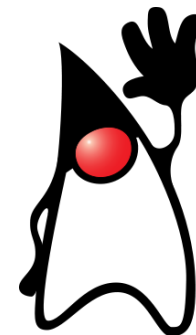




Android Con Java



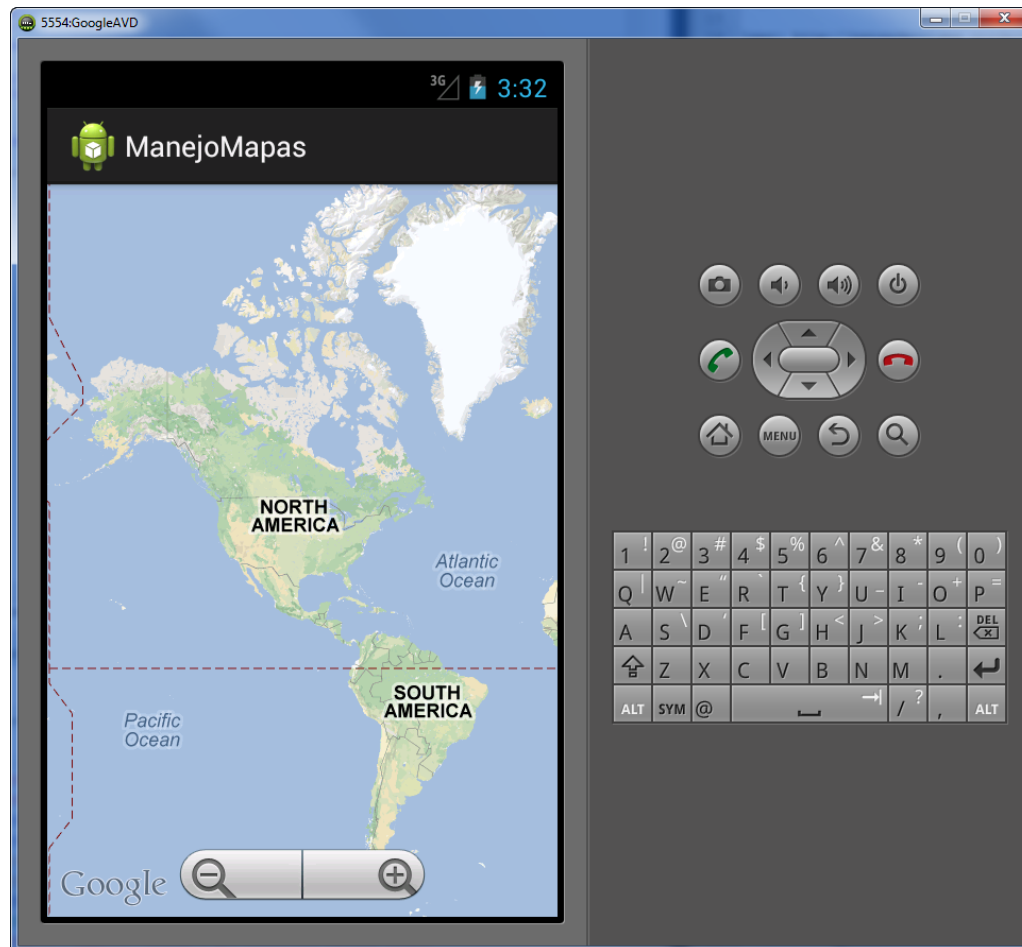
Ejercicio 24



Manejo de Mapas
en Android

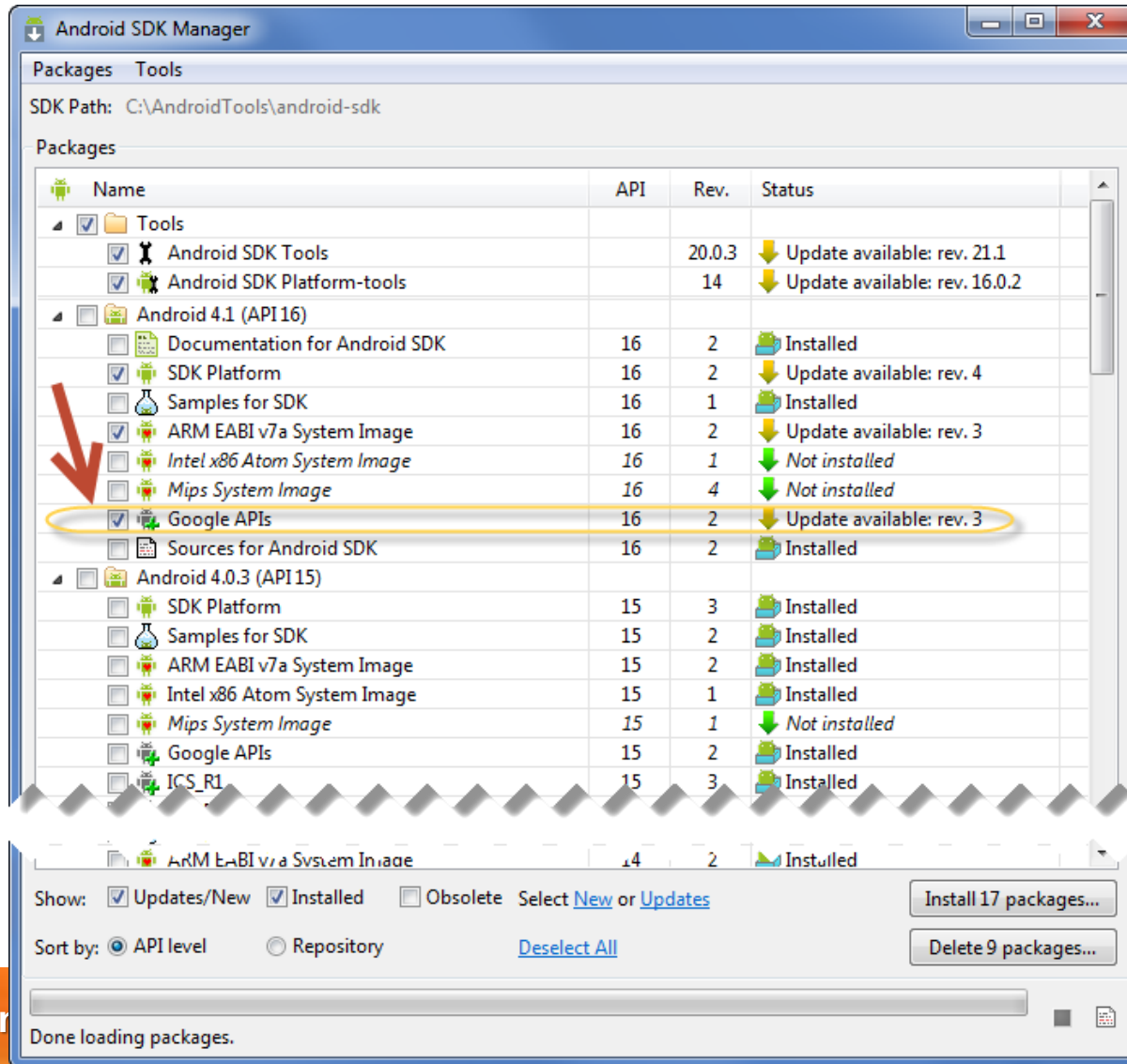
Objetivo del Ejercicio

- El objetivo del ejercicio poner en práctica el Manejo de Mapas en Android.



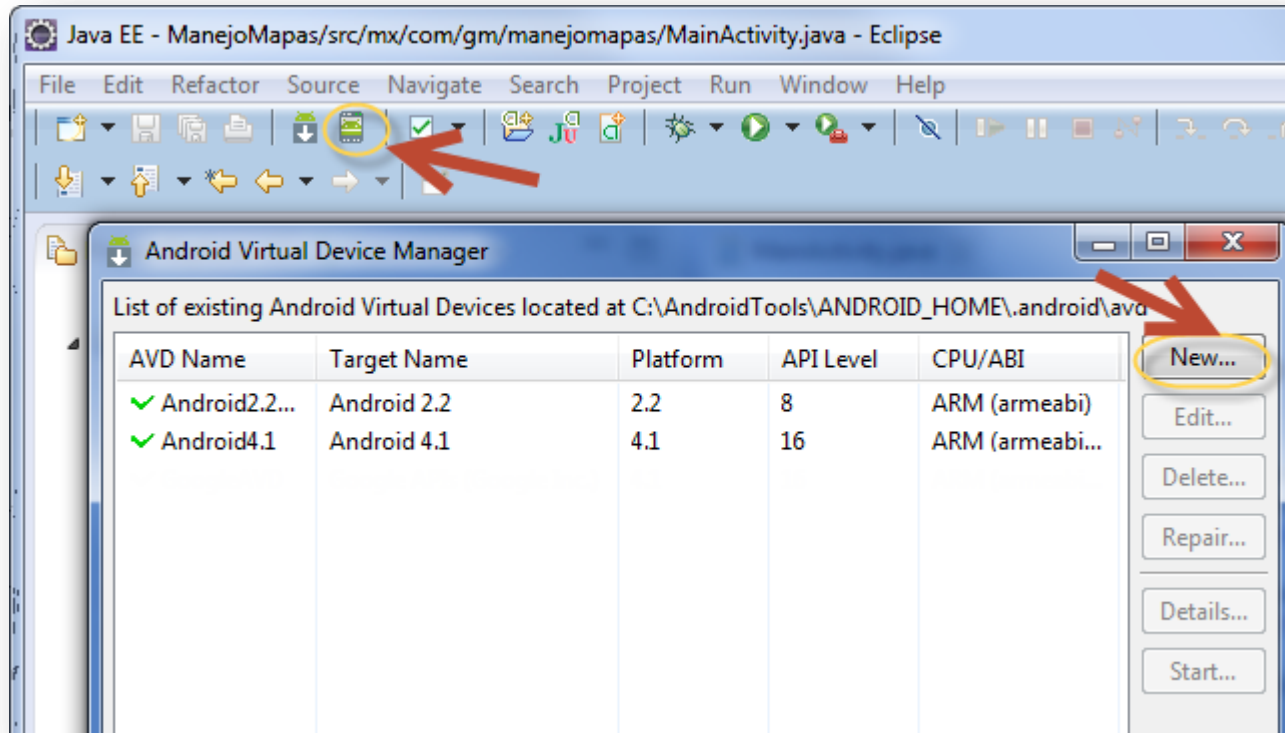
Paso 1. Instalar alguna versión del SDK de Google

Abrir el SDK y asegurarnos de tener alguna versión del SDK de Google:



Paso 2. Crear un AVD con el API de Google

Crear un nuevo emulador basado en la versión del API de Google:



Paso 2. Crear un AVD con el API de Google (cont)

Crear un nuevo emulador basado en la versión del API de Google:

Create new Android Virtual Device (AVD)

Name: GoogleAVD

Target: Google APIs (Google Inc.) - API Level 16

CPU/ABI: ARM (armeabi-v7a)

SD Card:

- ☒ Size: 500 MiB
- ☐ File: Browse...

