

INFORME GENERAL
[APP - SYSPARK]

Historia del Documento

Fecha	Versión	Comentarios	Autores
26 Enero 2016	1.0	Informe Final app android	Nelson Pupiales Santiago Mantilla Michael Gonzalez

Tabla de Contenidos

1. Vision General.
2. Pantallas.
3. Flujo de Comportamiento.
- 3.1. Mapa.
- 3.2. Lista.
- 3.2.1. Detalles.
- 3.2.1.1 Mapa Detalles.
- 3.3. Busquedas.
4. Bloqueos y Excepciones.
- 5 Velocidad de Aplicación.
- 6 Eventos
- 7 Implementacion.
- 8 Arquitectura de Aplicación.
9. Conclusiones.

1. Visión general.

El proyecto actual se engloba dentro del área de desarrollo de aplicaciones móviles pensado y desarrollado como el cliente de una arquitectura cliente - servidor, desarrollado en la plataforma Android Studio. De las diferentes propuestas del equipo se ha tomado como opción de el Gestor de Mapa y GPS para la plataforma Android.

La idea inicial se basa en desarrollar una aplicación para dispositivos Android, que permite al usuario geolocalizar, desde su posición actual, podrá seleccionar los diferentes parqueaderos que se encuentren a su alrededor los mismos que entregaran información si existen plazas disponibles.

Estos puntos se mostrarán como pines dentro del mapa y al pulsar en los mismos un cuadro de dialogo permitirá realizar las acciones oportunas: abrirá una pantalla la información del parqueadero antes seleccionado.

Inicialmente para facilitar la gestión de la aplicación el ámbito geográfico se limitará a la Ciudad de Loja.

2. Pantallas.

La aplicación cuenta con 6 (seis) pantallas para la interacción del usuario, estas son:

1. Pantalla de Carga
2. Pantalla de Principal
 - a. Pantalla del mapa Principal
 - b. Pantalla de Listado de parqueaderos
 - i. Pantalla de detalles del parqueadero
 1. Ubicación del parqueadero específico en el mapa.
 - c. Pantalla Buscador.

3. Flujo del comportamiento.

La aplicación inicia con una pantalla de carga con una barra de carga, luego se nos presenta la pantalla principal de nuestra aplicación, la cual consta de 3 (tres) botones los cuales son:

1. Mapa - esta opción nos llevará al mapa principal de nuestra aplicación dónde están identificados visualmente todos los parqueaderos disponibles, el número de plazas por cada parqueadero mostrado y nuestra aplicación actual.

2. Lista - en esta opción nos muestra una lista con todos los parqueaderos existentes en nuestra base de datos.
3. Buscar - nuestro buscador cuenta con una interacción gráfica ya que se cargan todos nuestros parqueaderos en un spinner (barra seleccionable superior) y el mapa cargado en la parte central de la pantalla se moverá dependiendo de nuestra selección.

3.1. MAPA.

La primera opción de nuestra aplicación es una pantalla con el mapa de la ciudad de Loja donde nos ubicara por medio de GPS en donde estemos ubicados y todos los parqueaderos que se encuentren en nuestra base de datos, cada uno de los marcadores mostrados en esta pantalla tienen asociados el nombre del parqueadero en cuestión y el número de plazas o de espacios disponibles en el parqueadero.

3.2. LISTA.

En cuanto a la lista, en esta pantalla nos muestra como dice su nombre un listado de todos los parqueaderos disponibles en nuestra base de datos con un resumen de cada ítem de esta lista, esta pantalla cuenta con un botón que nos permite actualizar la lista de manera rapida.

3.2.1. Detalles.

Cuando seleccionamos algún registro de la lista mostrada anteriormente automáticamente nos muestra una pantalla con todos los detalles disponibles de este parqueadero.

3.2.1.1. Mapa Detalles.

En la pantalla de detalles en la parte inferior izquierda la cual nos muestra la ubicacion en especifico del parqueadero mostrado anteriormente.

