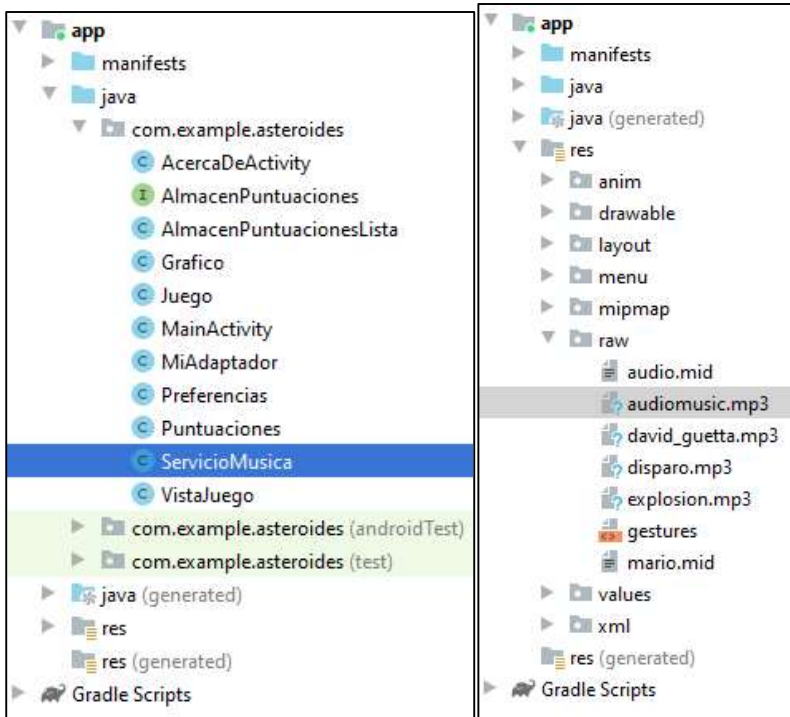


## PRACTICA 4: ASTEROIDES XIII

### Parte I: Uso del servicio de música en Asteroides

**Paso 1.** Copia la clase `ServicioMusica` realizada para la práctica 15 de la UF2 al proyecto Asteroides. También se debe de copiar el mp3 usado en el proyecto anterior, con el nombre `audiomusic.mp3`.



**Paso 2.** Corrige los errores que hayan aparecido para adaptarla al nuevo proyecto.

```
public class ServicioMusica extends Service {  
  
    private static final int ID_NOTIFICATION_CREAR = 1;  
    MediaPlayer reproductor;  
  
    @Override  
    public void onCreate() {  
        Toast.makeText( context: this, text: "Servicio creado", Toast.LENGTH_SHORT).show();  
        reproductor = MediaPlayer.create( context: this, R.raw.audio);  
    }  
}
```



```

public class ServicioMusica extends Service {

    private static final int ID_NOTIFICATION_CREAR = 1;
    MediaPlayer reproductor;

    @Override
    public void onCreate() {
        Toast.makeText( context: this, text: "Servicio creado", Toast.LENGTH_SHORT).show();
        reproductor = MediaPlayer.create( context: this, R.raw.audiomusic);
    }
}

```

**Paso 3.** Cuando se visualiza los detalles de la notificación se podía lanzar la actividad **ActividadPrincipal**. Modifica el código para que se lance la actividad **Asteroides**.

```

@Override
public int onStartCommand(Intent intenc, int flags, int idArranque) {
    Toast.makeText( context: this, text: "Servicio arrancado " + idArranque, Toast.LENGTH_SHORT).show();
    reproductor.start();

    NotificationManager notificationManager =
        (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
    String CHANNEL_ID = "my_channel_01";

    if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {
        CharSequence name = "my_channel";
        String Description = "This is my channel";
        int importance = NotificationManager.IMPORTANCE_HIGH;
        NotificationChannel mChannel = new NotificationChannel(CHANNEL_ID, name, importance);
        mChannel.setDescription(Description);
        mChannel.enableLights(true);
        mChannel.setLightColor(Color.RED);
        mChannel.enableVibration(true);
        mChannel.setVibrationPattern(new long[]{100, 200, 300, 400, 500, 400, 300, 200, 400});
        mChannel.setShowBadge(false);
        notificationManager.createNotificationChannel(mChannel);
    }

    Intent intent = new Intent( packageContext: this, MainActivity.class);
    PendingIntent pendingIntent = PendingIntent.getActivity( context: this, requestCode: 0 , intent, flags: 0);

    NotificationCompat.Builder nm = new NotificationCompat.Builder( context: this, CHANNEL_ID)
        .setContentIntent(pendingIntent)
        .setSmallIcon(R.drawable.ic_launcher_background)
        .setContentTitle("Reproduciendo música")
        .setContentText("Información adicional");

    notificationManager.notify(ID_NOTIFICATION_CREAR, nm.build());

    return START_STICKY;
}

```

**Paso 4.** Lanza el servicio en el método **onCreate()** de la actividad **MainActivity** de **Asteroides**. Para el servicio en el método **onDestroy()**.

