

Universidad Autónoma Metropolitana Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

Reporte del Proyecto Terminal

“Visualización de Incidencias en la Ciudad de México”

Modalidad: Proyecto Tecnológico

Alumna: Montiel Saavedra Alejandra
209364851
al209364851@alumnos.azc.uam.mx

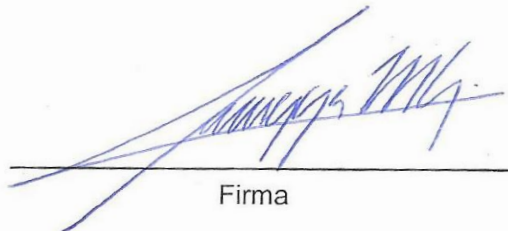
Asesor: Dr. Francisco Javier Zaragoza Martínez
Profesor Titular
Departamento de Sistemas
franz@correo.azc.uam.mx

Trimestre 17 Invierno

Fecha de entrega: 10 de abril del 2017

DECLARATORIA

Yo, Francisco Javier Zaragoza Martínez declaro que apruebo el contenido del presente reporte de Proyecto de Integración y doy la autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Firma

Asesor:

Dr. Francisco Javier Zaragoza Martínez
Profesor Titular
Departamento de Sistemas
franz@correo.azc.uam.mx

Yo, Alejandra Montiel Saavedra, doy mi autorización a la Coordinación de Servicio de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento para la Biblioteca Digital, así como en el Repositorio de UAM Azcapotzalco.



Firma

Alumno:

Alejandra Montiel Saavedra
Matrícula: 209364851
al209364851@alumnos.azc.uam.mx

RESUMEN

El presente reporte detalla el desarrollo del Proyecto de Integración Visualización de incidencias en la Ciudad de México.

El programa ha sido desarrollado desde cero, desde su diseño hasta la implementación de cada uno de los módulos en las que se puede dividir el trabajo: Implementación del diseño mediante Layouts; conexión a la base de datos y tratamiento de mensajes obtenidos en formato XML, otras funciones como pueden ser implementar mapas de Google o añadir listas dinámicas.

La aplicación trabaja para dispositivos móviles con sistema operativo Android, la finalidad es visualizar la información de incidencias en la Ciudad de México; por medio de tres opciones las cuales son:

- Consultar por medio de las coordenadas.
- Consulta por el rango de fecha.
- Consulta por tipos de incidencias.

Dependiendo la opción que se elija será la forma en la que la aplicación mostrara los datos al usuario.

TABLA DE CONTENIDO

1. INTRODUCCIÓN	1
2. ANTECEDENTES	2
2.1. Proyectos terminales	2
2.2. Software	2
2.3. Tesis	4
3. JUSTIFICACIÓN	4
4. OBJETIVOS	5
4.1. Objetivo General	5
4.2. Objetivos Específicos	5
5. MARCO TEÓRICO	5
5.1. Sistema operativo Android como plataforma de desarrollo de una aplicación	5
5.2. XML	6
5.3. Componentes principales de una aplicación Android	7
6. DESARROLLO	8
6.1 Descripción de las actividades a detalle	8
6.1.1 Actividad: IntroActivity.java	9
6.1.2 Actividad: Menu.java	11
6.1.3 Actividad: SamplesAdapter.java	12
6.1.4 Actividad: ListaDeIncidencias.java	13
6.1.5 Actividad: Mapa.java	16
6.1.6 Actividad: RangoDeFecha.java	20
6.2 Modificaciones en el AndroidManifest.xml de todas las actividades	22
7. RESULTADOS	24
8. CONCLUSIONES	25
9. GLOSARIO	26
10. APÉNDICES	28
10.1 Clases de actividades	28

i.	IntroActivity.java	28
ii.	Información.java	31
iii.	Menu.java	31
iv.	SamplesAdapter.java	32
v.	ListasDeIncidencias.java	33
vi.	ListViewAdapter.java	36
vii.	VariablesDeIncidencias.java	38
viii.	XMLParser.java	39
ix.	Mapa.java	41
x.	RangoDeFecha.java	51
xi.	ListViewAdapterRangoFecha.java	53
10.2	Directorio res/layout (clases de vista).	56
i.	activity_intro.xml	56
ii.	slide1.xml	57
iii.	slide2.xml	58
iv.	slide3.xml	59
v.	menu.xml	60
vi.	portada_autora.xml	61
vii.	listview_incidencias.xml	63
viii.	listview_item.xml	63
ix.	mapa_coordenadas.xml	65
x.	listview_main_rango_fecha.xml	64
xi.	listview_item_rango_fecha.xml	67
10.3	Otros	68
i.	AndroidManifest.xml	68
ii.	colors.xml	69
iii.	google_maps_api.xml	69
iv.	strings.xml	70
v.	styles.xml	70
10.4	Gradle Scripts	71
i.	build.gradle (Dependencias de la aplicación)	71
11.	RECURSOS	72
12.	REFERENCIAS BIBLIOGRÁFICAS	73

INDICE DE FIGURAS

Figura 1. Aplicación “AGUMovil”	3
Figura 2. Aplicación “Mi Policía”	3
Figura 3. Aplicación “El botón de Pánico Jalisco”	4
Figura 4. Sistema de capas del sistema operativo Android	6
Figura 5. Los tres módulos de la aplicación	8
Figura 6. Diseños de slide1.xml, slide2.xml y slide3.xml	9
Figura 7. Diseño de activity_intro.xml	10
Figura 8. Diapositivas de la Introducción	10
Figura 9. Menú	11
Figura 10. Información de la aplicación	11
Figura 11. Dependencia de RecyclerView	12
Figura 12. Búsqueda de la lista de incidencias	13
Figura 13. Diálogo de progreso	14
Figura 14. Diseño de listview_item.xml	15
Figura 15. SDK Manager	16
Figura 16. Google Play Services	16
Figura 17. Dependencia de google-play-services-places	17
Figura 18. Google Console Developers	18
Figura 19. Polyline: Rectángulo	18
Figura 20. Marcadores de tipos de incidencias	19
Figura 21. Diseño de listview_item_rango_fecha.xml	20
Figura 22. Rango de fecha	21
Figura 23. Permisos en el archivo AndroidManifest.xml	22
Figura 24. Icono de la aplicación	23

1. INTRODUCCIÓN

En ocasiones el transcurso normal de las situaciones o eventos de la vida diaria se ven interrumpidos por situaciones que afectan su cotidianidad, a estas situaciones las llamamos incidencias.

En la mayoría de los casos se presentan sin aviso y en otras son el resultado de la indiferencia, descuido, o deterioro de un sistema o evento, como por ejemplo, las incidencias presentadas en una ciudad de un gran peso demográfico como lo es la Ciudad de México, ejemplos bastos se pueden encontrar en casi cualquier momento y rubro. Fallas del sistema eléctrico, sistemas de iluminación con fallos recurrentes, bloqueos de drenaje, limpieza de calles irregular, vialidades saturadas, puntos ciegos en el patrullaje de elementos de seguridad y así podremos enumerar más incidencias que obstruyen el paso normal de la vida citadina.

Los datos abiertos son los que cualquier persona es libre de utilizar, reutilizar y redistribuir. Uno de los datos abiertos que en mayor medida beneficia a la ciudadanía son los de incidencia, estos datos son adquiridos e implementados en algunas herramientas tales como aplicaciones móviles como por ejemplo La app “Mi Policía”, desarrollada por la Secretaría de Seguridad Pública que permite a la ciudadanía ubicar el cuadrante de la ciudad en que se ubica el usuario, y contactar al policía asignado como jefe de cuadrante para atención a emergencias e incidencias. Esto ha permitido una disminución considerable en los tiempos de respuesta por parte de la policía (actualmente se atienden las emergencias en menos de tres minutos) y focalizar el servicio en zonas de alta incidencia delictiva [1].

El artículo “Ciudades inteligentes” de la revista ¿Cómo ves? aborda los sistemas que pueden proporcionar miles de datos por segundo para prever o mitigar problemas. Aprovechamiento de herramientas digitales y plataformas interactivas, conexión del gobierno y los ciudadanos, realización de trámites por internet. Esta propuesta coincide en la utilización de un conjunto de datos para generar información con el uso de medios interactivos [2].

Como hemos visto existe una gran cantidad de tipos de incidencias que ocasionan alteraciones en el curso de las vidas de los ciudadanos. En respuesta proponemos nuestra aplicación, la cual trata de ser una herramienta donde el ciudadano pueda obtener un registro en tiempo real de las incidencias que lo aquejen en ese momento dando como resultado una mínima afectación de sus actividades.

2. ANTECEDENTES

A continuación se presentan varios proyectos relacionados con el proyecto de la Visualización de Incidencias en la Ciudad de México, por lo tanto, aquellos están clasificados en: proyectos terminales, software y tesis.

2.1. Proyectos terminales

“Sistema de gestión escolar en Android” es muy similar pues incluye la información académica del alumno que se encuentra en la plataforma en línea Sistema de Integración e Información UAM (SIIUAM), a diferencia de esta propuesta nuestro contenido son datos abiertos sobre incidencias, sin embargo esta propuesta busca poner a disposición de los usuarios la información académica relativa a horarios y UEA de manera inmediata, además de que proporcionará a los alumnos la posibilidad de generar esquemas de horarios para inscripción, así como de orientarlos en la elección de un área de concentración al finalizar la carrera algo similar a la manipulación y gestión de los datos de incidencia pudiendo hacer incluso graficas sobre las incidencias y otorgando al usuario información detallada para trazar un plan, ruta o simplemente en lo que más le convenga usar dicha información. Con esto concluimos que a grandes rasgos la diferencia de más peso entre la propuesta y el antecedente son los datos manejados [3].

“Aplicación para dispositivos con Android que encuentre la mejor ruta entre dos estaciones del metro” guarda relación con esta propuesta en el sentido de que ambas tienen el objetivo de facilitar el traslado de los usuarios en la Ciudad de México, tanto en tiempo y distancia, trazando la mejor ruta o alertando de incidencias para poder diagnosticar ésta [4].

2.2. Software

“AGUMóvil” es una plataforma de vanguardia, que da información en tiempo real de afectaciones viales a través de un mapa interactivo de la Ciudad de México, podemos crear rutas para facilitar nuestra movilidad como se muestra en la figura 1. La relación que existe entre este software y mi propuesta es que va enfocado para la mejor movilidad vial del usuario teniendo a su disposición datos sobre las incidencias viales y su estatus del tiempo real [5].



Figura 1. Aplicación “AGUMóvil”

“Mi policía” es una herramienta que forma parte de un programa integral de seguridad pública, denominado Programa de Cuadrantes (Policía de Proximidad), que consiste en realizar una división territorial de la ciudad en 847 cuadrantes, delimitados por factores geográficos, incidencia delictiva, vialidades, habitantes y población flotante. La aplicación integra la tecnología a la seguridad pública, permitiendo a los ciudadanos, de manera gratuita, acceder a información del cuadrante en el que se ubica, al contacto del policía más cercano a su ubicación, la posibilidad de realizar llamadas de emergencia y tener una respuesta más rápida de la policía como se muestra en la figura 2 [6].

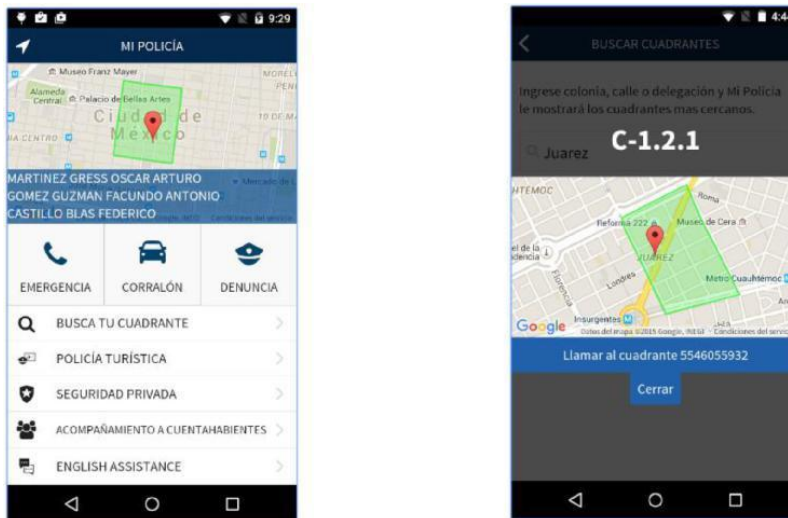


Figura 2. Aplicación “Mi Policía”.

Ambas aplicaciones utilizan la base de datos de incidencias en seguridad pública, aunque a diferencia de esta propuesta, nuestra app gestiona los datos que “Mi policía” va recabando y alimentando a dicha base de datos.

“El Botón de Pánico Jalisco” es una aplicación para dispositivos móviles del Estado de Jalisco que permite que el usuario registrado pueda enviar una alerta inmediata al Centro Integral de Comunicaciones (Ceinco 066), en donde un operador calificado recibe el

caso, efectuando todos los pasos establecidos para dar curso a la situación y resolver el problema de manera rápida y efectiva. La relación que existe entre este software y mi propuesta es la similitud entre la forma en que se enlazan a centrales de datos gubernamentales recibiendo y compartiendo información como se muestra en la figura 3. [7].

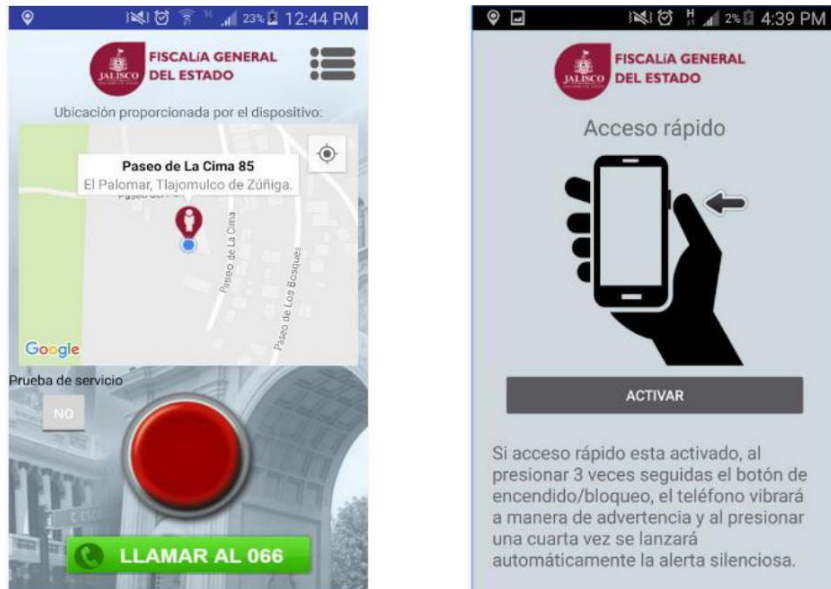


Figura 3. Aplicación “El botón de Pánico Jalisco”.

2.3. Tesis

“La Utilización de servicios Web en dispositivos móviles Android” es una la tesis que como la propuesta coinciden en el desarrollo de una aplicación Android© para dispositivos móviles con el fin de explotar servicios web, aunque la tesis ensancha las funcionalidades de la aplicación con el objetivo de integrar varias tecnologías que en la propuesta no están incluidas, como geolocalización y recuperación de información para fines comerciales [8].

3. JUSTIFICACIÓN

En la Ciudad de México las incidencias han llegado a ser de magnitudes descontroladas y por consiguiente vemos a muchas personas afectadas por ello. Ya no es raro encontrar embotellamientos kilométricos o transporte público tan deficiente que día a día es un reto que la gente llegue a sus trabajos o destinos en tiempo y forma, las inundaciones son cotidianas e incrementan año tras año, iluminación y seguridad son otro rubro donde el gobierno se ve sobrepasado y asaltos, robos y otros delitos son cada vez más comunes dentro de nuestro círculos sociales, por esta razón nuestro tiempo nos exige una estrategia más inteligente para resolver estas situaciones.

Nuestra propuesta es una herramienta digital que proporciona al usuario datos, registros y estadísticas que le permitan superar o al menos confrontar de una manera más equilibrada a todo este conjunto de incidencias.

Si bien esta propuesta no busca acabar con las incidencias de manera directa, si trata de ser una aportación eficaz para el usuario de la Ciudad de México que necesite estar al tanto de ellas y mejorar tanto su movilidad como su seguridad dentro de la ciudad.

