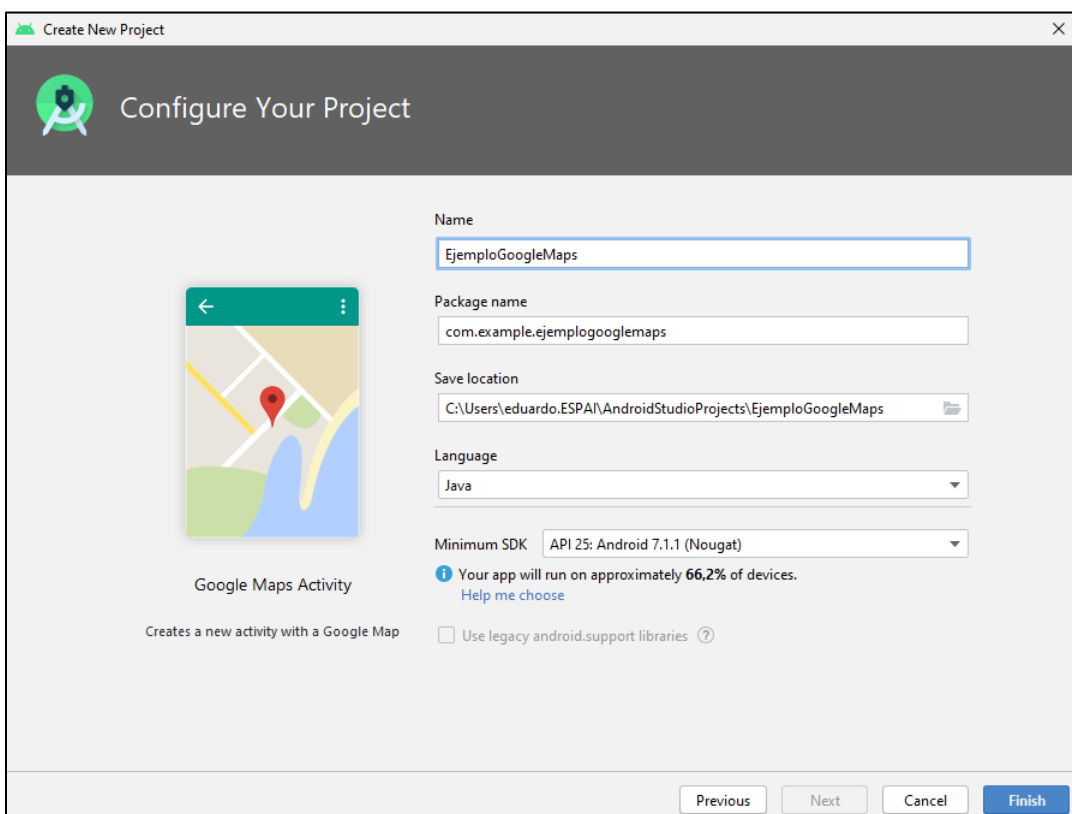
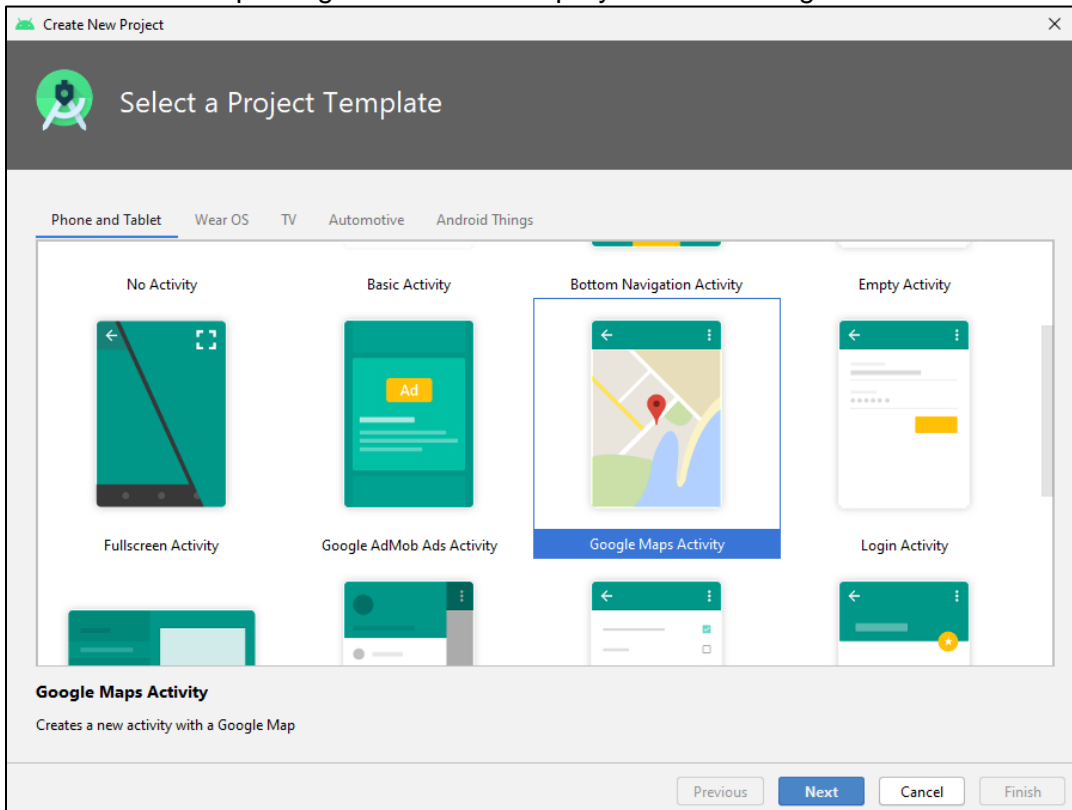