La idea es desactivar (comentar) el código que activa el objeto **mediaplayer** en **MainActivity** de **Asteroides**, ya que a partir de ahora de esta funcionalidad se va a encargar el servicio **ServicioMusica**.

```

//////////onCreate de MainActivity//////////
//if (mp==null) {
//    mp = MediaPlayer.create(this, R.raw.audio);
//    mp.start();
//}
startService(new Intent( packageContext: this, ServicioMusica.class));
Toast.makeText( context: this, text: "onCreate", Toast.LENGTH_SHORT).show();

```

```

@Override protected void onDestroy() {
    Toast.makeText( context: this, text: "onDestroy", Toast.LENGTH_SHORT).show();
    stopService(new Intent( packageContext: this, ServicioMusica.class));
    super.onDestroy();
    //mp.stop();
}

```

Comentamos el objeto mediaplayer de Asteroides porque ahora utilizaremos el objeto mediaplayer de ServicioMusica.

```

public class MainActivity extends AppCompatActivity
    implements GestureOverlayView.OnGesturePerformedListener {

    //private MediaPlayer mp=null;

    private GestureLibrary libreria;
    public static AlmacenPuntuaciones almacen= new AlmacenPuntuacionesLista();
    private Button bAcercaDe;
    private Button bSalir;
    private Button bPlay;
    private Button bConfigurar;
    private Animation animacion;

```

```

@Override protected void onResume() {
    super.onResume();
    //Toast.makeText(this, "onResume", Toast.LENGTH_SHORT).show();
    //mp.start();
}

@Override protected void onStop() {
    super.onStop();
    //Toast.makeText(this, "onStop", Toast.LENGTH_SHORT).show();
    //mp.pause();
}

```



```

@Override protected void onSaveInstanceState(Bundle estadoGuardado){
    super.onSaveInstanceState(estadoGuardado);
    //if (mp != null) {
    //    int pos = mp.getCurrentPosition();
    //    estadoGuardado.putInt("posicion", pos);
    //}
}

@Override protected void onRestoreInstanceState(Bundle estadoGuardado){
    super.onRestoreInstanceState(estadoGuardado);
    //if (estadoGuardado != null && mp != null) {
    //    int pos = estadoGuardado.getInt("posicion");
    //    mp.seekTo(pos);
    //}
}

```

**Paso 5.** Edita el fichero **AndroidManifest.xml** y añade la siguiente línea dentro de la etiqueta **<application>**.

```
<service android:name=".ServicioMusica"/>
```

```

<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_mi_icono"
    android:label="Asteroides"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name=".AcercaDeActivity"
        android:label="Acerca de ..."
        android:theme="@style/Theme.AppCompat.Light.Dialog" />
    <activity
        android:name=".Preferencias"
        android:label="Preferencias" />
    <activity android:name=".Puntuaciones" />
    <activity android:name=".Juego" android:screenOrientation="landscape"
        android:theme="@style/Theme.AppCompat.Light.NoActionBar"></activity>
    <service android:name=".ServicioMusica"/>
</application>

