

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO
DIRECCIÓN ACADÉMICA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
CUADERNO DE PRÁCTICAS

Carrera	Plan de estudios	Clave materia	Nombre de la materia	Semestre	Grupo	Periodo
ISC	ISIC-2010-224	DWD-1703	DESARROLLO DE APLICACIONES ANDROID II	9		2020-1

Practica No.	Laboratorio de:	Nombre de la practica	Duración (Hora)
1	CÓMPUTO	3.1 Manejo de API de Google maps 3.1.1 Configuración 3.1.2 Manejo de Propiedades 3.2 Uso de LocationManager de Android 3.3 Uso de LocationProvider de Android 3.4 Uso de la clase Criteria	24HRS

Unidad temática: Geolocalización y mapas	Subtema: 2.3 Sistema de archivos interno y externo
Competencia específica materia:	
Competencia de la práctica: Programa e implementa geolocalización y mapas en aplicaciones móviles.	
Evaluación: 50% prácticas de laboratorio	
<p>Introducción: Mapas en Android – Google Maps Android API (1) 12/09/2016</p> <p>Curso Programación Android Este artículo forma parte del Curso de Programación Android que tienes disponible de forma completamente gratuita en sgoliver.net</p> <p>La API de Google Maps se integró con los Google Play Services allá por finales de 2012. Este cambio trajo consigo importantes mejoras, como la utilización de mapas vectoriales y mejoras en el sistema de caché, lo que proporcionaba mayor rendimiento, mayor velocidad de carga, y menor consumo de datos.</p> <p>También llegó con un cambio en la forma en que los desarrolladores interactuaríamos con los mapas, pasando de los antiguos MapActivity y MapView a un nuevo tipo de fragment llamado MapFragment, con las ventajas que conlleva el uso de este tipo de componentes.</p> <p>Antes de empezar a utilizar esta API en nuestras aplicaciones será necesario realizar algunos preparativos, y es que para hacer uso de los servicios de Google Maps es necesario que</p>	

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO
DIRECCIÓN ACADÉMICA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
CUADERNO DE PRÁCTICAS

previamente generemos una Clave de API (o API key) asociada a nuestra aplicación. Éste es un proceso sencillo y se realiza accediendo a la [Consola de Desarrolladores](#) de Google.

Una vez hemos accedido, tendremos que crear un nuevo proyecto desde la lista desplegable que aparece en la parte superior derecha y seleccionando la opción «Crear proyecto...»

Aparecerá entonces una ventana que nos solicitará el nombre del proyecto. Introducimos algún nombre descriptivo, se generará automáticamente un ID único (que podemos editar aunque no es necesario), y aceptamos pulsando «Crear».

Una vez creado el proyecto llegamos a una página donde se nos permite seleccionar las APIs de Google que vamos a utilizar (como podéis ver la lista es bastante extensa). En nuestro caso particular vamos a seleccionar «*Google Maps Android API*».

Aparecerá entonces una ventana informativa con una breve descripción de la API y una advertencia indicando que se necesitará una clave de API para poder utilizarla. Por el momento vamos a activar la API haciendo click sobre la opción «*HABILITAR*» que aparece en la parte superior.

Una vez activada nos vuelve a aparecer la advertencia sobre la necesidad de obtener credenciales para el uso de la API por lo que, ahora sí, tendremos que iniciar el proceso de obtención de la clave. Pulsaremos para ello sobre el botón «*Ir a las credenciales*».

Esto iniciará un pequeño asistente. En el primer paso nos volverán a preguntar qué API vamos a utilizar (aunque debería aparecer seleccionada por defecto la de Google Maps para Android), y desde dónde vamos a utilizarla. Indicamos «Android» y pulsamos el botón «¿Qué tipo de credenciales necesito?«.

En el segundo paso tendremos que poner un nombre descriptivo a la clave de API que se va a generar, no es demasiado relevante, por lo que podemos dejar el que nos proponen por defecto. También en este paso se nos da la posibilidad de poder restringir el uso de este proyecto a determinadas aplicaciones concretas. Como es una práctica bastante recomendable vamos a ver cómo hacerlo. Pulsaremos sobre el botón «+ Añadir nombre de paquete y huella digital» y veremos que se solicitan dos datos:

- Nombre de paquete
- Huella digital de certificado SHA-1

El primero de ellos es simplemente el paquete java principal que utilizaremos en nuestra aplicación. Lo indicaremos al crear nuestro proyecto en Android Studio (o si ya tenemos una aplicación creada podemos encontrarlo en el fichero AndroidManifest.xml, en el atributo package del elemento principal). En mi caso de ejemplo utilizaré el paquete «net.sgoliver.android.mapas».