Snapshot: ☒ Enabled

Skin:

- ☒ Built-in: Default (WVGA800)
- ☐ Resolution: x

Hardware:

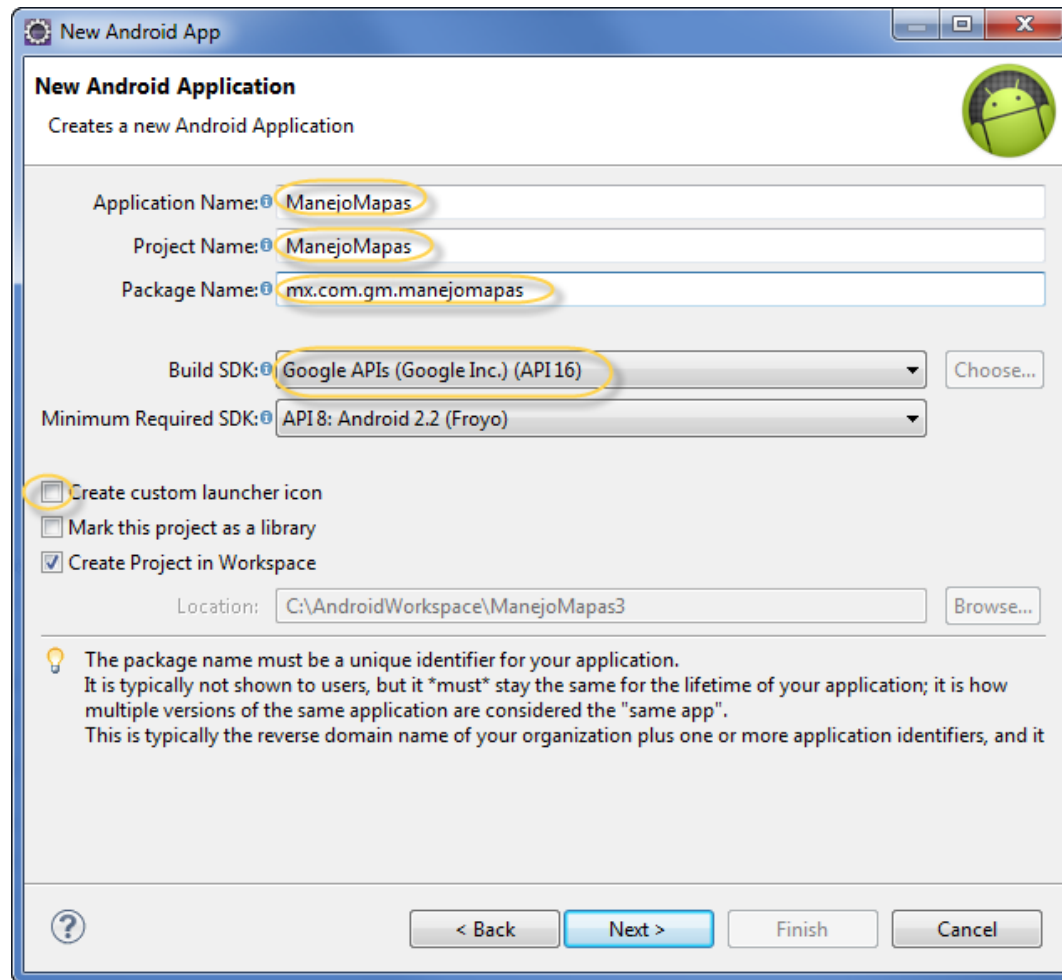
Property	Value
Abstracted LCD density	240
Max VM application heap size	48
Device ram size	512

☐ Override the existing AVD with the same name

Create AVD Cancel

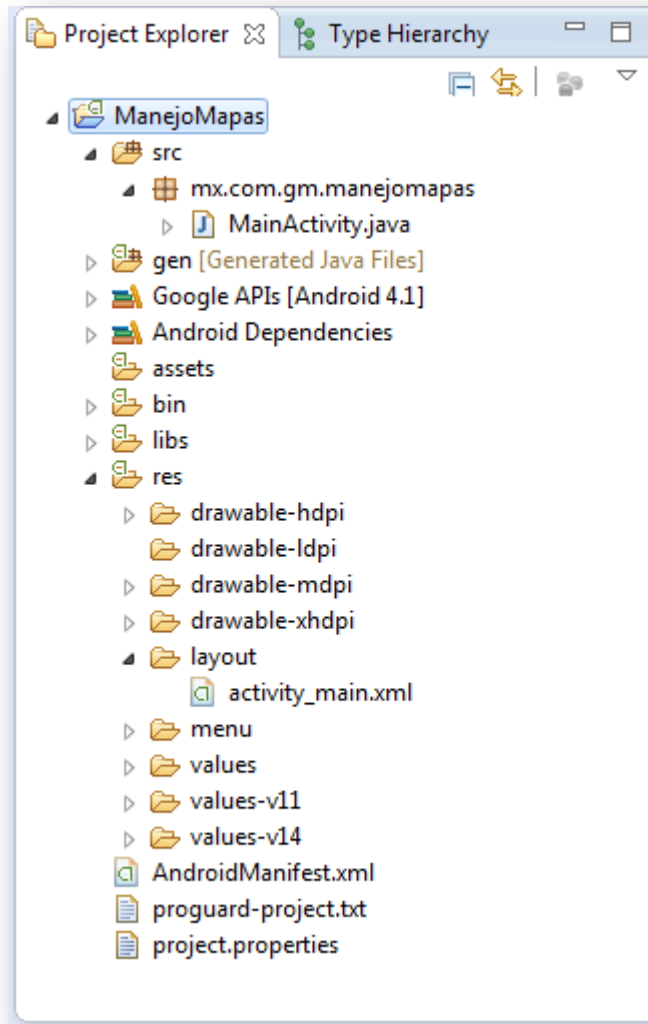
Paso 3. Crear el proyecto ManejoMapas

Creamos el proyecto ManejoMapas, debemos seleccionar el API de Google del AVD:



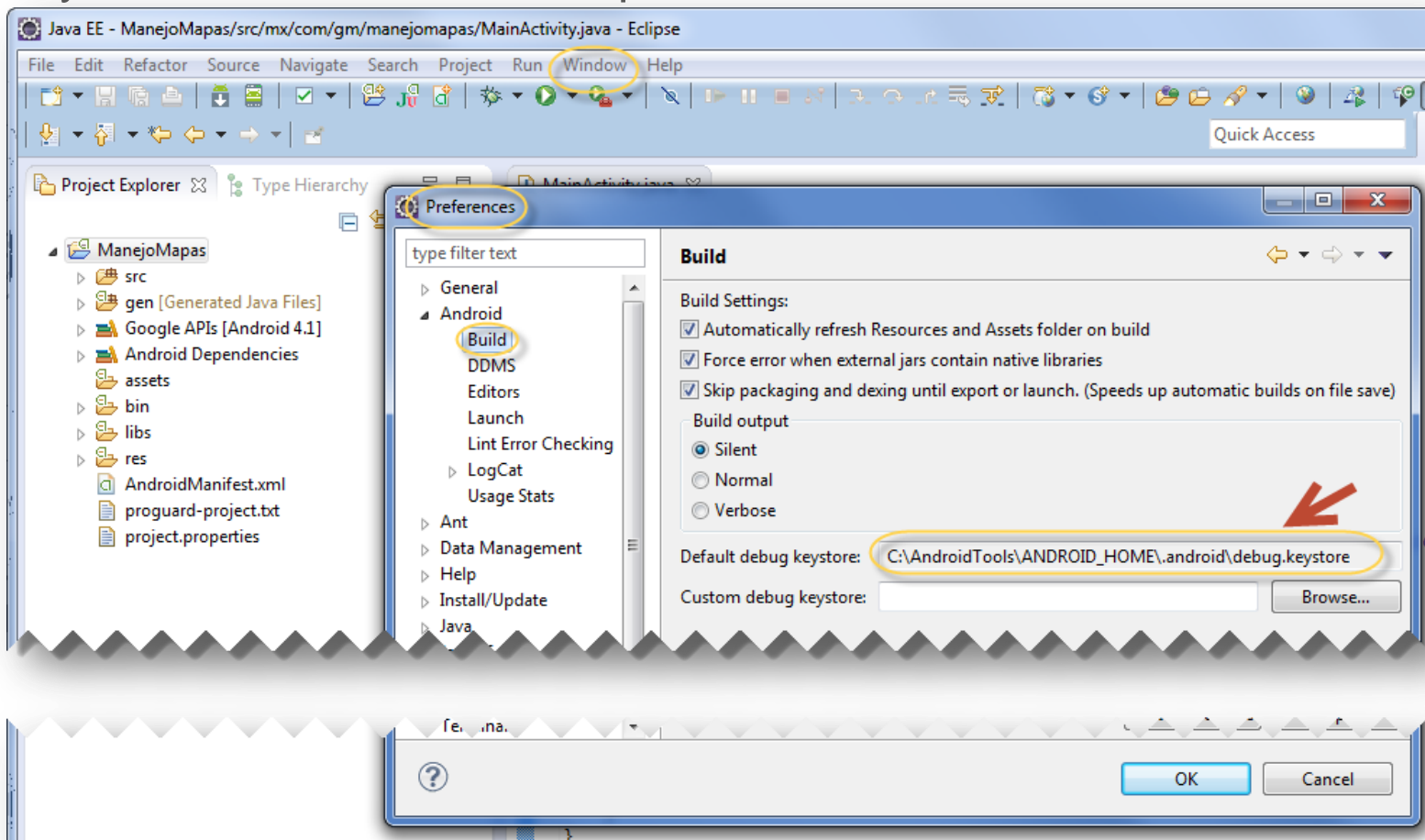
Paso 3. Crear el proyecto ManejoMapas (cont)

Creamos el proyecto ManejoMapas:



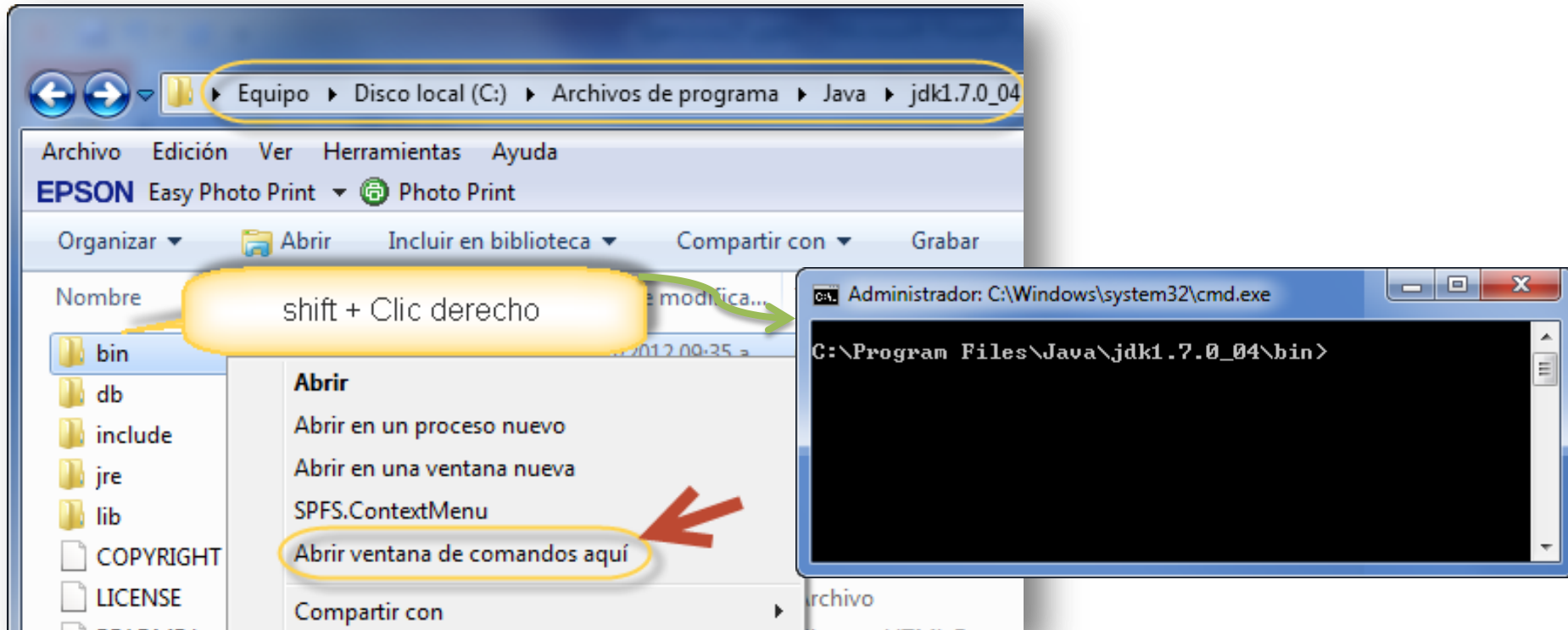
Paso 4. Obtener API Google Maps

Para trabajar Google Maps necesitamos generar un API que valide el uso de los mapas. En primer lugar copiamos la ruta donde se encuentre el keystore de Android. En Eclipse ir a Window->Preferences:



Paso 4. Obtener API Google Maps (cont)

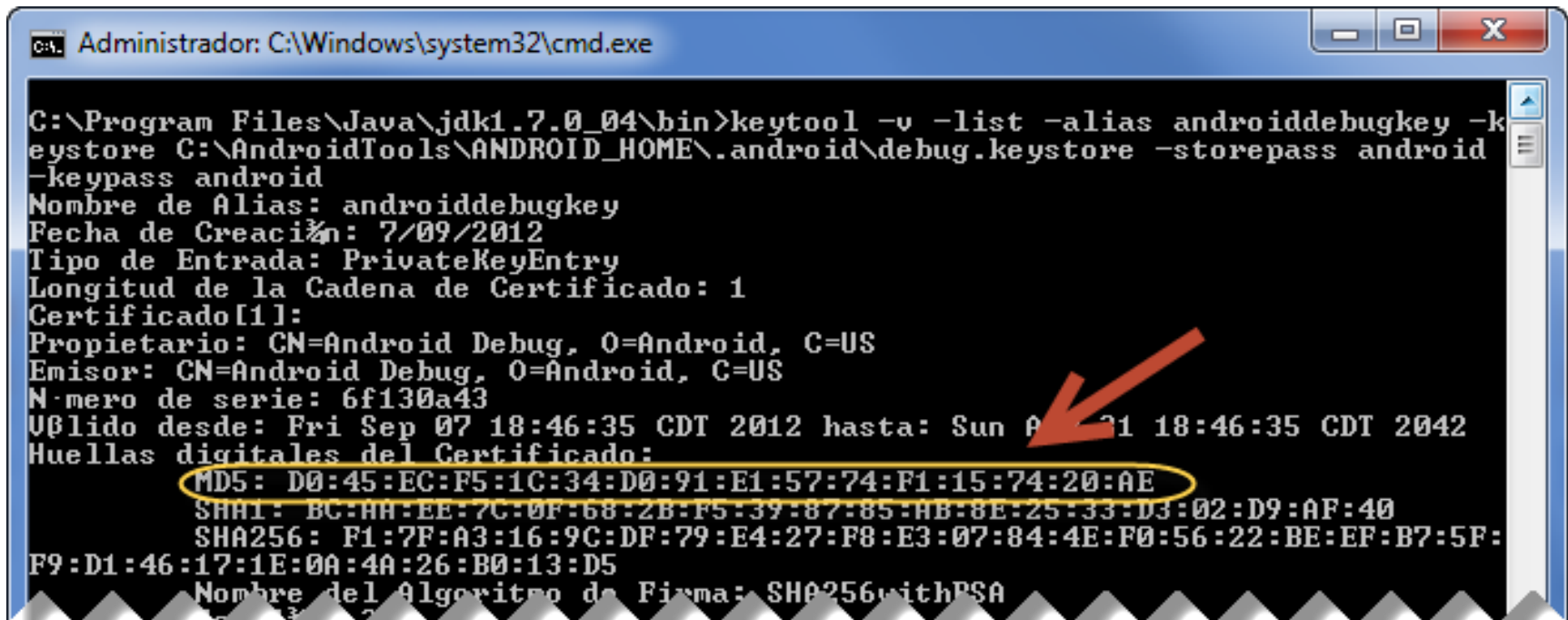
Ahora generamos el MD5 fingerprint de nuestro equipo utilizando el comando de Java. El comando se encuentra dentro de la instalación de Java, ej: C:\Program Files\Java\jdk1.7.0_04\bin. Abrimos una consola de ms-dos y ejecutamos el siguiente comando:



Paso 4. Obtener API Google Maps (cont)

Ahora generamos el MD5 fingerprint ejecutamos el siguiente comando:

```
keytool -v -list -alias androiddebugkey -keystore  
C:\AndroidTools\ANDROID_HOME\.android\debug.keystore -storepass  
android -keypass android
```

