

# 1 Cámara fotográfica

## 1.1 Sumario

- 1 Introducción
- 2 A Cámara
- 3 Caso práctico
  - ♦ 3.1 O xml do layout
  - ♦ 3.2 O Código java da aplicación
- 4 Afondar sobre o uso da cámara e vídeo

## 1.2 Introducción

- Neste curso só se vai introducir como manexar a cámara de fotos e amosar a foto obtida.
- Para ver como gardar as imaxes e como incluso gravar e manipula vídeos hai un apartado final onde se indica como proceder.

As clases que interveñen son:

- **Clase Camera:** danos acceso á cámara, as súas características. Usarémosla se facemos unha aplicación na que queiramos xestionar a cámara por nos mesmos.

Para facer uso da cámara podemos utilizar un Intent, evitándonos ter que deseñar a aplicación para xestionar a cámara.

Permisos necesarios a engadir no arquivo AndroidManifest.xml:

- Permiso para facer uso da cámara:

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

Nota: Se usamos a opción de chamar á cámara cun Intent non necesitamos dito permiso.

- Permiso para indicar que a nosa aplicación fai uso da cámara (Olo que é uses-feature):

```
<uses-feature android:name="android.hardware.camera" required="true"/>
```

Se queremos facer uso doutras características?, tanto da cámara coma doutro hardware que ten o dispositivo móbil, consultar:

<http://developer.android.com/guide/topics/manifest/uses-feature-element.html#hw-features>

Desta forma Google Play impedirá que se instale á aplicación se o dispositivo non ten o hardware necesario coas características especificadas.

Neste caso, a nosa aplicación está requirindo o uso da cámara, pero pode ocorrer que non a necesite para que funcione (podemos limitar a funcionalidade da nosa aplicación se o dispositivo non ten cámara, por exemplo). Para indicar isto temos que poñer:

```
<uses-feature android:name="android.hardware.camera" android:required="false" />
```

Chamaremos a unha aplicación e recolleremos o resultado de volta, que será a foto.

Os pasos a seguir son:

- Crear un Intent que sexa do tipo:

♦ **MediaStore.ACTION\_IMAGE\_CAPTURE** : Captura unha imaxe.

```
Intent intento = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
```

- Chamar o método `startActivityResult()`.

Neste momento o S.O. lanzará unha aplicación para recoller a imaxe.

Lembrar que con esta forma de chamar a unha activity, esperamos un resultado (a foto).

```
startActivityResult(intento, REQUEST_CODE_CAMARA);
```

- Obter o resultado da aplicación lanzada. Para iso temos que desenrolar o método `onActivityResult()` da nosa Activity.
- No método `onActivityResult()` imos recoller a imaxe vén no campo "data" dentro de `getExtras` do obxecto `Bundle`:

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    Bitmap bitMap= data.getExtras().get("data");
    .....
}
```

Unha vez temos o `Bitmap` podemos visualizalo nun control `ImageView` da seguinte forma:

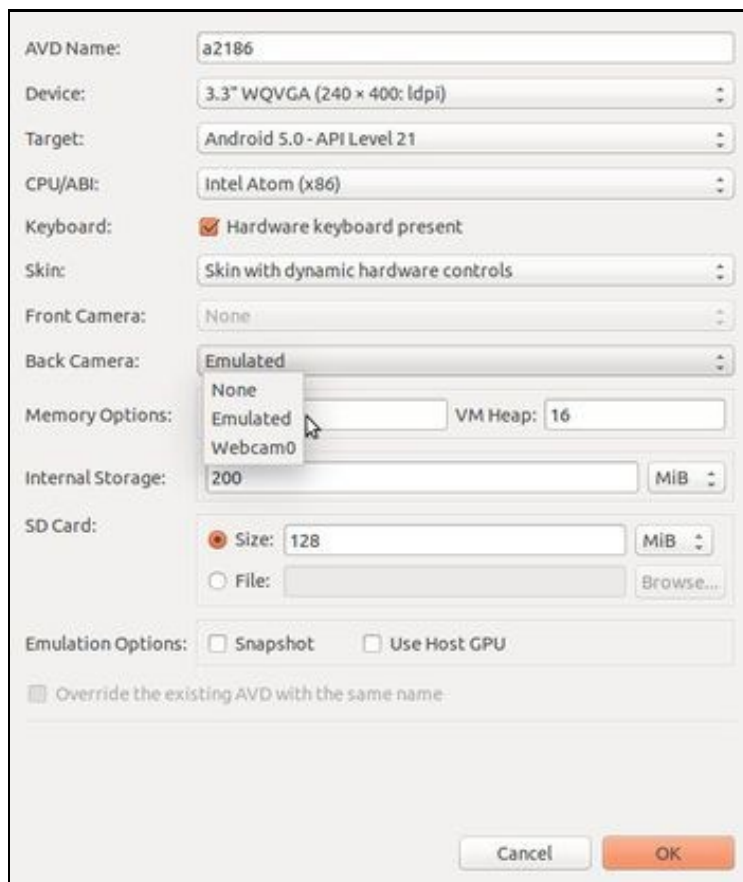
```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    Bitmap bitMap= data.getExtras().get("data");
    imageView.setImageBitmap(bitMap);
}
```

Sendo `imageView` un obxecto da clase `ImageView`.

## 1.3 A Cámara

- Segundo o AVD/Dispositivo este pode ter cámara traseira e tamén dianteira, para os famosos autoretratos, que todos damos en chamar Selfies
- No caso dun AVD a cámara pode ser emulada ou usar a do ordenador real se dispón dela.
- Nesta imaxe amósase que se vai usar a cámara emulada.

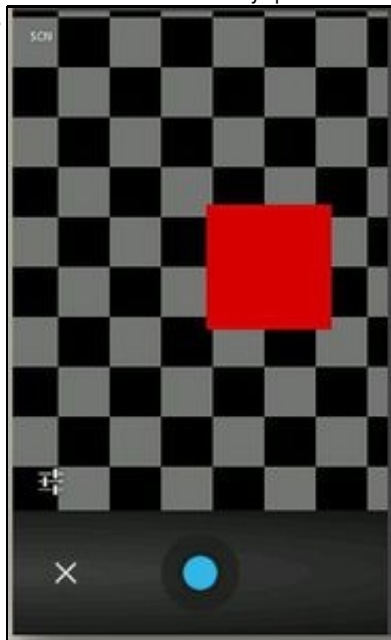


## 1.4 Caso práctico

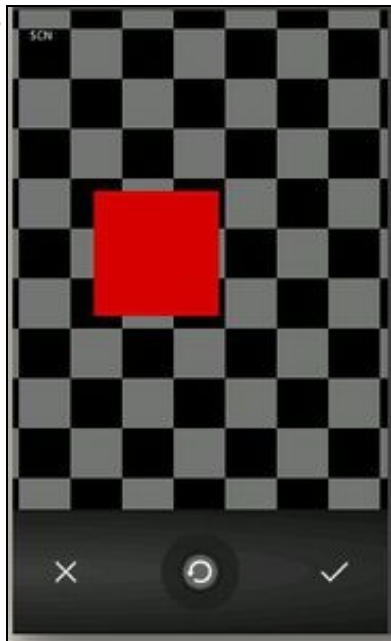
- Crear o proxecto: **U5\_10\_Fotos**
- Realizar unha foto



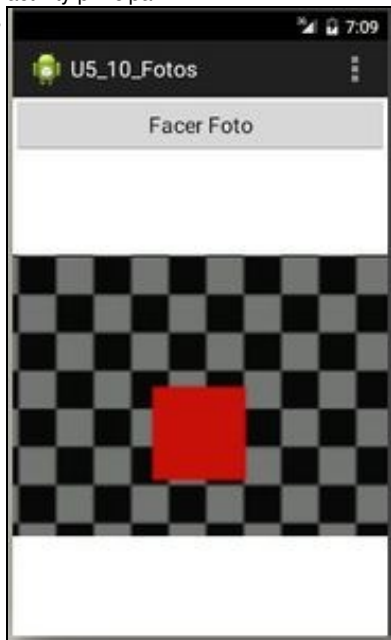
A aplicación dispón dun botón que vai lanzar a través dun intent. E dunha view "ImageView" que vai amosar a foto que nos devolve a cámara cando volvemos da activity que a manexa. Nestes intres ese View ten o debuxo de Android.



Como a cámara é emulad, anda por aí un cadrado bailando até que premamos o botón do obturador para realizar a foto.



Unha vez premido o obturador vemos que podemos descartar a foto (x) ou aceptala (marca de verificación), en calquera caso volvemos á activity principal.



No caso de aceptar a foto, vemos na para inferior da activity principal a foto realizada coa cámara na activity secundaria.

### 1.4.1 O xml do layout

```
<LinearLayout 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:orientation="vertical" >

    <Button
        android:id="@+id/bFoto"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="onBotonClick"
        android:text="Facer Foto" />

    <ImageView
        android:id="@+id/imgvwFoto"
        android:layout_width="match_parent"
```

```

        android:layout_height="match_parent"
        android:contentDescription="foto a sacar"
        android:src="@drawable/ic_launcher" />

</LinearLayout>

```

## 1.4.2 O Código java da aplicación

```

package com.example.u5_10_fotos;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageView;
import android.widget.Toast;

public class U5_10_Fotos extends Activity {
    private final int REQUEST_CODE_CAMARA = 1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_u5_10__fotos);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.u5_10__fotos, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

    public void onBotonClick(View v) {
        Intent intento = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        startActivityForResult(intento, REQUEST_CODE_CAMARA);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        // TODO Auto-generated method stub
        if (requestCode == REQUEST_CODE_CAMARA) {
            if (resultCode == RESULT_OK) {

                if (data == null) {
                    Toast.makeText(this, "NON HAI IMAXE", Toast.LENGTH_LONG).show();
                    return;
                }

                ImageView imgview = (ImageView) findViewById(R.id.imgvwFoto);
                imgview.setImageBitmap((Bitmap) data.getExtras().get("data"));

            } else if (resultCode == RESULT_CANCELED) {
                // Foto cancelada
            }
        }
    }
}

```

```
} else {  
    // Fallo na captura da foto.  
}  
}  
  
}  
  
}
```

- **Liñas 43-44:** Creamos un intent e lanzámolo esperando un resultado (a foto, neste caso).
- **Liñas 48-68:** Recibimos o resultado da activity secundaria (a que manexa a cámara) e comprobamos se o código devolto foi o mesmo que usamos para lanzar a cámara e tamén comprobamos se recibimos unha imaxe que nese caso imos amosar na vista ImageView, unha vez que a recollemos da chave "data" do campo data.

## 1.5 Afondar sobre o uso da cámara e vídeo

- Para ter un coñecemento máis profundo do uso da cámara:
  - ♦ Gardar as fotos,
  - ♦ Gravar e gardar vídeos,
  - ♦ Audio,
  - ♦ Etc.
- Pódese consultar o seguinte enlace do curso avanzado de Android: [PDM Avanzado Captura de Vídeo / Imaxes](#)

-- Ángel D. Fernández González e Carlos Carrión Álvarez -- (2015).