

# 1 Imaxes

## 1.1 Sumario

- 1 Introducción
- 2 Captura de Vídeo / Audio
- 3 Accedendo ó Vídeo / Foto
  - ◆ 3.1 Caso Práctico
    - ◇ 3.1.1 Creamos a Activity
- 4 Gardando o Vídeo / Foto
  - ◆ 4.1 Caso Práctico
    - ◇ 4.1.1 Creamos a Activity
  - ◆ 4.2 Información Extra
  - ◆ 4.3 Control para o VideoView
- 5 Almacenar fotos e vídeos en APIS >=23 e >=24
  - ◆ 5.1 Creamos a Activity
  - ◆ 5.2 Código para API>=24
- 6 Métodos útiles no manexo de Imaxes
- 7 Método útil no manexo de Vídeos
- 8 Visualización de múltiples imaxes
  - ◆ 8.1 Caso Práctico
    - ◇ 8.1.1 Preparación
    - ◇ 8.1.2 Creamos a Activity
    - ◇ 8.1.3 Creamos o ImageView personalizado

## 1.2 Introducción

As clases que interveñen na gravación son:

- **Clase Camera:** danos acceso á cámara, as súas características. Usarémosla se facemos unha aplicación na que queiramos xestionar a cámara por nos mesmos.
- **Clase SurfaceView:** para previsualizar o que imos gravar.
- **Clase MediaRecorder:** permítenos gravar vídeo dende a cámara.

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

Permisos necesarios a engadir no arquivo AndroidManifest.xml:

- Permiso de lectura sobre a tarxeta SD Externa (no caso de utilizar un dispositivo cunha API 23 ou superior).

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

- 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:

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

- Permisión de almacenamento: se a aplicación vai gardar os datos nunha tarxeta SD externa:

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

- Permisión para capturar audio (se é o caso):

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

## 1.3 Captura de Vídeo / Audio

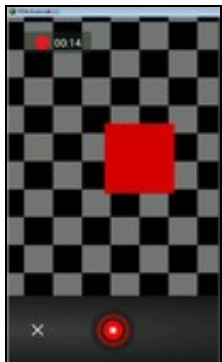
Neste manual imos aprender como facer uso das aplicacións que nos proporciona o S.O. Android para sacar unha foto ou vídeo.

Nos chamaremos a unha destas aplicacións e recolleremos o resultado de volta (que será a foto / vídeo sacado).

---

Nota: Facer esta parte utilizando o emulador é extremadamente lenta.

Cando prememos o botón de gravar temos que agardar ata que o rectángulo do interior da pantalla se poña vermello:



---

Os pasos a seguir son:

- Crear un Intent que sexa dun destes tipo:

◊ `MediaStore.ACTION_IMAGE_CAPTURE` : Captura unha imaxe.

◊ `MediaStore.ACTION_VIDEO_CAPTURE` : Captura un vídeo.

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

- Chamar ó método `startActivityForResult()`.

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

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

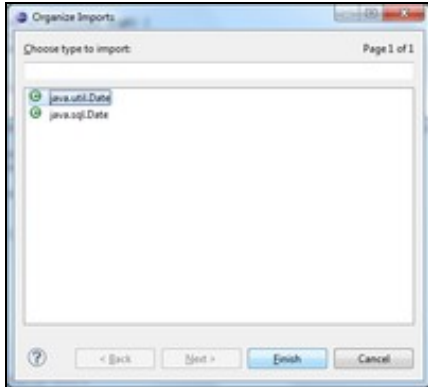
```
startActivityForResult(intento, REQUEST_CODE_GRAVACION_OK);
```

- Obter o resultado da aplicación lanzada. Para iso temos que desenrolar o **método onActivityResult()** da nosa Activity.

Cando chamamos a activity que lanza a aplicación de sacar foto ou gravar vídeo, podemos enviarlle como parámetros (no obxecto Bundle) unha serie de datos extras, como a calidade da foto, a calidade de vídeo, o tamaño máximo de gravación, onde queremos gardar a foto/vídeo...

Máis información en [http://developer.android.com/reference/android/provider/MediaStore.html#EXTRA\\_OUTPUT](http://developer.android.com/reference/android/provider/MediaStore.html#EXTRA_OUTPUT).

**Nota:** Para asignar un nome único a cada imaxe / vídeo capturado imos facer uso da clase Date. Cando importemos dita clase lembrar escoller á que se atopa no paquete util:



No caso de non enviar como información extra ó Intent onde queremos gardar o vídeo /foto, estes van vir no obxecto Intent dentro do método onActivityResult():

**No caso das imaxes,** a imaxe ven 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.

**No caso do vídeo,** obtemos o vídeo chamando ó método getData() do obxecto bundle. Isto devolve a URI do vídeo.

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

    Uri uri = data.getData();
    .....
}
```

Para visualizalo necesitamos un obxecto da clase VideoView.

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

    vidView.setVideoURI(data.getData());
    vidView.start();
}
```

Sendo vidView un obxecto da clase VideoView.

## 1.4 Accedendo ó Vídeo / Foto

Cando chamamos a activity que vai sacar a foto / vídeo podemos non indicar onde gardar esa foto

Neste punto imos ver como podemos acceder ó vídeo / foto que ven no Intent dentro do método onActivityResult().

### 1.4.1 Caso Práctico

O obxectivo desta práctica é visualizar nun control ImageView / VideoView a imaxe / vídeo obtida dende a aplicación do S.O. Android.



Nesta aplicación, debaixo do botón, se atopa un control ImageView e un control VideoView.

Dependendo da opción escollida no RadioButton, un deles estará invisible (propiedade visibility="gone").