3.3 BUSCAR.

En la pantalla de búsqueda nos muestra un mapa con una barra en la parte superior de la aplicación (spinner), en la cual se muestran los parqueaderos existentes en nuestra base de datos y la interacción de estos consiste en que el mapa se ubicará en la dirección de un parqueadero cuando seleccionemos un elemento de la barra superior o spinner.

4. Bloqueos y excepciones.

Dentro de la aplicación se cuenta con dos partes que se ha creído indispensable para máxima eficiencia a la hora de utilizarla:

- a. No se tiene acceso a la página de pantalla de carga, la misma que es generada solo por primera vez cuando se inicia la aplicación.
- b. La página se mantendrá una posición fija la misma que evitará distorsión en la gráfica a la hora de ejecutar la aplicación.

5. Velocidad de la aplicación.

La velocidad de la aplicación dependerá de la conexión del internet. Los datos son consumidos de una base de datos que se encuentra en el servidor.

La velocidad de la conexión es estable con los consumos de datos se utiliza Json para la carga de la información se más eficiente dinámica y rápida en el tiempo de respuesta.

6. Eventos.

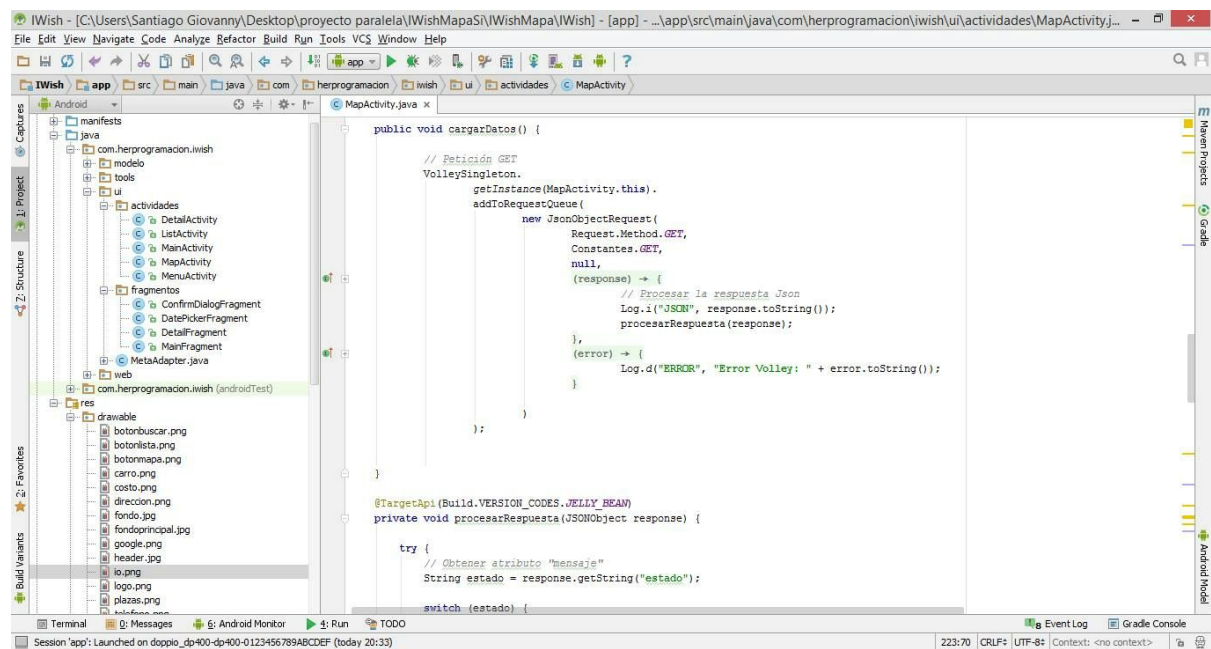
Los eventos principales de nuestra aplicación se ubican en la carga de los respectivos mapas los cuales son el mapa principal, el mapa para los detalles de cada registro e incluso en el mapa de búsquedas.

