de datos tiene por objeto generar datos, como los que se pudieran generar en un escenario real para comprobar cómo los procesa nuestro sistema.

Almacenamiento de datos

A la hora de decidirnos como almacenar los datos ha primado la sencillez en la programación. Habida cuenta de que es una simulación y de que no atendemos a especificaciones de rapidez o escalabilidad. Por ello, dada la amplia documentación al respecto y la sencillez en el manejo, hemos organizado los datos en archivos de texto plano. Cada archivo guarda el número de minutos que quedan a cada parada, como se muestra en la figura 4.34.

Cada archivo tiene como nombre la palabra *parada* seguida de un número único para cada una y la extensión del archivo (txt).

El manejo de esta información es fácil con cualquier código de programación.

Para su gestión hemos decidido usar 2 códigos php, ejecutado con la ayuda de la herramienta XAMPP, de la que ya hemos hablado anteriormente. Los códigos cumplen con la funcionalidad descrita para el módulo funcional explicado. Los ejecutamos en un servidor Apache, que escucha las peticiones HTTP que hagamos. Con las que gobernamos su comportamiento.

En la figura 4.35 mostramos un fragmento del código escrito para la funcionalidad de actualización. Donde puede verse en primer lugar un pequeño algoritmo que pide una clave para la ejecución del código. A continuación vemos el principio de una serie de bucles *for* en cascada. Dichos bucles son los que crean cada serie de archivos de parada. Hay un bucle *for* por cada rango de IDs de paradas. Que están organizadas tal y como mostramos en el anexo A.

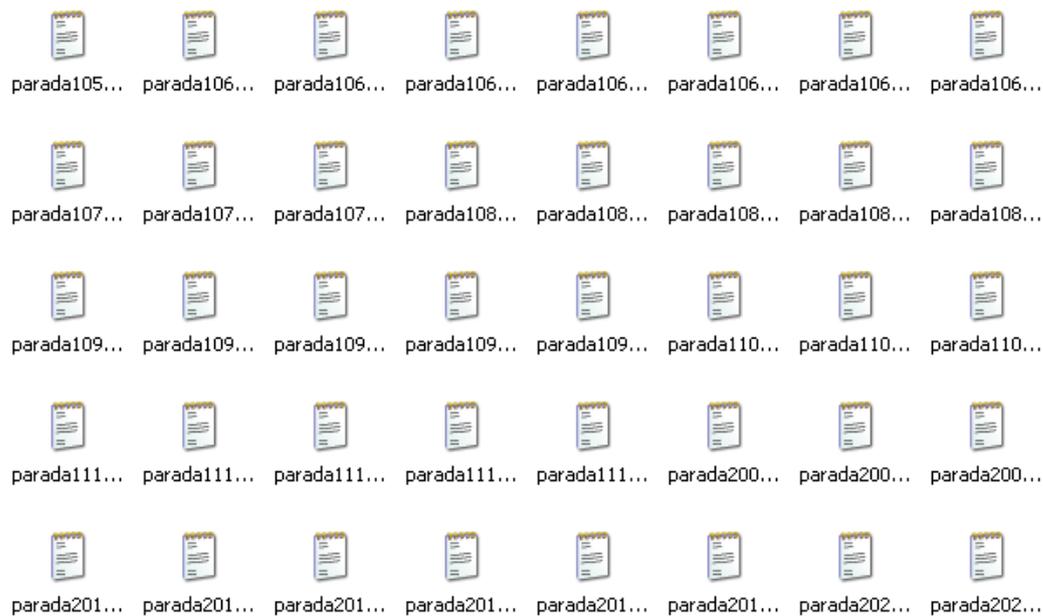


Figura 4.34 Organización de datos de minutos restantes por parada.

```

<?php
$clave = $_GET['clave'];
if($clave!='2013si') exit("&uacute; acceso no permitidos");

for ( $i = 1001 ; $i <= 1117 ; $i ++ ) {
    $nombre = "paradas/parada".$i.".txt";
    $archivo=file($nombre);
    $row = explode("##", $archivo[0]);
    $iscambio = FALSE;
    if ( $row[1] != 0 && $row[1]>0 )
    {
        $linea1=$row[1]-1;
    }
    else { $linea1=rand(10, 50);}

    $fh = fopen($nombre, 'w') or die("can't open file");

    fwrite($fh, "L1:");
    fwrite($fh, "##"); #se corresponde con la linea de numero
    fwrite($fh, $linea1);
    fwrite($fh, "##");

    fclose($fh);
}
for ( $i = 2001 ; $i <= 2119 ; $i ++ ) {
    $nombre = "paradas/parada".$i.".txt";
    $archivo=file($nombre);

```

Figura 4.35: Fragmento del código php de actualización.

5. Descripción a nivel usuario

En este capítulo hacemos un recorrido visual por la aplicación. Nos movemos desde el punto de vista de un usuario y vemos las opciones presentadas en el terminal móvil, que es el que ve el viajero.

5.1. Preparando el terminal

Es recomendable, aunque no obligatorio, activar la opción de GPS de nuestro dispositivo Android.

Aunque nuestra aplicación móvil puede funcionar sin el uso del GPS, debemos tenerlo activado si queremos usar todo su potencial.

En primer lugar, debemos deslizar el dedo desde la parte superior a la inferior de la pantalla, para desplegar el menú de notificaciones. Como puede verse en la

figura 5.1. En la barra de notificaciones el botón *localización*  debe estar encendido, de no ser así debemos pulsarlo para que quede en estado encendido .

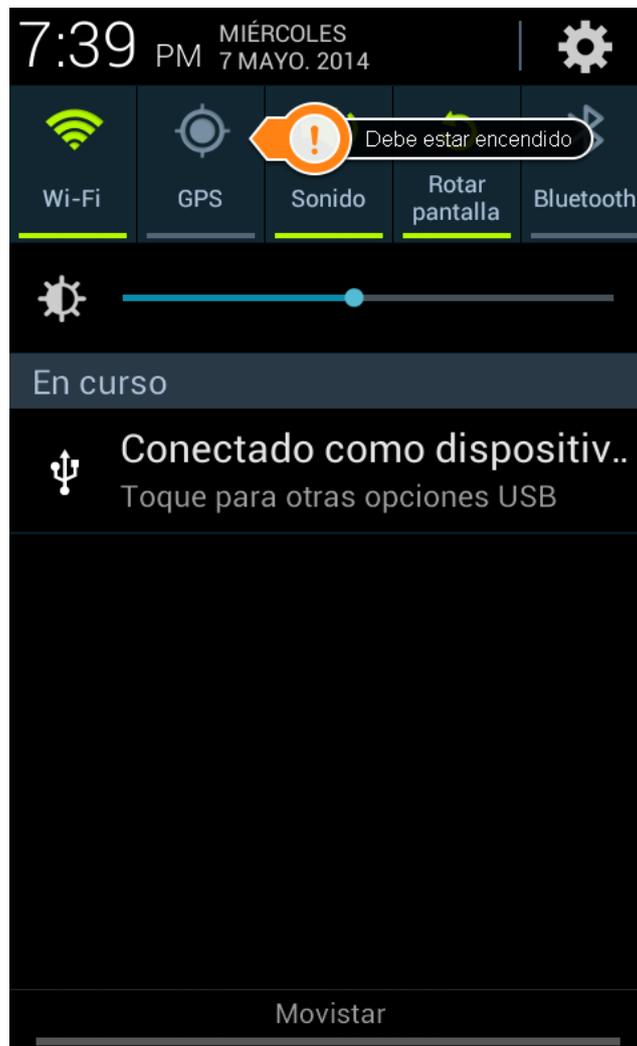
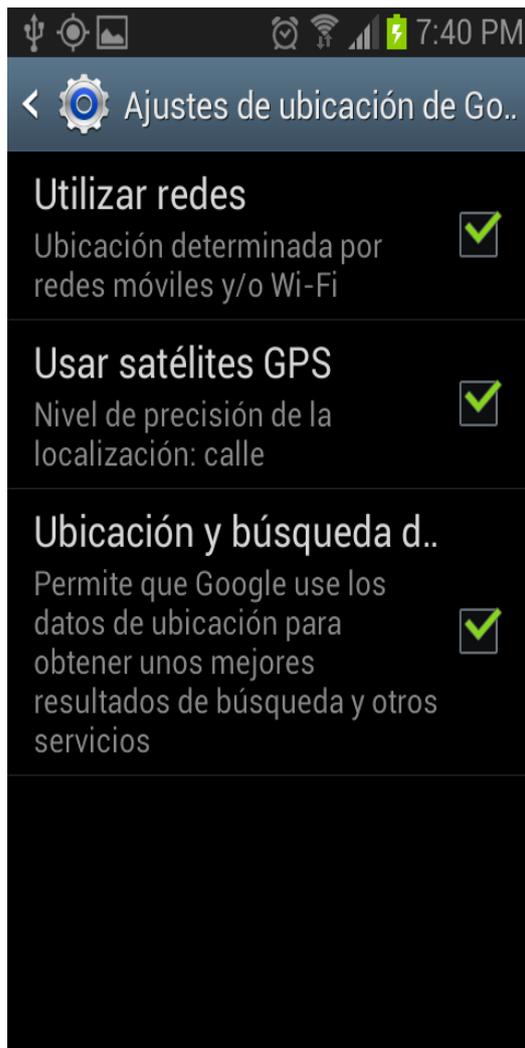


Figura 5.1: Localización del acceso al menú de GPS.



(a)



(b)

Figura 5.2: a) Menú de Ajustes de ubicación de Google, b) Captura de la barra de notificaciones.

Adicionalmente, manteniendo el dedo sobre el citado botón, accedemos al menú de *Ajustes de ubicación de Google*.

Debemos marcar todas las *checkboxs*. Tal y como vemos en la figura 5.2.a. Ello mejora la precisión de nuestra localización, cosa que como sabemos la aplicación utiliza en algunos modos de funcionamiento.

5.2. Iniciar la aplicación

Para iniciar la aplicación debemos buscar el icono correspondiente dentro del menú del dispositivo. Para acceder a este pulsamos la tecla *menú*. Posteriormente desplazarnos por las pantallas donde se encuentran los iconos disponibles. Nuestro icono es el que mostramos en la figura 5.3.a. El usuario busca ese icono y toca en el mismo para acceder a la aplicación por el menú principal.

No obstante, existe un lanzador alternativo, que mostramos en la figura 5.3.b.

Este último nos muestra directamente en pantalla la información de la parada que hemos previamente establecido como favorita. Hemos pensado en los usuarios que tienen una rutina hecha y no desean abrir un mapa cada vez que necesitan consultar cuando vendría su guagua.

Estas son las dos opciones de acceso a nuestro software, en la sección siguiente, empezamos por el menú principal.

5.3. Menú principal

El menú principal de la aplicación, nos muestra los distintos caminos por los que llegar a localizar la parada que buscamos.

La figura 5.4 muestra la pantalla de inicio de la aplicación, con las siguientes opciones:

- *Mostrar mapa*: muestra el mapa completo de la ciudad de Las Palmas de Gran Canaria con *todas* las paradas dibujadas.



(a)



(b)

Figura 5.3: a) Icono del lanzador principal de la aplicación, b) Icono del lanzador de *Ir a parada favorita*.

- *Paradas cercanas a mí*: muestra un mapa centrado en nuestra posición actual. En él están las paradas situadas a 200 metros a la redonda. Usamos esta opción cuando estemos cerca de la parada objetivo. Al entrar en el menú principal esta opción se encuentra deshabilitada. La aplicación intenta obtener nuestra posición. En caso de encontrar una posición precisa, nos ofrece esta opción. En caso contrario podemos usar el botón siguiente.
- *Paradas cercanas a una zona*: muestra un mapa similar al de la primera opción. Solo que en este mapa no hay ninguna parada dibujada. Debemos tocar cualquier punto del mapa. Al tocar el mapa, la aplicación muestra aquellas paradas contenidas en un radio de 200 m a la redonda a la vez que hace zoom sobre el lugar pulsado. Esta operación la podemos repetir cuantas veces queramos.
- *Ir a favorita*: permite establecer una parada como favorita. Con este botón accedemos a ella directamente y nos saltamos el paso intermedio de la visualización de un mapa. Más adelante vemos cómo establecer cual parada es la favorita.

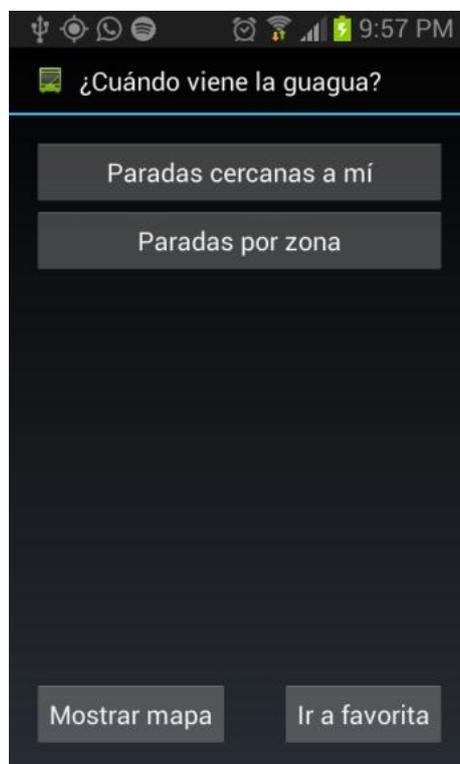


Figura 5.4: Captura del menú principal de la aplicación Android.

Escogemos la opción con la que queremos trabajar, principalmente trabajamos con las dos últimas. Si tenemos GPS activado y nos encontramos cerca de la parada objetivo, pulsamos *Paradas cercanas a mí*, si nos encontramos lejos de la parada objetivo o no disponemos de GPS usamos *Paradas cercanas a una zona*, para posteriormente presionar nosotros la zona en la que creemos que se encuentra situada la parada que queremos ver.

5.4. Opción Mostrar mapa

En este modo, vemos dibujadas todas las paradas de la red desde una vista alejada, haciendo el gesto de pinza con dos dedos podemos alejarnos o acercarnos así como cambiar nuestra posición moviendo un dedo por la pantalla. Presionando el botón  podemos centrarnos en nuestra ubicación si esta estuviera disponible. Recomendamos el uso de otras opciones menos costosas para el terminal y que filtran las paradas, mostrando únicamente las que realmente nos interesan. Como vemos a más adelante. Véanse figuras 5.5 y 5.6.

