

Tema 5

1. Sensores
2. Google Play Service
3. Google Maps
4. Publicar

Tema 5

Sensores

Posicionamiento.

- Obtener el servicio

```
// Acquire a reference to the system Location Manager
LocationManager locationManager = (LocationManager) this.getSystemService(Context.LOCATION_SERVICE);
```

- Escuchar cambios de posición

```
// Define a listener that responds to location updates
LocationListener locationManager = new LocationListener() {
    public void onLocationChanged(Location location) {
        // Called when a new location is found by the network location provider.
        makeUseOfNewLocation(location);
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {}

    public void onProviderEnabled(String provider) {}

    public void onProviderDisabled(String provider) {}
};
```

Tema 5

Sensores

Posicionamiento.

- Registrar el servicio

```
LocationProvider locationProvider = LocationManager.NETWORK_PROVIDER;  
// Or, use GPS location data:  
// LocationProvider locationProvider = LocationManager.GPS_PROVIDER;  
  
locationManager.requestLocationUpdates(locationProvider, 0, 0, locationListener);
```

- Permisos

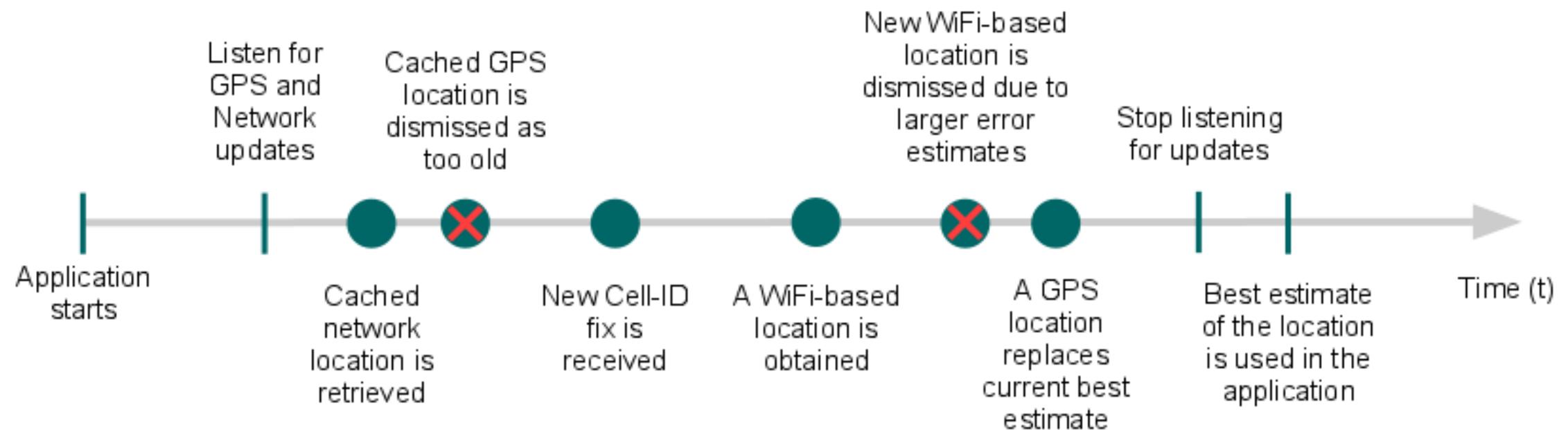
```
<manifest ... >  
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />  
    ...  
</manifest>
```

Tema 5

Sensores

Posicionamiento.

- Flujo de datos
- Iniciar la aplicación
- Escuchar actualizaciones de posición
- Estimar la mejor actualización
- Dejar de escuchar actualizaciones



Tema 5

Sensores

Posicionamiento.

- Obtener última posición conocida

```
LocationProvider locationProvider = LocationManager.NETWORK_PROVIDER;  
// Or use LocationManager.GPS_PROVIDER  
  
Location lastKnownLocation = locationManager.getLastKnownLocation(locationProvider);
```

- Estimar la mejor posición
 - Posición mas reciente
 - Mayor precisión
 - Fuente de mayor precisión

Tema 5

Sensores

Posicionamiento.

- Detener actualizaciones

```
// Remove the listener you previously added  
locationManager.removeUpdates(locationListener);
```

- Activar GPS

```
Intent intent = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);  
startActivityForResult(intent, RESULT_ENABLE_GPS_FROM_SETTING);
```

Tema 5

Sensores

Posicionamiento.

- Geocoder

```
Geocoder gc = new Geocoder(this);  
List<Address> direcciones = gc.getFromLocation(mCurrentLocation.getLatitude(), mCurrentLocation.getLongitude(), 1);
```

```
if(direcciones.size() > 0){  
    StringBuilder sb = new StringBuilder();  
    sb.append(direcciones.get(0).getLocality()).append("\n");  
    sb.append(direcciones.get(0).getPostalCode()).append("\n");  
    sb.append(direcciones.get(0).getCountryName()).append("\n");  
}
```

```
Geocoder gc = new Geocoder(this);  
List<Address> direcciones = gc.getFromLocationName(direccion, 1);
```

```
if(direcciones.size() > 0){  
    Location loc = new Location(LocationManager.GPS_PROVIDER);  
    loc.setLatitude(direcciones.get(0).getLatitude());  
    loc.setLongitude(direcciones.get(0).getLongitude());  
}
```

Tema 5

Sensores

Tipos de sensores.

TYPE_ACCELEROMETER

TYPE_ALL

TYPE_GRAVITY

TYPE_GYROSCOPE

TYPE_LIGHT

TYPE_LINEAR_ACCELERATION

TYPE_MAGNETIC_FIELD

TYPE_ORIENTATION

TYPE_PRESSURE

TYPE_PROXIMITY

TYPE_ROTATION_VECTOR

TYPE_TEMPERATURE

Tema 5

Gestor de sensores.

Sensores

- Recuperar un sensor

```
mSensorManager = (SensorManager) getSystemService (SENSOR_SERVICE) ;  
mAccelerometer = mSensorManager.getDefaultSensor (Sensor.TYPE_ACCELEROMETER) ;
```

- Registrar un sensor

```
mSensorManager.registerListener (this, mAccelerometer, SensorManager.SENSOR_DELAY_NORMAL) ;
```

- Eliminar un sensor

```
mSensorManager.unregisterListener (this) ;
```

- Escuchar un sensor

```
public void onAccuracyChanged (Sensor sensor, int accuracy) {  
}  
  
public void onSensorChanged (SensorEvent event) {  
}
```

Tema 5

Sensores

Gestor de sensores.

- Refresco de un sensor

SENSOR_DELAY_FASTEST

SENSOR_DELAY_GAME

SENSOR_DELAY_NORMAL

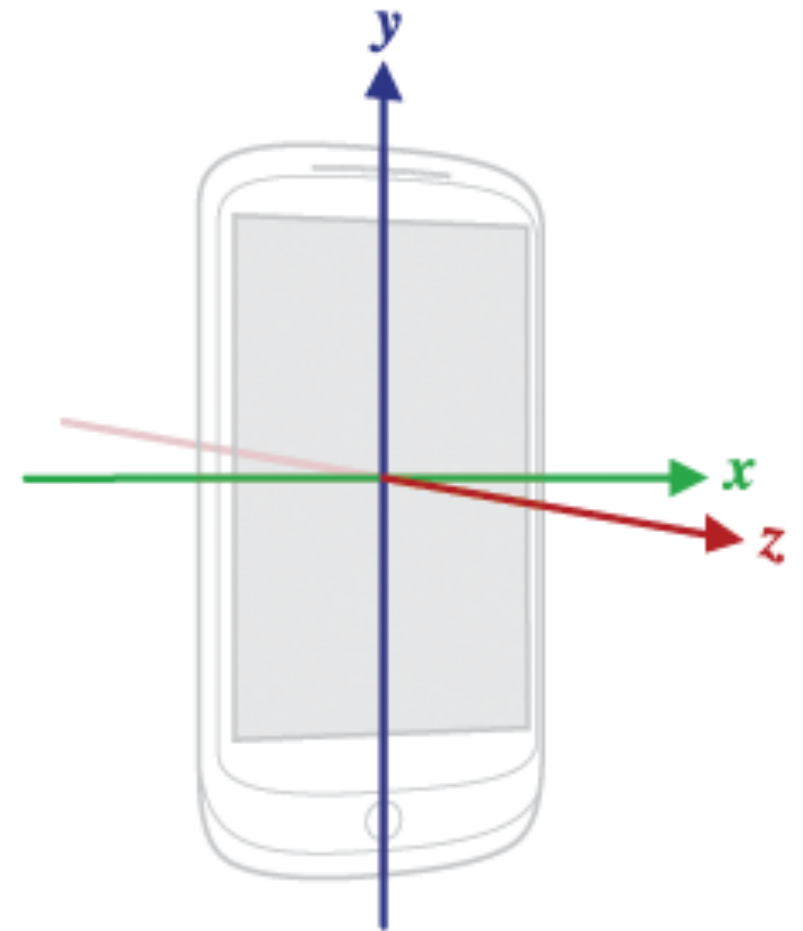
SENSOR_DELAY_UI

Tema 5

Sensores

Eventos del sensor.

