

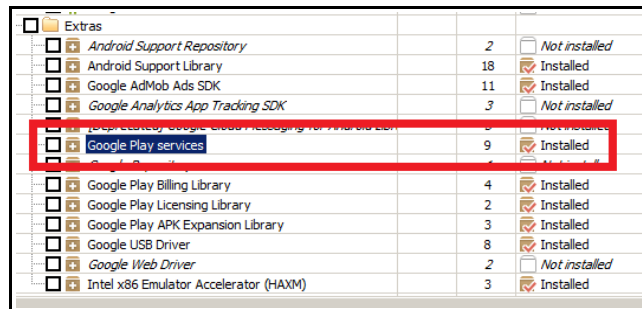
## Google Maps

Para usar la biblioteca de mapas de Google (Google Maps Android API v2) deberemos realizar los siguientes pasos:

- 1) Bajar y configurar “Google Play services SDK”
- 2) Obtener una clave para poder usar los mapas en nuestra aplicación (API KEY)
- 3) Configurar el Manifest.xml

### 1 - Instalando Google Play services SDK

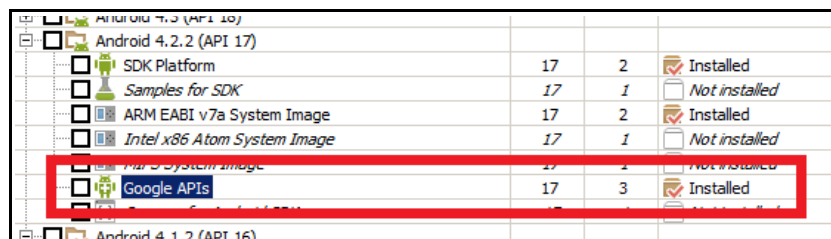
En el SDK manager, deberemos instalar el ítem “Google Play services”:



Estas herramientas se instalarán en **android-sdk/extras/google/google\_play\_services/**

**NOTA:** Los mapas solo podrán ser utilizados en un dispositivo con Android 2.3 o superior y que incluya Google Play Store.

También deberemos instalar, en el SDK manager, las versiones de las APIs de Android de Google:



Esto nos permitirá crear un emulador en donde funcionan los mapas.

**NOTA:** Para ejecutar los mapas en un emulador, éste debe poseer una version de Android 4.2.2 o superior.

### Utilizando Google Play services library

Al instalar Google Play services SDK, se instaló una biblioteca que deberemos incluir en el proyecto de nuestra aplicación Android, en la ruta:

android-sdk/extras/google/google\_play\_services/libproject/google-play-services\_lib/

Para incluir esta biblioteca a nuestro proyecto Android, la agregamos en las dependencias en el archivo build.gradle del módulo app:

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.3'
    compile 'com.google.android.gms:play-services:10.2.0'
}
```

**NOTA:** La versión puede cambiar por una más actual

## 2 - Obteniendo la APY KEY

Cada aplicación posee un certificado, existen dos tipos de certificados:

- **Debug certificate:** Lo genera el Android Studio automáticamente al ejecutar la app.
- **Release certificate:** Se genera al firmar la aplicación para poder subirla en Google Play Store.

Antes de obtener la API -KEY necesitamos obtener el hash SHA1 del certificado de debug/release. Mostraremos como obtener el hash para el certificado de debug (debug.keystore) cuando se firma la aplicación se generará un nuevo archivo de certificado, y deberá utilizarse para obtener un nuevo hash y una nueva API-KEY.

### **Modo Manual:**

abrimos una terminal, y nos posicionamos en `~/.android/` (sobre Linux) o en `C:\Users\your_user_name\.android\` (sobre Windows)

Luego ejecutamos:

Para Linux:

```
keytool -list -v -keystore ~/.android/debug.keystore -alias androiddebugkey
-storepass android -keypass android
```

Para Windows:

```
keytool -list -v -keystore "C:\Users\your_user_name\.android\debug.keystore" -alias
androiddebugkey -storepass android -keypass android
```

Debería aparecer la siguiente información:

```
Alias name: androiddebugkey
Creation date: Jan 01, 2013
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Android Debug, O=Android, C=US
Issuer: CN=Android Debug, O=Android, C=US
Serial number: 4aa9b300
Valid from: Mon Jan 01 08:04:04 UTC 2013 until: Mon Jan 01 18:04:04 PST 2033
Certificate fingerprints:
MD5: AE:9F:95:D0:A6:86:89:BC:A8:70:BA:34:FF:6A:AC:F9
SHA1: BB:0D:AC:74:D3:21:E1:43:07:71:9B:62:90:AF:A1:66:6E:44:5D:75
Signature algorithm name: SHA1withRSA
Version: 3
```

En donde el código SHA1 es el que deberemos obtener, ya que lo utilizaremos a continuación para obtener la API KEY.

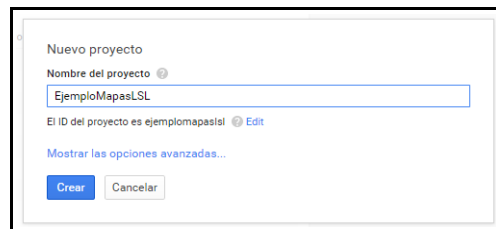
**NOTA:** Si no se reconoce el comando *keytool*, posicionarse en el directorio bin de Java o en "android-studio/jre/bin/"

A continuación, deberemos acceder a la Google API Console:  
<https://console.developers.google.com/project>

En donde nos aparecerá la siguiente página:

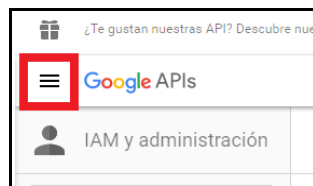


Presionamos Crear proyecto.

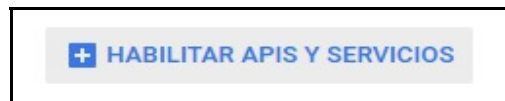


Ponemos como nombre del proyecto “EjemploMapasLSL” y lo creamos.

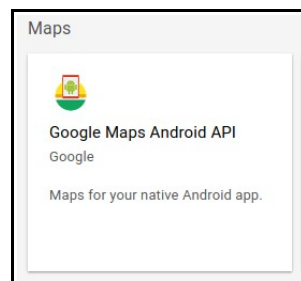
Seleccionamos el boton del menu y luego la opción “APIs y Servicios”



Presionar “Habilitar APIs y Servicios” para activar la API de mapas para este proyecto.

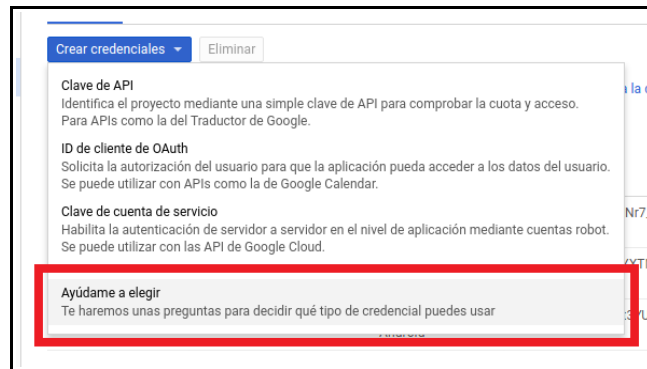


En la sección “Maps” seleccionar “Google Maps Android API”

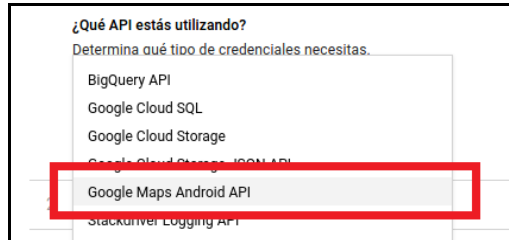




A continuación, en el menú de la izquierda, seleccionamos “Credenciales”, y hacemos click sobre “**Crear credenciales**” y seleccionamos “Ayúdame a elegir”



Seleccionamos “Google Maps Android API”:

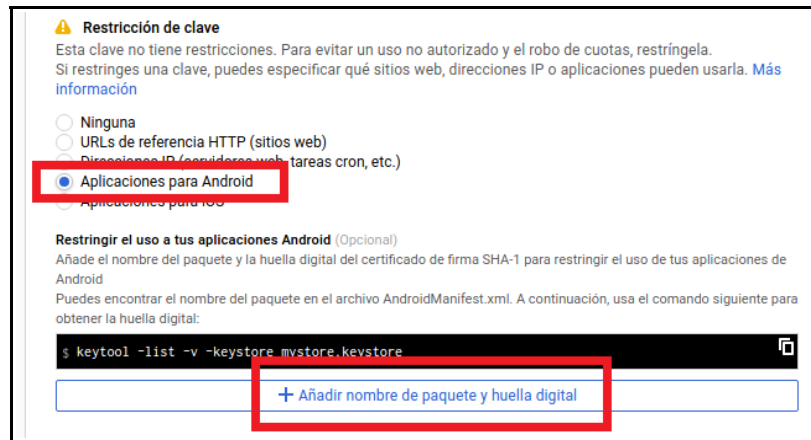


Aparecerá la API KEY, pero debemos restringir su uso solo a apps android.



para ello, seleccionamos “Restringir la clave”.

Luego seleccionamos “Aplicaciones para Android ” y “Añadir nombre de paquete y huella digital”



**Restricción de clave**

Esta clave no tiene restricciones. Para evitar un uso no autorizado y el robo de cuotas, restringela. Si restringes una clave, puedes especificar qué sitios web, direcciones IP o aplicaciones pueden usarla. [Más información](#)

☐ Ninguna  
☐ URLs de referencia HTTP (sitios web)  
☐ Direcciones IP (servidores web, tareas cron, etc.)  
☒ **Aplicaciones para Android**  
☐ Aplicaciones para iOS

**Restringir el uso a tus aplicaciones Android** (Opcional)

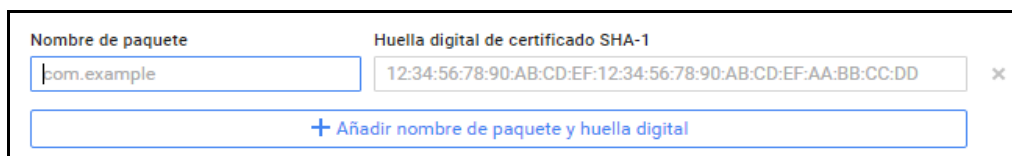
Añade el nombre del paquete y la huella digital del certificado de firma SHA-1 para restringir el uso de tus aplicaciones de Android

Puedes encontrar el nombre del paquete en el archivo AndroidManifest.xml. A continuación, usa el comando siguiente para obtener la huella digital:

```
$ keytool -list -v -keystore mystore.keystore
```

**+ Añadir nombre de paquete y huella digital**

En el espacio que nos aparece para escribir, deberemos poner el código SHA1 que obtuvimos del certificado, junto con el nombre del package de nuestra aplicación



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

com.example 12:34:56:78:90:AB:CD:EF:12:34:56:78:90:AB:CD:EF:AA:BB:CC:DD

**+ Añadir nombre de paquete y huella digital**

*Ejemplo:*

com.utnfra.ejemplomapas

13:EE:8F:D1:A4:87:AE:69:A9:1F:78:DA:98:EA:1A:59:6E:C3:2F:A2

A continuación se mostrará en pantalla la API KEY, que tiene un formato como el siguiente:

***AlzaSyCBajT5wt3YUrAQDtIILMVYipEjz2qH3dQ***

Para más información de cómo obtener la API Key:

<https://developers.google.com/maps/documentation/android/start>

## 3 - Configurar el Manifest de nuestra aplicación

Deberemos agregar los siguientes permisos en el manifest:

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
<!-- The following two permissions are not required to use
      Google Maps Android API v2, but are recommended. -->
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

También deberemos agregar dentro del tag de manifest, lo siguiente:

```
<uses-feature
    android:glEsVersion="0x00020000"
    android:required="true"/>
```

Esto indica que la aplicación utiliza OpenGL para tratar los gráficos de los mapas.

Por último, dentro del tag Application, colocamos la API KEY y la versión de Google Play Services:

```
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="poner api key aqui"/>

<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
```

## Colocando el mapa en pantalla

Modificaremos nuestra Activity, haciendo que herede de "FragmentActivity", ya que los mapas utilizan fragments para representarse (En el caso de usar appcompat y heredar de ActionBarActivity, esto no es necesario):

```
public class MainActivity extends FragmentActivity{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Luego en nuestro archivo xml de layout, colocaremos un tag de "Fragment" en donde el mapa aparecerá.

```
<?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" >

    <fragment
        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:id="@+id/map"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