Una vez tengamos a la vista la para objetivo, damos un toque en su icono para abrir la *InfoWindow* correspondiente. La *InfoWindows* consiste en una ventana emergente que nos muestra la información de la parada que estamos seleccionando.



Figura 5.5: Vista del modo Mostrar mapa.



Figura 5.6: Vista en detalle del modo Mostrar mapa.

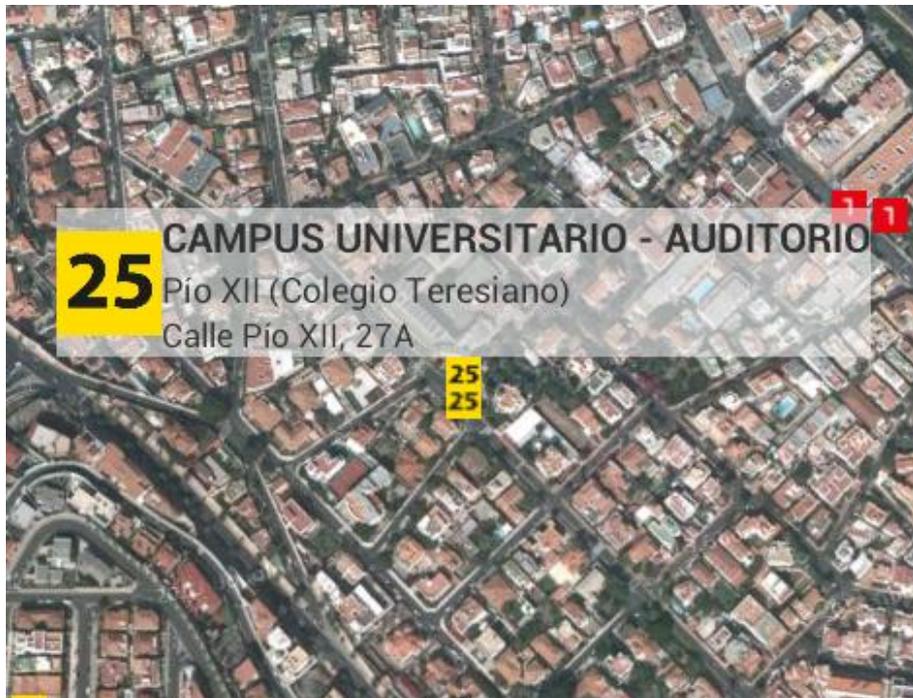


Figura 5.7: Vista de la pequeña ventana desplegable de cada parada o InfoWindow.

En ella vemos información relevante de la parada. Haciendo clic en otro icono, se nos cierra este. Pulsando sobre él pasamos a la pantalla de información de parada.

5.5. Opción Paradas cercanas a mí

Por lo general esta opción suele estar disponible pasados 4 segundos de arranque de nuestra aplicación. Aunque el tiempo varía según las condiciones de red y la posición. Es el tiempo que tarda nuestro dispositivo, valiéndose de los proveedores de localización que tenga a mano en encontrar una localización fiable y precisa (la aplicación exige 50 m de precisión).

Llegamos a esta opción a través del menú principal cuando el botón correspondiente se habilita.

Nos muestra una circunferencia transparente en un tono verde de 200 metros de radio con centro en nuestra posición.

Nuestra aplicación solo dibuja las paradas contenidas en dicha figura. A parte de esto el zoom está ajustado para enfocar el mapa en nuestra zona de interés. Y así identificar rápidamente la parada que estamos buscando. Véase figura 5.9.

Al igual que en el caso anterior, si presionamos cualquier icono de parada, obtenemos una *InfoWindow* emergente que nos da más datos, con el fin de confirmar que se trata de la parada que buscamos.

Dando un toque con el dedo sobre la ventanita emergente por fin conocemos el tiempo de llegada. Tal y como explicamos más adelante.

Dado que en todos los modos de funcionamiento seguimos teniendo la libertad de movernos dentro del mapa, la opción *Paradas cercanas a mí* ofrece la posibilidad de cambiar nuestra zona de foco. Cosa que explicamos en la sección que viene a continuación.

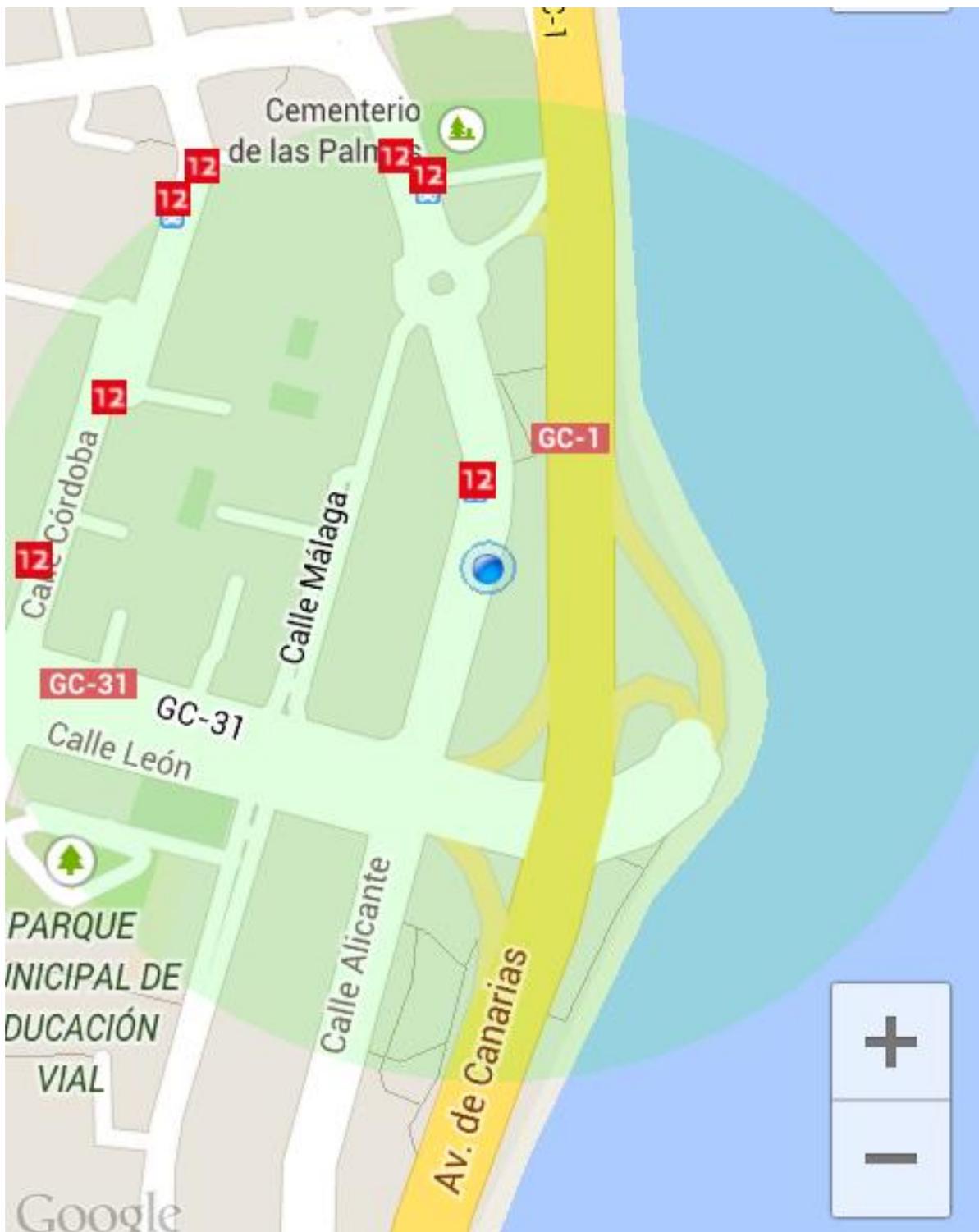


Figura 5.8: Captura del modo paradas cercanas a mí.



Figura 5.9: Modo Paradas cercanas a mí.

5.6. Opción Paradas cercanas a una zona

En esta opción el mapa se nos presenta como en el primer cuadro de la figura 5.11. Obtenemos una vista general de Las Palmas de Gran Canaria.

Usamos este modo, cuando o bien no disponemos de GPS o bien la parada que buscar no está en nuestro área. En esta opción podemos navegar, alejar y acercar el mapa. Aunque en principio no se nos muestra ninguna parada. Es al dar un toque en algún punto del mapa cuando se inicia una animación. Dicha animación nos lleva al punto de toque con un zoom más cercano. Al mismo tiempo, alinea el mapa con el norte arriba, ajusta la inclinación y dibuja un círculo verde transparente, véase figura 5.11. A la vez que esto sucede va dibujando las paradas contenidas en dicho círculo.

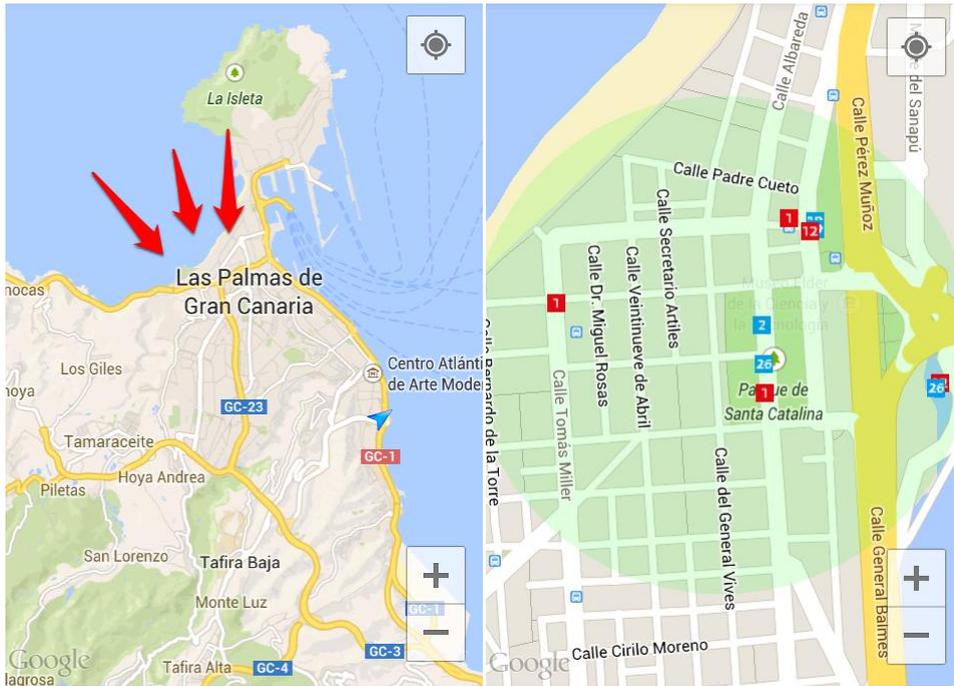


Figura 5.10: Captura del funcionamiento del mapa en modo Paradas cercanas a mí.

A partir de aquí, funciona como en el modo anterior. Podemos dar un toque de dedo a los iconos que nos son mostrados o volver a cambiar la zona de interés.

5.7. Información de parada

Una vez hemos elegido en el modo mapa la parada que queremos, entramos en esta pantalla de la aplicación. En ella se nos vuelve a mostrar la información de la parada, y se lanza una petición al servidor donde se encuentra la información. En un instante se muestra el tiempo previsto de llegada. La cual va refrescándose cada pocos segundos. Como se muestra en la figura 5.12, disponemos de un botón. El botón *Establecer como favorita* ofrece una interesante opción. Ya habíamos hablado de un lanzador alternativo para acceder directamente a la pantalla de información de parada. Presionar el botón es la forma de establecer la parada favorita. De esta forma la aplicación la guarda y cuando volvamos a necesitarla, buscamos el icono  en nuestro menú de aplicaciones. Ello nos ahorra tener que navegar por la aplicación.

En este punto podemos cerrar el móvil, ya que este nos avisa cuando quede poco tiempo para la llegada de la guagua. Lo haría con una notificación.



Figura 5.11: Captura de la ventana de resultado.

6. Conclusiones y posibles ampliaciones

En este capítulo presentamos las principales conclusiones obtenidas después de llevar a cabo el PFC y analizamos posibles ampliaciones.

6.1. Conclusiones

Últimamente vemos producirse grandes avances en cuanto a las prestaciones que los dispositivos móviles nos ofrecen. Algunas soluciones llegan a ser muy ingeniosas por su sencillez y potencia. Ofreciendo una muy satisfactoria experiencia a los usuarios.

En este contexto, la aparición de *Android* ha sido muy importante. Dado su carácter abierto, su amplia documentación y la facilidad que se brinda a los desarrolladores para entrar en él. Hecho que también y como es obvio, beneficia sobre todo al consumidor el cual encuentra un amplio abanico de diferentes opciones para sacar partido a su dispositivo. Que en el caso concreto de *Android* maneja los más punteros y potentes smartphones de gama alta, pero también teléfonos cada vez más asequibles. Pasando por módulos de *smartTV*, tabletas, *phablets*...

En este contexto, este PFC pretende desarrollar una aplicación que acerque una información al pasajero de guaguas de Las Palmas de Gran Canaria para que éste tenga un uso del transporte público más cómodo, flexible y dinámico.

A continuación, se analizan los resultados obtenidos tras la realización de este PFC:

- En primer lugar, se analizan los principales logros alcanzados en función de los objetivos marcados al comienzo de la realización del PFC. En segundo lugar, se han estudiado las posibles ampliaciones que se pueden realizar en el futuro.
- Una vez finalizada la realización del proyecto se puede afirmar que se han cumplido satisfactoriamente todos los objetivos marcados al inicio.
- Hemos conseguido una aplicación rápida, intuitiva y sencilla que da a los usuarios la información que necesita.
- Diseñamos una interfaz gráfica que consigue hacer todo el trabajo con tres toques como máximo. Pensada para consultarse en el trasiego diario. Y la hemos podido testar en un entorno simulado, para el cual hemos implementado nuestro propio servidor de datos.

Por otro lado, hemos obtenido desde cero conocimientos de Java, programación orientada a objetos y de desarrollo en Android utilizando el IDE Eclipse, con Java. Con los cuales nos vemos capaces de enfrentarnos a proyectos más grandes con las posibilidades que nos brinda dicha programación.

6.2. Posibles ampliaciones

Aunque cumplimos con los objetivos que habíamos previsto. Hemos comprobado que cabrían unas cuantas ampliaciones que mejorarían la aplicación. Posibilidades que se bajarían si este proyecto se pone en marcha en un futuro.