#### 1.4.1.1 Creamos a Activity

- Nome do proxecto: **UD2\_04\_MultimediaFotoVideo**
- Nome da activity: **UD2\_04\_MultimediaFotoVideo.java**

#### Código do layout xml

```
<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" >

    <RadioGroup
        android:id="@+id/UD2_04_rgrpOpciones"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:gravity="center_horizontal"
        android:orientation="horizontal" >

        <RadioButton
            android:id="@+id/UD2_04_rbbtnFoto"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sacar Foto"
        android:textSize="15sp"
        android:checked="true" />

<RadioButton
    android:id="@+id/UD2_04_rbtnVideo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="15sp"
    android:text="Sacar Vídeo" />
</RadioGroup>

<Button
    android:id="@+id/UD2_04_btnGravarVideoFoto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/UD2_04_rgrpOptions"
    android:textSize="@dimen/botons"
    android:text="Lanzar Aplicación" />

<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/UD2_04_btnGravarVideoFoto">

    <ImageView
        android:id="@+id/UD2_04_imgvwFoto"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:contentDescription="foto a sacar"
        android:layout_gravity="center"
        android:src="@drawable/ic_launcher" />

    <VideoView
        android:id="@+id/UD2_04_vidvwVideo"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="center"
        android:visibility="gone" />

</FrameLayout>

</RelativeLayout>

```

## Código da clase UD2\_04\_MultimediaFotoVideo

**Obxectivo:** Comprobar como capturar unha imaxe / vídeo e visualizala nun control ImageView / VideoView

```

import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.RadioGroup.OnCheckedChangeListener;
import android.widget.Toast;
import android.widget.VideoView;

public class UD2_04_MultimediaFotoVideo extends Activity {

    /**
     * Código para verificar que o resultado ven do intent de gravación
     */

```

```

private final int REQUEST_CODE_GRAVACION_OK = 1;

/**
 * Obtemos a imaxe ou video que ven da aplicacion Android
 */
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == REQUEST_CODE_GRAVACION_OK) {
        if (resultCode == RESULT_OK) {

            if (data == null) {
                Toast.makeText(getApplicationContext(), "NON HAI IMAXE/VIDEO A GARDAR", Toast.LENGTH_LONG).show();
                return;
            }

            RadioButton rb = (RadioButton) findViewById(R.id.UD2_04_rbbtnFoto);
            if (rb.isChecked()) { // Saca foto
                ImageView imgview = (ImageView) findViewById(R.id.UD2_04_imgvwFoto);
                imgview.setImageBitmap((Bitmap) data.getExtras().get("data"));
            } else { // Saca vídeo
                VideoView vidview = (VideoView) findViewById(R.id.UD2_04_vidvwVideo);
                vidview.setVideoURI(data.getData());
                vidview.start();
            }

            } else if (resultCode == RESULT_CANCELED) {
                // Video ou Foto cancelada
            } else {
                // Fallo na captura do Video ou foto.
            }
        }

    /**
     * Programa o código dos click's os botóns
     */
    private void xestionarEventos(){
        Button gravar = (Button) findViewById(R.id.UD2_04_btnGravarVideoFoto);
        gravar.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                RadioButton rb = (RadioButton) findViewById(R.id.UD2_04_rbbtnFoto);

                if (rb.isChecked()){ // Saca foto

                    Intent intento = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                    startActivityForResult(intento, REQUEST_CODE_GRAVACION_OK);

                }
                else { // Grava vídeo
                    Intent intento = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
                    startActivityForResult(intento, REQUEST_CODE_GRAVACION_OK);

                }

            }

        });

        RadioGroup rgroup = (RadioGroup) findViewById(R.id.UD2_04_rgrpOptions);
        rgroup.setOnCheckedChangeListener(new OnCheckedChangeListener() {

            @Override
            public void onCheckedChanged(RadioGroup group, int checkedId) {
                // TODO Auto-generated method stub
                ImageView imgview;
                VideoView videoview;

                switch (checkedId) {

```

```

case R.id.UD2_04_rbtnFoto:// OCULTAMOS O VIDEO
imgview = (ImageView) findViewById(R.id.UD2_04_imgvwFoto);
imgview.setVisibility(View.VISIBLE);
videoview = (VideoView) findViewById(R.id.UD2_04_vidvwVideo);
videoview.setVisibility(View.GONE);
break;
case R.id.UD2_04_rbtnVideo:// OCULTAMOS AS FOTOS
imgview = (ImageView) findViewById(R.id.UD2_04_imgvwFoto);
imgview.setVisibility(View.GONE);
videoview = (VideoView) findViewById(R.id.UD2_04_vidvwVideo);
videoview.setVisibility(View.VISIBLE);
break;
}
}
});
}

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_ud2_04__multimedia_foto_video);

xestionarEventos();
}
}

```

- Liñas 38-39: Nestas liñas obtemos a foto da aplicación do S.O. e a visualiza no ImageView.
- Liñas 42-44: Nestas liñas obtemos o vídeo do Intent e o cargamos no VideoView e comezamos a reprodución.
- Liñas 70-71: Evento click sobre o botón. Se está a opción do RadioButton de sacar unha foto lanzamos o Intent ACTION\_IMAGE\_CAPTURE, esperando o resultado.
- Liñas 75-76: Evento click sobre o botón. Se está a opción do RadioButton de sacar un vídeo lanzamos o Intent ACTION\_VIDEO\_CAPTURE, esperando o resultado.

## 1.5 Gardando o Vídeo / Foto

Neste punto imos facer unha modificación da práctica anterior e imos enviar ao Intent información sobre onde debe gardar a foto / vídeo.

Neste caso, o Intent do método onActivityResult **NON DEVOLVERÁ NINGÚN DATO**.

### 1.5.1 Caso Práctico

O obxectivo desta práctica é visualizar nun control ImageView / VideoView a imaxe / vídeo obtida dende a aplicación do S.O. Android PERO GARDANDO NA TARXETA SD O RESULTADO DE SACAR ESE VÍDEO / FOTO.



