
Universidad
Tecmilenio®

Universidad Tecmilenio

Desarrollo de aplicaciones en plataforma Android

Emisiones, hilos y servicios

Introducción

Las apps de Android pueden enviar o recibir mensajes de emisión desde el sistema de Android y otras apps para Android, dichas emisiones son enviadas cuando ocurre un evento de interés.

Un ejemplo son los avisos de batería baja, un mensaje de que se ha conectado la batería, etc.

Explicación

Las apps pueden recibir emisiones de dos maneras: mediante receptores declarados en el manifiesto y mediante receptores registrados en el contexto.

Receptores declarados en el manifiesto

- Al declarar un receptor de emisión en tu manifiesto, el sistema inicia la app (si aún no está en ejecución) cuando se envía la emisión.

  Requiere los siguientes pasos:
  1. Especificar el elemento <receiver> en el manifiesto de la app.

```
<receiver android:name=".MyBroadcastReceiver"  android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED"/>
        <action android:name="android.intent.action.INPUT_METHOD_CHANGED" />
    </intent-filter>
</receiver>
```

Explicación

2. Crear la subclase **BroadcastReceiver** y, luego, implementa **onReceive**(Context, Intent).

Ejemplo:

```
public class MyBroadcastReceiver extends BroadcastReceiver {
        private static final String TAG = "MyBroadcastReceiver";
        @Override
        public void onReceive(Context context, Intent intent) {
            StringBuilder sb = new StringBuilder();
            sb.append("Action: " + intent.getAction() + "\n");
            sb.append("URI: " + intent.toUri(Intent.URI_INTENT_SCHEME).toStrin
            String log = sb.toString();
            Log.d(TAG, log);
            Toast.makeText(context, log, Toast.LENGTH_LONG).show();
        }
    }
```

Explicación

Receptores que se registran en el contexto:

1. A través de crear la instancia **BroadcastReceiver**

```
BroadcastReceiver br = new MyBroadcastReceiver();
```

2. Crear un **IntentFilter** y registrar el receptor mediante una llamada a **registerReceiver(BroadcastReceiver, IntentFilter):**

```
IntentFilter filter = new IntentFilter(ConnectivityManager.CONNECTIVITY_ACTION)
    filter.addAction(Intent.ACTION_AIRPLANE_MODE_CHANGED);
    this.registerReceiver(br, filter);
```

Explicación

Los receptores registrados en el contexto reciben emisiones siempre que su contexto de registro sea válido. Por ejemplo, si se registra en un contexto Activity, se reciben emisiones siempre y cuando esté vigente la actividad.

En cambio si se registra en el contexto de la aplicación, se reciben emisiones mientras la app esté en ejecución.

Explicación

Una aplicación Android crea siempre un subproceso o un hilo principal de ejecución. Todo el código de la aplicación se ejecuta dentro de este subproceso, lo que incluye dibujar y actualizar la interfaz de usuario.

Para mantener la respuesta de la aplicación, es esencial evitar el uso del hilo principal para realizar cualquier operación que pueda terminar bloqueándola.

Dos reglas para evitar bloquear la aplicación:
1. No bloquear el hilo principal.
2. Todas las operaciones de interfaz gráfica se hacen en el hilo principal.

Actividad

- ¿Cuál es la diferencia entre un BroadcastReceiver vs un BroadcastReceptor?
- ¿Cómo se reciben las emisiones en el contexto Android?
- ¿Qué pasa si se registra un receptor en una activity y ésta ya no se encuentra vigente?

Cierre

Tener este conocimiento como programador te apoya para que puedas diseñar aplicaciones que cuando ejecuten una tarea se desarrolle con éxito y evitando que el usuario se desespere, ya que si empieza a tocar muchas veces la pantalla del dispositivo, seguramente terminará por bloquearse, haciendo una mala experiencia con esa aplicación.

El saber hacerlo te facilita mucho el manejo de datos fuera de la aplicación, sobre todo imágenes o procesamiento masivo de datos, entre otros.