En el mapa principal hacemos que la vista del mapa se dirija a la posición actual del dispositivo que está usando la aplicación junto con un acercamiento o zoom del mapa que permita una visualización óptima, por parte de las búsquedas y del mapa de detalles hacemos que se muestre un zoom óptico y que el mapa se dirija hacia el primer o único marcador de la lista.

7. Implementación.

Durante esta fase se ha realizado la codificación de la aplicación intentando seguir el método de desarrollo ágil, creando múltiples iteraciones del producto hasta llegar al resultado final. Las diferentes iteraciones se han almacenado en un gestor de versiones GIT.

El código se ha desarrollado con la ayuda de la plataforma de desarrollo Android Studio (Android development tools)



8. Arquitectura de Aplicación.

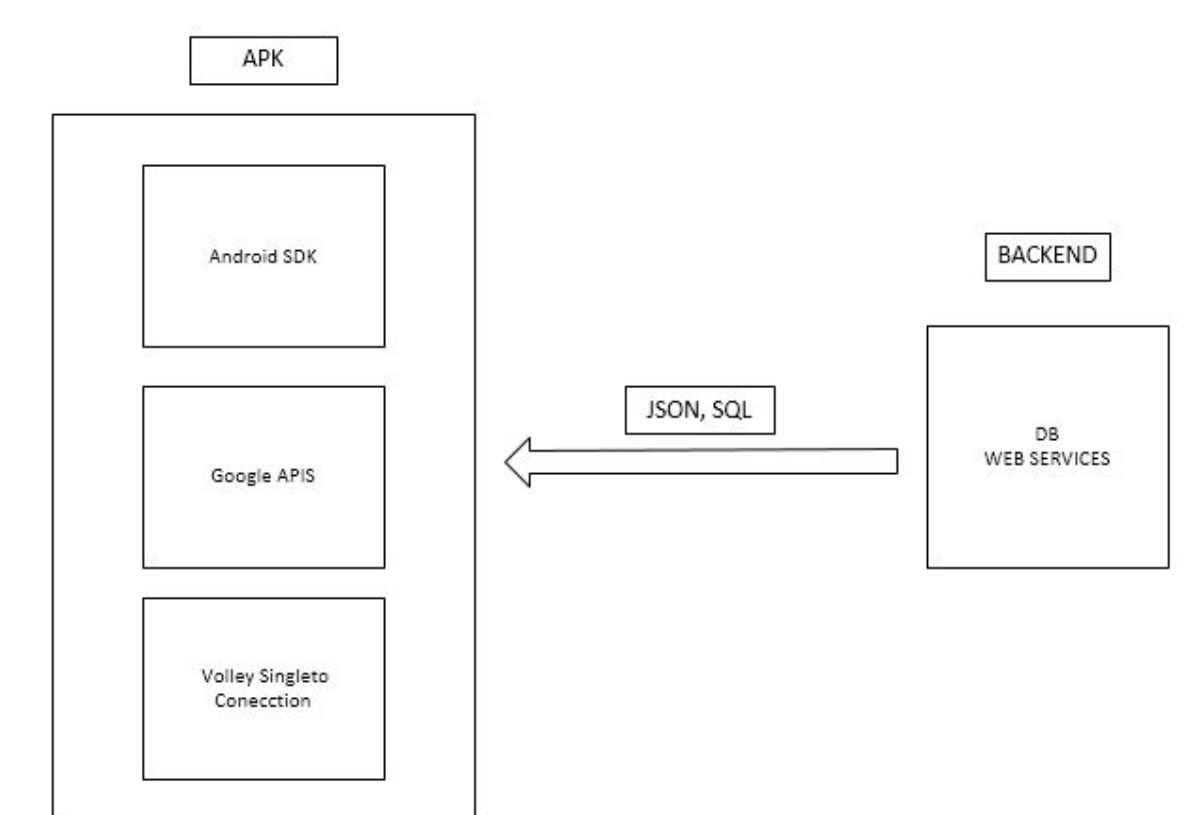
La documentación de arquitectura se ha propuesto hasta la fase de implementación, en la que finalmente se podrá mostrar en forma completa.