Neste caso as fotos gardadas se almacenan na tarxeta SD no cartafol PICTURES e os vídeos no cartafol MOVIES.

#### 1.5.1.1 Creamos a Activity

- Nome do proxecto: **UD2\_05\_MultimediaFotoVideo**
- Nome da activity: **UD2\_05\_MultimediaFotoVideo.java**
- O código do layout vai ser o mesmo que na activity anterior.

#### Código da clase UD2\_05\_MultimediaFotoVideo

**Obxectivo:** Variación da práctica anterior no que indicamos como dato extra a enviar ó intent que obtén a foto / vídeo onde queremos que o garde.

```
import java.io.File;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.RadioGroup.OnCheckedChangeListener;
import android.widget.VideoView;

public class UD2_05_MultimediaFotoVideo extends Activity {

    /**
     * Código para verificar que o resultado ven do intent de gravación
     */
    private final int REQUEST_CODE_GRAVACION_OK = 1;

    final private String nomeVideo="video.mp4";
    final private String nomeFoto="foto.jpg";
```



```

/**
 * Obtemos a imaxe ou video que ven da aplicacion Android
 */
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == REQUEST_CODE_GRAVACION_OK) {
        if (resultCode == RESULT_OK) {

            RadioButton rb = (RadioButton) findViewById(R.id.UD2_04_rbbtnFoto);
            if (rb.isChecked()) { // Saca foto
                File ruta = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES);
                File arquivo = new File(ruta,nomeFoto);
                if (!arquivo.exists()) return;// Non hai foto

                ImageView imgview = (ImageView) findViewById(R.id.UD2_04_imgvwFoto);
                Bitmap bitmap = BitmapFactory.decodeFile(arquivo.getAbsolutePath());
                imgview.setImageBitmap(bitmap);

            } else { // Saca vídeo
                File ruta = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_MOVIES);
                File arquivo = new File(ruta,nomeVideo);
                if (!arquivo.exists()) return;// Non hai foto

                VideoView vidview = (VideoView) findViewById(R.id.UD2_04_vidvwVideo);
                vidview.setVideoURI(Uri.fromFile(arquivo));
                vidview.start();
            }

        } else if (resultCode == RESULT_CANCELED) {
            // Video ou Foto cancelada
        } else {
            // Fallo na captura do Video ou foto.
        }
    }

}

/**
 * Programa o código dos click's os botóns
 */
private void xestionarEventos(){
    Button gravar = (Button)findViewById(R.id.UD2_04_btnGravarVideoFoto);
    gravar.setOnClickListener(new OnClickListener(){

        @Override
        public void onClick(View arg0) {
            // TODO Auto-generated method stub
            RadioButton rb = (RadioButton)findViewById(R.id.UD2_04_rbbtnFoto);

            if (rb.isChecked()){// Saca foto
                File ruta = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES);
                File arquivo = new File(ruta,nomeFoto);

                Intent intento = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                intento.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(arquivo));
                startActivityForResult(intento, REQUEST_CODE_GRAVACION_OK);

            }
            else {// Grava vídeo
                File ruta = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_MOVIES);
                File arquivo = new File(ruta,nomeVideo);

                Intent intento = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
                intento.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(arquivo));

                startActivityForResult(intento, REQUEST_CODE_GRAVACION_OK);

            }

        }

    });
}

```

```

RadioGroup rgroup = (RadioGroup) findViewById(R.id.UD2_04_rgrpOptions);
rgroup.setOnCheckedChangeListener(new OnCheckedChangeListener() {

    @Override
    public void onCheckedChanged(RadioGroup group, int checkedId) {
        // TODO Auto-generated method stub
        ImageView imgview;
        VideoView videoview;

        switch(checkedId){
            case R.id.UD2_04_rbbtnFoto:// OCULTAMOS O VIDEO
                imgview = (ImageView) findViewById(R.id.UD2_04_imgvwFoto);
                imgview.setVisibility(View.VISIBLE);
                videoview = (VideoView) findViewById(R.id.UD2_04_vidvwVideo);
                videoview.setVisibility(View.GONE);
                break;
            case R.id.UD2_04_rbbtnVideo:// OCULTAMOS AS FOTOS
                imgview = (ImageView) findViewById(R.id.UD2_04_imgvwFoto);
                imgview.setVisibility(View.GONE);
                videoview = (VideoView) findViewById(R.id.UD2_04_vidvwVideo);
                videoview.setVisibility(View.VISIBLE);
                break;
        }
    }
});
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ud2_05__multimedia_foto_video);

    xestionarEventos();
}
}

```

- Liñas 40-42: Obtemos un obxecto File que apunta ó nome da imaxe gardada. Este dato foi enviado como dato extra ó intent cando lanzamos a aplicación de obter unha foto.
- Liña 45: Obtemos un obxecto da clase BitMap a partires do obxecto File.
- Liñas 49-51: Obtemos un obxecto File que apunta ó nome do vídeo gardado. Este dato foi enviado como dato extra ó intent cando lanzamos a aplicación de obter o vídeo.
- Liña 54: Obtemos a Uri a partires do obxecto File.
- Liña 84: Enviamos como información extra no Intent a ruta onde debe gardar a foto sacada.
- Liña 93: Enviamos como información extra no Intent a ruta onde debe gardar o vídeo sacado.

**Importante:** Como xa comentamos antes, getData() no Intent do método onActivityResult() non trae nada.

## 1.5.2 Información Extra

Como comentamos antes podemos enviar o Intent diversa información (como a ruta onde gardar a foto / vídeo).