PRACTICA 14: GOOGLMAPS

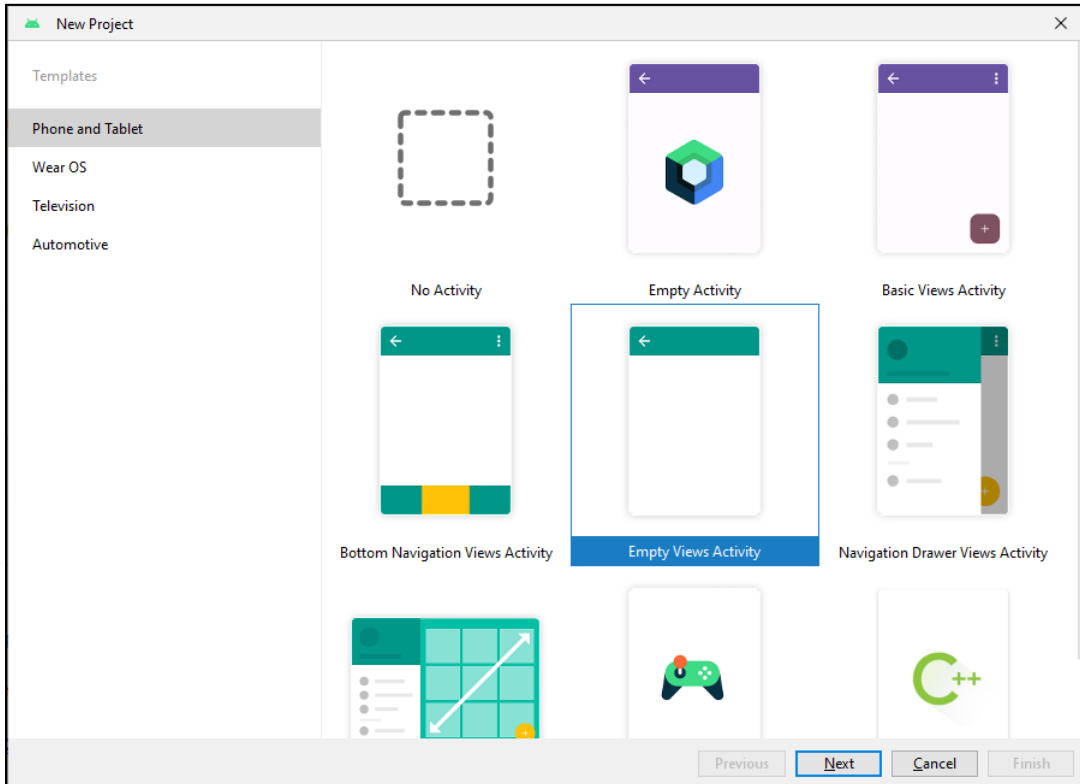
Parte I: Un ejemplo simple con Google Maps

Veamos un sencillo ejemplo que nos permite visualizar un mapa centrado en las coordenadas geográficas detectadas por el sistema de posicionamiento.

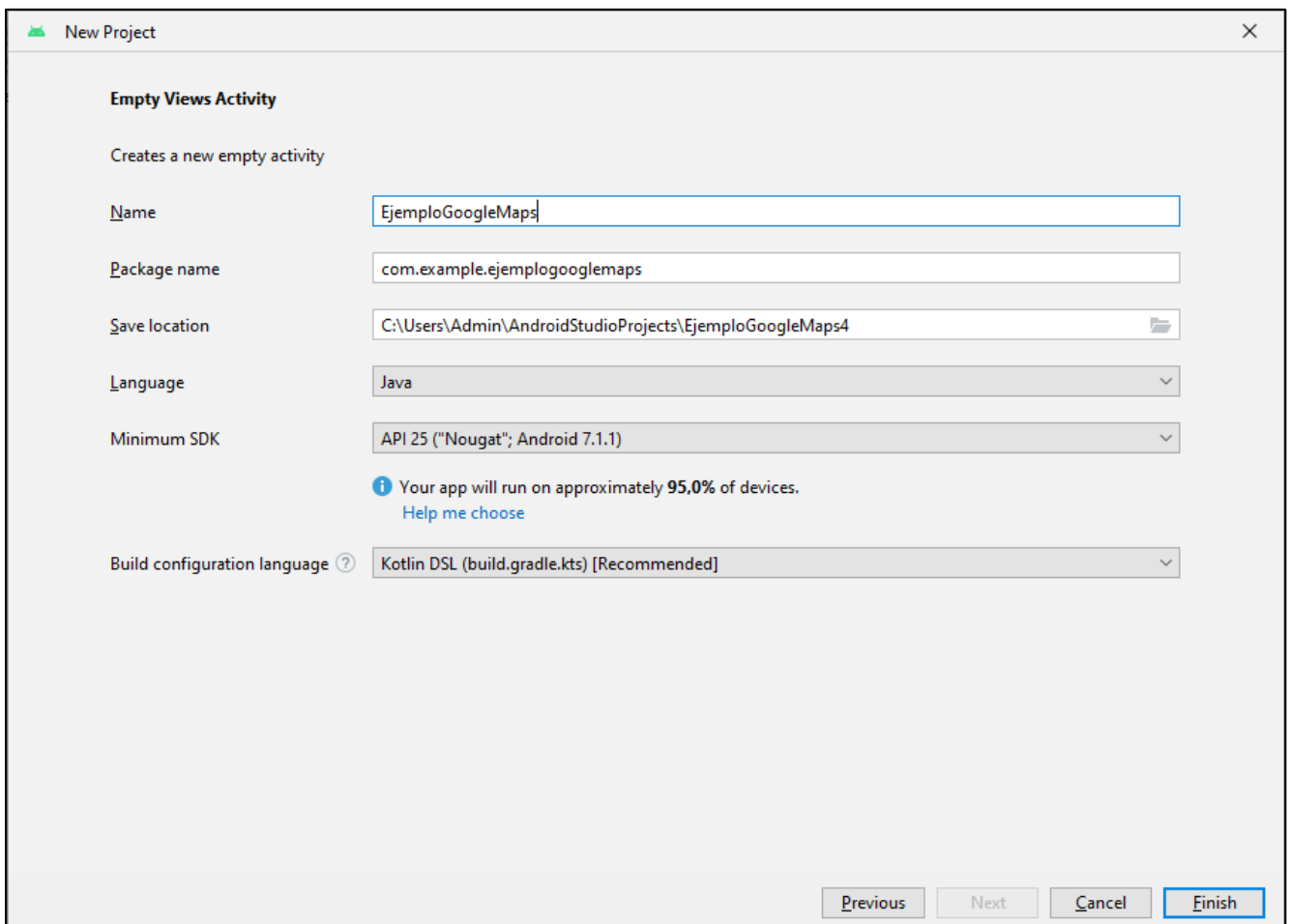
Paso 1. Antes se podía generar un nuevo proyecto con los siguientes datos:



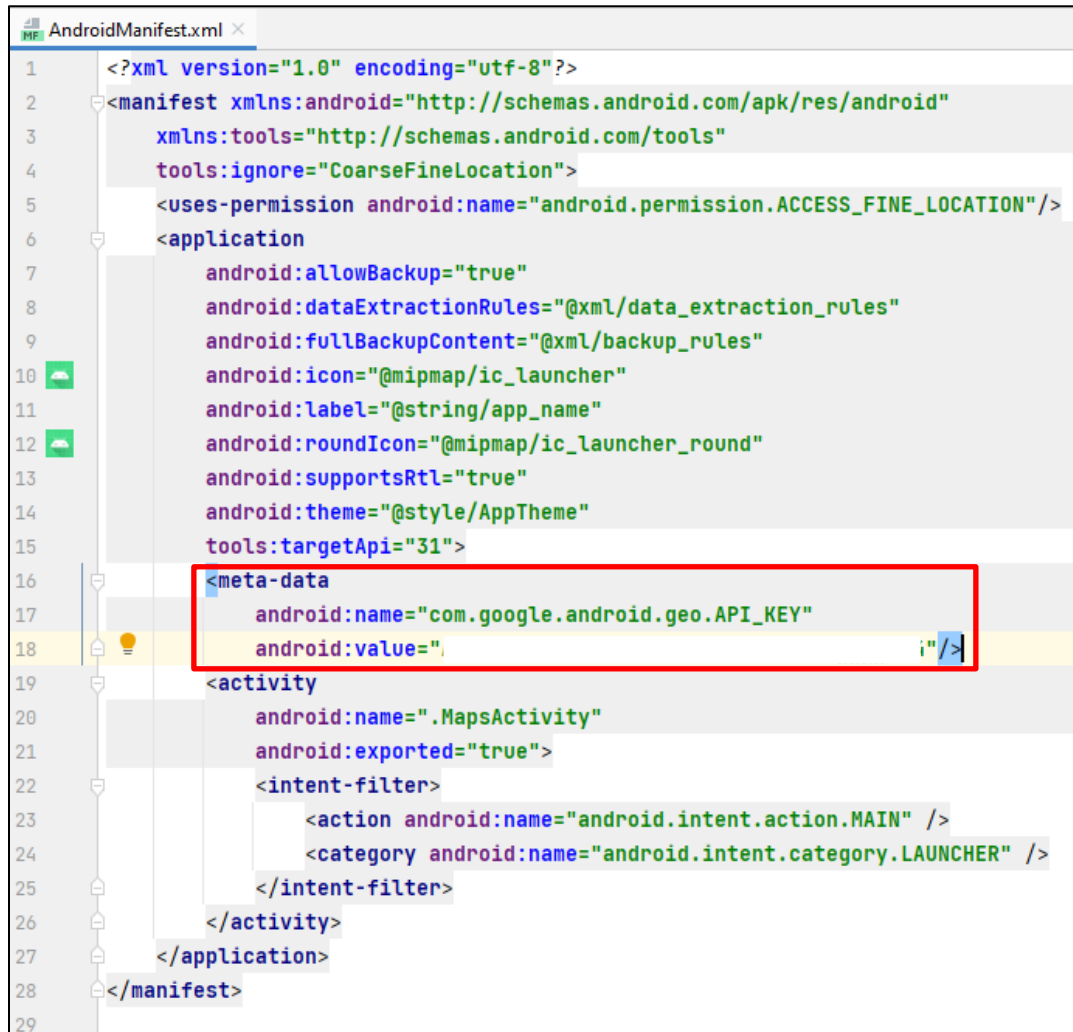
Verás que ya no aparece esta plantilla ya. La generaremos manualmente.



Con el mismo nombre de EjemploGoogleMaps.

