Apuntes

Descargar estos apuntes

Tema 3 . Componentes fundamentales de una aplicación Android

Índice

Indroducción

En este tema estudiaremos los componentes de una aplicación Android y sus respectivas funciones.
Para cualquier aplicación Android, en general, será necesario construir una ventana en la que se irán añadiendo componentes, cada componente es un punto de entrada por el que el sistema o un usuario interactúa con tu aplicación. Algunos componentes dependen de otros.
Lo primero que haremos será comprender que es el contexto de una aplicación y como permite relacionar los componentes.
Luego veremos el concepto de **Actividades, Intents, Servicios, BroadCast Receivers y Content Providers**.

Contexto de una aplicación

El contexto de una aplicación es una interfaz entre la aplicación y el sistema operativo, la cual describe la información que representa tu aplicación dentro del ambiente del sistema operativo. También permite acceder a los recursos de la aplicación y coordinar el funcionamiento de los bloques de la aplicación.
El contexto representa toda la meta-información sobre las relaciones que tiene la aplicación con otras aplicaciones o el sistema y podemos implementarlo a través de la **clase abstracta Context**.
En Android nos encontramos con los siguientes tipos de contextos:

- **Aplicación**. Este contexto engloba a todos los demás, y cubre todo el ciclo de vida de la aplicación desde que la arrancamos hasta que muere. Por lo que cada aplicación tiene ***un único contexto de aplicación***. El Context de la aplicación vive hasta que se termina por completo la aplicación (hasta que muere, no hasta que se pausa)
  Se puede acceder desde una ***Activity*** o un ***Service*** con `getApplication()` o desde cualquiera que herede de Context con `getApplicationContext()`.
- **Activity o Service**. Como hemos dicho, un Context vive tanto como el elemento al que pertenece, por lo que depende del ciclo de vida. Así, un Context de una Activity vivirá el mismo tiempo que vive esta la activity y siempre será vivirá menos que el de la Aplicación.

Componentes de una aplicación

Activity

Las Activitys son los componentes visibles de la aplicación, tienen una interface de usuario que permite interactuar con ellas.
La mayoría de las aplicaciones permiten la ejecución de varias acciones a través de la existencia de una o más pantallas. Por ejemplo: *una aplicación de mensajes de texto podrá tener la lista de*

contactos que se muestra en una ventana, mediante el despliegue de una segunda ventana el usuario puede escribir el mensaje al contacto elegido, y en otra tercera puede repasar su historial de mensajes enviados o recibidos.
Cada una de estas ventanas puede estar representada a través de un componente Activity, de forma que navegar de una ventana a otra implica lanzar una actividad o dormir otra.

Services

Un servicio es una entidad que **ejecuta instrucciones en segundo plano** sin que el usuario lo note en la interfaz. Son muy utilizados para realizar acciones de larga duración mientras las actividades muestran otro tipo de información. Por ejemplo guardar la información en la base de datos, escuchar música mientras se ejecuta la aplicación, administrar conexiones de red, etc.
Los servicios también tienen un ciclo de vida como las actividades, pero este es más corto. Solo se comprende de los estados de Creación, Ejecución y Destrucción.

Intens

Un Intent **es un objeto de mensajería** que puedes usar para solicitar una acción de otro componente de una app. Los intents facilitan la comunicación entre componentes de varias formas, desde la comunicación entre actividades, entre fragments, servicios, etc.
Los Intents pueden ser **implícitos o explícitos**. Son explícitos cuando ejecutamos un componente en específico. Son implícitos cuando se entrega una referencia genérica sobre alguna condición, y aquellas entidades que cumplan ese requisito serán presentadas como candidatas para la tarea.
Un ejemplo de **Intent implícito**: *sería cuando deseas compartir un sitio web en alguna red social. Para ello Android nos ofrece en un pequeño dialogo de lista para que elijamos entre cuales de todas las aplicaciones sociales deseamos elegir para compartir. No especificamos que aplicación queríamos, simplemente se mostraron todas las aplicaciones que podrían responder a ese tipo de acción*.
Un I**ntent explicito se** da cuando invocamos una actividad con un tipo de clase especifico. Por ejemplo, *supón que dentro de nuestra aplicación tenemos dos actividades llamadas A y B. Si decidimos iniciar la actividad B desde la actividad A usaríamos un Intent explicito debido a que "B" es un tipo definido*.

Content provider

Con el componente **Content Provider**, cualquier aplicación en Android puede almacenar datos en un fichero, en una base de datos SQLite o en cualquier otro formato que considere.
Además, estos **datos pueden ser compartidos entre distintas aplicaciones**. Una clase que implemente el componente Content Provider contendrá una serie de métodos que permite almacenar, recuperar, actualizar y compartir los datos de una aplicación.
Por defecto, el API de Android trae consigo Content Providers predefinidos para intercambiar

información de audio, vídeo, imágenes, e información personal. Pero si en algún momento deseas intercambiar información con una estructura personalizada debes crear tu propia subclase heredada de la clase ContentProvider.

