



Desarrollo de aplicaciones móviles en Android



Índice

- 🖱 Historia
- 🖱 ¿Qué es Android?
- 🖱 Arquitectura del sistema
- 🖱 Entorno de desarrollo
- 🖱 Arquitectura de las aplicaciones
- 🖱 Almacenamiento de datos
- 🖱 *Multimedia*
- 🖱 Práctica final



Objetivos

Multimedia

- ▶ Conocer principales componentes multimedia de la plataforma Android.
- ▶ Identificar las posibilidades que ofrece el uso de la cámara como herramienta multimedia en nuestra aplicaciones.
- ▶ Aprender a utilizar el video para las app android.
- ▶ Conocer las posibilidades del canvas Android para la realización de interfaces ricos.



MediaPlayer

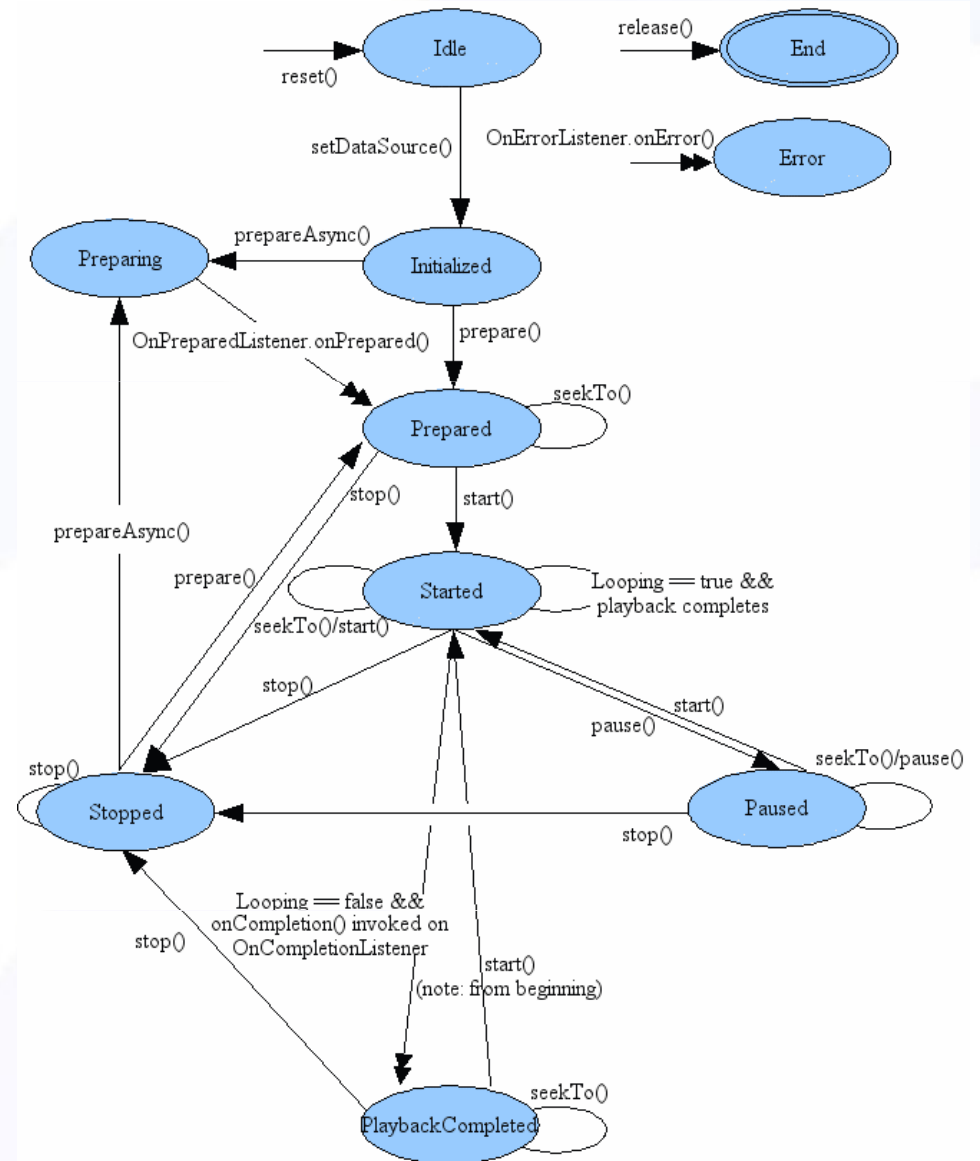
- La clase MediaPlayer se utiliza para reproducir Audio y Video de una forma simple y directa
- Funciona como máquina de estados
- Permite reproducción, parada del multimedia, pausa, etc
- Soporta la mayoría de los formatos de audio y vídeo más conocidos
- Los formatos que soporta Android para audio son: WAV, AAC, MP3, WMA, AMR, OGG y MIDI.



```
MediaPlayer mp = new MediaPlayer();  
mp.setDataSource(PATH_TO_FILE);  
mp.prepare();  
mp.start();
```