Outra información que podemos enviar:

- No caso de captura de fotos, podemos pasarlle o Intent (chamando o método putExtra da clase Intent):

MediaStore.EXTRA\_OUTPUT : Directorio e arquivo (URI) onde se vai gardar a foto. Se non se especifica, o Intent gardará o imaxe cun nome e directorio por defecto que ven especificado no intent de retorno Intent.getData().

- No caso do vídeo, algúns dos datos que podemos enviarlle a Intent son:

MediaStore.EXTRA\_OUTPUT : Directorio e arquivo (URI) onde se vai gardar o vídeo.

Se non se especifica, o Intent gardará o imaxe cun nome e directorio por defecto que ven especificado no intent de retorno `Intent.getData()`.

`MediaStore.EXTRA_VIDEO_QUALITY`: Calidade da imaxe. Se valor 0 sería calidade baixa (para mms) e se vale 1 calidade alta.

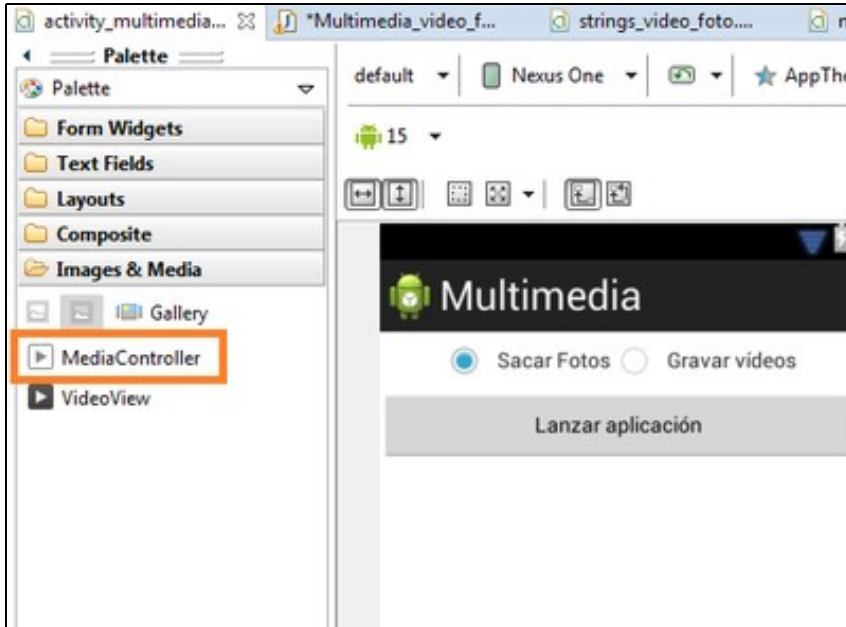
`MediaStore.EXTRA_DURATION_LIMIT`: Nº de segundos máximos para gravación.

`MediaStore.EXTRA_SIZE_LIMIT`: Tamaño máximo do arquivo en bytes.

### 1.5.3 Control para o VideoView

Cando creamos o layout e arrastramos o `VideoView` ó mesmo, podemos facer uso doutro control asociado ó mesmo que permite ter as opcións de 'Pause', 'Stop',...

Este control denomínase `MediaController` e o podemos asociar a un `VideoView` graficamente:



Ou por programación:

```
MediaController controller = new MediaController(this);
```

Con calquera das dúas formas é necesario asocialo a o `VideoView` da forma:

```
VideoView videoview = (VideoView)findViewById(R.id.vidvwVideo);  
videoview.setMediaController(controller);
```

Se o aplicamos á nosa práctica:

```
import android.widget.MediaController;  
  
private void controlVideo() {  
    MediaController controller = new MediaController(this);  
    VideoView videoview = (VideoView)findViewById(R.id.UD2_04_vidvwVideo);  
    videoview.setMediaController(controller);  
}  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_ud2_04__multimedia_foto_video);  
  
    controlVideo();  
    xestionarEventos();  
}
```

## 1.6 Almacenar fotos e vídeos en APIS >=23 e >=24

- Se temos unha API >=23 debemos solicitar o permiso para escribir na SD Card en tempo de execución.
- Se temos unha API >=24 o sistema non deixa compartir por defecto as URLs de referencia a ficheiros entre distintas aplicacións. Nestes casos obténse a excepción **android.os.FileUriExposedException**. No noso caso estaríase compartindo unha referencia a un ficheiro (URI) da nosa aplicación coa aplicación de Android que xestiona a cámara e que lle pasamos ao chamala ao través do intent.
  - ♦ Neste enlace tense máis información sobre esa excepción:  
<https://medium.com/@ali.muzaffar/what-is-android-os-fileuriexposedexception-and-what-you-can-do-about-it-70b9eb17c6d0>
- Para solucionalo creárase un recurso **provider**. A través deste mecanismo poderase xestionar que datos comparte unha aplicación con outra. Neste caso, a nosa app coa aplicación que xestiona a cámara.

### 1.6.1 Creamos a Activity

- Nome do proxecto: **UD2\_05B\_MultimediaFotoVideo**
- Nome da activity: **UD2\_05B\_MultimediaFotoVideo.java**

#### Código do layout xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".UD2_05B_MultimediaFotoVideo">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <RadioGroup
            android:id="@+id/UD2_05B_rgrpOpciones"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_alignParentTop="true"
            android:gravity="center_horizontal"
            android:orientation="horizontal" >

            <RadioButton
                android:id="@+id/UD2_05B_rbtnFoto"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Sacar Foto"
                android:textSize="15sp"
                android:checked="true" />

            <RadioButton
                android:id="@+id/UD2_05B_rbtnVideo"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textSize="15sp"
                android:text="Sacar Vídeo" />

        </RadioGroup>

        <Button
            android:id="@+id/UD2_05B_btnGravarVideoFoto"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_below="@+id/UD2_05B_rgrpOpciones"
            android:textSize="14sp"
```

```

        android:text="Lanzar Cámara" />

<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/UD2_05B_btnGravarVideoFoto">