La aplicación consta de dos capas diferenciadas el cliente que se ejecuta en un dispositivo Android y la parte servidora, donde se alojan los datos de la aplicación.



Para realizar la aplicación Syspark se ha decidido prescindir del uso de un backend en lenguaje de servidor y base de datos para usar un tipo de servicio en la nube denominado Backend as a Service.

La estructura arquitectónica usada en la aplicación android es la siguiente:



Se hace uso de Android SDK y las librerías Google Apis para generar los mapas. La librería parse permite la comunicación con el Backend.

8. Arquitectura Cliente Syspark.

La aplicación se puede dividir en diferentes capas:

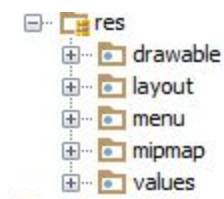


- La **capa de presentación** formada por los recursos textuales y la interfaz gráfica.
- Los **permisos** que permiten a la aplicación acceder a diferentes características del dispositivo.
- La **lógica de aplicación** formada por las diferentes clases que dan forma a la aplicación.

Esta separación de capas es la habitual en las aplicaciones para Android.

Capa de presentación

La capa de presentación se encuentra en el directorio \res de la aplicación



Dentro de las carpetas que se nombran como drawable encontramos las imágenes de las que hace uso la aplicación para las diferentes resoluciones de dispositivo. Las carpetas nombradas como values contienen documentos XML que incluyen los textos para poder internacionalizar la aplicación.


```

<resources>

    <!-- Base application theme. -->
    <style name="AppTheme.Base" parent="Theme.AppCompat.Light.DarkActionBar">
        <item name="colorPrimary">@color/primary</item>
        <item name="colorPrimaryDark">@color/primary</item>
        <item name="colorAccent">@color/accent</item>
        <item name="android:textColorSecondary">@color/textColorSecondary</item>
        <item name="colorControlHighlight">@color/accent</item>
