

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**PRACTICA DE GOOGLEMAPS**

Curso: Soluciones Móviles I

Docente: Ing. Alberto Johnatan Flor Rodríguez

Alumno:

**RODRIGUEZ MAMANI, Juan Rigoberto**  **(2017057862)**

**Tacna – Perú**

**2020**

**INDICE GENERAL**

[**INTRODUCCIÓN** 3](#_Toc42501291)

[**I.** **OBJETIVOS** 4](#_Toc42501292)

[**1.1. General** 4](#_Toc42501293)

[**1.2. Específicos** 4](#_Toc42501294)

[**II.** **DESARROLLO** 4](#_Toc42501295)

[**2.1. Desarrollo de la aplicación** 4](#_Toc42501296)

[**2.2. Repositorio GitHub** 27](#_Toc42501297)

[**CONCLUSIONES** 28](#_Toc42501298)

[**BIBLIOGRAFÍA** 29](#_Toc42501299)

# **INTRODUCCIÓN**

El Android Studio es la plataforma libre desarrollada por Google, ampliamente utilizada en multitud de dispositivos como móviles, tabletas, TV, wearables e Internet de las cosas. Su expansión ha sido espectacular, siendo el S.O. más utilizado en la actualidad. Tras realizar este curso conocerás los fundamentos del desarrollo de aplicaciones en Android y podrás realizar sencillas aplicaciones, que incluyan los aspectos más importantes y novedosos de esta plataforma.

Android es un sistema operativo que fue creado especialmente para teléfonos con pantalla táctil, los llamados de nueva generación o los inteligentes, las tablets comunes y las que funcionan con líneas telefónica; entrando en esta gama los relojes inteligentes, televisores y algunos aditamentos de los nuevos automóviles, esta empresa de nombre Android Inc, fue respaldada por Google, que en el año 2005 la misma google se apodera comprando por varios millones de dólares, previendo que esta nueva plataforma en la tecnología de avanzada de los estándares en los dispositivos móviles resultaba un buen negocio a futuro.

# **OBJETIVOS**

## **1.1. General**

* Realizar la practica encargada del laboratorio de Googlemaps.

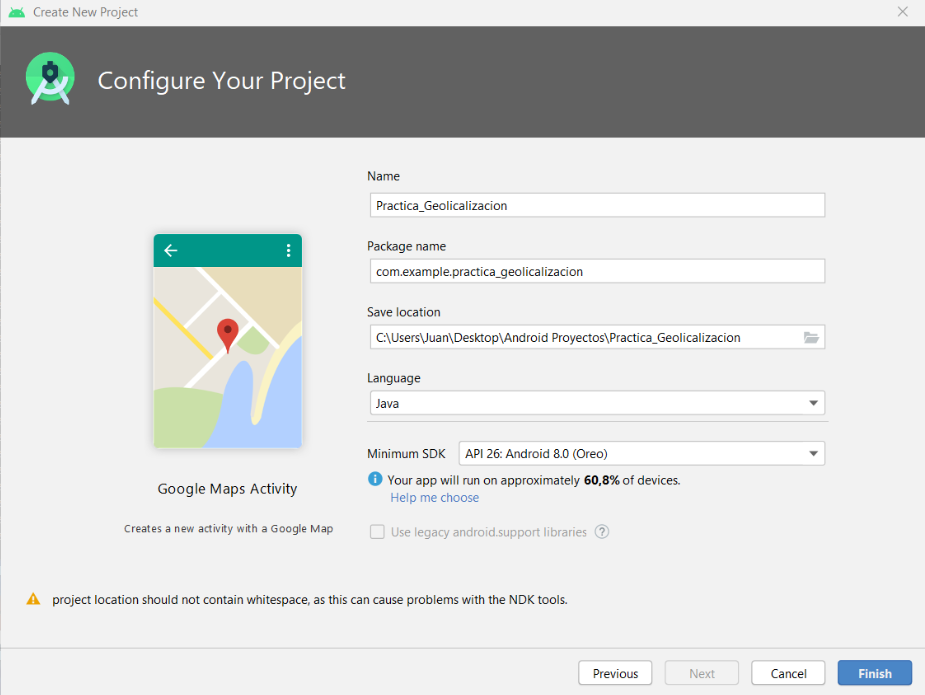
## **1.2. Específicos**

* Investigar la problemática propuesta de la practica encargada.
* Utilizar herramientas de desarrollo en Android Studio.
* Proponer una solución al problema planteado.