Universidad Tecmilenio

Desarrollo de aplicaciones en plataforma Android

Servicios

Introducción

En algunas ocasiones se requiere añadir un nuevo componente a la aplicación que se está creando, con el propósito de ejecutar algún tipo de acción en segundo plano, esto es, que no requiera una interacción directa con el usuario, sin embargo se requiere que ese componente siga activo aunque el usuario cambie de actividad, por ejemplo: escuchar música mientras se revisa otra aplicación. Entonces es el momento de crear un servicio.

Explicación

Para crear un servicio, debes crear una subclase de **Service** o usar una de sus subclases existentes.

Los siguientes son los métodos de devolución de llamada más importantes que se deben anular que manejan aspectos fundamentales del ciclo de vida del servicio y que proporcionan un mecanismo que permita que los componentes se enlacen con el servicio (si corresponde).

onStartCommand()

- El sistema invoca a este método llamando a startService() cuando otro componente, como una actividad, solicita que se inicie el servicio.
- Una vez que se ejecuta este método, el servicio se inicia y se puede ejecutar en segundo plano de manera indefinida. El implementarlo, será tu responsabilidad detener el servicio al terminar el trabajo.

Explicación

onBind()

- El sistema invoca a este método llamando a bindService() cuando otro componente quiere enlazarse con el servicio.
- Se debe proporcionar una interfaz que los clientes utilicen para comunicarse con el servicio devolviendo una Ibinder al implementar este método.
- Importante considerar que se debe devolver NULL si no deseas permitir los enlaces.

Explicación

onCreate()

- Se invoca este método para realizar procedimientos únicos de configuración cuando el servicio se crea por primera vez (antes de llamar a onStartCommand() o onBind()).
- Si el servicio ya se está ejecutando, el método no se llama.

onDestroy()

- El sistema invoca a este método cuando el servicio ya no se usa y se va a destruir.
- Esta es la última llamada que el servicio recibe.

Explicación

Los servicios Android ejercen doble función:

1. Indicar al sistema que el elemento que se está creando se debe ejecutar en segundo plano y por largo tiempo.

- Este tipo de servicio se inicia con el método startService(), que le indica su ejecución de forma indefinida hasta que se le indique lo contrario.

2. Permitir que la aplicación se comunique con otras aplicaciones. Este tipo de servicios son iniciados a través del método bindService(), que establece la conexión con el servicio.

Recuerda:

- Cada vez que un servicio es creado de acuerdo a una de las razones antes mencionadas, el sistema instancia el servicio y llama al método onCreate().

- El método implementará el comportamiento adecuado, generalmente creando un hilo de ejecución (thread) secundario para realizar el trabajo.

Actividad

- ¿Cuál es la diferencia entre las dos funciones de los servicios de Android ?
- ¿Por qué es importante conocer y entender el ciclo de vida de un servicio?
- Los servicios como reproducir música, realizar entrada o salida de archivos ¿corresponde a servicios de primer o segundo plano? ¿por qué?

Cierre

Los servicios de Android son esos procesos que se ejecutan en segundo plano y no tienen una interfaz de usuario.

Como ya sabes, los servicios de Android son ideales para situaciones en las que una aplicación necesita continuar realizando tareas, pero no necesariamente necesita una interfaz de usuario para ser visible para el usuario.

Universidad Tecmilenio

Desarrollo de aplicaciones en plataforma Android

Bases de Datos

Introducción

Las bases de datos

Las aplicaciones móviles que no necesitan almacenar al menos una cierta cantidad de datos persistentes son pocas y espaciadas. El uso de bases de datos es un aspecto esencial de la mayoría de las aplicaciones, que van desde aplicaciones que se basan casi en su totalidad en datos, hasta aquellas que simplemente necesitan almacenar pequeñas cantidades de datos, como la puntuación predominante de un juego.

Explicación

La importancia de SQLite.

SQLite es un sistema completo de base datos que llega a soportar gran cantidad de tablas, índices entre otros.