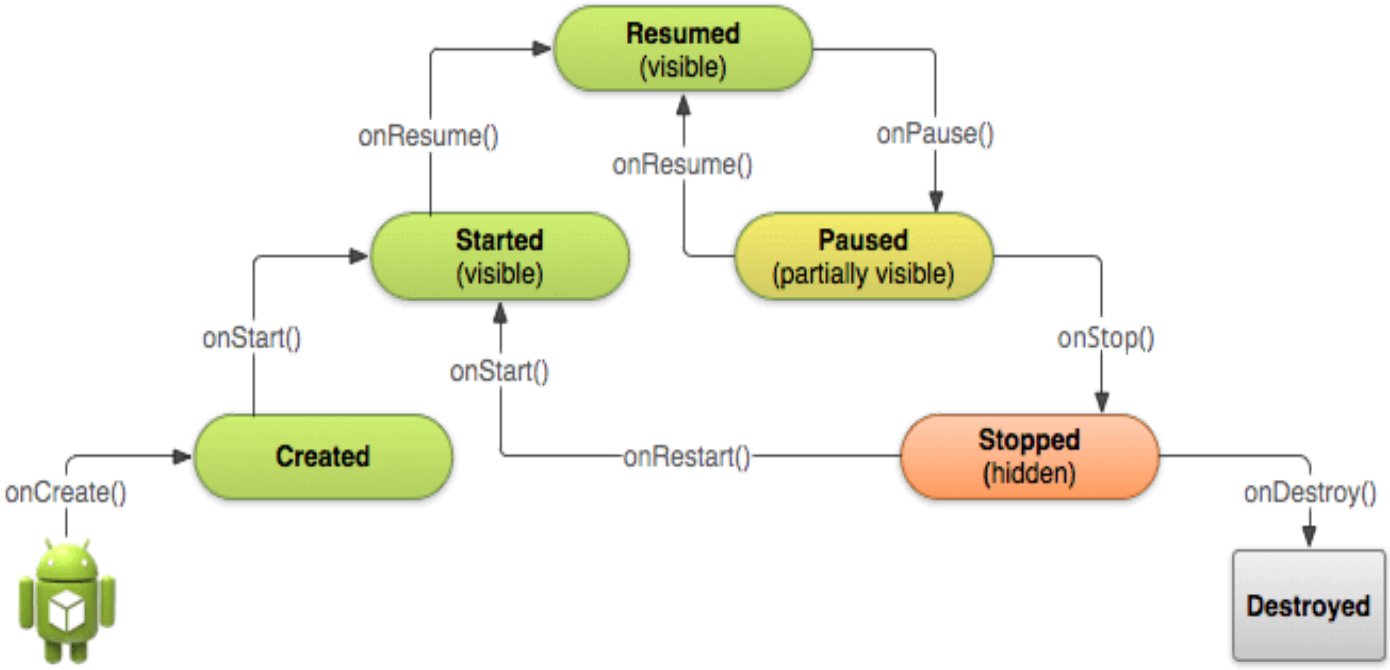
Broadcast receiver

Finalmente, el último componente clave son los **Broadcast Receives**, o receptores de emisión, que **reaccionan a intents específicos pudiendo a su vez ejecutar una acción** o iniciando una activity o pueden devolver otro intent al sistema para que siga la ejecución. Realmente este tipo de componentes son capaces de procesar los mensajes del sistema operativo.

Ciclo de vida de un Activity

Como hemos comentado en el punto anterior. La activity es uno de los componentes esenciales de una aplicación Android, además que es el componente que tiene asociada la interfaz de usuario. Una misma aplicación puede tener una o varias actividades y Android permite controlar por completo el ciclo de vida de cada una de estas.
Los estados por los cuales puede pasar una Activity son los siguientes: **Creación, Ejecución, Reanudación, Pausa, Parada y Destrucción**.

A la relación entre ellos se le llama **Ciclo de vida de una Actividad**.



El ciclo de vida de una Activity nos describe los estados y las transiciones entre estados que una determinada Activity, como hemos comentado son:

- Creación: una actividad se ha creado cuando su estructura se encuentra en memoria, pero esta no es visible aun. Cuando el usuario presiona sobre el icono de la aplicación en su

dispositivo, el método `onCreate()` es ejecutado inmediatamente para cargar el layout de la actividad principal en memoria.

- Ejecución-Reanudación: después de haber sido cargada la actividad se ejecutan en secuencia el método `onStart() y onResume()`. Aunque `onStart()` hace visible la actividad, es `onResume()` quien le transfiere el foco para que interactúe con el usuario.
- Pausa: una actividad está en pausa cuando se encuentra en la pantalla parcialmente visible. Un ejemplo: *cuando se abren diálogos que toman el foco superponiéndose a la actividad*. El método llamado para la transición hacia la pausa es `onPause()`.
- Detención: una actividad está detenida cuando no es visible en la pantalla, pero aún se encuentra en memoria y en cualquier momento puede ser reanudada. Cuando una aplicación es enviada a segundo plano se ejecuta el método `onStop()`. Al reanudar la actividad, se pasa por el método `onRestart()` hasta llegar a el estado de ejecución y luego al de reanudación.
- Destrucción: cuando la actividad ya no existe en memoria se encuentra en estado de destrucción. Antes de pasar a destruir la aplicación se ejecuta el método `onDestroy()`. Es común que la mayoría de actividades no implementen este método, a menos que deban destruir procesos como servicios en segundo plano.

Aunque tu aplicación puede tener varias actividades en su estructura, se debe definir una actividad principal. Para hacerlo se debe especificar en tu archivo Android Manifest un componente `<actitivy>`, con un componente hijo `<intent-filter>`. Dentro de este componente declararas dos nuevos componentes, `<action>` y `<category>`. Al primero le asignaras el elemento enumerado `MAIN` y al segundo el elemento enumerado `LAUNCHER`.

Probar el código del ciclo de vida de una actividad. EjercicioResueltoCicloVida

Comunicación de Activitys a través de Intents