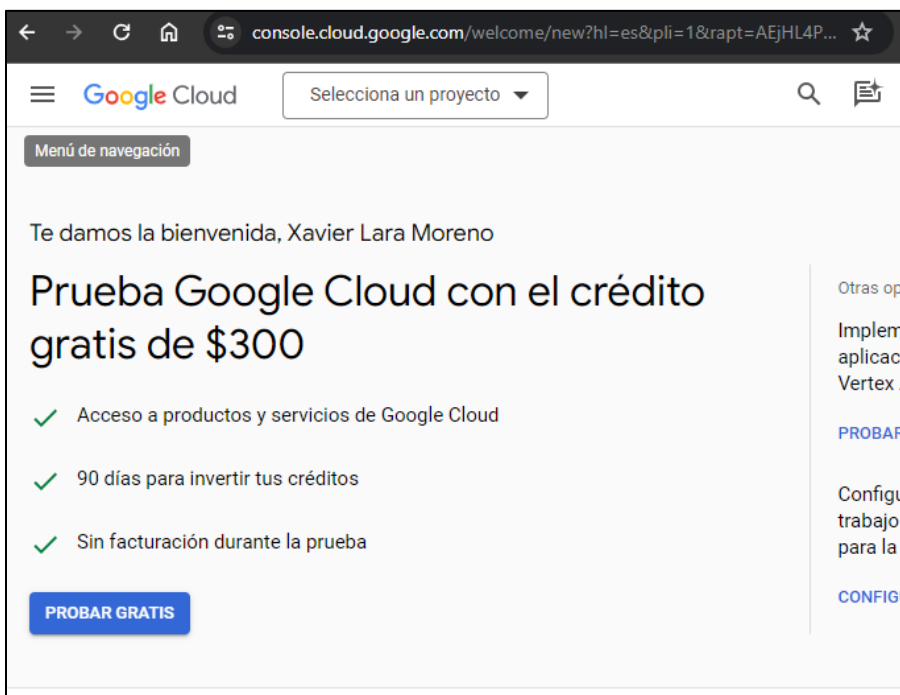


Paso 2. Abre AndroidManifest.xml y añade la pestaña de meta-data antes de la de activity.

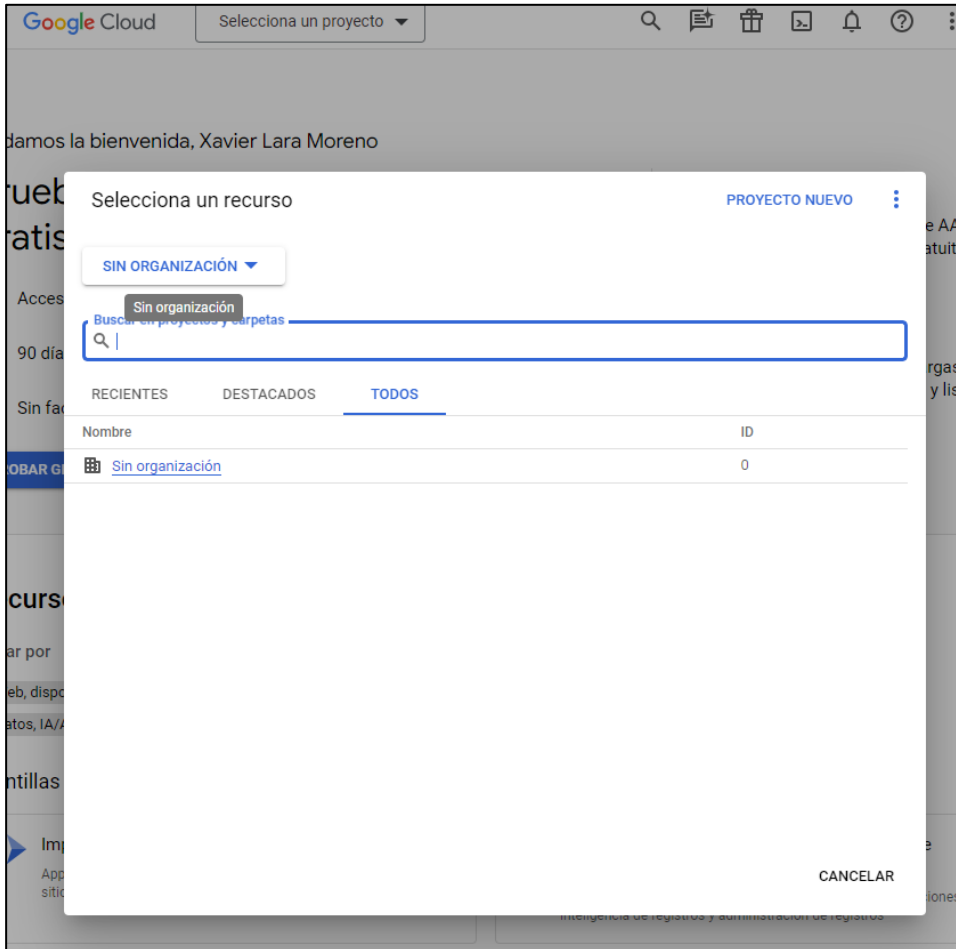


```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   tools:ignore="CoarseFineLocation">
5   <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
6   <application
7     android:allowBackup="true"
8     android:dataExtractionRules="@xml/data_extraction_rules"
9     android:fullBackupContent="@xml/backup_rules"
10    android:icon="@mipmap/ic_launcher"
11    android:label="@string/app_name"
12    android:roundIcon="@mipmap/ic_launcher_round"
13    android:supportsRtl="true"
14    android:theme="@style/AppTheme"
15    tools:targetApi="31">
16     <meta-data
17       android:name="com.google.android.geo.API_KEY"
18       android:value="" />
19   <activity
20     android:name=".MapsActivity"
21     android:exported="true">
22     <intent-filter>
23       <action android:name="android.intent.action.MAIN" />
24       <category android:name="android.intent.category.LAUNCHER" />
25     </intent-filter>
26   </activity>
27 </application>
28 </manifest>
```

Paso 3. Ve a Google Cloud Platform, selecciona un proyecto nuevo. Primero en clicas en Selecciona un Proyecto.