Uno o más *widgets* en la pantalla de inicio, mostrando información de ciertas paradas que el usuario configure.

Posibilidad de activación del tiempo en el que la aplicación considera que queda poco. Para guardarlo dentro de las *SharedPreferences*.

Notificaciones de aviso basadas en un alimentador de noticias especial de la compañía de transportes.

Pr. Pliego de condiciones y presupuesto

En este capítulo se enumera el trabajo realizado y el software necesario para mantener la funcionalidad de la aplicación. Además, se presenta un presupuesto detallado de los costes para la realización del proyecto.

Pr.1 Pliego de condiciones

En este apartado se indican los requisitos mínimos necesarios para poder ejecutar la aplicación desarrollada en este proyecto.

Pr.1.1 Descripción del proyecto

En este proyecto se ha desarrollado un sistema de localización de productos. Las tareas a llevar a cabo para la realización del mismo se desglosan a continuación:

- Adquisición de conocimientos sobre Java.
- Adquisición de conocimientos sobre desarrollo en Android.
- Adquisición de conocimientos servicios web.
- Adquisición de conocimientos sobre servidores Apache.
- Desarrollo del módulo servicio Web.
- Desarrollo del módulo del cliente móvil.
- Documentación del proyecto.

Pr.1.2 Requisitos de hardware

Para el correcto funcionamiento del sistema se especifican los requisitos mínimos a nivel de hardware y software, que debe tener tanto el servidor como los clientes.

Requisitos hardware del servidor

El servidor debe ser un equipo con al menos las siguientes características:

- PC compatible con procesador *PENTIUM 4 2.4Ghz*.
- 1GB de memoria *RAM*.
- 100MB de espacio libre en el disco duro.
- Conexión a Internet.

El sistema operativo puede ser cualquier Linux o cualquier Windows a partir de Windows Xp.

Requisitos hardware de la aplicación móvil

Para que la aplicación funcione se necesita lo siguiente:

- Sistema Operativo Android 4.0.1 Ice Cream Sandwich o superior.
- 4 megas de espacio libre en la memoria interna.
- Algún tipo de conexión a internet.
- No obstante, recomendamos usar la aplicación en un dispositivo que cumpla las siguientes características.
- Sistema Operativo *Android 4.1.2 Jelly Bean* o superior.
- 10 megas de espacio libre en la memoria interna.
- Conexión internet móvil.
- Conexión *Wireless Fidelity (WiFi)*.
- Sistema de GPS.

Otros elementos técnicos a tener en cuenta

Nuestro software funciona de forma correcta y no se ha detectado anomalías importantes. Sin embargo, es importante poner en aviso al usuario de los siguientes detalles sobre su funcionamiento:

- Si no se nos han mostrado los minutos en la pantalla de visualización de información de parada después de un tiempo. Es mejor volver a atrás y repetir la operación. Ello se debe a que para evitar bucles, la aplicación intenta obtener la información un número limitado de veces.
- Una vez hemos obtenido una notificación de llegada de guagua, es mejor cerrar totalmente la app antes de volver a usarla.

Pr.2 Presupuesto

En este apartado se presenta el presupuesto de elaboración del proyecto que consta de dos partes: presupuesto de la mano de obra y presupuesto del material utilizado.

Para calcular el coste de la mano de obra, se ha evaluado en función del tiempo empleado en cada fase de desarrollo del proyecto. A continuación se detallan estas fases:

- Información.
- Definición.
- Análisis.
- Implementación.
- Pruebas.
- Redacción de la memoria.

El coste de cada una de estas fases se calcula en función del número de horas empleadas en el proyecto, utilizando los baremos del *Colegio Oficial de Ingenieros Técnicos de Telecomunicación (COITT)*.

Pr.2.1 Descripción de las fases del proyecto

En este punto se describen las etapas en las que se divide el proyecto definiendo el trabajo realizado en cada una de ellas.

Información

En esta etapa investigamos, consultamos y en definitiva recogemos información con la que llevar a cabo nuestro proyecto. Parte de ello consiste en comparar diversas plataformas para solucionar los diferentes retos de cada bloque funcional. También el aprendizaje a distinto grado de lenguajes de programación como puede ser el lenguaje *Java*, el *php*. Y así como el manejo de herramientas como son el Eclipse con el plugin de para el desarrollo en Android...

Definición

En este punto se definen los objetivos del proyecto y cómo afrontar su ejecución dividiendo el proceso en partes lo más simples posible. Esto proporciona una mayor rapidez en la búsqueda de soluciones y en realizar una implementación con errores mínimos.

Análisis

Este apartado consiste en estudiar el problema a resolver. En la fase de análisis se tiene que poner especial interés debido a que se realiza el proyecto basándose en las pautas definidas en esta etapa. Para ello se debe dedicar todo el tiempo que sea necesario.

Es imprescindible partir de unas especificaciones lo más exactas y detalladas posibles. El resultado debe ser un modelo preciso del entorno del problema, de los datos y de los objetivos propuestos al inicio del proyecto.

Implementación

Este proceso consiste, principalmente, en implementar en el lenguaje de programación elegido, todo lo diseñado en la fase anterior. Es común pensar que la fase de implementación es la que consume la mayor parte del trabajo de desarrollo del software; sin embargo, esto es relativo ya que las etapas previas son cruciales, críticas y pueden llevar bastante más tiempo.

Una vez que se ha desarrollado cada módulo, se realizan las pertinentes pruebas y se depuran los posibles errores.

Pruebas

En este punto se realiza una comprobación global de la aplicación buscando posibles errores o conflictos entre los diferentes módulos. En el caso de existir, éstos se solucionan y se vuelve a validar. Este proceso se repite hasta que se considera que el sistema es válido.

Redacción de la memoria

En esta fase se elabora el presente documento en el que se describe paso a paso todo el proceso de desarrollo del proyecto. Además, se realiza un manual de usuario con toda la información necesaria para el uso de la aplicación y se desarrolla un presupuesto de los costes necesarios que supone llevar a cabo el proyecto.

Pr.2.2 Mano de obra

En este apartado, se realiza el presupuesto en función del número de horas dedicado a cada fase. Tomando como media 4 horas al día, 5 días a la semana y 4 semanas al mes, se obtiene una duración total del proyecto de 12 meses.

En la Tabla Pr.1 se muestra el número de horas invertidas en cada fase del proyecto. Los honorarios calculados por dichas horas de trabajo se calculan en base a la fórmula establecida por el *COITT* del año 2003 cuyo significado es el siguiente:

$$\mathbf{H= 78 \cdot He + 65 \cdot Hn.}$$

Donde:

He= Horas especiales · C.

Hn= Horas normales · C.

C= Coeficiente reductor.

Utilizando como coeficiente reductor $C=0,4$, los honorarios por mano de obra son los siguientes:

$$\mathbf{H= 65 \cdot 960 \cdot 0,4 = 24960 \text{ euros.}}$$

Pr.2.3 Coste del material

En este apartado se describen los costes del material utilizado durante el desarrollo de este proyecto.

En la Tabla Pr.2 se muestra el desglose de los costes de los materiales utilizados.

Fases	Horas
Información	240
Definición	80
Análisis	80
Implementación	320
Pruebas	80
Redacción de memoria	160
TOTAL	960

Tabla Pr.1: Número de horas trabajadas.

Concepto	Tiempo amortización	Importe (€)
PC portátil + S.O.	12 meses	800,00 €
Samsung Galaxy S3 Mini	18 meses	280,00 €
TOTAL		1.080,00 €

Tabla Pr.2: Coste de los materiales.

Concepto	Coste (€)
Mano de obra	24.960,00 €
Coste del material	1.080,00 €
TOTAL	26.040,00 €

Tabla Pr.3: Coste total del proyecto.

El plazo de amortización del coste de los materiales utilizados durante el desarrollo del proyecto se estima en 12 meses, ya que es la duración de elaboración del mismo.

Pr.2.4 Coste total del proyecto

En la Tabla Pr.3 se muestra el coste total del proyecto dividido por mano de obra y material utilizado.

El presupuesto total de este proyecto es de VEINTISEIS MIL CUARENTA euros.

Anexo A: Datos de paradas

Id	Línea Num	sentido	Num Parada	Cod Línea	lineas	title	LatLng	snippet
1001	1	0	1	10	TEATRO - PUERTO	Teatro.	28.104409,- 15.413878	Av. Rafael Cabrera, 4
1002	1	0	2	10	TEATRO - PUERTO	Rafael Cabrera (San Telmo)	28.10914,- 15.416084	Av. Rafael Cabrera, 30
1003	1	0	3	10	TEATRO - PUERTO	Venegas (Usos Múltiples)	28.111689,- 15.418132	Calle Venegas, 54
1004	1	0	4	10	TEATRO - PUERTO	Venegas (Fuente Luminosa)	28.114254,- 15.419264	Calle Archivero Municipal Pedro Cullen del Castillo, 49
1005	1	0	5	10	TEATRO - PUERTO	Luis Doreste Silva, 22	28.116158,- 15.421211	Calle Luis Doreste Silva, 6-8
1006	1	0	6	10	TEATRO - PUERTO	Luis Doreste Silva, 50	28.118893,- 15.423591	Calle Luis Doreste Silva, 52-54
1007	1	0	7	10	TEATRO - PUERTO	León y Castillo (C.N. Metropole)	28.122059,- 15.42661	Calle León y Castillo, 227
1008	1	0	8	10	TEATRO - PUERTO	León y Castillo (Oficinas Municipales)	28.125433,-15.429	Calle León y Castillo, 274
1009	1	0	9	10	TEATRO - PUERTO	León y Castillo (Torre de Las Palmas)	28.128809,- 15.429806	Calle León y Castillo, 295A
1010	1	0	10	10	TEATRO - PUERTO	León y Castillo (Club Náutico)	28.132898,- 15.429523	Calle León y Castillo, 363
1011	1	0	11	10	TEATRO - PUERTO	Pl. San Juan Bautista (Base Naval)	28.135621,- 15.429001	Calle General Balmes
1012	1	0	12	10	TEATRO - PUERTO	Juan Manuel Durán González, 20	28.135456,- 15.4327632	Calle Juan Manuel Durán González, 22
1013	1	0	13	10	TEATRO - PUERTO	Tomás Miller, frente 31	28.137497,- 15.433434	Calle Tomás Miller, 22
1014	1	0	14	10	TEATRO - PUERTO	Tomás Miller (Playa de Las Canteras)	28.141092,- 15.433718	Calle Tomás Miller, 70
1015	1	0	15	10	TEATRO - PUERTO	Alfredo L. Jones, 1	28.142163,- 15.430423	Calle Alfredo L Jones, 1-3
1016	1	0	16	10	TEATRO - PUERTO	Eduardo Benot, frente 23	28.143825,- 15.429793	Calle Eduardo Benot, 19
1017	1	0	17	10	TEATRO - PUERTO	Agustín Millares Sall (Edificio Mapfre)	28.147872,- 15.427967	Calle Poeta Agustin Millares Sall
1018	1	0	18	10	TEATRO - PUERTO	Juan Rejón (Castillo de La Luz)	28.149171,- 15.425367	Calle Juan Rejón, 73
1019	1	0	19	10	TEATRO - PUERTO	Manuel Becerra (Puerto)	28.150375,- 15.422071	Calle la Naval, 203
1101	1	1	1	11	PUERTO - TEATRO	Manuel Becerra (Puerto)	28.150396,- 15.422149	Calle la Naval, 201
1102	1	1	2	11	PUERTO - TEATRO	Juan Rejón, 67	28.149154,- 15.425972	Calle Juan Rejón, 63
1103	1	1	3	11	PUERTO - TEATRO	Albareda (Mercado del Puerto)	28.148608,- 15.428989	Calle Juan Rejón, 2-4
1104	1	1	4	11	PUERTO - TEATRO	Albareda, 43	28.144897,- 15.430324	Calle Albareda, 43
1105	1	1	5	11	PUERTO - TEATRO	Parque Santa Catalina	28.139964,- 15.430761	Calle León y Castillo, 439
1106	1	1	6	11	PUERTO - TEATRO	Presidente Alvear, 17	28.13508,- 15.430341	Calle Presidente Alvear, 17-19
1107	1	1	7	11	PUERTO - TEATRO	León y Castillo (Edificio José Antonio)	28.132614,- 15.430047	Calle León y Castillo, 361

1108	1	1	8	11	PUERTO - TEATRO	León y Castillo (Oficinas Municipales)	28.125544,- 15.429258	Calle León y Castillo, 261
1109	1	1	9	11	PUERTO - TEATRO	León y Castillo (Colegio Salesiano)	28.122895,- 15.427527	Calle León y Castillo, 229
1110	1	1	10	11	PUERTO - TEATRO	León y Castillo, 209	28.12004,- 15.425596	Calle León y Castillo, 211
1111	1	1	11	11	PUERTO - TEATRO	León y Castillo, 185	28.117899,- 15.423799	Calle León y Castillo, 183
1112	1	1	12	11	PUERTO - TEATRO	León y Castillo, 145	28.116556,- 15.422596	Calle León y Castillo, 149
1113	1	1	13	11	PUERTO - TEATRO	León y Castillo (Pl. de La Feria)	28.113406,- 15.419977	Calle León y Castillo, 79
1114	1	1	14	11	PUERTO - TEATRO	León y Castillo, 13	28.109941,- 15.418244	Calle León y Castillo, 15
1115	1	1	15	11	PUERTO - TEATRO	Triana (Parque San Telmo)	28.108268,- 15.417443	Calle Triana, 109
1116	1	1	16	11	PUERTO - TEATRO	Munguía, 8	28.106269,- 15.415236	Calle Munguía, 10
1117	1	1	17	11	PUERTO - TEATRO	Teatro	28.103804,- 15.41372	Av Rafael Cabrera
2001	2	0	1	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Guiniguada	28.10075,-15.419	Calle Bernardino Correa Viera
2002	2	0	2	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Alameda de Colón	28.1030638,- 15.4171632	Calle Doctor Domingo Déniz, 13
2003	2	0	3	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Primero de Mayo (Correos)	28.106143,- 15.420125	Av Primero de Mayo, 62
2004	2	0	4	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Ingeniero J. Bosch y Sintes, 1	28.108483,- 15.421034	Calle Senador Castillo Olivares, 1
2005	2	0	5	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Pl. de la Constitución (Obelisco)	28.112301,- 15.423372	Calle Alfonso XIII, 7
2006	2	0	6	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Tomás Morales, frente 69	28.115001,- 15.425162	Paseo Tomás Morales, 94
2007	2	0	7	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Tomás Morales, 120	28.116932,- 15.426284	Paseo Tomás Morales, 118
2008	2	0	8	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Emilio Ley (Piscinas Julio Navarro)	28.121336,- 15.429834	Calle Emilio Ley, 4
2009	2	0	9	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Pío XII (Colegio Teresiano)	28.125041,- 15.432122	Calle Pío XII, 27A
2010	2	0	10	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Pío XII (Estadio Insular)	28.128925,- 15.432851	Calle Pío XII, 62
2011	2	0	11	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Galicia (Mercado Central)	28.131953,- 15.432921	Calle Pio XII, 81
2012	2	0	12	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Galicia, 32	28.135192,- 15.433256	Calle Galicia, 32
2013	2	0	13	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Tomás Miller, frente 31	28.137497,- 15.433434	Calle Tomás Miller, 22
2014	2	0	14	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Tomás Miller (Playa de Las Canteras)	28.141092,- 15.433718	Calle Tomás Miller, 70
2015	2	0	15	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Alfredo L. Jones, 1	28.142163,- 15.430423	Calle Alfredo L Jones, 1-3
2016	2	0	16	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Eduardo Benot, frente 23	28.143825,- 15.429793	Calle Eduardo Benot, 19