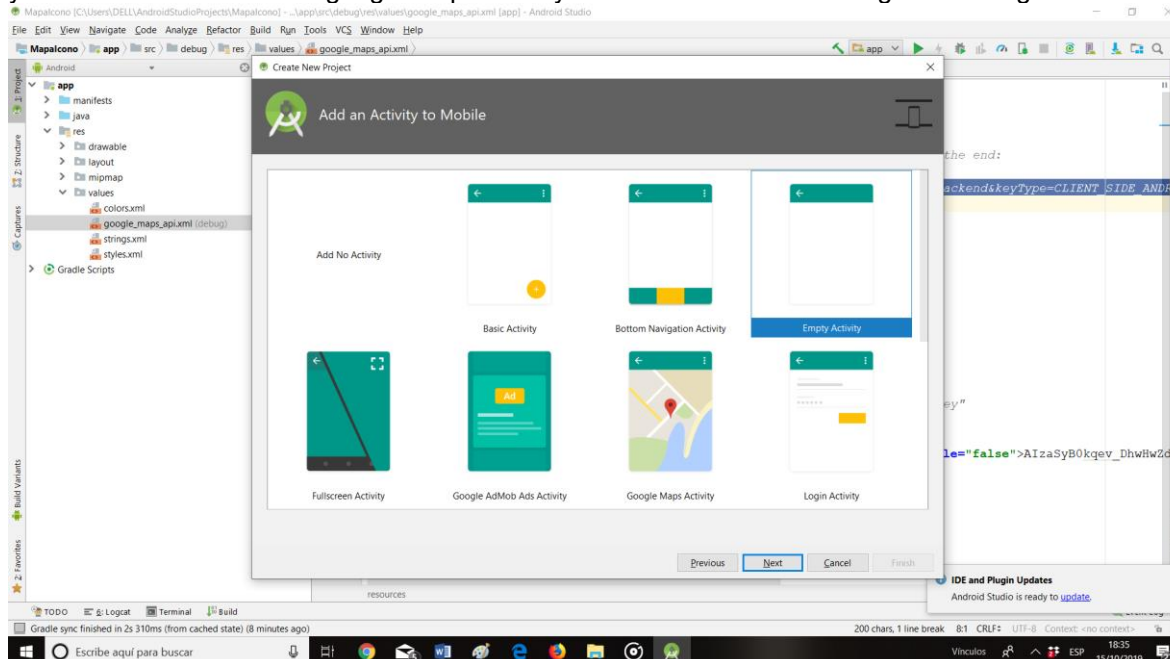
El segundo de los datos requiere más explicación. Toda aplicación Android debe ir firmada para poder ejecutarse en un dispositivo, tanto físico como emulado. Este proceso de firma es uno de los pasos que tenemos que hacer siempre antes de distribuir públicamente una aplicación. Adicionalmente, durante el desarrollo de la misma, para realizar las pruebas y la depuración del código, aunque no seamos conscientes de ello también estamos firmado la aplicación con un

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO
DIRECCIÓN ACADÉMICA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
CUADERNO DE PRÁCTICAS

“certificado de pruebas”. Pues bien, para obtener la huella digital del certificado con el que estamos firmando la aplicación podemos utilizar la utilidad *keytool* que se proporciona en el SDK de Java. **Importante:** en este ejemplo vamos a obtener el SHA1 del certificado de pruebas, que es el que se utilizará para probar en el emulador, pero en caso de que nuestra aplicación se firmara de nuevo para subirla a la tienda de Google tendríamos que obtener de nuevo el SHA1 del nuevo certificado, o de lo contrario los mapas no se visualizarán.