Se almacena en un archivador único, actualmente es usado por un gran número de aplicaciones móviles.

Uno de sus principios fundamentales es el esquema, ya que esta es una declaración de la forma en la que se encuentra organizada la base de datos.

Explicación

Una vez definido el diseño de la base de datos se deben aplicar los métodos que se creen, las cuales mantendrán dicha base de datos y las tablas.

Por ejemplo, si se usa **SQLiteOpenHelper** se debe crear una subclase que llegue anular todos los métodos de devoluciones llamadas onCreate y onUpgrate.
Para crear nuestra subclase de base de datos aplicaremos **SQLiteOpenHelper**.

Explicación

Para ingresar la información a la base de datos, se debe insertar el objeto **ContentValues** al método insert (), uno de los argumentos de insert () es el nombre de la tabla, en el segundo argumento le indica al framework que hacer cuando ContentValues este vacío, es decir que no tengan ningún valor.

Recuerda: el métodos **insert** () muestra la identificación de la fila recién creada y si este tiene un error cuando hayan insertado los datos.

Explicación

Para leer la información en la base de datos se utiliza el método **query()** donde pasan por criterios de selección y columnas que requiera, este método llega a combinar los elementos insert () y update ().

Para el tercer y cuarto argumento (selection y selectionArgs) que se llegan a combinar para crear una cláusula WHERE. Como los argumentos se proporcionan por separado de la consulta de selección, se escapan antes de combinarse, lo que hace que tus instrucciones de selección sean inmunes a la inserción SQL.

Explicación

Para borrar filas de una tabla, debes proporcionar criterios de selección que identifiquen las filas para el método delete(). El mecanismo funciona igual que los argumentos de selección del método query(). Divide la especificación de selección en una cláusula de selección y argumentos de selección. La cláusula define las columnas que se comprobarán y también permite combinar pruebas de columnas. Los argumentos son valores para probar que están vinculados a la cláusula.

Para actualizar una base de datos se debe modificar un subconjunto de los valores de la base de datos, a través del método **update().**

La actualización de la tabla combina la sintaxis ContentValues de insert() con la sintaxis WHERE de delete().

Actividad

- ¿Cuál es la función de la clase HandlerDataBase?
- ¿Por qué es importante sabe identificar la llave primaria en una base de datos? ¿Es posible tener varias llaves primarias?
- ¿Cuáles usos tienen las bases de datos en las aplicaciones móviles?

Cierre

Es importante que estés practicando el uso de las base de datos. La práctica es importante para lograr más control al implementarlas.

Haz aprendido la descripción general del sistema de administración de bases de datos SQLite con el sistema operativo Android, junto con un esquema de las clases de SDK de Android que facilitan la creación y almacenamiento de bases de datos, así como la ingresar, borrar y actualizar información.

Universidad Tecmilenio

Desarrollo de aplicaciones en plataforma Android

Multimedia

Introducción

Multimedia en Android

Es prácticamente un requisito integrar en las aplicaciones elementos multimedia, como el audio o el video. Pues si pensamos por ejemplo en aplicaciones turísticas, audio, para aprender algún idioma, o simplemente reproductores de música, entonces pensamos en la necesidad de diseñar aplicaciones multimedia.

Android ofrece la posibilidad de trabajar con elementos multimedia integrados previamente en la aplicación o incluso obtenidos de forma dinámica desde Internet (streaming).