- Tipo de sensor
- Huella de tiempo
- Datos del sensor
 - Coordenadas relativas a la pantalla
 - Los ejes no cambian con la orientación



Tema 5

Sensores

Eventos del sensor.

- Accuracy
 - Precisión del sensor
- Sensor
 - Tipo de sensor
- Timestamp
 - Huella de tiempo
- Values
 - Valores emitidos por el sensor

public int	accuracy
public Sensor	sensor
public long	timestamp
public final float[]	values

Tema 5

Sensores

Datos del sensor.

- TYPE_ACCELEROMETER
 - m/s^2
 - Values[0]
 - Aceleración en el eje X
 - Values[1]
 - Aceleración en el eje Y
 - Values[2]
 - Aceleración en el eje Z

Tema 5

Sensores

Datos del sensor.

- TYPE_MAGNETIC_FILED
 - uT (Micro Tesla)
 - Values[0]
 - Campo magnético en el eje X
 - Values[1]
 - Campo magnético en el eje Y
 - Values[2]
 - Campo magnético en el eje Z

Tema 5

Sensores

Datos del sensor.

- TYPE_GYROSCOPE
 - Radianes/segundo
 - Values[0]
 - Velocidad angular en el eje X
 - Values[1]
 - Velocidad angular en el eje Y
 - Values[2]
 - Velocidad angular en el eje Z

Tema 5

Sensores

Datos del sensor.

- TYPE_LIGHT
 - Lux
 - Values[0]
 - Intensidad de la luz
- TYPE_PROXIMITY
 - Centimetros
 - Values[0]
 - Distancia registrada por el sensor
 - Algunos sensores solo admiten
 - cerca (0) y lejos (1)

Tema 5

Sensores

Datos del sensor.

- TYPE_GRAVITY
 - m/s^2
 - Vector tridimensional que representa la gravedad
- TYPE_LINEAR_ACCELERATION
 - m/s^2
 - Vector tridimensional que representa la aceleración del dispositivo

Tema 5

Sensores

Datos del sensor.

- TYPE_ORIENTATION
 - Grados
- Values[0]
 - Azimuth, 0 a 359
 - Angulo eje Y y Norte magnetico
- Values[1]
 - Pitch, -180 a 180
 - Rotación alrededor del eje X
- Values[2]
 - Roll, -90 a 90
 - Rotación alrededor del eje Y

Tema 5

Google Play Services

Google ofrece una variedad de servicios que nos ayudan a crear nuevas fuentes de ingresos, administrar la distribución de aplicaciones, seguimiento de uso de aplicaciones y mejorar nuestra aplicación con características como mapas, inicio de sesión, y los mensajes de la nube.

Aunque estos servicios de Google no están incluidos en la plataforma Android son compatibles con la mayoría de los dispositivos Android.

Al usar estos servicios, podemos distribuir nuestra aplicación en Google Play para todos los dispositivos con Android 2.2 o superior.

El fin de la fragmentación o el comienzo de un nuevo modelo.

Google Play Services + Compability Pack => [Dispositivos < 2.2 \approx 1%]

Tema 5

Google Play Services

Google Maps

Completamente integrado en dispositivos Android.

Nos permite personalizar el mapa con marcadores y superposiciones, controlar la perspectiva del usuario, dibujar líneas y formas, y mucho más.

Google+

Permite a los usuarios iniciar sesión con su cuenta de Google, personalizar la experiencia del usuario con información Google+, sacar a la gente en nuestra aplicación con mensajes interactivos, y añadir el botón +1 para que los usuarios puedan recomendar su contenido.

GCM

Permite notificar de forma inmediata a los usuarios sobre los eventos puntuales mediante la entrega de mensajes livianos desde su servidor web. No hay cuotas o cargos para usar Google Mensajería Cloud.

Tema 5

Google Play Services

Google Play In-App Billing

Construir una aplicación con un flujo de ingresos constante ofreciendo nuevos contenidos o recursos a los usuarios directamente desde la aplicación. Todas las transacciones son manejadas por Google Play Store para una experiencia de usuario sencilla.

Google Analytics

Permite medir el éxito de una aplicación y obtener información sobre cómo los usuarios se relacionan con el contenido de la aplicación. Podemos hacer seguimiento a las compras, el número de usuarios activos, patrones de interacción, y mucho más.

Google de AdMob

Visualizar los anuncios de AdMob es una opción para monetizar nuestras aplicaciones que aprovecha múltiples redes publicitarias con anuncios orientados y varios formatos de visualización.

Tema 5

Google Play Services

Google Play Game Services

Podemos añadir logros, tablas de clasificación, multijugador en tiempo real y otras características. Permite a los jugadores ingresar usando su cuenta de Google+ y compartir su experiencia de juego con los amigos.

Google Play Distribution

Nos permite gestionar la distribución de aplicaciones. Ofrece características que nos permiten controlar lo que los usuarios pueden descargar de nuestra aplicación, así como entregar versiones separadas de la aplicación sobre la base de ciertas características como la versión de la plataforma.

Tema 5

Location Service

LocationService mantiene automáticamente la ubicación actual del usuario, por lo que toda la aplicación puede recuperarla cuando sea necesario.

La precisión de la localización se basa en los permisos de ubicación que hemos solicitado y en los sensores de localización que están actualmente activos.

Envía la posición actual hasta la aplicación a través del cliente LocationClient.

Las aplicaciones que utilizan servicios de localización deben solicitar permisos de ubicación.

Android tiene dos permisos de ubicación:
ACCESS_COARSE_LOCATION y ACCESS_FINE_LOCATION.

El permiso que usemos definirá la exactitud de la ubicación actual.

Tema 5

Location Service

LocationServices es parte de la Google Play Services.

Ya que es difícil anticipar el estado del dispositivo del usuario, siempre deberemos comprobar que se ha instalado el APK antes de intentar conectarnos a los servicios de localización.

Para comprobar que se ha instalado el APK, usamos `isGooglePlayServicesAvailable()` que devuelve un `ConnectionResult`.

Si se produce un error llamaremos a `GooglePlayServicesUtil.getErrorDialog ()` para recuperar diálogo que solicita a los usuarios descargar el APK de Google Play Services

Tema 5

Location Service

```
private boolean servicesConnected() {
    // Check that Google Play services is available
    int resultCode =
        GooglePlayServicesUtil.
            isGooglePlayServicesAvailable(this);
    // If Google Play services is available
    if (ConnectionResult.SUCCESS == resultCode) {
        // In debug mode, log the status
        Log.d("Location Updates",
            "Google Play services is available.");
        // Continue
        return true;
    }
    // Google Play services was not available for some reason
    else {
        // Get the error code
        int errorCode = connectionResult.getErrorCode();
        // Get the error dialog from Google Play services
        Dialog errorDialog = GooglePlayServicesUtil.getErrorDialog(
            errorCode,
            this,
            CONNECTION_FAILURE_RESOLUTION_REQUEST);

        // If Google Play services can provide an error dialog
        if (errorDialog != null) {
            // Create a new DialogFragment for the error dialog
            AlertDialogFragment errorFragment =
                new AlertDialogFragment();
            // Set the dialog in the DialogFragment
            errorFragment.setDialog(errorDialog);
            // Show the error dialog in the DialogFragment
            errorFragment.show(getSupportFragmentManager(),
                "Location Updates");
        }
    }
}
```

Tema 5

Location Service

Para obtener la ubicación actual, creamos un cliente de ubicación, conectarse a servicios de localización, y luego llamar a su método `getLastLocation ()`.

El valor devuelto es la mejor ubicación basada en los permisos que solicita la aplicación y los sensores de localización actualmente habilitados.

Tenemos que implementar las interfaces que utiliza `ServiceLocation` para comunicarse con su aplicación:

`ConnectionCallbacks`

Avisa cuando un cliente local está conectado o desconectado.