DESARROLLO

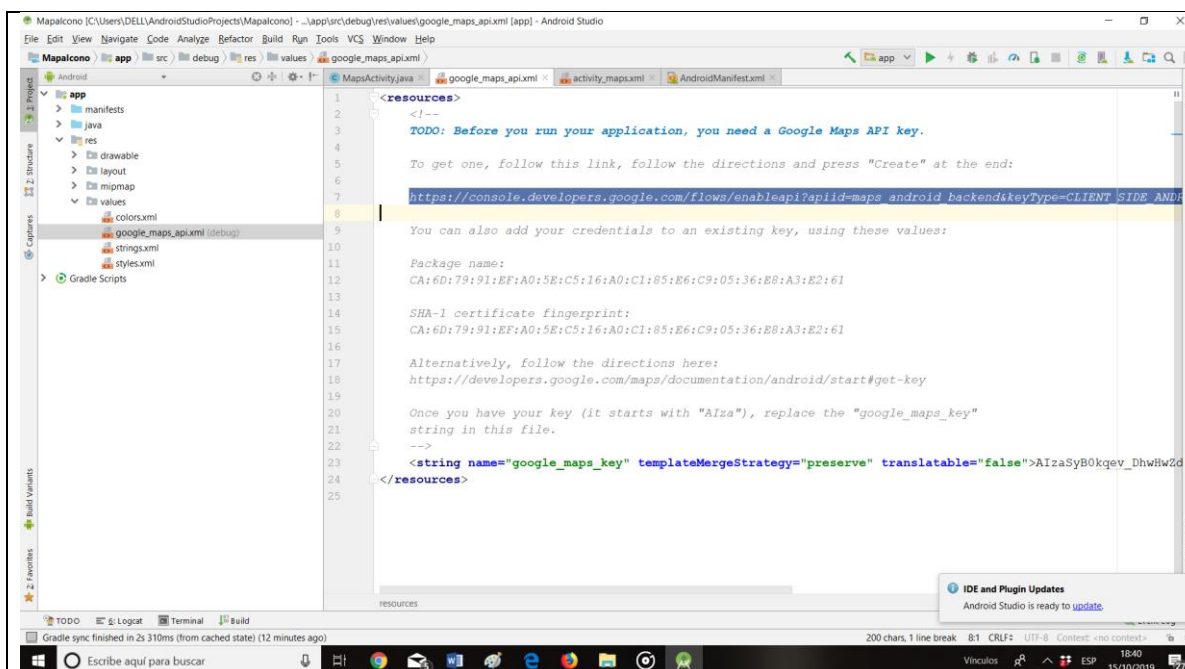
Otra forma de hacerlo sin tener una cuenta google play services como desarrollador en la cual tendras que pagar 25 USD, es la siguiente alternativa; crea un nuevo proyecto en Android Studio y selecciones la actividad “google maps activity” como su observa en la siguiente imagen



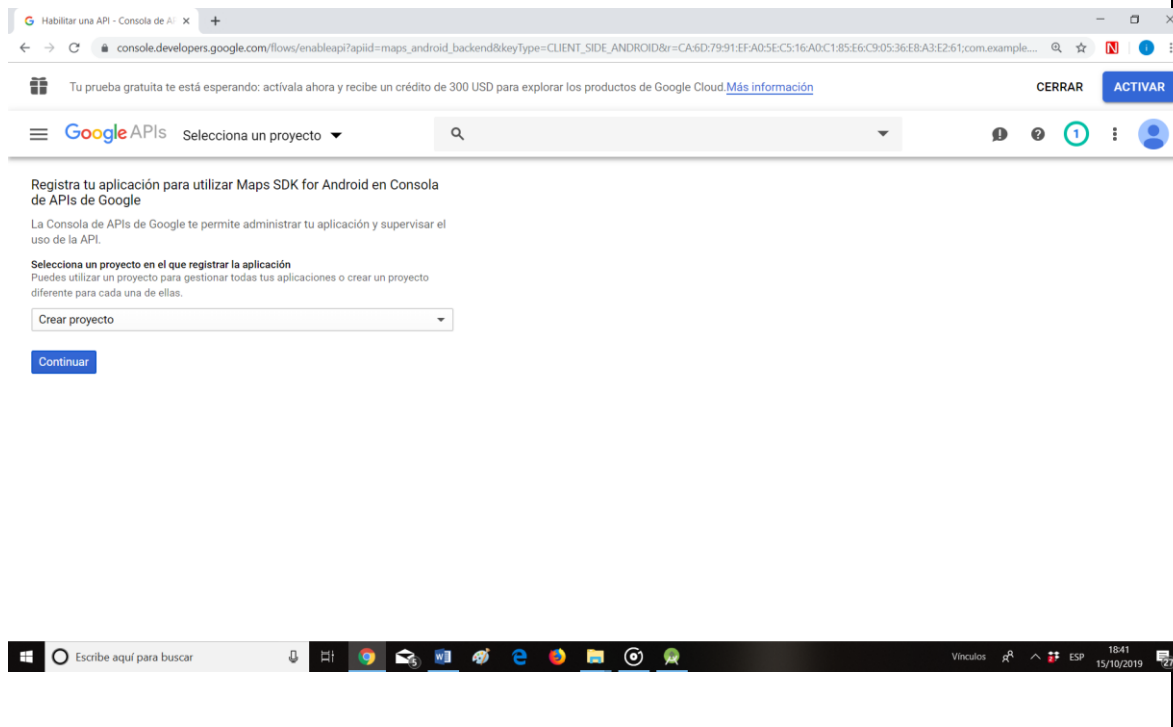
Dirígete al archivo `google_maps_api.xml` que se encuentra dentro de la carpeta “values” dentro de los recursos y en el código comentado encuentra la url que genera la llave, es parecida a esta https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&keyType=CLIENT_SIDE_ANDROID&r=CA:6D:79:91:EF:A0:5E:C5:16:A0:C1:85:E6:C9:05:36:E8:A3:E2:61%3Bcom.example.dell.mapaicono

Esta indicado en la siguiente figura en donde puedes copiar la url.