4. OBJETIVOS

4.1. Objetivo General

- Diseñar e implementar una aplicación móvil para que la ciudadanía visualice la información de incidencias en la Ciudad de México.

4.2. Objetivos Específicos

- Diseñar e implementar el módulo de lectura de datos abiertos de las incidencias
- Diseñar e implementar el módulo de la consulta de usuario.
- Diseñar e implementar el módulo de la visualización de respuesta.

5. MARCO TEÓRICO

A continuación se presenta el marco teórico del proyecto Visualización de Incidencias en la Ciudad de México, en esta parte se plantea la definición del sistema operativo Android que es el sistema operativo para el cual se desarrolla el proyecto.

5.1 Sistema operativo Android como plataforma de desarrollo de una aplicación

Android es un sistema operativo basado en el núcleo Linux, fue diseñado para dispositivos móviles tales como teléfonos inteligentes y tabletas. Android permite programar en un entorno de trabajo (framework) de Java.

Además, Android cuenta con aplicaciones Google, que permiten acceso a Google Maps, YouTube, Gmail, y muchas más aplicaciones oficiales con solo ingresar una cuenta Google [9].

Las cuatro capas que forman la estructura del sistema operativo Android como se muestra en la Figura 4 son las siguientes:

- Aplicaciones: Se encuentran en el primer nivel contienen las aplicaciones incluidas en Android, un navegador, un cliente de correo electrónico, un calendario, mapas y todas las aplicaciones que el usuario instale en su dispositivo.
- Framework: Esta segunda capa proporciona el conjunto de herramientas que nos van a permitir desarrollar una aplicación.
- Librerías y máquinas virtual: Esta capa incluye tanto las librerías nativas (bibliotecas de Java) como la máquina virtual.
- Kernel de Linux: El Kernel es el que hace de intermediario entre el software y el hardware. Además se encarga de gestionar los mecanismos de protección de accesos a la memoria central y de administrar las interrupciones, excepciones y la comunicación entre procesos.



Figura 4. Sistema de capas del sistema operativo Android.

Durante el desarrollo de este proyecto, se utilizó la versión 4.4.2 de Android conocida como Android KitKat [10], la cual cuenta con las siguientes características:

- Pantalla completa para gran parte de las aplicaciones.
- Se incluye una opción de autocompletado más eficiente.

5.2 XML

Para poder programar en Android se necesitan unos ciertos conocimientos sobre la estructura del lenguaje de marcado extensible Markup Language (XML). Android utiliza el lenguaje XML para definir toda la parte del diseño de la interfaz que contiene una aplicación. Toda la forma gráfica de la interfaz, audio, botones, imágenes, controles, etc., es indicada en un fichero XML y cargada posteriormente en una clase (.java) donde se le asigna de funcionalidad.

Se puede seguir que la estructura que sigue Android a la hora de disponer sus herramientas para desarrollar interfaces es la del Modelo Vista Controlador (MVC) [11]. El principal objetivo del MVC es separar completamente los datos de la aplicación de la forma visual de ésta. El MVC se divide en tres partes básicas:

1. *Modelo*: Son las representaciones sobre los distintos modelos de almacenamiento de la información que utiliza una aplicación Android, como base de datos, web services, etc.
2. *Vista*: Es la apariencia básica que tiene una aplicación Android.
3. *Controlador*: Son las clases de java (.java) que se encargan de relacionar la información con la apariencia gráfica. El controlador se encarga por tanto de integrar el modelo con la vista.

5.3 Componentes principales de una aplicación Android

Una vez descrito el funcionamiento del sistema Android, se describen, a modo de introducción, los componentes o clases principales que aparecen con mayor frecuencia en una aplicación Android.

Activity: La clase Activity (actividad) representa el componente principal de la interfaz gráfica de una aplicación Android. Las Activities pueden verse como las distintas pantallas o ventanas de las que se compone una aplicación.

Service: La clase Service (servicios) son componentes que no presentan interfaz gráfica y que funcionan en segundo plano. Los objetos Service son utilizados para crear tareas específicas como el lanzamiento de notificaciones, la actualización de datos de la interfaz principal, etc.

View: La clase View representa el componente básico en la que se apoyan todos los elementos que construyen una interfaz. En Android existen multitud de clases para generar interfaces. Todas las clases que generan una interfaces, como botones o etiquetas de texto, heredan directa o indirectamente de la clase View.

Content Provider: La clase Content Provider representa el mecanismo básico para compartir datos entre aplicaciones Android. Mediante los Content Provider, es posible utilizar datos determinados en diversas aplicaciones sin necesidad de mostrar detalles sobre la estructura o su almacenamiento interno.

Broadcast Receiver: Esa clase representa el componente destinado a detectar y reaccionar ante determinados eventos que genera el sistema operativo Android. Algunos de esos eventos pueden ser la conexión a una red Wi-Fi, la entrada de una llamada, la llegada de un SMS, etc [12].

Intent: La clase Intent representa el componente de comunicación entre objetos de clase Activity u objetos de clase Service. Esta clase permite la transferencia de datos entre Activities o Services. También permite iniciar nuevas Activities o incluso iniciar aplicaciones externas.

6. DESARROLLO

Primero que nada presento el diagrama de componentes compuesto por tres módulos: 1) De consulta de usuario; 2) De lectura de datos abiertos; y, 3) De visualización de respuesta en la Figura 5.

1.- Módulo de consulta de usuario.

Su objetivo es obtener la consulta. Su entrada viene de una interfaz gráfica que permite escoger un rango de coordenadas, un rango de fechas y los tipos de incidencia de interés.

2.- Módulo de lectura de datos abiertos.

Su objetivo es generar los datos abiertos a partir de las solicitudes de los usuarios obteniendo la información de los archivos de datos abiertos. Su entrada son un rango de fechas, área geográfica y tipo de incidencia y su salida es la lista de incidencias que cumplan esas características

3.- Módulo de visualización de respuesta.

Este módulo será el encargado de mostrar la información requerida por el usuario. Su entrada es una lista de incidencia de interés y su salida es el mapa de incidencias.

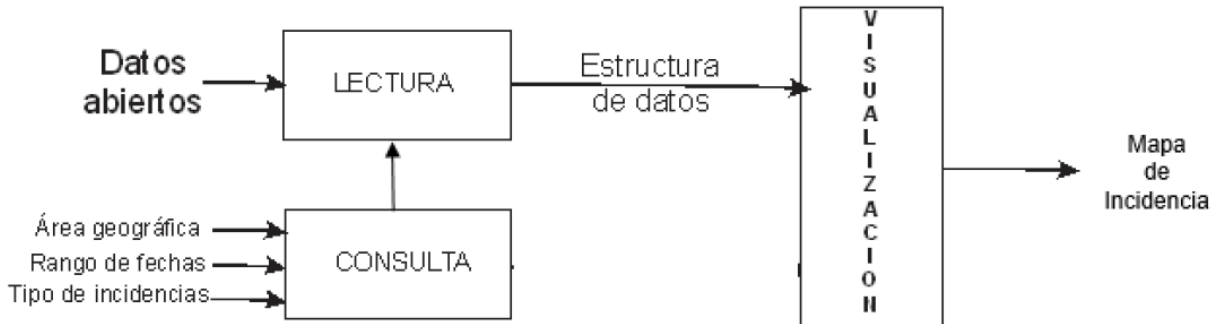


Figura 5. Los tres módulos de la aplicación.

6.1 Descripción de las actividades a detalle.

Se creó un nuevo proyecto en Android Studio, donde rellenaré los detalles y el nombre del proyecto Visualización De Incidencias.

- Nombre de la Aplicación: VisualizaciónDeIncidencias
- Nombre del Proyecto: VisualizaciónDeIncidencias

Comenzaremos abriendo las siguientes actividades:

6.1.1. Actividad: IntroActivity.java

Esta actividad es la actividad inicial de la aplicación, es como si fuera la pantalla de bienvenida o introducción de la aplicación.

Esta contiene tres diapositivas de introducción e incluye los botones “siguiente” y “omitir” para la navegación dentro de la aplicación. El usuario puede navegar por cada diapositiva con un gesto de desplazamiento (tocando la pantalla táctil) o con el botón siguiente.

`colors.xml` situado bajo `res -> value` añade los valores de color a las diapositivas por lo que puede cargarlos fácilmente en la actividad.

Posteriormente, `string.xml` situado bajo `res -> value` añade los siguientes valores de cadena (texto) para mencionar un título y una descripción para cada diapositiva.

Así que hice tres diseños separados XML denominados `slide1.xml`, `slide2.xml` y `slide3.xml` bajo `res -> layouts` para tres diapositivas (Ver Figura 6.). El diseño de cada diapositiva contiene las imágenes, textos y los colores.

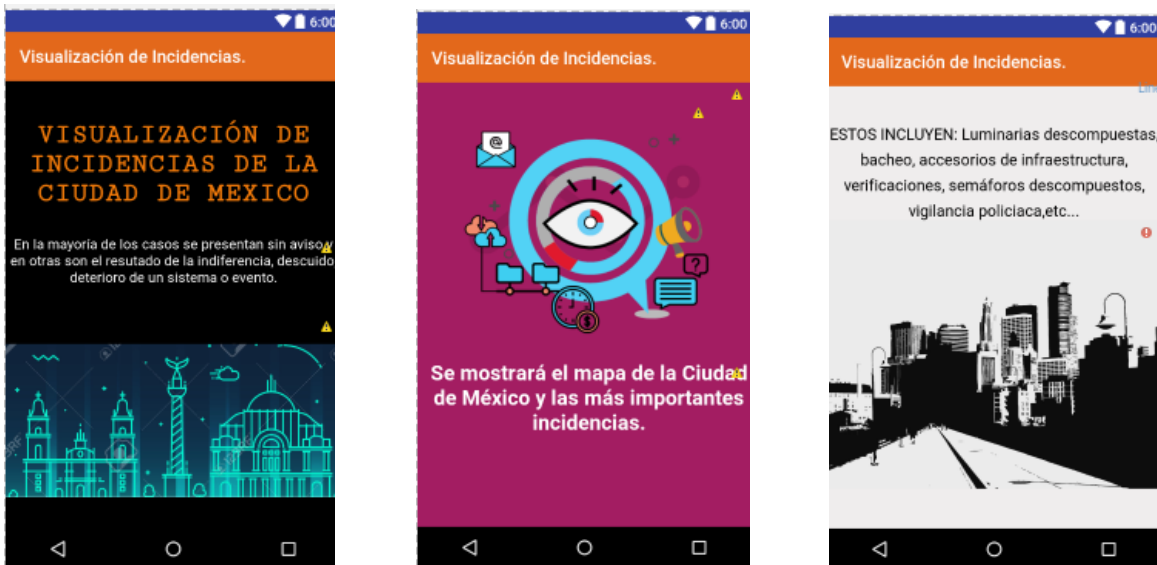


Figura 6. Diseños de slide1.xml, slide2.xml y slide3.xml

Una vez que los diseños están listos, crear una nueva actividad denominada `IntroActivity.java` para activar todas las funciones de navegación necesarias en la parte de bienvenida de la aplicación..

En el diseño o layout de `activity_intro.xml` que contienen ViewPager de control deslizante, LinearLayout de puntos de fondo y dos botones para Omitir y Siguiente (Ver Figura 7).

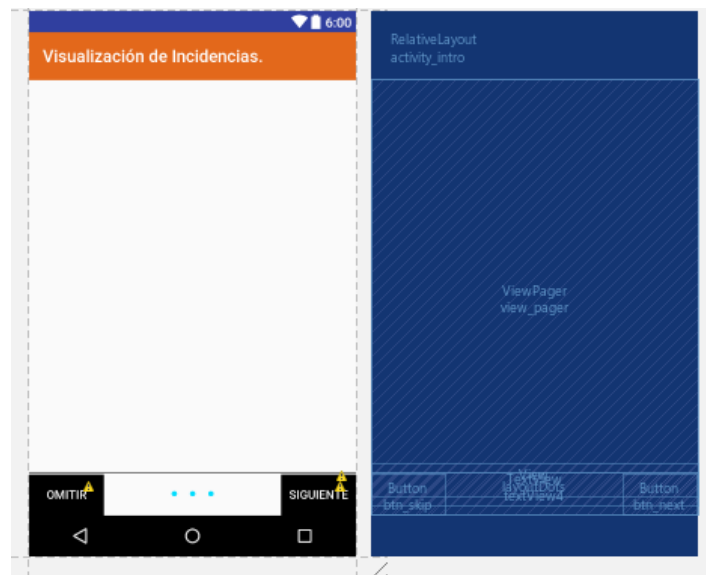


Figura 7. Diseño de activity_intro.xml

En la Figura 8, codifique en la clase de IntroActivity.java donde me he ocupado de pocas cosas tales como:

- La Creación de un ViewPager la cual visualizara todas las pantallas (diapositivas) que he creado antes.
- El Añadir los eventos “Omitir”, “Siguiete” y “Empezar”. Si se hace clic en el botón “siguiete”, se mostrará la siguiente diapositiva. Si se hace clic en el botón “Omitir”, la actividad principal se iniciará directamente.



Figura 8. Diapositivas de la Introducción

6.1.2. Actividad: Menu.java

En el botón de Empezar en el diseño de *activity_intro.xml* se iniciará el proceso *menu.xml* ya que contienen ListView y dos iconos de información y otro de Salir de la aplicación como se muestra la figura 9.

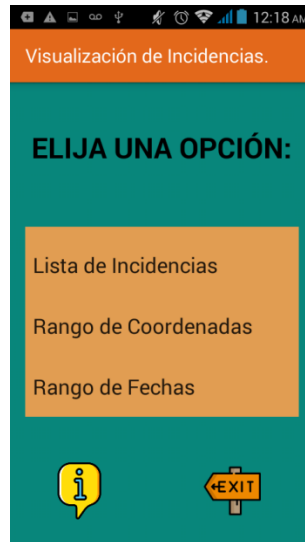


Figura 9. Menú

En la parte inferior izquierda muestra el icono información entonces al oprimirse comienza el proceso denominado `id = info`, este redirecciona a la actividad de *portada_autora.xml* que muestra la pantalla que contiene el título, versión, el logotipo, y fecha de creación de la aplicación (Ver Figura 10)

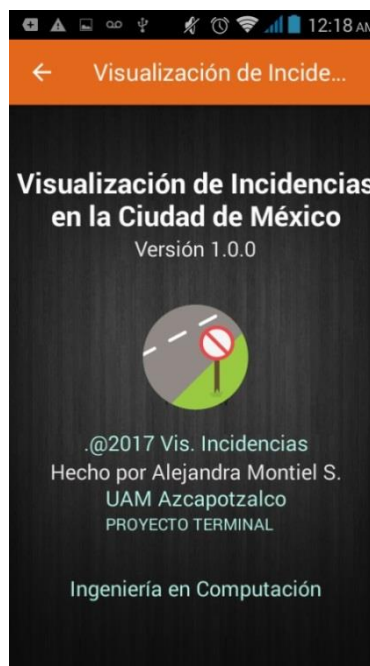


Figura 10. Información de la aplicación

Y en la parte inferior derecha muestra el icono de Exit para salir de la aplicación mediante:

```
salir.setOnClickListener(view)->{
finish();
Intent intent = new Intent(Intent.ACTION_MAIN);
intent.addCategory(Intent.CATEGORY_HOME);
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
startActivity(intent);
});
```

Sin embargo, creando la clase de `Menu.java` codifique:

- RecyclerView (sucesor de ListView con id = lista) para manejar la recopilación de datos de las tres clases:
 - `ListasDeIncidencias.java`
 - `Mapa.java`
 - `RangoDeFechas.java`

Y vincularla para que se muestre en la aplicación. También comienza aplicar el patrón de ViewHolder.

6.1.3. Actividad: SamplesAdapter.java

- Creación y Detección de clics SamplesAdapter (`SamplesAdapter.java`) en ListView tiene una `View.OnClickListener` interfaz para vincular a los eventos de clic para los elementos individuales de la lista (ListView: id=lista). En el método de `onBindViewHolder` es para que actualice los `RecyclerView.ViewHolder` contenidos con el elemento en la posición dada. Y el método importante de `getItemCount` es el que devuelve el número total de elementos de conjunto de datos almacenados en el adaptador.
- Añadido de eventos de clic oyente a dos iconos de Información y Salir.

NOTA: Modificar la biblioteca de soporte RecyclerView en las dependencias en su `app/build.gradle` como se muestra en la Figura 11.