MediaPlayer: estados y transiciones

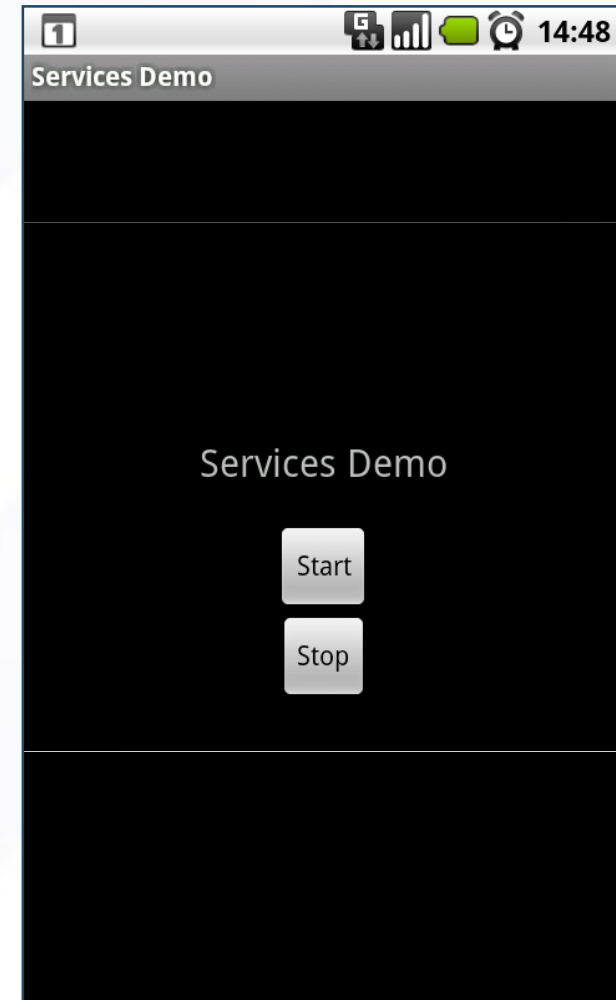
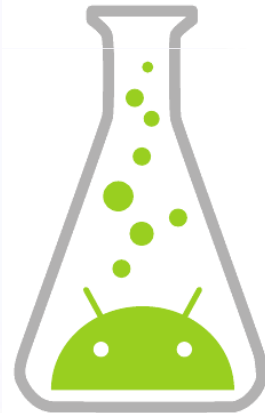
- El ciclo de vida de una reproducción sin errores pasa por los estados:
- IDLE (MediaPlayer creado),
- INITIALIZED (MediaPlayer informado de cuál es el media a reproducir)
- PREPARED (MediaPlayer tiene toda la información necesaria para reproducir)
- STARTED (MediaPlayer reproduciendo)
- STOPPED (MediaPlayer finalizado)



Ejemplos y ejercicios

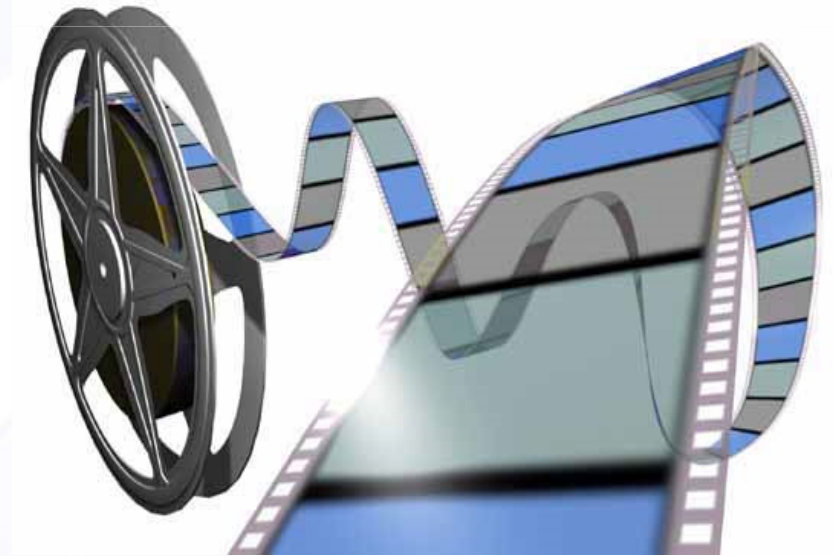
➤ **Service-MusicDemo: cómo inicializar una instancia de MediaPlayer para reproducción de un audio local:**