2017	2	0	17	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Agustín Millares Sall (Edificio Mapfre)	28.147872,- 15.427967	Calle Poeta Agustín Millares Sall
2018	2	0	18	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Juan Rejón (Castillo de La Luz)	28.149171,- 15.425367	Calle Juan Rejón, 73
2019	2	0	19	20	GUINIGUADA - PUERTO (POR TOMAS MORALES)	Manuel Becerra (Puerto)	28.15024,-15.42191	Calle la Naval, 205
2101	2	1	1	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Manuel Becerra (Puerto)	28.150396,- 15.422149	Calle la Naval, 201
2102	2	1	2	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Juan Rejón, 67	28.149154,- 15.425972	Calle Juan Rejón, 63
2103	2	1	3	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Albareda (Mercado del Puerto)	28.148608,- 15.428989	Calle Juan Rejón, 2-4
2104	2	1	4	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Albareda, 43	28.144897,- 15.430324	Calle Albareda, 43
2105	2	1	5	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Parque Santa Catalina	28.140812,- 15.430811	Parque Santa Catalina
2106	2	1	6	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Presidente Alvear (Iglesia del Pino)	28.135636,- 15.430434	Calle Juan Manuel Durán González, 8-10
2107	2	1	7	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Mesa y López, 38	28.134783,- 15.433043	Calle José Mesa López, 40
2108	2	1	8	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Galicia (Mercado Central)	28.1333778,- 15.433161	Calle Galicia, 24
2109	2	1	9	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Galicia	28.131203,- 15.4329634	Calle Pío XII, 75
2110	2	1	10	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Pío XII (Estadio Insular)	28.128104,- 15.433072	Calle Antonio Zerolo, 2
2111	2	1	11	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Pío XII (Colegio Teresiano)	28.124881,- 15.43217	Calle Pío XII, 27A
2112	2	1	12	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Emilio Ley (Parque Doramas)	28.120219,- 15.429505	Calle Emilio Ley, 9
2113	2	1	13	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Pérez del Toro (Pl. Dr. Juan Bosch Millares)	28.117206,- 15.427311	Calle Pérez del Toro, 89
2114	2	1	14	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Pérez del Toro, 13	28.113648,- 15.425118	Calle Pérez del Toro, 11-13
2115	2	1	15	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Pl. de la Constitución (Obelisco)	28.111873,- 15.42374	Plaza de la Constitución, 5-6
2116	2	1	16	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Tomás Morales, 5	28.108348,- 15.420749	Paseo Tomás Morales, 7
2117	2	1	17	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Primero de Mayo, 35	28.105602,- 15.41994	Av. Primero de Mayo
2118	2	1	18	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Primero de Mayo, 1	28.1023833,- 15.4175833	Av. Primero de Mayo, 4
2119	2	1	19	21	PUERTO - GUINIGUADA (POR TOMAS MORALES)	Guiniguada	28.1005667,- 15.4191167	Calle Bernardino Correa Viera
6001	6	0	1	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Hoya de La Plata	28.070003,- 15.418131	Calle Hoya de la Plata, 2F

6002	6	0	2	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Párroco José C. Quintana, 8	28.068169,- 15.417786	Calle Párroco José C Quintana S, 65
6003	6	0	3	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Párroco José C. Quintana, 30	28.066982,- 15.418597	Calle Párroco José C Quintana S, 26
6004	6	0	4	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Párroco José C. Quintana, 46	28.066479,- 15.419741	Calle Párroco José C Quintana S, 32
6005	6	0	5	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Marianao, 12	28.063778,- 15.422441	Calle Marianao, 16-18
6006	6	0	6	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Veracruz, 16	28.06314,- 15.424441	Calle Veracruz, 16
6007	6	0	7	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Veracruz, 36	28.062454,- 15.426603	Calle Veracruz, 51
6008	6	0	8	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Marianao, 60	28.061437,- 15.428892	Calle Marianao, 58
6009	6	0	9	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Monterrey, 1	28.061035,- 15.430507	Calle Potosi, 1
6010	6	0	10	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Guantánamo, frente 121	28.060191,- 15.42848	Calle Guantánamo, 92
6011	6	0	11	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Guantánamo, 108	28.059558,- 15.430479	Calle Guantánamo, 86-90
6012	6	0	12	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Guantánamo (Centro P. Salto del Negro)	28.059258,- 15.432256	Calle Guantánamo, 86-90
6013	6	0	13	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Tablero de Gonzalo	28.057781,- 15.439606	Calle Isla de la Graciosa, 92-122
6014	6	0	14	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Isla de La Graciosa, frente 72	28.057408,- 15.443136	Calle Isla de la Graciosa, 70
6015	6	0	15	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Isla de La Graciosa, frente 42	28.05669,- 15.444761	Calle Isla de la Graciosa, 51
6016	6	0	16	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Isla de La Graciosa, 41	28.055975,- 15.445925	Calle Isla de la Graciosa, 41
6017	6	0	17	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Lomo del Capón	28.05468,-15.44743	Lugar Lomo del Capon
6018	6	0	18	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Isla de La Montaña Clara, frente 5	28.055263,- 15.446715	Calle Isla de Montaña Clara, 6
6019	6	0	19	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Arquitecto José Luís Jiménez, 3	28.056422,- 15.448332	Montañeta de Tafira, 1
6020	6	0	20	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Arquitecto Laureano Arroyo, 80	28.058341,- 15.449411	Calle Arquitecto Laureano Arroyo, 78
6021	6	0	21	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Arquitecto Laureano Arroyo, 73	28.057639,- 15.450729	Calle Arquitecto Laureano Arroyo, 48-50
6022	6	0	22	60	HOYA DE LA PLATA - SAN FRANCISCO DE PAULA	Arquitecto Laureano Arroyo (San Francisco de Paula)	28.056679,- 15.451378	Calle Arquitecto López Echegarreta, 17
6101	6	1	1	61	SAN FRANCISCO DE PAULA - HOYA DE LA PLATA	Arquitecto Laureano Arroyo (San Francisco de Paula)	28.056679,- 15.451378	Calle Arquitecto López Echegarreta, 17
6102	6	1	2	61	SAN FRANCISCO DE PAULA - HOYA DE	Arquitecto Laureano Arroyo,	28.057625,- 15.450642	Calle Arquitecto Laureano Arroyo, 48

					LA PLATA	64		
6103	6	1	3	61	SAN FRANCISCO DE PAULA - HOYA DE LA PLATA	Arquitecto Laureano Arroyo, 102 B	28.058998,- 15.447172	Calle Arquitecto Laureano Arroyo, 104
6104	6	1	4	61	SAN FRANCISCO DE PAULA - HOYA DE LA PLATA	Miguel Martín Fernández de La Torre, 7	28.057813,- 15.444879	Calle Miguel Martin Fernandez De La Torre, 7
6105	6	1	5	61	SAN FRANCISCO DE PAULA - HOYA DE LA PLATA	Miguel Martín Fernández de La Torre (Centro de Menores)	28.057854,- 15.441827	Calle Isla de la Graciosa, 90
6106	6	1	6	61	SAN FRANCISCO DE PAULA - HOYA DE LA PLATA	Tablero de Gonzalo, frente 7	28.057632,- 15.43929	Calle Isla de la Graciosa, 92-122
6107	6	1	7	61	SAN FRANCISCO DE PAULA - HOYA DE LA PLATA	Guantánamo, 215	28.059183,- 15.432206	Calle Guantánamo, 215
6108	6	1	8	61	SAN FRANCISCO DE PAULA - HOYA DE LA PLATA	Guantánamo, 165	28.059521,- 15.430304	Calle Guantánamo, 171
6109	6	1	9	61	SAN FRANCISCO DE PAULA - HOYA DE LA PLATA	Guantánamo, 115	28.060229,- 15.428147	Calle Guantánamo, 129
6110	6	1	10	61	SAN FRANCISCO DE PAULA - HOYA DE LA PLATA	Marianao, frente 82	28.06095,- 15.429669	Calle Marianao, 80
6111	6	1	11	61	SAN FRANCISCO DE PAULA - HOYA DE LA PLATA	Marianao, 103	28.06148,- 15.427917	Calle de Mariano, 99
6112	6	1	12	61	SAN FRANCISCO DE PAULA - HOYA DE LA PLATA	Marianao, 53	28.062571,- 15.424149	Calle de Mariano, 35
6113	6	1	13	61	SAN FRANCISCO DE PAULA - HOYA DE LA PLATA	Marianao, 5	28.063365,- 15.422578	Calle Marianao, 33-37
6114	6	1	14	61	SAN FRANCISCO DE PAULA - HOYA DE LA PLATA	Párroco José C. Quintana, 41	28.066574,- 15.41932	Calle Párroco José C Quintana S, 33-35
6115	6	1	15	61	SAN FRANCISCO DE PAULA - HOYA DE LA PLATA	Párroco José C. Quintana, 15	28.067179,- 15.418124	Calle Párroco José C Quintana S, 7
6116	6	1	16	61	SAN FRANCISCO DE PAULA - HOYA DE LA PLATA	Párroco José C. Quintana, frente 8	28.068356,- 15.417744	Calle Párroco José C Quintana S, 69A
6117	6	1	17	61	SAN FRANCISCO DE PAULA - HOYA DE LA PLATA	Hoya de La Plata	28.070003,- 15.418131	Calle Hoya de la Plata, 2F
7001	7	0	1	70	TEATRO - CAMPUS UNIVERSITARIO	Teatro	28.104377,- 15.41383	Av Rafael Cabrera, 4
7002	7	0	2	70	TEATRO - CAMPUS UNIVERSITARIO	Lentini (Monopol)	28.102325,- 15.415268	Calle Lentini
7003	7	0	3	70	TEATRO - CAMPUS UNIVERSITARIO	Ctra. del Centro (Antigua Cárcel)	28.09343,- 15.425415	Carretera del Centro
7004	7	0	4	70	TEATRO - CAMPUS UNIVERSITARIO	Ctra. del Centro (Colegio S. Juan Bosco)	28.097665,- 15.422628	Carretera del Centro
7005	7	0	5	70	TEATRO - CAMPUS UNIVERSITARIO	Real de S. Roque, frente 11	28.098249,- 15.421711	Calle Real de San Roque, 13
7006	7	0	6	70	TEATRO - CAMPUS UNIVERSITARIO	Farnesio, frente 15	28.098419,- 15.424381	Calle Farnesio, 11-13
7007	7	0	7	70	TEATRO - CAMPUS UNIVERSITARIO	Farnesio, frente 59	28.099565,- 15.425914	Calle Farnesio, 57
7008	7	0	8	70	TEATRO - CAMPUS UNIVERSITARIO	Fama, frente 85	28.099234,- 15.426504	Calle Fama, 85
7009	7	0	9	70	TEATRO - CAMPUS UNIVERSITARIO	Andenes, frente 15	28.098588,- 15.428395	Calle Andenes, 11
7010	7	0	10	70	TEATRO - CAMPUS UNIVERSITARIO	Andenes, frente 39	28.098295,- 15.429819	Calle Andenes, 37
7011	7	0	11	70	TEATRO - CAMPUS	Nueva, frente 25	28.097346,-	Calle Nueva, 27