Figura 11. Dependencia de RecyclerView

6.1.4. Actividad: ListaDeIncidencias.java

En la Figura 12. comenzaré describiendo el xml correspondiente (`listview_incidencias.xml`) este contienen los siguientes controles: EditText, ListView y ScrollView.

Implementaré una funcionalidad de búsqueda en el analizador de XML que filtra textos en la aplicación Android.

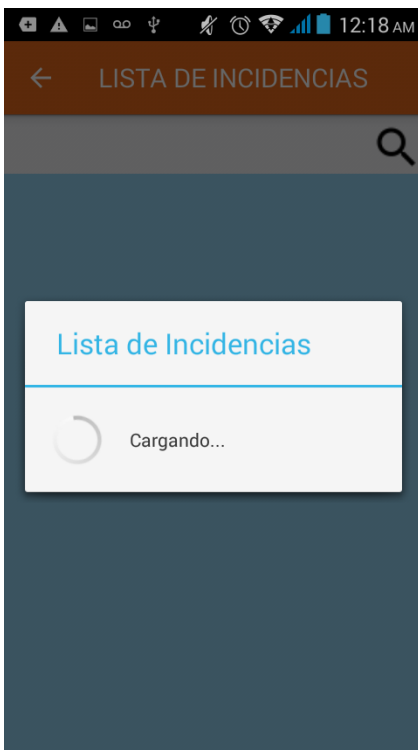


Figura 12. Búsqueda de la lista de incidencias.

Un listview puede ser filtrado por la entrada del usuario está habilitado usando el método `addTextChangedListener`. La función de búsqueda filtrará la vista de lista con una cadena correspondiente de la entrada del usuario.

La búsqueda a través del listview proporciona a los usuarios una manera fácil de encontrar la información que necesitan. Crearé un listview con un EditText colocado en la parte superior y en la entrada de Texto se filtrará los resultados de los textos en ListView. Analizaré la lista de incidencias de una URL utilizando un analizador XML ([ListasDeIncidencias.java](#)).

En la figura 13. del dialogo de progreso, he creado un AsyncTask como una tarea en segundo plano para cargar los datos o nodos XML en una matriz de cadenas y pasarlo a ListViewAdapter.



```
private class DownloadXML extends AsyncTask<Void, Void, Void> {

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        // Crear un dialogo
        mProgressDialog = new ProgressDialog(MainActivity.this);
        // Titulo del dialogo
        mProgressDialog.setTitle("Lista de Incidencias");
        // Mensaje del dialogo
        mProgressDialog.setMessage("Cargando...");
        mProgressDialog.setIndeterminate(false);
        // Vista del dialogo
        mProgressDialog.show();
    }
}
```

Figura 13. Diálogo de progreso

En esta actividad, he alojado un archivo XML en el servidor y he utilizado el analizador XML para recuperar nodos de la dirección URL.

El enlace de archivo XML del gobierno es:

<http://www.gobiernoabierto.cdmx.gob.mx/sigdata/resources/datasets/IncidenciasCDMXOctubre2014-incidenciaoct2014.xml>

La lista de incidencias está descrita por un xml donde los nodos tienen la siguiente estructura:

```
<dataset>
  <registro>
    <Numeroticket>112014-292801</Numeroticket>
    <Motivo>Luminarias descompuestas</Motivo>
    <Latitud>19.490657</Latitud>
    <Longitud>-99.1732</Longitud>
    <FechaHoraapertura>31/10/2014 22:55</FechaHoraapertura>
  </registro>
  <registro>
  ...
  </registro>
  ...
</dataset>
```

Como se ve cada nodo tiene las propiedades tales como: Motivo, Latitud, Longitud y Fecha y Hora de apertura.

A continuación, sea crea una clase de arreglos en la clase `VariablesDeIncidencias.java`.

Posteriormente, una clase `XMLParser.java` descarga el archivo XML y comprueba si se construye en consecuencia.

A continuación, utilizo un adaptador listview personalizado (`ListViewAdapter.java`) donde los arreglos de cadena se pasan al ListViewAdapter se establecen en TextViews y ImageViews seguidos por las posiciones.

Creando un diseño gráfico XML para su elemento listview denominado `listview_item.xml` como se muestra en la Figura 14.

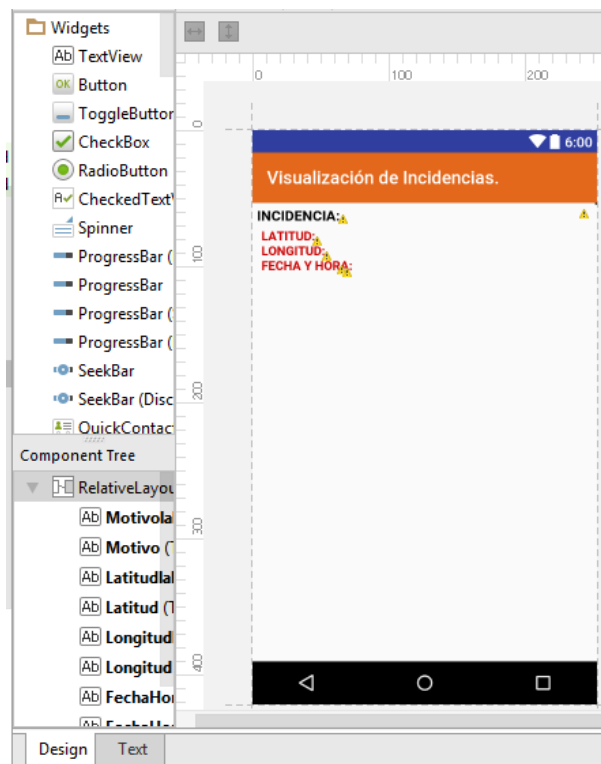


Figura 14. Diseño de listview_item.xml

6.1.5. Actividad: Mapa.java

Para agregar la dependencia en el proyecto primero deberá instalar el complemento disponible en el SDK.

Dentro de Android Studio dirigo a Tools -> Android -> SDK Manager como se muestra en la Figura 15.

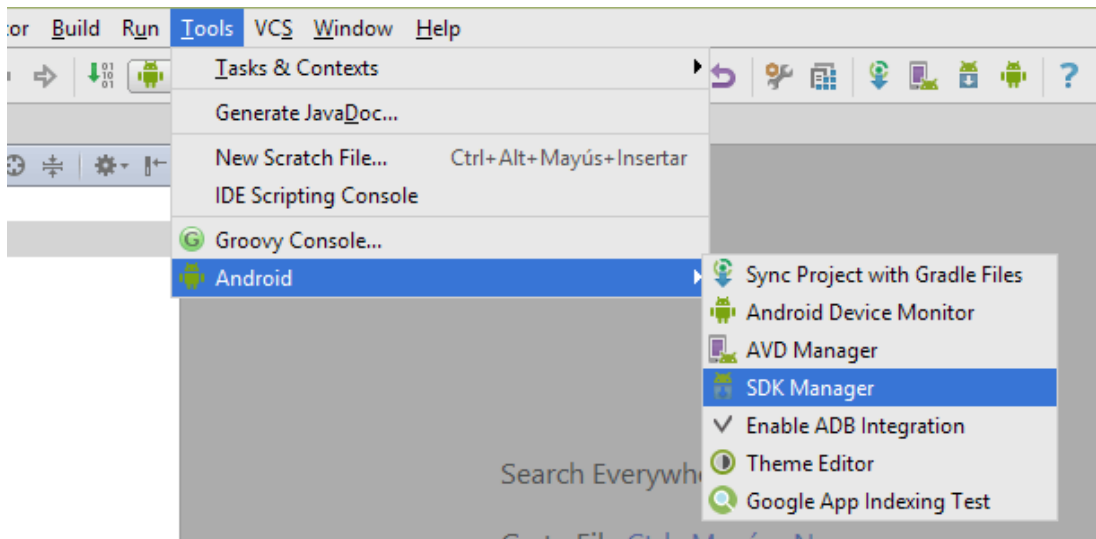


Figura 15. SDK Manager

En la Figura 16. selecciono la pestaña SDL Tools, localizo el paquete Google Play Services y marco su respectivo checkbox, el siguiente paso es aceptar la licencia. Y esperar a que termine la descarga y pegado de los archivos.

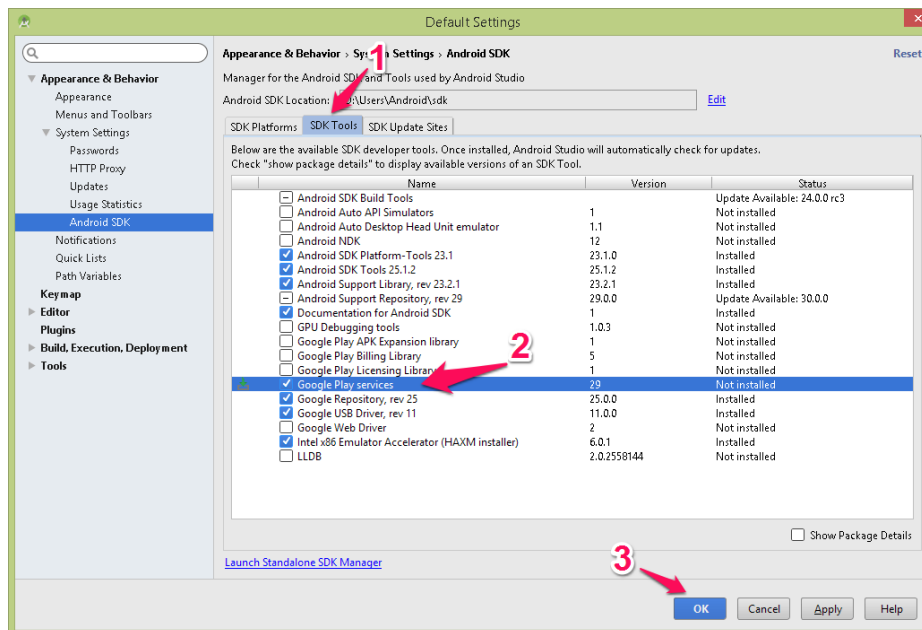


Figura 16. Google Play Services

Añado dependencia a build.gradle para poder ver el mapa; al confirmarla se actualizará y la aplicación será sincronizada, por lo tanto usaré el siguiente comando *compile* como se muestra en la Figura 17.:



Figura 17. Dependencia de google-play-services-places

Deberá habilitar el servicio web en los servidores de Google para proveer datos desde la API de Google Maps, crearé la credencial para la aplicación Android, se pedirá una clave que determina la existencia real de esta.

Por lo tanto, hay que obtener lo siguiente:

Localizar el archivo con la keystore usada para una aplicación en modo debug. La documentación específica para dicho archivo denominada debug.keystore, usar la utilidad keytool para extraer la huella digital SHA-1.

En la Figura 18 para crear nuevo proyecto en Google Console Developers; para usar una API de Google es necesario crear una cuenta Google.

Veamos cómo hacerlo:

- Ingresar a Google Console Developers
- Seleccionar el menú desplegable en la parte derecha de la toolobar que dice Selecciona un proyecto y asignar Crear proyecto.
- Aparecerá el diálogo de creación asigna el nombre Visualización de Incidencias confirmar.
- En la visión general del servicio presiona Habilitar.
- Añadiendo el nombre del paquete y huella digital SHA-1.
- Finalmente copiar la clave de la API y pegarlo al AndroidManifest.xml

O antes de explicar cómo funciona esta actividad para usar Google Maps se tienen que hacer un conjunto de detalles, en la página de internet es:

<https://developers.google.com/maps/documentation/android-api/start?hl=es-419>

se

describe específicamente los pasos a seguir para el uso de la biblioteca.

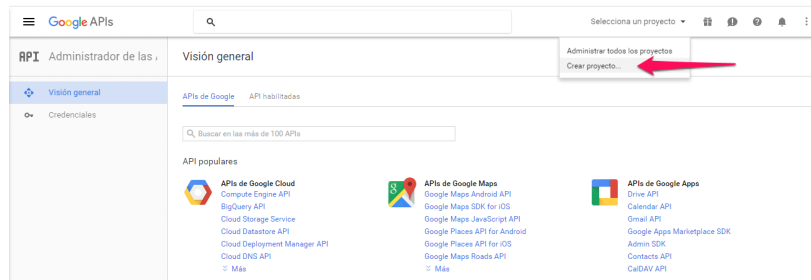


Figura 18. Google Console Developers

El xml denominado a esta actividad es `mapa_coordenadas.xml` que contiene el componente pertinente para el uso de mapas en esa clase.

Entonces hay que estar configurada la biblioteca, con esta API todas las acciones se hacen en función del objeto `GoogleMap` así que lo primero es que necesitamos es construir este objeto. Para desplegar datos de Google Maps API usaremos clase `MapFragment` en versiones mayores o igual a 12 y lo crearemos de la siguiente forma:

```
mMapFragment = (SupportMapFragment) getSupportFragmentManager()
    .findFragmentById(R.id.map);
```

Este fragmento administra totalmente la creación, actualización y destrucción del mapa en la vista.

Lo segundo es implementar `OnMapReadyCallback` en esa actividad, sobrescribir `onMapReady()` y lo último llamar a `getMapAsync()`. Por otro lado, el método `OnMapReadyCallback` escucha e implementa el modelo asíncrono de los servicios de Google, para avisarte en qué momento el mapa está listo y te entrega una referencia del mismo que pueda manipular.

A continuación, para elaborar el método es establecer un objeto `Polyline`, éste podremos trazar un rectángulo sobre el mapa de la Ciudad de México, crearemos algunas propiedades del rectángulo tales como: el color y el ancho del rectángulo como se muestra en la Figura 19.

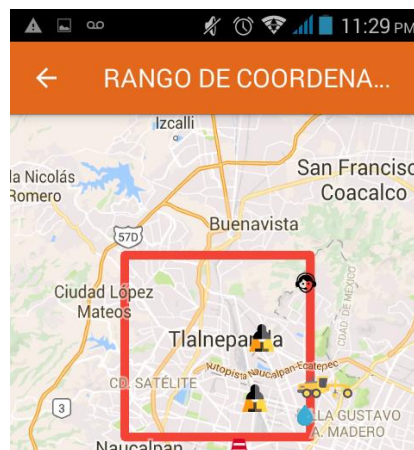


Figura 19. Polyline: Rectángulo

Después añadiremos unos marcadores (marker) mediante el procedimiento `addMarker`, por lo cual le pasamos un objeto de tipo `MarkerOptions`, por medio de éste crearemos la posición (donde irá el marcador), el título del tipo de incidencia y el icono de la imagen de cada uno. Por lo tanto obtendremos la posición mediante las propiedades latitud y longitud de cada elemento y también el nombre como se muestra en la Figura 20.

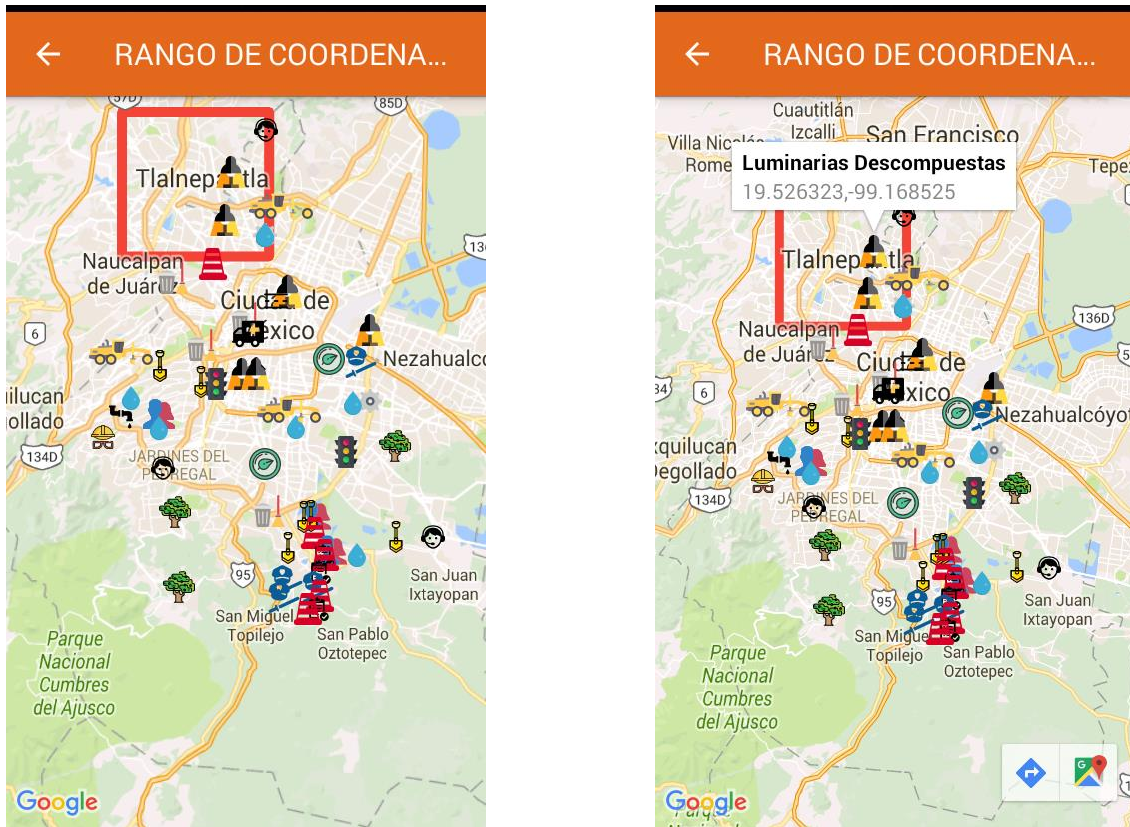


Figura 20. Marcadores de tipos de incidencias

Finalmente centraremos el mapa de la Ciudad de México aplicando un zoom adecuado con la función de la clase `CameraPosition` y el método `animateCamera` de la siguiente forma:

```
import com.google.android.gms.maps.CameraUpdateFactory;
...
...
mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(new
    LatLng(19.325250, -99.136070),10));
```

6.1.6. Actividad: RangoDeFecha.java

Lo primero es crear un diseño gráfico XML para su elemento listview denominado `listview_item_rango_fecha.xml` como se muestra en la Figura 21.

