

**Nuevos Paradigmas de Interacción (2015-2016)**  
GRADO EN INGENIERÍA INFORMÁTICA  
UNIVERSIDAD DE GRANADA

---

## Tutorial appGestosFoto

---

Pedro Antonio Ruiz Cuesta  
Ignacio Martín Requena

8 de febrero de 2016

## Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Descripción</b>	<b>3</b>
<b>3. Desarrollo</b>	<b>3</b>
3.1. Reconocer un gesto . . . . .	3
3.2. Haciendo la foto automáticamente . . . . .	4
3.3. Ver la foto . . . . .	4

## Índice de figuras

## 1. Introducción

En este tutorial veremos como podemos desarrollar una aplicación android que reconozca un patrón dibujado en la pantalla, un gesto, y haga automáticamente una foto pasados unos segundos, en nuestro caso 3. Después la foto se mostrará.

## 2. Descripción

La aplicación constará de 3 actividades, una para detectar el gesto, otra para hacer la foto y otra final para visualizar la imagen capturada. Para ello, hemos partido de un ejemplo que nos ofrece google directamente para el uso de la cámara llamado Camera2Basic. Este ejemplo utiliza la nueva API para la cámara, la cuál está disponible para versiones de Android a partir de 5.0.<sup>1</sup>

## 3. Desarrollo

Para el desarrollo de nuestra app utilizaremos el IDE Android Studio.

Antes de nada en el AndroidManifest deberíamos añadir permisos de uso de la cámara:

```
<uses-permission android:name=.android.permission.CAMERA/>
<uses-feature android:name=.android.hardware.camera/>
<uses-feature android:name=.android.hardware.camera.autofocus/>
```

### 3.1. Reconocer un gesto

Lo primero que vamos a hacer es descargarnos una app que podemos encontrar en la play store llamada **Gesture Builder**, con ella es muy sencillo crear el gesto que deseemos.

Una vez creado nuestro gesto, tenemos que ir (en nuestro terminal) al directorio de la app y buscar el archivo *gestures.txt*. Este archivo lo copiamos y lo guardamos en la carpeta de recursos de nuestro proyecto android (nosotros hemos decididido guardarlo en un nuevo directorio llamado "raw"). Ya tenemos nuestro gesto creado, ahora toca reconocerlo en nuestra app. Para ello tenemos que añadir a la interfaz un GestureOverlayView que será el encargado de reconocer el patrón que dibujamos, también debemos hacer que la activity donde vamos a reconocer el gesto implemente GestureOverlayView.OnGesturePerformedListener, que nos hará implementar un método:

```
1 public void onGesturePerformed(GestureOverlayView overlay, Gesture
    gesture) {...}
```

---

<sup>1</sup><http://developer.android.com/intl/es/samples/Camera2Basic/index.html>

Declaramos un objeto gLibrary de tipo GestureLibrary como atributo de la clase. Ahora en onCreate de nuestra activity tenemos que añadir alguna cosa más:

```
1      GestureOverlayView gesture;  
2      //Relacionamos con el XML  
3      gesture = (GestureOverlayView)findViewById(R.id.  
4          gestureOverlayView1);  
5      //Le añadimos el listener  
6      gesture.addOnGesturePerformedListener(this);  
7      //Creamos la carpeta res/raw y añadimos el archivo gestures  
8      //añadimos el raw al gLibrary  
9      gLibrary = GestureLibraries.fromRawResource(this, R.raw.  
10         gestures);  
11      //lo cargamos...  
12      gLibrary.load();
```

Ya tenemos en nuestra activity un reconocedor de gestos hecho, ahora en el método que hemos tenido que implementar tenemos que recoger los datos producidos por cada gesto que realizemos:

```
1      ArrayList<Prediction> predictions = gLibrary.recognize(gesture);
```

Una lista de gestos reconocidos, nos quedaremos con la primera, que es la más fiable. Por último nos queda lanzar la siguiente activity que es la que hará la foto, para ello simplemente decimos que si la precision.score es mayor que el valor de precisión que nosotros queramos aplicar, se lance la activity correspondiente.

### 3.2. Haciendo la foto automáticamente

Puesto que hemos cogido el ejemplo anteriormente comentado, tenemos en nuestra activity "CameraActivity" ya todo preparado para hacer la foto. Para que se lance automáticamente la captura, hemos puesto un cronómetro (CountDownTimer) que a los 3 segundos hace la foto, todo esto se ha hecho en el método onOpened del ejemplo que habíamos descargado.

### 3.3. Ver la foto

Al hacer la foto en la CameraActivity lanzamos la siguiente activity que será la que nos muestra la captura realizada, eso lo conseguimos simplemente pasando en el intent que pasa lanza la PictureActivity desde CameraActivity un Extra con el archivo que ha creado. Este archivo lo recogemos en la PictureActivity y lo mostramos con un ImageView.