Explicación

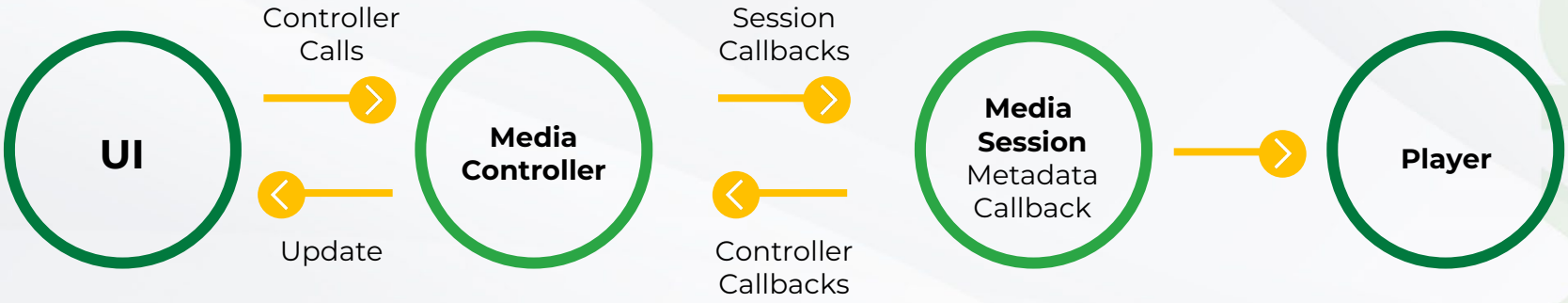
En Android, es posible compilar tu propio reproductor desde cero o a través de las siguientes opciones:

- Clase MediaPlayer, que proporciona las funciones básicas para un reproductor sencillo que admita los formatos de audio y video y las fuentes de datos más comunes.
- ExoPlayer es una biblioteca de código abierto que expone las API de audio de Android de niveles inferiores. ExoPlayer admite funciones de alto rendimiento.

Explicación

El marco de trabajo de Android define dos clases (una sesión multimedia y un controlador multimedia) que imponen una estructura bien definida para compilar una app de reproductor multimedia.

La sesión multimedia y el controlador multimedia se comunican entre sí a través de devoluciones de llamada predefinidas con acciones estándar del reproductor (reproducir, pausar, detener, etc.), además de llamadas personalizadas extensibles propias de tu aplicación.



Explicación

Una sesión multimedia se encarga de toda la comunicación con el reproductor. Mientras que el controlador multimedia aísla la Interfaz gráfica (IU). El código de la IU se comunica solamente con el controlador multimedia, no con el reproductor en sí.

El controlador multimedia convierte las acciones del control de transporte en devoluciones de llamada a la sesión multimedia.

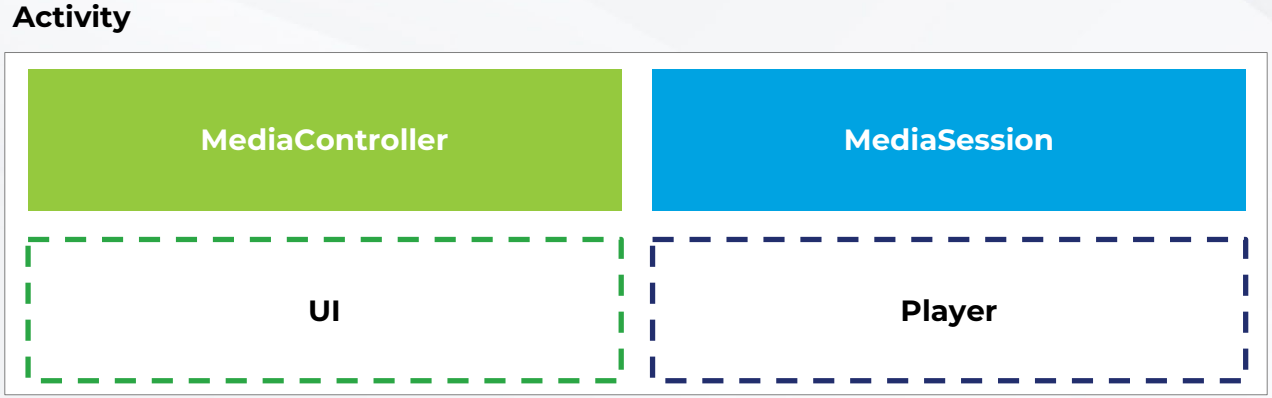
Explicación

Diferencias entre apps de video y de audio

Cuando se reproduce un video, involucra la vista y el oído. Pero al reproducir audio, solamente se usa el oído, por ello hay un diseño distinto para cada caso.