Permitiré un adaptador listview personalizado (`ListAdapter.java`) los arreglos de cadena se pasan al `ListAdapter` se establecen en `TextViews`.

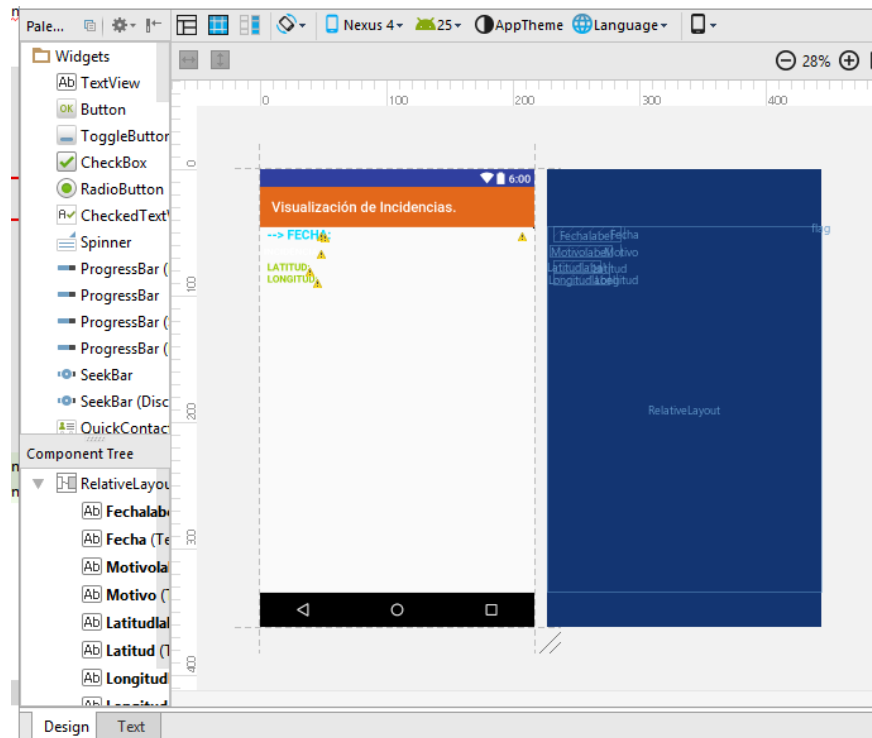


Figura 21. Diseño de `listview_item_rango_fecha.xml`

Un listview puede ser filtrado por la entrada del usuario y está habilitado usando el método `addTextChangedListener`. La función de búsqueda filtrará la vista de lista con una cadena correspondiente a la entrada del usuario.

Añadiremos un listview y en la entrada de Texto se filtrará los resultados de las fechas de apertura en ListView. Averiguaré la fecha de apertura de cada tipo de incidencias de una URL utilizando un analizador XML (`RangoDeFecha.java`).

En esta actividad, he introducido un archivo XML en el servidor y he utilizado el analizador XML para recuperar nodos de la dirección URL.

El enlace de archivo XML del gobierno es:

<http://www.gobiernoabierto.cdmx.gob.mx/sigdata/resources/datasets/IncidenciasCDMXOctubreDel2014-incidenciaoct2014.xml>

La fecha de apertura de incidencias está descrita por un xml donde el nodo denominado `<FechaHoraapertura>` tiene la siguiente estructura:

```

<dataset>
  <registro>
    <Numeroticket>112014-292801</Numeroticket>
    <Motivo>Luminarias descompuestas</Motivo>
    <Latitud>19.490657</Latitud>
    <Longitud>-99.1732</Longitud>
    <FechaHoraapertura>31/10/2014 22:55</FechaHoraapertura>
  </registro>
  <registro>
  ...
  </registro>
  ...
</dataset>

```

A continuación, creo una clase de arreglos en la clase `VariablesDeIncidencias.java`.

Creo una clase `XMLParser.java` ésta descarga el archivo XML y comprueba si se construye en consecuencia.

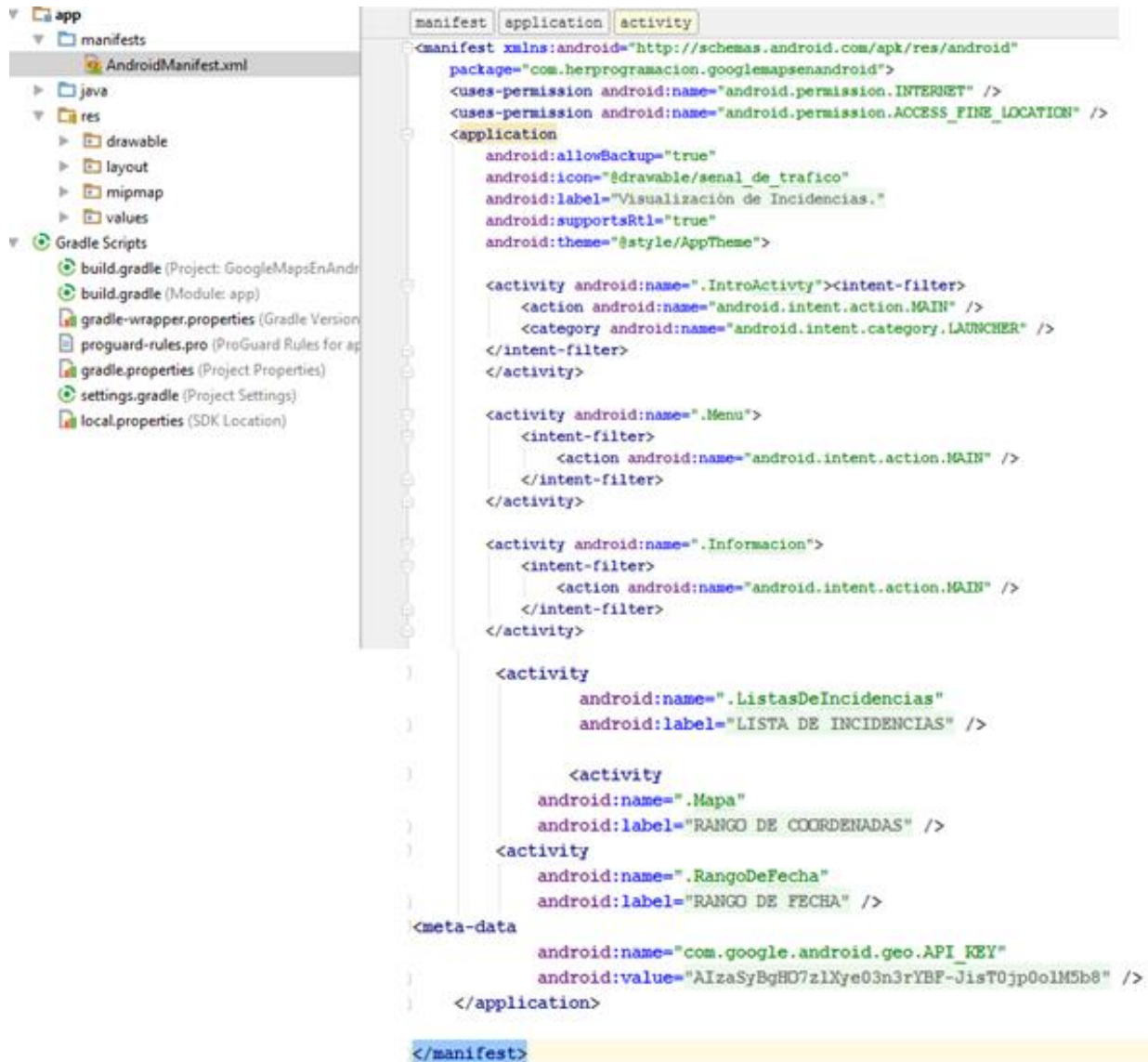
Al final, describiré el xml correspondiente (`listview_main_rango_fecha.xml`) este contiene los siguientes controles: EditText, ListView y ScrollView. Desarrollaré una utilidad de búsqueda en el analizador de XML que filtra textos (Ver Figura 22).



Figura 22. Rango de fecha.

6.2 Modificaciones en el AndroidManifest.xml de todas las actividades.

Situado en la raíz de esas aplicaciones que he hecho AndroidManifest.xml, es un archivo de configuración donde podemos aplicar las configuraciones básicas de la aplicación. Su configuración puede realizarse a través de una interfaz gráfica, por lo tanto tenemos que declarar permisos para permitir la aplicación para escribir un almacenamiento externo, conectarse a Internet y a los servicios de Google como se muestra en la Figura 23.



```
manifest application activity
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.herprogramacion.googlemapsenandroid"
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <application
        android:allowBackup="true"
        android:icon="@drawable/senal_de_trafico"
        android:label="Visualización de Incidencias."
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".IntroActivty"><intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
        </activity>
        <activity android:name=".Menu">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
            </intent-filter>
        </activity>
        <activity android:name=".Informacion">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
            </intent-filter>
        </activity>
        <activity
            android:name=".ListasDeIncidencias"
            android:label="LISTA DE INCIDENCIAS" />
        <activity
            android:name=".Mapa"
            android:label="RANGO DE COORDENADAS" />
        <activity
            android:name=".RangoDeFecha"
            android:label="RANGO DE FECHA" />
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="AIzaSyBqHD7zIXye03n3rYBF-JisT0jp0o1M5b8" />
    </application>
</manifest>
```

Figura 23. Permisos en el archivo AndroidManifest.xml

En esa figura contiene las siguientes actividades tales como:

Icono de la aplicación (ver la Figura 24.) que aparecerá posteriormente al instalarse en los menús y nombre.

```

<application
    android:allowBackup="true"
    android:icon="@drawable/senal_de_trafico"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">

```

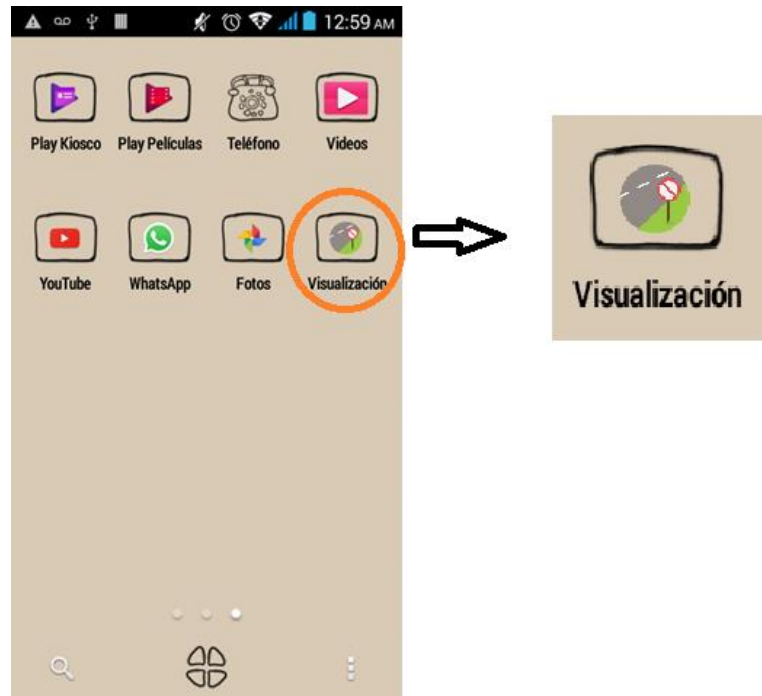


Figura 24. Icono de la aplicación

La actividad inicial de .IntroActivity debe ser declarada de tipo “LAUNCHER”.

Las otras actividades de .Menu, .Información, .ListasDelIncidencias y .RangoDeFecha deben añadir el tipo “MAIN”, sino no funcionarán.

La actividad de .Mapa deberá agregar un hijo <meta-data> al nodo <application> con la siguiente descripción:

```

<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:name="CLAVE_DE_API"/>

```

7. RESULTADOS

- Se logró implementar la lista de incidencias de un tamaño variable de entre 1 a 1000 incidencias dependiendo el año.
- Se logró implementar la consulta por rango de fecha y aunque quedo acotada por la información alojada en los servidores del gobierno de la CDMX esta aplicación esta diseñada para la consulta de cualquier fecha siempre y cuando exista en estos.
- Las incidencias ocurren prácticamente en cualquier rincón de la CDMX pero se concentran en mayor medida dentro de la delegación Iztapalapa.

8. CONCLUSIONES.

Unos de los objetivos que más rápido pude alcanzar fue el módulo de lectura de datos abiertos de las incidencias, aunque en un principio carecía de los elementos técnicos y teóricos para su realización una breve investigación me condujo a las herramientas de programación necesarias para su implementación, las cuales fueron las funciones XML Parser cuya función es enlazar mi aplicación con los servidores necesarios donde obtenemos la información de las incidencias en forma de listas.

La implementación del módulo de consulta fue un objetivo que necesito de más tiempo y una investigación mucho mas intensiva debido a que desconocía el uso e importancia de los plugin necesarios, plugins como “dependences” eran necesarios para la compilación y los filtros de la información consultados por el usuario.

Diseñar el módulo de la visualización de respuesta, fue un proceso lleno de errores ya que de nueva cuenta los plugin resaltaron su importancia aunado a que desconocía el hecho de hacer una cuenta especial Google Maps API para poder descargar todo lo necesario para la utilización de Google Maps, una vez resuelto esto un nuevo problema salto, la huella digital SHA-1 proporcionada tendría que ser incluida en el código de la App hecho que también desconocía.

En una breve opinion personal quisiera comentar que llegue a este proyecto terminal con casi ningún tipo de información o practica sobre dispositivos Android, este hecho en momentos fue frustrante por el tiempo que tuvo que dedicar a empaparme de la teoría necesaria para comenzar a realizarlo, o para sobreponerme a todas las dificultades que dentro de el me iba encontrando a cuanta vez avanzaba. Para terminar esta opinión creo que necesite que dentro de mi formación académica dentro de la Uam existieran UEAs que proporcionen los elementos al menos básicos para afrontar las dificultades que un proyecto de este tipo tiene.

Dentro de mi proyecto me hubiera gustado que las incidencias incluyeran una descripción especifica tanto de su estatus como de sus causas así como imágenes de las mismas, lamentablemente esto tiene que ver más con los datos que proporciona el mismo gobierno de la CDMX en sus servidores y de los cuales mi aplicación descarga la información, y aunque yo he habilitado dentro del código la gestión de imágenes por si en un futuro las implementas por el momento no las hay.

La experiencia que me dio este proyecto me a dejado un agradable deseo de adentrarme al mundo de las aplicaciones móviles, encuentro en este campo una gran oportunidad para materializar mi creatividad y algunas ideas que he tenido rondando mi mente hace años.

9. GLOSARIO.

Activity: Está compuesto por otros dos nodos o atributos en su interior: name label.