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO
DIRECCIÓN ACADÉMICA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
CUADERNO DE PRÁCTICAS



Copie y pegue la url en un navegador web y observará la siguiente ventana en la cual continuará creando el proyecto



TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO
DIRECCIÓN ACADÉMICA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
CUADERNO DE PRÁCTICAS

Genera la clave API como se muestra en las siguientes ventanas

The first screenshot shows the Google Cloud Platform console with the 'APIs and Services' page. The 'Credentials' tab is selected, and a 'Create API key' dialog box is open. The dialog box contains the text: 'Clave de API creada', 'Para usar esta clave en tu aplicación, transfírela como un parámetro key=API_KEY', and 'Tu clave de API: AIzaSyA6By0k1sKpgakVnIQ7D9db2v10z2BTMBU'. There is also a warning icon and text: '⚠ Restringe la clave para impedir el uso no autorizado en producción.' and buttons for 'CERRAR' and 'RESTRINGIR CLAVE'.

The second screenshot shows the same console page, but the 'API key created' dialog box is still open, displaying the same information as the first screenshot.

Copia y pega tu llave en el archivo google_maps_api.xml donde se indica

```
<string name="google_maps_key" templateMergeStrategy="preserve"
translatable="false">AIzaSyBuw5GCP0Vj8YhPyZdTgE5MayL8L1wZeVv</string>
</resources>
```

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO
DIRECCIÓN ACADÉMICA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
CUADERNO DE PRÁCTICAS

En este momento si cuentas con un emulador o teléfono inteligente con internet puedes observar el mapa que apunta a Sydney Australia, que es la marca que tiene de defecto el asistente

Materiales:

Equipo de cómputo de alto desempeño core i7 de Intel con 8GB en RAM, IDE Android Studio 3.5 y JDK 1.8

Desarrollo:

Práctica 1a.

Ahora haremos una modificación al código en la actividad principal MapsActivity de Java como se muestra

```
package com.example.dell.mapaicono;

import android.Manifest;
import android.content.pm.PackageManager;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
import android.widget.Toast;

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.UiSettings;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

public class MapsActivity extends FragmentActivity implements
OnMapReadyCallback {

    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        // Obtain the SupportMapFragment and get notified when the map
        is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment)
        getSupportFragmentManager()
            .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }
}
```

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO
DIRECCIÓN ACADÉMICA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
CUADERNO DE PRÁCTICAS

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

    //AQUÍ OBSERVA QUE ESTA HERRAMIENTA DEL OBJETO GOOGLEMAP PERMITE
    //AGREGAR BOTONES DE ZOOM EN EL MAPA
    UiSettings mapSettings;
    mapSettings = mMap.getUiSettings();
    mapSettings.setZoomControlsEnabled(true);

    LatLng eiffel= new LatLng(48.8583701,2.2922873);
    mMap.addMarker(new
    MarkerOptions().icon(BitmapDescriptorFactory.fromResource(
    R.mipmap.ivan)).anchor(0.0f,1.0f).position(eiffel).title("Francia"));
    mMap.moveCamera(CameraUpdateFactory.newLatLng(eiffel));

    LatLng gym = new LatLng(19.295461,-99.02934);
    mMap.addMarker(new
    MarkerOptions().icon(BitmapDescriptorFactory.fromResource(
    R.mipmap.ivan)).anchor(0.0f,1.0f).position(gym).title("tecamachalco"));
    mMap.moveCamera(CameraUpdateFactory.newLatLng(gym));

    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(gym,20));
}
}
```

Como puede diferenciar entre el código por defecto y este último podemos encontrar las diferencias en el método onMapReady, primeramente observamos la siguiente terna de instrucciones que sirven para colocar los controles de zoom en el mapa

```
UiSettings mapSettings;
mapSettings = mMap.getUiSettings();
mapSettings.setZoomControlsEnabled(true);
```

Posteriormente se habrá dado cuenta que se pueden agregar tantos marcadores como usted desee como se aprecia el siguiente encadenamiento de métodos