App de video

Una app de video necesita una ventana para ver el contenido, por lo que se suelen implementar como una sola actividad de Android. Ejemplo de la activity de video:

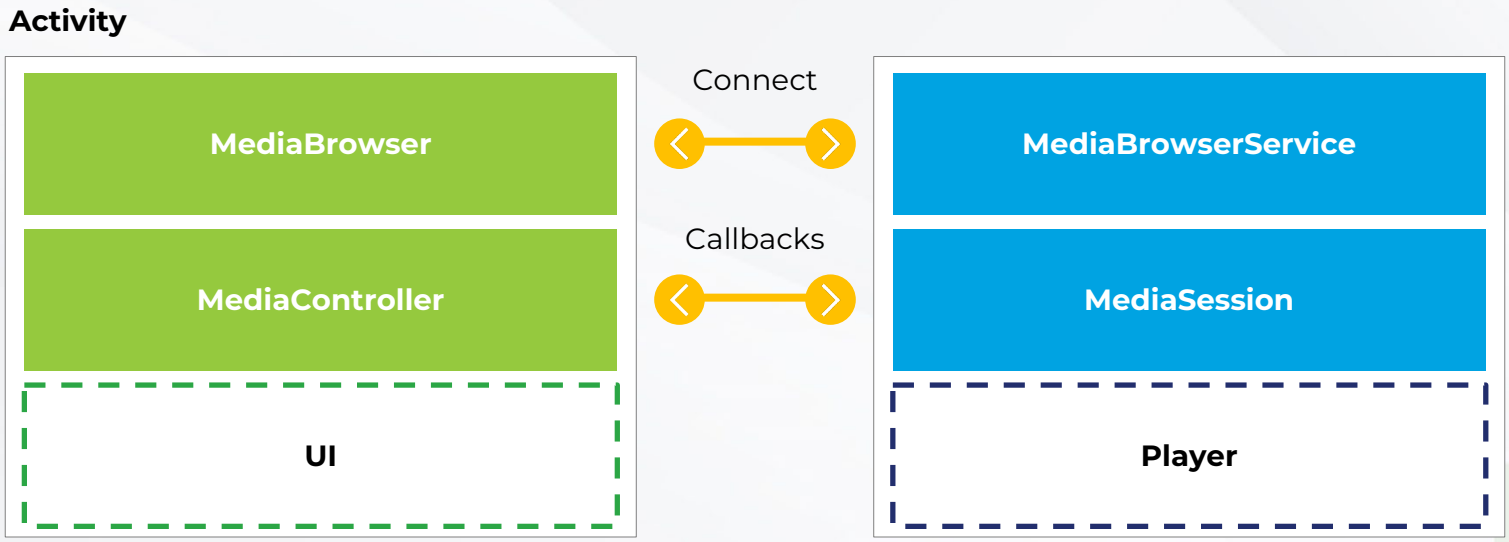


Explicación

Un reproductor de audio no siempre necesita que su interfaz gráfica esté visible. Una vez que comienza a reproducir audio, puede ejecutarse como una tarea en segundo plano, ya que el usuario puede usar otra app y seguir escuchando el contenido.

Una interfaz gráfica suele tener una duración corta en comparación con un reproductor, que puede ejecutarse durante mucho tiempo sin necesidad de esa interfaz.

Ejemplo de una activity de audio:



Actividad

- ¿Cuál es la principal diferencia entre las Apps de audio y video?
- ¿Cuál es la diferencia entre una sesión multimedia y un controlador multimedia?
- ¿Por qué es posible que una app de audio pueda ejecutarse en segundo plano?

Cierre

Hay una serie de aplicaciones cuyo factor importante de éxito lo han hecho el poder tomar una foto o hacer un video, ya que actualmente las personas usan mucho de su tiempo en estar viendo este tipo de multimedia, mientras realizan otra actividad como ir en transporte, hacer ejercicio o seguir programas de televisión vía internet.