APK: Es la extensión de las aplicaciones Android, variante del Java.

Broadcast Receiver: Es un mensaje que cualquier aplicación puede recibir.

Drawable: Un tipo de recurso visual que podemos utilizar al programar en Android.

EditText: Es un componente que permite la edición y escritura de texto.

Intent: Intercambiar datos entre actividades y servicios, por lo que podemos también utilizarlas para hacer llamados a otras actividades.

Interfaz: es la capa que existe entre el usuario y el dispositivo, que le permite interactuar con este último.

Kernel: Es el núcleo de los sistemas operativos basados en Linux.

KitKat: Es la denominación de la versión 4.4 del sistema operativo Android.

Launcher: Parte de la interfaz de usuario de Android en pantallas de inicio que le permite lanzar aplicaciones.

Layout: Fichero XML que describe la composición de una pantalla.

ListView: Es un componente que permite desplegar una serie de elementos.

Google Maps: Emplea la información vía satélite.

Método: Conjunto de comportamientos o acciones que puede realizarse en el objeto.

ProgressDialog: Es un elemento en Android para hacer entender al usuario que debe esperar mientras la aplicación hace una tarea en segundo plano, ya sea un cálculo, una autenticación, una descarga, etc.

SDK (en español, "kit de desarrollo de software"): Conjunto de herramientas que nos sirve para crear aplicaciones de manera simple para un sistema o plataforma en particular.

Secure hash algorithm (SHA1): Es un algoritmo seguro con el que firmamos nuestra aplicación que se utiliza para poder hacer uso de varias API que Google proporciona para los desarrolladores, como por ejemplo Google Maps.

XML: Formato flexible que permite la gestión y el intercambio de datos estructurados mediante simples archivos de texto.

Widget: Elemento que puede añadirse al escritorio de nuestro dispositivo Android, ofreciéndonos información y determinadas funcionalidades.

10. APÉNDICES

10.1 Clases de actividades.

i. IntroActivity.java

```
package com.herprogramacion.googlemapsenandroid;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.support.v4.view.PagerAdapter;
import android.support.v4.view.ViewPager;
import android.view.*;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;

public class IntroActivty extends Activity {

    private ViewPager viewPager;
    private ViewPagerAdapter viewPagerAdapter;
    private LinearLayout dotsLayout;
    private TextView[] dots;
    private int[] layouts;
    private Button btnSkip, btnNext;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_intro);

        viewPager = (ViewPager) findViewById(R.id.view_pager);
        dotsLayout = (LinearLayout) findViewById(R.id.layoutDots);
        btnSkip = (Button) findViewById(R.id.btn_skip);
        btnNext = (Button) findViewById(R.id.btn_next);

        layouts = new int[]{
            R.layout.slide1,
            R.layout.slide2,
            R.layout.slide3};

        addBottomDots(0);

        viewPagerAdapter = new ViewPagerAdapter();
        viewPager.setAdapter(viewPagerAdapter);
        viewPager.addOnPageChangeListener(viewPagerPageChangeListener);
```

```

}

public void btnSkipClick(View v)
{
    launchHomeScreen();
}

public void btnNextClick(View v)
{
    int current = getItem(1);
    if (current < layouts.length) {
        // move to next screen
        viewPager.setCurrentItem(current);
    } else {
        launchHomeScreen();
    }
}

ViewPager.OnPageChangeListener viewPagerPageChangeListener = new
ViewPager.OnPageChangeListener() {

    @Override
    public void onPageSelected(int position) {
        addBottomDots(position);

        /*Cambiando el siguiente texto del botón 'SIGUIENTE' */
        if (position == layouts.length - 1) {
            // last page. make button text to GOT IT
            btnNext.setText(getString(R.string.start));
            btnSkip.setVisibility(View.GONE);

        } else {

            btnNext.setText(getString(R.string.next));
            btnSkip.setVisibility(View.VISIBLE);
        }
    }

    @Override
    public void onPageScrolled(int arg0, float arg1, int arg2) {

    }

    @Override
    public void onPageScrollStateChanged(int arg0) {

    }
};

private void addBottomDots(int currentPage) {
    dots = new TextView[layouts.length];

}

private int getItem(int i) {
    return viewPager.getCurrentItem() + i;
}

```

```

private void launchHomeScreen() {
    startActivity(new Intent(this, Menu.class));
    finish();
}

public class ViewPagerAdapter extends PagerAdapter {
    private LayoutInflater inflater;

    public ViewPagerAdapter() {

    }

    @Override
    public Object instantiateItem(ViewGroup container, int position) {
        inflater = (LayoutInflater)
getSystemService(Context.LAYOUT_INFLATER_SERVICE);

        View view = inflater.inflate(layouts[position], container,
false);
        container.addView(view);

        return view;
    }

    @Override
    public int getCount() {
        return layouts.length;
    }

    @Override
    public boolean isViewFromObject(View view, Object obj) {
        return view == obj;
    }

    @Override
    public void destroyItem(ViewGroup container, int position, Object
object) {
        View view = (View) object;
        container.removeView(view);
    }
}
}

```

ii. Informacion.java

```
package com.herprogramacion.googlemapsenandroid;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;

public class Informacion extends AppCompatActivity{

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.portada_autora);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }

    @Override
    public boolean onSupportNavigateUp() {
        onBackPressed();
        return super.onSupportNavigateUp();
    }
}
```

iii. Menu.java

```
package com.herprogramacion.googlemapsenandroid;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.RecyclerView;
import android.view.View;
import android.widget.ImageView;

public class Menu extends AppCompatActivity implements
SamplesAdapter.OnItemClickListener {

    ImageView info;
    ImageView salir;
    public final static Class[] options = {

        ListasDeIncidencias.class,
        Mapa.class,
        RangoDeFecha.class

    };

    private SamplesAdapter mAdapter;
    private RecyclerView mRecyclerView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.menu);
        // botón de información
        info = (ImageView) findViewById(R.id.info);
    }
}
```

```

        info.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent info = new Intent(Menu.this, Informacion.class);
                startActivity(info);
            }
        });
// botón de salir
salir=(ImageView) findViewById(R.id.exit);
salir.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        finish();
        Intent intent = new Intent(Intent.ACTION_MAIN);
        intent.addCategory(Intent.CATEGORY_HOME);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
    }
});

mAdapter = new SamplesAdapter(this, this);
mRecyclerView = (RecyclerView) findViewById(R.id.lista);
mRecyclerView.setAdapter(mAdapter);

@Override
public void onClick(SamplesAdapter.ViewHolder holder, int opcion) {
    mostrarOpcion(opcion);
}

private void mostrarOpcion(int indice) {
    Intent i = new Intent(this, options[indice]);
    startActivity(i);
}
}

```

iv. SamplesAdapter.java

```

package com.herprogramacion.googlemapsenandroid;

import android.content.Context;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

public class SamplesAdapter extends
RecyclerView.Adapter<SamplesAdapter.ViewHolder> {
    private final Context mContext;

    private OnItemClickListener callback;

    interface OnItemClickListener {
        public void onClick(ViewHolder holder, int opcion);
    }
}

```

```

public class ViewHolder extends RecyclerView.ViewHolder
    implements View.OnClickListener {

    public TextView opcion;

    public ViewHolder(View v) {
        super(v);
        opcion = (TextView) v.findViewById(android.R.id.text1);
        v.setOnClickListener(this);
    }

    @Override
    public void onClick(View view) {
        callback.onClick(this, getAdapterPosition());
    }
}

public SamplesAdapter(Context contexto, OnItemClickListener escucha) {
    this.mContext = contexto;
    this.callback = escucha;
}

@Override
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View v = LayoutInflater.from(mContext)
        .inflate(android.R.layout.simple_selectable_list_item, parent,
false);
    return new ViewHolder(v);
}

@Override
public void onBindViewHolder(ViewHolder holder, int position) {

    // Asignación UI
    holder.opcion.setText(ITEMS[position]);
}

@Override
public int getItemCount() {
    return ITEMS.length;
}

public static String[] ITEMS = {
    "Lista de Incidencias",
    "Rango de Coordenadas",
    "Rango de Fechas"
};
}

```

V. ListasDeIncidencias.java

```

package com.herprogramacion.googlemapsenandroid;

import android.app.ProgressDialog;
import android.os.AsyncTask;

```

```

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
import android.widget.EditText;
import android.widget.ListView;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.NodeList;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;
public class ListasDeIncidencias extends AppCompatActivity {

    ListView listview;
    ListViewAdapter adapter;
    ProgressDialog mProgressDialog;
    EditText editsearch;

    static String MOTIVO = "Motivo";
    static String LATITUD = "Latitud";
    static String LONGITUD = "Longitud";
    static String FECHAHORAAPERTURA = "FechaHoraapertura";

    private List<VariablesDeIncidencias> datasetlist = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // Get the view from listview main.xml
        setContentView(R.layout.listview_incidencias);
        // Execute DownloadXML AsyncTask
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        new DownloadXML().execute();
    }

    private class DownloadXML extends AsyncTask<Void, Void, Void> {

        @Override
        protected void onPreExecute() {
            super.onPreExecute();

            mProgressDialog = new ProgressDialog(ListasDeIncidencias.this);
            mProgressDialog.setTitle("Lista de Incidencias");
            mProgressDialog.setMessage("Cargando...");
            mProgressDialog.setIndeterminate(false);

            mProgressDialog.show();
        }

        @Override
        protected Void doInBackground(Void... params) {
            // Create the array
            datasetlist = new ArrayList<VariablesDeIncidencias>();
            XMLParser parser = new XMLParser();

```



```

// Retrive nodes from the given website URL in XMLParser.class
String xml = parser

.getXmlFromUrl("http://www.gobiernoabierto.cdmx.gob.mx/sigdata/resources/datasets/IncidenciasCDMXOctubredel2014-incidenciaoct2014.xml");
// Retrive DOM element
Document doc = parser.getDomElement(xml);

try {
// Locate the NodeList name
NodeList nl = doc.getElementsByTagName("registro");
for (int i = 0; i < nl.getLength(); i++) {
Element e = (Element) nl.item(i);
VariablesDeIncidencias map = new VariablesDeIncidencias();

map.setmotivo(parser.getValue(e, MOTIVO));
map.setlatitud(parser.getValue(e, LATITUD));
map.setlongitud(parser.getValue(e, LONGITUD));
map.setfechahoraapertura(parser.getValue(e,
FECHAHORAAPERTURA));

datasetlist.add(map);
}
} catch (Exception e) {
Log.e("Error", e.getMessage());
e.printStackTrace();
}
return null;
}

@Override
protected void onPostExecute(Void args) {

listview = (ListView) findViewById(R.id.listview);

adapter = new ListViewAdapter(ListasDeIncidencias.this,
datasetlist);

listview.setAdapter(adapter);
mProgressDialog.dismiss();
editsearch = (EditText) findViewById(R.id.search);
editsearch.addTextChangedListener(new TextWatcher() {

@Override
public void afterTextChanged(Editable arg0) {
// TODO Auto-generated method stub
String text = editsearch.getText().toString()
.toLowerCase(Locale.getDefault());
adapter.filter(text);
}

@Override
public void beforeTextChanged(CharSequence arg0, int arg1,
int arg2, int arg3) {
// TODO Auto-generated method stub
}

@Override

```

```

        public void onTextChanged(CharSequence arg0, int arg1,
                                int arg2, int arg3) {
            // TODO Auto-generated method stub
        }

    });
}

}
@Override
public boolean onSupportNavigateUp() {
    onBackPressed();
    return super.onSupportNavigateUp();
}
}
}

```

vi. ListViewAdapter.java

```

package com.herprogramacion.googlemapsenandroid;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;

public class ListViewAdapter extends BaseAdapter {

    private final List<VariablesDeIncidencias> datasetlist;

    Context context;
    LayoutInflater inflater;
    ArrayList<HashMap<String, String>> data;
    ImageLoader imageLoader;
    private List<VariablesDeIncidencias> dataset = null;
    private ArrayList<VariablesDeIncidencias> arraylist;

    public ListViewAdapter(Context context,
                           List<VariablesDeIncidencias> datasetlist) {
        this.context = context;
        this.datasetlist = datasetlist;
        inflater = LayoutInflater.from(context);
        this.arraylist = new ArrayList<VariablesDeIncidencias>();
        this.arraylist.addAll(datasetlist);
        imageLoader = new ImageLoader(context);
    }

    public class ViewHolder {

```

```

    TextView Motivo;
    TextView Latitud;
    TextView Longitud;
    TextView FechaHoraapertura;

}

@Override
public int getCount() {
    return datasetlist.size();
}

@Override
public Object getItem(int position) {
    return datasetlist.get(position);
}

@Override
public long getItemId(int position) {
    return position;
}

public View getView(final int position, View view, ViewGroup parent) {
    final ViewHolder holder;
    if (view == null) {
        holder = new ViewHolder();
        view = inflater.inflate(R.layout.listview_item, null);

        holder.Motivo = (TextView) view.findViewById(R.id.Motivo);
        holder.Latitud = (TextView) view.findViewById(R.id.Latitud);
        holder.Longitud = (TextView) view.findViewById(R.id.Longitud);
        holder.FechaHoraapertura = (TextView)
view.findViewById(R.id.FechaHoraapertura);

        view.setTag(holder);
    } else {
        holder = (ViewHolder) view.getTag();
    }
    // Set the results into TextViews

    holder.Motivo.setText(datasetlist.get(position).getmotivo());
    holder.Latitud.setText(datasetlist.get(position).getlatitud());
    holder.Longitud.setText(datasetlist.get(position).getlongitud());//getRank

    holder.FechaHoraapertura.setText(datasetlist.get(position).getfechahoraapertura(
));

    view.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View arg0) {
            }
    });
    return view;
}
// Filter Class
public void filter(String charText) { // para filtrar datos
    charText = charText.toLowerCase(Locale.getDefault());

```

```

        datasetlist.clear();
        if (charText.length() == 0) {
            datasetlist.addAll(arraylist);
        } else {
            for (VariablesDeIncidencias wp : arraylist) {
                if
(wp.getmotivo().toLowerCase(Locale.getDefault()).contains(charText)) {
                    datasetlist.add(wp);
                }
            }
        }
        notifyDataSetChanged();
    }
}

```

vii. VariablesDeIncidencias.java

```

package com.herprogramacion.googlemapsenandroid;

```

```

public class VariablesDeIncidencias {
    private String Motivo;
    private String Latitud;
    private String Longitud;
    private String FechaHoraapertura;

    public String getmotivo() {
        return Motivo;
    }

    public void setmotivo(String Motivo) {
        this.Motivo = Motivo;
    }

    //-----

    public String getlatitud() {
        return Latitud;
    }

    public void setlatitud(String Latitud) {
        this.Latitud = Latitud;
    }

    //-----

    public String getlongitud() {
        return Longitud;
    }

    public void setlongitud(String Longitud) {
        this.Longitud = Longitud;
    }

    //-----
}

```

```

    public String getfechahoraapertura() {
        return FechaHoraapertura;
    }

    public void setfechahoraapertura(String FechaHoraapertura) {
        this.FechaHoraapertura = FechaHoraapertura;
    }
}

```

viii. XMLParser.java

```

package com.herprogramacion.googlemapsenandroid;

import android.util.Log;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.util.EntityUtils;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.InputSource;
import org.xml.sax.SAXException;

import java.io.IOException;
import java.io.StringReader;
import java.io.UnsupportedEncodingException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

public class XMLParser {

    public XMLParser() {

    }

    // Retrive XML from URL
    public String getXmlFromUrl(String url) {
        String xml = null;

        try {
            DefaultHttpClient httpClient = new DefaultHttpClient();
            HttpPost httpPost = new HttpPost(url);

            HttpResponse httpResponse = httpClient.execute(httpPost);
            HttpEntity httpEntity = httpResponse.getEntity();
            xml = EntityUtils.toString(httpEntity);

        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        } catch (ClientProtocolException e) {