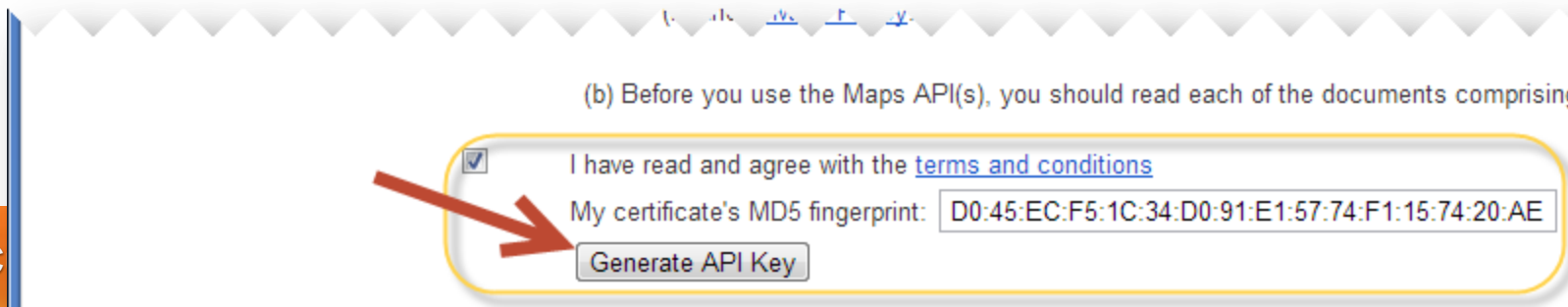


```
Administrador: C:\Windows\system32\cmd.exe  
  
C:\Program Files\Java\jdk1.7.0_04\bin>keytool -v -list -alias androiddebugkey -keystore C:\AndroidTools\ANDROID_HOME\.android\debug.keystore -storepass android -keypass android  
Nombre de Alias: androiddebugkey  
Fecha de Creación: 7/09/2012  
Tipo 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  
Número de serie: 6f130a43  
Válido desde: Fri Sep 07 18:46:35 CDT 2012 hasta: Sun Aug 21 18:46:35 CDT 2042  
Huellas digitales del Certificado:  
MD5: D0:45:EC:F5:1C:34:D0:91:E1:57:74:F1:15:74:20:AE  
SHA1: BC:AA:EE:7C:0F:68:2B:F5:39:87:85:AB:8E:25:33:D3:02:D9:AF:40  
SHA256: F1:7F:A3:16:9C:DF:79:E4:27:F8:E3:07:84:4E:F0:56:22:BE:EF:B7:5F:F9:D1:46:17:1E:0A:4A:26:B0:13:D5  
Nombre del Algoritmo de Firma: SHA256withRSA
```

Paso 4. Obtener API Google Maps (cont)

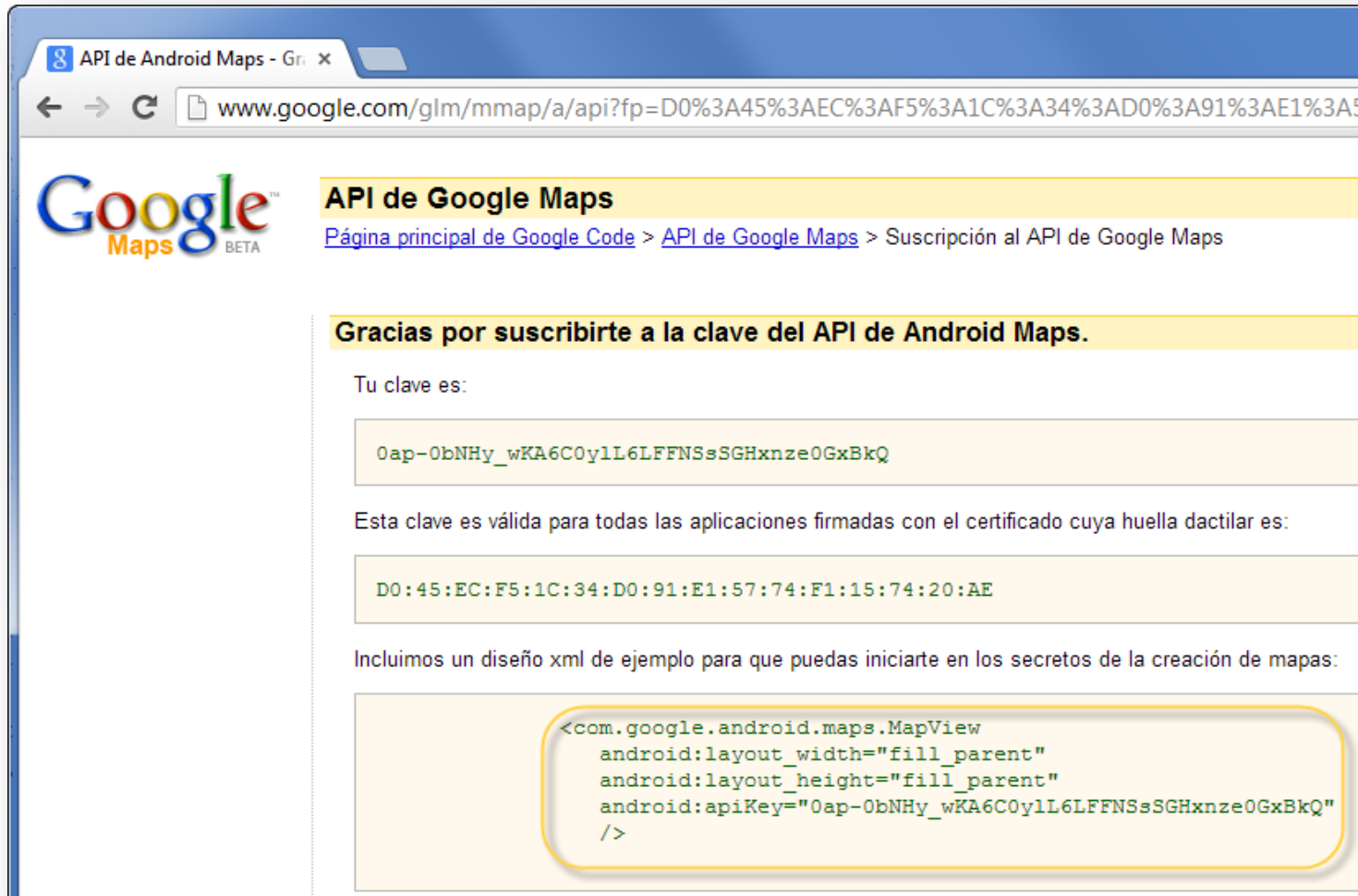
Obtener el API de Google para validar las aplicaciones que utilizan Mapas. Ir a la siguiente página:

<https://developers.google.com/maps/documentation/android/v1/maps-api-signup?hl=es>



Paso 4. Obtener API Google Maps (cont)

Debemos tener una cuenta de Gmail, hacemos login y obtendremos nuestra clave para el uso de Mapas de Google para aplicaciones Android:



The screenshot shows a web browser window with the URL `www.google.com/glm/mmap/a/api?fp=D0%3A45%3AEC%3AF5%3A1C%3A34%3AD0%3A91%3AE1%3A5`. The page title is "API de Android Maps - Google". The Google Maps logo is visible on the left. The main heading is "API de Google Maps". Below it, there is a breadcrumb trail: "Página principal de Google Code > API de Google Maps > Suscripción al API de Google Maps". A yellow banner states "Gracias por suscribirte a la clave del API de Android Maps." Below this, it says "Tu clave es:" followed by a text box containing the API key: `0ap-0bNH_ywKA6C0y1L6LFFNSsSGHxnze0GxBkQ`. Then, it says "Esta clave es válida para todas las aplicaciones firmadas con el certificado cuya huella dactilar es:" followed by a text box containing the fingerprint: `D0:45:EC:F5:1C:34:D0:91:E1:57:74:F1:15:74:20:AE`. Finally, it says "Incluimos un diseño xml de ejemplo para que puedas iniciarte en los secretos de la creación de mapas:" followed by a code block containing an XML snippet for a MapView.

API de Google Maps

[Página principal de Google Code](#) > [API de Google Maps](#) > Suscripción al API de Google Maps

Gracias por suscribirte a la clave del API de Android Maps.

Tu clave es:

```
0ap-0bNH_ywKA6C0y1L6LFFNSsSGHxnze0GxBkQ
```

Esta clave es válida para todas las aplicaciones firmadas con el certificado cuya huella dactilar es:

```
D0:45:EC:F5:1C:34:D0:91:E1:57:74:F1:15:74:20:AE
```

Incluimos un diseño xml de ejemplo para que puedas iniciarte en los secretos de la creación de mapas:

```
<com.google.android.maps.MapView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:apiKey="0ap-0bNH_ywKA6C0y1L6LFFNSsSGHxnze0GxBkQ"
/>
```

Paso 5. Modificar la actividad activity_main.xml

Modificamos la actividad activity_main.xml. Utilizamos el código que nos generó el API de Google y agregamos un id e indicamos que podamos hacer clic sobre el mapa:

```
<RelativeLayout 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" >

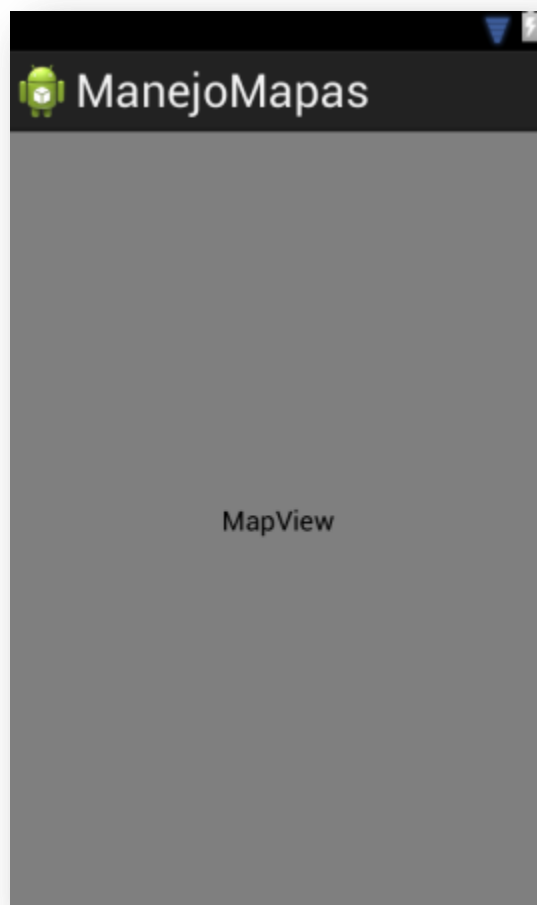
    <com.google.android.maps.MapView
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:id="@+id/mapa"
        android:clickable="true"
        android:apiKey="0ap-0bNH_ywKA6C0yLL6LFFNSsSGHxnze0GxBkQ"
    />

</RelativeLayout>
```



Paso 5. Modificar la actividad activity_main.xml (cont)

El resultado debe ser similar al siguiente:



Paso 6. Modificamos la clase MainActivity.java

Modificamos la clase MainActivity.java. Esta clase debe extender de la clase MapActivity:

```
package mx.com.gm.manejomapas;

import android.os.Bundle;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapView;

public class MainActivity extends MapActivity {

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

        MapView map = (MapView) findViewById(R.id.mapa);
        map.setBuiltInZoomControls(true);
    }

    @Override
    protected boolean isRouteDisplayed() {
        // TODO Auto-generated method stub
        return false;
    }
}
```


Paso 7. Agregamos el permiso

Modificamos el archivo de manifiesto, para agregar los permisos respectivos e indicamos que vamos a utilizar la librería de mapas de android. El archivo queda como sigue:

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

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="15" />
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <uses-library android:name="com.google.android.maps"/>
        <activity
            android:name=".MainActivity"
            android:label="@string/title_activity_main" >