```

**Paso 6.** Si realizas el punto anterior simplemente lanzando la actividad **Mainactivity** de **Asteroides**, cuando el usuario pulse sobre la notificación el sistema lanzará una nueva tarea, aunque ya exista una previa. Si te interesa que no se lance una nueva tarea cuando ya exista una previa añade la línea en **AndroidManifest.xml**.

```

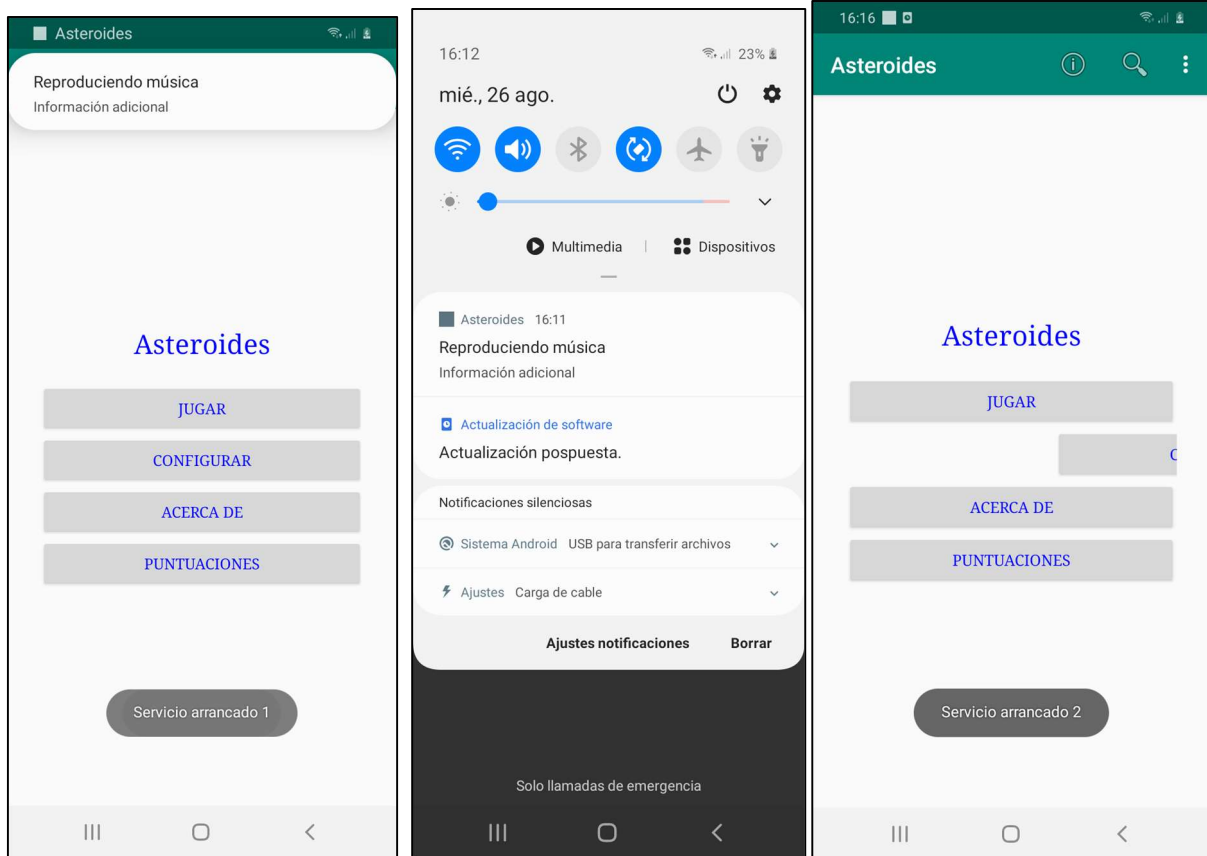
<activity
    android:name=".MainActivity"

```

**android:label="@string/app\_name"**

**android:launchMode="singleTask">**

El Servicio de musica lanza una notificación al Servicio de notificaciones de Android. Si hacemos click sobre la notificación se lanzará una nueva instancia de la aplicación de asteroides (aunque ya exista una previa).