`OnConnectionFailedListener`

Especifica si se produce un error al intentar conectar con el servicio.

```
public class MainActivity extends FragmentActivity implements
    GooglePlayServicesClient.ConnectionCallbacks,
    GooglePlayServicesClient.OnConnectionFailedListener {
```

Tema 5

Location Service

```
...
/*
 * Called by Location Services when the request to connect the
 * client finishes successfully. At this point, you can
 * request the current location or start periodic updates
 */
@Override
public void onConnected(Bundle dataBundle) {
    // Display the connection status
    Toast.makeText(this, "Connected", Toast.LENGTH_SHORT).show();
}
...
/*
 * Called by Location Services if the connection to the
 * location client drops because of an error.
 */
@Override
public void onDisconnected() {
    // Display the connection status
    Toast.makeText(this, "Disconnected. Please re-connect.",
        Toast.LENGTH_SHORT).show();
}
```

Tema 5

Location Service

```
/*
 * Called by Location Services if the attempt to
 * Location Services fails.
 */
@Override
public void onConnectionFailed(ConnectionResult connectionResult) {
    /*
     * Google Play services can resolve some errors it detects.
     * If the error has a resolution, try sending an Intent to
     * start a Google Play services activity that can resolve
     * error.
     */
    if (connectionResult.hasResolution()) {
        try {
            // Start an Activity that tries to resolve the error
            connectionResult.startResolutionForResult(
                this,
                CONNECTION_FAILURE_RESOLUTION_REQUEST);
        }
        /*
         * Thrown if Google Play services canceled the original
         * PendingIntent
         */
        catch (IntentSender.SendIntentException e) {
            // Log the error
            e.printStackTrace();
        }
    } else {
        /*
         * If no resolution is available, display a dialog to the
         * user with the error.
         */
        showErrorDialog(connectionResult.getErrorCode());
    }
}
```

Tema 5

Location Service

Debemos crear el cliente en onCreate().

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    /*
     * Create a new location client, using the enclosing class to
     * handle callbacks.
     */
    mLocationClient = new LocationClient(this, this, this);
    ...
}
```

Tema 5

Location Service

Lo conectaremos en onStart ().

```
/*
 * Called when the Activity becomes visible.
 */
@Override
protected void onStart() {
    super.onStart();
    // Connect the client.
    mLocationClient.connect();
}
```


Tema 5

Location Service

Desconectamos el cliente en onStop ()

```
/*  
 * Called when the Activity is no longer visible.  
 */  
@Override  
protected void onStop() {  
    // Disconnecting the client invalidates it.  
    mLocationClient.disconnect();  
    super.onStop();  
}
```

Tema 5

Location Service

Cuando necesitemos una ubicación hacemos uso de `getLastLocation()`

```
public class MainActivity extends FragmentActivity implements
    GooglePlayServicesClient.ConnectionCallbacks,
    GooglePlayServicesClient.OnConnectionFailedListener {
    ...
    // Global variable to hold the current location
    Location mCurrentLocation;
    ...
    mCurrentLocation = mLocationClient.getLastLocation();
    ...
}
```


Tema 5

Location Service

LocationService envía actualizaciones periódicas de la ubicación del dispositivo.

El método `onLocationChanged()` que se usa para recibir las actualizaciones se especifica en la interfaz `LocationListener`.

El argumento es un objeto `Location` que contiene la latitud y longitud del lugar.

```
public class MainActivity extends FragmentActivity implements
    GooglePlayServicesClient.ConnectionCallbacks,
    GooglePlayServicesClient.OnConnectionFailedListener,
    LocationListener {
    ...
    // Define the callback method that receives location updates
    @Override
    public void onLocationChanged(Location location) {
        // Report to the UI that the location was updated
        String msg = "Updated Location: " +
            Double.toString(location.getLatitude()) + "," +
            Double.toString(location.getLongitude());
        Toast.makeText(this, msg, Toast.LENGTH_SHORT).show();
    }
}
```

Tema 5

Location Service

Nos permite controlar el intervalo entre las actualizaciones y la precisión de la ubicación establecimiento de los valores de un objeto `LocationRequest`.

`LocationRequest.setInterval ()`.

Este método establece la periodicidad en milisegundos de las actualizaciones de ubicación. Si no hay otras aplicaciones que esten conectadas recibiremos las actualizaciones a este ritmo.

`LocationRequest.setFastestInterval ()`.

Este método establece la tasa más rápida en milisegundos que nuestra aplicación es capaz de gestionar.

Es necesario establecer esta tasa porque otras aplicaciones también afectan a la velocidad a la que se envían las actualizaciones.

`LocationRequest.setPriority()`

Establece la precisión de las ubicaciones

Tema 5

Location Service

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // Create the LocationRequest object
    mLocationRequest = LocationRequest.create();
    // Use high accuracy
    mLocationRequest.setPriority(
        LocationRequest.PRIORITY_HIGH_ACCURACY);
    // Set the update interval to 5 seconds
    mLocationRequest.setInterval(UPDATE_INTERVAL);
    // Set the fastest update interval to 1 second
    mLocationRequest.setFastestInterval(FATEST_INTERVAL);
    ...
}
```

Tema 5

Location Service

Iniciar y detener las actualizaciones.

Es una buena practica detener las actualizaciones cuando la actividad pasa a segundo plano y volver a activarlas, si procede, al volver a estar activa.

```
/**
 * In response to a request to start updates, send a request
 * to Location Services
 */
private void startPeriodicUpdates() {
    mLocationClient.requestLocationUpdates(mLocationRequest, this);
    mConnectionState.setText(R.string.location_requested);
}

/**
 * In response to a request to stop updates, send a request to
 * Location Services
 */
private void stopPeriodicUpdates() {
    mLocationClient.removeLocationUpdates(this);
    mConnectionState.setText(R.string.location_updates_stopped);
}
```

Tema 5

Google Maps

Con la API de Google Maps podemos añadir mapas basados en datos de Google.

El API se encarga de automatizar el acceso a los servidores de Google Maps, la descarga de datos, visualización del mapa, y la respuesta a los gestos del mapa.

También puede utilizar llamadas a la API para añadir marcadores, polígonos, y las superposiciones de un mapa básico, y para cambiar la visión del usuario de un mapa de la zona en particular.

Estos objetos proporcionan información adicional para las ubicaciones de los mapa y permiten la interacción del usuario con el mapa.

Tema 5

Google Maps

El API permite agregar estos gráficos a un mapa:

1. Iconos fijados a posiciones específicas en el mapa (marcadores).
2. Conjuntos de segmentos de líneas (polilíneas).
3. Segmentos cerrados (polígonos).
4. Imágenes (BMP) fijadas a posiciones específicas en el mapa.
5. Los conjuntos de imágenes que se muestran encima de los tiles de los mosaicos del mapa base.

Tema 5

Google Maps

Crear una aplicación que utiliza la API de Google Maps v2 requiere varios pasos.

El proceso general de la adición de un mapa a una aplicación de Android es la siguiente:

Descargar y configurar Google Play Services SDK.

El API de Google Maps se distribuye como parte de este SDK.

Obtener una clave del API. Para ello, tendrá que registrar un proyecto en el API de Google Console y obtener un certificado para firmar su aplicación.

Configurar los permisos y el API Key en el manifest de aplicación.

Añadir el mapa a la aplicación.

Tema 5

Google Maps

Obtener Api Key.

La obtención de una clave para la aplicación requiere varios pasos.

- Recuperar información sobre el certificado de su aplicación.
- Registrar un proyecto en la Consola de Google.
- Agregar el servicio de mapas.
- Solicitar la clave.
- Añadir la clave para la aplicación y comenzar el desarrollo.