```
LatLng eiffel= new LatLng(48.8583701,2.2922873);
mMap.addMarker(new MarkerOptions().icon(BitmapDescriptorFactory.fromResource(
```

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO
DIRECCIÓN ACADÉMICA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
CUADERNO DE PRÁCTICAS

```
R.mipmap.ivan)).anchor(0.0f,1.0f).position(eiffel).title("Francia"));
mMap.moveCamera(CameraUpdateFactory.newLatLng(eiffel));
```

En el código anterior se especifica la latitud y longitud de un lugar en específico, posteriormente se agrega un icono (deberá crearlo en mipmap de los recursos en la opción image assets), un título y el efecto de cámara para dirigir a la marca.

Finalmente la instrucción:

```
mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(gym,20));
```

Coloca un zoom en la referencia marcada, solamente aplica a una marca.

Nota final, por default se manifestó el uso de la localización fina en el “manifest”, téngalo en cuenta ya que si no usa el asistente lo deberá de habilitar usted mismo

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Práctica 1b

Ahora combinará sus conocimientos de anteriores como por ejemplo SQLite entre otros de esta asignatura con los de mapa, deberá quizás usar todos sus conocimientos de programación para realizar un mapa que habilita el GPS y guarda en una base de datos SQLite la latitud, longitud y obtiene la calle de dónde ha estado el usuario.

Primero deberá crear un proyecto con una actividad de mapa como lo realizó en la práctica 1a y también generar la llave que deberá agregar en el archivo google_mpas_api.xml como estudio en la metodología. Posteriormente habilite los siguientes recursos en el manifest

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"></uses-permission>
```

Crea el siguiente diseño en el layout principal de los recursos “res”

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

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```


TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO
DIRECCIÓN ACADÉMICA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
CUADERNO DE PRÁCTICAS

```
style="@style/Widget.AppCompat.Button.ButtonBar.AlertDialog"
    android:text="Localizaciones"
    android:id="@+id/bLocalizacion"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:id="@+id/tvLocalizacion"/>

</LinearLayout>

<fragment
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:map="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MapsActivity" />
```

Como observa se pretende agregar un mapa con un botón y un etiqueta TextView

Ahora, agrega una clase llamada `SqlLocalizacion` en donde crearas una base de datos con una tabla llamada ubicaciones y cuatro campos: id, latitud, longitud y calle, como se muestra en el siguiente código

```
package com.example.dell.mapareto;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class SqlLocalizacion extends SQLiteOpenHelper {
    public String tabla= ("CREATE TABLE ubicaciones (id integer
primary key autoincrement,calle text,latitud real,longitud real)");

    public SqlLocalizacion(Context contexto, String name,
SQLiteDatabase.CursorFactory factory, int version) {
        super(contexto, "Direcciones", null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(tabla);
    }

    @Override
```

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO
DIRECCIÓN ACADÉMICA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
CUADERNO DE PRÁCTICAS

```
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
    db.execSQL("DROP TABLE ubicaciones");
}
}
```

Ahora modifica la clase principal MapsActivity con el siguiente código que deberás explicar en tu reporte

```
package com.example.dell.mapareto;

import android.Manifest;
import android.content.ContentValues;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.sqlite.SQLiteDatabase;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.provider.Settings;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.maps.CameraUpdate;
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.UiSettings;
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 java.io.IOException;
import java.util.List;
import java.util.Locale;

public class MapsActivity extends FragmentActivity implements
OnMapReadyCallback, View.OnClickListener {
    private static int PETICION_PERMISO_LOCALIZACION = 101;
    private GoogleMap mMap;
    Button ver;
```

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO
DIRECCIÓN ACADÉMICA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
CUADERNO DE PRÁCTICAS

```
double lat = 0.0, lng = 0.0;
LatLng coordenadas;
private Marker marcador;
String calle2 = "";
TextView calle;
Button lugares;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maps);
    // Obtain the SupportMapFragment and get notified when the map
    is ready to be used.
    SupportMapFragment mapFragment = (SupportMapFragment)
    getSupportFragmentManager()
        .findFragmentById(R.id.map);
    mapFragment.getMapAsync(this);

    lugares = (Button) findViewById(R.id.bLocalizacion);
    calle = (TextView) findViewById(R.id.tvLocalizacion);
    calle.setText(calle2);
    lugares.setOnClickListener(this);
}

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

    mMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
    UiSettings settings = mMap.getUiSettings();
    settings.setCompassEnabled(true);
    settings.setRotateGesturesEnabled(true);
    settings.setScrollGesturesEnabled(true);
    settings.setZoomControlsEnabled(true);
    settings.setZoomGesturesEnabled(true);

    mMap.setOnMapLongClickListener(new
    GoogleMap.OnMapLongClickListener() {
        @Override
        public void onMapLongClick(LatLng latLng) {
            mMap.addMarker(new
            MarkerOptions().icon(BitmapDescriptorFactory.fromResource(R.mipmap.punto))
                .anchor(0.0f,
                1.0f).position(latLng)).setTitle("" + latLng);
            Toast.makeText(getApplicationContext(), "aquí
            click"+latLng, Toast.LENGTH_SHORT).show();
        }
    });
}
```