En este ejemplo se muestra cómo se lanza un Service que prepara y lanza el reproductor MediaPlayer. En este caso al ejecutar `MediaPlayer.create(...)` ya se le informa del audio a reproducir y queda en un estado de PREPARED.



Camera

- Android soporta el uso de cámaras integradas en los dispositivos
- Se pueden capturar imágenes o previsualizar lo que recibe la cámara
- Se utiliza las clases Camera y SurfaceHolder y SurfaceView para recibir y visualizar la imagen de la cámara
- Se hereda de la clase SurfaceView y se implementa SurfaceHolder.Callback.
- El emulador está limitado y no puede recibir imágenes reales.

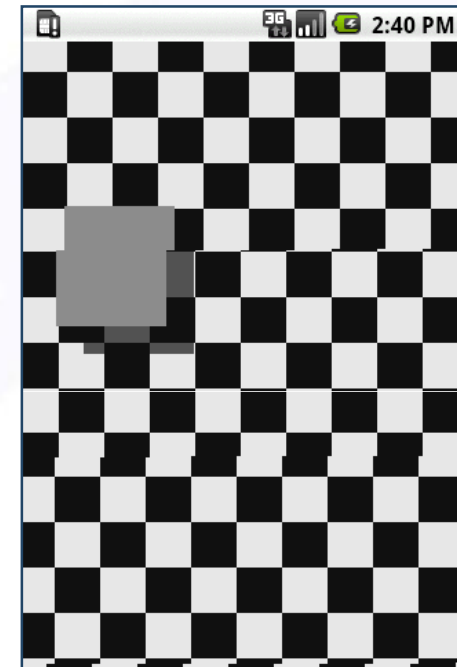
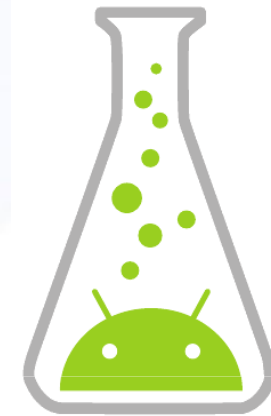


Ejemplos y ejercicios

➤ Camera-Demo: cómo utilizar la cámara del dispositivo móvil:

En este ejemplo se puede ver cómo la Activity principal crea la clase Preview que hereda de SurfaceView implementa SurfaceHolder.Callback. Extendiendo SurfaceView se puede utilizar como contenedor de imágenes que recibirá de la cámara, y por lo tanto utilizarse como View principal para la aplicación (método setContentView()).

Por otro lado, al implementar SurfaceHolder.Callback debe sobrescribir los métodos surfaceCreated(), surfaceDestroyed() y surfaceChanges(), donde se realizarán las conexiones entre la Surface y la Camera.



Reproducción de video

- Android permite la reproducción de vídeo dentro de la aplicación.
- La vista VideoView permite insertar contenido de tipo Video
- Formatos MP4, H.263 o 3GP, y H.264 o AVC entre otros
- La clase MediaController ofrece una vista con botones de Play, Pause, Rewind, FastForward y ProgressSlider