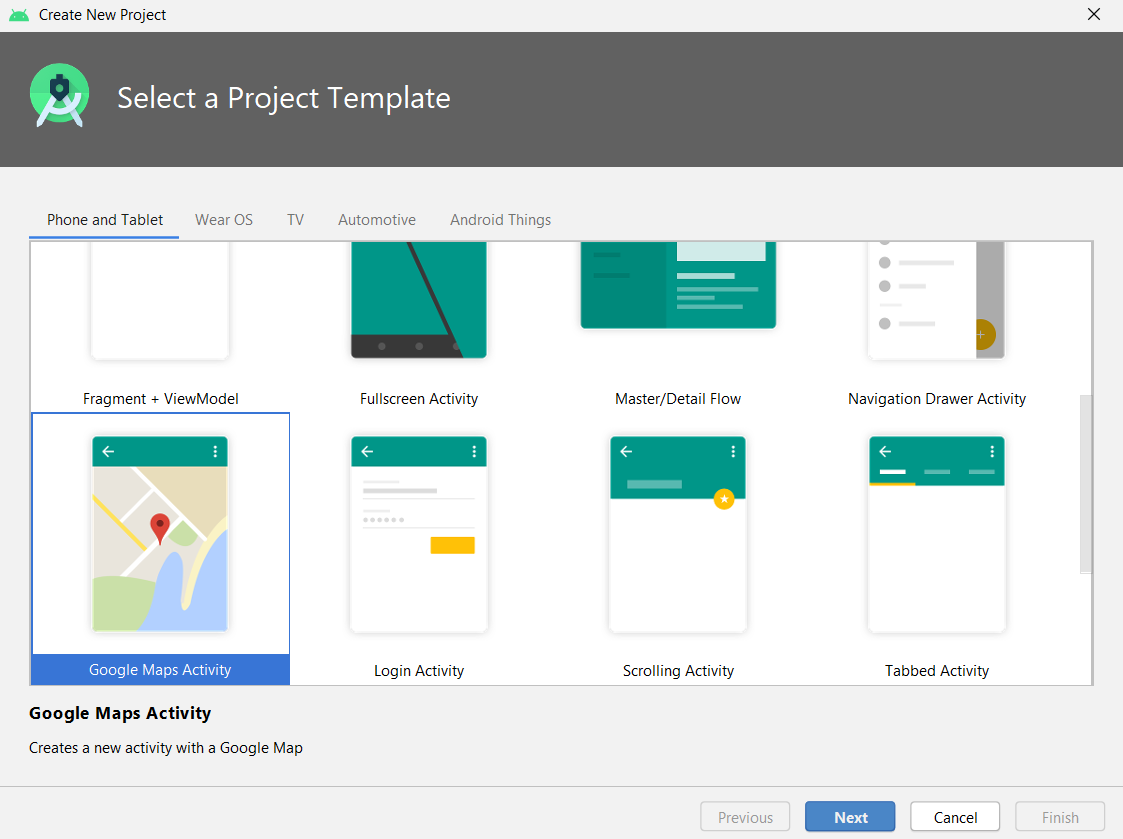
# **DESARROLLO**

## **2.1. Desarrollo de la aplicación**

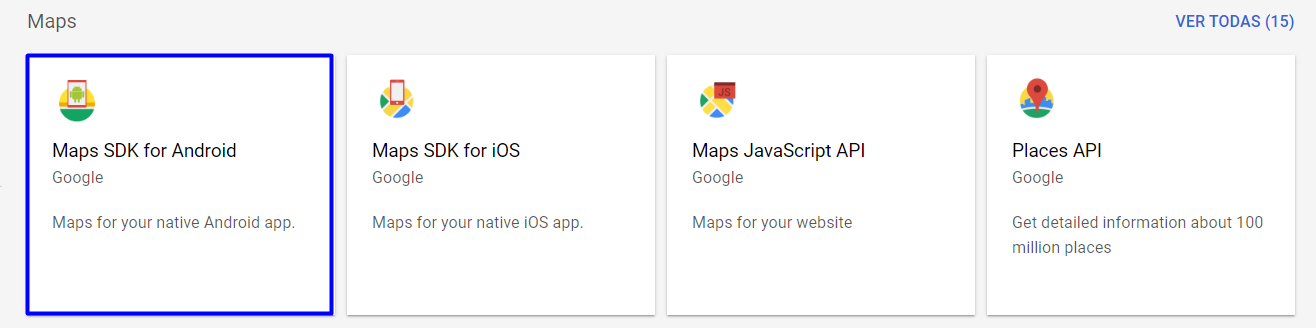
* Lo primero que haremos será crear un nuevo proyecto con el nombre: **Practica\_Geolocalización** luego elegiremos el Minimum SDK la cual será el **API 26.**



* Seguidamente elegiremos el template de **Google Maps Activity.**



* Ahora nos dirigiremos a la página <https://console.cloud.google.com/> en el cual nos registraremos con nuestra cuenta de Google
* Una vez creada la cuenta, debemos habilitar los servicios, para eso nos vamos al Panel y buscamos **Maps SDK for Android** y lo habilitamos.

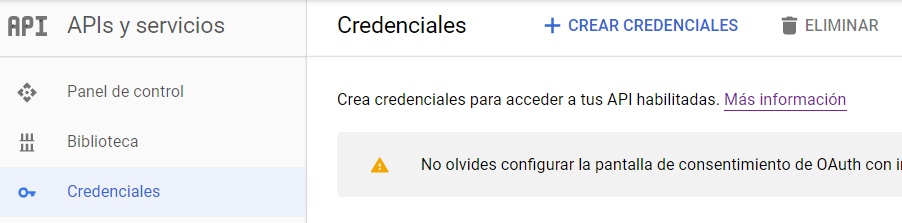




* Luego de haber habilitado nuestro servicio, crearemos una nueva API, para eso nos vamos a Panel de control y elegimos Habilitar **APIS y Servicios.**



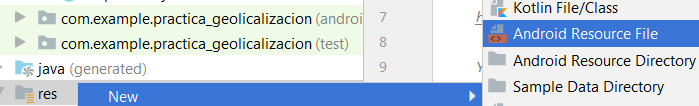
* Después nos vamos a credenciales y crearemos una nueva credencial. Una vez hecho esto ingresaremos a la credencial creada y esta nos darán varios datos como nuestro KEY en el nosotros tenemos que copiar y pegarlo a nuestro proyecto.



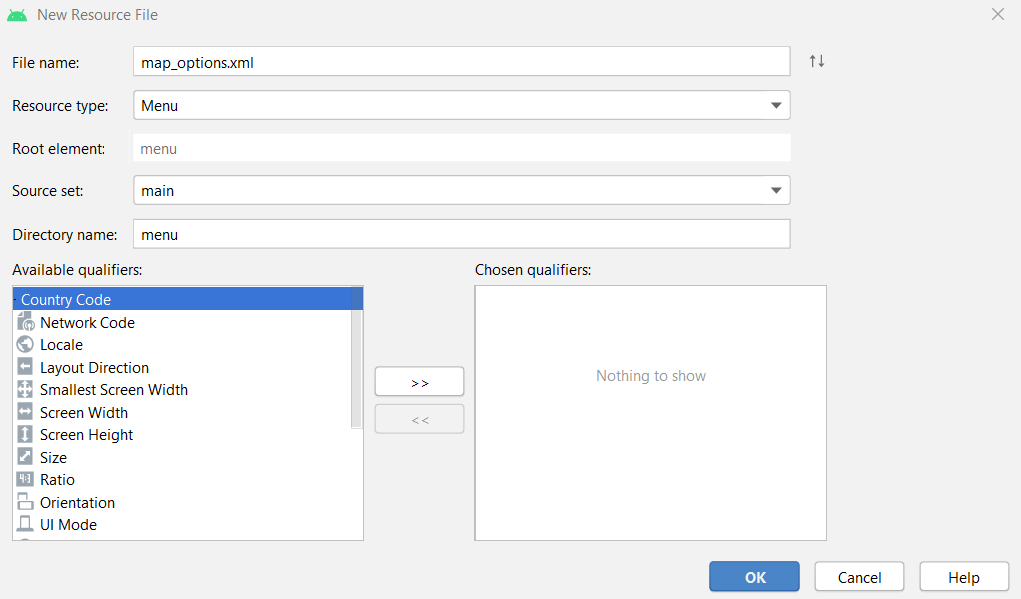
* Seguidamente nos dirigiremos a nuestro proyecto al apartado de **values**, **google\_maps\_api.xml** y pegamos nuestra key generada en el siguiente apartado.