					UNIVERSITARIO		15.430817	
7012	7	0	12	70	TEATRO - CAMPUS UNIVERSITARIO	Nueva, frente 67	28.095837,- 15.431752	GC-112
7013	7	0	13	70	TEATRO - CAMPUS UNIVERSITARIO	Barahona, 77	28.094613,- 15.433187	GC-112
7014	7	0	14	70	TEATRO - CAMPUS UNIVERSITARIO	Barahona, frente 19	28.09196,- 15.434075	GC-112
7015	7	0	15	70	TEATRO - CAMPUS UNIVERSITARIO	La Matula, frente 25	28.089778,- 15.435745	GC-112
7016	7	0	16	70	TEATRO - CAMPUS UNIVERSITARIO	La Matula, frente 61	28.087858,- 15.437024	Calle la Matula, 32-34
7017	7	0	17	70	TEATRO - CAMPUS UNIVERSITARIO	La Matula, frente 95	28.08559,- 15.438838	Calle la Matula, 19
7018	7	0	18	70	TEATRO - CAMPUS UNIVERSITARIO	Puente La Matula	28.082737,- 15.441266	GC-112
7019	7	0	19	70	TEATRO - CAMPUS UNIVERSITARIO	Ctra. Lomo Blanco (Hermanos Toledo Suárez)	28.084794,- 15.441123	GC-112
7020	7	0	20	70	TEATRO - CAMPUS UNIVERSITARIO	Ctra. Lomo Blanco (Lomo Verdejo)	28.087842,- 15.43922	GC-112
7021	7	0	21	70	TEATRO - CAMPUS UNIVERSITARIO	Ctra. Lomo Blanco (Felipe Martel)	28.084023,- 15.442834	Calle Felipe Martel, 12
7022	7	0	22	70	TEATRO - CAMPUS UNIVERSITARIO	Felipe Martel (Lindero)	28.08552,- 15.442247	Calle Felipe Martel, 2
7023	7	0	23	70	TEATRO - CAMPUS UNIVERSITARIO	San José Artesano, 24	28.084876,- 15.443285	Calle San José Artesano, 24
7024	7	0	24	70	TEATRO - CAMPUS UNIVERSITARIO	San José Artesano, 46	28.083667,- 15.444719	Calle San José Artesano, 48
7025	7	0	25	70	TEATRO - CAMPUS UNIVERSITARIO	Ctra. Lomo Blanco, (Puente Aéreo)	28.0840166666667, -15.4456166666667	GC-112
7026	7	0	26	70	TEATRO - CAMPUS UNIVERSITARIO	Ctra. Lomo Blanco (IES Felo Monzón)	28.081858,- 15.447164	GC-112
7027	7	0	27	70	TEATRO - CAMPUS UNIVERSITARIO	Ctra. Lomo Blanco, 54	28.079941,- 15.447413	GC-112, 50-58
7028	7	0	28	70	TEATRO - CAMPUS UNIVERSITARIO	Ctra. Lomo Blanco, 94	28.07805,- 15.448313	Carretera Lomo Blanco, 88
7029	7	0	29	70	TEATRO - CAMPUS UNIVERSITARIO	Ctra. Lomo Blanco (Urb. El Zurbarán)	28.076102,- 15.449683	Carretera Lomo Blanco, 113
7030	7	0	30	70	TEATRO - CAMPUS UNIVERSITARIO	Ciencias Básicas / Informática	28.07282,- 15.451986	Facultad de Informática
7031	7	0	31	70	TEATRO - CAMPUS UNIVERSITARIO	Campus Universitario (Ingenierías)	28.0708666666667, -15.4555666666667	Calle la Naval
7101	7	1	1	71	CAMPUS UNIVERSITARIO - TEATRO	Campus Universitario (Ingenierías)	28.0708666666667, -15.4555666666667	Calle la Naval
7102	7	1	2	71	CAMPUS UNIVERSITARIO - TEATRO	Ciencias Básicas / Teología	28.072523,- 15.451691	Carretera Lomo Blanco, 137
7103	7	1	3	71	CAMPUS UNIVERSITARIO - TEATRO	Ctra. Lomo Blanco (Urb. El Zurbarán)	28.075806,- 15.449603	Carretera Lomo Blanco, 117
7104	7	1	4	71	CAMPUS UNIVERSITARIO - TEATRO	Ctra. Lomo Blanco, 81	28.079779,- 15.447262	Carretera Lomo Blanco, 85
7105	7	1	5	71	CAMPUS UNIVERSITARIO - TEATRO	Ctra. Lomo Blanco, frente 51	28.081874,- 15.447016	GC-112
7106	7	1	6	71	CAMPUS UNIVERSITARIO - TEATRO	Carretera del Lomo Blanco, 5	28.08322,-15.44370	Calle Practicante Calixto Rodríguez, 17
7107	7	1	7	71	CAMPUS UNIVERSITARIO -	Felipe Martel (Lindero)	28.08552,- 15.442247	Calle Felipe Martel, 2

					TEATRO			
7108	7	1	8	71	CAMPUS UNIVERSITARIO - TEATRO	San José Artesano, 24	28.084876,- 15.443285	Calle San José Artesano, 24
7109	7	1	9	71	CAMPUS UNIVERSITARIO - TEATRO	San José Artesano, 46	28.083667,- 15.444719	Calle San José Artesano, 48
7110	7	1	10	71	CAMPUS UNIVERSITARIO - TEATRO	Ctra. Lomo Blanco (Felipe Martel)	28.083994,- 15.442648	GC-112
7111	7	1	11	71	CAMPUS UNIVERSITARIO - TEATRO	Ctra. Lomo Blanco (El Hornillo)	28.087828,- 15.439376	GC-112
7112	7	1	12	71	CAMPUS UNIVERSITARIO - TEATRO	Ctra. Lomo Blanco (Hermanos Toledo Suárez)	28.084655,- 15.441175	GC-112
7113	7	1	13	71	CAMPUS UNIVERSITARIO - TEATRO	Ctra. Lomo Blanco (Puente La Matula)	28.082502,- 15.442044	GC-112
7114	7	1	14	71	CAMPUS UNIVERSITARIO - TEATRO	La Matula, 97	28.085516,- 15.438801	Calle la Matula, 18
7115	7	1	15	71	CAMPUS UNIVERSITARIO - TEATRO	La Matula, 65	28.087472,- 15.437067	Calle la Matula, 32-34
7116	7	1	16	71	CAMPUS UNIVERSITARIO - TEATRO	La Matula, 25	28.08958,- 15.435729	Ladera de Matula, 19
7117	7	1	17	71	CAMPUS UNIVERSITARIO - TEATRO	Barahona, 19	28.092007,- 15.433814	GC-112
7118	7	1	18	71	CAMPUS UNIVERSITARIO - TEATRO	Nueva, 77	28.093892,- 15.433512	GC-112
7119	7	1	19	71	CAMPUS UNIVERSITARIO - TEATRO	Nueva, 73	28.095672,- 15.43177	GC-112
7120	7	1	20	71	CAMPUS UNIVERSITARIO - TEATRO	Nueva, 35	28.096963,- 15.430798	Calle Nueva, 37
7121	7	1	21	71	CAMPUS UNIVERSITARIO - TEATRO	Andenes, 19	28.098327,- 15.429269	Calle Andenes, 21
7122	7	1	22	71	CAMPUS UNIVERSITARIO - TEATRO	Andenes, 9	28.098452,- 15.428015	Calle Belingo, 26
7123	7	1	23	71	CAMPUS UNIVERSITARIO - TEATRO	Fama, 71	28.098873,- 15.426134	Calle Fama, 71
7124	7	1	24	71	CAMPUS UNIVERSITARIO - TEATRO	Farnesio, 53	28.099458,- 15.425872	Calle Farnesio, 55
7125	7	1	25	71	CAMPUS UNIVERSITARIO - TEATRO	Farnesio, 13	28.098216,- 15.424184	Calle Farnesio, 9
7126	7	1	26	71	CAMPUS UNIVERSITARIO - TEATRO	Real de S. Roque, 13	28.098135,- 15.421658	Calle Real de San Roque, 13D
7127	7	1	27	71	CAMPUS UNIVERSITARIO - TEATRO	Juan de Quesada (Rectorado ULPGC)	28.098981,- 15.420548	Calle Juan de Quesada, 31
7128	7	1	28	71	CAMPUS UNIVERSITARIO - TEATRO	Ctra. del Centro (Teatro Guinguada)	28.102166,- 15.414957	C-811
7129	7	1	29	71	CAMPUS UNIVERSITARIO - TEATRO	Teatro	28.103993,- 15.413801	Av Rafael Cabrera, 4
1200	12	0	1	120	PUERTO - HOYA DE	Manuel Becerra	28.149954,-	Calle Juan Rejón, 62

1					LA PLATA	(Puerto)	15.421738	
1200	12	0	2	120	PUERTO - HOYA DE LA PLATA	Juan Rejón, 67	28.149154,- 15.425972	Calle Juan Rejón, 63
1200	12	0	3	120	PUERTO - HOYA DE LA PLATA	Albareda (Mercado del Puerto)	28.148147,- 15.429239	Calle Albareda, 82
1200	12	0	4	120	PUERTO - HOYA DE LA PLATA	Albareda, 43	28.144897,- 15.430324	Calle Albareda, 43
1200	12	0	5	120	PUERTO - HOYA DE LA PLATA	Parque Santa Catalina	28.140338,- 15.430777	Calle León y Castillo, 439
1200	12	0	6	120	PUERTO - HOYA DE LA PLATA	Presidente Alvear, 17	28.13508,- 15.430341	Calle Presidente Alvear, 17-19
1200	12	0	7	120	PUERTO - HOYA DE LA PLATA	León y Castillo (Edificio José Antonio)	28.132614,- 15.430047	Calle León y Castillo, 361
1200	12	0	8	120	PUERTO - HOYA DE LA PLATA	Alc. J. Ramírez Bethencourt (Parque Romano)	28.126104,- 15.428078	Av Alcalde José Ramírez Bethencourt, 129
1200	12	0	9	120	PUERTO - HOYA DE LA PLATA	Alc. J. Ramírez Bethencourt (Carvajal)	28.118532,- 15.421208	Av Alcalde José Ramírez Bethencourt, 13-21
1201	12	0	10	120	PUERTO - HOYA DE LA PLATA	Alc. J. Ramírez Bethencourt (Centro Insular de Deportes)	28.1132,-15.417206	Av Alcalde José Ramírez Bethencourt, 6
1201	12	0	11	120	PUERTO - HOYA DE LA PLATA	Av. de Canarias (San Telmo)	28.108938,- 15.41523	Av de Canarias, 4
1201	12	0	12	120	PUERTO - HOYA DE LA PLATA	Mercado de Vegueta	28.1026563,- 15.4130011	Av Alcalde Díaz Saavedra Navarro, 1
1201	12	0	13	120	PUERTO - HOYA DE LA PLATA	Alc. Díaz Saavedra Navarro (Prudencio Guzmán)	28.096582,- 15.412393	Plaza de Santa Isabel, 2
1201	12	0	14	120	PUERTO - HOYA DE LA PLATA	Eufemiano Jurado (Juzgados)	28.0946629,- 15.4135542	Av Eufemiano Jurado, 1D
1201	12	0	15	120	PUERTO - HOYA DE LA PLATA	Eufemiano Jurado (Reyes Católicos)	28.0960648,- 15.4146458	Av Eufemiano Jurado, 4,
1201	12	0	16	120	PUERTO - HOYA DE LA PLATA	Córdoba (Bomberos Vegueta)	28.094959,- 15.415373	Calle Córdoba, 12
1201	12	0	17	120	PUERTO - HOYA DE LA PLATA	Córdoba, 40	28.092851,- 15.416967	Calle Córdoba, 40-42
1201	12	0	18	120	PUERTO - HOYA DE LA PLATA	Córdoba, frente 31	28.090733,- 15.41828	Calle Córdoba, 16
1201	12	0	19	120	PUERTO - HOYA DE LA PLATA	Córdoba, frente 49	28.088019,- 15.419573	Calle Córdoba, 42-44
1202	12	0	20	120	PUERTO - HOYA DE LA PLATA	Córdoba, 54	28.085908,- 15.417808	Calle Córdoba, 52
1202	12	0	21	120	PUERTO - HOYA DE LA PLATA	Alicante (Colegio Islas Canarias)	28.084727,- 15.417194	Calle Alicante, 34
1202	12	0	22	120	PUERTO - HOYA DE LA PLATA	Blas Cabrera Felipe (Iglesia Sta. Clara)	28.084012,- 15.419253	Paseo Blas Cabrera Felipe - Físico, 306
1202	12	0	23	120	PUERTO - HOYA DE LA PLATA	Blas Cabrera Felipe (Parking Hosp. Materno Insular)	28.08095,- 15.4185666666667	Paseo Blas Cabrera Felipe - Físico, 310
1202	12	0	24	120	PUERTO - HOYA DE LA PLATA	Blas Cabrera Felipe (Martín Freire)	28.07735,-15.41755	Paseo Blas Cabrera Felipe - Físico, 3D
1202	12	0	25	120	PUERTO - HOYA DE LA PLATA	Blas Cabrera Felipe, 14	28.074878,- 15.417308	Paseo Blas Cabrera Felipe - Físico, 14
1202	12	0	26	120	PUERTO - HOYA DE LA PLATA	Blas Cabrera Felipe (Carrefour)	28.071834,- 15.417817	Paseo Blas Cabrera Felipe (Físico), 40

1202 7	12	0	27	120	PUERTO - HOYA DE LA PLATA	Hoya de La Plata	28.070003,- 15.418131	Calle Hoya de la Plata, 2F
1210 1	12	1	1	121	HOYA DE LA PLATA - PUERTO	Hoya de La Plata	28.070003,- 15.418131	Calle Hoya de la Plata, 2F
1210 2	12	1	2	121	HOYA DE LA PLATA - PUERTO	Blas Cabrera Felipe (Carrefour)	28.072086,- 15.41763	Paseo Blas Cabrera Felipe - Físico, 40
1210 3	12	1	3	121	HOYA DE LA PLATA - PUERTO	Blas Cabrera Felipe, frente 16	28.074645,- 15.417144	Paseo Blas Cabrera Felipe - Físico, 17
1210 4	12	1	4	121	HOYA DE LA PLATA - PUERTO	Blas Cabrera Felipe (Martín Freire)	28.077533,- 15.417487	Paseo Blas Cabrera Felipe - Físico, 3D
1210 5	12	1	5	121	HOYA DE LA PLATA - PUERTO	Blas Cabrera Felipe (Parking Hosp. Materno Insular)	28.080343,- 15.418243	Paseo Blas Cabrera Felipe - Físico, 312
1210 6	12	1	6	121	HOYA DE LA PLATA - PUERTO	Blas Cabrera Felipe (Iglesia Sta. Clara)	28.08225,- 15.418766	Paseo Blas Cabrera Felipe - Físico, 310
1210 7	12	1	7	121	HOYA DE LA PLATA - PUERTO	Blas Cabrera Felipe, frente 306	28.084759,- 15.41906	Paseo Blas Cabrera Felipe - Físico, 306
1210 8	12	1	8	121	HOYA DE LA PLATA - PUERTO	Alicante (Colegio Islas Canarias)	28.085059,- 15.41711	Calle Alicante, 11
1210 9	12	1	9	121	HOYA DE LA PLATA - PUERTO	Córdoba, 71	28.086074,- 15.417922	Calle Córdoba, 52
1211 0	12	1	10	121	HOYA DE LA PLATA - PUERTO	Córdoba, 49	28.088141,- 15.419385	Calle Córdoba, 2
1211 1	12	1	11	121	HOYA DE LA PLATA - PUERTO	Córdoba, 29	28.090827,- 15.418112	Calle Córdoba, 13
1211 2	12	1	12	121	HOYA DE LA PLATA - PUERTO	Córdoba, 19	28.093752,- 15.416175	Calle Córdoba, 11-19
1211 3	12	1	13	121	HOYA DE LA PLATA - PUERTO	Córdoba (Bomberos Vegueta)	28.095124,- 15.415107	Calle Córdoba, 11
1211 4	12	1	14	121	HOYA DE LA PLATA - PUERTO	Eufemiano Jurado (Juzgados)	28.0948475,- 15.4137379	Av Eufemiano Jurado, 1D
1211 5	12	1	15	121	HOYA DE LA PLATA - PUERTO	Alicante (C. S. San José)	28.09265,- 15.413848	Calle Alicante, 4
1211 6	12	1	16	121	HOYA DE LA PLATA - PUERTO	Rafael Cabrera (Teatro)	28.1041617,- 15.4136967	Av Rafael Cabrera, 4
1211 7	12	1	17	121	HOYA DE LA PLATA - PUERTO	Rafael Cabrera (San Telmo)	28.108817,- 15.415927	Av Rafael Cabrera, 30
1211 8	12	1	18	121	HOYA DE LA PLATA - PUERTO	Av. Marítima del Sur (Juan XXIII)	28.121403,- 15.423873	Av Alcalde José Ramírez Bethencourt, 25
1211 9	12	1	19	121	HOYA DE LA PLATA - PUERTO	León y Castillo (Club Náutico)	28.132898,- 15.429523	Calle León y Castillo, 363
1212 0	12	1	20	121	HOYA DE LA PLATA - PUERTO	Intercambiador Sta. Catalina	28.140082,- 15.428278	Calle Nicolás Estevanez
1212 1	12	1	21	121	HOYA DE LA PLATA - PUERTO	Eduardo Benot, frente 5	28.1423,- 15.4301166666667	Calle Eduardo Benot, 12
1212 2	12	1	22	121	HOYA DE LA PLATA - PUERTO	Eduardo Benot, frente 23	28.143825,- 15.429793	Calle Eduardo Benot, 19
1212 3	12	1	23	121	HOYA DE LA PLATA - PUERTO	Agustín Millares Sall (Edificio Mapfre)	28.147872,- 15.427967	Calle Poeta Agustín Millares Sall
1212 4	12	1	24	121	HOYA DE LA PLATA - PUERTO	Juan Rejón (Castillo de La Luz)	28.149171,- 15.425367	Calle Juan Rejón, 73
1212 5	12	1	25	121	HOYA DE LA PLATA - PUERTO	Manuel Becerra (Puerto)	28.149954,- 15.421738	Calle Juan Rejón, 62
1700 1	17	0	1	170	TEATRO - AUDITORIO	Teatro	28.104377,- 15.41383	Av Rafael Cabrera, 4
1700 2	17	0	2	170	TEATRO - AUDITORIO	Rafael Cabrera (San Telmo)	28.108817,- 15.415927	Av Rafael Cabrera, 30
1700 3	17	0	3	170	TEATRO - AUDITORIO	Venegas (Usos Múltiples)	28.111689,- 15.418132	Calle Venegas, 54
1700 4	17	0	4	170	TEATRO - AUDITORIO	Venegas (Fuente Luminosa)	28.114254,- 15.419264	Calle Archivero Municipal Pedro Cullen

