

# Descripción general de MediaRecorder

El marco de trabajo de contenido multimedia de Android admite la captura y la **codificación de una variedad de formatos comunes de audio y video**. Puedes usar las API de [MediaRecorder](#) si son compatibles con el hardware del dispositivo.

En este documento, se muestra cómo usar MediaRecorder para escribir una app que capture audio del micrófono de un dispositivo, lo guarde y lo reproduzca (con [MediaPlayer](#)). Para grabar video, tendrás que usar la cámara del dispositivo junto con [Camera2](#). Esto se describe en la guía de la [cámara](#).

**Nota:** Android Emulator no puede grabar audio. Asegúrate de probar el código en un dispositivo real que pueda grabar.

## Cómo solicitar permiso para grabar audio

Para poder grabar, la app le debe avisar al usuario que tendrá acceso a la entrada de audio del dispositivo. Tienes que incluir esta etiqueta de permiso en el archivo de manifiesto de la app:

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

RECORD\_AUDIO se considera un **permiso "peligroso"** porque puede representar un riesgo para la privacidad del usuario. A partir de Android 6.0 (API nivel 23), toda app que use un permiso peligroso tiene que solicitar la aprobación del usuario en el **momento de ejecución**. Una vez que el usuario haya otorgado el permiso, la aplicación debe recordarlo y no volver a preguntar.

## Cómo crear y ejecutar un MediaRecorder

Inicializa una nueva instancia de [MediaRecorder](#) con las siguientes llamadas:

- Configura la fuente de audio mediante [setAudioSource\(\)](#). Probablemente uses [MIC](#).

**Nota:** La mayoría de las fuentes de audio (incluido **DEFAULT**) procesan la señal de audio. Para grabar audio sin procesar, selecciona **UNPROCESSED**. Algunos dispositivos no admiten entradas no procesadas. Primero, llama al [AudioManager.getProperty\(AudioManager.PROPERTY\\_SUPPORT\\_AUDIO\\_SOURCE\\_UNPROCESSED\)](#) para verificar que esté disponible. Si no es así, intenta usar **VOICE\_RECOGNITION** en su lugar, que no emplea AGC ni supresión de ruido. Puedes usar **UNPROCESSED** como fuente de audio incluso cuando la propiedad no sea compatible, pero no hay garantía de que la señal se vaya a procesar (o no) en ese caso.

- Configura el formato del archivo de salida mediante [setOutputFormat\(\)](#). Ten en cuenta que a partir de Android 8.0 (API nivel 26) MediaRecorder es compatible con el formato MPEG2\_TS, que sirve para transmitir:
- `mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.MPEG_2_TS)`
- Configura el nombre del archivo de salida mediante [setOutputFile\(\)](#). Tienes que especificar un descriptor del archivo que represente un archivo real.
- Configura el codificador de audio con [setAudioEncoder\(\)](#).
- Llama a [prepare\(\)](#) para completar la inicialización.

Llama a [start\(\)](#) y [stop\(\)](#) para iniciar la grabadora y para detenerla, respectivamente.

Cuando hayas terminado con la instancia MediaRecorder, llama a [release\(\)](#) para liberar sus recursos lo antes posible.

**Nota:** En los dispositivos que ejecutan Android 9 (API nivel 28) o superior, las apps que se ejecutan en segundo plano no pueden acceder al micrófono. Por lo tanto, tu app tiene que grabar audio solo cuando está en primer plano o cuando incluyes una instancia de **MediaRecorder** en un [servicio en primer plano](#).

## Cómo usar MediaMuxer para grabar varios canales

A partir de Android 8.0 (API nivel 26), puedes usar un [MediaMuxer](#) para grabar varias transmisiones de video y audio simultáneas. Con versiones anteriores de Android, solo puedes grabar una pista de audio y/o una pista de video por vez.

Usa el método [addTrack\(\)](#) para mezclar varias pistas.

También puedes agregar una pista de metadatos o más con información personalizada para cada marco, pero solo a contenedores MP4. La app define el formato y el contenido de los metadatos.

## Metadatos

Los metadatos pueden ser útiles para el procesamiento sin conexión. Por ejemplo, los datos que captura el sensor giroscópico se podrían usar para llevar a cabo la estabilización de video.