    <ImageView
        android:id="@+id/UD2_05B_imgvwFoto"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:contentDescription="foto a sacar"
        android:layout_gravity="center"
        />

    <VideoView
        android:id="@+id/UD2_05B_vidvwVideo"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="center"
        android:visibility="gone" />

</FrameLayout>

</RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

## Rutas compartidas que usará provider

- As posibles rutas que se poden usar están no seguinte enlace:  
<https://developer.android.com/reference/android/support/v4/content/FileProvider#SpecifyFiles>
- Nesta aplicación as rutas que se van compartir entre a aplicación e a aplicación que xestiona a cámara son: /sdcard/Movies /sdcard/Pictures.
- Ficheiro: **res/xml/rutas\_provider.xml**

```

<?xml version="1.0" encoding="utf-8"?>
<paths xmlns:android="http://schemas.android.com/apk/res/android">
    <external-path name="external_files_pictures" path="Pictures/" />
    <external-path name="external_files_movies" path="Movies/" />
</paths>

```

## AndroidManifest.xml: Indicar o provider que permitirá compartir datos

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.ud2_05b_multimediafotovideo">

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.UD2_05B_MultimediaFotoVideo">
        <activity android:name=".UD2_05B_MultimediaFotoVideo">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <provider
            android:name="androidx.core.content.FileProvider"
            android:authorities="${applicationId}.provider"
            android:exported="false"

```

```

        android:grantUriPermissions="true">
        <meta-data
            android:name="android.support.FILE_PROVIDER_PATHS"
            android:resource="@xml/rutas_provider" />
    </provider>

</application>

</manifest>

```

- Liña 24: a variable **`\${applicationId}** tomará o valor do atributo **applicationId** do ficheiro **build.gradle**.
- Liñas 28-29: indica que se use o provedor de rutas para compartir e o ficheiro que contén información sobre esas rutas.

## 1.6.2 Código para API<=24

- **getUriForFile**: este método vai facer uso do provider definido no AndroidManifest.xml para que á aplicación da cámara poida usar as rutas compartidas pola nosa aplicación e así poder almacenar as fotos e os vídeos.

### • Clase: **UD2\_05B\_MultimediaFotoVideo**

```

package com.example.ud2_05b_multimediamfotovideo;

import androidx.appcompat.app.AppCompatActivity;

import android.Manifest;
import android.content.pm.PackageManager;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.content.Intent;
import android.graphics.Bitmap;
import android.os.Environment;
import android.provider.MediaStore;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.MediaController;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.RadioGroup.OnCheckedChangeListener;
import android.widget.Toast;
import android.widget.VideoView;

import java.io.File;

import static androidx.core.content.FileProvider.getUriForFile;

//public class UD2_05B_MultimediaFotoVideo extends AppCompatActivity {
public class UD2_05B_MultimediaFotoVideo extends AppCompatActivity {

    //Código para verificar que o resultado ven do intent de gravación de imaxe
    private final int REQUEST_CODE_GRAVACION_FOTO = 1;

    //Código para verificar que o resultado ven do intent de gravación de video
    private final int REQUEST_CODE_GRAVACION_VIDEO = 2;

    final private String nomeVideo="video.mp4";
    final private String nomeFoto="foto.jpg";
    private final int CODIGO_IDENTIFICADOR=1; // Usado por si necesitamos diferentes permisos, para identificar cual de ellos es

    /**
     * Se é API >= 23 pedir permisos de escritura en SDCARD
     */

    public void pedirPermiso() {

```

```

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M)
            requestPermissions( new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE},CODIGO_IDENTIFICADOR);
    }

    /**
     * Activar o botón Lanzar Cámara se hai permisos.
     */
    @Override
    public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {

        switch (requestCode) {
            case CODIGO_IDENTIFICADOR: {
                // Se o usuario premeu o boton de cancelar o array volve cun null
                Button btn = findViewById(R.id.UD2_05B_btnGravarVideoFoto);
                if (grantResults.length > 0
                    && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                    btn.setEnabled(true);
                    // PERMISO CONCEDIDO
                } else {
                    // PERMISO DENEGADO
                    btn.setEnabled(false); //Desabilitamos o botón se non temos o permiso de escritura
                    Toast.makeText(this,"É NECESARIO O PERMISO DE ESCRITURA NA SDCARD",Toast.LENGTH_LONG).show();
                }
                return;
            }
            // Comprobamos outros permisos
        }
    }

    /**
     * Para introducirle controis de avance/retroceso/pause/play ao á view do vídeo
     */
    private void controlVideo(){
        MediaController controller = new MediaController(this);
        VideoView videoview = (VideoView)findViewById(R.id.UD2_05B_vidvwVideo);
        videoview.setMediaController(controller);
    }

    /**
     * Obtemos a foto ou vídeo que vén da aplicación Android
     */
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (resultCode == RESULT_OK) {

            File ruta, arquivo;
            switch (requestCode) {
                case REQUEST_CODE_GRAVACION_FOTO:
                    ruta = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES);
                    arquivo = new File(ruta, nomeFoto);
                    if (!arquivo.exists()) return; // Non hai foto

                    ImageView imgview = (ImageView) findViewById(R.id.UD2_05B_imgvwFoto);
                    Bitmap bitmap = BitmapFactory.decodeFile(arquivo.getAbsolutePath());
                    imgview.setImageBitmap(bitmap);
                    break;
                case REQUEST_CODE_GRAVACION_VIDEO:
                    ruta = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_MOVIES);
                    arquivo = new File(ruta, nomeVideo);
                    if (!arquivo.exists()) return; // Non hai vídeo