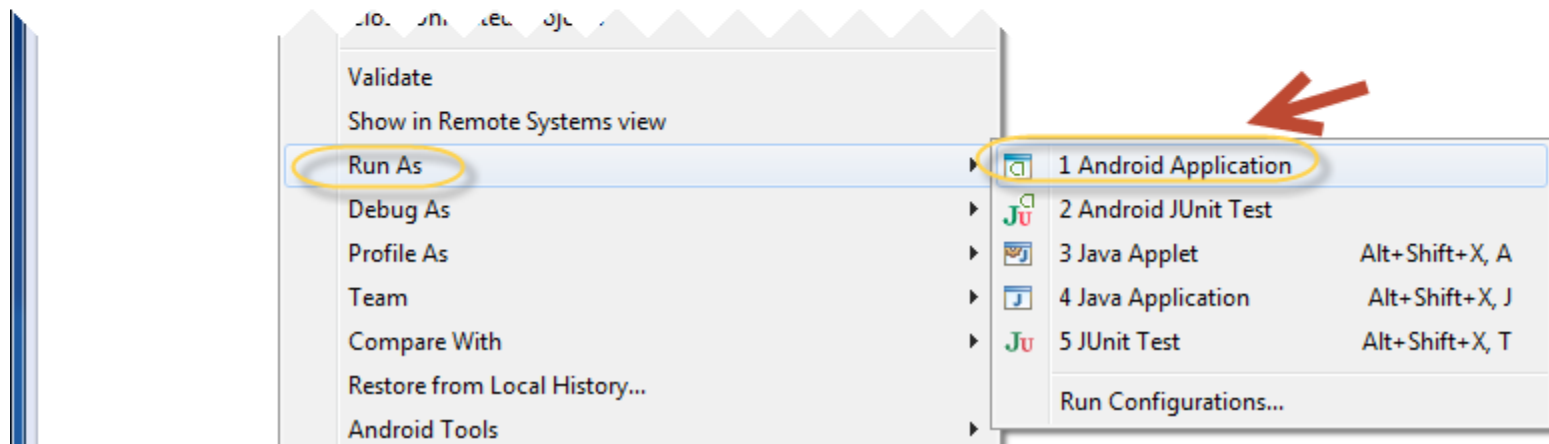
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

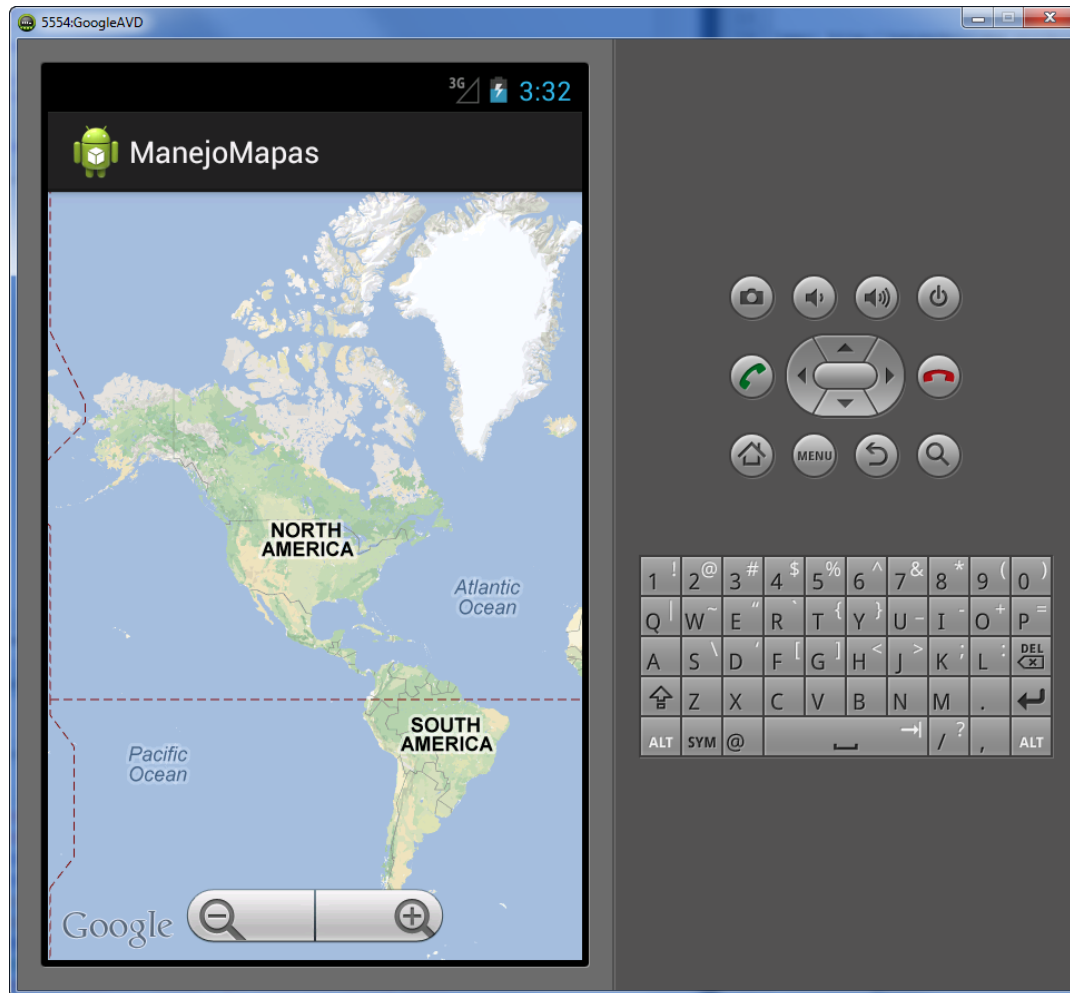
Paso 8. Ejecutamos el proyecto

Ejecutamos el proyecto ManejoMapas utilizando el AVD de Google:



Paso 8. Ejecutamos el proyecto (cont)

Observamos que si utilizamos un emulador que soporte el API de Google y con la última versión podemos visualizar el siguiente resultado:





Conclusión

Con este ejercicio hemos visto cómo configurar el uso de mapas, y cómo implementarlo directamente en nuestra aplicación Android.

En el siguiente ejercicio veremos cómo podemos interactuar con el mapa y ubicar alguna coordenada emulando un GPS.



www.globalmentoring.com.mx

Pasión por la tecnología Java

Experiencia y Conocimiento para tu vida