* Hecho esto ahora pasaremos a desarrollar nuestra aplicación, lo primero que haremos será crear un Menu, para eso nos dirigiremos a la carpeta res, new, **Android rosource file.**



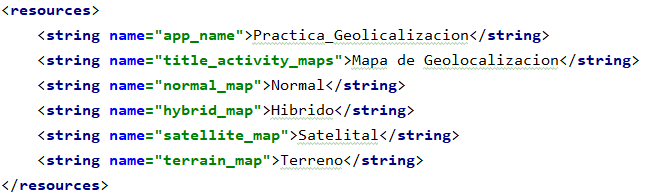
* Le asignaremos el nombre de **map\_options.xml**



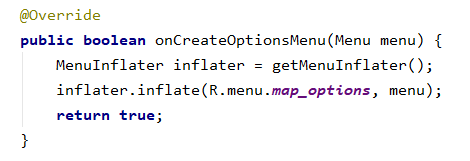
* Dentro de **map\_options.xml** copiaremos el siguiente código, en caso de que nos de algún error presionaremos alt+enter sobre dicho error para solucionarlo.



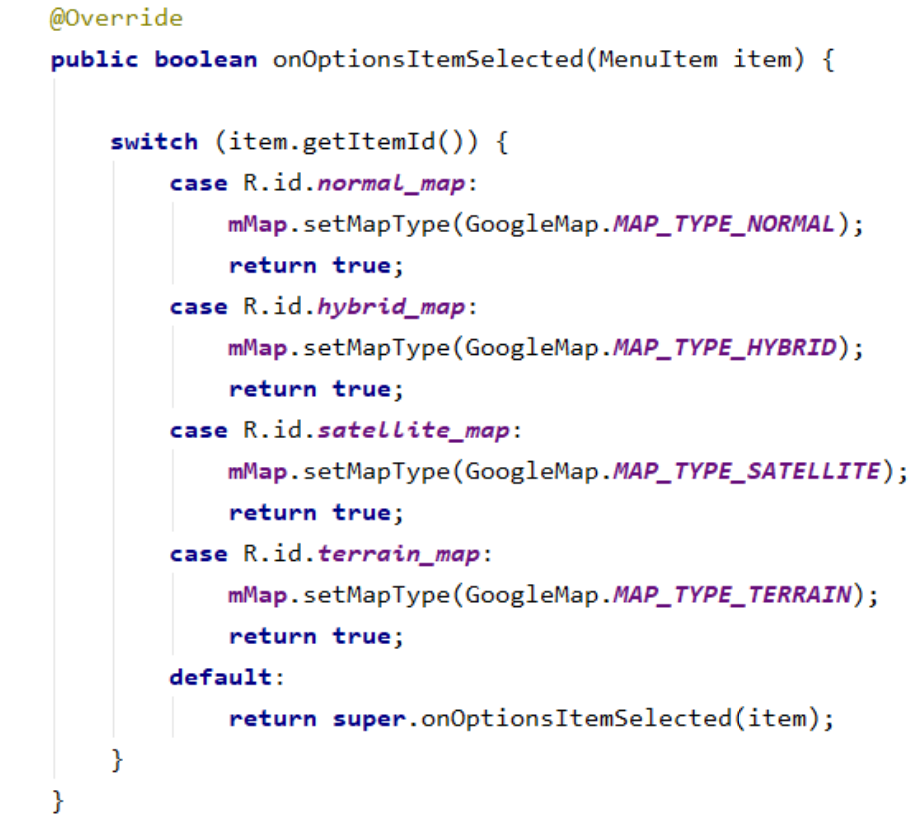
* En nuestro values, **strings.xml** debería tener los siguientes valores.



* Ahora llamaremos el menú a nuestra aplicación, para eso dentro de nuestro MainActivity en el apartado de **onCreateOptionsMenu** copiaremos el siguiente código.



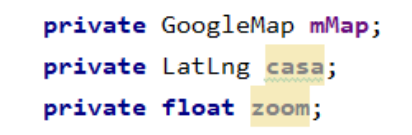
* Agregaremos los ítems del menú.



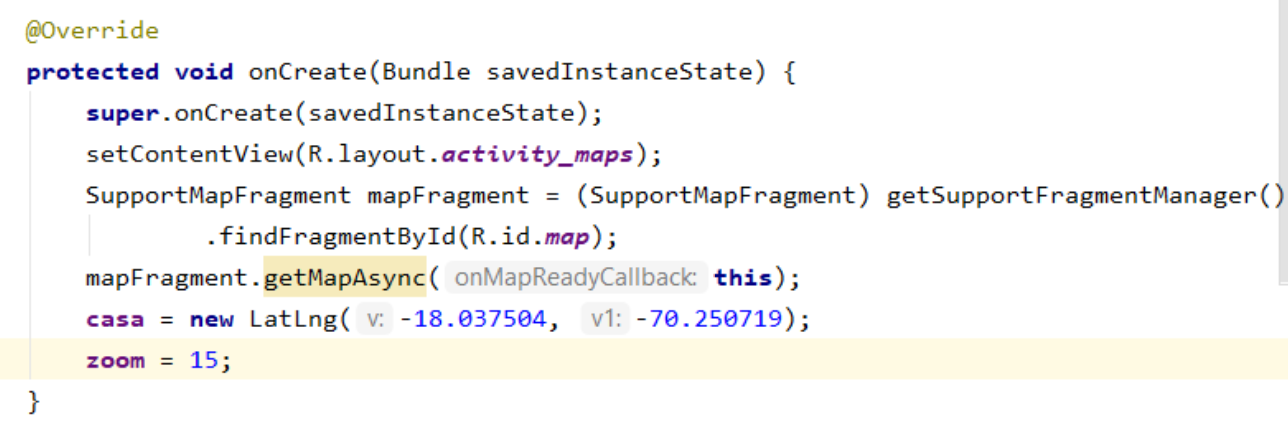
* Si compilamos nos mostrará lo siguiente.