                    VideoView vidview = (VideoView) findViewById(R.id.UD2_05B_vidvwVideo);
                    vidview.setVideoURI(Uri.fromFile(arquivo));
                    vidview.start();
                    break;
            }
        } else if (resultCode == RESULT_CANCELED) {
            // Video ou Foto cancelada
        } else {
            // Fallo na captura do Video ou foto.
        }
    }

```

```

}

/**
 * Programa o código dos click's os botões
 */
private void xestionarEventos(){
    Button gravar = (Button)findViewById(R.id.UD2_05B_btnGravarVideoFoto);
    gravar.setOnClickListener(new OnClickListener(){

        @Override
        public void onClick(View arg0) {
            // TODO Auto-generated method stub
            RadioButton rb = (RadioButton)findViewById(R.id.UD2_05B_rbtnFoto);

            if (rb.isChecked()){ // Sacar foto
                File ruta = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES);
                File archivo = new File(ruta,nomeFoto);

                Uri contentUri=null;
                if (Build.VERSION.SDK_INT >= 24) {
                    contentUri = getUriForFile(getApplicationContext(), getApplicationContext().getPackageName() + ".provider", archivo);
                }
                else {
                    contentUri = Uri.fromFile(archivo);
                }

                Intent intento = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                intento.putExtra(MediaStore.EXTRA_OUTPUT, contentUri);
                startActivityForResult(intento, REQUEST_CODE_GRAVACION_FOTO);
            }

            else { // Grava vídeo
                File ruta = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_MOVIES);
                File archivo = new File(ruta,nomeVideo);

                Uri contentUri=null;
                if (Build.VERSION.SDK_INT >= 24) {
                    contentUri = getUriForFile(getApplicationContext(), getApplicationContext().getPackageName() + ".provider", archivo);
                }
                else {
                    contentUri = Uri.fromFile(archivo);
                }

                Intent intento = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
                intento.putExtra(MediaStore.EXTRA_OUTPUT, contentUri);
                startActivityForResult(intento, REQUEST_CODE_GRAVACION_VIDEO);
            }
        }
    });

    RadioGroup rgroup = (RadioGroup)findViewById(R.id.UD2_05B_rgrpOptions);
    rgroup.setOnCheckedChangeListener(new OnCheckedChangeListener() {

        @Override
        public void onCheckedChanged(RadioGroup group, int checkedId) {
            // TODO Auto-generated method stub
            ImageView imgview;
            VideoView videoview;

            switch(checkedId){
                case R.id.UD2_05B_rbtnFoto: // OCULTAMOS O VIDEO
                    imgview = (ImageView)findViewById(R.id.UD2_05B_imgvwFoto);
                    imgview.setVisibility(View.VISIBLE);
                    videoview = (VideoView)findViewById(R.id.UD2_05B_vidvwVideo);
                    videoview.setVisibility(View.GONE);
                    break;
                case R.id.UD2_05B_rbtnVideo: // OCULTAMOS AS FOTOS
                    imgview = (ImageView)findViewById(R.id.UD2_05B_imgvwFoto);
                    imgview.setVisibility(View.GONE);
                    videoview = (VideoView)findViewById(R.id.UD2_05B_vidvwVideo);

```



```

        videoview.setVisibility(View.VISIBLE);
        break;
    }
}
});
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ud2_05b__multimedia_foto_video);
    pedirPermiso();
    controlVideo();
    xestionarEventos();
}
}

```

- Liñas 48-52\*: Solicítase permiso para escribir (e por tanto ler) no almacenamento externo.
- Liñas 58-77\*: Se hai permisos para escribir na SD CARD habilítase o botón que lanza a cámara.
- Liñas 83-87\*: Actívanse os controis de reprodución de vídeo (avanzar, retroceder, pausar e continuar).
- Liñas 139,157: cada unha destas liñas recolle o directorio correspondente ás fotos ou aos vídeos da SDCARD, respectivamente.
- Liñas 142-145 e 160-164: No caso de que a API $\geq$ 24 entón a uri fórmase tendo en conta o provider definido no AndroidManifest.xml para poder compartir datos entre distintas aplicacións: a nosa e a que xestiona a cámara.

## 1.7 Métodos útiles no manexo de Imaxes

- **Bitmap.createScaledBitmap(bitmaporiginal, width, height, boolean filter)**: devolve un obxecto da clase Bitmap e vains servir para escalar un bitmap a outro tamaño indicado por width e height.

O filter debería ser posto a true se facemos a imaxe máis grande e a false se a facemos máis pequena. Para determinar o tamaño adecuado do Bitmap en función dos puntos por polgada do dispositivo onde nos atopamos podemos facer uso da propiedade densityDpi:

```

switch (getResources().getDisplayMetrics().densityDpi) {
    case DisplayMetrics.DENSITY_LOW:
        break;
    case DisplayMetrics.DENSITY_MEDIUM:
        break;
    case DisplayMetrics.DENSITY_HIGH:
        break;
    case DisplayMetrics.DENSITY_XHIGH:
        break;
}

```

En función do valor de densidade escalaremos a imaxe a un tamaño adecuado tendo en conta a proporción:

- xhdpi: 2.0
- hdpi: 1.5
- mdpi: 1.0 (baseline)
- ldpi: 0.75

Nota: Normalmente cando facemos fotos e queremos visualizalas todas xuntas (por exemplo utilizando un horizontalScrollView) faise uso deste método para ter unha imaxe en pequeno da foto e despois cargar a imaxe orixinal cando a visualicemos a pantalla completa.

- **Obxectobitmap.compress(CompressFormat.JPEG, 100,os)**: comprime un bitmap e o escribe a disco utilizando un FileOutputStream. O primeiro parámetro indica o tipo de compresión (se temos transparencias na imaxe orixinal deberemos usar BMP), o segundo indica o nivel de compresión (se é 50 sería o 50%, no exemplo o deixaría tal cal) e o terceiro parámetro é un obxecto da clase FileOutputStream para pasalo a disco.

Este método pode servírnos para pasar un bitmap a disco utilizando un obxecto da clase File (o veremos na unidade de Datos Persistentes) e obtendo o FileOutputStream a partires del. É importante asinarlle unha extensión ó arquivo que vai ser a imaxe gardada.

Se a foto a temos asinada a un ImageView e queremos recuperala coma un obxecto da clase Bitmap teremos que facer:

```
imageView.setDrawingCacheEnabled(true);  
Bitmap imaxe = imageView.getDrawingCache();
```

Sendo imageView o obxecto que fai referencia ó ImageView.

O proceso contrario, é dicir, cargar unha foto nun ImageView será o seguinte (o veremos outra vez na unidade de datos persistentes):

```
Bitmap bitmap = BitmapFactory.decodeFile(Obxecto_file_apuntando_á_imaxe_gardada);  
img.setImageBitmap(bitmap);
```

Sendo img un obxecto da clase ImageView.

## 1.8 Método útil no manexo de Vídeos

Pode resultarnos necesario a partires dun vídeo obter unha foto del.

Para facelo debemos de utilizar a clase [ThumbnailUtils](#)

Vexamos un exemplo:

```
Bitmap preview_bitmap = ThumbnailUtils.createVideoThumbnail(uri_al_video.toString(), Thumbnails.MICRO_KIND);
```

## 1.9 Visualización de múltiples imaxes

Pode darse o caso de que necesitemos visualizar múltiples imaxes na nosa activity.

Nota: Como está comentado en liñas anteriores, o lóxico será ter unha versión 'en pequeno' da foto para non encher a memoria utilizando o método compress da clase Bitmap ou ben utilizando o método `Bitmap.createScaledBitmap`, como está feito no exemplo que ven a continuación.

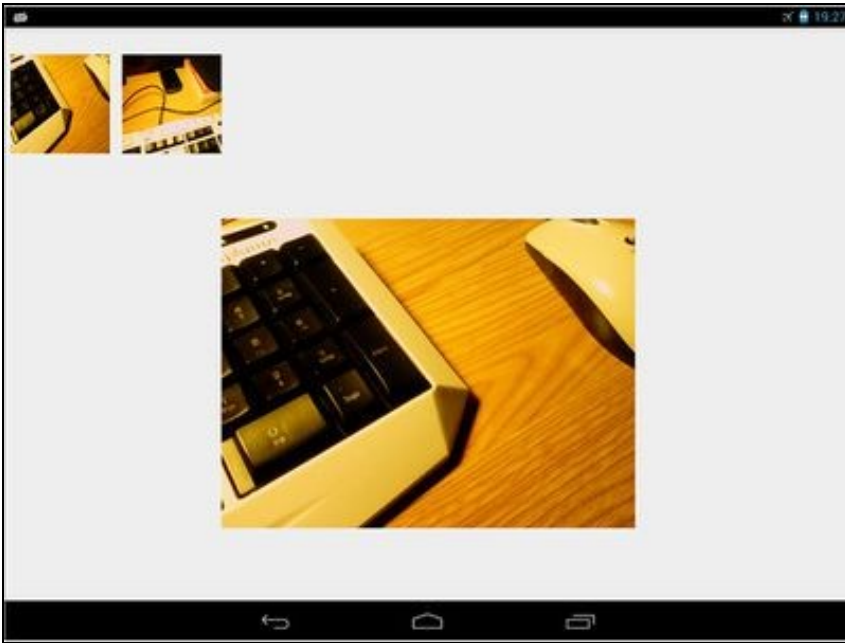
Temos varias posibilidades para facer isto:

- [Control Gallery](#): deprecated a partires da API 16.
- [Control HorizontalScrollView](#).
- [Control ViewPager](#).

Neste apartado imos ver como utilizar un **HorizontalScrollView**.

### 1.9.1 Caso Práctico

O obxectivo desta práctica é ver como podemos utilizar o HorizontalScrollView para visualizar imaxes en pequeno.



Como o seu propio nome indica, o `HorizontalScrollView` é un control que permite facer un scroll horizontal de `View`'s.

A idea é ter dentro deste control un `LinearLayout` coas imaxes.

Cada imaxe vai ter un layout cun tamaño (o tamaño da miniatura) e xestionaremos o evento de Click sobre a imaxe en miniatura para ver dita imaxe nun `ImageView` en grande.

### 1.9.1.1 Preparación

Será necesario copiar ó cartafol da SD onde o S.O. Android garda as imaxes por defecto, dúas imaxes calquera de nome **foto1.jpg** e **foto2.jpg**.

Utilizamos a clase `Environment` (`Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES)`) para indicar a ruta onde se atopan as fotos.

Normalmente esta ruta se atopa en `/sdcard/Pictures`.

- Se utilizades o emulador podeades copiar as fotos utilizando a perspectiva DDMS.
- Se estades a utilizar un dispositivo real teredes que copiar as fotos conectando o dispositivo ó computador.

O alumno é libre de utilizar outra ruta calquera modificando o código convenientemente.

### 1.9.1.2 Creamos a Activity

- Nome do proxecto: **UD2\_06\_MultimediaHorizontalScrollView**
- Nome da activity: **UD2\_06\_MultimediaHorizontalScrollView.java**

### Código do layout xml

```
<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"
    tools:context="${relativePackage}.${activityClass}" >

    <HorizontalScrollView
        android:id="@+id/UD2_06_hsvviewFotosPai"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
```

```

        android:layout_marginTop="26dp" >

        <LinearLayout
            android:id="@+id/UD2_06_linearLayoutFotosFillo"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal" />

    </HorizontalScrollView>

    <ImageView
        android:id="@+id/UD2_06_imgvwFotoAmpliada"
        android:layout_width="500dp"
        android:layout_height="500dp"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="26dp"
        android:src="@drawable/ic_launcher" />

</RelativeLayout>

```

Como vemos temos un HorizontalScrollView e un LinearLayout dentro do mesmo.

### Código da clase UD2\_06\_MultimediaHorizontalScrollView

**Obxectivo:** Carga nun HorizontalScrollView dúas fotos gardadas na tarxeta SD do dispositivo.

**Nota:** Vos dará un erro xa que fai uso dunha clase explicada a continuación.