        <item name="android:windowBackground">@color/background</item>
    </style>

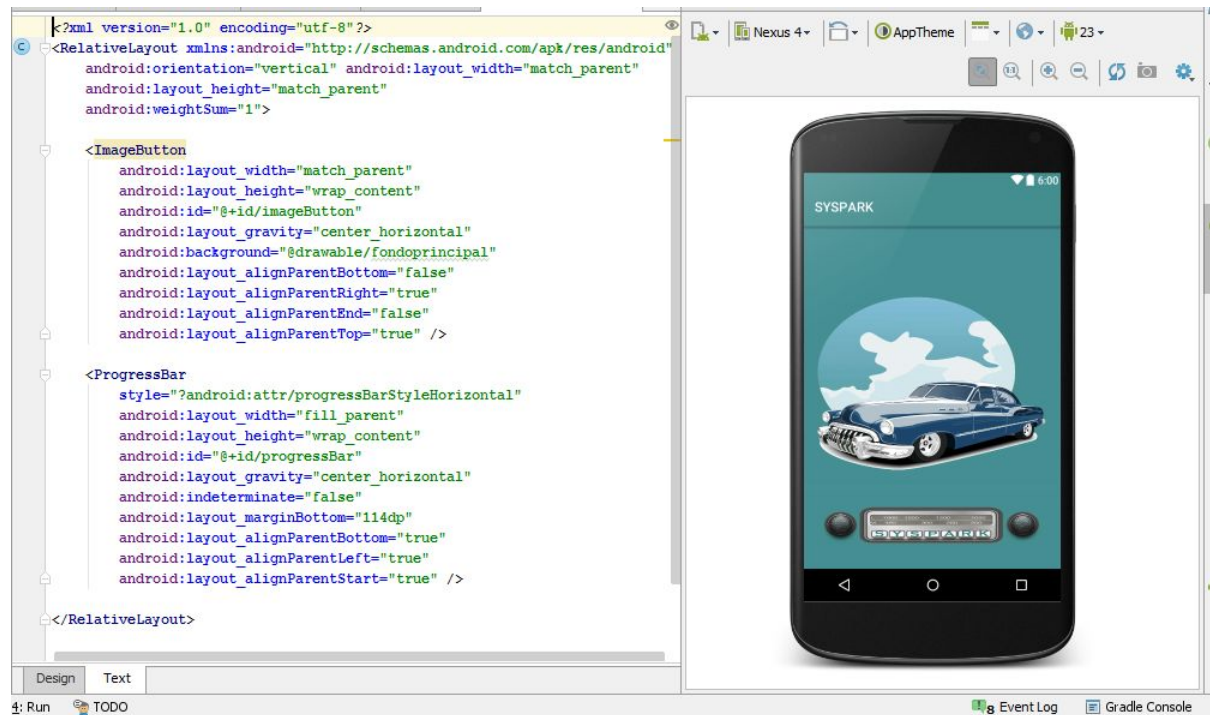
    <style name="AppTheme" parent="AppTheme.Base" />

    <style name="AppTheme.Detail">
        <item name="colorPrimaryDark">@android:color/black</item>
        <item name="windowActionBarOverlay">true</item>
        <item name="actionBarStyle">@style/ActionBar.Detail</item>
    </style>

    <style name="ActionBar.Detail" parent="Widget.AppCompat.Light.ActionBar.Solid">
        <item name="background">@null</item>
    </style>
</resources>

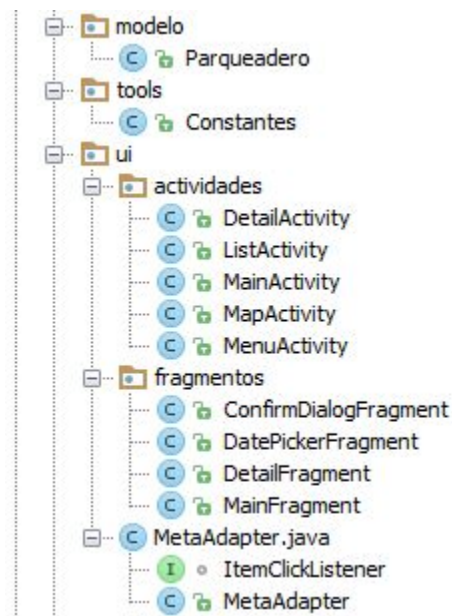
```

La carpeta layout contiene documentos XML donde se ha realizado el diseño de las diferentes actividades (pantallas) principales que contiene la aplicación.



Lógica aplicación

La lógica de la aplicación contiene todas las clases en lenguaje Java de las que hace uso la aplicación.



9. Conclusiones.

Las conclusiones que se pueden derivar de la realización de este trabajo de final de carrera son muy positivas.

Los requisitos iniciales que se presentaban tras la elección del proyecto se han cumplido en su mayoría si son revisados uno a uno, lo que se puede considerar un éxito.

El hecho de trabajar con la metodología de trabajo de diseño centrado en el usuario nos ha permitido mejorar los conocimientos y valorar positivamente este tipo de técnicas para incluirlas en futuros proyectos en los que pueda participar.

Utilizar nuevas tecnologías actuales como el uso de nuevo software, herramientas de test de usuarios o el propio desarrollo en Android que evoluciona constantemente ha supuesto un reto y una fuente de aprendizaje inimaginable con respecto a los objetivos iniciales.

Reiterar que el aprendizaje del conocimiento conseguido del desarrollo nativo en Android ha sido enriquecedor.

El resultado final no es perfecto, pero después de haber atravesado algunos problemas laborales, los resultados conseguidos ha sido satisfactorios.