								del Castillo, 49
1700 5	17	0	5	170	TEATRO - AUDITORIO	Luis Doreste Silva, 22	28.116158,- 15.421211	Calle Luis Doreste Silva, 6-8
1700 6	17	0	6	170	TEATRO - AUDITORIO	Luis Doreste Silva, 50	28.118893,- 15.423591	Calle Luis Doreste Silva, 52-54
1700 7	17	0	7	170	TEATRO - AUDITORIO	Av. Marítima del Sur (Juan XXIII)	28.121403,- 15.423873	Av Alcalde José Ramírez Bethencourt, 25
1700 8	17	0	8	170	TEATRO - AUDITORIO	León y Castillo (Club Náutico)	28.132898,- 15.429523	Calle León y Castillo, 363
1700 9	17	0	9	170	TEATRO - AUDITORIO	Mesa y López, 42	28.134709,- 15.433731	Calle José Mesa López, 46
1701 0	17	0	10	170	TEATRO - AUDITORIO	Mesa y López (C.S. Alcaravaneras)	28.13365,- 15.436255	Av José Mesa y López, 56
1701 1	17	0	11	170	TEATRO - AUDITORIO	Mesa y López (Madera y Corcho)	28.131869,- 15.438488	Av José Mesa y López, 72
1701 2	17	0	12	170	TEATRO - AUDITORIO	Fernando Guanarteme, 118	28.132406,- 15.440781	Calle Fernando Guanarteme, 151-153
1701 3	17	0	13	170	TEATRO - AUDITORIO	Simancas, 70	28.129722,- 15.444062	Calle Simancas, 55
1701 4	17	0	14	170	TEATRO - AUDITORIO	Numancia, 68	28.130373,- 15.445503	Calle Numancia, 63
1701 5	17	0	15	170	TEATRO - AUDITORIO	Auditorio	28.1296833333333, -15.4480333333333	Calle Sagunto, 15
1710 1	17	1	1	171	AUDITORIO - TEATRO	Auditorio	28.129302,- 15.449076	Carretera del rincón, s- n
1710 2	17	1	2	171	AUDITORIO - TEATRO	Ind. José Sánchez Peñate (C. C. Las Arenas)	28.127667,- 15.446565	Av José Sánchez Peñate, 25
1710 3	17	1	3	171	AUDITORIO - TEATRO	Castillejos, 71	28.128738,- 15.444231	Calle Castillejos, 57
1710 4	17	1	4	171	AUDITORIO - TEATRO	Castillejos, 23	28.129927,- 15.442708	Calle Castillejos, 23
1710 5	17	1	5	171	AUDITORIO - TEATRO	Costa Rica, frente 33	28.130646,- 15.440087	Calle Costa Rica, 568
1710 6	17	1	6	171	AUDITORIO - TEATRO	Mesa y López (Madera y Corcho)	28.131422,- 15.438624	Av de José Mesa y López, 67
1710 7	17	1	7	171	AUDITORIO - TEATRO	Mesa y López (C.S. Alcaravaneras)	28.133275,- 15.436327	Av José Mesa y López, 37-39
1710 8	17	1	8	171	AUDITORIO - TEATRO	Mesa y López, 7	28.134699,- 15.431396	Calle José Mesa López, 13
1710 9	17	1	9	171	AUDITORIO - TEATRO	León y Castillo (Edificio José Antonio)	28.132614,- 15.430047	Calle León y Castillo, 361
1711 0	17	1	10	171	AUDITORIO - TEATRO	Alc. J. Ramírez Bethencourt (Parque Romano)	28.126104,- 15.428078	Av Alcalde José Ramírez Bethencourt, 129
1711 1	17	1	11	171	AUDITORIO - TEATRO	Alc. J. Ramírez Bethencourt (Carvajal)	28.118532,- 15.421208	Av Alcalde José Ramírez Bethencourt, 13-21
1711 2	17	1	12	171	AUDITORIO - TEATRO	Alc. J. Ramírez Bethencourt (Centro Insular de Deportes)	28.1132,-15.417206	Av Alcalde José Ramírez Bethencourt, 6
1711 3	17	1	13	171	AUDITORIO - TEATRO	Av. de Canarias (San Telmo)	28.108938,- 15.41523	Av de Canarias, 4
1711 4	17	1	14	171	AUDITORIO - TEATRO	Teatro	28.104377,- 15.41383	Av Rafael Cabrera, 4,
1900 1	19	0	1	190	SANTA CATALINA - EL SEBADAL	Eduardo Benot (Parque Sta. Catalina)	28.1420333333333, -15.43005	Calle Eduardo Benot, 5
1900 2	19	0	2	190	SANTA CATALINA - EL SEBADAL	Eduardo Benot, frente 23	28.143825,- 15.429793	Calle Eduardo Benot, 19
1900 3	19	0	3	190	SANTA CATALINA - EL SEBADAL	Agustín Millares Sall (Edificio	28.147872,- 15.427967	Calle Poeta Agustín Millares Sall

						Mapfre)		
1900 4	19	0	4	190	SANTA CATALINA - EL SEBADAL	Juan Rejón (Castillo de la Luz)	28.149171,- 15.425367	Calle Juan Rejón, 73,
1900 5	19	0	5	190	SANTA CATALINA - EL SEBADAL	Manuel Becerra (Puerto)	28.149954,- 15.421738	Calle Juan Rejón, 62
1900 6	19	0	6	190	SANTA CATALINA - EL SEBADAL	Roque Nublo, 4	28.152294,- 15.421606	Calle Roque Nublo, 21
1900 7	19	0	7	190	SANTA CATALINA - EL SEBADAL	Profesor Lozano (Catamarca)	28.1544333333333, -15.4209833333333	Calle Profesor Lozano, 37
1900 8	19	0	8	190	SANTA CATALINA - EL SEBADAL	Profesor Lozano (Rotonda Arequipa)	28.155936,- 15.420337	Calle Profesor Lozano, 27
1900 9	19	0	9	190	SANTA CATALINA - EL SEBADAL	Arequipa (Guaguas Municipales)	28.156674,- 15.422017	Calle Arequipa, 236
1901 0	19	0	10	190	SANTA CATALINA - EL SEBADAL	Profesor Lozano (Canarias 7)	28.155916,- 15.417239	Calle Profesor Lozano, 5
1901 1	19	0	11	190	SANTA CATALINA - EL SEBADAL	Dr. Juan Domínguez Pérez, 2	28.156314,- 15.415595	Calle Doctor Juan Domínguez Pérez, 133
1901 2	19	0	12	190	SANTA CATALINA - EL SEBADAL	Dr. Juan Domínguez Pérez, 18	28.159812,- 15.412026	Calle Doctor Juan Domínguez Pérez, 54- 94
1901 3	19	0	13	190	SANTA CATALINA - EL SEBADAL	Dr. Juan Domínguez Pérez (El Sebadal)	28.161521,- 15.410644	Calle Doctor Juan Domínguez Pérez, 71
1910 1	19	1	1	191	EL SEBADAL - SANTA CATALINA	Dr. Juan Domínguez Pérez (El Sebadal)	28.161521,- 15.410644	Calle Doctor Juan Domínguez Pérez, 71
1910 2	19	1	2	191	EL SEBADAL - SANTA CATALINA	Dr. Juan Domínguez Pérez, 49	28.160132,- 15.411964	Calle Doctor Juan Domínguez Pérez, 71- 105
1910 3	19	1	3	191	EL SEBADAL - SANTA CATALINA	Dr. Juan Domínguez Pérez, 25	28.158429,- 15.413707	Calle Doctor Juan Domínguez Pérez, 107- 157
1910 4	19	1	4	191	EL SEBADAL - SANTA CATALINA	Profesor Lozano, 10	28.156143,- 15.416702	Calle Profesor Lozano, 18-20
1910 5	19	1	5	191	EL SEBADAL - SANTA CATALINA	Profesor Lozano, 34	28.156257,- 15.419856	Calle Profesor Lozano, 34
1910 6	19	1	6	191	EL SEBADAL - SANTA CATALINA	Arequipa (Guaguas Municipales)	28.156674,- 15.422017	Calle Arequipa, 236
1910 7	19	1	7	191	EL SEBADAL - SANTA CATALINA	Catamarca, frente 2	28.152401,- 15.418598	Calle Vía 8010, 8A
1910 8	19	1	8	191	EL SEBADAL - SANTA CATALINA	Juan Rejón (frente Bomberos La Isleta)	28.150397,- 15.419249	Calle Juan Rejón, 141
1910 9	19	1	9	191	EL SEBADAL - SANTA CATALINA	Manuel Becerra (Puerto)	28.150375,- 15.422071	Calle la Naval, 203
1911 0	19	1	10	191	EL SEBADAL - SANTA CATALINA	Juan Rejón, 67	28.149154,- 15.425972	Calle Juan Rejón, 63
1911 1	19	1	11	191	EL SEBADAL - SANTA CATALINA	Albareda (Mercado del Puerto)	28.148147,- 15.429239	Calle Albareda, 82
1911 2	19	1	12	191	EL SEBADAL - SANTA CATALINA	Albareda, 43	28.144897,- 15.430324	Calle Albareda, 43
1911 3	19	1	13	191	EL SEBADAL - SANTA CATALINA	Eduardo Benot (Parque Sta. Catalina)	28.1421333333333, -15.43005	Calle Eduardo Benot, 5
2000 1	20	0	1	200	SANTA CATALINA - LA ISLETA	Eduardo Benot (Parque Sta. Catalina)	28.1420333333333, -15.43005	Calle Eduardo Benot, 5
2000 2	20	0	2	200	SANTA CATALINA - LA ISLETA	Eduardo Benot, frente 23	28.143825,- 15.429793	Calle Eduardo Benot, 19
2000 3	20	0	3	200	SANTA CATALINA - LA ISLETA	López Socas, 10	28.148165,- 15.428633	Calle López Socas, 10
2000	20	0	4	200	SANTA CATALINA -	Luján Pérez, 20	28.149822,-	Calle Luján Pérez, 30