```

```

        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

    return xml;
}

public Document getDomElement(String xml) {
    Document doc = null;
    DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
    try {

        DocumentBuilder db = dbf.newDocumentBuilder();

        InputSource is = new InputSource();
        is.setCharacterStream(new StringReader(xml));
        doc = db.parse(is);

    } catch (ParserConfigurationException e) {
        Log.e("Error: ", e.getMessage());
        return null;
    } catch (SAXException e) {
        Log.e("Error: ", e.getMessage());
        return null;
    } catch (IOException e) {
        Log.e("Error: ", e.getMessage());
        return null;
    }

    return doc;
}

public final String getElementValue(Node elem) {
    Node child;
    if (elem != null) {
        if (elem.hasChildNodes()) {
            for (child = elem.getFirstChild(); child != null; child = child
                .getNextSibling()) {
                if (child.getNodeType() == Node.TEXT_NODE) {
                    return child.getNodeValue();
                }
            }
        }
    }
    return "";
}

// Retrive Node Value
public String getValue(Element item, String str) {
    NodeList n = item.getElementsByTagName(str);
    return this.getElementValue(n.item(0));
}
}

```

ix. Mapa.java

```
package com.herprogramacion.googlemapsenandroid;

import android.graphics.Color;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.maps.model.Polyline;
import com.google.android.gms.maps.model.PolylineOptions;
public class Mapa extends AppCompatActivity
    implements OnMapReadyCallback {

    private SupportMapFragment mMapFragment;
    private GoogleMap mMap;
    private Marker markerCdmx;
    private Marker markerCdmx1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.mapa_coordenadas);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        mMapFragment = (SupportMapFragment) getSupportFragmentManager()
            .findFragmentById(R.id.map);
        mMapFragment.getMapAsync(this);
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        LatLng cdmx = new LatLng(19.490657,-99.1732);
        markerCdmx = googleMap.addMarker(new MarkerOptions()
            .position(cdmx)
            .title("Luminarias descompuestas")
            .snippet(cdmx.latitude + "," + cdmx.longitude)
            .icon(BitmapDescriptorFactory.fromResource(R.drawable.lampara)
        );

        LatLng cdmx1 = new LatLng(19.458073,-99.183016);
        markerCdmx1 = googleMap.addMarker(new MarkerOptions()
            .position(cdmx1)
            .title("Accesorios de Infraestructura")
            .snippet(cdmx1.latitude + "," + cdmx1.longitude)
            .icon(BitmapDescriptorFactory.fromResource(R.drawable.pilon)
        );
    }
}
```

```

LatLng cdmx2 = new LatLng(19.38462471,-99.0650643);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx2)
    .title("Vigilancia Policiaca")
    .snippet(cdmx2.latitude + "," + cdmx2.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.vigpol))
);

LatLng cdmx3 = new LatLng(19.368475,-99.180777);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx3)
    .title("Semáforos descompuestos")
    .snippet(cdmx3.latitude + "," + cdmx3.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.sem))
);

LatLng cdmx4 = new LatLng(19.408167,-99.058405);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx4)
    .title("Luminarias descompuestas")
    .snippet(cdmx4.latitude + "," + cdmx4.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.lampara))
);

LatLng cdmx5 = new LatLng(19.489069,-99.128942);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx5)
    .title("Bacheo")
    .snippet(cdmx5.latitude + "," + cdmx5.longitude)

.icon(BitmapDescriptorFactory.fromResource(R.drawable.asfaltadora)
);

LatLng cdmx6 = new LatLng(19.321788,-99.038753);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx6)
    .title("Poda de Árboles")
    .snippet(cdmx6.latitude + "," + cdmx6.longitude)

.icon(BitmapDescriptorFactory.fromResource(R.drawable.arbol_frutal)
);

LatLng cdmx7 = new LatLng(19.34658, -99.25537);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx7)
    .title("Coladeras (mantenimiento)")
    .snippet(cdmx7.latitude + "," + cdmx7.longitude)

.icon(BitmapDescriptorFactory.fromResource(R.drawable.tuberia_perdiendo_agua)
);

LatLng cdmx8 = new LatLng(19.38694, -99.090753);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx8)
    .title("Mantenimiento a Áreas Verdes")
    .snippet(cdmx8.latitude + "," + cdmx8.longitude)

.icon(BitmapDescriptorFactory.fromResource(R.drawable.certificacion)
);

```



```

LatLng cdmx9 = new LatLng(19.356583, -99.072156);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx9)
    .title("Fuga de Agua")
    .snippet(cdmx9.latitude + "," + cdmx9.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.soltar)
);

LatLng cdmx10 = new LatLng(19.358112, -99.058639);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx10)
    .title("Mantenimiento al alumbrado")
    .snippet(cdmx10.latitude + "," + cdmx10.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.ajustes)
);

LatLng cdmx11 = new LatLng(19.482397,-99.141674);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx11)
    .title("Fuga de Agua")
    .snippet(cdmx11.latitude + "," + cdmx11.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.ajustes)
);

LatLng cdmx12 = new LatLng(19.44723213,-99.2141525);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx12)
    .title("Limpieza de Calles")
    .snippet(cdmx12.latitude + "," + cdmx12.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.barrer)
);

LatLng cdmx13 = new LatLng(19.369873,-99.191924);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx13)
    .title("Desazolve")
    .snippet(cdmx13.latitude + "," + cdmx13.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.pala)
);

LatLng cdmx14 = new LatLng(19.375758,-99.16004);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx14)
    .title("Luminarias descompuestas")
    .snippet(cdmx14.latitude + "," + cdmx14.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.lampara)
);

LatLng cdmx15 = new LatLng(19.376499,-99.148715);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx15)
    .title("Luminarias faltantes")
    .snippet(cdmx15.latitude + "," + cdmx15.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.farola)
);

LatLng cdmx16 = new LatLng(19.375432,-99.14893);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx16)

```

```

        .title("Luminarias descompuestas")
        .snippet(cdmx16.latitude + "," + cdmx16.longitude)
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.lampara)
    );

    LatLng cdmx17 = new LatLng(19.4380314,-99.12369002);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx17)
        .title("Luminarias descompuestas")
        .snippet(cdmx17.latitude + "," + cdmx17.longitude)
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.lampara)
    );

    LatLng cdmx18 = new LatLng(19.338941,-99.117121);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx18)
        .title("Fuga de Agua")
        .snippet(cdmx18.latitude + "," + cdmx18.longitude)
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.soltar)
    );

    LatLng cdmx19 = new LatLng(19.43688814,-99.1238295);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx19)
        .title("Luminarias Descompuestas")
        .snippet(cdmx19.latitude + "," + cdmx19.longitude)
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.lampara)
    );

    LatLng cdmx20 = new LatLng(19.482397,-99.141674);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx20)
        .title("Fuga de Agua")
        .snippet(cdmx20.latitude + "," + cdmx20.longitude)
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.soltar)
    );

    LatLng cdmx21 = new LatLng(19.417331,-99.156869);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx21)
        .title("Limpieza de calles")
        .snippet(cdmx21.latitude + "," + cdmx21.longitude)
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.barrer)
    );

    LatLng cdmx22 = new LatLng(19.482397,-99.141674);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx22)
        .title("Fuga de Agua")
        .snippet(cdmx22.latitude + "," + cdmx22.longitude)
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.soltar)
    );

    LatLng cdmx23 = new LatLng(19.482397,-99.141674);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx23)
        .title("Fuga de Agua")
        .snippet(cdmx23.latitude + "," + cdmx23.longitude)
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.soltar)
    );

    LatLng cdmx24 = new LatLng(19.405312,-99.15532);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx24)

```

```

        .title("Retiro de Ambulantes")
        .snippet(cdmx24.latitude + "," + cdmx24.longitude)

.icon(BitmapDescriptorFactory.fromResource(R.drawable.ambulancia)
);
LatLng cdmx25 = new LatLng(19.256299,-99.008031);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx25)
    .title("Apoyo Logísticos para eventos públicos")
    .snippet(cdmx25.latitude + "," + cdmx25.longitude)

.icon(BitmapDescriptorFactory.fromResource(R.drawable.trabajador_del_centro_de_a
tencion_telefonica_con_el_receptor_de_cabeza)
);
LatLng cdmx26 = new LatLng(19.482397,-99.141674);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx26)
    .title("Fuga de Agua")
    .snippet(cdmx26.latitude + "," + cdmx26.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.soltar)
);
LatLng cdmx27 = new LatLng(19.360496,-99.247601);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx27)
    .title("Fuga de Agua")
    .snippet(cdmx27.latitude + "," + cdmx27.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.soltar)
);
LatLng cdmx28 = new LatLng(19.338408,-99.226004);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx28)
    .title("Fuga de Agua")
    .snippet(cdmx28.latitude + "," + cdmx28.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.soltar)
);
LatLng cdmx29 = new LatLng(19.269409,-99.105294);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx29)
    .title("Desazolve")
    .snippet(cdmx29.latitude + "," + cdmx29.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.pala)
);
LatLng cdmx30 = new LatLng(19.345446,-99.225835);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx30)
    .title("Asistencia Social")
    .snippet(cdmx30.latitude + "," + cdmx30.longitude)

.icon(BitmapDescriptorFactory.fromResource(R.drawable.asistencia_social)
);
LatLng cdmx31 = new LatLng(19.482397,-99.141674);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx31)
    .title("Fuga de Agua")
    .snippet(cdmx31.latitude + "," + cdmx31.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.soltar)
);
LatLng cdmx32 = new LatLng(19.38029781,-99.25581118);

```

```

        markerCdmx1 = googleMap.addMarker(new MarkerOptions()
            .position(cdmx32)
            .title("Bacheo")
            .snippet(cdmx32.latitude + "," + cdmx32.longitude)

        .icon(BitmapDescriptorFactory.fromResource(R.drawable.asfaltadora)
        );
        LatLng cdmx33 = new LatLng(19.245183,-99.122955);
        markerCdmx1 = googleMap.addMarker(new MarkerOptions()
            .position(cdmx33)
            .title("Desazolve")
            .snippet(cdmx33.latitude + "," + cdmx33.longitude)
            .icon(BitmapDescriptorFactory.fromResource(R.drawable.pala)
        );
        LatLng cdmx34 = new LatLng(19.345509,-99.225683);
        markerCdmx1 = googleMap.addMarker(new MarkerOptions()
            .position(cdmx34)
            .title("Asistencia Social")
            .snippet(cdmx34.latitude + "," + cdmx34.longitude)

        .icon(BitmapDescriptorFactory.fromResource(R.drawable.asistencia_social)
        );
        LatLng cdmx35 = new LatLng(19.345509,-99.225683);
        markerCdmx1 = googleMap.addMarker(new MarkerOptions()
            .position(cdmx35)
            .title("Asistencia Social")
            .snippet(cdmx35.latitude + "," + cdmx35.longitude)

        .icon(BitmapDescriptorFactory.fromResource(R.drawable.asistencia_social)
        );

        LatLng cdmx36 = new LatLng(19.397567,-99.191633);
        markerCdmx1 = googleMap.addMarker(new MarkerOptions()
            .position(cdmx36)
            .title("Limpieza de calles")
            .snippet(cdmx36.latitude + "," + cdmx36.longitude)
            .icon(BitmapDescriptorFactory.fromResource(R.drawable.barrer)
        );
        LatLng cdmx37 = new LatLng(19.345509,-99.225683);
        markerCdmx1 = googleMap.addMarker(new MarkerOptions()
            .position(cdmx37)
            .title("Asitencia social")
            .snippet(cdmx37.latitude + "," + cdmx37.longitude)

        .icon(BitmapDescriptorFactory.fromResource(R.drawable.asistencia_social)
        );
        LatLng cdmx38 = new LatLng(19.381275,-99.224528);
        markerCdmx1 = googleMap.addMarker(new MarkerOptions()
            .position(cdmx38)
            .title("Desazolve")
            .snippet(cdmx38.latitude + "," + cdmx38.longitude)
            .icon(BitmapDescriptorFactory.fromResource(R.drawable.pala)
        );
        LatLng cdmx39 = new LatLng(19.381275,-99.224528);
        markerCdmx1 = googleMap.addMarker(new MarkerOptions()
            .position(cdmx39)
            .title("Desazolve")
            .snippet(cdmx39.latitude + "," + cdmx39.longitude)

```

```

        .icon(BitmapDescriptorFactory.fromResource(R.drawable.pala)
    );
    LatLng cdmx40 = new LatLng(19.345509,-99.225683);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx40)
        .title("Asistencia social")
        .snippet(cdmx40.latitude + "," + cdmx40.longitude)

    .icon(BitmapDescriptorFactory.fromResource(R.drawable.asistencia_social)
    );
    LatLng cdmx41 = new LatLng(19.273385,-99.138733);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx41)
        .title("Limpieza de calles")
        .snippet(cdmx41.latitude + "," + cdmx41.longitude)
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.barrer)
    );
    LatLng cdmx42 = new LatLng(19.482397,-99.141674);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx42)
        .title("Fuga de Agua")
        .snippet(cdmx42.latitude + "," + cdmx42.longitude)
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.soltar)
    );
    LatLng cdmx43 = new LatLng(19.268594,-99.110323);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx43)
        .title("Desazolve")
        .snippet(cdmx43.latitude + "," + cdmx43.longitude)
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.pala)
    );
    LatLng cdmx44 = new LatLng(19.243314,-99.068773);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx44)
        .title("Fuga de Agua")
        .snippet(cdmx44.latitude + "," + cdmx44.longitude)
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.soltar)
    );
    LatLng cdmx45 = new LatLng(19.2701381,-99.1022051);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx45)
        .title("Asistencia Social")
        .snippet(cdmx45.latitude + "," + cdmx45.longitude)

    .icon(BitmapDescriptorFactory.fromResource(R.drawable.asistencia_social)
    );
    LatLng cdmx46 = new LatLng(19.242221,-99.093585);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx46)
        .title("Verificaciones")
        .snippet(cdmx46.latitude + "," + cdmx46.longitude)

    .icon(BitmapDescriptorFactory.fromResource(R.drawable.verificacion_de_la_lista_d
e_entrega_simbolo_de_portapapeles)
    );
    LatLng cdmx47 = new LatLng(19.23999999,-99.09423065);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx47)

```

```

        .title("Accesorios de Infraestructura")
        .snippet(cdmx47.latitude + "," + cdmx47.longitude)
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.pilon))
    );
    LatLng cdmx48 = new LatLng(19.23049,-99.097172);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx48)
        .title("Verificaciones")
        .snippet(cdmx48.latitude + "," + cdmx48.longitude)

        .icon(BitmapDescriptorFactory.fromResource(R.drawable.verificacion_de_la_lista_d
e_entrega_simbolo_de_portapapeles))
    );
    LatLng cdmx49 = new LatLng(19.243631,-99.089571);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx49)
        .title("Asistencia Social")
        .snippet(cdmx49.latitude + "," + cdmx49.longitude)

        .icon(BitmapDescriptorFactory.fromResource(R.drawable.asistencia_social))
    );
    LatLng cdmx50 = new LatLng(19.25651,-99.101846);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx50)
        .title("Accesorios de Infraestructuras")
        .snippet(cdmx50.latitude + "," + cdmx50.longitude)
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.pilon))
    );
    LatLng cdmx51 = new LatLng(19.20234742,-99.09836792);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx51)
        .title("Verificaciones")
        .snippet(cdmx51.latitude + "," + cdmx51.longitude)

        .icon(BitmapDescriptorFactory.fromResource(R.drawable.verificacion_de_la_lista_d
e_entrega_simbolo_de_portapapeles))
    );
    LatLng cdmx52 = new LatLng(19.336757,-99.122861);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx52)
        .title("Bacheo")
        .snippet(cdmx52.latitude + "," + cdmx52.longitude)

        .icon(BitmapDescriptorFactory.fromResource(R.drawable.asfaltadora))
    );
    LatLng cdmx53 = new LatLng(19.19978808,-99.10744271);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx53)
        .title("Accesorios de Infraestructura")
        .snippet(cdmx53.latitude + "," + cdmx53.longitude)
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.pilon))
    );
    LatLng cdmx54 = new LatLng(19.20971,-99.097992);
    markerCdmx1 = googleMap.addMarker(new MarkerOptions()
        .position(cdmx54)
        .title("Accesorios de Infraestructura")
        .snippet(cdmx54.latitude + "," + cdmx54.longitude)
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.pilon))
    );