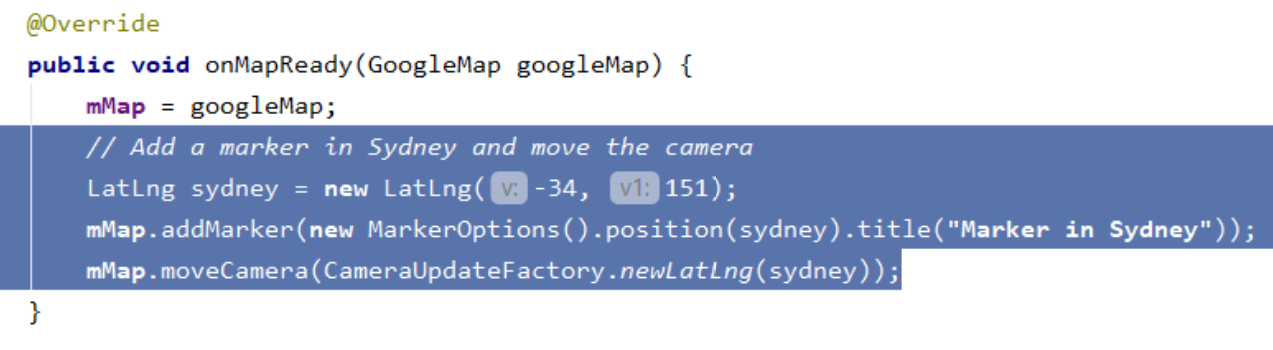
* Ahora vamos a declarar las siguientes variables en nuestro **MainActivity**.



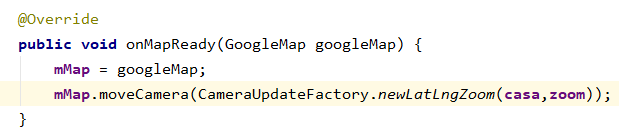
* En nuestro **onCreate** agregaremos lo siguiente:



* En nuestro **onMapReady** modificaremos estas líneas de código.



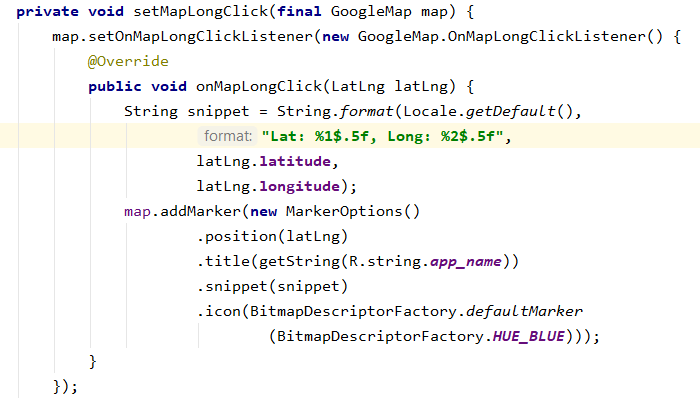
* Y agregaremos estas.



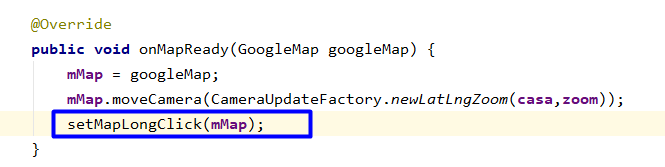
* Si compilamos la aplicación nos mostrará lo siguiente.



* Incorporamos un nuevo método que lo hará será capturar la posición inicial.



* En **onMapReady** agregaremos la siguiente línea.



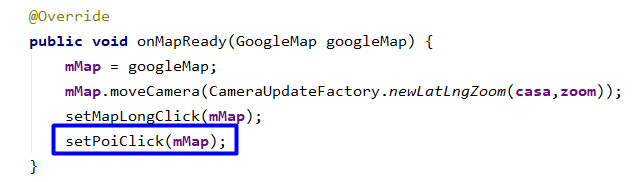
* El resultado es que cuando nosotros hagamos un clic pronunciado en un determinado punto del mapa, este generara un marcador y si hacemos nuevamente clic sobre ese marcador, nos mostrara el nombre de la aplicación como la latitud y longitud.



* Ahora lo que haremos será que cuando hagamos clic en algún determinado lugar del mapa ya no nos muestre el nombre de la aplicación, sino el nombre del sitio de interés que hemos elegido, para eso agregaremos el siguiente método llamado **setPoiClick**.



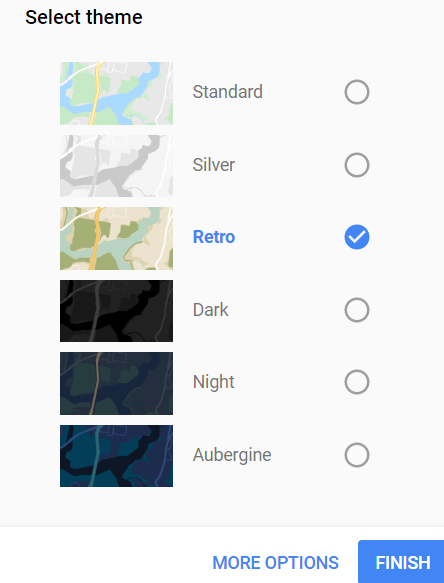
* Ahora para que funcione, lo que haremos será llamarlo en el onMapReady.



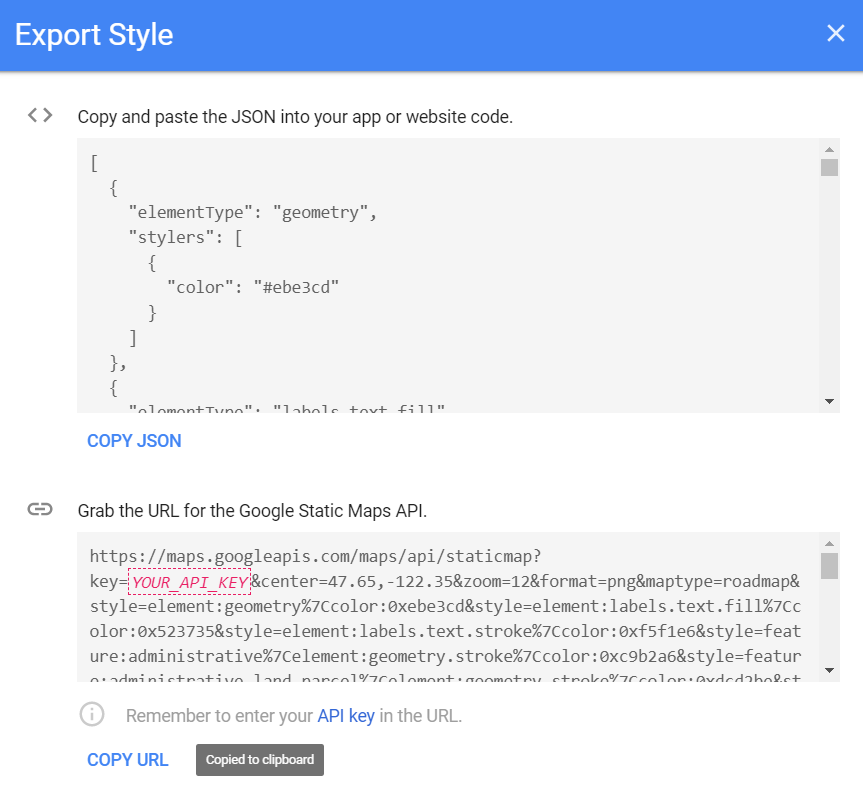
* Ahora cuando demos clic sobre un punto de interés, nos aparecerá otro marcador y mostrará el nombre de dicho lugar.