4					LA ISLETA		15.429676	
2000 5	20	0	5	200	SANTA CATALINA - LA ISLETA	Luján Pérez, 66	28.151496,- 15.430004	Calle Luján Pérez, 76
2000 6	20	0	6	200	SANTA CATALINA - LA ISLETA	Tinguaro, 18	28.153103,- 15.429358	Calle Tinguaro, 20
2000 7	20	0	7	200	SANTA CATALINA - LA ISLETA	Gumidafe, 30	28.152805,- 15.428099	Calle Gumidafe, 9
2000 8	20	0	8	200	SANTA CATALINA - LA ISLETA	Osorio, 28	28.152692,- 15.426303	Calle Osorio, 26
2000 9	20	0	9	200	SANTA CATALINA - LA ISLETA	Osorio, 58	28.152531,- 15.424636	Calle Osorio, 58
2001 0	20	0	10	200	SANTA CATALINA - LA ISLETA	Manuel Becerra (Puerto)	28.149954,- 15.421738	Calle Juan Rejón, 62
2010 1	20	1	1	201	LA ISLETA - SANTA CATALINA	Manuel Becerra (Puerto)	28.149954,- 15.421738	Calle Juan Rejón, 62
2010 2	20	1	2	201	LA ISLETA - SANTA CATALINA	Palmar, frente 82	28.153324,- 15.423433	Calle Palmar, 86
2010 3	20	1	3	201	LA ISLETA - SANTA CATALINA	Bandama, 25	28.154367,- 15.424529	Calle Guadalfia, 12-14
2010 4	20	1	4	201	LA ISLETA - SANTA CATALINA	Eduardo Peñate Santana, local 4	28.154456,- 15.427189	Calle Coronel Rocha, 36
2010 5	20	1	5	201	LA ISLETA - SANTA CATALINA	José Luis Guerra de Armas (Alonso Quijano)	28.154541,- 15.428299	Calle Alonso Quijano, 20
2010 6	20	1	6	201	LA ISLETA - SANTA CATALINA	Coronel Rocha (Nueva Isleta)	28.155387,- 15.430764	Calle Coronel Rocha, 12
2010 7	20	1	7	201	LA ISLETA - SANTA CATALINA	Faro, 99	28.153114,- 15.430773	Calle Faro, 100
2010 8	20	1	8	201	LA ISLETA - SANTA CATALINA	Faro, 43	28.150697,- 15.430608	Calle Faro, 46
2010 9	20	1	9	201	LA ISLETA - SANTA CATALINA	Faro, 11	28.148962,- 15.430312	Calle Faro, 12
2011 0	20	1	10	201	LA ISLETA - SANTA CATALINA	Albareda (Mercado del Puerto)	28.148147,- 15.429239	Calle Albareda, 82
2011 1	20	1	11	201	LA ISLETA - SANTA CATALINA	Albareda, 43	28.144897,- 15.430324	Calle Albareda, 43
2011 2	20	1	12	201	LA ISLETA - SANTA CATALINA	Eduardo Benot (Parque Sta. Catalina)	28.1420333333333, -15.43005	Calle Eduardo Benot, 5
2500 1	25	0	1	250	AUDITORIO - CAMPUS UNIVERSITARIO	Auditorio	28.129257,- 15.449235	Carretera del Rincón, S/N
2500 2	25	0	2	250	AUDITORIO - CAMPUS UNIVERSITARIO	Ind. José Sánchez Peñate (C. C. Las Arenas)	28.127667,- 15.446565	Av José Sánchez Peñate, 25
2500 3	25	0	3	250	AUDITORIO - CAMPUS UNIVERSITARIO	Castillejos, 71	28.128738,- 15.444231	Calle Castillejos, 57
2500 4	25	0	4	250	AUDITORIO - CAMPUS UNIVERSITARIO	Castillejos, 23	28.129927,- 15.442708	Calle Castillejos, 23
2500 5	25	0	5	250	AUDITORIO - CAMPUS UNIVERSITARIO	Costa Rica, frente 33	28.130646,- 15.440087	Calle Costa Rica, 568
2500 6	25	0	6	250	AUDITORIO - CAMPUS UNIVERSITARIO	Mesa y López (Madera y Corcho)	28.131422,- 15.438624	Av de José Mesa y López, 67
2500 7	25	0	7	250	AUDITORIO - CAMPUS UNIVERSITARIO	Mesa y López (C.S. Alcaravaneras)	28.133275,- 15.436327	Av José Mesa y López, 37-39
2500 8	25	0	8	250	AUDITORIO - CAMPUS UNIVERSITARIO	Galicia (Mercado Central)	28.1333778,- 15.433161	Calle Galicia, 24
2500 9	25	0	9	250	AUDITORIO - CAMPUS UNIVERSITARIO	Galicia	28.131203,- 15.4329634	Calle Pío XII, 75
2501 0	25	0	10	250	AUDITORIO - CAMPUS	Pío XII (Estadio Insular)	28.128104,- 15.433072	Calle Antonio Zerolo, 2

					UNIVERSITARIO			
2501 1	25	0	11	250	AUDITORIO - CAMPUS UNIVERSITARIO	Pío XII (Colegio Teresiano)	28.124881,- 15.43217	Calle Pío XII, 27A
2501 2	25	0	12	250	AUDITORIO - CAMPUS UNIVERSITARIO	Emilio Ley (Parque Doramas)	28.120219,- 15.429505	Calle Emilio Ley, 9
2501 3	25	0	13	250	AUDITORIO - CAMPUS UNIVERSITARIO	Pérez del Toro (Pl. Dr. Juan Bosch Millares)	28.117206,- 15.427311	Calle Pérez del Toro, 89
2501 4	25	0	14	250	AUDITORIO - CAMPUS UNIVERSITARIO	Pérez del Toro, 13	28.113648,- 15.425118	Calle Pérez del Toro, 11- 13
2501 5	25	0	15	250	AUDITORIO - CAMPUS UNIVERSITARIO	Pl. de la Constitución (Obelisco)	28.111873,- 15.42374	Plaza de la Constitución, 5-6
2501 6	25	0	16	250	AUDITORIO - CAMPUS UNIVERSITARIO	Tomás Morales, 5	28.108348,- 15.420749	Paseo Tomás Morales, 7
2501 7	25	0	17	250	AUDITORIO - CAMPUS UNIVERSITARIO	Primero de Mayo, 35	28.105602,- 15.41994	Avenida Primero de Mayo
2501 8	25	0	18	250	AUDITORIO - CAMPUS UNIVERSITARIO	Primero de Mayo, 1	28.1023833,- 15.4175833	Av Primero de Mayo, 4
2501 9	25	0	19	250	AUDITORIO - CAMPUS UNIVERSITARIO	Guiniguada	28.1005667,- 15.4191167	Calle Bernardino Correa Viera
2502 0	25	0	20	250	AUDITORIO - CAMPUS UNIVERSITARIO	Bernardino Correa Viera, frente 13	28.101083,- 15.418798	Calle Bernardino Correa Viera, 5
2502 1	25	0	21	250	AUDITORIO - CAMPUS UNIVERSITARIO	Arminda, frente 1	28.0848667,- 15.43735	Calle Arminda, 27
2502 2	25	0	22	250	AUDITORIO - CAMPUS UNIVERSITARIO	Ctra. Pico Viento (Asilo de Ancianos)	28.07416,- 15.446978	Carretera Pico Viento, 69
2502 3	25	0	23	250	AUDITORIO - CAMPUS UNIVERSITARIO	Ingenierías	28.07147,- 15.455304	Comedores Universitarios
2502 4	25	0	24	250	AUDITORIO - CAMPUS UNIVERSITARIO	Ciencias Básicas / Teología	28.072523,- 15.451691	Carretera Lomo Blanco, 137
2502 5	25	0	25	250	AUDITORIO - CAMPUS UNIVERSITARIO	Ciencias Básicas / Informática	28.07282,- 15.451986	Facultad de Informática
2502 6	25	0	26	250	AUDITORIO - CAMPUS UNIVERSITARIO	Biblioteca Universitaria	28.075587,- 15.451742	Calle Saulo Torón, 13
2502 7	25	0	27	250	AUDITORIO - CAMPUS UNIVERSITARIO	Ciencias Jurídicas (Módulo B)	28.07817,- 15.449235	Calle Farmacéutico Enrique Arroyo, 4
2502 8	25	0	28	250	AUDITORIO - CAMPUS UNIVERSITARIO	Parque Tecnológico	28.079484,- 15.453412	Edificio de Termodinámica
2502 9	25	0	29	250	AUDITORIO - CAMPUS UNIVERSITARIO	Residencias (Campus Universitario)	28.079581,- 15.452474	Edificio de Termodinámica
2510 1	25	1	1	251	CAMPUS UNIVERSITARIO - AUDITORIO	Residencias (Campus Universitario)	28.079581,- 15.452474	Edificio de Termodinámica
2510 2	25	1	2	251	CAMPUS UNIVERSITARIO - AUDITORIO	Aulario Ciencias Jurídicas	28.078113,- 15.449491	Calle Farmacéutico Enrique Arroyo, 6-8
2510 3	25	1	3	251	CAMPUS UNIVERSITARIO - AUDITORIO	Biblioteca Universitaria	28.07562,- 15.452062	Calle Arquitecto Fernando Navarro
2510 4	25	1	4	251	CAMPUS UNIVERSITARIO -	Ciencias Básicas	28.073407,- 15.452819	Facultad de Ciencias del Mar

					AUDITORIO			
2510 5	25	1	5	251	CAMPUS UNIVERSITARIO - AUDITORIO	Ingenierías	28.070647,- 15.455137	Calle la Naval
2510 6	25	1	6	251	CAMPUS UNIVERSITARIO - AUDITORIO	Ctra. Pico Viento (Asilo de Ancianos)	28.074094,- 15.446685	Carretera Pico Viento, 67
2510 7	25	1	7	251	CAMPUS UNIVERSITARIO - AUDITORIO	Ctra. subida a Tafira (La Favorita)	28.084836,- 15.436839	GC-110
2510 8	25	1	8	251	CAMPUS UNIVERSITARIO - AUDITORIO	Rafael Cabrera (Teatro)	28.104388,- 15.413797	Av Rafael Cabrera, 4
2510 9	25	1	9	251	CAMPUS UNIVERSITARIO - AUDITORIO	Rafael Cabrera (San Telmo)	28.10914,- 15.416084	Av Rafael Cabrera, 30
2511 0	25	1	10	251	CAMPUS UNIVERSITARIO - AUDITORIO	Tomás Morales, 4	28.1085,-15.42075	Paseo Tomás Morales, 16
2511 1	25	1	11	251	CAMPUS UNIVERSITARIO - AUDITORIO	Alameda de Colón	28.1030638,- 15.4171632	Calle Doctor Domingo Déniz, 1
2511 2	25	1	12	251	CAMPUS UNIVERSITARIO - AUDITORIO	Primero de Mayo (Correos)	28.106143,- 15.420125	Av Primero de Mayo, 62
2511 3	25	1	13	251	CAMPUS UNIVERSITARIO - AUDITORIO	Ingeniero J. Bosch y Síntes, 1	28.108483,- 15.421034	Calle Senador Castillo Olivares, 1
2511 4	25	1	14	251	CAMPUS UNIVERSITARIO - AUDITORIO	Pl. de la Constitución (Obelisco)	28.112268,- 15.4233608	Calle Alfonso XIII, 7
2511 5	25	1	15	251	CAMPUS UNIVERSITARIO - AUDITORIO	Tomás Morales, frente 69	28.115001,- 15.425162	Paseo Tomás Morales, 94
2511 6	25	1	16	251	CAMPUS UNIVERSITARIO - AUDITORIO	Tomás Morales, 120	28.116932,- 15.426284	Paseo Tomás Morales, 118
2511 7	25	1	17	251	CAMPUS UNIVERSITARIO - AUDITORIO	Emilio Ley (Piscinas Julio Navarro)	28.121336,- 15.429834	Calle Emilio Ley, 4
2511 8	25	1	18	251	CAMPUS UNIVERSITARIO - AUDITORIO	Pío XII (Colegio Teresiano)	28.125041,- 15.432122	Calle Pío XII, 27A
2511 9	25	1	19	251	CAMPUS UNIVERSITARIO - AUDITORIO	Pío XII (Estadio Insular)	28.128925,- 15.432851	Calle Pío XII, 62
2512 0	25	1	20	251	CAMPUS UNIVERSITARIO - AUDITORIO	Galicia (Mercado Central)	28.131953,- 15.432921	Calle Pío XII, 81
2512 1	25	1	21	251	CAMPUS UNIVERSITARIO - AUDITORIO	Mesa y López, 42	28.134709,- 15.433731	Calle José Mesa López, 46
2512 2	25	1	22	251	CAMPUS UNIVERSITARIO - AUDITORIO	Mesa y López (C.S. Alcaravaneras)	28.13365,- 15.436255	Av José Mesa y López, 56
2512 3	25	1	23	251	CAMPUS UNIVERSITARIO - AUDITORIO	Mesa y López (Madera y Corcho)	28.131869,- 15.438488	Av José Mesa y López, 72
2512 4	25	1	24	251	CAMPUS UNIVERSITARIO - AUDITORIO	Fernando Guanarteme, 118	28.132406,- 15.440781	Calle Fernando Guanarteme, 151-153
2512 5	25	1	25	251	CAMPUS UNIVERSITARIO - AUDITORIO	Simancas, 70	28.129722,- 15.444062	Calle Simancas, 55
2512 6	25	1	26	251	CAMPUS UNIVERSITARIO - AUDITORIO	Numancia, 68	28.130373,- 15.445503	Calle Numancia, 63
2512 7	25	1	27	251	CAMPUS UNIVERSITARIO -	Auditorio	28.1296833,- 15.4480333	Calle Sagunto, 15

					AUDITORIO			
2600 1	26	0	1	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Intercambiador Sta. Catalina	28.140028,- 15.428336	Calle Nicolás Estevanez
2600 2	26	0	2	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Parque Santa Catalina	28.140338,- 15.430777	Calle León y Castillo, 439
2600 3	26	0	3	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Mesa y López, 38	28.134783,- 15.433043	Calle José Mesa López, 40
2600 4	26	0	4	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Mesa y López (C.S. Alcaravanas)	28.13365,- 15.436255	Av José Mesa y López, 56
2600 5	26	0	5	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Mesa y López (Madera y Corcho)	28.131869,- 15.438488	Av José Mesa y López, 72
2600 6	26	0	6	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Mesa y López, 86	28.1300833,- 15.4405667	Av José Mesa y López, 86
2600 7	26	0	7	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Rotonda Hosp. Dr. Negrín	28.120113,- 15.443103	Av Juan Carlos I, 3
2600 8	26	0	8	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Juan Carlos I (Hosp. Dr. Negrín)	28.119023,- 15.444233	Av Juan Carlos I, 3
2600 9	26	0	9	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Juan Carlos I, 10	28.11818,- 15.445241	Av Juan Carlos I, 3
2601 0	26	0	10	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Juan Carlos I, 20	28.115724,- 15.447926	Av Juan Carlos I
2601 1	26	0	11	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Juan Carlos I, 50	28.113198,- 15.45011	Av Juan Carlos I
2601 2	26	0	12	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Pintor Felo Monzón, 2	28.11145,-15.45095	Av Pintor Felo Monzón, 4
2601 3	26	0	13	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Pintor Felo Monzón (Bambú)	28.1084667,- 15.45165	Av Pintor Felo Monzón
2601 4	26	0	14	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Pintor Felo Monzón, 22	28.106397,- 15.452327	Av Pintor Felo Monzón, 15
2601 5	26	0	15	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Pintor Felo Monzón, 28	28.10448,- 15.452786	Av Pintor Felo Monzón, 28
2601 6	26	0	16	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Pintor Felo Monzón, 38	28.100852,- 15.453749	Av Pintor Felo Monzón, 29
2601 7	26	0	17	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Pintor Felo Monzón (C.C. Siete Palmas)	28.098602,- 15.454777	Av Pintor Felo Monzón, S-N
2601 8	26	0	18	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Pintor Felo Monzón (Hipercor)	28.0952,- 15.4561833	Av Pintor Felo Monzón, 38
2601 9	26	0	19	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Ctra. Lomo Blanco, (Puente Aéreo)	28.0840167,- 15.4456167	GC-3
2602 0	26	0	20	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Ctra. Lomo Blanco (IES Felo Monzón)	28.081858,- 15.447164	GC-112
2602 1	26	0	21	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Aulario Ciencias Jurídicas	28.078113,- 15.449491	Calle Farmacéutico Enrique Arroyo, 6-8
2602 2	26	0	22	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Biblioteca Universitaria	28.07562,- 15.452062	Calle Arquitecto Fernando Navarro
2602 3	26	0	23	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Ciencias Básicas	28.073407,- 15.452819	Facultad de Ciencias del Mar