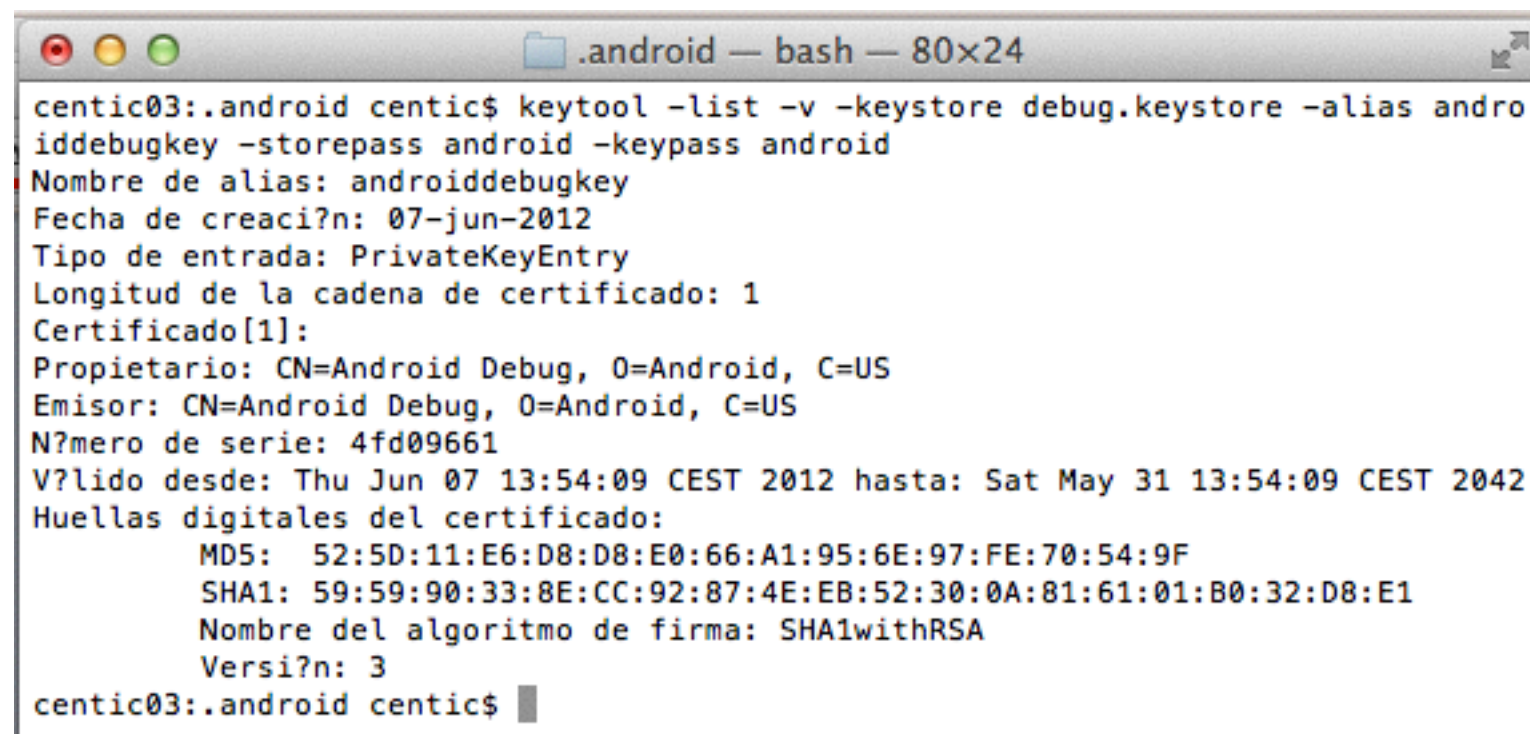
Tema 5

Google Maps

Obtener Api Key.

Recuperar información sobre el certificado de su aplicación.

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



```
.android — bash — 80x24
centic03:~ centic$ keytool -list -v -keystore debug.keystore -alias andro
iddebugkey -storepass android -keypass android
Nombre de alias: androiddebugkey
Fecha de creaci?n: 07-jun-2012
Tipo de entrada: PrivateKeyEntry
Longitud de la cadena de certificado: 1
Certificado[1]:
Propietario: CN=Android Debug, O=Android, C=US
Emisor: CN=Android Debug, O=Android, C=US
N?mero de serie: 4fd09661
V?lido desde: Thu Jun 07 13:54:09 CEST 2012 hasta: Sat May 31 13:54:09 CEST 2042
Huellas digitales del certificado:
    MD5: 52:5D:11:E6:D8:D8:E0:66:A1:95:6E:97:FE:70:54:9F
    SHA1: 59:59:90:33:8E:CC:92:87:4E:EB:52:30:0A:81:61:01:B0:32:D8:E1
    Nombre del algoritmo de firma: SHA1withRSA
    Versi?n: 3
centic03:~ centic$
```

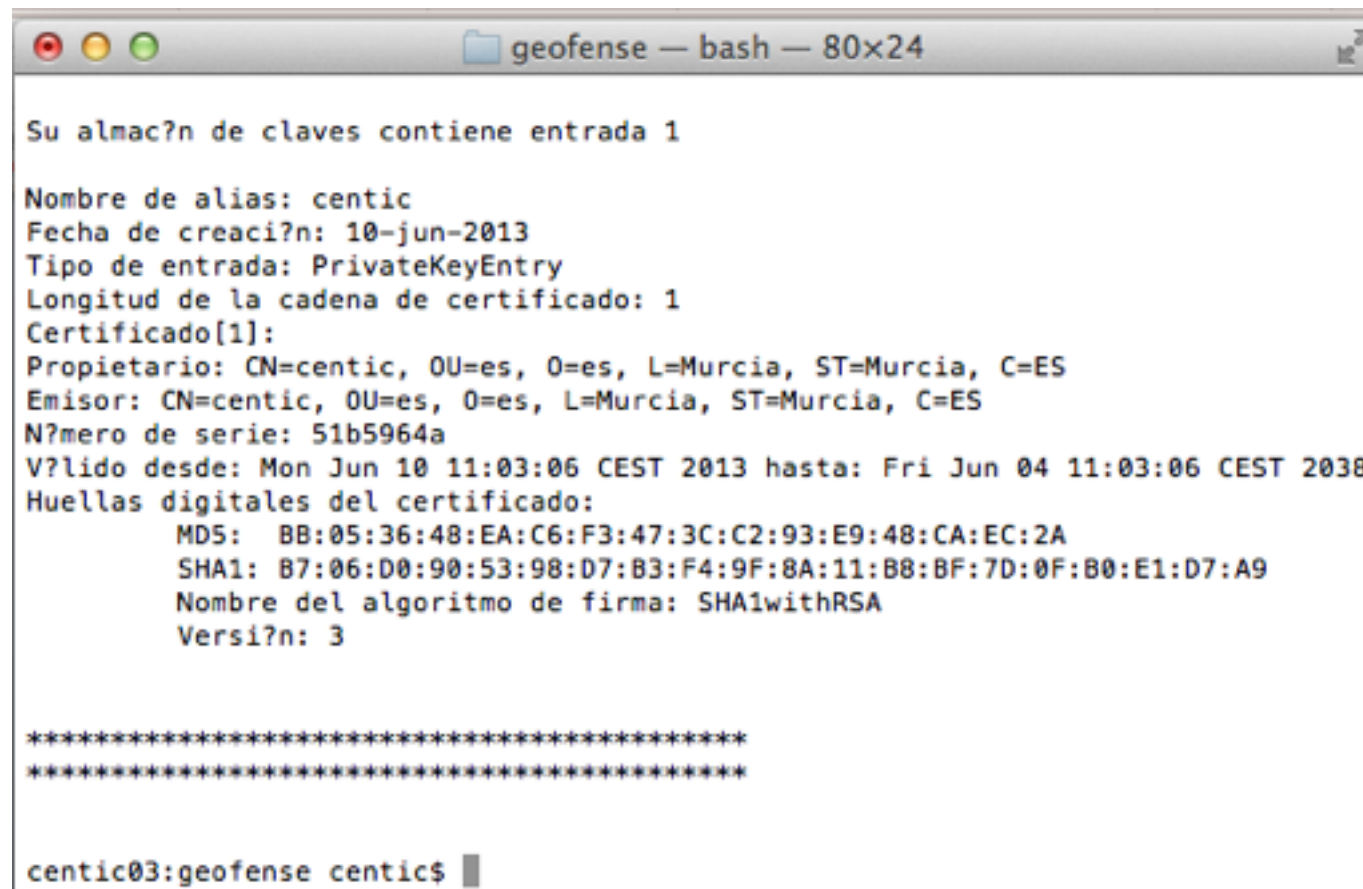
Tema 5

Google Maps

Obtener Api Key.