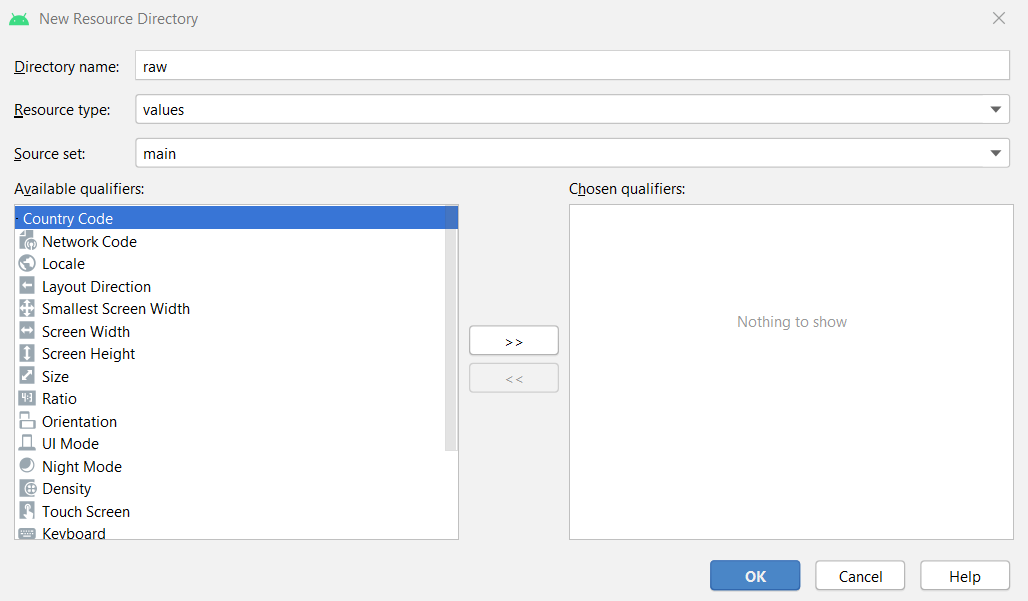
* Ahora le daremos algo de estilo a nuestro mapa, para eso vamos a la página: <https://mapstyle.withgoogle.com> y crearemos nuestro estilo, una vez lo hayamos hecho copiaremos el json generado en dicha web.



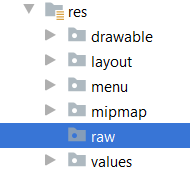
* Copiaremos el código generado en Json.

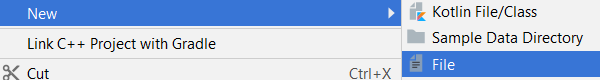


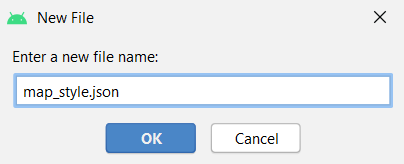
* Ahora nos vamos a **res** y crearemos una carpeta llamado **raw**.



* Dentro de la carpeta raw crearemos un nuevo file llamado **map\_style.json**.



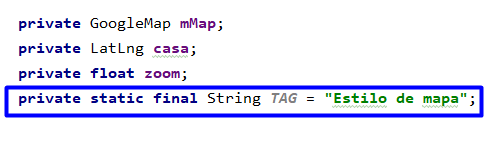




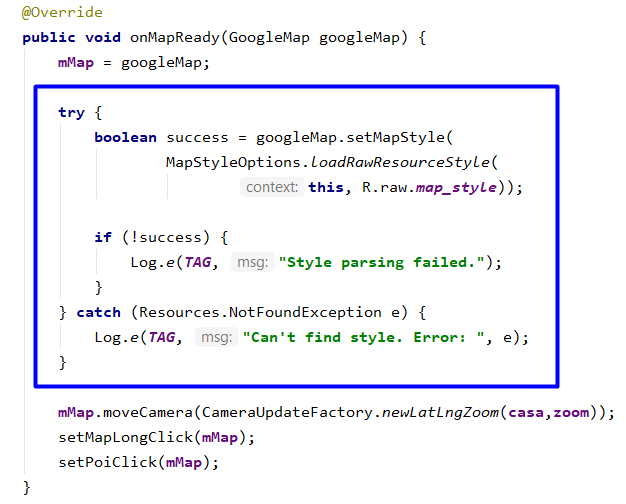
* Y aquí es donde pegaremos todo el código Json generado de nuestro estilo antes creado.



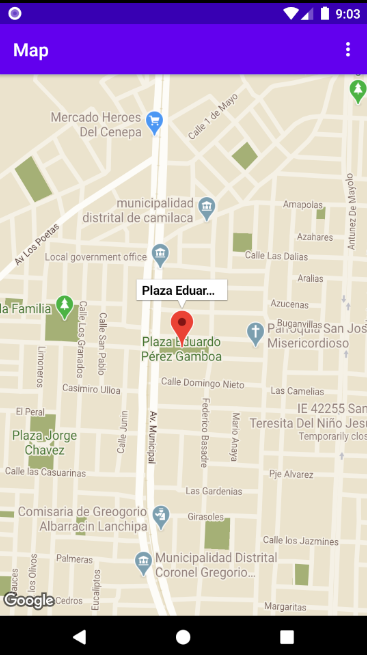
* Luego vamos a agregar una contante llamado TAG a la clase **MapsActivity**.