2602 4	26	0	24	260	SANTA CATALINA - CAMPUS UNIVERSITARIO	Campus Universitario (Ingenierías)	28.070647,- 15.455137	Calle la Naval
2610 1	26	1	1	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Campus Universitario (Ingenierías)	28.07147,- 15.455304	Comedores Universitarios
2610 2	26	1	2	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Ciencias Básicas / Teología	28.072523,- 15.451691	Carretera Lomo Blanco, 137
2610 3	26	1	3	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Ciencias Básicas / Informática	28.07282,- 15.451986	Facultad de Informática
2610 4	26	1	4	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Biblioteca Universitaria	28.075587,- 15.451742	Calle Saulo Torón, 13
2610 5	26	1	5	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Ciencias Jurídicas (Módulo B)	28.07817,- 15.449235	Calle Farmacéutico Enrique Arroyo, 4
2610 6	26	1	6	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Ctra. Lomo Blanco, frente 51	28.081874,- 15.447016	GC-112
2610 7	26	1	7	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Rotonda Lomo Blanco	28.0834667,- 15.4465167	GC-112
2610 8	26	1	8	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Pintor Felo Monzón (frente Hipercor)	28.095408,- 15.45576	Av Pintor Felo Monzón, 49
2610 9	26	1	9	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Pintor Felo Monzón, 37	28.098103,- 15.454533	Av Pintor Felo Monzón, 26
2611 0	26	1	10	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Pintor Felo Monzón, 29	28.100802,- 15.453437	Av Pintor Felo Monzón, 29
2611 1	26	1	11	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Pintor Felo Monzón, 25	28.104392,- 15.452452	Av Pintor Felo Monzón, 28
2611 2	26	1	12	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Pintor Felo Monzón, 17	28.106252,- 15.451929	Av Pintor Felo Monzón, 17
2611 3	26	1	13	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Pintor Felo Monzón (Trópico)	28.1087,- 15.4512833	Av Pintor Felo Monzón
2611 4	26	1	14	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Pintor Felo Monzón, 1	28.1112833,- 15.4506833	Av Pintor Felo Monzón, 4
2611 5	26	1	15	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Juan Carlos I (C.C. Las Ramblas)	28.113718,- 15.449282	Av Juan Carlos I
2611 6	26	1	16	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Juan Carlos I, 23	28.115552,- 15.447654	Av Juan Carlos I,
2611 7	26	1	17	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Juan Carlos I, 9	28.118017,- 15.444953	Av Juan Carlos I, 3
2611 8	26	1	18	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Juan Carlos I (Rotonda Hosp. Dr. Negrín)	28.119479,-15.4432	Av Juan Carlos I, 3
2611 9	26	1	19	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Mesa y López, 85	28.129736,- 15.440707	Av José Mesa y López, 88
2612 0	26	1	20	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Mesa y López (Madera y Corcho)	28.131422,- 15.438624	Av de José Mesa y López, 67
2612 1	26	1	21	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Mesa y López (C.S. Alcaravaneras)	28.133275,- 15.436327	Av José Mesa y López, 37-39
2612 2	26	1	22	261	CAMPUS UNIVERSITARIO - SANTA CATALINA	Mesa y López, 3	28.134834,- 15.429695	Av José Mesa Y López, 1
2612	26	1	23	261	CAMPUS	Intercambiador	28.140028,-	Calle Nicolás Estevanez

3					UNIVERSITARIO - SANTA CATALINA	Sta. Catalina	15.428336	
4800 1	48	0	1	480	ESCALERITAS - CAMPUS UNIVERSITARIO	Cronista Navarro Ruiz (Escaleritas)	28.1236166666667, -15.4354333333333	Calle Cronista Navarro Ruiz, 8
4800 2	48	0	2	480	ESCALERITAS - CAMPUS UNIVERSITARIO	Av. Escaleritas, 48	28.121634,- 15.436521	Av Escaleritas, 46
4800 3	48	0	3	480	ESCALERITAS - CAMPUS UNIVERSITARIO	Obispo Romo (Parque Hnos. Millares)	28.116851,- 15.437046	Calle Obispo Romo, 46
4800 4	48	0	4	480	ESCALERITAS - CAMPUS UNIVERSITARIO	Obispo Romo (Mercado Altavista)	28.114456,- 15.435344	Calle Obispo Romo, 17
4800 5	48	0	5	480	ESCALERITAS - CAMPUS UNIVERSITARIO	Zaragoza, 27	28.111085,- 15.432905	Calle Zaragoza, 27
4800 6	48	0	6	480	ESCALERITAS - CAMPUS UNIVERSITARIO	Don Pedro Infinito, 54	28.109715,- 15.432285	Calle Don Pedro Infinito, 52
4800 7	48	0	7	480	ESCALERITAS - CAMPUS UNIVERSITARIO	Don Pedro Infinito, 120	28.108443,- 15.434695	Calle de Pedro Infinito, 96-120
4800 8	48	0	8	480	ESCALERITAS - CAMPUS UNIVERSITARIO	Don Pedro Infinito, 152	28.107725,- 15.436093	Calle Don Pedro Infinito, 152
4800 9	48	0	9	480	ESCALERITAS - CAMPUS UNIVERSITARIO	Don Pedro Infinito, 194	28.106583,- 15.438321	Calle de Pedro Infinito, 194-196
4801 0	48	0	10	480	ESCALERITAS - CAMPUS UNIVERSITARIO	Don Pedro Infinito, 204	28.105911,- 15.439728	Calle Don Pedro Infinito, 206
4801 1	48	0	11	480	ESCALERITAS - CAMPUS UNIVERSITARIO	Ctra. del Norte (C.C. La Ballena)	28.100343,- 15.442313	Carretera del Norte, 112
4801 2	48	0	12	480	ESCALERITAS - CAMPUS UNIVERSITARIO	Ctra. Lomo Blanco, (Puente Aéreo)	28.0834667,- 15.4465167	GC-112
4801 3	48	0	13	480	ESCALERITAS - CAMPUS UNIVERSITARIO	Ctra. Lomo Blanco (IES Felo Monzón)	28.081858,- 15.447164	GC-112
4801 4	48	0	14	480	ESCALERITAS - CAMPUS UNIVERSITARIO	Aulario Ciencias Jurídicas	28.078113,- 15.449491	Calle Farmacéutico Enrique Arroyo, 6-8
4801 5	48	0	15	480	ESCALERITAS - CAMPUS UNIVERSITARIO	Biblioteca Universitaria	28.07562,- 15.452062	Calle Arquitecto Fernando Navarro
4801 6	48	0	16	480	ESCALERITAS - CAMPUS UNIVERSITARIO	Ciencias Básicas	28.073407,- 15.452819	Facultad de Ciencias del Mar
4801 7	48	0	17	480	ESCALERITAS - CAMPUS UNIVERSITARIO	Campus Universitario (Ingenierías)	28.0708667,- 15.45556667	Calle la Naval
4810 1	48	1	1	481	CAMPUS UNIVERSITARIO - ESCALERITAS	Campus Universitario (Ingenierías)	28.0708666666667, -15.4555666666667	Calle la Naval
4810 2	48	1	2	481	CAMPUS UNIVERSITARIO - ESCALERITAS	Ciencias Básicas / Teología	28.072523,- 15.451691	Carretera Lomo Blanco, 137
4810 3	48	1	3	481	CAMPUS UNIVERSITARIO - ESCALERITAS	Ciencias Básicas / Informática	28.07282,- 15.451986	Facultad de Informática
4810 4	48	1	4	481	CAMPUS UNIVERSITARIO - ESCALERITAS	Biblioteca Universitaria	28.075587,- 15.451742	Calle Saulo Torón, 13
4810 5	48	1	5	481	CAMPUS UNIVERSITARIO - ESCALERITAS	Ciencias Jurídicas (Módulo B)	28.07817,- 15.449235	Calle Farmacéutico Enrique Arroyo, 4
4810 6	48	1	6	481	CAMPUS UNIVERSITARIO -	Ctra. Lomo Blanco, frente 51	28.081874,- 15.447016	GC-112

					ESCALERITAS			
48107	48	1	7	481	CAMPUS UNIVERSITARIO - ESCALERITAS	Rotonda Lomo Blanco	28.0840167,- 15.4456167	GC-3
48108	48	1	8	481	CAMPUS UNIVERSITARIO - ESCALERITAS	Av. Escaleritas (C.C. La Ballena)	28.102818,- 15.442719	Av Escaleritas, 193
48109	48	1	9	481	CAMPUS UNIVERSITARIO - ESCALERITAS	Av. Escaleritas (Urb. Sansofé)	28.104065,- 15.440898	Av Escaleritas, 191
48110	48	1	10	481	CAMPUS UNIVERSITARIO - ESCALERITAS	Mariucha, 171	28.107169,- 15.439785	Calle Mariucha, 173
48111	48	1	11	481	CAMPUS UNIVERSITARIO - ESCALERITAS	Mariucha, 139	28.109234,- 15.437861	Calle Mariucha, 139
48112	48	1	12	481	CAMPUS UNIVERSITARIO - ESCALERITAS	Mariucha (Canódromo)	28.11125,- 15.4343666666667	Calle Mariucha, 123
48113	48	1	13	481	CAMPUS UNIVERSITARIO - ESCALERITAS	Obispo Romo, frente 27	28.115508,- 15.435786	Calle Obispo Romo, 27
48114	48	1	14	481	CAMPUS UNIVERSITARIO - ESCALERITAS	Av. Escaleritas, 51	28.118586,- 15.437041	Calle Francisco de Torquemada, 38
48115	48	1	15	481	CAMPUS UNIVERSITARIO - ESCALERITAS	Av. Escaleritas (Cáritas)	28.120676,- 15.436559	Calle Padre Anchieta, 6,
48116	48	1	16	481	CAMPUS UNIVERSITARIO - ESCALERITAS	Cronista Navarro Ruiz (Escaleritas)	28.12361667,- 15.4354333333	Calle Cronista Navarro Ruiz, 8

Referencias

- [1] W. Frank Ableson, Robi Sen, Chris King. (2011) Android Guía para desarrolladores. Segunda Edición.978-84-415-2958-8.
- [2] **Android WebKit:**
<http://developer.android.com/reference/android/webkit/package-summary.html>
- [3] **Android Developer:**
<http://developer.android.com/guide/topics/graphics/opengl.html>
- [4] **Android Studio | Android Developer:**
<https://developer.android.com/sdk/installing/studio.html>
- [5] **Android SDK | Android Developer**
<http://developer.android.com/sdk/index.html>
- [6] **Android DDMS | Android Developer:**
<http://developer.android.com/tools/debugging/ddms.html>
- [7] **Eclipse:**
<http://www.eclipse.org/>
- [8] **XML |World Wide Web Consortium**
<http://www.w3.org/XML/>
- [9] **SOAP | World Wide Web Consortium**
<http://www.w3.org/TR/soap/>
- [10] **XAMMP**
<https://www.apachefriends.org/es/about.html>
- [11] **Apache**
https://httpd.apache.org/ABOUT_APACHE.html
- [12] **The Apache Software Foundation**
<http://axis.apache.org/axis2/java/core/>
- [13] **Microsoft Excel**
<http://office.microsoft.com/es-es/microsoft-excel-software-de-hoja-de-calculo-FX010048762.aspx>
- [14] **API de Google Maps**
<https://developers.google.com/maps/?hl=es>
- [15] **The Apache Software Foundation**
<http://tomcat.apache.org/>
- [16] **Web life123**
<http://www.life123.com/technology/computer-software/applet/what-is-an-applet.shtml>
- [17] **Web StackOverFlow**
<http://stackoverflow.com/tags/servlets/info>

- [18] **Web DesarrolloWeb.com**
<http://www.desarrolloweb.com/articulos/731.php>

- [19] **Web del Jakarta Project**
<http://jakarta.apache.org/>

- [20] **Wikipedia**
<http://es.wikipedia.org/wiki/POJO>

- [21] Roberto Montero Miguel. (2011) Java 7. 978-84-415-2988-5. ANAYA.

Las referencias a páginas webs, se comprobaron en julio del año 2014.

Glosario

3D	Tres dimensiones
AOP	<i>Aspect Oriented Programming</i>
API	<i>Application Programming Interface</i>
COITT	<i>Colegio Oficial de Ingenieros Técnicos de Telecomunicación</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i>
IP	<i>Internet Protocol</i>
JRE	<i>Java Runtime Environment</i>
PFC	<i>Proyecto Final de Carrera</i>
SQL	<i>Structured Query Language</i>
src	<i>SouRCe</i>
ULPGC	<i>Universidad de Las Palmas de Gran Canaria</i>
URL	<i>Uniform Resource Locator</i>
WTP	<i>Web Tools Platform</i>
WWW	<i>World Wide Web</i>
XML	<i>eXtensible Markup Language</i>
POJO	<i>Plain Old Object Java</i>
SDK	<i>Software Development Kit</i>
ADT	<i>Android Development Tools</i>
NDK	<i>Native Development Kit</i>
OpenGL	<i>Open Graphics Library</i>
W3C	<i>World Wide Web Consortium</i>
SOAP	<i>Simple Object Access Protocol</i>
WSDL	<i>Web Service Description Language</i>
MPEG-4	<i>Moving Pictures Experts Group 4</i>

MP3	MPEG-2 Audio Layer III
OGG	Codec del Vorbis Ogg project
AAC	Advanced Audio Coding
AMR	<i>Adaptive Multi-Rate</i>
JPG	JPEG: Joint Photographic Experts Group
PNG	Portable Network Graphics
GIF	Graphics Interchange Format
SMS	Short Message Service
JDT	Java Development Tools
DDMS	Dalvik Debug Monitor Service
DVM	Dalvik Virtual Machine
AVDM	Android Virtual Devide Manager
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
SQL	Structured Query Language
SQLite	Light SQL
MS-SQL	Microsoft SQL
JVM	Java Virtual Machine
TITSA	Transportes Interurbanos de Tenerife S.A.
PHP	PHP Hypertext Pre-processor
GNU	GNU is Not Unix
COITT	Colegio Oficial de Ingenieros Técnicos de Telecomunicación
JSP	JavaServer Pages
StAX	Streaming API for XML
WiFi	Wireless Fidelity