Paso 4. Ahora clicas en PROYECTO NUEVO.



Paso 5. El nombre del proyecto será EjemploGraficosMaps

Google Cloud

Proyecto nuevo

Tienes 12 projects restantes en tu cuota. Solicita un incremento o borra algunos proyectos. [Más información](#)

[MANAGE QUOTAS](#)

Nombre del proyecto *
EjemploGoogleMaps

Project ID: tranquil-leaf-412817. It cannot be changed later. [EDIT](#)

Organización *
iticbcn.cat

Selecciona una organización para vincularla a un proyecto. No podrás cambiar esta selección más adelante.

Ubicación *
iticbcn.cat [EXPLORAR](#)

Organización o carpeta superior

[CREAR](#) [CANCELAR](#)

Paso 6. Ahora ve a Incorpora mapas de Google Maps Platform. Deberéis añadir la API de forma gratuita sin cargos, pero os obligará a dejar los datos de una tarjeta de crédito.

The screenshot shows the Google Cloud console interface. At the top, there's a navigation bar with the Google Cloud logo, a dropdown menu showing 'EjemploGoogleMaps', and a search bar. Below the navigation bar, a welcome message says 'Te damos la bienvenida, Xavier Lara Moreno'. The main heading is 'Prueba Google Cloud con el crédito gratis de \$300'. There are three checkmarks listing benefits: 'Acceso a productos y servicios de Google Cloud', '90 días para invertir tus créditos', and 'Sin facturación durante la prueba'. A 'PROBAR GRATIS' button is visible. To the right, there's a section for 'Otras opciones' with links for 'PROBAR VERTEX AI' and 'CONFIGURACIÓN DE GOOGLE CLOUD'. Below this, the 'Recursos populares para comenzar' section is shown, with filters for 'Web, dispositivos móviles, juegos, almacenamiento', 'Contenedores, VMs, nubes híbridas y múltiples, mover carga de trabajo', 'Datos, IA/AA, SAP', and 'Maps, APIs'. The 'General' filter is selected. Under 'Plantillas de soluciones prediseñadas', there are three cards: 'Implementa una app web de tres niveles', 'Implementa VMs administradas con balanceo de cargas', and 'Crea un almacén de datos con BigQuery'. A 'Ver todas las soluciones' button is below. The 'Productos' section contains a grid of cards: 'Crea una VM Compute Engine', 'Entrena y aloja modelos de AA Vertex AI', 'Crea una base de datos Cloud SQL', 'Analiza y administra datos BigQuery', 'Crea un bucket de almacenamiento Cloud Storage', 'Ejecuta apps alojadas en contenedores Cloud Run', 'Crea una app alojada en contenedores Kubernetes Engine', and 'Incorpora mapas Google Maps Platform'. The 'Incorpora mapas' card is highlighted with a red border.

Google Cloud EjemploGoogleMaps Buscar (/) recursos, documentos, productos y más Buscar 1 ?

de navegación

Te damos la bienvenida, Xavier Lara Moreno

Prueba Google Cloud con el crédito gratis de \$300

- ✓ Acceso a productos y servicios de Google Cloud
- ✓ 90 días para invertir tus créditos
- ✓ Sin facturación durante la prueba

PROBAR GRATIS

Otras opciones

Implementa y entrena modelos de AA y aplicaciones de IA con el nivel gratuito de Vertex AI.

[PROBAR VERTEX AI](#)

Configura Google Cloud en las cargas de trabajo empresariales escalables y listas para la producción.

[CONFIGURACIÓN DE GOOGLE CLOUD](#)

Recursos populares para comenzar

Filtrar por

Web, dispositivos móviles, juegos, almacenamiento Contenedores, VMs, nubes híbridas y múltiples, mover carga de trabajo Datos, IA/AA, SAP Maps, APIs

General

Plantillas de soluciones prediseñadas ?

Implementa una app web de tres niveles

Apps web, sitios de rich media, sitios web de comercio electrónico y sitios web respaldados...

Implementa VMs administradas con balanceo de cargas

Análisis de datos, canalización de datos, registros de aplicaciones, inteligencia de...

Crea un almacén de datos con BigQuery

Almacenes de datos, paneles, ETL, estadísticas y análisis de datos

[Ver todas las soluciones](#)