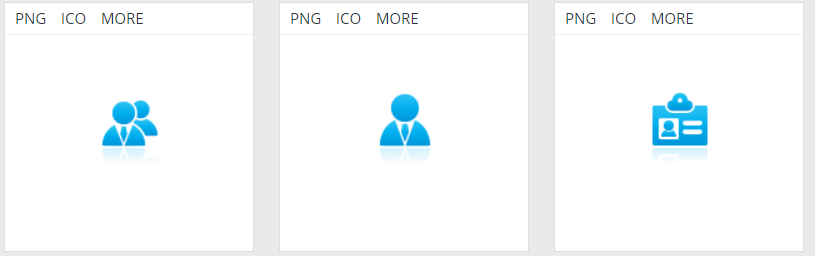
* Ahora vamos a modificar nuestro **onMapReady**, para eso agregaremos las siguientes líneas de código. Esto nos va a permitir entregarle a dicho objeto nuestro fichero de estilo para que este lo adquiera.



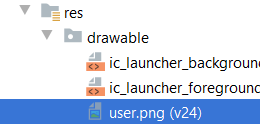
* Ahora lo compilaremos y podremos ver el nuevo estilo creado.



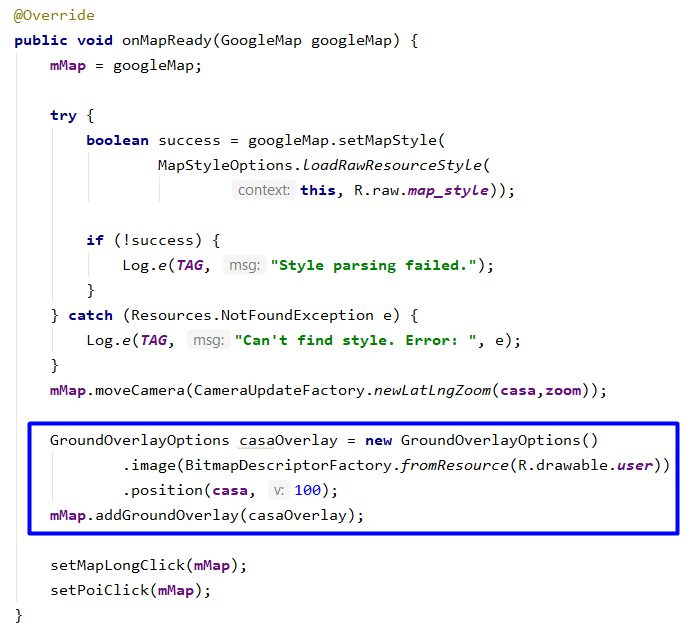
* Acto seguido, vamos a cambiar nuestro icono para eso nos dirigimos a la página <https://www.shareicon.net/> y elegiremos el icono que más nos guste.



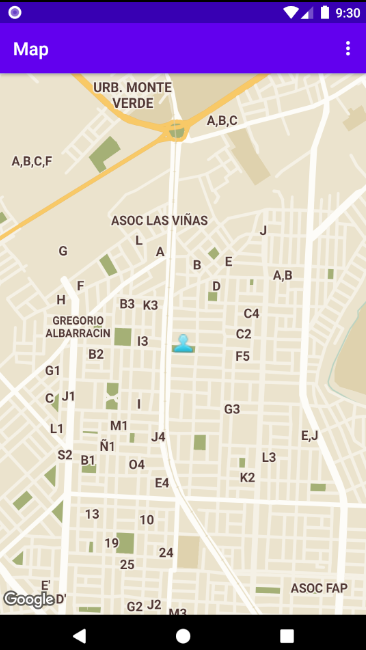
* Cuando tengamos el icono lo copiaremos (ctrl + c) y lo pegaremos (ctrl + v) dentro de la carpeta res, **drawable**.



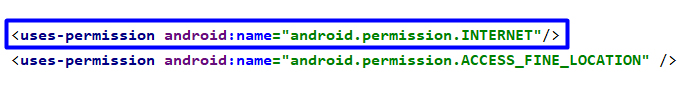
* Ahora pasaremos a agregar las siguientes líneas de código dentro de **onMapReady** donde especificaremos el nombre de nuestro icono, en este caso el del ejemplo se llama **user**.



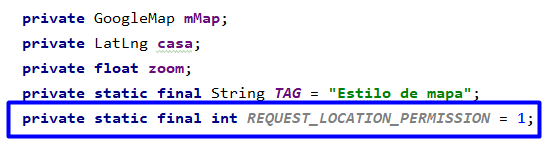
* Ahora podemos compilar nuestra aplicación y podremos ver como nuestro icono de inicio a cambiado.



* Ahora nos dirigiremos a nuestro manifests, **AndroidManifests.xml** y agregaremos el permiso que nos permitirá conectarnos a internet. Esto para que se active el WIFI y hacer la geolocalización utilizándolo como referencia, en caso de no encontrar un punto de referencia.



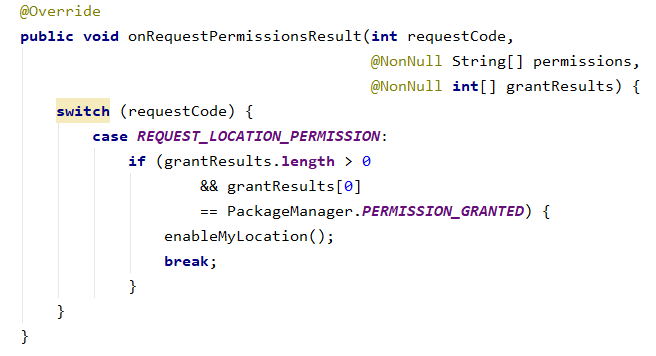
* En la clase **MapsActivity** agregaremos la siguiente constante.



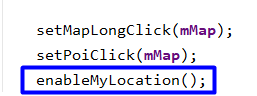
* Ahora crearemos un nuevo método llamado enableMyLocation, lo que hará será habilitar el botón por el cual el mapa va a poder dirigirse hacia donde nos encontramos.