```

import java.io.File;

import android.app.Activity;
import android.os.Bundle;
import android.os.Environment;
import android.widget.LinearLayout;

public class UD2_06_MultimediaHorizontalScrollView extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ud2_06__multimedia_horizontal_scroll_view);

        LinearLayout linear = (LinearLayout) findViewById(R.id.UD2_06_linearLayoutFotosFillo);

        String ruta = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES).getAbsolutePath();

        UD2_06_ImaxeHorizontalScrollView imaxe = new UD2_06_ImaxeHorizontalScrollView(this, ruta+File.separator+"foto1.jpg");
        linear.addView(imaxe.getLayout());
        imaxe = new UD2_06_ImaxeHorizontalScrollView(this, ruta+File.separator+"foto2.jpg");
        linear.addView(imaxe.getLayout());

    }
}

```

- Liña 15: Obtemos a referencia ó LinearLayout que se atopa dentro do HorizontalScrollView.
- Liña 17: Ruta onde se atopan as imaxes.
- Liña 19,21: Creamos dous obxectos da clase UD2\_06\_ImaxeHorizontalScrollView (explicada posteriormente).
- Liñas 20,22: Engadimos ditas imaxes ó LinearLayout.

#### 1.9.1.3 Creamos o ImageView personalizado

Agora imos explicar a clase UD2\_06\_ImaxeHorizontalScrollView.

A idea é moi sinxela.

Vou engadir ó LinearLayout obxectos da clase ImageView.

A diferenza é que vou personalizar ditos obxectos para que estean (a imaxe) dentro dun Layout creado por min y cun tamaño concreto (o veremos no código).

Ademais xestionaremos o evento Click sobre a imaxe para facer que apareza en grande no ImageView da Activity principal.

O que hai que ter moi claro é que esta clase representa cada unha das imaxes en miniatura que vemos dentro do control HorizontalScrollView.

### Código da clase UD2\_06\_ImaxeHorizontalScrollView

**Obxectivo:** Facemos un ImageView personalizado cun tamaño específico e xestionamos o evento Click sobre a imaxe.

```
import java.io.File;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.view.Gravity;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup.LayoutParams;
import android.widget.ImageView;
import android.widget.LinearLayout;

public class UD2_06_ImaxeHorizontalScrollView extends ImageView implements OnClickListener {
    protected UD2_06_MultimediaHorizontalScrollView context;
    private File file;
    private String path;
    private LinearLayout layout;

    /**
     * Crea unha imaxe e a engade nun layout que vai ser o que visualice o HorizontalView
     * @param context
     * @param path: ruta a imaxe
     */
    public UD2_06_ImaxeHorizontalScrollView(UD2_06_MultimediaHorizontalScrollView context,String path) {
        super(context);
        this.context=context;
        this.path=path;
        file = new File(path);

        setOnClickListener(this);
        crearLayout();
    }

    private void crearLayout() {
        LinearLayout layout = new LinearLayout(context);
        layout.setLayoutParams(new LayoutParams(135, 135));
        layout.setGravity(Gravity.CENTER);

        this.setLayoutParams(new LayoutParams(120, 120)); // TAMAÑO DA IMAXE
        this.setScaleType(ImageView.ScaleType.CENTER_CROP);
        Bitmap bit = BitmapFactory.decodeFile(file.getAbsolutePath());
        if (bit==null) { // Non a pode cargar
            this.layout=layout;
            return;
        }

        Bitmap bitmap;
        bitmap = Bitmap.createScaledBitmap(bit, 220, 220,false);
        this.setImageBitmap(bitmap);

        layout.addView(this);
        this.layout = layout;
    }

    public LinearLayout getLayout(){
        return layout;
    }

    public File getFile(){
        return file;
    }

    public String getPath(){
        return path;
    }

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
```

```

Bitmap bp = BitmapFactory.decodeFile(getPath());
ImageView imgview = (ImageView) context.findViewById(R.id.UD2_06_imgvwFotoAmpliada);
imgview.setImageBitmap(bp);
}

}

```

- Liña 12: Fixarse como facemos unha clase que deriva de ImageView e implementa a interface OnClickListener para xestionar o evento click sobre a imaxe
- Liña 23: No constructor recibimos unha instancia da clase que utiliza o ImageView. Isto é necesario xa que cando prememos sobre a imaxe en miniatura temos que atopar o ImageView da activiry para cargar a imaxe en grande. Como segundo parámetro enviamos a ruta ó arquivo de imaxe.
- Liñas 25-27: Gardamos os datos enviados.
- Liña 29: Facemos que o evento de click se xestione dentro da clase.
- Liña 30: Chamamos ó método que vai crea o Layout onde vai ir a imaxe.
- Liñas 33-51: Creamos o layout (LinearLayout) que vai albergar a foto.

- ◊ Liñas 34-36: Layout que alberga a imaxe.
- ◊ Liñas 38-39: Tamaño e tipo de escala da imaxe dentro do layout.
- ◊ Liña 40-47: Cargamos a imaxe dende a SD Externa e a asignamos ó ImageView.
- ◊ Liñas 48-51: Engadimos o layout creado o ImageView cargado de disco.

- Liñas 64-71: Xestionamos o evento do Click sobre a imaxe. Buscamos o control ImageView da activity principal (UD2\_06\_imgvwFotoAmpliada) e cargamos a partires do path a imaxe no ImageView.