Productos

Crea una VM

Compute Engine

Entrena y aloja modelos de AA

Vertex AI

Crea una base de datos

Cloud SQL

Analiza y administra datos

BigQuery

Crea un bucket de almacenamiento

Cloud Storage

Ejecuta apps alojadas en contenedores

Cloud Run

Crea una app alojada en contenedores

Kubernetes Engine

Incorpora mapas

Google Maps Platform

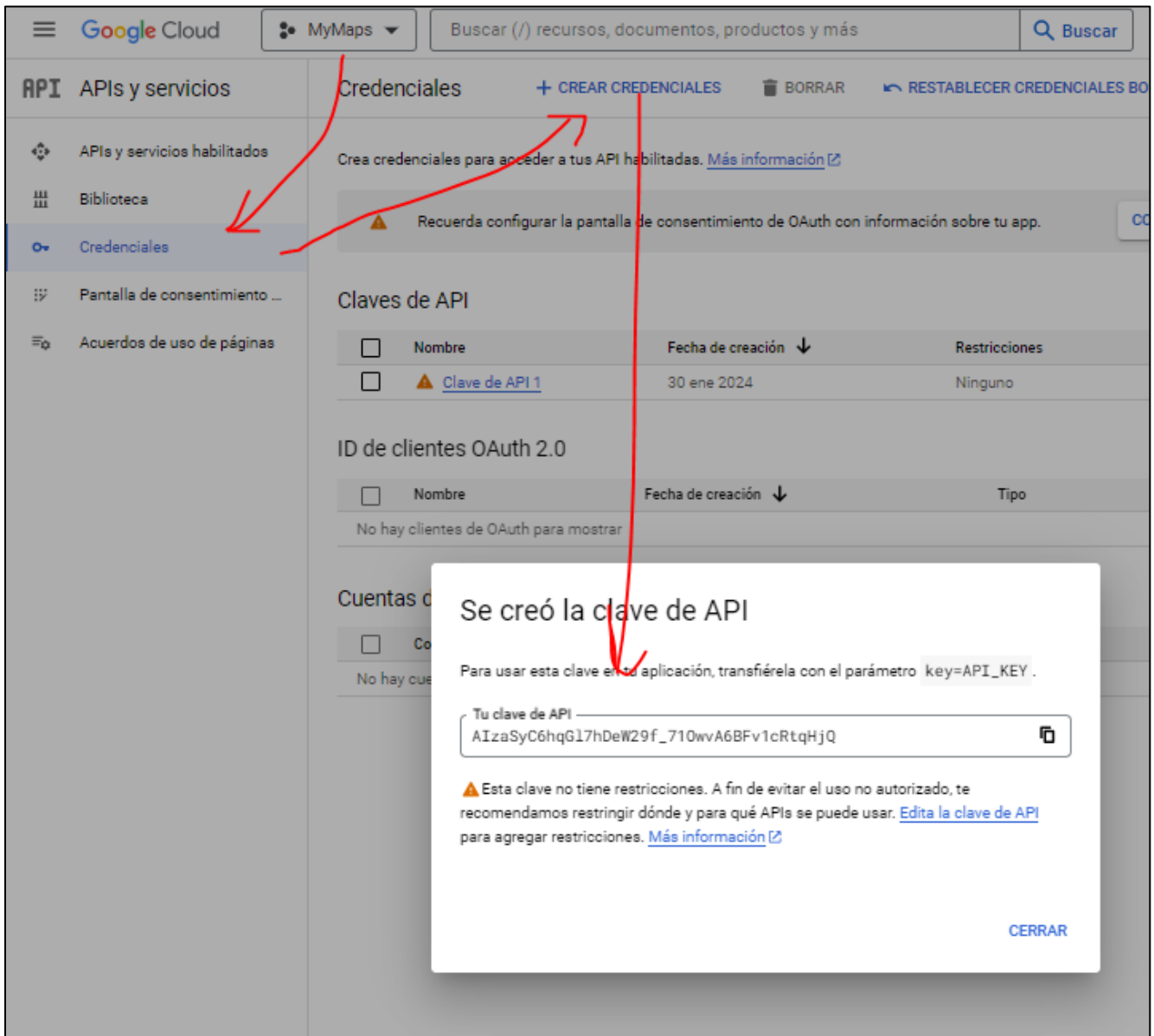
Convierte texto a voz

API de Text-to-Speech

Convierte voz a texto

API de Speech-to-Text

Paso 7. Una vez hayas añadido la API de Google Maps a tu proyecto EjemploGraficos/API y servicios/Credenciales/CREAR CREDENCIALES



Paso 8. Copia la clave de API en la pestaña de meta-data que hemos creado en el paso 2.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="EjemploGoogleMaps3"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.EjemploGoogleMaps3"
        tools:targetApi="31">
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="AIzaSyC6hqG17hDeW29f_710wvA6BFv1cRtqHjQ"/>
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
```

Paso 9. Examina el permiso dentro de la sección manifest, que se ha generado automáticamente:

```
AndroidManifest.xml x
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.ejemplogooglemaps">

    <!--
        The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
        Google Maps Android API v2, but you must specify either coarse or fine
        Location permissions for the "MyLocation" functionality.
    -->
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="EjemploGoogleMaps"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
```

Paso 10. Añade la implementación del gms de play-services-maps en el modulo de gradle.

```
build.gradle.kts (:app) x
You can use the Project Structure dialog to view and edit your project configuration Open (Ctrl+Alt+Mayús+S) H
8
9      defaultConfig { this: ApplicationDefaultConfig
10          applicationId = "com.example.ejemplogooglemaps3"
11          minSdk = 25
12          targetSdk = 33
13          versionCode = 1
14          versionName = "1.0"
15
16          testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
17      }
18
19      buildTypes { this: NamedDomainObjectContainer<ApplicationBuildType>
20          release { this: ApplicationBuildType
21              isMinifyEnabled = false
22              proguardFiles(
23                  getDefaultProguardFile( name: "proguard-android-optimize.txt"),
24                  "proguard-rules.pro"
25              )
26          }
27      }
28      compileOptions { this: CompileOptions
29          sourceCompatibility = JavaVersion.VERSION_1_8
30          targetCompatibility = JavaVersion.VERSION_1_8
31      }
32  }
33
34  dependencies { this: DependencyHandlerScope
35
36      implementation("androidx.appcompat:appcompat:1.6.1")
37      implementation("com.google.android.material:material:1.11.0")
38      implementation("androidx.constraintlayout:constraintlayout:2.1.4")
39      implementation("com.google.android.gms:play-services-maps:18.2.0")
40      testImplementation("junit:junit:4.13.2")
41      androidTestImplementation("androidx.test.ext:junit:1.1.5")
42      androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
43  }
```

Paso 11. Examina el contenido del layout activity_maps.xml:

```
activity_maps.xml x
<?xml version="1.0" encoding="utf-8"?>
<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" />
```

Paso 12. Ejecuta la aplicación y comprueba el resultado:



Paso 13. Abre MapsActivity.java y observa que esta clase hereda de FragmentActivity y que implemente la interfaz OnMapReadyCallback:

```
MapsActivity.java ×
package com.example.ejemplogooglemaps;

import ...