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO
DIRECCIÓN ACADÉMICA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
CUADERNO DE PRÁCTICAS

```
    }  
    });  
  
    mMap.setOnMarkerClickListener(new  
GoogleMap.OnMarkerClickListener() {  
    @Override  
    public boolean onMarkerClick(Marker marker) {  
        Toast.makeText(getApplicationContext(), "click "+  
marker.getPosition(), Toast.LENGTH_SHORT).show();  
        return true;  
    }  
    });  
  
    direccionAct();  
}  
  
private void direccionAct() {  
    if (ActivityCompat.checkSelfPermission(this,  
Manifest.permission.ACCESS_FINE_LOCATION) !=  
PackageManager.PERMISSION_GRANTED &&  
        ActivityCompat.checkSelfPermission(this,  
Manifest.permission.ACCESS_COARSE_LOCATION) !=  
PackageManager.PERMISSION_GRANTED) {  
        ActivityCompat.requestPermissions(this, new  
String[]{Manifest.permission.ACCESS_FINE_LOCATION},  
PETICION_PERMISO_LOCALIZACION);  
        return;  
    } else {  
        LocationManager locationManager = (LocationManager)  
getSystemService(Context.LOCATION_SERVICE);  
        Location location =  
locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);  
        updateLocalizacion(location);  
  
locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER  
, 20000, 0, locationListener);  
    }  
  
}  
  
private void Marcador(double lat, double lng) {  
    coordenadas = new LatLng(lat, lng);  
    CameraUpdate ubicacionCam =  
CameraUpdateFactory.newLatLngZoom(coordenadas, 16);  
    if (marcador != null) marcador.remove();  
    marcador = mMap.addMarker(new  
MarkerOptions().position(coordenadas).title("Direccion: " + calle2+  
" (" + coordenadas + ")")  
        .icon(BitmapDescriptorFactory.defaultMarker()));  
    mMap.animateCamera(ubicacionCam);  
}
```

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO
DIRECCIÓN ACADÉMICA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
CUADERNO DE PRÁCTICAS

```
private void updateLocalizacion(Location location) {
    if (location != null) {
        lat = location.getLatitude();
        lng = location.getLongitude();
        Marcador(lat, lng);
        guardar();
    }
}

LocationListener locationListener = new LocationListener() {
    @Override
    public void onLocationChanged(Location location) {
        updateLocalizacion(location);
        setLocation(location);
    }
    @Override
    public void onStatusChanged(String provider, int status, Bundle
extras) {}

    @Override
    public void onProviderEnabled(String provider) {
        Toast.makeText(getApplicationContext(), "GPS
ACTIVADO", Toast.LENGTH_SHORT).show();
    }
    @Override
    public void onProviderDisabled(String provider) {
        Toast.makeText(getApplicationContext(), "GPS
DESACTIVADO", Toast.LENGTH_SHORT).show();
        locationStart();
    }
};

public void guardar(){
    SqlLocalizacion lugar = new
SqlLocalizacion(this, "ubicaciones", null, 1);
    SQLiteDatabase db = lugar.getWritableDatabase();
    String direccionCom = calle2;
    Double latit = coordenadas.latitude;
    Double longit = coordenadas.longitude;
    ContentValues valores = new ContentValues();
    valores.put("calle", direccionCom);
    valores.put("latitud", latit);
    valores.put("longitud", longit);
    db.insert("ubicaciones", null, valores);
    db.close();
}

public void setLocation(Location location) {
    if (location.getLatitude() != 0.0 && location.getLongitude() !=
0.0) {
        try {
            Geocoder geocoder = new Geocoder(this,
Locale.getDefault());
```

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO
DIRECCIÓN ACADÉMICA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
CUADERNO DE PRÁCTICAS

```

        List<Address> list =
geocoder.getFromLocation(location.getLatitude(),
location.getLongitude(), 1);
        if (!list.isEmpty()) {
            Address DirCalle = list.get(0);
            calle2 = (DirCalle.getAddressLine(0));
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}

private void locationStart() {
    LocationManager locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
    final boolean gpsEnabled =
locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);
    if (!gpsEnabled) {
        Intent settingsIntent = new
Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
        startActivity(settingsIntent);
    }
}

@Override
public void onClick(View v) {
    switch (v.getId()){
        case R.id.bLocalizacion:
            Intent verUbicacion = new
Intent(MapsActivity.this, Ubicacion.class);
            startActivity(verUbicacion);
            break;
    }
}
}

