

Captura de Audio y Video [Java]

Hemos visto que Android nos ofrece una mecánica para acceder a otros **componentes** de nuestro dispositivo mediante el uso de los **Intents Implícitos**. Vamos a ver cómo podemos acceder así a la cámara del dispositivo. Date cuenta de que gracias a cómo se gestionan los componentes no vamos a tener que conocer ni programar nuestra app específicamente para un modelo concreto de cámara; nos bastará con solicitar al sistema operativo que busque los componentes capaces de responder a mis necesidades. Los haga funcionar, y nos retorne la foto en cuanto finalice. Esto nos **abstrae** también del funcionamiento exacto de cada componente llamado (y puede que ni sepamos cuál es).

Permisos

Para poder utilizar la cámara o cualquier componente debemos comenzar dándole permisos en el **Manifest**. Diferentes componentes necesitarán diferentes permisos, obviamente. Dado que para nuestra app la cámara es imprescindible, indicamos que el atributo required es true. Esto entre otras cosas hace que Google Play no permita que dispositivos sin cámara puedan descargar nuestra app.

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

Es nuestra responsabilidad comprobar la disponibilidad de la cámara en tiempo de ejecución. Si no disponemos de ella, habrá que deshabilitar las partes de nuestro código que dependan de la cámara. Podemos hacerlo con:

```
PackageManager packageManager = getPackageManager ();
boolean cameraAvailable = packageManager.hasSystemFeature(Pack-
ageManager.FEATURE_CAMERA_ANY);
```

Captura de la imagen

Para capturar una imagen, hay que crear un **Intent** e indicarle **la acción** que se quiere realizar. Obviamente, tendremos que conocer el nombre de la acción correspondiente para cada uno de los diferentes componentes del Android; o incluso podremos crear nuestros propios componentes y atender a acciones completamente nuevas. En este caso, la acción para todas las cámaras de los dispositivos móviles es:

MediaStore.ACTION_IMAGE_CAPTURE.

Por tanto, podemos declarar un Intent Implícito con dicha acción. **Nótese** que no tenemos mucho más que hacer hasta que el componente cámara de fotos termina su ejecución, por lo tanto, nuestra siguiente línea de código será simplemente tratar la respuesta de la cámara. Como siempre, hay que conocer de antemano exactamente qué y cómo envían las respuestas los diferentes componentes a los que llamamos. En este caso:

```
static final int CAPTURA_IMAGEN = 1;

Intent hacerFotoIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
if (hacerFotoIntent.resolveActivity(getPackageManager()) != null) {
    startActivityForResult(hacerFotoIntent, CAPTURA_IMAGEN);
}
```

El método **resolveActivity ()** nos devuelve la actividad que debe manejar el Intent que hemos lanzado. Si es nulo, entonces es que el dispositivo no tenga una componente capaz de atender a la acción mencionada (manejar la cámara).

En este caso y por defecto, la foto nos es devuelta de la cámara como un *thumbnail*. El thumbnail va integrado dentro del propio Intent como un dato extra, el cual podemos procesar a través del método **onActivityResult ()** de la forma habitual.

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == CAPTURA_IMAGEN && resultCode == RESULT_OK) {
        Bundle extras = data.getExtras();
        Bitmap imageBitmap = (Bitmap) extras.get("data");
        imageView.setImageBitmap(imageBitmap);
    }
}
```

Práctica 44

[Java] Realiza una app que sea capaz de tomar fotos y mostrarlas luego. El Android Studio **permite** que puedas hacer todo esto mediante el emulador, dado que ‘simula’ una habitación virtual para la cámara. Investiga un poco su funcionamiento.

Práctica 45

[Kotlin] Lo mismo que antes, pero en Kotlin.

Captura de vídeo

Android ofrece dos posibilidades para grabar video:

- Usar **Intent**, delegando a alguna aplicación nativa de Android su grabación.
- Usar la clase **MediaRecorder** si no queremos usar una aplicación nativa de Android.

La primera opción suele ser la más adecuada en la mayoría de los casos. Al igual que antes, debemos de configurar el **Manifest** indicando:

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

Siguiendo con la lógica anterior, la **acción** para **capturar vídeo** es:

MediaStore.ACTION_VIDEO_CAPTURE.

El Intent se realiza de forma similar con los siguientes atributos:

- MediaStore.EXTRA_VIDEO_QUALITY: Para la calidad del vídeo, siendo 0 la más baja.
- MediaStore.EXTRA_DURATION_LIMIT: Para limitar el tiempo que estaremos grabando, en segundos.

Práctica 46

Realiza la práctica propuesta en el documento **10 - Captura de Fotos y Video [Java]**

Práctica 47

Realiza la práctica propuesta en el documento **10 - Captura de Fotos y Video [Kotlin]**