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;
```

Paso 14. Examina el método onCreate de MapsActivity.java:

```
@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(new OnMapReadyCallback() {
        @Override
        public void onMapReady(GoogleMap googleMap) {
            // Add a marker in Sydney and move the camera
            LatLng sydney = new LatLng(-34, 151);
            mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));
            mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
        }
    });
}
```

Paso 15. Examina el método onMapReady que se debe de implementar, que será llamado en el momento en que el mapa está disponible. Es en esta función donde podremos manipular el mapa. Una implementación mínima sería la siguiente:

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

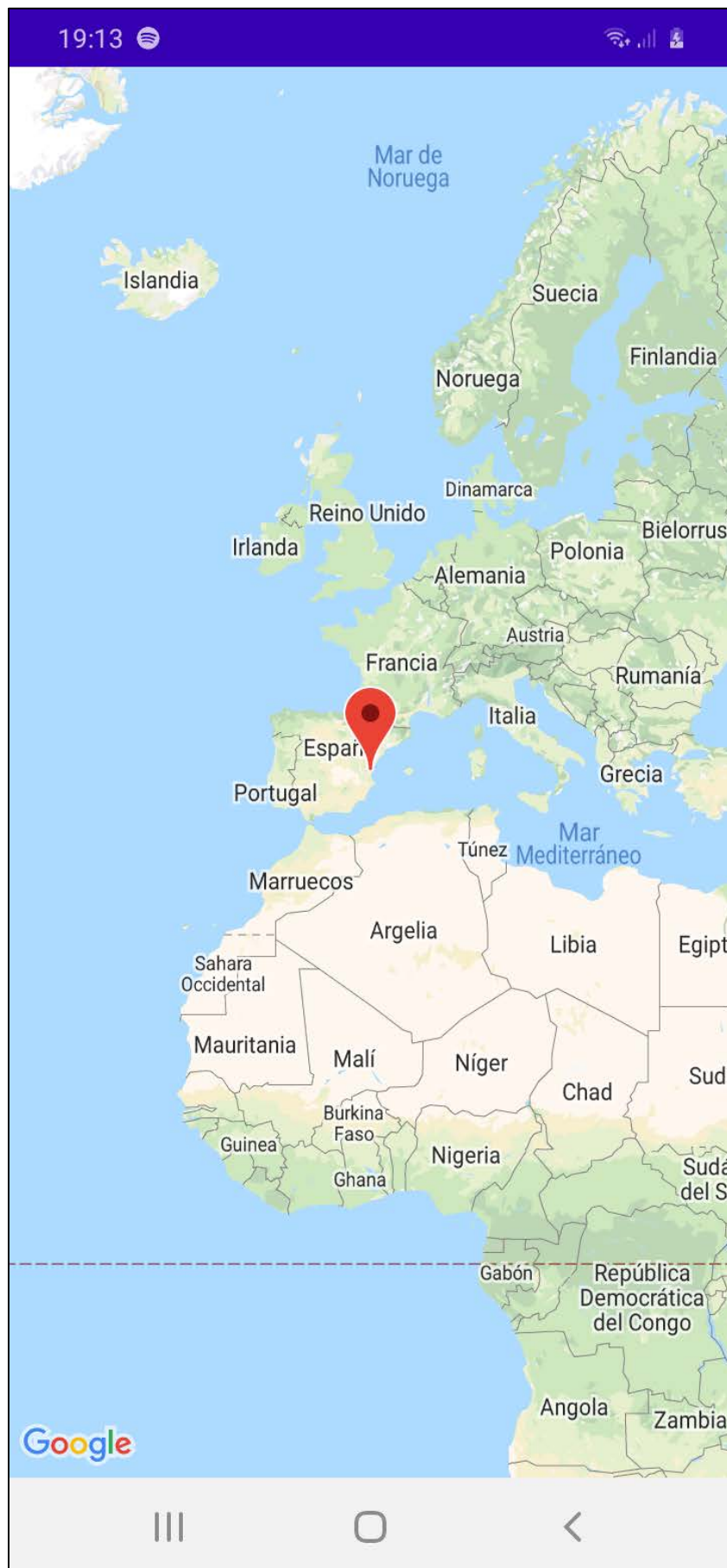
    // Add a marker in Sydney and move the camera
    LatLng sydney = new LatLng(-34, 151);
    mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));
    mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
}
```

Paso 16. Cambia el código y apunta a la Universidad Politécnica de Valencia:

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

    // Add a marker in Sydney and move the camera
    //LatLng sydney = new LatLng(-34, 151);
    LatLng UPV = new LatLng(39.481106, -0.340987); //Nos ubicamos en la UPV
    mMap.addMarker(new MarkerOptions().position(UPV).title("Marker UPV"));
    mMap.moveCamera(CameraUpdateFactory.newLatLng(UPV));
}
```

Paso 17. Ejecuta la aplicación y comprueba el resultado:



Parte II: Introduciendo código en Google Maps

En el ejercicio anterior hemos visto un ejemplo muy básico donde solo se mostraba un mapa con las opciones predeterminadas. En este ejercicio aprenderemos a configurarlo y añadir marcadores desde el código.

Paso 18. Abre el *layout activity_main.xml* y añade los siguientes tres botones dentro del `<LinearLayout>` (tras el `<fragment ...>`):