Recuperar información sobre el certificado de su aplicación.

```
keytool -list -keystore your_keystore_name
```



```
geofense — bash — 80x24

Su almacén de claves contiene entrada 1

Nombre de alias: centic
Fecha de creación: 10-jun-2013
Tipo de entrada: PrivateKeyEntry
Longitud de la cadena de certificado: 1
Certificado[1]:
Propietario: CN=centic, OU=es, O=es, L=Murcia, ST=Murcia, C=ES
Emisor: CN=centic, OU=es, O=es, L=Murcia, ST=Murcia, C=ES
Número de serie: 51b5964a
Válido desde: Mon Jun 10 11:03:06 CEST 2013 hasta: Fri Jun 04 11:03:06 CEST 2038
Huellas digitales del certificado:
    MD5: BB:05:36:48:EA:C6:F3:47:3C:C2:93:E9:48:CA:EC:2A
    SHA1: B7:06:D0:90:53:98:D7:B3:F4:9F:8A:11:B8:BF:7D:0F:B0:E1:D7:A9
Nombre del algoritmo de firma: SHA1withRSA
Versión: 3

*****
*****

centic03:geofense centic$
```

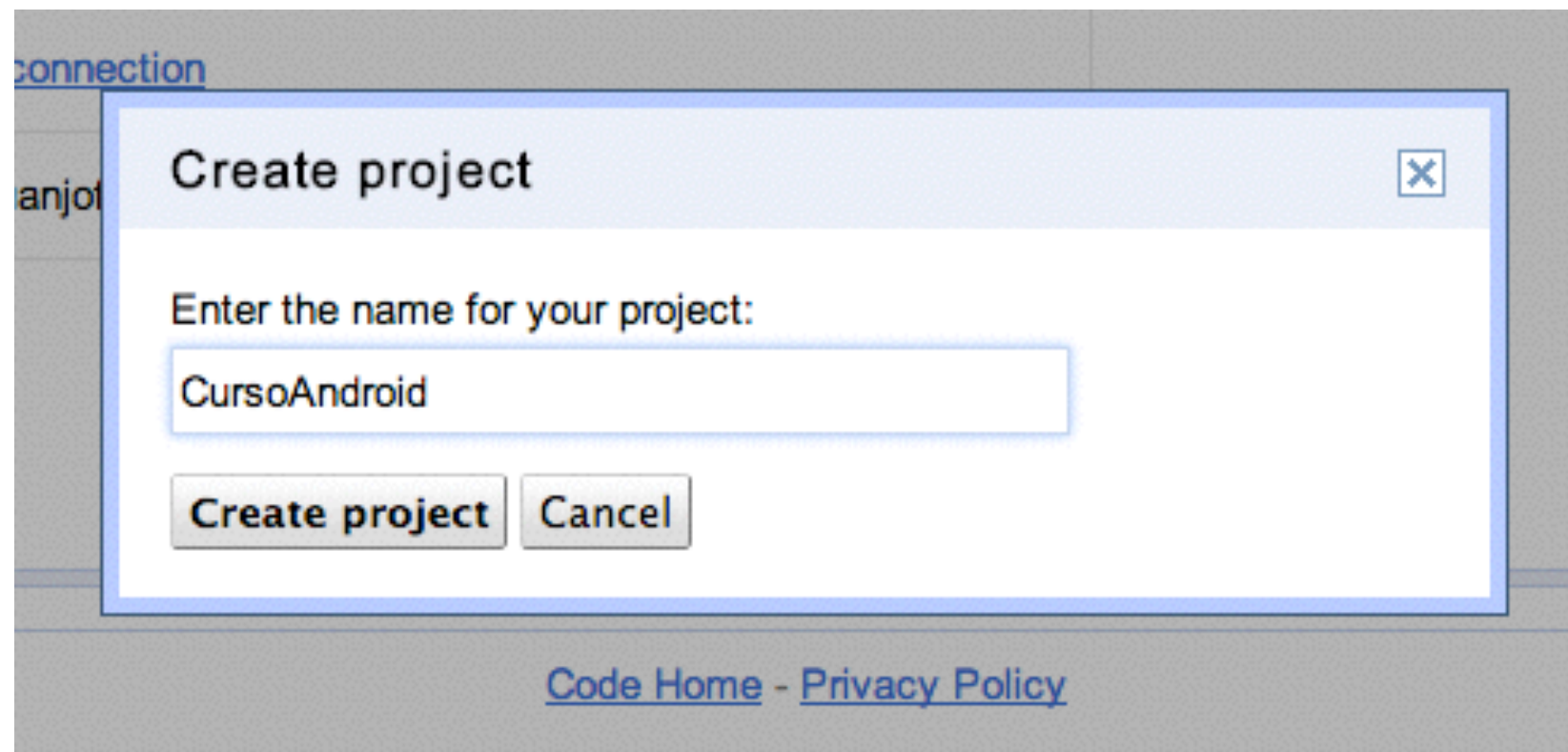
Tema 5

Google Maps

Obtener Api Key.

Crear el proyecto en la consola de Google.

<https://code.google.com/apis/console/>

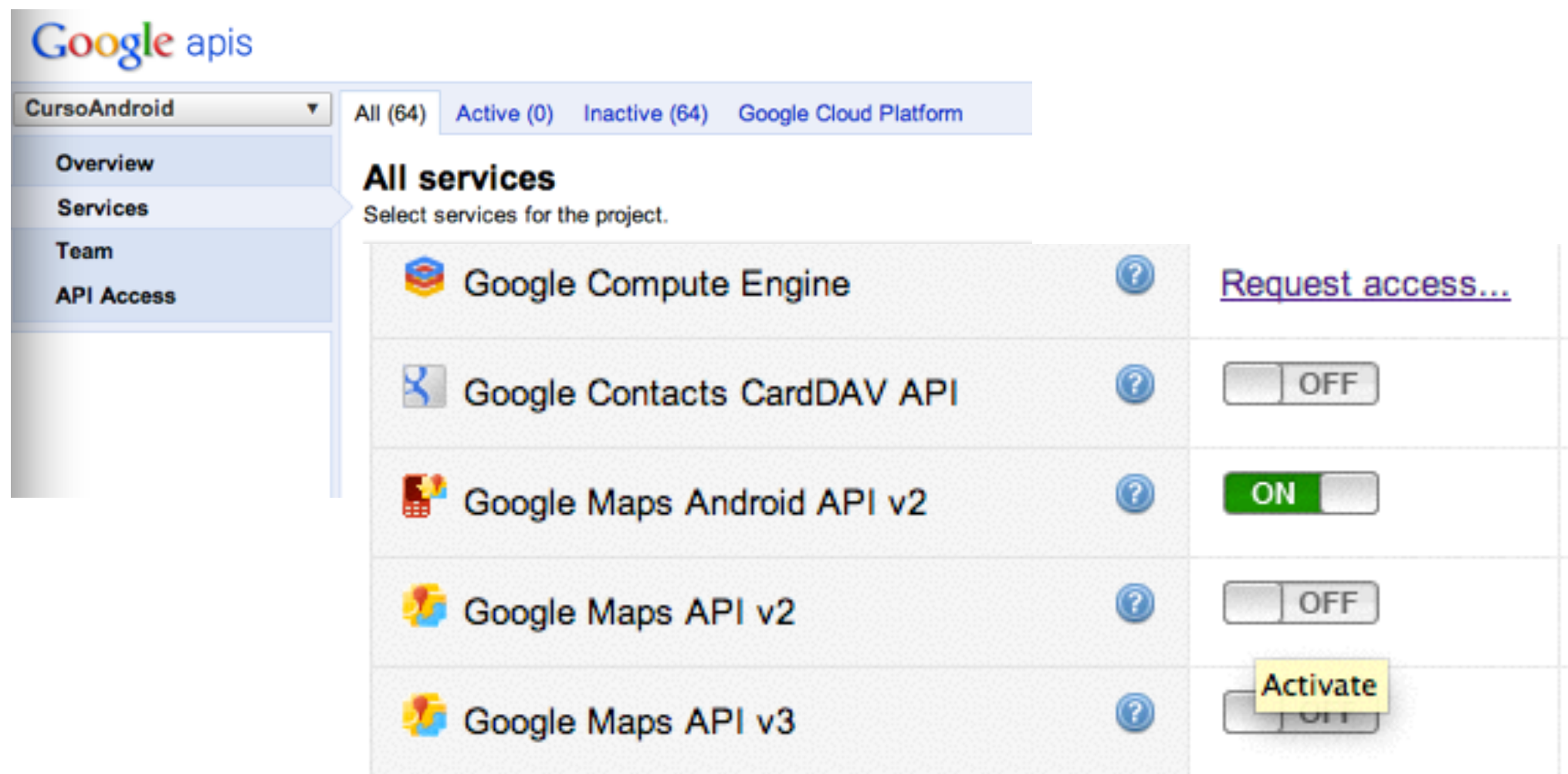


Tema 5

Google Maps

Obtener Api Key.