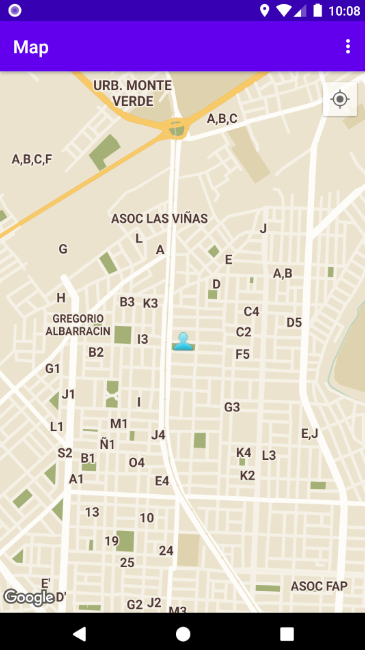
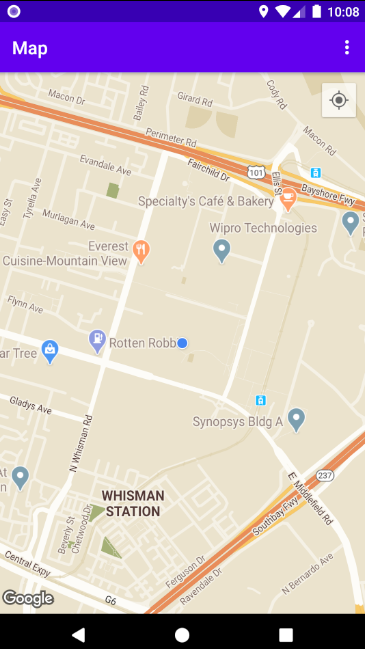
* Además, agregaremos el siguiente código.



* Y no olvidemos llamar a nuestro nuevo método en **onMapReady**.



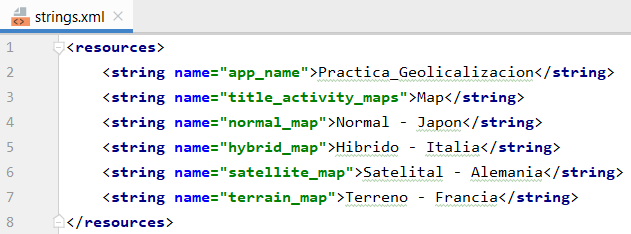
* Ahora pasaremos a compilar la aplicación, cuando arranca la aplicación aparecemos en la dirección por defecto que hemos elegido en un inicio del trabajo, pero si le damos clic al botón de Ubicar nos moverá automáticamente a nuestra “ubicación real”.

* Como acto seguido lo que haremos será crear 4 ubicaciones virtuales las cuales serán Japón, Alemania, Italia y Francia, esto con la intensión de poder mostrar la distancia desde nuestra ubicación hacia esos lugares.

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de Mapa | Icono | Localización | Lati, Long |
| Normal | Mundo | Japón | (35.680513, 139.769051) |
| Híbrido | Montañas | Italia | (41.902609, 12.494847) |
| Satélite | Satélite | Alemania | (52.516934, 13.403190) |
| Terrarian | Planos | Francia | (48.843489, 2.355331) |

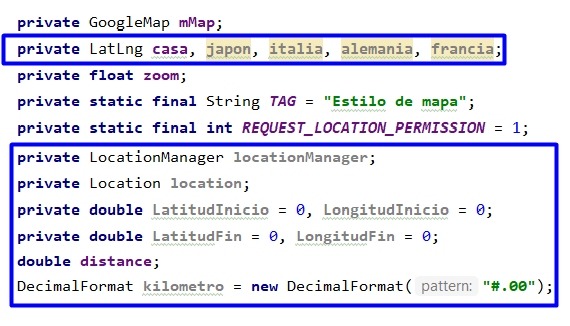
* Para eso primero modificaremos nuestro values, **string.xml** de tal manera para que quede tal como el recuadro.



* Nos sirve de referencia ya que enlazará sitio al que queremos calcular la distancia.



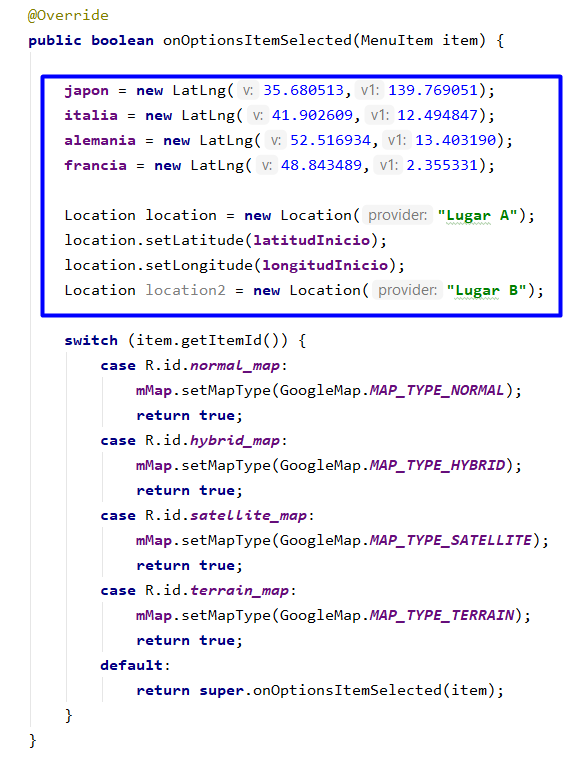
* Acto seguido, agregaremos los países como también la location, la latitud y longitud en nuestro **MapsActivity**.



* En nuestro **onCreate** agregaremos las siguientes líneas de código.



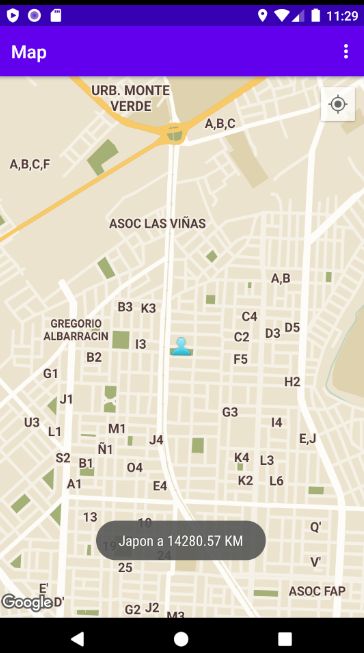
* Luego en nuestro **onOptionItemSelected** agregaremos las coordenadas de dichos países, de tal manera que quede igual al cuadro antes visto.



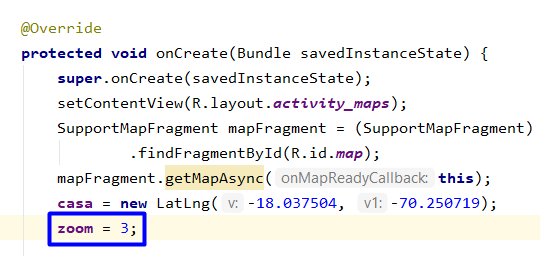
* Ahora pasaremos a configurar nuestro primer menú que mostraba la vista Normal. Para eso agregaremos lo siguiente. **JAPÓN**



* Ahora si compilamos nuestra aplicación veremos que cuando damos clic al menú Japón, este nos muestra un mensaje indicando la distancia de punto a punto representado en Kilómetros.



