

PDM Avanzado Captura de Vídeo / Imaxes

Sumario

Introdución

Captura de Fotos / Vídeos

Accedendo ó Vídeo / Foto

- Caso Práctico
- Creamos a Activity

Gardando o Vídeo / Foto API < 23

- Caso Práctico
- Creamos a Activity
- Información Extra
- Control para o VideoView

Gardando o Vídeo / Foto. Caso especial API >= 23 e API >= 24

Métodos útiles no manexo de Imaxes

Métodos útil no manexo de Vídeos

Visualización de múltiples imaxes

- Caso Práctico
- Preparación
- Creamos a Activity
- Creamos o ImageView personalizado

Introdución

As clases que interveñen na gravación son:

- Clase Camera (<http://developer.android.com/reference/android/hardware/Camera.html>): danos acceso á cámara, as súas características. Usarémosla se facemos unha aplicación na que queiramos xestionar a cámara por nos mesmos.
- Clase SurfaceView (<http://developer.android.com/reference/android/view/SurfaceView.html>): para previsualizar o que imos gravar.
- Clase MediaRecorder (<http://developer.android.com/reference/android/media/MediaRecorder.html>): 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).

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

- Permiso para facer uso da cámara:

```
1 <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:

```
1 <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:

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

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

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

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

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

IMPORTANTE: A partires da API 23 (Android 6), Android incorpora outro nivel de seguridade, de tal forma que agora, a maiores, teremos que solicitar dito permiso a nivel de programación. Máis información de como facelo [neste enlace](http://wiki.cifprodolfoucha.es/index.php?title=PDM_Avanzado_Permisos_AndroidManifest.xml) (http://wiki.cifprodolfoucha.es/index.php?title=PDM_Avanzado_Permisos_AndroidManifest.xml).

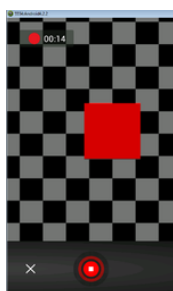
Captura de Fotos / Vídeos

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` (http://developer.android.com/reference/android/provider/MediaStore.html#ACTION_IMAGE_CAPTURE) : Captura unha imaxe.
 - `MediaStore.ACTION_VIDEO_CAPTURE` (http://developer.android.com/reference/android/provider/MediaStore.html#ACTION_VIDEO_CAPTURE) : Captura un vídeo.

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

- Chamar ó método `startActivityResult()` (<http://developer.android.com/reference/android/app/Activity.html#startActivityResult%28android.content.Intent,%20int%29>).

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).

```
1 startActivityResult(intento, REQUEST_CODE_GRAVACION_OK);
```

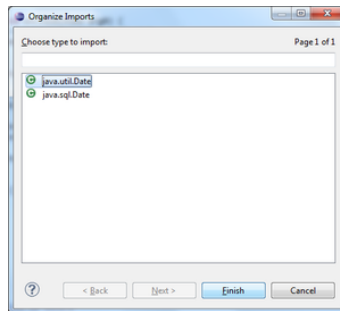
- Obter o resultado da aplicación lanzada. Para iso temos que desenrolar o método `onActivityResult()` (<http://developer.android.com/reference/android/app/Activity.html#onActivityResult%28int,%20int,%20android.content.Intent%29>) da nosa Activity.

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.

Nota: Para asinar 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:

```
1 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
2
3     Bitmap bitMap= data.getExtras().get("data");
4     .....
5 }
```

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

```
1 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
2
3     Bitmap bitMap= data.getExtras().get("data");
4     imageView.setImageBitmap(bitMap);
5 }
```

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.

```
1 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
2
3     Uri uri = data.getData();
4     .....
5 }
```

Para visualizalo necesitamos un obxecto da clase `VideoView`.

```

1 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
2
3     vidView.setVideoURI(data.getData());
4     vidView.start();
5 }

```

Sendo vidView un obxecto da clase VideoView.

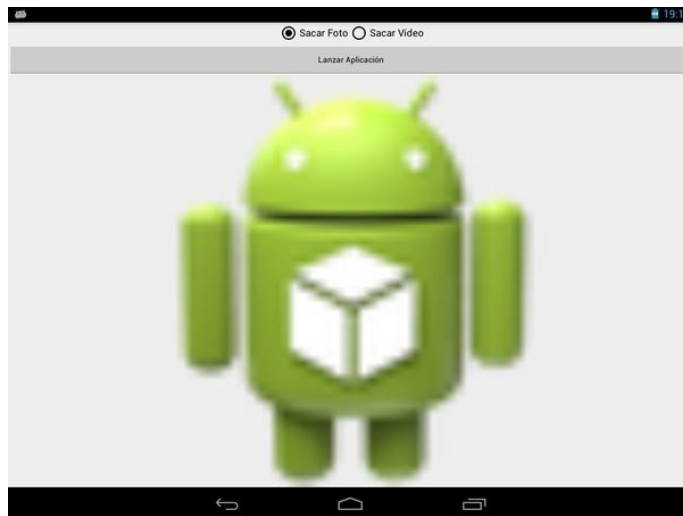
Accedendo ó Vídeo / Foto

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().

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").