Activar los servicios necesarios. En este caso Android Maps v2



The screenshot shows the Google APIs console interface. On the left, there is a sidebar with the 'Google apis' logo and a navigation menu with options: 'CursoAndroid', 'Overview', 'Services', 'Team', and 'API Access'. The 'Services' option is selected. The main area displays 'All services' for the project 'CursoAndroid'. It includes filters for 'All (64)', 'Active (0)', and 'Inactive (64)', along with a 'Google Cloud Platform' link. Below the filters, a table lists various services. The 'Google Maps Android API v2' service is highlighted with a yellow box, and a yellow label 'Activate' is placed over its toggle switch. Other services listed include Google Compute Engine, Google Contacts CardDAV API, Google Maps API v2, and Google Maps API v3.

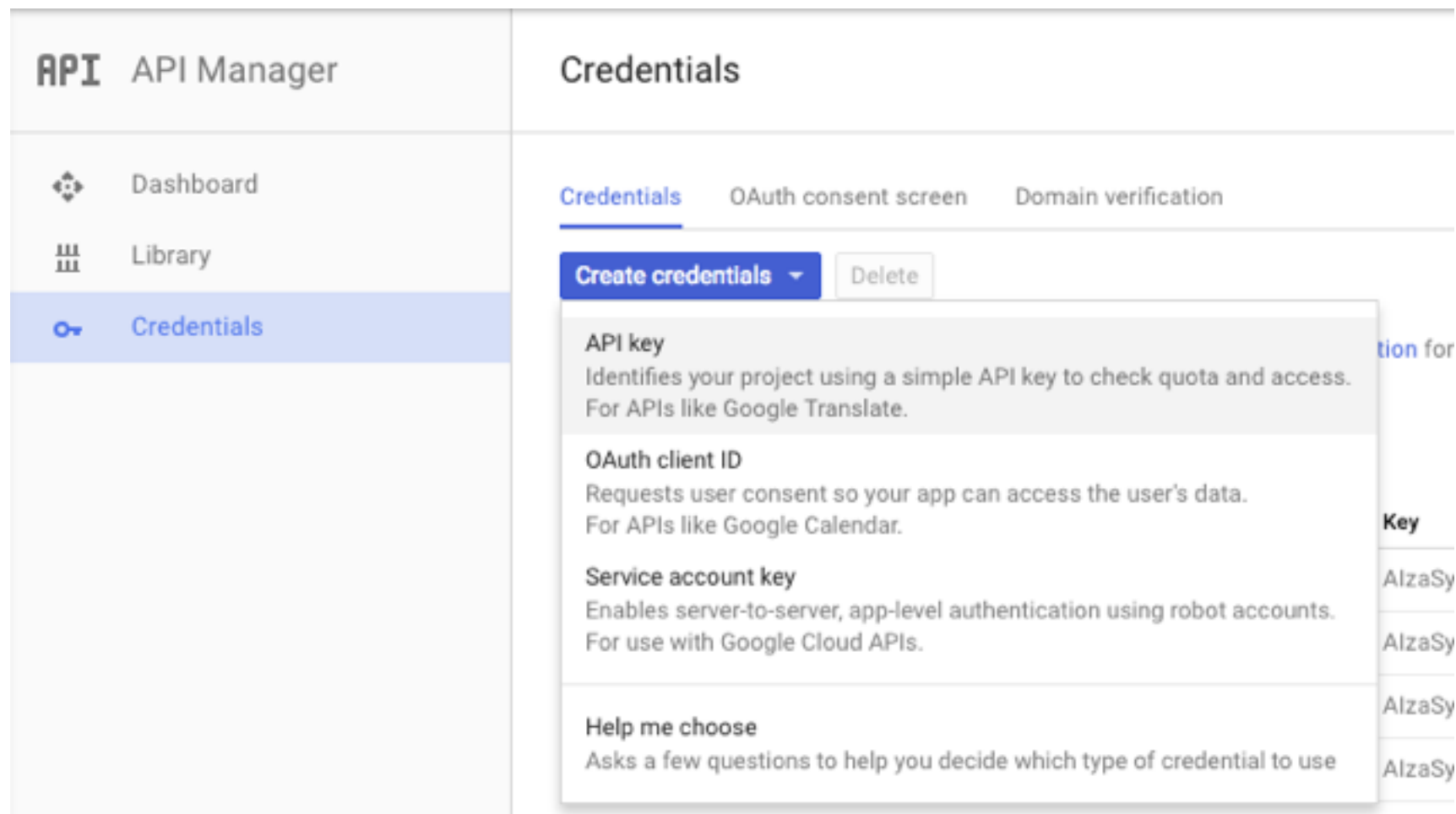
Service Name	Status	Action
Google Compute Engine	?	Request access...
Google Contacts CardDAV API	?	OFF
Google Maps Android API v2	ON	Activate
Google Maps API v2	?	OFF
Google Maps API v3	?	OFF

Tema 5

Google Maps

Obtener Api Key.

Generar el API Key usando la clave obtenida anteriormente y el nombre de paquete de la aplicación.



Tema 5

Google Maps

Añadir Api Key al manifest.

```
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >

    <meta-data android:name="com.google.android.maps.v2.API_KEY"
               android:value="AIzaSyD5wYJ5BSXYF3hebj9z8TkU90K1SWlPhbc" />

    <activity
        android:name="com.cursoandroid.mapssamplev2.MainActivity"
        android:label="@string/app_name" >
        <intent-filter>
```


Tema 5

Google Maps

Añadir permisos para recibir los mapas.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.cursoandroid.mapssamplev2"
    android:versionCode="1"
    android:versionName="1.0" >

    <permission
        android:name="com.cursoandroid.mapssamplev2.permission.MAPS_RECEIVE"
        android:protectionLevel="signature"/>
    <uses-permission android:name="com.cursoandroid.mapssamplev2.permission.MAPS_RECEIVE"/>

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="16" />
```

Tema 5

Google Maps

Añadir permisos para usar GMaps.

`android.permission.INTERNET`

Usado por la API para descargar los mapas desde los servidores de Google Maps.

`android.permission.ACCESS_NETWORK_STATE`

Permite la API para comprobar el estado de la conexión con el fin de determinar si los datos se pueden descargar.

`com.google.android.providers.gsf.permission.READ_GSERVICES`

Permite la API para acceder a servicios basados en la web de Google.

`android.permission.WRITE_EXTERNAL_STORAGE`

Permite la API para almacenar en caché los datos del mapa en la zona de almacenamiento externo del dispositivo.

Tema 5

Google Maps

Añadir permisos para usar GMaps.

Se recomiendan los siguientes permisos, pero se pueden ignorar si la aplicación no necesita acceso a la ubicación actual del usuario.

`android.permission.COARSE_LOCATION`

Permite la API de usar datos de la celda WiFi o móvil (o ambos) para determinar la ubicación del dispositivo.

`android.permission.ACCESS_FINE_LOCATION`

Permite la API para utilizar el Sistema de Posicionamiento Global (GPS) para determinar la ubicación del dispositivo para dentro de un área muy pequeña.

Tema 5

Google Maps

Añadir permisos para usar GMaps.

```
<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"/>
```

Tema 5

Google Maps

Gmaps v2 está construido usando OpenGL 2.

Dado que la versión 2 del API de Google Maps requiere OpenGL ES Versión 2, debemos agregar un elemento `<uses-feature>` que filtre los dispositivos que no lo soportan

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

Tema 5

Google Maps

MapFragment.

MapFragment es una subclase de la clase Fragment, permite colocar un mapa en otro fragment o una activity.

Actúa como un contenedor para el mapa y facilita el acceso al objeto GoogleMap.

Representa un comportamiento o una porción de interfaz de usuario en una actividad.

Se pueden combinar varios fragments en una sola actividad para construir una interfaz de usuario de varios paneles y reutilizar un fragment en múltiples actividades.

MapFragment requiere API 12 o superior aunque podemos acceder a las mismas funciones a través de la clase SupportMapFragment.

Tema 5

Google Maps

MapFragment.

```
<?xml version="1.0" encoding="utf-8"?>
<fragment
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.MapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

```
mMapFragment = MapFragment.newInstance();
FragmentTransaction fragmentTransaction =
    getFragmentManager().beginTransaction();
fragmentTransaction.add(R.id.my_container, mMapFragment);
fragmentTransaction.commit();
```

Tema 5

Google Maps

MapFragment.