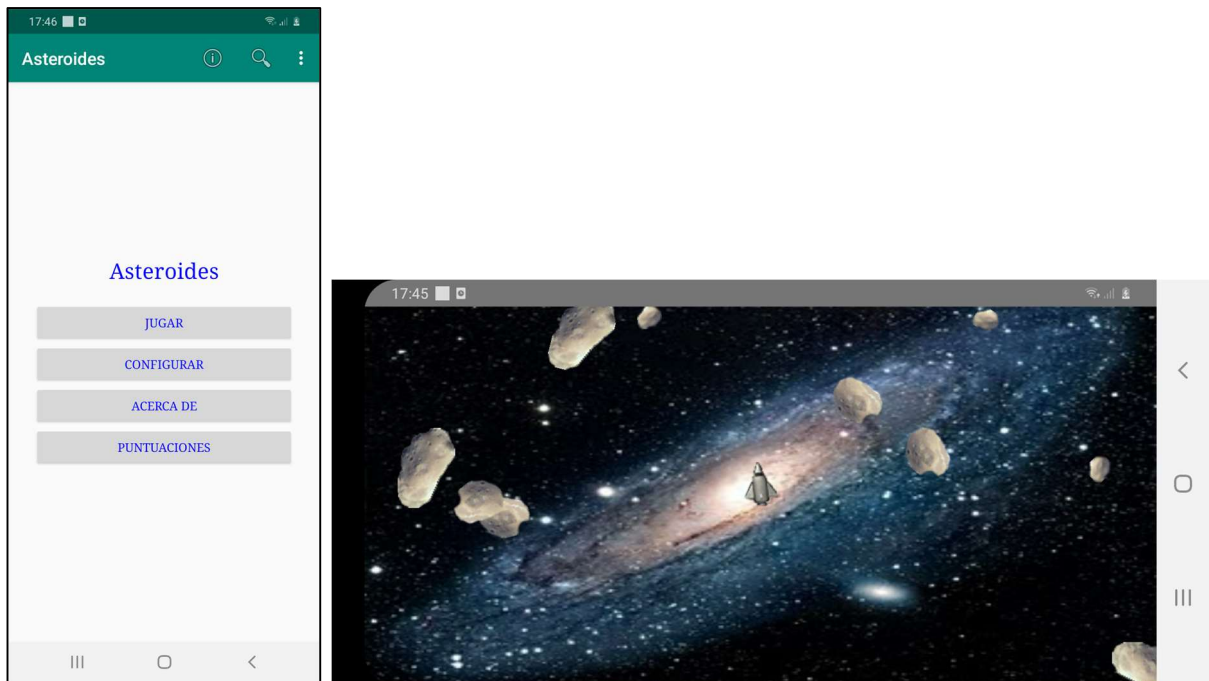


Mediante la cláusula `singleTask`, solo se permite una instancia de esa actividad, por lo tanto si hacemos click en la notificación, no abrirá otra instancia de la aplicación.

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_mi_icono"
    android:label="Asteroides"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity"
        android:label="@string/app_name"
        android:launchMode="singleTask">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
```

**Paso 7.** Ejecuta la aplicación.

La musica suena tanto en la pantalla principal como durante el juego propiamente dicho. Esto es debido a que la finalización del servicio se encuentra definida en el evento `onDestroy` de `asteroides.java`, que ocurre cuando finaliza completamente la aplicación.

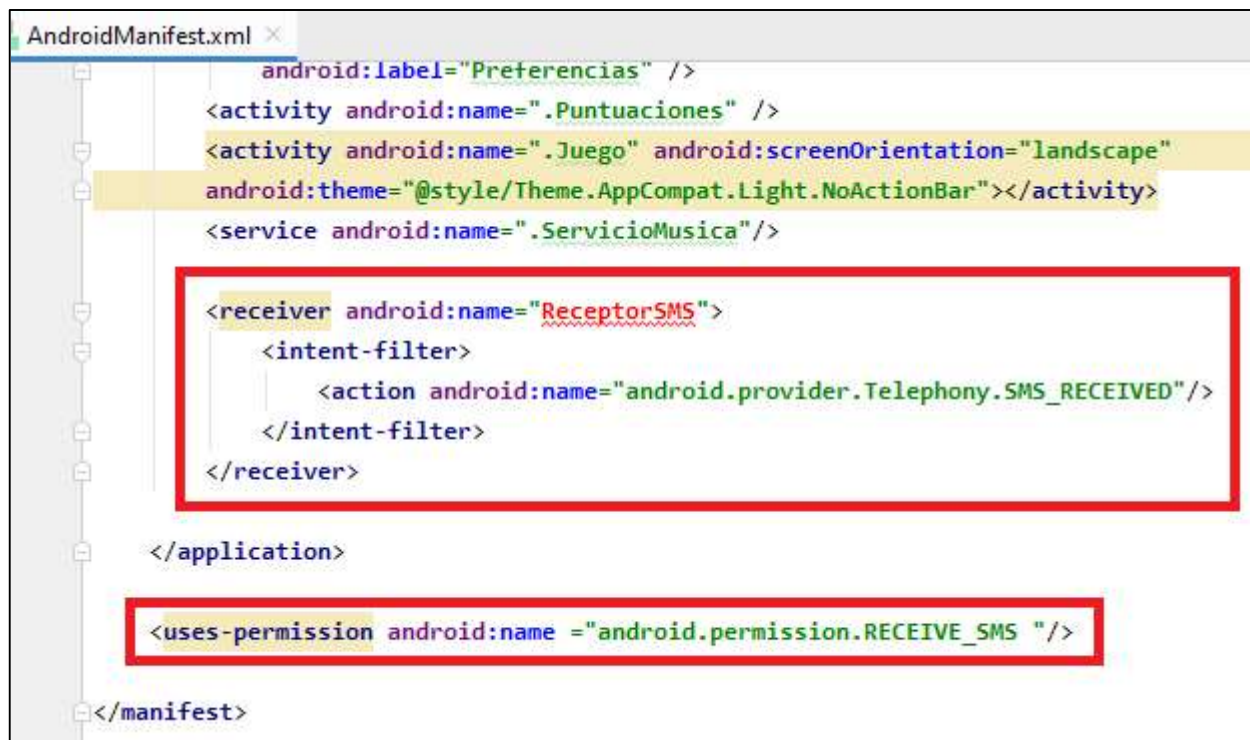


**Parte II: Arranque de una actividad al llegar un SMS.**

Vamos a modifica el proyecto *Asteroides* para que se arranque automáticamente la actividad *AcercaDe* al llegar un SMS cualquiera.

**Paso 1.** Abre el proyecto *Asteroides*.

**Paso 2.** En *AndroidManifest.xml* pide el permiso adecuado y registra el receptor de anuncios:



**Paso 3.** Crea una nueva clase con el siguiente código:

```

public class ReceptorSMS extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        Intent i = new Intent(context, AcercaDeActivity.class);
        i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        Toast.makeText(context, text: "SMS RECIBIDO", Toast.LENGTH_LONG).show();
        context.startActivity(i);
    }
}

```

La forma de arrancar una actividad desde un receptor de anuncios es muy similar a la que estudiamos en el capítulo 3. La única diferencia es que ahora hemos necesitado añadir un *flag* a la intención, que indica que la actividad sea creada en una nueva tarea.

**Paso 4.** Ejecuta la aplicación. Envía un SMS al dispositivo y verifica que se abre la actividad *Acerca de ...*

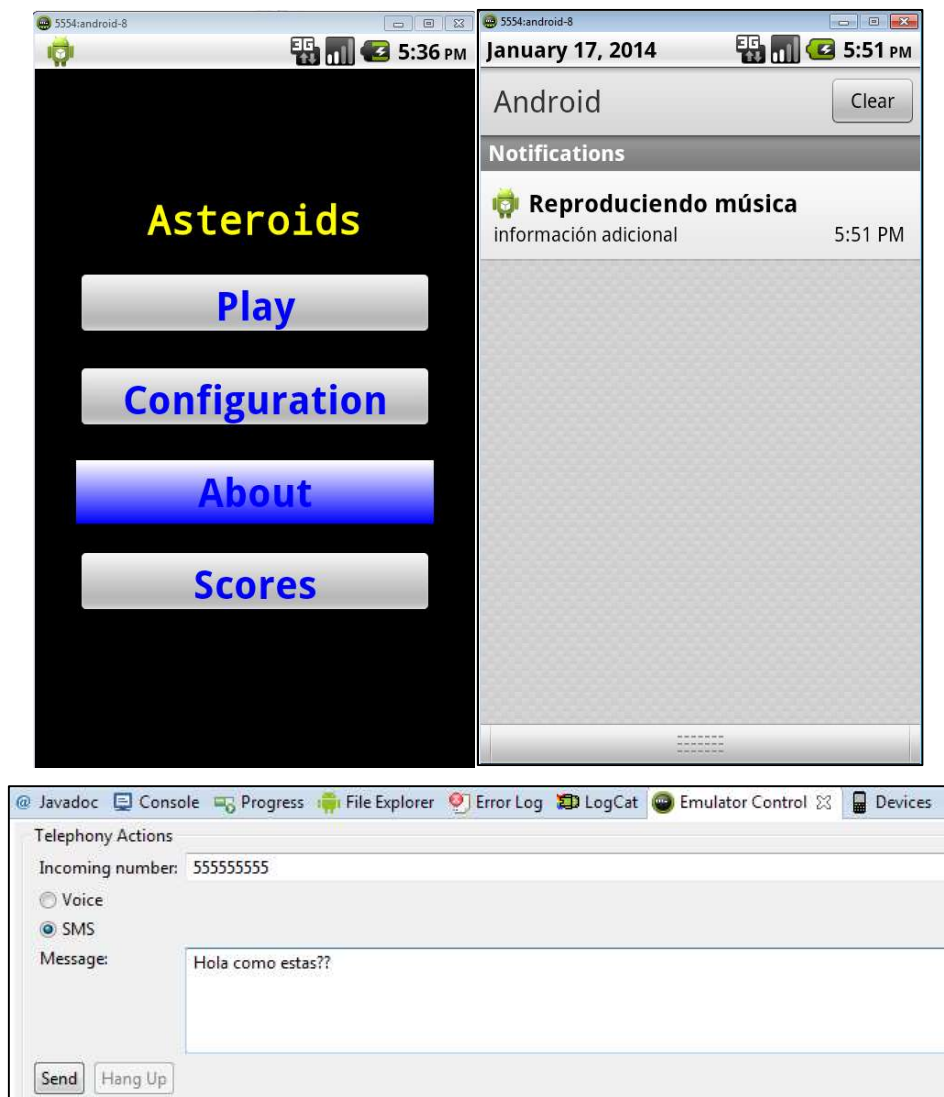


Enviamos un SMS al móvil.