Creamos a Activity

- Nome do proxecto: **UD6_04_MultimediaFotoVideo**
- Nome da activity: **UD6_04_MultimediaFotoVideo.java**

Código do layout xml

```

1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent" >
5
6   <RadioGroup
7       android:id="@+id/UD6_04_rgrpOpciones"
8       android:layout_width="match_parent"
9       android:layout_height="wrap_content"
10      android:layout_alignParentLeft="true"
11      android:layout_alignParentTop="true"
12      android:gravity="center_horizontal"
13      android:orientation="horizontal" >
14
15      <RadioButton
16          android:id="@+id/UD6_04_rbtnFoto"
17          android:layout_width="wrap_content"
18          android:layout_height="wrap_content"
19          android:text="Sacar Foto"
20          android:textSize="15sp"
21          android:checked="true" />

```

```

22
23     <RadioButton
24         android:id="@+id/UD6_04_rbtnVideo"
25         android:layout_width="wrap_content"
26         android:layout_height="wrap_content"
27         android:textSize="15sp"
28         android:text="Sacar Video" />
29 </RadioGroup>
30 <Button
31     android:id="@+id/UD6_04_btnGravarVideoFoto"
32     android:layout_width="match_parent"
33     android:layout_height="wrap_content"
34     android:layout_alignParentLeft="true"
35     android:layout_below="@+id/UD6_04_rgrpOptions"
36     android:textSize="14sp"
37     android:text="Lanzar Aplicación" />
38
39
40 <FrameLayout
41     android:layout_width="match_parent"
42     android:layout_height="match_parent"
43     android:layout_alignParentLeft="true"
44     android:layout_below="@+id/UD6_04_btnGravarVideoFoto">
45
46     <ImageView
47         android:id="@+id/UD6_04_imgvwFoto"
48         android:layout_width="match_parent"
49         android:layout_height="match_parent"
50         android:contentDescription="foto a sacar"
51         android:layout_gravity="center"
52         android:src="@drawable/ic_launcher" />
53
54     <VideoView
55         android:id="@+id/UD6_04_vidvwVideo"
56         android:layout_width="match_parent"
57         android:layout_height="match_parent"
58         android:layout_gravity="center"
59         android:visibility="gone" />
60
61 </FrameLayout>
62
63 </RelativeLayout>

```

Código da clase UD6_04_MultimediaFotoVideo

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

```

1 import android.app.Activity;
2 import android.content.Intent;
3 import android.graphics.Bitmap;
4 import android.os.Bundle;
5 import android.provider.MediaStore;
6 import android.view.View;
7 import android.view.View.OnClickListener;
8 import android.widget.Button;
9 import android.widget.ImageView;
10 import android.widget.RadioGroup;
11 import android.widget.RadioGroup.OnCheckedChangeListener;
12 import android.widget.Toast;
13 import android.widget.VideoView;
14
15
16 public class UD6_04_MultimediaFotoVideo extends Activity {
17
18     /**
19      * Código para verificar que o resultado ven do intent de gravación
20      */
21     private final int REQUEST_CODE_GRAVACION_OK = 1;
22
23
24     /**
25      * Obtemos a imaxe ou video que ven da aplicacion Android
26      */
27     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
28
29         if (resultCode == RESULT_OK) {
30
31             File ruta,archivo;
32             switch(requestCode) {
33                 case REQUEST_CODE_GRAVACION_IMAXE:
34                     ImageView imgview = (ImageView) findViewById(R.id.UD6_04_imgvwFoto);
35                     imgview.setImageBitmap((Bitmap) data.getExtras().get("data"));
36
37                     break;
38
39                 case REQUEST_CODE_GRAVACION_VIDEO:
40                     VideoView vidview = (VideoView) findViewById(R.id.UD6_04_vidvwVideo);
41                     vidview.setVideoURI(data.getData());
42                     vidview.start();

```

```

43
44         break;
45     }
46 } else if (resultCode == RESULT_CANCELED) {
47
48     // Video ou Foto cancelada
49 } else {
50
51     // Fallo na captura do Video ou foto.
52 }
53 }
54 }
55
56 /**
57  * Programa o código dos click's os botóns
58  */
59 private void xestionarEventos(){
60     Button gravar = (Button)findViewById(R.id.UD6_04_btnGravarVideoFoto);
61     gravar.setOnClickListener(new View.OnClickListener(){
62
63         @Override
64         public void onClick(View arg0) {
65             // TODO Auto-generated method stub
66             RadioButton rb = (RadioButton)findViewById(R.id.UD6_04_rbbtnFoto);
67
68             if (rb.isChecked()){ // Sacar foto
69
70                 Intent intento = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
71                 startActivityForResult(intento, REQUEST_CODE_GRAVACION_IMAXE);
72
73             }
74             else { // Grava vídeo
75                 Intent intento = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
76                 startActivityForResult(intento, REQUEST_CODE_GRAVACION_VIDEO);
77
78             }
79         }
80     }
81 }
82
83 });
84
85 RadioGroup rgroup = (RadioGroup)findViewById(R.id.UD6_04_rgrpOptions);
86 rgroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
87
88     @Override
89     public void onCheckedChanged(RadioGroup group, int checkedId) {
90         // TODO Auto-generated method stub
91         ImageView imgview;
92         VideoView videoview;
93
94         switch(checkedId){
95             case R.id.UD6_04_rbbtnFoto: // OCULTAMOS O VIDEO
96                 imgview = (ImageView)findViewById(R.id.UD6_04_imgvwFoto);
97                 imgview.setVisibility(View.VISIBLE);
98                 videoview = (VideoView)findViewById(R.id.UD6_04_vidvwVideo);
99                 videoview.setVisibility(View.GONE);
100                 break;
101             case R.id.UD6_04_rbbtnVideo: // OCULTAMOS AS FOTOS
102                 imgview = (ImageView)findViewById(R.id.UD6_04_imgvwFoto);
103                 imgview.setVisibility(View.GONE);
104                 videoview = (VideoView)findViewById(R.id.UD6_04_vidvwVideo);
105                 videoview.setVisibility(View.VISIBLE);
106                 break;
107         }
108     }
109 });
110 }
111
112
113 @Override
114 protected void onCreate(Bundle savedInstanceState) {
115     super.onCreate(savedInstanceState);
116     setContentView(R.layout.activity_ud6_04__multimedia_foto_video);
117
118     xestionarEventos();
119 }
120 }

```

- **Liñas 34-35:** Nestas liñas obtemos a foto da aplicación do S.O. e a visualiza no ImageView.
- **Liñas 40-42:** Nestas liñas obtemos o vídeo do Intent e o cargamos no VideoView e comezamos a reprodución.
- **Liñas 71-72:** 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 76-77:** Evento click sobre o botón. Se está a opción do RadioButton de sacar un vídeo lanzamos o Intent ACTION_VIDEO_CAPTURE, esperando o resultado.

Gardando o Vídeo / Foto API < 23

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.**

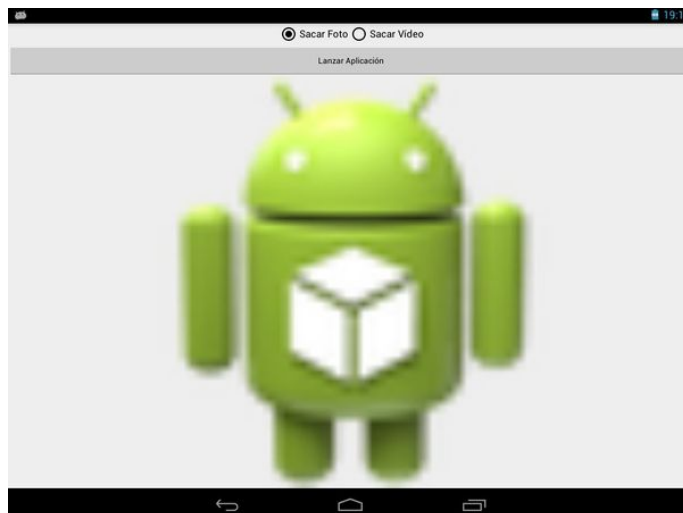
- **NOTA IMPORTANTE:** Neste caso imos a enviar a ruta onde se vai gardar a foto/vídeo. **NON PODEMOS EMPREAGAR A SD INTERNA**, como a ruta que devolve `getFilesDir()` (https://wiki.cifprodolfoucha.es/index.php?title=PDM_Avanzado_Datos_Persistentes_Arquivos#A_clase_File_e_a_clase_Environment) xa que quen vai gardar a foto é unha aplicación externa e polo tanto non ten permiso.

Se queremos gardala nese cartafol temos esta alternativa:

<https://stackoverflow.com/questions/7720383/camera-intent-not-saving-photo>

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.

Creamos a Activity

- Nome do proxecto: **UD6_05_MultimediaFotoVideo**
- Nome da activity: **UD6_05_MultimediaFotoVideo.java**
- O código do layout vai ser o mesmo que na activity anterior.

Código da clase UD6_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.

```
1 import java.io.File;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.graphics.Bitmap;
6 import android.graphics.BitmapFactory;
7 import android.net.Uri;
8 import android.os.Bundle;
9 import android.os.Environment;
10 import android.provider.MediaStore;
11 import android.view.View;
12 import android.view.View.OnClickListener;
13 import android.widget.Button;
14 import android.widget.ImageView;
15 import android.widget.RadioButton;
16 import android.widget.RadioGroup;
17 import android.widget.RadioGroup.OnCheckedChangeListener;
18 import android.widget.VideoView;
19
20 public class UD6_05_MultimediaFotoVideo extends Activity {
21
22     /**
```

```

23  * Código para verificar que o resultado ven do intent de gravación
24  */
25  private final int REQUEST_CODE_GRAVACION_IMAXE = 1;
26  private final int REQUEST_CODE_GRAVACION_VIDEO = 2;
27
28  final private String nomeVideo="video.mp4";
29  final private String nomeFoto="foto.jpg";
30
31
32  /**
33   * Obtemos a imaxe ou video que ven da aplicacion Android
34   */
35  protected void onActivityResult(int requestCode, int resultCode, Intent data) {
36
37      if (resultCode == RESULT_OK) {
38
39          File ruta,archivo;
40          switch(requestCode) {
41              case REQUEST_CODE_GRAVACION_IMAXE:
42                  ruta = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES);
43                  archivo = new File(ruta, nomeFoto);
44                  if (!archivo.exists()) return; // Non hai foto
45
46                  ImageView imgview = (ImageView) findViewById(R.id.UD6_04_imgvwFoto);
47                  Bitmap bitmap = BitmapFactory.decodeFile(archivo.getAbsolutePath());
48                  imgview.setImageBitmap(bitmap);
49                  break;
50              case REQUEST_CODE_GRAVACION_VIDEO:
51                  ruta = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_MOVIES);
52                  archivo = new File(ruta, nomeVideo);
53                  if (!archivo.exists()) return; // Non hai foto
54
55                  VideoView vidview = (VideoView) findViewById(R.id.UD6_04_vidvwVideo);
56                  vidview.setVideoURI(Uri.fromFile(archivo));
57                  vidview.start();
58                  break;
59
60          }
61      } else if (resultCode == RESULT_CANCELED) {
62
63          // Video ou Foto cancelada
64      } else {
65
66          // Fallo na captura do Video ou foto.
67      }
68  }
69
70  /**
71   * Programa o código dos click's os botóns
72   */
73  private void xestionarEventos(){
74      Button gravar = (Button)findViewById(R.id.UD6_04_btnGravarVideoFoto);
75      gravar.setOnClickListener(new View.OnClickListener(){
76
77          @Override
78          public void onClick(View arg0) {
79              // TODO Auto-generated method stub
80              RadioButton rb = (RadioButton)findViewById(R.id.UD6_04_rbtnFoto);
81
82
83              if (rb.isChecked()){ // Saca foto
84                  File ruta = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES);
85                  File archivo = new File(ruta,nomeFoto);
86
87                  Intent intento = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
88                  intento.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(archivo));
89                  startActivityForResult(intento, REQUEST_CODE_GRAVACION_IMAXE);
90
91              }
92              else { // Grava vídeo
93                  File ruta = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_MOVIES);
94                  File archivo = new File(ruta,nomeVideo);
95
96                  Intent intento = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
97                  intento.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(archivo));
98
99                  startActivityForResult(intento, REQUEST_CODE_GRAVACION_VIDEO);
100
101              }
102
103          }
104      });
105
106      RadioGroup rgroup = (RadioGroup)findViewById(R.id.UD6_04_rgrpOptions);
107      rgroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
108
109          @Override
110          public void onCheckedChanged(RadioGroup group, int checkedId) {
111              // TODO Auto-generated method stub
112              ImageView imgview;
113              VideoView videoview;
114
115              switch(checkedId){
116

```



```

117         case R.id.UD6_04_rbtnFoto: // OCULTAMOS O VIDEO
118             imgview = (ImageView)findViewById(R.id.UD6_04_imgvwFoto);
119             imgview.setVisibility(View.VISIBLE);
120             videoview = (VideoView)findViewById(R.id.UD6_04_vidvwVideo);
121             videoview.setVisibility(View.GONE);
122             break;
123         case R.id.UD6_04_rbtnVideo: // OCULTAMOS AS FOTOS
124             imgview = (ImageView)findViewById(R.id.UD6_04_imgvwFoto);
125             imgview.setVisibility(View.GONE);
126             videoview = (VideoView)findViewById(R.id.UD6_04_vidvwVideo);
127             videoview.setVisibility(View.VISIBLE);
128             break;
129     }
130 }
131 });
132 }
133
134
135 @Override
136 protected void onCreate(Bundle savedInstanceState) {
137     super.onCreate(savedInstanceState);
138     setContentView(R.layout.activity_ud6_05_multimedia_foto_video);
139
140     xestionarEventos();
141 }
142 }

```

- Liñas 42-43: 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 47: Obtemos un obxecto da clase BitMap a partires do obxecto File.
- Liñas 51-52: 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 56: Obtemos a Uri a partires do obxecto File.
- Liña 88: Enviamos como información extra no Intent a ruta onde debe gardar a foto sacada.
- Liña 97: 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.

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.

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 (<http://developer.android.com/reference/android/widget/MediaController.html>).

- Para asocialo a un `VideoView`:

```
1 MediaController controller = new MediaController(this);
2 VideoView videoview = (VideoView)findViewById(R.id.videovideo);
3
4 videoview.setMediaController(controller);
```

- Se o aplicamos á nosa práctica:

```
1 import android.widget.MediaController;
2
3 private void controlVideo(){
4     MediaController controller = new MediaController(this);
5     VideoView videoview = (VideoView)findViewById(R.id.UD6_04_videovideo);
6     videoview.setMediaController(controller);
7
8 }
9
10 @Override
11 protected void onCreate(Bundle savedInstanceState) {
12     super.onCreate(savedInstanceState);
13     setContentView(R.layout.activity_ud6_04_multimedia_foto_video);
14
15     controlVideo();
16     xestionarEventos();
17 }
```

Gardando o Vídeo / Foto. Caso especial API >= 23 e API >= 24

- Como comentamos antes, se queremos gardar unha imaxe/vídeo nun cartafol na SDCard Externa e temos unha API >= 23, temos que solicitar dito permiso por programación.

Este punto xa está explicado neste punto da Wiki ([http://wiki.cifprodolfoucha.es/index.php?title=PDM_Avanzado o Permisos AndroidManifest.xml](http://wiki.cifprodolfoucha.es/index.php?title=PDM_Avanzado_o_Permisos_AndroidManifest.xml)).

Neste caso necesitaríamos pedir permiso para escribir na tarxeta SDCard (ao facelo xa temos o permiso de lectura tamén). Mirar como facelo no enlace anterior.

NOTA IMPORTANTE: Neste exemplo, a activity deriva da clase Activity. Lembrar que se o facedes cunha clase que derive da clase AppCompatActivity a chamada a `requestPermissions` é diferente. Mirade o enlace anterior.

- Pero a maiores temos un problema, xa que se executamos o código anterior nun S.O. Android cunha API >=24 daranos unha excepción `android.os.FileUriExposedException`.

Isto é debido a que a partires desta versión xa non podemos empregar unha Uri como forma de envío á aplicación da cámara, xa que esta Uri está sendo compartida pola nosa aplicación e a aplicación da cámara e a partires desa versión non está permitido.

Para solucionalo temos que crear un provider (<https://developer.android.com/guide/topics/providers/content-providers?hl=es-419>) (un recurso de Android que permite xestionar o acceso a información dende diferentes aplicacións, permitindo o acceso doutras aplicación para recuperar o gardar información na aplicación que ofrece o provider).

- **Paso 1:** Definimos no `AndroidManifest.xml` o provider da forma:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="es.cursoandroid.cifprodolfoucha.aprendiendo">
4
5     <uses-permission android:name="android.permission.CALL_PHONE" />
6     <uses-permission android:name="android.permission.INTERNET" />
7     <uses-permission android:name="android.permission.WAKE_LOCK" />
8     <uses-permission android:name="android.permission.READ_CONTACTS" />
9     <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

```

10
11
12
13 <application
14     android:allowBackup="true"
15     android:icon="@mipmap/ic_launcher"
16     android:label="@string/app_name"
17     android:roundIcon="@mipmap/ic_launcher_round"
18     android:supportRtl="true"
19     android:theme="@style/AppTheme">
20
21     <provider
22         android:name="android.support.v4.content.FileProvider"
23         android:authorities="${applicationId}.provider"
24         android:exported="false"
25         android:grantUriPermissions="true">
26     </provider>

```

- Na liña 23, `applicationId` vaise substituír pola propiedade **applicationId** do arquivo **build.gradle** que coincide co nome do paquete.

- **Paso 2:** Debemos de definir un arquivo en `/res/xml/` que terá as rutas nas que deixaremos escribir.

Podemos consultar as diferentes rutas neste enlace (<https://developer.android.com/reference/android/support/v4/content/FileProvider#SpecifyFiles>).

No noso caso imos deixar escribir nos cartafolios `'/sdcard/Movies'` y `'/sdcard/Pictures'`:

Arquivo **/res/xml/provides_path.xml**

```

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

```

Fixarse que neste caso estamos a empregar a etiqueta `<external-path...>` que vai devolver a seguinte ruta: `Environment.getExternalStorageDirectory()` ([https://developer.android.com/reference/android/os/Environment.html#getExternalStorageDirectory\(\)](https://developer.android.com/reference/android/os/Environment.html#getExternalStorageDirectory())). Sobre dita ruta estamos engadindo os cartafolios `'Pictures'` e `'Movies/'` para crear dous `'external-path'` que serán os `path's` que se poderán compartir en forma de Uri entre aplicacións.

- **Paso 3:** Modificamos o posto no arquivo **AndroidManifest.xml** para que teña en conta as rutas postas no arquivo anterior:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="es.cursodolfocha.aprendiendo">
4
5     <uses-permission android:name="android.permission.CALL_PHONE" />
6     <uses-permission android:name="android.permission.INTERNET" />
7     <uses-permission android:name="android.permission.WAKE_LOCK" />
8     <uses-permission android:name="android.permission.READ_CONTACTS" />
9     <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
10
11
12
13 <application
14     android:allowBackup="true"
15     android:icon="@mipmap/ic_launcher"
16     android:label="@string/app_name"
17     android:roundIcon="@mipmap/ic_launcher_round"
18     android:supportRtl="true"
19     android:theme="@style/AppTheme">
20
21     <provider
22         android:name="android.support.v4.content.FileProvider"
23         android:authorities="${applicationId}.provider"
24         android:exported="false"
25         android:grantUriPermissions="true">
26         <meta-data
27             android:name="android.support.FILE_PROVIDER_PATHS"
28             android:resource="@xml/provides_path" />
29
30     </provider>

```

- Paso 4:** Modificamos a aplicación para que no caso de que esteamos sobre un S.O. Api >=24 faga uso do **método** **getUriForFile** o cal vai facer uso do provider definido antes. Desta forma a cámara vai poder acceder tamén a este mesmo Uri para gardar a foto/vídeo:

Activity UD6_05_MultimediaFotoVideo.java modificada.

```

1 package es.cursoandroid.cifprodolfoucha.aprendiendo.Multimedia;
2
3 import android.Manifest;
4 import android.app.Activity;
5 import android.content.Intent;
6 import android.content.pm.PackageManager;
7 import android.graphics.Bitmap;
8 import android.graphics.BitmapFactory;
9 import android.net.Uri;
10 import android.os.Build;
11 import android.os.Bundle;
12 import android.os.Environment;
13 import android.provider.MediaStore;
14 import android.view.View;
15 import android.widget.Button;
16 import android.widget.ImageView;
17 import android.widget.RadioButton;
18 import android.widget.RadioGroup;
19 import android.widget.Toast;
20 import android.widget.VideoView;
21
22 import java.io.File;
23
24 import es.cursoandroid.cifprodolfoucha.aprendiendo.R;
25
26 import static android.support.v4.content.FileProvider.getUriForFile;
27
28 public class UD6_05_MultimediaFotoVideo extends Activity {
29
30     //Código para verificar que o resultado ven do intent de gravación de imaxe
31     private final int REQUEST_CODE_GRAVACION_IMAXE = 1;
32     //Código para verificar que o resultado ven do intent de gravación de video
33     private final int REQUEST_CODE_GRAVACION_VIDEO = 2;
34
35     final private String nomeVideo="video.mp4";
36     final private String nomeFoto="foto.jpg";
37     private final int CODIGO_IDENTIFICADOR=1; // Usado por si necesitamos diferentes permisos, para identificar cual
de ellos es
38
39     /**
40      * Obtemos a imaxe ou video que ven da aplicacion Android
41      */
42     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
43         if (resultCode == RESULT_OK) {
44
45             File ruta,archivo;
46             switch(requestCode) {
47                 case REQUEST_CODE_GRAVACION_IMAXE:
48                     ruta = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES);
49                     archivo = new File(ruta, nomeFoto);
50                     if (!archivo.exists()) return; // Non hai foto
51
52                     ImageView imgview = (ImageView) findViewById(R.id.UD6_04_imgvwFoto);
53                     Bitmap bitmap = BitmapFactory.decodeFile(archivo.getAbsolutePath());
54                     imgview.setImageBitmap(bitmap);
55                     break;
56                 case REQUEST_CODE_GRAVACION_VIDEO:
57                     ruta = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_MOVIES);
58                     archivo = new File(ruta, nomeVideo);
59                     if (!archivo.exists()) return; // Non hai foto
60
61                     VideoView vidview = (VideoView) findViewById(R.id.UD6_04_vidvwVideo);
62                     vidview.setVideoURI(Uri.fromFile(archivo));
63                     vidview.start();
64                     break;
65             }
66         }
67         else if (resultCode == RESULT_CANCELED) {
68
69             // Video ou Foto cancelada
70         } else {
71
72             // Fallo na captura do Video ou foto.
73         }
74     }
75
76     public void pedirPermiso(){
77
78         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
79             requestPermissions( new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, CODIGO_IDENTIFICADOR);
80         }
81     }
82
83     @Override
84     public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {

```

```

86
87     switch (requestCode) {
88         case CODIGO_IDENTIFICADOR: {
89             // Se o usuario premeu o boton de cancelar o array volve cun null
90             Button btn = findViewById(R.id.UD6_04_btnGravarVideoFoto);
91             if (grantResults.length > 0
92                 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
93                 btn.setEnabled(true);
94                 // PERMISO CONCEDIDO
95             } else {
96                 // PERMISO DENEGADO
97                 btn.setEnabled(false); //Desabilitamos o botón se non temos o permiso de escritura
98                 Toast.makeText(this, "É NECESARIO O PERMISO DE ESCRITURA NA SD CARD", Toast.LENGTH_LONG).show();
99             }
100             return;
101         }
102     }
103     // Comprobamos os outros permisos
104 }
105 }
106 }
107
108 /**
109  * Programa o código dos click's os botóns
110  */
111 private void xestionarEventos(){
112     Button gravar = (Button)findViewById(R.id.UD6_04_btnGravarVideoFoto);
113     gravar.setOnClickListener(new View.OnClickListener(){
114
115         @Override
116         public void onClick(View arg0) {
117             // TODO Auto-generated method stub
118             RadioButton rb = (RadioButton)findViewById(R.id.UD6_04_rbbtnFoto);
119
120
121             if (rb.isChecked()){ // Sacar foto
122                 File ruta = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES);
123                 File arquivo = new File(ruta, nomeFoto);
124
125                 Uri contentUri=null;
126                 if (Build.VERSION.SDK_INT >= 24) {
127                     contentUri = getUriForFile(getApplicationContext(), getApplicationContext()
128                         .getPackageName() + ".provider", arquivo);
129                 }
130                 else {
131                     contentUri = Uri.fromFile(arquivo);
132                 }
133
134                 Intent intento = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
135                 intento.putExtra(MediaStore.EXTRA_OUTPUT, contentUri);
136
137                 startActivityForResult(intento, REQUEST_CODE_GRAVACION_IMAXE);
138             }
139             else { // Grava vídeo
140                 File ruta = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_MOVIES);
141                 File arquivo = new File(ruta, nomeVideo);
142
143                 Uri contentUri=null;
144                 if (Build.VERSION.SDK_INT >= 24) {
145                     contentUri = getUriForFile(getApplicationContext(), getApplicationContext()
146                         .getPackageName() + ".provider", arquivo);
147                 }
148                 else {
149                     contentUri = Uri.fromFile(arquivo);
150                 }
151
152                 Intent intento = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
153                 intento.putExtra(MediaStore.EXTRA_OUTPUT, contentUri);
154
155                 startActivityForResult(intento, REQUEST_CODE_GRAVACION_VIDEO);
156             }
157         }
158     }
159 }
160 });
161
162 RadioGroup rgroup = (RadioGroup)findViewById(R.id.UD6_04_rgrpOptions);
163 rgroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
164
165     @Override
166     public void onCheckedChanged(RadioGroup group, int checkedId) {
167         // TODO Auto-generated method stub
168         ImageView imgview;
169         VideoView videoview;
170
171         switch(checkedId){
172             case R.id.UD6_04_rbbtnFoto: // OCULTAMOS O VIDEO
173                 imgview = (ImageView)findViewById(R.id.UD6_04_imgvwFoto);
174                 imgview.setVisibility(View.VISIBLE);
175                 videoview = (VideoView)findViewById(R.id.UD6_04_vidvwVideo);
176                 videoview.setVisibility(View.GONE);
177                 break;
178             case R.id.UD6_04_rbbtnVideo: // OCULTAMOS AS FOTOS
179                 imgview = (ImageView)findViewById(R.id.UD6_04_imgvwFoto);

```

```

180         imgview.setVisibility(View.GONE);
181         videoview = (VideoView)findViewById(R.id.UD6_04_vidvwVideo);
182         videoview.setVisibility(View.VISIBLE);
183         break;
184     }
185 }
186 });
187 }
188
189 @Override
190 protected void onCreate(Bundle savedInstanceState) {
191     super.onCreate(savedInstanceState);
192     setContentView(R.layout.activity_ud05_01__foto__reproducir);
193
194     pedirPermiso();
195     xestionarEventos();
196 }
197 }

```

- Liñas 76-82: Se a API >=23 hai que pedir permiso de escritura por código.
- Liñas 93-97: En caso de que o usuario non dea permiso, desactivamos o botón.
- Liñas 125-133: En caso de que a API >=24 necesitamos ao 'método getUriForFile' para obter a URI que lle imos pasar como dato extra a cámara de fotos. En caso contrario o facemos como ata o de agora.

Fixarse como o segundo parámetro pon: `getApplicationContext().getPackageName() + ".provider"`, que se corresponde coa entrada **android:authorities="\${applicationId}.provider"** do AndroidManifest.xml.

- Liñas 143-150: O mesmo para o caso do vídeo.

- A forma anterior é unha das posibilidades que temos de dar permiso a certa Uri para que sexa compartida. Neste caso o facemos enviamos con `setData` a Uri.

Existen outras formas como podemos consultar neste enlace (<https://developer.android.com/reference/android/support/v4/content/FileProvider#Permissions>).

- Máis información:
 - <https://developer.android.com/reference/android/support/v4/content/FileProvider#Permissions> (<https://developer.android.com/reference/android/support/v4/content/FileProvider#Permissions>).
 - Exemplo de uso (<https://medium.com/@ali.muzaffar/what-is-android-os-fileuriexposedexception-and-what-you-can-do-about-it-70b9eb17c6d0>).

Métodos útiles no manexo de Imaxes

- **Bitmap.createScaledBitmap(bitmaporixinal, 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`:

```

1     switch (getResources().getDisplayMetrics().densityDpi) {
2         case DisplayMetrics.DENSITY_LOW:
3             break;
4         case DisplayMetrics.DENSITY_MEDIUM:
5             break;
6         case DisplayMetrics.DENSITY_HIGH:
7             break;
8         case DisplayMetrics.DENSITY_XHIGH:
9             break;
10    }

```

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:

```
1 Bitmap bitmap = ((BitmapDrawable)imvFoto.getDrawable()).getBitmap();
```

Sendo imgview 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):

```
1 Bitmap bitmap = BitmapFactory.decodeFile(Ruta_apuntando_á_imaxe_gardada);
2 img.setImageBitmap(bitmap);
```

Sendo img un obxecto da clase ImageView.

Métodos útil no manexo de Vídeos

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

Para facelo debemos de utilizar a clase [ThumbnailUtils](http://developer.android.com/reference/android/media/ThumbnailUtils.html) (<http://developer.android.com/reference/android/media/ThumbnailUtils.html>)

Vexamos un exemplo:

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

- Para cargar un vídeo localmente gardado en /res/raw/ poñeríamos:

```
1 mVideoView.setVideoURI(Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.small_video));
```

- Para cargar un vídeo dende Internet poñeríamos:

```
1 mVideoView.setVideoPath("http://direccion_internet/cartafof/video.mp4");
```

Nota: Lembrar ter o permiso de acceso a internet no AndroidManifest.xml

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](http://developer.android.com/reference/android/widget/Gallery.html) (<http://developer.android.com/reference/android/widget/Gallery.html>): deprecated a partires da API 16.
- [Control HorizontalScrollView](http://developer.android.com/reference/android/widget/HorizontalScrollView.html) (<http://developer.android.com/reference/android/widget/HorizontalScrollView.html>).
- [Control ViewPager](http://developer.android.com/reference/android/support/v4/view/ViewPager.html) (<http://developer.android.com/reference/android/support/v4/view/ViewPager.html>).

Actualización 2018

A partires da API 22 aparece o [Widget RecyclerView](https://developer.android.com/reference/android/support/v7/widget/RecyclerView) (<https://developer.android.com/reference/android/support/v7/widget/RecyclerView>) que permite facer o mesmo que o HorizontalScrollView pero de forma moito máis eficiente.

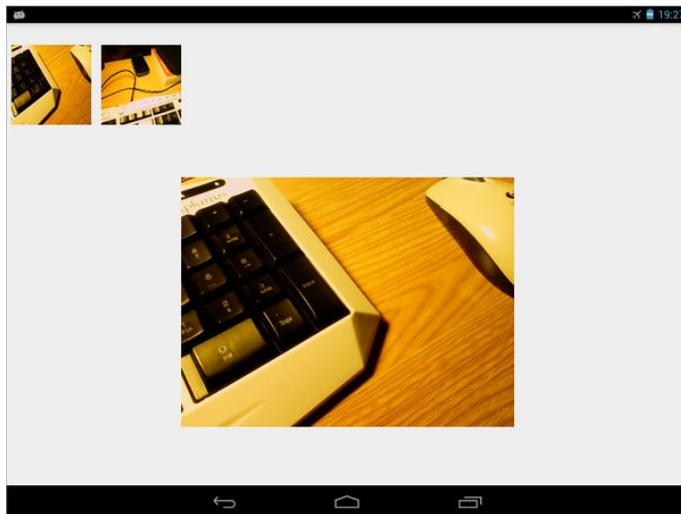
Para cargar dito compoñente nunha API inferior á 22 (vos tedes que facer a aplicación cun API>=19) teredes que facer uso dunha biblioteca de compatibilidade, como xa vimos na Wiki (http://wiki.cifprodolfoucha.es/index.php?title=PDM_UD1_Bibliotecas_de_compatibilidade).

Xa está explicado o seu funcionamento nesta WIKI (https://wiki.cifprodolfoucha.es/index.php?title=PDM_RecyclerView_e_CardView).

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

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.

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 podedes 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.

Creemos a Activity

- Nome do proxecto: **UD6_06_MultimediaHorizontalScrollView**
- Nome da activity: **UD6_06_MultimediaHorizontalScrollView.java**

Código do layout xml

```
1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   tools:context="${relativePackage}.${activityClass}" >
6
```



```

7      <HorizontalScrollView
8          android:id="@+id/UD6_06_hsvviewFotosPai"
9          android:layout_width="match_parent"
10         android:layout_height="wrap_content"
11         android:layout_alignParentTop="true"
12         android:layout_centerHorizontal="true"
13         android:layout_marginTop="26dp" >
14
15         <LinearLayout
16             android:id="@+id/UD6_06_linearLayoutFotosFillo"
17             android:layout_width="wrap_content"
18             android:layout_height="wrap_content"
19             android:orientation="horizontal" />
20
21     </HorizontalScrollView>
22
23     <ImageView
24         android:id="@+id/UD6_06_imgvwFotoAmpliada"
25         android:layout_width="500dp"
26         android:layout_height="500dp"
27         android:layout_alignParentBottom="true"
28         android:layout_centerHorizontal="true"
29         android:layout_marginBottom="26dp"
30         android:src="@drawable/ic_launcher" />
31
32 </RelativeLayout>

```

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

Código da clase UD6_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.

```

1 import java.io.File;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.os.Environment;
6 import android.widget.LinearLayout;
7
8 public class UD6_06_MultimediaHorizontalScrollView extends Activity {
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_ud6_06_multimedia_horizontal_scroll_view);
14
15         LinearLayout linear = (LinearLayout)findViewById(R.id.UD6_06_linearLayoutFotosFillo);
16
17         String ruta = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES).getAbsolutePath();
18
19         UD6_06_ImaxeHorizontalScrollView imaxe = new UD6_06_ImaxeHorizontalScrollView(this, ruta+File.separator+"foto1.jpg");
20         linear.addView(imaxe.getLayout());
21         imaxe = new UD6_06_ImaxeHorizontalScrollView(this, ruta+File.separator+"foto2.jpg");
22         linear.addView(imaxe.getLayout());
23     }
24 }
25 }

```

- 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 UD6_06_ImaxeHorizontalScrollView (explicada posteriormente).
- Liñas 20,22: Engadimos ditas imaxes ó LinearLayout.

Creamos o ImageView personalizado

Agora imos explicar a clase **UD6_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 UD6_o6_ImaxeHorizontalScrollView

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

```

1 import java.io.File;
2
3 import android.graphics.Bitmap;
4 import android.graphics.BitmapFactory;
5 import android.view.Gravity;
6 import android.view.View;
7 import android.view.View.OnClickListener;
8 import android.view.ViewGroup.LayoutParams;
9 import android.widget.ImageView;
10 import android.widget.LinearLayout;
11
12 public class UD6_o6_ImaxeHorizontalScrollView extends ImageView implements OnClickListener {
13     protected UD6_o6_MultimediaHorizontalScrollView context;
14     private File file;
15     private String path;
16     private LinearLayout layout;
17
18     /**
19      * Crea unha imaxe e a engade nun layout que vai ser o que visualice o HorizontalView
20      * @param context
21      * @param path: ruta a imaxe
22      */
23     public UD6_o6_ImaxeHorizontalScrollView(UD6_o6_MultimediaHorizontalScrollView context,String path) {
24         super(context);
25         this.context=context;
26         this.path=path;
27         file = new File(path);
28
29         setOnClickListener(this);
30         crearLayout();
31     }
32
33     private void crearLayout() {
34         LinearLayout layout = new LinearLayout(context);
35         layout.setLayoutParams(new LayoutParams(135, 135));
36         layout.setGravity(Gravity.CENTER);
37
38         this.setLayoutParams(new LayoutParams(120, 120)); // TAMAÑO DA IMAXE
39         this.setScaleType(ImageView.ScaleType.CENTER_CROP);
40         Bitmap bit = BitmapFactory.decodeFile(file.getAbsolutePath());
41         if (bit==null) { // Non a pode cargar
42             this.layout=layout;
43             return;
44         }
45         Bitmap bitmap;
46         bitmap = Bitmap.createScaledBitmap(bit, 220, 220,false);
47         this.setImageBitmap(bitmap);
48
49         layout.addView(this);
50         this.layout = layout;
51     }
52
53     public LinearLayout getLayout(){
54         return layout;
55     }
56
57     public File getFile(){
58         return file;
59     }
60     public String getPath(){
61         return path;
62     }
63     @Override
64     public void onClick(View v) {
65         // TODO Auto-generated method stub
66
67         Bitmap bp = BitmapFactory.decodeFile(getPath());
68         ImageView imgview = (ImageView)context.findViewById(R.id.UD6_o6_imgvwFotoAmpliada);
69         imgview.setImageBitmap(bp);
70     }
71
72
73 }

```

- 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 (UD6_06_imgvwFotoAmpliada) e cargamos a partires do path a imaxe no ImageView.

Enlace a la página principal de la UD7 (https://wiki.cifprodolfoucha.es/index.php?title=Programaci%C3%B3n_de_dispositivos_m%C3%B3viles#UNIDADE_7:_Multimedia)

Enlace a la página principal del curso (https://wiki.cifprodolfoucha.es/index.php?title=Programación_de_dispositivos_móviles)

-- [Ángel D. Fernández González](#) e [Carlos Carrión Álvarez](#) -- (2014).

Obtenido de «https://wiki.cifprodolfoucha.es/index.php?title=PDM_Avanzado_Captura_de_Vídeo_/_Imaxes&oldid=12091»

Esta página se editó por última vez el 12 ene 2021 a las 20:30.