* Ahora vamos a cambiar la distancia de visión de nuestro mapa, esto lo haremos en onCreate, donde aparecía en 15 lo cambiaremos a 3.



* Esto lo hacemos con la finalidad de poder ver todo el globo y así poder ubicar las coordenadas de los puntos generados, tales como Japón.



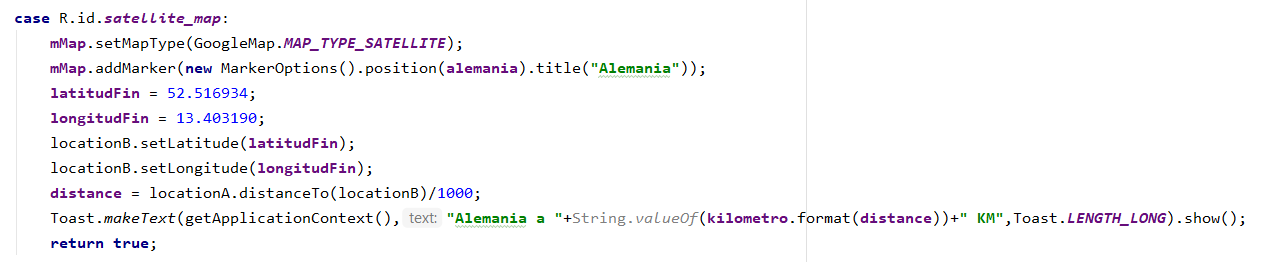
* Ahora como hemos visto que nuestro primer menú funciona correctamente, haremos el mismo procedimiento para los demás menús. **ITALIA**



* Resultado.



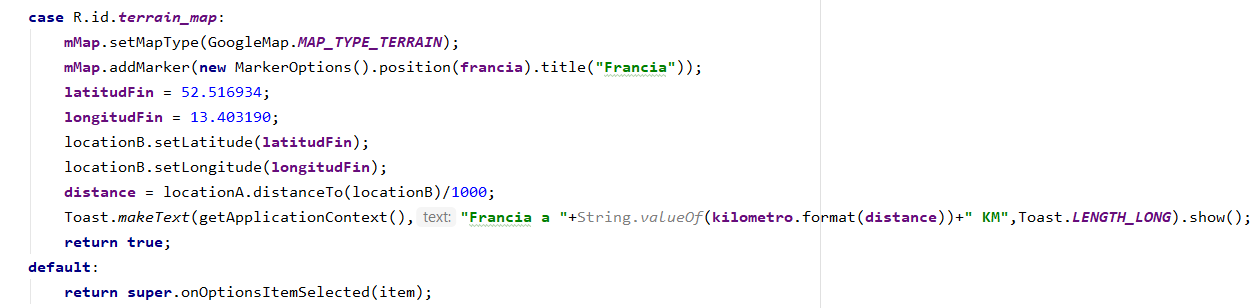
* Repetiremos el proceso, pero esta vez será para **ALEMANIA**.



* Resultado.



* Repetiremos el proceso, pero esta vez será para **FRANCIA**.



* Resultado.



* Y así hemos concluido con nuestra practica perteneciente al tema de Googlemaps.

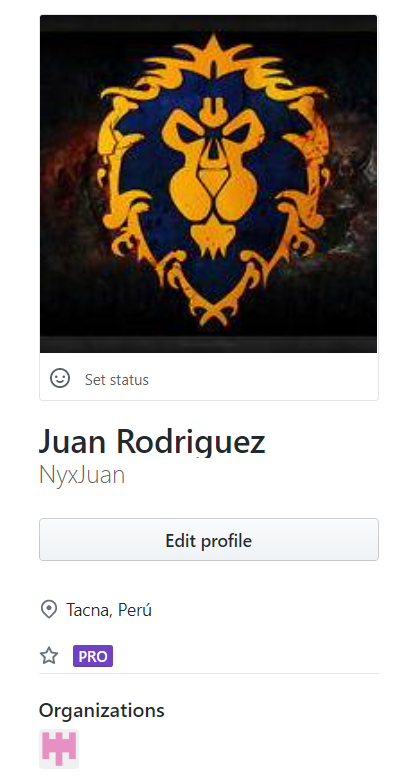
**NOTA:**

* En caso de que falle la aplicación, intentar limpiar el Emulador dándole clic en **WIPE DATA** o intentar probar con otro emulador.
* Otorgar permisos a la aplicación.
* Intentar correr con el emulador utilizando el **API 26** en adelante.

## **2.2. Repositorio GitHub**

**Aplicación Googlemaps:**

[**https://github.com/NyxJuan/Moviles\_I\_Practica\_IV\_Googlemaps**](https://github.com/NyxJuan/Moviles_I_Practica_IV_Googlemaps)



# **CONCLUSIONES**

* Se logró realizar la practica encargada del laboratorio de Googlemaps.
* Se pudo Investigar la problemática propuesta de la practica encargada.
* Se utilizó varias herramientas para el desarrollo de la aplicación en Android Studio.
* Se pudo proponer la solución del problema planteado.

# **BIBLIOGRAFÍA**

1. Gutiérrez, G. (2019). *Android.* Recuperado el 7 de mayo de 2020, de Concepto Definición. Sitio web: <https://conceptodefinicion.de/android/>
2. Google. (2018). *Google Maps Intents for Android.* Recuperado el 7 de junio de 2020, de Android Studio. Sitio web: <https://developers.google.com/maps/documentation/urls/android-intents>
3. Google. (2019). *Google Cloud Platform.* Recuperado el 7 de junio de 2020, de Google. Sitio web: <https://console.cloud.google.com/>
4. MapStyle (2018). *Google Maps Style.* Recuperado el 7 de junio de 2020, de Google. Sitio web: <https://mapstyle.withgoogle.com/>
5. ShareIcon. (2018). *Share Icon.* Recuperado el 8 de junio de 2020, de ShareIcon. Sitio web: <https://www.shareicon.net/>
6. Google. (2018). *Google Maps Platform Documentation.* Recuperado el 8 de junio de 2020, de Google. Sitio web: <https://developers.google.com/maps/documentation?hl=es>
7. Xomena. (2017). *Android, how to pass latitude longitude route info to Google Maps App.* Recuperado el 8 de junio de 2020, de Stackoverflow. Sitio web: <https://stackoverflow.com/questions/42677389/android-how-to-pass-lat-long-route-info-to-google-maps-app>