Recuperar un MapFragment declarado en XML.

```
private GoogleMap mMap;  
...  
mMap = ((MapFragment) getSupportFragmentManager().findFragmentById(R.id.map)).getMap();
```

```
private void setUpMapIfNeeded() {  
    // Do a null check to confirm that we have not already instantiated the map.  
    if (mMap == null) {  
        mMap = ((MapFragment) getSupportFragmentManager().findFragmentById(R.id.map))  
                .getMap();  
        // Check if we were successful in obtaining the map.  
        if (mMap != null) {  
            // The Map is verified. It is now safe to manipulate the map.  
        }  
    }  
}
```

Tema 5

Google Maps

MapFragment.

Ofrece cuatro tipos de mapas:

Normal

Mapa de carreteras.

Híbrido

Datos de fotografías por satélite con mapas de carreteras..

Satélite

Datos de fotografías de satélite.

Terreno

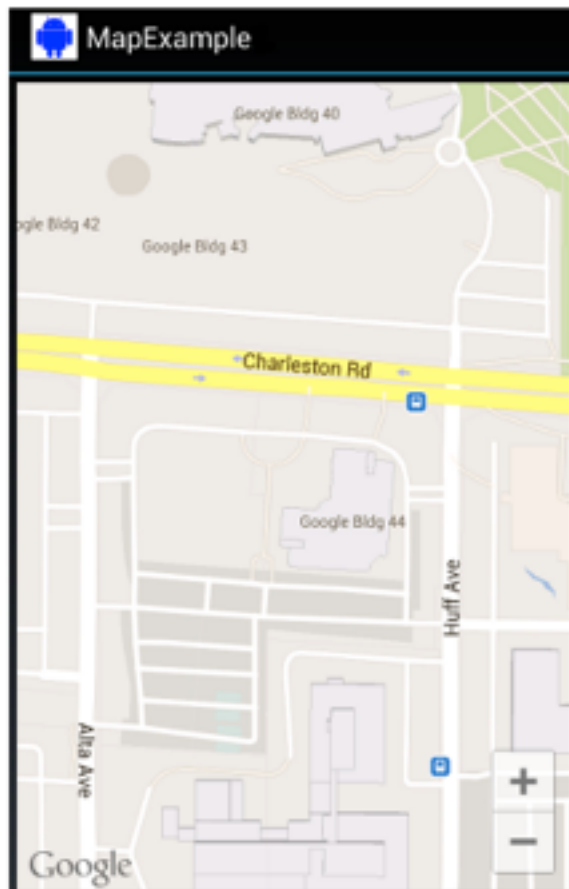
Los datos topográficos. El mapa incluye colores, curvas de nivel y las etiquetas y sombreado perspectiva.

Tema 5

Google Maps

MapFragment.

```
GoogleMap map;  
...  
// Sets the map type to be "hybrid"  
map.setMapType(GoogleMap.MAP_TYPE_HYBRID);
```



Tema 5

Google Maps

MapFragment.

```
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    class="com.google.android.gms.maps.SupportMapFragment"
    map:cameraBearing="112.5"
    map:cameraTargetLat="-33.796923"
    map:cameraTargetLng="150.922433"
    map:cameraTilt="30"
    map:cameraZoom="13"
    map:mapType="normal"
    map:uiCompass="false"
    map:uiRotateGestures="true"
    map:uiScrollGestures="false"
    map:uiTiltGestures="true"
    map:uiZoomControls="false"
    map:uiZoomGestures="true"/>
```

```
options.mapType(GoogleMap.MAP_TYPE_SATELLITE)
        .compassEnabled(false)
        .rotateGesturesEnabled(false)
        .tiltGesturesEnabled(false);
```

Tema 5

Publicación

Publicar una aplicación en Google PLAY.

Abrir la consola de Google Play.


<https://play.google.com/apps/publish/>.

Si no tenemos cuenta de desarrollador deberemos pagar una cuota de \$25 mediante Google Wallet.

Si somos una empresa es mejor crear una cuenta de gmail asociada a la empresa en lugar de usar una cuenta personal.

Tema 5

Publicación

 Google play | Developer Console

≡ All applications



Financial reports

Settings

Announcements

ALL APPLICATIONS

+ Add new application

APP NAME	PRICE	ACTIVE / TOTAL INSTALLS	AVG. RATING / TOTAL #	CRASHES & ANRS	LAST UPDATE	STATUS
 Animal Translator 1.1	Free	12,078 / 185,410	★ 3.29 / 566		Apr 21, 2010	Published
 Earthquake! 3.8	Free	71,426 / 785,829	★ 4.15 / 6,212		Feb 1, 2013	Published

Page 1 of 1

Tema 5

Publicación

Detalles del desarrollador.

ACCOUNT DETAILS Saved

DEVELOPER PROFILE

Developer name *

Company or developer name...

The developer name will appear to users under the name of your application.

Email address *

your.address@gmail.com

Website

http://your.site.com

Phone Number *

+1-650-555-1212

Include plus sign, country code and area code.
For example, +1-650-253-0000.


Email updates

☐ I'd like to get occasional emails about development and Google Play opportunities.

Tema 5

Publicación

Detalles de la aplicación.

 EARTHQUAKE! – com.radioactiveyak.earthquake Published

STORE LISTING Saved Fields marked with * need to be filled before publishing.

PRODUCT DETAILS

English (United States) Add translations

Title *
English (United States)
11 of 30 characters

Description *
English (United States)

Get a head start on the apocalypse with Earthquake!

Last 24hrs of earthquakes, with damage and rumble areas shown on an interactive map. Features notifications and vibration to indicate quake magnitude, and a dynamic widget.

Now optimized for tablets!


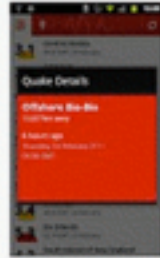
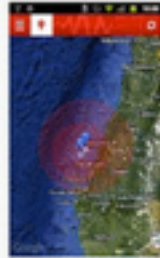

Promo text
English (United States)

Recent changes
English (United States)


GRAPHIC ASSETS

If you haven't added localized graphics for each language, graphics for your default language will be used.
[Learn more about graphic assets.](#)


Screenshots *
Default – English (United States)
320 x 480 or 480 x 800 or 480 x 854 or 1280 x 720 or 1280 x 800. JPG or 24-bit PNG (no alpha)
Drag to reorder. At least two are required.



High-res icon *
Default – English (United States)
512 x 512
32-bit PNG (with alpha)



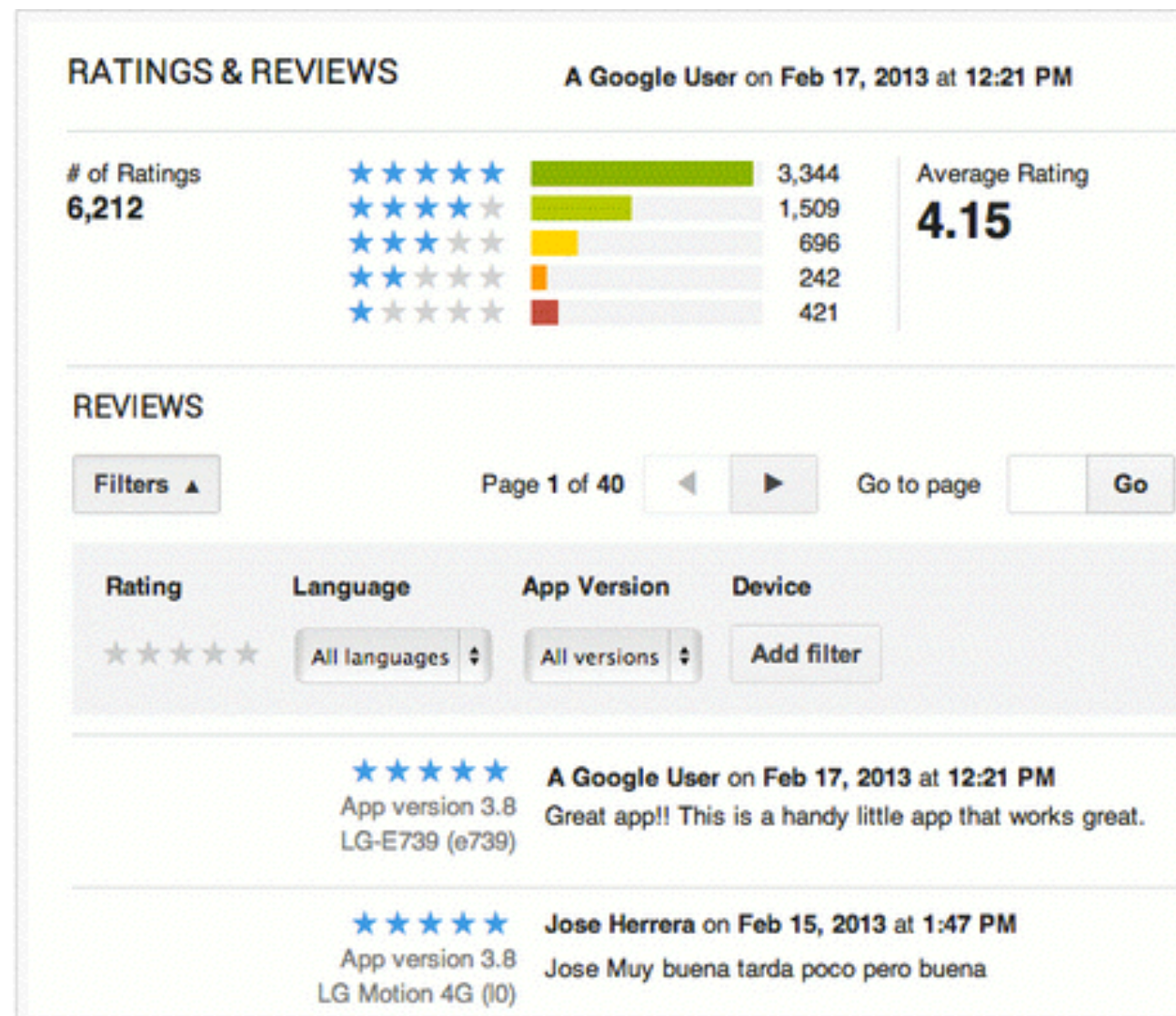
Feature Graphic
Default – English (United States)
1024 w x 500 h
JPG or 24-bit PNG (no alpha)