```

Como habrás observado esta aplicación tiene diferentes métodos es necesario realizar un esquema de la secuencia de los métodos esto denotará que has entendió muy bien el código. Ahora es momento de crear la siguiente actividad “Ubicación” como te habrás dado cuenta al presionar el botón no llevará a la siguiente actividad en donde consultamos a la base de datos de los lugares de donde has estado con tu GPS, mostrando la latitud, longitud y calle.

```

package com.example.dell.mapareto;

import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

```

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO
DIRECCIÓN ACADÉMICA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
CUADERNO DE PRÁCTICAS

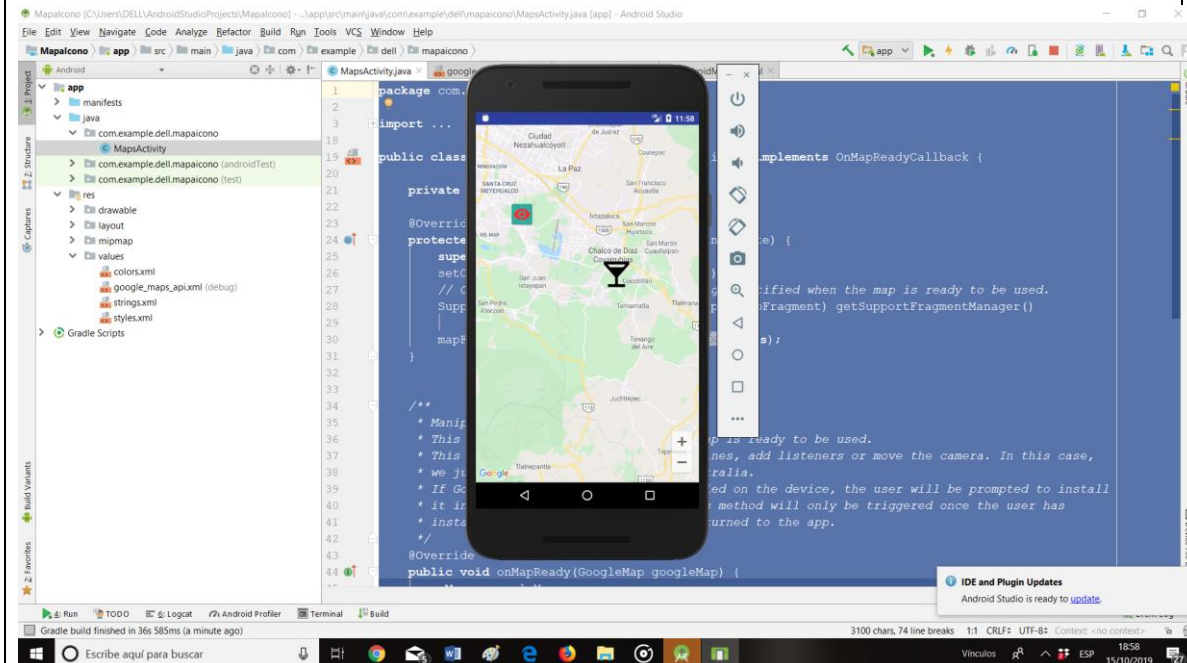
```
public class Ubicacion extends AppCompatActivity {
    TextView vRegistros;
    Cursor c;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ubicacion2);
        vRegistros = (TextView) findViewById(R.id.tvUbicacion);

        SqlLocalizacion lugar = new
        SqlLocalizacion(this, "ubicaciones", null, 1);
        SQLiteDatabase db = lugar.getWritableDatabase();
        c = db.rawQuery(" SELECT * FROM ubicaciones ", null);

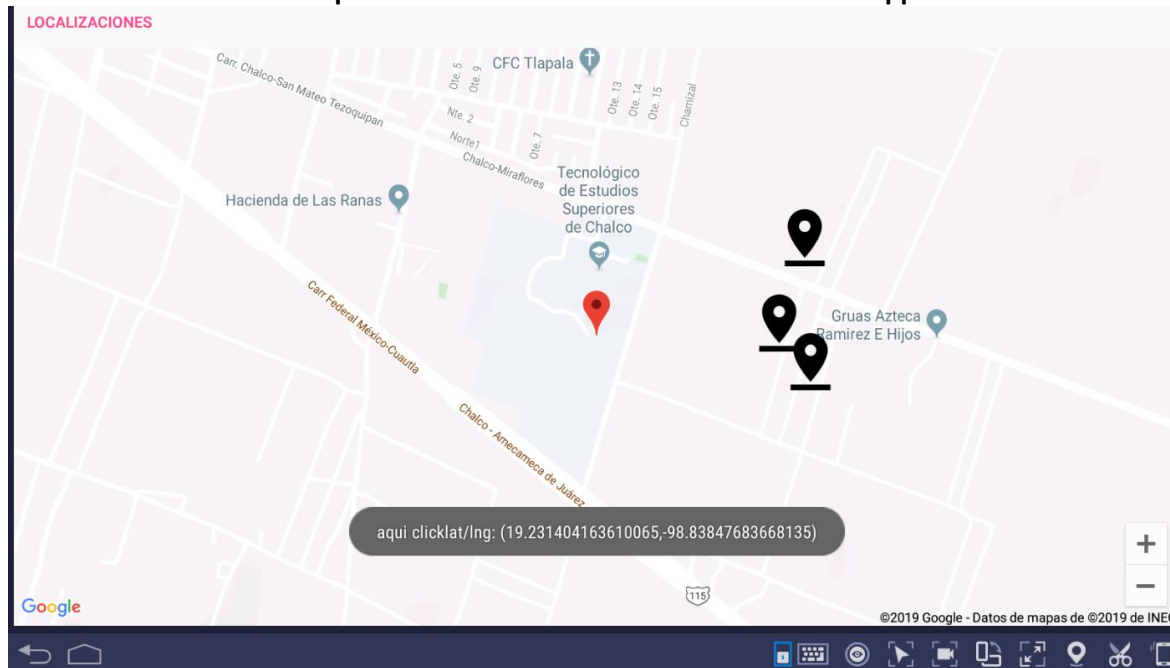
        if (c.moveToFirst()) {
            do {
                Integer id= c.getInt(0);
                String calle = c.getString(1);
                Double latitud = c.getDouble(2);
                Double longitud = c.getDouble(3);
                vRegistros.append("ID" + id + "\n"+"Direccion:" + calle
+ " \n" + "Latitud"+ latitud + "\n"+"Longitud:" + longitud + "\n\n\n");
            } while(c.moveToNext());
        }
        db.close();
    }
}
```

