



Memoria del proyecto

Programación de aplicaciones para dispositivos
móviles

Integrantes

Ángel Luis Ortiz Folgado

Óscar Eduardo Pérez la Madrid

Memoria del proyecto

Introducción

La aplicación, InfoMadrid, muestra información sobre distintos servicios de la ciudad de Madrid. En concreto localización de farmacias y de aparcamientos públicos, así como la ruta desde tu localización hasta allí. Para ello se obtiene la información del Catálogo de Información Pública del Sector Público¹.

Manual de Instalación

InfoMadrid, ha sido desarrollada usando el IDE Eclipse con el plugin de Android.

Para instalar el código sólo hay que importar el proyecto PadSearcher, mediante la secuencia: File->Import->Android-> Existing Android Code Into Workspace y en la ventana Root Directory seleccionar PadSearcher. Tal como se observa en la figura 1

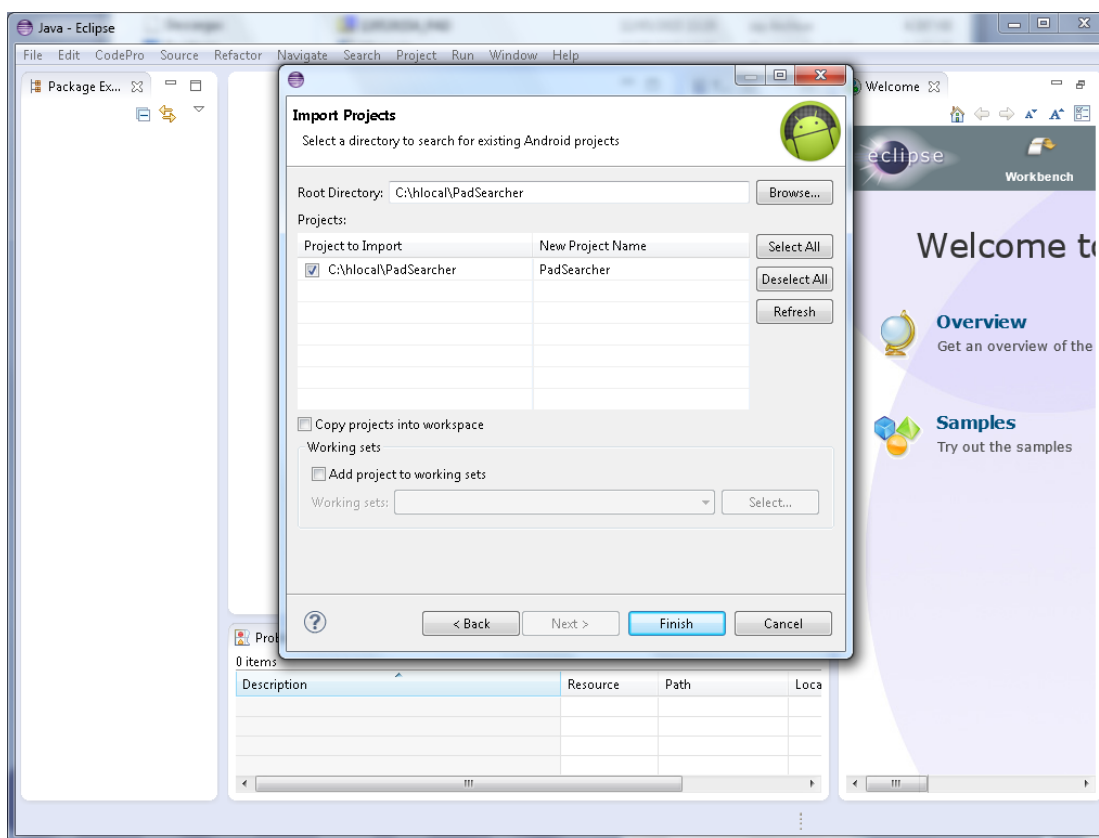


Figura 1: Instalación

Además hay que importar también, siguiendo los pasos anteriores la librería appcompat_v7.