Tema 5

Publicación

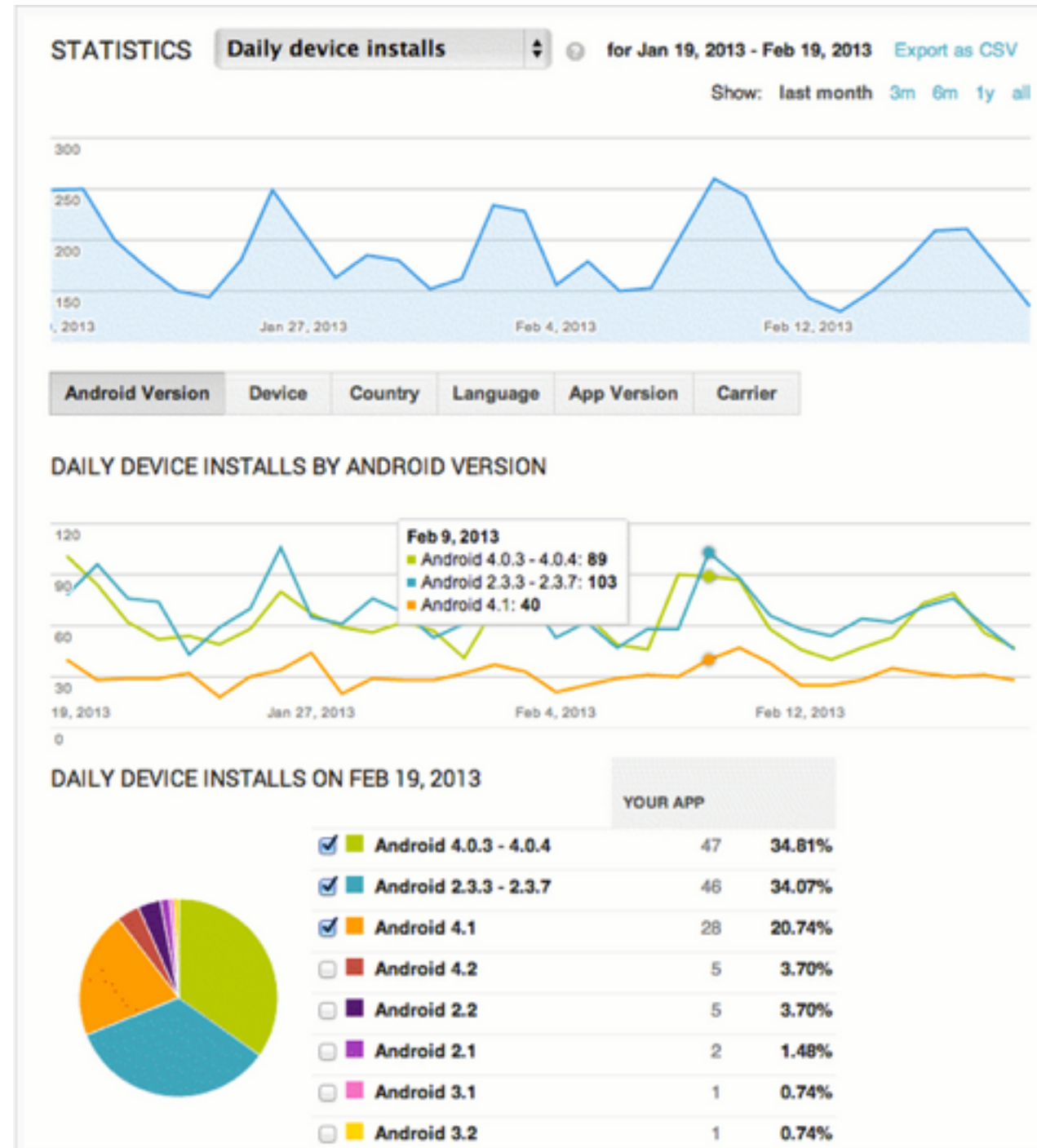
Comentarios y puntuaciones.



Tema 5

Publicación

Estadísticas.



Tema 5

Publicación

Check List (L10N).

Identificar los idiomas y regiones a los que queremos optar.

Adaptar la aplicación a las distintas regiones.

Aplicar layouts alternativos cuando sea necesario.

Soporte RTL

Soporte de fechas, número y monedas.

Disponer de los ficheros strings.xml para cada idioma

Disponer de las imágenes que necesiten ser adaptadas a cada idioma.

Tema 5

Publicación

Check List (L10N).

Identificar los idiomas y regiones a los que queremos optar.

Adaptar la aplicación a las distintas regiones.

Aplicar layouts alternativos cuando sea necesario.

Soporte RTL

Soporte de fechas, número y monedas.

Disponer de los ficheros strings.xml para cada idioma

Disponer de las imágenes que necesiten ser adaptadas a cada idioma.

Alcohol, Tobacco and Drugs

- Apps that contain illegal content are not permitted in Google Play. Apps that include references to drugs, alcohol or tobacco products must be rated **medium maturity** or above. Apps that primarily focus on the consumption or sale of drugs, alcohol or tobacco must be rated **high maturity**.

Gambling

- Apps that facilitate real gambling are not permitted in Google Play. Apps with gambling themes or simulated gambling must be rated **medium maturity** or **high maturity**.

Hate

- Hate speech is not allowed in Google Play. Apps that contain inflammatory content must be rated **medium maturity** or **high maturity**.

Location

- Applications rated **everyone** must not ask users for their location. Apps that access location data must be rated **low maturity** or above. Apps that publish or share users' location with others must be rated **medium maturity** or **high maturity**.

Profanity and Crude Humor

- Apps that include profanity or crude humor should be rated **medium maturity** or **high maturity**.

Sexual and Suggestive Content

- Pornography is not permitted in Google Play. Apps that include suggestive or sexual references must be rated **medium maturity** or **high maturity**. Apps that focus on suggestive or sexual references must be rated **high maturity**.

User Generated Content and User to User Communication

- Apps rated **everyone** must not host any user generated content or enable communication between users. Apps that focus on allowing users to find each other and communicate should be rated **medium maturity** or **high maturity**.

Violence

- Apps that contain mild cartoon or fantasy violence must be rated **low maturity** or above. Apps that contain realistic or intense fantasy violence must be rated **medium maturity** or **high maturity**, and those that contain graphic violence must be rated **high maturity**. Gratuitous real violence is not allowed in Google Play.

Tema 5

Publicación

Check List.

Añadir los elementos de pago si los hubiese.

Añadir los precios para cada país de los elementos anteriores.

Preparar los gráficos promocionales.

Firmar y subir el APK

Publicar en Alfa o Beta

Publicar en Play