

Software para Dispositivos Móviles
Grado en Ingeniería Informática del Software
Escuela de Ingeniería Informática – Universidad de Oviedo

Configuración inicial mapas Android

Desarrollo del mapa básico

Juan Ramón Pérez Pérez

Departamento de Informática

jrpp@uniovi.es

API de Google Maps para Android

- Nos permite realizar distintas operaciones con los mapas:
 - Mostrar mapas / imágenes “satélite”
 - Mostrar información personalizada sobre ellos.
 - Permitir al usuario interactuar con ellos

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

Pasos necesarios para utilizar Google Maps



Instalar
Google Play
Services
SDK

Establecer
configuración
del
proyecto

Obtener
una Map
API Key

Comprobar
versiones
librerías

Personalización
objeto
GoogleMap

Instalación y configuración de Google Play services

- Google Maps Android API se distribuye como parte de Google Play Services SDK
- Instalar en el sistema Google Play Services
 - Librería de desarrollo (Sistema de desarrollo)
 - Ejecución (Dispositivo móvil / emulador)



Instalación de Google play services



Descargar Google Play services que aparece en la sección de SDK Tools del SDK Manager

- Comprobar Google APIs de la plataforma SDK instalada
- Android Support Repository (Fragments)

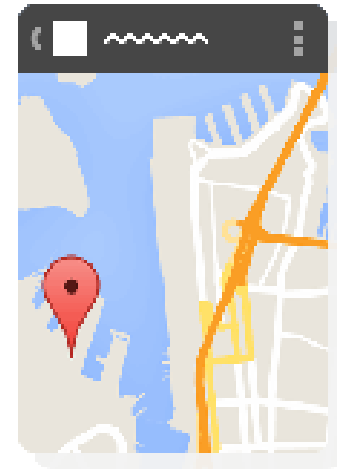


Debemos crear un AVD (Android Virtual Device) con Google APIS

- Para probarlo en el emulador debemos usar al menos la versión 4.2.2 (API 17)

Proyecto App Android con Mapas (Android Studio)

- Plantilla para la actividad inicial:
 - Google Maps Activity



Google Maps Activity



Casi está listo:
Sólo falta un detalle
La API Key

API Key para mapas

- Maps API Key está basada en el certificado digital único de nuestra aplicación.
 - SHA-1 fingerprint
- Dos tipos de certificados:
 - Debug, para pruebas de desarrollo (es único para cada máquina),
 - Release, para subirlo al Google play store.
- Google utiliza este sistema → para controlar quién se conecta y el consumo que hace del servicio.

Obtención de la API key de debug (procedimiento completo)

Localización almacén
de claves de debug

- (Eclipse ADT) **Windows > Prefs > Android > Build**
- Dir. almacén: users\<usuario>\.android

Uso de keytool para
obtener la huella
digital SHA1

- `keytool -list -alias androiddebugkey -keystore C:\Users\<usuario>\.android\debug.keystore -storepass android -keypass android`

Consola de APIs de
Google

- <https://console.developers.google.com>
- Panel de control: Crear un proyecto
- Biblioteca de APIs > Maps Android > Habilitar

Crear API key

- Credenciales
- Crear nueva clave de API
- Restringir la clave

Consola de APIs de Google

1. Debemos autenticarnos con nuestra cuenta de Google
2. Seleccionar un Proyecto (crear uno nuevo)
3. Biblioteca de APIs
 - Google Maps Android API → Activar este servicio
4. Credenciales
 - Clave de API
5. Restringir clave > Aplicaciones para Android :
 - Nombre **completo** del paquete:
es.uniovi.ingenieriainformatica.sdm.mapabasico
 - SHA-1 fingerprint

Método abreviado con la URL que genera Android Studio


Incluir la API KEY en nuestra app

(Android Studio)

- La API KEY es un código alfanumérico de cifras y letras
- Copiar en values/google_maps_api.xml
- Si la hemos restringido: esta API KEY es exclusiva para este proyecto en esta máquina de desarrollo

Ejecución de la aplicación con el mapa

- Emulador:
 - AVD (Android Virtual Device) con Google APIS
 - Versión de Android 4.2.2 o superior
- Dispositivo real:
 - Debe disponer del componente Google play services (Servicios de Google Play)
 - "Google Play Services" debe estar actualizado



¿De dónde viene la magia?

¿Qué ha creado la plantilla para que todo esto funcione?

Enlace con las librerías en el fichero Gradle

- Fichero build.gradle (Module: app) sección *dependencies*
- Aquí se incluyen las librerías que incorpora el proyecto
 - Android Support Repository
 - 'com.android.support:appcompat-v7:23.4.0'
 - Google play Services
 - 'com.google.android.gms:play-services:9.8.0'

Manifiesto: Referencia a la API KEY

- Fichero AndroidManifest.xml
- Como elemento hijo de <application>

```
<meta-dataandroid:name="com.google.android.geo.API_KEY"  
    android:value="AIzaSyB_hjtiMKAdU3OZjSCqRAZ8YfOm3jrH2L0"  
>
```

Layout: MapFragment

- El XML de layout: layout/activity_maps.xml contiene un fragment.
- Componente embebido en la Activity que permite mostrar mapas a través de Fragments
- Los fragments son dinámicos → el editor gráfico no los muestra directamente.

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


Qué son los Fragment

- Un Fragment representa una porción de la interfaz dentro de una Activity.
- Surgen debido a la necesidad de soportar una UI más flexible y dinámica en pantallas grandes (tablets)
 - A partir de Android 3.0 (Anteriores Android library-support)
- Describen su propio layout

MainActivity

- Desde el código de la actividad principal activamos el Fragment con el mapa
- El mapa por defecto dispone de interactividad sin programar.