```

```

);
LatLng cdmx55 = new LatLng(19.253514,-99.037284);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx55)
    .title("Desazolve")
    .snippet(cdmx55.latitude + "," + cdmx55.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.pala)
);

LatLng cdmx56 = new LatLng(19.208687,-99.12649);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx56)
    .title("Vigilancia Policiaca")
    .snippet(cdmx56.latitude + "," + cdmx56.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.vigpol)
);

LatLng cdmx57 = new LatLng(19.526323,-99.168525);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx57)
    .title("Luminarias Descompuestas")
    .snippet(cdmx57.latitude + "," + cdmx57.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.lampara)
);

LatLng cdmx58 = new LatLng(19.433342,-99.133097);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx58)
    .title("Banquetas o Guarniciones")
    .snippet(cdmx58.latitude + "," + cdmx58.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.calle)
);

LatLng cdmx59 = new LatLng(19.216187,-99.099383);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx59)
    .title("Vigilancia Policiaca")
    .snippet(cdmx59.latitude + "," + cdmx59.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.vigpol)
);

LatLng cdmx60 = new LatLng(19.308634,-99.141632);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx60)
    .title("Mantenimiento a Áreas Verdes")
    .snippet(cdmx60.latitude + "," + cdmx60.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.certificacion)
);

LatLng cdmx61 = new LatLng(19.331246,-99.270142);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx61)
    .title("Protección Civil")
    .snippet(cdmx61.latitude + "," + cdmx61.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.casco)
);

LatLng cdmx62 = new LatLng( 19.220804,-99.124388);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx62)
    .title("Vigilancia Policiaca")
    .snippet(cdmx62.latitude + "," + cdmx62.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.vigpol)
);

```

```

);
LatLng cdmx63 = new LatLng( 19.216187,-99.209383);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx63)
    .title("Poda de Árboles")
    .snippet(cdmx63.latitude + "," + cdmx63.longitude)

.icon(BitmapDescriptorFactory.fromResource(R.drawable.arbol_frutal)
);
LatLng cdmx64 = new LatLng(19.272258,-99.212723);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx64)
    .title("Poda de Árboles")
    .snippet(cdmx64.latitude + "," + cdmx64.longitude)

.icon(BitmapDescriptorFactory.fromResource(R.drawable.arbol_frutal)
);
LatLng cdmx65 = new LatLng(19.308551,-99.222679);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx65)
    .title("Apoyo Logísticos para eventos publicos")
    .snippet(cdmx65.latitude + "," + cdmx65.longitude)

.icon(BitmapDescriptorFactory.fromResource(R.drawable.trabajador_del_centro_de_a
tencion_telefonica_con_el_receptor_de_cabeza)
);
LatLng cdmx66 = new LatLng(19.560761,-99.140797);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx66)
    .title("Apoyo Logísticos para eventos publicos")
    .snippet(cdmx66.latitude + "," + cdmx66.longitude)

.icon(BitmapDescriptorFactory.fromResource(R.drawable.trabajador_del_centro_de_a
tencion_telefonica_con_el_receptor_de_cabeza)
);
LatLng cdmx67 = new LatLng(19.317299,-99.076939);
markerCdmx1 = googleMap.addMarker(new MarkerOptions()
    .position(cdmx67)
    .title("Semáforos descompuestos")
    .snippet(cdmx67.latitude + "," + cdmx67.longitude)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.sem)
);

// Delimitar a Ciudad de México con un rectángulo
PolylineOptions cdmxRect = new PolylineOptions()

    .add(new LatLng(19.583579, -99.254859)) // P1
    .add(new LatLng(19.583579, -99.138130)) // P2
    .add(new LatLng(19.475506,-99.138130)) // P3
    .add(new LatLng(19.475506,-99.254859)) // P4
    .add(new LatLng(19.583579,-99.254859)) // P1
    .color(Color.parseColor("#f44336")); // Rojo 500

Polyline polyline = mMap.addPolyline(cdmxRect);

mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(
    new LatLng(19.325250, -99.136070), 10));

```



```

    }

    @Override
    public boolean onSupportNavigateUp() {
        onBackPressed();
        return super.onSupportNavigateUp();
    }
}

```

x. RangoDeFecha.java

```

package com.herprogramacion.googlemapsenandroid;

import android.app.ProgressDialog;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
import android.widget.EditText;
import android.widget.ListView;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.NodeList;

import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

/**
 * Created by AleMontielSaavedra on 01/04/2017.
 */

public class RangoDeFecha extends AppCompatActivity {

    // Declare Variables
    ListView listviewfecha;
    ListViewAdapterRangoFecha adapterrangofecha;
    ProgressDialog mProgressDialog;
    EditText editsearch;

    static String MOTIVO = "Motivo";
    static String LATITUD = "Latitud";
    static String LONGITUD = "Longitud";
    static String FECHAHORAAPERTURA = "FechaHoraapertura";

    private List<VariablesDeIncidencias> datasetlist = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.listiview_main_rango_fecha);
    }
}

```

```

    new DownloadXML().execute();
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
}

private class DownloadXML extends AsyncTask<Void, Void, Void> {

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        // Create a progressdialog
        mProgressDialog = new ProgressDialog(RangoDeFecha.this);
        // Set progressdialog title
        mProgressDialog.setTitle("Rango de fecha");
        // Set progressdialog message
        mProgressDialog.setMessage("Cargando...");
        mProgressDialog.setIndeterminate(false);
        // Show progressdialog
        mProgressDialog.show();
    }

    @Override
    protected Void doInBackground(Void... params) {
        // Create the array
        datasetlist = new ArrayList<VariablesDeIncidencias>();
        XMLParser parser = new XMLParser();

        String xml = parser

.getXmlFromUrl("http://www.gobiernoabierto.cdmx.gob.mx/sigdata/resources/datasets/IncidenciasCDMXOctubredel2014-incidenciaoct2014.xml");
        // Retrive DOM element
        Document doc = parser.getDomElement(xml);

        try {

            NodeList nl = doc.getElementsByTagName("registro");
            for (int i = 0; i < nl.getLength(); i++) {
                Element e = (Element) nl.item(i);
                VariablesDeIncidencias map = new VariablesDeIncidencias();

                map.setmotivo(parser.getValue(e, MOTIVO));
                map.setlatitud(parser.getValue(e, LATITUD));
                map.setlongitud(parser.getValue(e, LONGITUD));
                map.setfechahoraapertura(parser.getValue(e,
FECHAHORAAPERTURA));

                datasetlist.add(map);
            }
        } catch (Exception e) {
            Log.e("Error", e.getMessage());
            e.printStackTrace();
        }
        return null;
    }

    @Override
    protected void onPostExecute(Void args) {

```

```

        listViewfecha = (ListView) findViewById(R.id.listViewfecha);

        adapterrangofecha = new ListViewAdapterRangoFecha (RangoDeFecha.this,
datasetlist);

        listViewfecha.setAdapter(adapterrangofecha);

        mProgressDialog.dismiss();
        // Locate the EditText in listView_main.xml
        editsearch = (EditText) findViewById(R.id.search);
        editsearch.addTextChangedListener(new TextWatcher() {

            @Override
            public void afterTextChanged(Editable arg0) {
                // TODO Auto-generated method stub
                String text = editsearch.getText().toString()
                    .toLowerCase(Locale.getDefault());
                adapterrangofecha.filter(text);
            }

            @Override
            public void beforeTextChanged(CharSequence arg0, int arg1,
                int arg2, int arg3) {
                // TODO Auto-generated method stub
            }

            @Override
            public void onTextChanged(CharSequence arg0, int arg1,
                int arg2, int arg3) {
                // TODO Auto-generated method stub
            }

        });
    }
    @Override
    public boolean onSupportNavigateUp() {
        onBackPressed();
        return super.onSupportNavigateUp();
    }
}

```

xi. ListViewAdapterRangoFecha.java

```

package com.herprogramacion.googlemapsenandroid;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

```

```

import java.util.Locale;

public class ListViewAdapterRangoFecha extends BaseAdapter {

    private final List<VariablesDeIncidencias> datasetlist; // <WorldPopulation> ES
UNA
    Context context;
    LayoutInflater inflater;
    ArrayList<HashMap<String, String>> data;
    ImageLoader imageLoader;
    private List<VariablesDeIncidencias> dataset = null;
    private ArrayList<VariablesDeIncidencias> arraylist;

    public ListViewAdapterRangoFecha(Context context,
                                     List<VariablesDeIncidencias> datasetlist) {
        this.context = context;
        this.datasetlist = datasetlist;
        inflater = LayoutInflater.from(context);
        this.arraylist = new ArrayList<VariablesDeIncidencias>();
        this.arraylist.addAll(datasetlist);
        imageLoader = new ImageLoader(context);
    }

    public class ViewHolder {
        TextView Motivo;
        TextView Latitud;
        TextView Longitud;
        TextView Fecha;
    }

    @Override
    public int getCount() {
        return datasetlist.size();
    }

    @Override
    public Object getItem(int position) {
        return datasetlist.get(position);
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    public View getView(final int position, View view, ViewGroup parent) {
        final ViewHolder holder;
        if (view == null) {
            holder = new ViewHolder();
            view = inflater.inflate(R.layout.listview_item_rango_fecha, null);

            holder.Motivo = (TextView) view.findViewById(R.id.Motivo);
            holder.Latitud = (TextView) view.findViewById(R.id.Latitud);
            holder.Longitud = (TextView) view.findViewById(R.id.Longitud);
            holder.Fecha = (TextView) view.findViewById(R.id.Fecha);
        }
    }
}

```

```

        view.setTag(holder);
    } else {
        holder = (ViewHolder) view.getTag();
    }

    holder.Motivo.setText(datasetlist.get(position).getmotivo());
    holder.Latitud.setText(datasetlist.get(position).getlatitud());

holder.Longitud.setText(datasetlist.get(position).getlongitud()); //getRank

holder.Fecha.setText(datasetlist.get(position).getfechahoraapertura()); //getRank

    view.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
            // Send single item click data to SingleItemView Class
            /* Intent intent = new Intent(context,
SingleItemViewRangoFecha.class);

            intent.putExtra("Motivo",
                (datasetlist.get(position).getmotivo()));

            intent.putExtra("Latitud",
                (datasetlist.get(position).getlatitud()));

            intent.putExtra("Longitud",
                (datasetlist.get(position).getlongitud()));

            intent.putExtra("FechaHoraapertura",
                (datasetlist.get(position).getfechahoraapertura()));

        }
    });
    return view;
}

// Filter Class
public void filter(String charText) { // para filtrar datos

    charText = charText.toLowerCase(Locale.getDefault());
    datasetlist.clear();
    if (charText.length() == 0) {
        datasetlist.addAll(arraylist);
    } else {
        for (VariablesDeIncidencias wp : arraylist) {
            if
(wp.getfechahoraapertura().toLowerCase(Locale.getDefault()).contains(charText))
{

                datasetlist.add(wp);

            }
        }
    }
}

```

```

    }
    notifyDataSetChanged();
}
}

```

10.2 Directorio res/layout (clases de vista).

i. activity_intro.xml

```

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_intro"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.herprogramacion.googlemapsenandroid.IntroActivty">

    <android.support.v4.view.ViewPager
        android:id="@+id/view_pager"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <LinearLayout
        android:id="@+id/layoutDots"
        android:layout_width="match_parent"
        android:layout_height="@dimen/layout_height"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="@dimen/layout_bottom_margin"
        android:gravity="center"
        android:orientation="horizontal">

        <TextView
            android:text=" . . ."
            android:layout_width="wrap_content"
            android:layout_height="50dp"
            android:id="@+id/textView4"
            android:layout_weight="1"
            android:textStyle="normal|bold"
            android:textSize="40sp"
            android:textColor="@android:color/holo_blue_bright"
            android:textAlignment="center" />
    </LinearLayout>

    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:alpha="5"
        android:layout_above="@id/layoutDots"
        android:background="?android:attr/colorForeground" />

    <Button
        android:id="@+id/btn_next"
        android:layout_width="wrap_content"

```

```

    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:background="@android:color/black"
    android:text="@string/next"
    android:onClick="btnNextClick"
    android:textColor="@android:color/white"
    android:textAlignment="center" />

```

```
<Button
```

```

    android:id="@+id/btn_skip"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    android:background="@android:color/black"
    android:text="@string/skip"
    android:onClick="btnSkipClick"
    android:textColor="@android:color/white"
    android:textAlignment="center" />

```

```
</RelativeLayout>
```

ii. slide1.xml

```
<?xml version="1.0" encoding="utf-8" ?>
```

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="wrap_content"
    android:background="@color/bg_screen4"
    tools:context=".MainActivity"
    android:layout_height="wrap_content">

```

```
<TextView
```

```

    android:layout_height="wrap_content"
    android:text="@string/title_slide1"
    android:textColor="?android:attr/colorMultiSelectHighlight"
    android:textSize="@dimen/slide_title"
    android:textStyle="bold"
    android:textAlignment="center"
    android:fontFamily="serif-monospace"
    android:layout_width="match_parent"

    android:layout_centerHorizontal="true"
    android:layout_marginTop="45dp" />

```

```
<TextView
```

```

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

```

```

    android:text="@string/desc_slide4"
    android:textAlignment="center"
    android:textColor="@android:color/white"
    android:textSize="16sp"
    android:layout_weight="0.88"
    android:layout_centerVertical="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginBottom="50dp"
    android:paddingBottom="100dp" />

```

```

<ImageView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:src="@drawable/mexico"
    android:layout_weight="1.10"
    android:scaleType="centerInside"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

```

```
</RelativeLayout>
```

iii. slide2.xml

```
<?xml version="1.0" encoding="utf-8" ?>
```

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/bg_screen3">

```

```

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:weightSum="1"
        android:orientation="vertical"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true">

```

```

        <ImageView
            android:layout_height="300dp"
            android:src="@drawable/data"

            android:layout_weight="0.67"
            android:layout_width="300dp"
            android:layout_marginTop="20dp" />

```

```

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

```



```

        android:text="@string/desc_slide_fecha_tipos"
        android:textAlignment="center"
        android:textColor="@android:color/white"
        android:textAppearance="@style/TextAppearance.AppCompat"
        android:textStyle="normal|bold"
        android:textSize="24sp"
        android:paddingStart="10dp" />
    </LinearLayout>
</RelativeLayout>

```

iv. slide3.xml

```

<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/bg_screen5">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:orientation="vertical"
        android:weightSum="1"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true">

        </LinearLayout>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/desc_slideproblems"
            android:textColor="@android:color/black"
            android:textSize="18sp"

            android:id="@+id/textView"
            android:layout_marginTop="45dp"
            android:lineSpacingExtra="8sp"
            android:typeface="sans"
            android:textAlignment="center" />

        <ImageView
            android:layout_width="match_parent"
            android:layout_height="310dp"
            android:src="@drawable/street"
            android:scaleType="centerCrop"
            android:layout_below="@+id/textView"
            android:layout_toRightOf="@+id/linearLayout"
            android:layout_toEndOf="@+id/linearLayout" />
    </RelativeLayout>

```

v. menu.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#08867b">

    <TextView
        android:text="ELIJA UNA OPCIÓN:"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="48dp"
        android:id="@+id/textView2"
        android:textStyle="normal|bold"
        android:textSize="30sp"
        android:fontFamily="sans-serif"
        android:textColor="@color/bg_screen4"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true" />

    <android.support.v7.widget.RecyclerView
        android:id="@+id/lista"
        app:layoutManager="LinearLayoutManager"
        android:layout_marginLeft="@dimen/activity_horizontal_margin"
        android:layout_marginRight="@dimen/activity_horizontal_margin"
        android:theme="@style/SeleccionAzul"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="8dp"
        android:background="#e7f79f4d"
        android:layout_marginTop="63dp"
        android:layout_below="@+id/textView2"
        android:layout_centerHorizontal="true" />

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/salida"
        android:layout_marginEnd="20dp"
        android:id="@+id/exit"
        android:clickable="true"
        android:layout_alignBottom="@+id/info"
        android:layout_alignEnd="@+id/textView2"
        android:layout_alignTop="@+id/info" />

    <ImageView
        android:layout_width="70dp"
        app:srcCompat="@drawable/informacion"
        android:id="@+id/info"
        android:layout_height="60dp"
        android:layout_marginStart="22dp"
        android:layout_marginBottom="28dp"
```

```

        android:clickable="true"
        android:layout_alignParentBottom="true"
        android:layout_alignStart="@+id/lista" />
</RelativeLayout>

```

vi. portada_autora.xml

```

<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/fondoblack">

    <TextView
        android:text="Versión 1.0.0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView5"
        android:layout_centerHorizontal="true"
        android:id="@+id/version"
        android:textSize="18sp"
        android:textColor="@color/bg_screen5" />

    <TextView
        android:text="Visualización de Incidencias en la Ciudad de México"
        android:layout_height="wrap_content"
        android:id="@+id/textView5"
        android:textStyle="normal|bold"
        android:textAlignment="center"
        android:textColor="#fcfcfc"
        android:textSize="24sp"
        android:layout_width="wrap_content"
        android:layout_marginTop="50dp"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true" />

    <TextView
        android:text="TextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/uamazc"
        android:layout_centerHorizontal="true"
        android:id="@+id/textView2" />

    <TextView
        android:text=".@2017 Vis. Incidencias"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/trimestre"
        android:layout_marginTop="16dp"
        android:textSize="18sp"
        android:textColor="@color/bg_screen1"
        android:layout_below="@+id/textView2"
        android:layout_centerHorizontal="true" />

```

```

<ImageView
    android:layout_width="90dp"
    android:layout_height="90dp"
    app:srcCompat="@drawable/senal_de_trafico"
    android:id="@+id/imageView2"
    android:layout_marginTop="36dp"
    android:layout_below="@+id/version"
    android:layout_centerHorizontal="true" />

<TextView
    android:text="Hecho por Alejandra Montiel S."
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/trimestre"
    android:layout_centerHorizontal="true"
    android:id="@+id/hechopor"
    android:textColor="@color/bg_screen5"
    android:textSize="18sp" />

<TextView
    android:text="UAM Azcapotzalco"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/uamazc"
    android:textSize="18sp"
    android:textColor="@color/bg_screen1"
    android:layout_marginTop="63dp"
    android:layout_below="@+id/imageView2"
    android:layout_centerHorizontal="true" />

<TextView
    android:text="PROYECTO TERMINAL"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/tituloproyecto"
    android:textColor="@color/bg_screen1"
    android:layout_below="@+id/uamazc"
    android:layout_alignStart="@+id/uamazc" />

<TextView
    android:text="Ingeniería en Computación"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="33dp"
    android:id="@+id/cbi"
    android:textSize="18sp"
    android:textColor="@color/bg_screen1"
    android:layout_below="@+id/tituloproyecto"
    android:layout_centerHorizontal="true" />

</RelativeLayout>

```

vii. listview_incidencias.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <EditText
        android:id="@+id/search"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textAppearance="@style/TextAppearance.AppCompat.Display1"

        android:background="#13676787">

        <requestFocus />
    </EditText>

    <ListView
        android:id="@+id/listview"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_below="@+id/search"
        android:background="#91d0ef"
        android:cacheColorHint="@android:color/black"
        android:drawSelectorOnTop="false"
        android:duplicateParentState="false" />

    <ImageView
        app:srcCompat="@drawable/magnify"
        android:id="@+id/imageView"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_above="@+id/listview"
        android:layout_alignParentEnd="true" />

</RelativeLayout>
```

viii. listview_item.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <TextView
        android:id="@+id/Motivolabel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:text="@string/Motivolabel"
```

```

        android:textStyle="normal|bold"
        android:textColor="@android:color/black"
        android:layout_margin="5dp"
        android:textSize="16sp"
        android:textAppearance="@style/TextAppearance.AppCompat" />
<TextView
    android:id="@+id/Motivo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:layout_toRightOf="@+id/Motivolabel"
    android:layout_margin="6dp"
    android:textSize="15sp"
    android:textStyle="normal|italic" />

<TextView
    android:id="@+id/Latitudlabel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/Motivolabel"
    android:text="@string/Latitudlabel"
    android:textStyle="normal|bold"
    android:textColor="@android:color/holo_red_dark"
    android:layout_marginStart="10dp" />
<TextView
    android:id="@+id/Latitud"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/Motivo"
    android:layout_toRightOf="@+id/Latitudlabel"
    android:layout_marginStart="10dp" />

<TextView
    android:id="@+id/Longitudlabel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/Latitudlabel"
    android:text="@string/Longitudlabel"
    android:textStyle="normal|bold"
    android:textColor="@android:color/holo_red_dark"
    android:layout_marginEnd="10dp"
    android:layout_marginStart="10dp" />
<TextView
    android:id="@+id/Longitud"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/Latitud"
    android:layout_toRightOf="@+id/Longitudlabel" />

<TextView
    android:id="@+id/FechaHoraaperturalabel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/Longitudlabel"
    android:text="@string/FechaHoraaperturalabel"
    android:textStyle="normal|bold"
    android:layout_marginLeft="10dp"

```

```

        android:textColor="@android:color/holo_red_dark" />
<TextView
    android:id="@+id/FechaHoraapertura"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/Longitud"
    android:layout_toRightOf="@+id/FechaHoraaperturaalabel"
    android:layout_marginLeft="5dp"
    android:layout_marginTop="2dp" />

<ImageView
    android:id="@+id/flag"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:background="#000000"
    android:padding="1dp" />

```

```
</RelativeLayout>
```

ix. mapa_coordenadas.xml

```

<?xml version="1.0" encoding="utf-8" ?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:name="com.google.android.gms.maps.SupportMapFragment" />

```

x. listview_main_rango_fecha.xml

```

<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:background="#171818"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

<ListView
    android:id="@+id/listviewfecha"
    android:layout_width="250dp"
    android:background="#bcfa4921"
    android:cacheColorHint="@android:color/black"
    android:drawSelectorOnTop="false"
    android:duplicateParentState="false"
    android:layout_height="300dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true" />

```

```

<TextView
    android:text="/10/2014"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textView3"
    android:textStyle="normal|bold"
    android:textSize="36sp"
    android:textColor="@android:color/darker_gray"
    android:layout_alignBaseline="@+id/search"
    android:layout_alignBottom="@+id/search"
    android:layout_alignEnd="@+id/search" />

<EditText
    android:id="@+id/search"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:background="#dcf1e56a"
    android:layout_height="80dp"
    android:layout_width="190dp"
    android:layout_marginStart="74dp"
    android:layout_marginTop="55dp"
    android:layout_alignParentTop="true"
    android:layout_alignParentStart="true">
    <requestFocus />
</EditText>

<TextView
    android:text="-----"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textView9"
    android:textColor="@android:color/holo_red_dark"
    android:textStyle="normal|bold"
    android:layout_marginBottom="20dp"
    android:layout_above="@+id/textView3"
    android:layout_alignStart="@+id/search" />

<TextView
    android:text="-----"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textView6"
    android:textColor="@android:color/holo_red_dark"
    android:textStyle="normal|bold"
    android:layout_below="@+id/search"
    android:layout_alignStart="@+id/search" />

</RelativeLayout>

```


xi. listview_item_rango_fecha.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <TextView
        android:id="@+id/Fechalabel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:text="@string/Fechalabel"
        android:textStyle="normal|bold"
        android:layout_marginLeft="10dp"
        android:textColor="@android:color/holo_blue_bright"
        android:textSize="18sp" />

    <TextView
        android:id="@+id/Fecha"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:layout_toRightOf="@+id/Fechalabel"
        android:layout_marginLeft="5dp"
        android:layout_marginTop="2dp" />

    <TextView
        android:id="@+id/Motivolabel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/Fechalabel"
        android:text="@string/Motivolabel"
        android:textStyle="normal|bold"
        android:textColor="@android:color/background_light"
        android:layout_margin="5dp"
        android:textSize="14sp"
        android:textAppearance="@style/TextAppearance.AppCompat" />

    <TextView
        android:id="@+id/Motivo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/Fecha"
        android:layout_toRightOf="@+id/Motivolabel"
        android:layout_margin="6dp"
        android:textSize="15sp"
        android:textStyle="normal|italic" />

    <TextView
        android:id="@+id/Latitudlabel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/Motivolabel"
        android:text="@string/Latitudlabel"
        android:textStyle="normal|bold"
```

```

        android:textColor="@android:color/holo_green_light"
        android:layout_marginStart="10dp" />
<TextView
    android:id="@+id/Latitud"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/Motivo"
    android:layout_toRightOf="@+id/Latitudlabel"
    android:layout_marginStart="10dp" />

<TextView
    android:id="@+id/Longitudlabel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/Latitudlabel"
    android:text="@string/Longitudlabel"
    android:textStyle="normal|bold"
    android:textColor="@android:color/holo_green_light"
    android:layout_marginEnd="10dp"
    android:layout_marginStart="10dp" />
<TextView
    android:id="@+id/Longitud"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/Latitud"
    android:layout_toRightOf="@+id/Longitudlabel" />

<ImageView
    android:id="@+id/flag"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:background="#000000"
    android:padding="1dp" />
</RelativeLayout>

```

10.3 Otros

i. AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.herprogramacion.googlemapsenandroid">
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <application
        android:allowBackup="true"
        android:icon="@drawable/senal_de_trafico"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <activity android:name=".IntroActivty"><intent-filter>

```

```

        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

<activity android:name=".Menu">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
    </intent-filter>
</activity>

<activity android:name=".Informacion">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
    </intent-filter>
</activity>

<activity
    android:name=".ListasDeIncidencias"
    android:label="@string/title_listview_main" />

    <activity
        android:name=".Mapa"
        android:label="@string/title_activity_polyline" />
    <activity
        android:name=".RangoDeFecha"
        android:label="@string/title_listview_main_rango_fecha" />
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="@string/google_maps_key" />
</application>

</manifest>

```

ii. colors.xml

```

<?xml version="1.0" encoding="utf-8" ?>
<resources>
    <color name="colorPrimary">#eee15d0b</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
    <color name="bg_screen1">#b7f7e9</color>
    <color name="bg_screen2">#46aa03</color>
    <color name="bg_screen3">#a31d62</color>
    <color name="bg_screen4">#000000</color>
    <color name="bg_screen5">#efeded</color>
    <color name="bg_screen6">#e5719f</color>
</resources>

```

iii. google_maps_api.xml

```

<resources>

    <string name="google_maps_key" templateMergeStrategy="preserve"
    translatable="false">AIzaSyBgHO7z1Xye03n3rYBF-Jist0jp0o1M5b8</string>
</resources>

```

iv. strings.xml

```
<resources>
  <string name="app_name">Visualización de Incidencias. </string>

  <!-- TODO: Remove or change this placeholder text -->
  <string name="hello_blank_fragment">Hello blank fragment</string>
  <string name="marker_detail_latlng">(%1$.2f, %2$.2f)</string>

  <string name="Motivolabel">"INCIDENCIA: "</string>
  <string name="Latitudlabel">"LATITUD: "</string>
  <string name="Longitudlabel">"LONGITUD: "</string>
  <string name="FechaHoraaperturalabel">"FECHA Y HORA: "</string>
  <string name="Fechalabel">"--> FECHA: "</string>
  <string name="title_listview_main">LISTA DE INCIDENCIAS</string>
  <string name="title_activity_polyline">RANGO DE COORDENADAS</string>
  <string name="title_listview_main_rango_fecha">RANGO DE FECHA</string>

  <!-- PANTALLA DE BIENVENIDA O INTRODUCCIÓN-->
  <string name="next">SIGUIENTE</string>
  <string name="skip">OMITIR</string>
  <string name="start">EMPEZAR</string>
  <string name="title_slidel">VISUALIZACIÓN DE INCIDENCIAS DE LA CIUDAD DE
MEXICO</string>
  <string name="desc_slidel">En la mayoría de los casos se presentan sin aviso
y en otras son el resultado de la indiferencia, descuido, deterioro de un sistema
o evento.</string>
  <string name="title_slide2">Battery status update</string>
  <string name="desc_slide2">It will keep you updated on battery charging
status through notification and home screen widget</string>
  <string name="title_slide3">Alert on 100%</string>
  <string name="desc_slide3">It will alert you when battery reaches
100%</string>
  <string name="title_slide4">VISUALIZACIÓN DE INCIDENCIAS DE LA CIUDAD DE
MEXICO</string>
  <string name="desc_slide4">En la mayoría de los casos se presentan sin aviso
y en otras son el resultado de la indiferencia, descuido, deterioro de un sistema
o evento.</string>
  <string name="desc_slideproblems">ESTOS INCLUYEN:
Luminarias descompuestas, bacheo, accesorios de infraestructura, verificaciones,
semáforos descompuestos, vigilancia policiaca,etc...</string>
  <string name="desc_slide_fecha_tipos">Se mostrará el mapa de la Ciudad de
México y las más importantes incidencias.</string>
</resources>
```

v. styles.xml

```
<resources>
  <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
  </style>

  <style name="SeleccionAzul" parent="Theme.AppCompat.Light">
```

```

        <item name="colorControlHighlight">#9FA8DA</item>
    </style>
</resources>

```

10.4 Gradle Scripts

i. build.gradle (Dependencias de la aplicación)

```

apply plugin: 'com.android.application'

android {
    android {
        compileSdkVersion 25
        buildToolsVersion "25.0.2"
        defaultConfig {
            applicationId "com.herprogramacion.googlemapsenandroid"
            minSdkVersion 19
            targetSdkVersion 25
            versionCode 1
            versionName "1.0"
            testInstrumentationRunner
            "android.support.test.runner.AndroidJUnitRunner"
            multiDexEnabled true
        }
        buildTypes {
            release {
                minifyEnabled false
                proguardFiles getDefaultProguardFile('proguard-android.txt'),
                'proguard-rules.pro'
            }
        }
        dependencies {
            compile fileTree(dir: 'libs', include: ['*.jar'])
            androidTestCompile('com.android.support.test.espresso:espresso-
            core:2.2.2', {
                exclude group: 'com.android.support', module: 'support-
            annotations'
            })
            compile 'com.android.support:appcompat-v7:25.1.1'
            compile
            'org.jbundle.util.osgi.wrapped:org.jbundle.util.osgi.wrapped.org.apache.http.cli
            ent:4.1.2'

            compile 'com.google.android.gms:play-services-places:10.2.0'
            compile 'com.android.support:support-v4:25.1.1'
            compile 'com.android.support:recyclerview-v7:25.1.1'
            compile
            'org.jbundle.util.osgi.wrapped:org.jbundle.util.osgi.wrapped.org.apache.http.cli
            ent:4.1.2'

            testCompile 'junit:junit:4.12'
        }
    }
}

```

RECURSOS:

El hardware que se utilizó para la realización del proyecto es el siguiente:

- Una Notebook acer Aspire E5-551-t04N, con un disco duro de 1Tb, memoria RAM de 12GB y un procesador AMD A10-7300 Radeon R6, 10 Compute Cores 4C+6G de 1.90GHz.
- Dispositivo móvil HUAWEI Ascend G700, con sistema operativo Android 4.4.1, con memoria interna de 8GB y un procesador a 1.2GHz. 2GB de RAM.

El software utilizado son:

- Microsoft Windows 10 Pro 64-bit (Notebook)
- Oracle WebLogic Server 12 Cr2
- Android Studio IDE
- Android SDK tools
- Android Studio 2.2.3
- Android emulator system image with Google APIs.
- Android 4.4 (KitKat)

REFERENCIAS BIBLIOGRÁFICAS

- [1] Almeida Hiram, “Portal de Datos Abiertos de la Ciudad de México”, http://www.infodf.org.mx/innovaciones/transparencia/2013/2013_02_CG_CedulaProyecto.pdf, 2013.
- [2] Cárdenas, Guillermo, “Ciudades Inteligentes”, ¿Cómo ves?, No. 202. pp. 10-14, 2015.
- [3] J. P. Alcántara Olivares y J. D. López Jaimes, “Sistema de gestión escolar en Android”, proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2007.
- [4] Hernández Muñoz Víctor, “Aplicación para dispositivos con Android que encuentre la mejor ruta entre dos estaciones del Metro”, proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2013.
- [5] Ciudad de México, “Agencia de Gestión Urbana de la Ciudad de México”, <http://www.agu.cdmx.gob.mx/>, 2013
- [6] Secretaría de Seguridad Pública, “Mi Policía”; <http://www.ssp.df.gob.mx/mi-policia.html>, 2016
- [7] Fiscalía General del Estado de Jalisco, “El botón de pánico Jalisco”, <http://fge.jalisco.gob.mx/>, 2014
- [8] S. Mohamed Amine, “Utilización de servicios Web en dispositivos móviles Android”, proyecto final de carrera, Escuela Técnica Superior de Ingeniería Informática, Universidad Politécnica de Valencia, España, 2014
- [9] Coordenadas Geográficas en Google Maps, “Sistema de Coordenadas”, <http://www.coordenadas-gps.com/>, 2015.
- [10] Android, “KitKat 4.4”, https://www.android.com/intl/es-419_mx/versions/kit-kat-4-4/, 2014
- [11] Zanini Viviana, “Android Studio 2”, Edit. USERS 1ª edición, Ciudad Autónoma de Buenos Aires, 2016.
- [12] Wolfson, Mike, “Android Developer Tools Essentials”, Edit. O’reilly, 2a. edición, USA, 2013