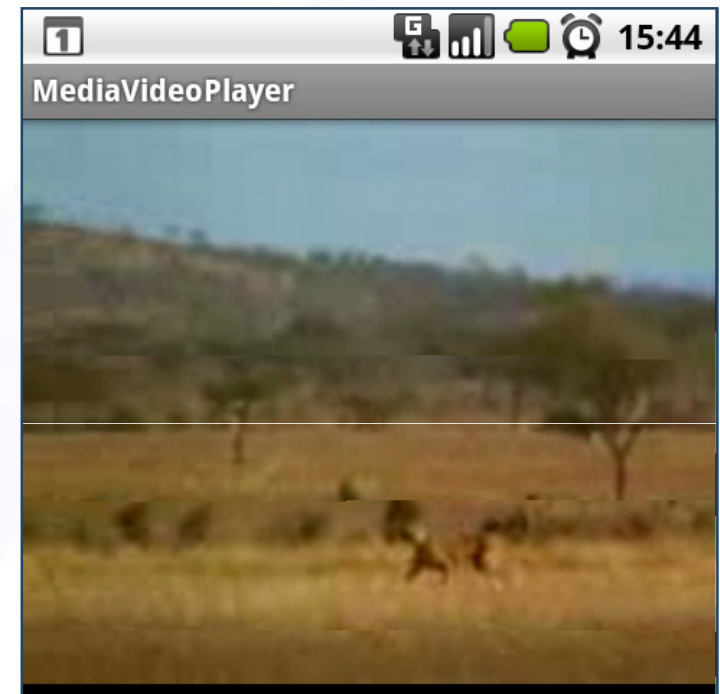
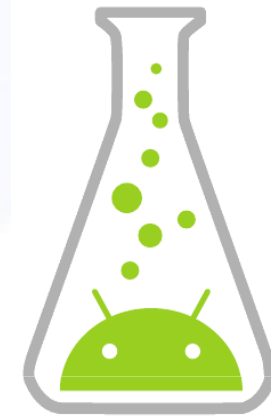
Ejemplos y ejercicios

➤ **Media-VideoPlayer: cómo reproducir Video en nuestra aplicación:**

En este ejemplo se puede ver cómo la Activity principal crea una vista de tipo `VideoView`, que asigna como contenido principal a nuestra aplicación.

Con esto, se setean los atributos del objeto `VideoView`, y el método `start()` comienza la reproducción del video.

El objeto `MediaController` añade botones de pausa y stop y la barra de progreso a la reproducción del vídeo.



Dibujado

- Para realizar aplicaciones más visuales, Android permite dibujar sobre objetos visuales.
- La librería `android.graphics` ofrece primitivas de dibujo de muchos tipos: puntos, rectángulos, líneas, y efectos de imagen.
- El dibujo se realiza sobre un canvas, sobrescribiendo el método `onDraw()` de una View
- Se pueden modificar Views existentes sobrescribiendo ese método y redibujando su Canvas.

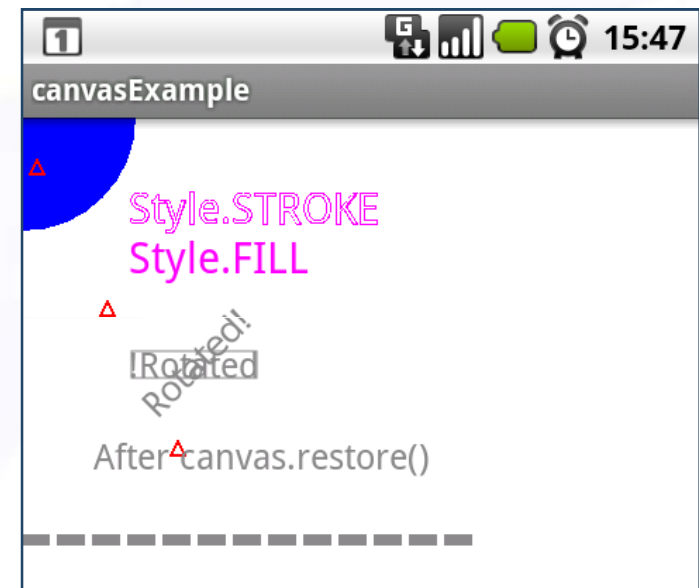
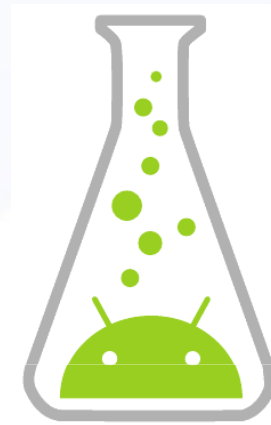


Ejemplos y ejercicios

➤ Media-Canvas: cómo dibujar en nuestra aplicación

En este ejemplo se puede ver cómo utilizar distintas clases del paquete `android.graphics` para dibujar textos y primitivas.

Todo se realiza en una clase `CanvasView`, que ofrece un método `onDraw(Canvas)` sobre el que realizaremos el dibujo.



Conclusiones

Multimedia

- ▶ Conocer principales componentes multimedia de la plataforma Android.
- ▶ Identificar las posibilidades que ofrece el uso de la cámara como herramienta multimedia en nuestra aplicaciones.
- ▶ Aprender a utilizar el video para las app android.
- ▶ Conocer las posibilidades del canvas Android para la realización de interfaces ricos.