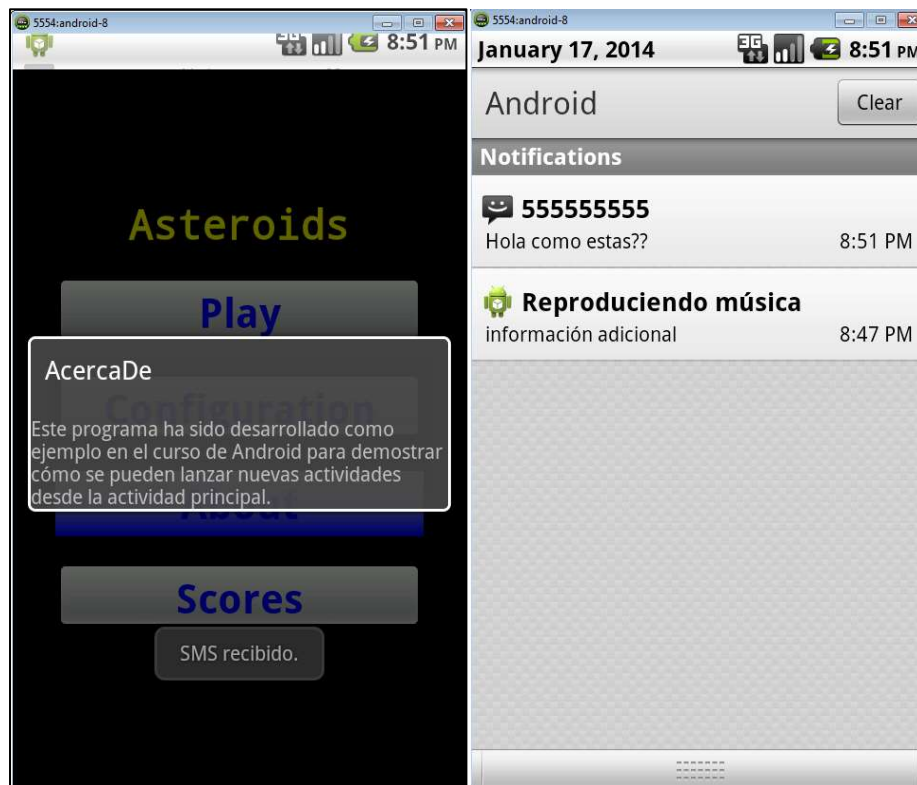




Probamos ahora en el emulador:







**NOTA:** Si utilizas el emulador, puedes utilizar la vista *EmulatorControl* para mandar simular el envío de un SMS.

Cuando lanzamos una nueva actividad, Android nos permite controlar en que tarea y en qué posición de la pila se situará. No obstante, se recomienda usar siempre el sistema estándar. Es decir, si lanzamos una nueva actividad desde otra actividad, la nueva actividad se sitúa en la misma tarea en la cima de la pila de actividades. Cuestión diferente es lanzar la actividad desde un receptor de anuncios, dado que cuando llegue el SMS podemos encontrarnos en cualquier tarea. En este caso, resulta imprescindible activar el flag **FLAG\_ACTIVITY\_NEW\_TASK**, así, en caso de no existir la tarea esta podrá ser creada.