Resultados:
Práctica 1a



TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO
DIRECCIÓN ACADÉMICA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
CUADERNO DE PRÁCTICAS

Practica 1b. Debes en tu reporte mostrar todo el funcionamiento de la app



Conclusiones:

Para crear un mapa en Android, la primera recomendación que puedo realizar como instructor especialista, es tener nuestro Android Studio actualizado, tener experiencia usando el IDE y tener en todo momento internet.

Otra consideración, es en el SDK, agrega la herramienta Google Play Services, ya que esta es necesaria para adquirir los servicios de los servidores de Google, mismo que en los mapas son usados.

Debes construir prácticas, pequeñas o pequeños módulos de esta práctica, nunca codifiques una práctica completa de inicio a fin, será difícil encontrar posibles errores de codificación, más sin embargo si codificas por funcionalidad y de forma incremental no sólo lograr concluir la prácticas de forma satisfactoria, además aprenderás de forma significativa. Otra ventaja es que si existe un código “deprecated” desactualizado, podrás consultar como se escribe la sintaxis de este porque los tienes perfectamente identificado.

Referencias bibliográficas:

Trabajos citados

- [1] Developers, (s.f.), Fragmentos, recuperado de <https://developer.android.com/guide/components/fragments.html>
- [2] sgoliver, (2013), interfaz de usuario en Android: fragments, recuperado de <http://www.sgoliver.net/blog/fragments-en-android/>

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO
DIRECCIÓN ACADÉMICA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
CUADERNO DE PRÁCTICAS

- [3] RamònInVarato, (2013), Fragments (fragmentos) en Android, recuperado de <https://jarroba.com/fragments-fragmentos-en-android/>
- [4] Hermosa programación, (2014), Android:Action bar, recuperado de <http://www.hermosaprogramacion.com/2014/09/android-action-bar/>
- [5] Android Developers, (s.f.), Cuadros de dialogo, recuperado de <https://developer.android.com/guide/topics/ui/dialogs.html?hl=es-419#DialogFragment>
- [6] Hermosa Programación, (s.f.), Como crear diálogos en Android recuperado de <http://www.hermosaprogramacion.com/2015/06/como-crear-dialogos-en-android/>
- [7] Dany Salas, (2011), la historia y comienzos de Android, el sistema operativo de google, recuperado de <https://elandroidelibre.com>.

FECHA DE ELABORACIÓN: 01-09-2019