```
activity_maps.xml ×
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MapsActivity">
    <fragment
        android:id="@+id/map"
        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_width="match_parent"
        android:layout_height="769dp"
        android:layout_weight="1" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="moveCamera"
            android:text="ir a UPV" />
        <Button
            android:id="@+id/button2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="animateCamera"
            android:text="animar a UPV" />
        <Button
            android:id="@+id/button3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="addMarker"
            android:text="marcador" />
    </LinearLayout>
</LinearLayout>
```

Paso 19. Sustituye el contenido de *MainActivity.java* por:

```
public class MapsActivity extends FragmentActivity implements OnMapReadyCallback,
    GoogleMap.OnMapClickListener {

    private GoogleMap mapa;
    private final LatLng UPV = new LatLng( v: 39.481106, v1: -0.340987);

    @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( onMapReadyCallback: this);
    }

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

        // Add a marker in Sydney and move the camera
        //LatLng sydney = new LatLng(-34, 151);
        //LatLng UPV = new LatLng(39.481106, -0.340987); //Nos ubicamos en la UPV
        //mMap.addMarker(new MarkerOptions().position(UPV).title("Marker UPV"));
        //mMap.moveCamera(CameraUpdateFactory.newLatLng(UPV));

        mapa = googleMap;
        mapa.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
        mapa.getUiSettings().setZoomControlsEnabled(false);
        mapa.moveCamera(CameraUpdateFactory.newLatLngZoom(UPV, v: 15));
        mapa.addMarker(new MarkerOptions()
            .position(UPV)
            .title("UPV")
            .snippet("Universidad Politécnica de Valencia")
            .icon(BitmapDescriptorFactory
                .fromResource(android.R.drawable.ic_menu_compass))
            .anchor( v: 0.5f, v1: 0.5f));
        mapa.setOnMapClickListener(this);
        if (ActivityCompat.checkSelfPermission( context: this,
            android.Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED) {
            mapa.setMyLocationEnabled(true);
            mapa.getUiSettings().setCompassEnabled(true);
        }
    }
}
```

```

public void moveCamera(View view) { mapa.moveCamera(CameraUpdateFactory.newLatLng(UPV)); }

public void animateCamera(View view) { mapa.animateCamera(CameraUpdateFactory.newLatLng(UPV)); }

public void addMarker(View view) {
    mapa.addMarker(new MarkerOptions().position(
        mapa.getCameraPosition().target));
}

@Override public void onMapClick(LatLng puntoPulsado) {
    mapa.addMarker(new MarkerOptions().position(puntoPulsado)
        .icon(BitmapDescriptorFactory
            .defaultMarker(BitmapDescriptorFactory.HUE_YELLOW)));
}
}

```

Comenzamos declarando dos objetos: **UPV** hacer referencia a la posición geográfica de la Universidad Politécnica de Valencia y **mapa**, que nos permitirá acceder al objeto **GoogleMap** que hemos insertado en un *fragment* de nuestro *Layout*. Este objeto es inicializado al comienzo de **onCreate()** y a continuación se utilizan una serie de métodos para configurarlo. **setMapType()** permite seleccionar el tipo de mapa (normal, satélite, híbrido o relieve). Para averiguar las constantes correspondientes te recomendamos que utilices la opción de autocompletar (escribes **GoogleMap** y podrás seleccionar las constantes de esta clase). El método **moveCamera()** desplaza el área de visualización a una determinada posición (**UPV**) a la vez que define el nivel de zoom (**15**). El nivel de zoom ha de estar en un rango de 2 (continente) hasta 21 (calle). El método **setMyLocationEnabled(true)** activa la visualización de la posición del dispositivo por medio del típico triángulo azul. El método **getUiSettings()** permite configurar las acciones del interfaz de usuario. En este ejemplo se han utilizado dos, desactivar los botones de zoom y visualizar una brújula. Puedes usar autocompletar para descubrir otras posibles configuraciones. El método **addMarker()** permite añadir los típicos marcadores que habrás visto en muchos mapas. En este ejemplo se indica la posición, el título, la descripción, un icono (en el ejemplo se utiliza el mismo *drawable* usado como icono de la aplicación) y el punto del icono que haremos coincidir con el punto exacto que queremos indicar en el mapa. Un valor de (0, 0) corresponde a la esquina superior izquierda del icono y (1, 1) a la esquina inferior derecha. Como nuestro icono tiene forma de círculo, hemos indicado el valor (0.5, 0.5) para que coincida con su centro. Finalmente, hemos registrado un escuchador de evento para detectar pulsaciones sobre la pantalla. El escuchador vamos a ser nosotros mismos (**this**), por lo que hemos implementado el interface **OnMapClickListener** y añadido el método **onMapClick()**.

A continuación se incluyen los tres métodos que se ejecutarán al pulsar sobre los botones añadidos al *layout*. El primero, **moveCamera()** desplaza el punto de visualización a la UPV. A diferencia del uso anterior, sin cambiar el nivel de zoom que el usuario tenga seleccionado.

El segundo, **animateCamera()**, nos desplaza hasta nuestra posición actual por medio de una animación (similar a la que a veces utilizan en el Tele Diario para mostrar un punto en conflicto). Observa como el método **getMyLocation()** permite obtener la posición del dispositivo sin usar el API Android de posicionamiento. Si usas este método verifica siempre que ya se dispone de una posición (**!=null**) y que has pedido permisos de localización.

El tercero, **addMarker()**, añade un nuevo marcador en el centro del mapa que estamos observando (**getCameraPosition()**). En este caso usaremos el marcador por defecto, sin información adicional.

Como hemos indicado **onMapClick()** será llamado cuando se pulse sobre el mapa. Se pasa como parámetro las coordenadas donde se ha pulsado, que utilizaremos para añadir un marcador. Esta vez el marcador por defecto, es de color amarillo.

Paso 20. Ejecuta la aplicación. El resultado se muestra a continuación:

