

Android: Camara

android.hardware.camera

Uso de la cámara

- ▶ Si la aplicación se basa en el uso de la cámara se debe indicar en el manifiesto.

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

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

Permiso de la cámara

- ▶ Si la aplicación hace uso directo de la cámara sin pasar por un intent implícito.

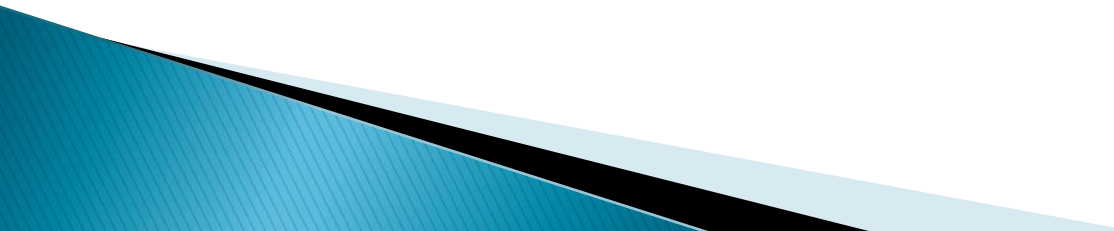
```
<uses-permission android:name  
    ="android.permission.CAMERA" />
```

Comprobar presencia de la cámara

- ▶ Se puede comprobar la presencia de la cámara:

```
if (getPackageManager().hasSystemFeature(  
    PackageManager.FEATURE_CAMERA)) {  
    return true;  
} else {  
    return false;  
}
```

Tipos de cámaras

- ▶ Cámara frontal:
PackageManager.*FEATURE_CAMERA_FRONT*
 - ▶ Cámara trasera:
PackageManager.*FEATURE_CAMERA*
 - ▶ Cualquier cámara:
PackageManager.*FEATURE_CAMERA_ANY*
- 

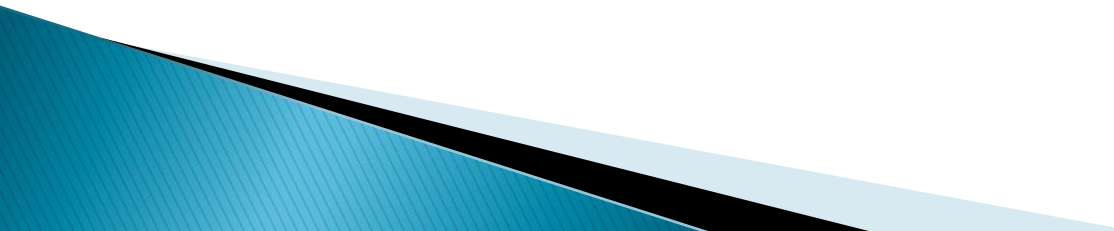
Intents de la cámara

- ▶ `MediaStore.ACTION_IMAGE_CAPTURE`, para la toma de fotos.
- ▶ `MediaStore.ACTION_VIDEO_CAPTURE`, para la toma de vídeos.

Comprobar disponibilidad de intent

- ▶ Antes de realizar la llamada del intent implícito se debe comprobar si está disponible.

```
public static boolean intentDisponible(Context
                                     contexto, String accion) {
    final PackageManager adminPaquetes = contexto.
                                     getPackageManager();
    final Intent intent = new Intent(accion);
    List<ResolveInfo> lista = adminPaquetes.
                                     queryIntentActivities(intent,
                                     PackageManager.MATCH_DEFAULT_ONLY);
    return lista.size() > 0;
}
```



Tomar una fotografía

```
Intent intent = new Intent  
    (MediaStore.ACTION_IMAGE_CAPTURE);  
startActivityForResult(intent, CTE);
```


Guardar la fotografía


Por el propio comportamiento, la guarda con un 1/8 de calidad

@Override

```
public void onActivityResult(int pet, int res, Intent data) {  
    if (res == RESULT_OK && pet == CTE) {  
        Bitmap foto = (Bitmap)data.getExtras().get("data");  
        FileOutputStream salida;  
        try {  
            salida = new FileOutputStream(getFilesDir()+  
                "/foto.jpg");  
            foto.compress(Bitmap.CompressFormat.JPEG, 90,  
                salida);  
        } catch (FileNotFoundException e) {  
        }  
    }  
}
```


Tomar fotografía y guardar

```
Intent intent = new Intent(  
    MediaStore.ACTION_IMAGE_CAPTURE);  
File archivo = new File(  
    Environment.  
        getExternalStoragePublicDirectory(  
            Environment.DIRECTORY_DCIM)  
            + "/foto.jpg");  
Uri fileUri = Uri.fromFile(archivo);  
intent.putExtra(MediaStore.EXTRA_OUTPUT,  
    fileUri);  
startActivityForResult(intent, ACCION_FOTO);
```



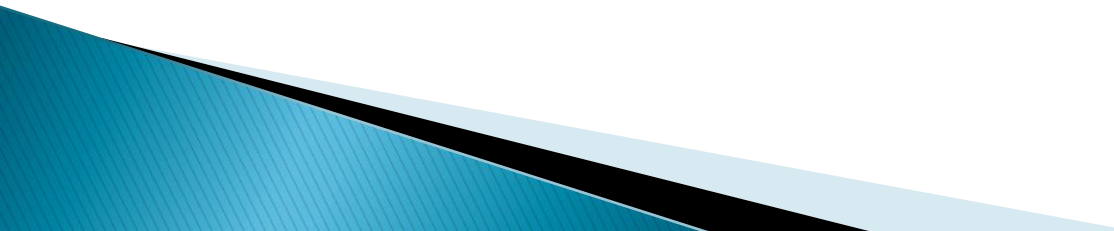
Tomar un vídeo

```
File archivo = new File(Environment.  
    getExternalStoragePublicDirectory(  
        Environment.DIRECTORY_MOVIES)  
        + "/video.mp4");  
Intent intent = new Intent(  
    MediaStore.ACTION_VIDEO_CAPTURE);  
Uri videoUri = Uri.fromFile(archivo);  
intent.putExtra(MediaStore.EXTRA_OUTPUT,  
    videoUri);  
startActivityForResult(intent, ACCION_VIDEO);
```



Parámetros para la captura de vídeo

```
intent.putExtra(  
    MediaStore.EXTRA_VIDEO_QUALITY, 0);  
    //calidad de 0 a 1  
intent.putExtra(MediaStore.EXTRA_SIZE_LIMIT,  
    1024*1024);  
    //número de bytes  
intent.putExtra(  
    MediaStore.EXTRA_DURATION_LIMIT, 4);  
    //segundos
```



Reproducir el vídeo

```
@Override
public void onActivityResult(int requestCode,
                             int resultCode, Intent data) {
    if (resultCode == Activity.RESULT_OK &&
        requestCode == ACCION_VIDEO) {
        VideoView vv = (VideoView)
                        findViewById(R.id.vv1);
        vv.setVideoURI(data.getData());
        vv.start();
    }
}
```