A la hora de usar un emulador usamos uno con la versión de Android 4.4.2 y SOC Arm.

¹ <http://datos.gob.es>

Manual de Usuario

VISTA PRINCIPAL

En la figura 2 se muestra la vista principal de nuestra aplicación. Como se observa es una interfaz simple, que consta del título de nuestra aplicación y cuatro botones.

Los botones Aparcamientos y Farmacias al pulsar sobre ellos llevan a la vista del mapa, donde muestra la ubicación de los aparcamientos/farmacias respectivamente.

Los otros dos botones, Búsquedas recientes, muestran las últimas búsquedas realizadas en la aplicación, como se observa en la figura 3. Cada uno de estos botones está asociado al botón principal más cercano.

VISTA DEL MAPA

En las figuras 4.1 y 4.2 se muestra la vista del mapa. En esta vista se ve un mapa centrado en tu posición actual (punto azul), además de los botones de control típicos de Google Maps (zoom, Street View, cambio de modo de mapa), aparecen marcadores con la ubicación del servicio buscado (farmacias o aparcamientos). Al pulsar sobre uno de estos marcadores, aparece su nombre y su dirección (que se guardará en búsquedas recientes), y además trazará la ruta desde tu posición hasta ese lugar en el mapa (ver figura 5).

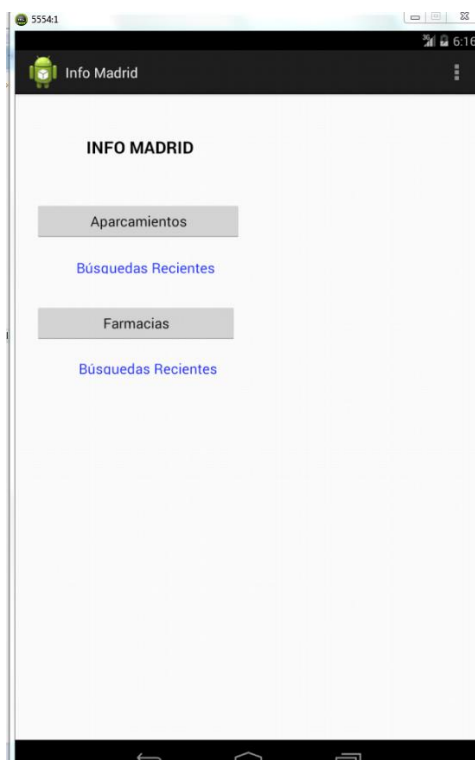


Figura 2: Pantalla inicio aplicación



Figura 3: Búsquedas recientes

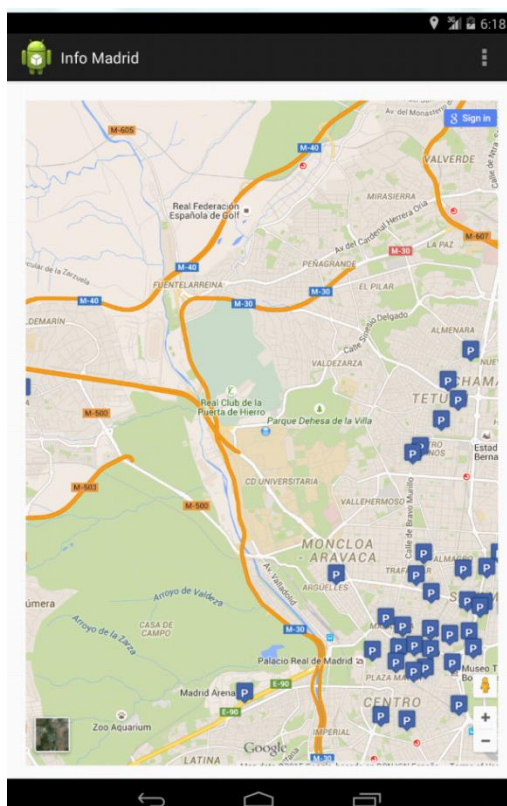


Figura 4.1: Vista mapa aparcamientos

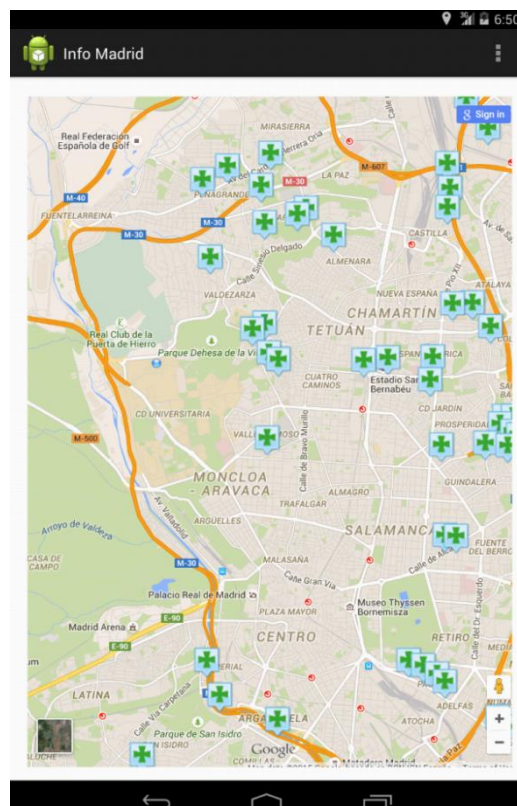


Figura 4.2: Vista mapa farmacias

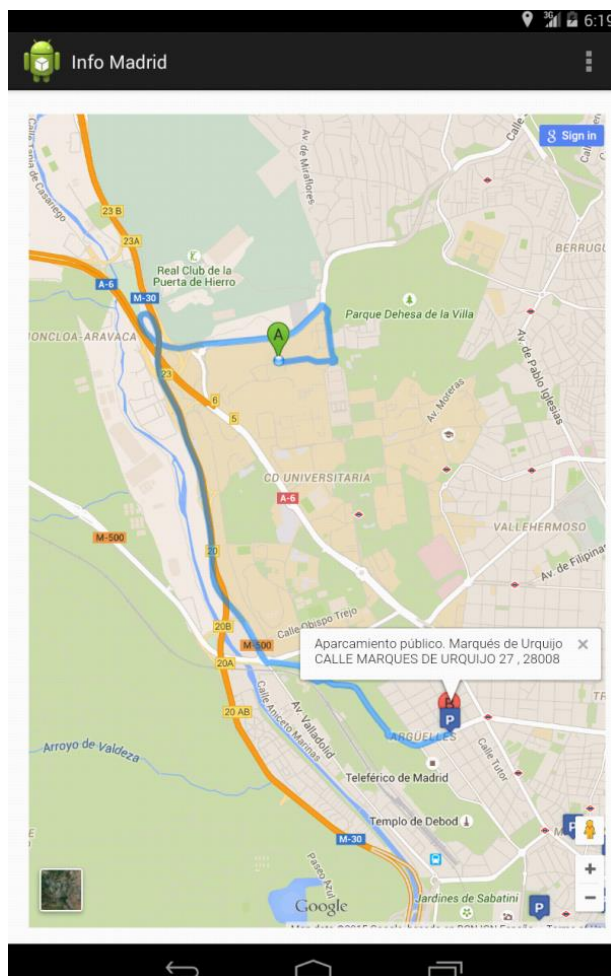


Figura 5: Generación de ruta

Diseño con modelos y comentarios

ARQUITECTURA APLICACIÓN

La aplicación consta de dos vistas, la principal (definida en `activity_formulario.xml`), que tiene asociado el formulario que contiene los botones, y la vista web (definida en `activity_parser.xml`), que contiene un `webview`, donde se muestra el mapa.