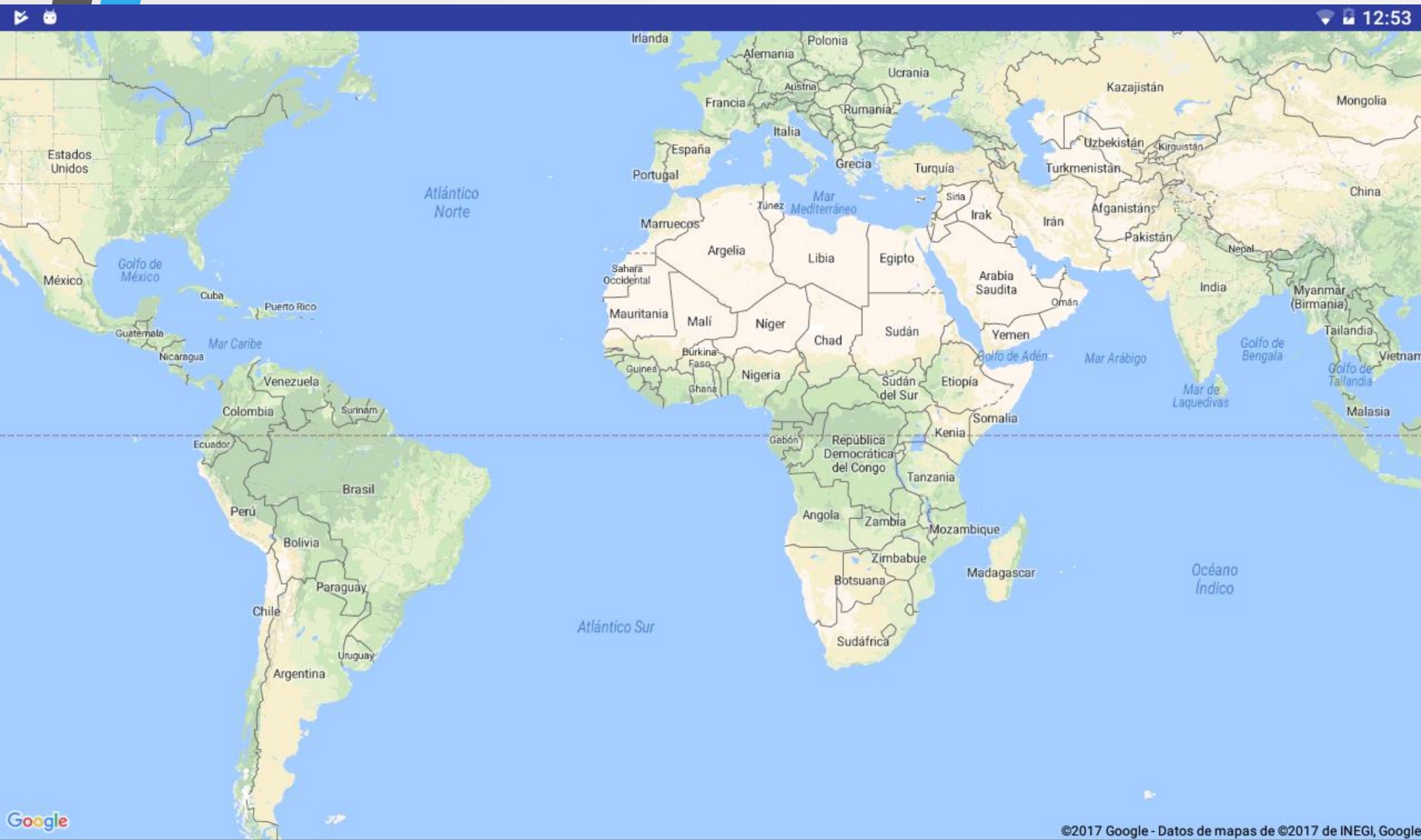
```
import android.support.v4.app.FragmentActivity;

import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.SupportMapFragment;

public class MapsActivity extends FragmentActivity
    implements OnMapReadyCallback {

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

Mapa inicial



Objeto GoogleMap

- La clase GoogleMap modela el objeto “mapa” que muestra nuestra aplicación
- Está contenido en el MapFragment que hemos creado
- Referencia:
 - <https://developers.google.com/android/reference/com/google/android/gms/maps/GoogleMap>

Funcionalidad de GoogleMap

- Gestión del mapa
 - Conectarse al servicio de Google Maps
 - Descargar los mapas
 - Mostrarlos en el dispositivo
- Gestión de la interacción del usuario
 - Mostrar controles (zoom, brújula...)
 - Responder ante gestos (pan y zoom)

Obtener el objeto GoogleMap

- Para obtener un objeto GoogleMap debemos de obtener el Fragment que contiene el mapa y después acceder al mapa.

```
MapFragment mapFragment =  
    (MapFragment) getFragmentManager()  
        .findFragmentById(R.id.map);  
  
mapFragment.getMapAsync(this);
```

Método callback GoogleMap

- Se debe implementar un método callback en el objeto MapFragment.
- Siempre proporciona instancia no nula de GoogleMap

@Override

```
public void onMapReady(GoogleMap map) {  
    map.addMarker(new MarkerOptions()  
        .position(new LatLng(43.354,-5.851))  
        .title("Marker"));  
}
```

Mapa inicial

- Mapa ortofotos + etiquetas
 - `setMapType`
- Información de tráfico (es necesario hacer más zoom para verla)
 - `setTrafficEnabled`
- Marcador en la Escuela de Ingeniería Informática