Cada una de esas vistas tiene asociado una clase java que se encarga de dotarlas de funcionalidad.

Formulario.java

Está asociada a la actividad formulario, implementa la funcionalidad a los botones, así como se encarga de recuperar la lista de visitas recientes del archivo correspondiente, para mostrarla en el diálogo que crea.

Parser.java

Asociada a la actividad parser, se encarga de crear el `webview`, y ejecutar la parte web, además parsea el xml adecuado, para obtener la información necesaria para mostrar en el mapa. También gestiona la geolocalización del usuario.

JSONTest.java

Clase auxiliar que crea un archivo JSON interpretable por JavaScript, con la información obtenida por el parser.

JavaScriptInterface.java

Esta clase sirve de puente entre la parte de Android nativo y la parte web.

Se encarga de pasar el objeto JSON con la información de los marcadores del mapa, así como las coordenadas en las que se encuentra el usuario.

Desde la parte Web, se encarga de recuperar la información de los marcadores seleccionados por el usuario y guardarlos en el archivo correspondiente (ya sean farmacias o aparcamientos), para que puedan ser recuperados en otro momento. Si el archivo no existe, lo crea.

Index.html

Muestra el mapa.

GoogleMap.js

Contiene los métodos JavaScript necesarios para mostrar el mapa.

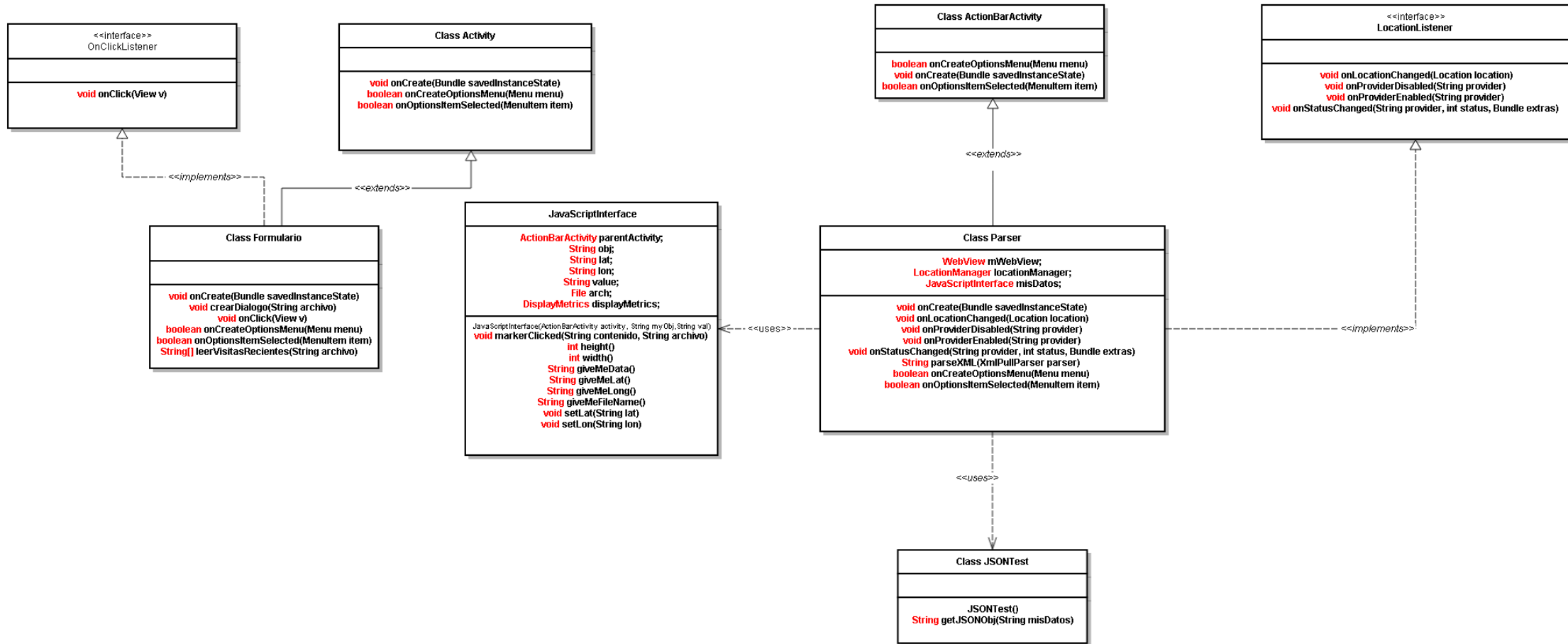


Figura6: Diagrama de clases

INTERACCIÓN CON LA APLICACIÓN

La aplicación tiene dos funciones básicas, mostrar el mapa adecuado, con la información requerida y mostrar la lista de búsquedas recientes.

Muestra del mapa

1. En el formulario principal se selecciona el botón del mapa que se desea visualizar.
2. Se llama a la actividad Parser.
3. En Parser, se trata el documento XML, para obtener la información.
4. La información obtenida se envía a JSONTest, para que cree el objeto JSON, interpretable por JavaScript.
5. Parser obtiene la geolocalización del usuario, y la envía junto con el JSON que ha recibido de JSONTest, a JavaScriptInterface, para comunicarlo a la parte web.
6. Parser carga el webview.
7. Index.html y GoogleMap.js, crean el mapa, centrado en las coordenadas recibidas y añaden los marcadores de posición del usuario, y de los establecimientos buscados.
8. Si en el mapa se pulsa sobre un marcador, GoogleMap.js se encarga de atender el evento mostrando la información asociada a dicho marcador. Esa información también se pasa a JavaScriptInterface, que se encargará de añadirla en un archivo, para su almacenamiento persistente. Finalmente GoogleMap.js, creará la ruta desde la posición actual hasta la posición marcador seleccionado, mostrándose en el mapa.

Mostrar lista de búsquedas recientes

1. En el formulario se pulsa sobre el botón de la lista que quieras visualizar.
2. Formulario, recupera la lista del fichero adecuado, de manera que quede ordenada de la búsqueda más reciente a la más antigua.
3. Formulario crea un diálogo donde se muestra dicha lista.

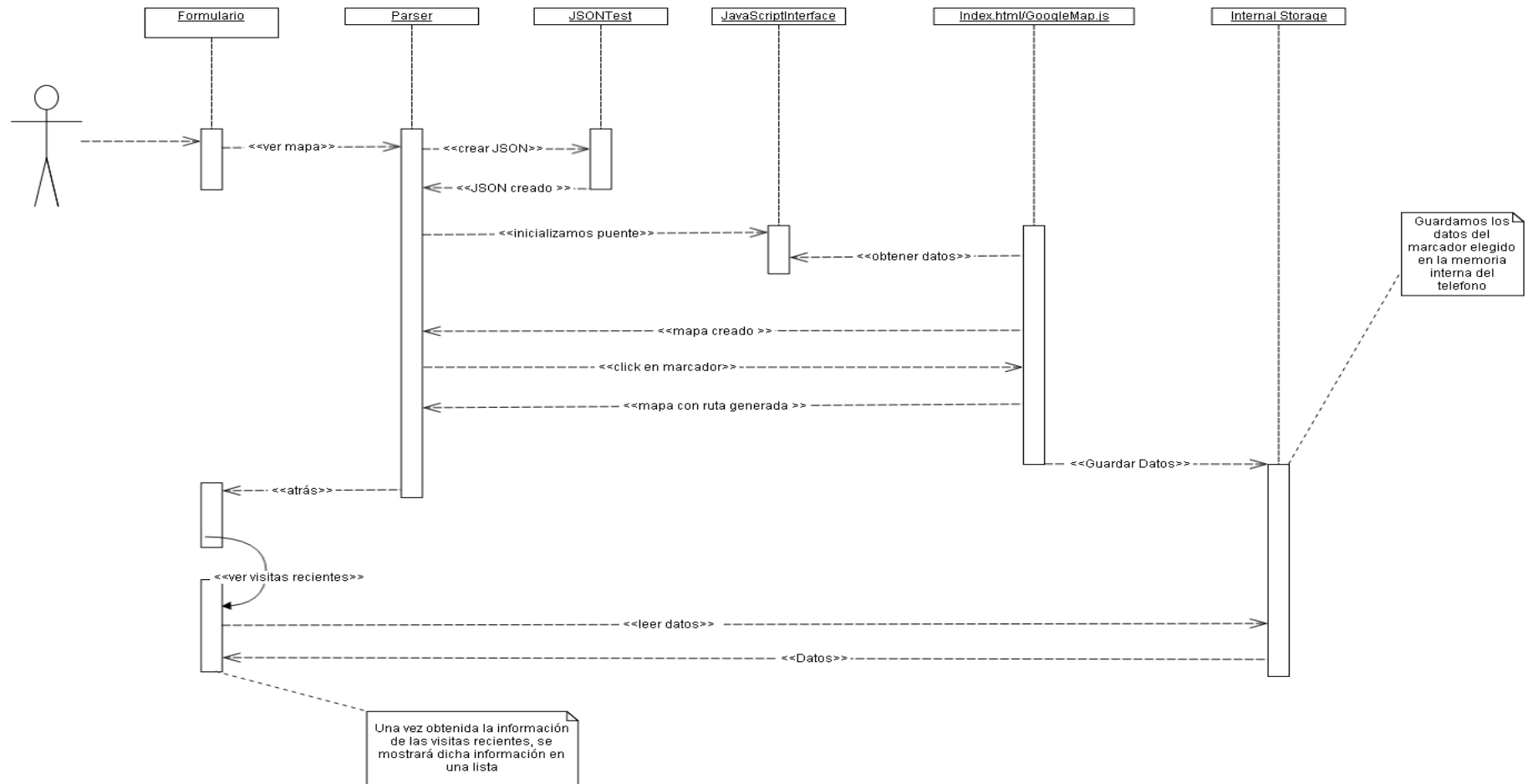


Figura7: Diagrama de Secuencia 1

Consideraciones técnicas fundamentales

En la realización de este proyecto se han encontrado las siguientes dificultades:

- A la hora de obtener la información para la generación de los marcadores mapa hubo que aprender a parsear un documento XML, para ello se hizo uso de la librería XMLPull².
- Una vez obtenidos los datos, se halló, el problema de comunicar los datos obtenidos a JavaScript. Para ello se creó una clase JavaScriptInterface, que hacía de puente entre ambas implementaciones.
- Otro inconveniente, fue la ejecución de la parte web en el dispositivo. En un principio usábamos un emulador con la versión 4.1.2 de Android, investigando descubrimos, que el motor webview implementado en esa versión, no ejecutaba de manera óptima JavaScript. Actualizamos el emulador a la versión 4.4.2, y encontramos problemas con la interacción con el mapa, lo solucionamos eligiendo como SOC del emulador ARM, en lugar de Intel atom.
- Otro escollo, encontrado durante la realización del proyecto, fue a la hora de obtener los datos de geolocalización. Para ello implementamos LocationListener y definimos los métodos que controlaban los eventos del GPS, pasando estos datos a la vista web mediante la interfaz creada anteriormente. También investigamos como pasar coordenadas al emulador, para poder realizar pruebas.
- Finalmente el último impedimento que se nos presentó fue la visualización del mapa, en pantallas de distintos tamaños, para solventarlo, definimos el ancho y el largo del mapa al 100% mediante el estilo, de esta forma queda ajustado a las dimensiones de la pantalla.

² <http://www.xmlpull.org/>