Android: Activity

1 Las actividades

La actividad (Activity) es el principal componente de una aplicación de Android y se encarga de gestionar gran parte de las interacciones con el usuario. A nivel de lenguaje de programación, una actividad hereda de la clase Activity y se traduce en una pantalla. Por tanto, y en términos generales, una aplicación de Android tendrá tantas Activities como pantallas distintas tenga.

Una aplicación generalmente consiste en múltiples actividades vinculadas de forma flexible entre sí. En el diseño de una aplicación, las Activities tendrán una secuencia de aparición según la lógica con la que se haya construido la misma. Aparecerán una u otras pantallas según el flujo del programa y las decisiones que en cada caso tome el usuario.

Otro punto a tener en cuenta es que si tu app tiene varias actividades es posible marcar una como la actividad principal (main). Esto nos permite que al presionar el icono de la app desde el Launcher, dicha actividad sea iniciada.

Desde una pantalla principal (main) se podrán realizar desplazamientos de una a otra Activity y, conforme se va navegando por las diferentes pantallas de la aplicación, el dispositivo va recordando la secuencia visitada, de tal manera que si retrocede utilizando los botones del dispositivo o la lógica de la aplicación, irán apareciendo pantallas en sentido inverso a como se habían visionado. Android almacena la posición de cada una de las Activities (pantallas) en una pila, Stack (también llamada Back Stack), en este caso, de tipo LIFO (Last In, First Out), acrónimo que significa que el último en entrar será el primero en salir.

El sistema de funcionamiento de esta pila es muy simple: cuando la actividad en curso inicia otra, esta nueva actividad obtiene el foco y es empujada a la parte superior de la pila; la actividad anterior permanece en la pila, pero detenida. El conjunto de actividades que tienen una relación lógica y que se encuentran en la pila se denomina tarea (Task).

Si el usuario regresa (Back), la actividad que tenía el foco se elimina, y la actividad previa y que se encontraba por debajo de ella en la pila se restaura y vuelve a coger el foco.

Si se continúa presionando Atrás, se irán eliminando sucesivamente tareas hasta llegar a la pantalla de inicio. Cuando todas las actividades se han eliminado de la pila, la tarea deja de existir y desaparece de la memoria.

Una tarea puede moverse en conjunto a un segundo plano cuando el usuario comienza una nueva tarea o va a la pantalla Inicio. Si está en segundo plano, todas las actividades de la tarea se paran, pero

el Back Stack de esta tarea se conserva. Si una tarea vuelve a situarse en primer plano, esta se volvería a posicionar en el lugar donde se había dejado.

Cuando se detiene una de las actividades, y teniendo en cuenta las características del dispositivo (especialmente la memoria), es posible que el sistema, si necesita recursos de memoria, elimine esa actividad completamente, pero aun así mantiene su lugar en la pila de actividades para cuando necesite volver a primer plano.

Cuando esto suceda, el sistema debe volver a crearla (en lugar de reanudarla), y para evitar que se pierda el trabajo que tenía en curso, se implementa el método onSaveInstanceState(), que asegura la recuperación de variables y otros valores ligados a ella.

Por lo que se ha visto hasta ahora, las actividades funcionan dentro del dispositivo como procesos que pueden estar en diferentes estados o momentos de ejecución.

- **Foreground Process**. Es el proceso en que se encuentra la actividad que se está mostrando en pantalla y con la que el usuario puede interactuar.
- **Visible Process**. La actividad es visible, pero aún no se puede interactuar con ella. Aunque este estado parezca tener poco sentido, su uso tendrá especial utilidad para el programador, ya que es una situación considerada importante por el sistema operativo.
- **Service Process**. Proceso que lleva una actividad que está funcionando como un servicio. Hace cosas en segundo plano y generalmente importantes.
- **Background Process**. Aquel que tiene una actividad no visible y situada en un nivel inferior de la pila.
- **Empty Process**. Se trata de aquellos procesos que no contienen nada y que Android suele usar para crear sobre ellos un proceso nuevo.

2 Ciclo de vida de una aplicación

Una actividad puede estar en diferentes periodos de funcionamiento que denominaremos estados, y que van a depender del flujo de la aplicación, de la interacción con el usuario y de las necesidades de recursos del sistema.

La actividad debe relacionarse con el sistema operativo y con otros programas que pueden estar en diferentes estados en el dispositivo móvil. Teniendo en cuenta el estado de funcionamiento de la misma, para Android, la actividad puede ser:

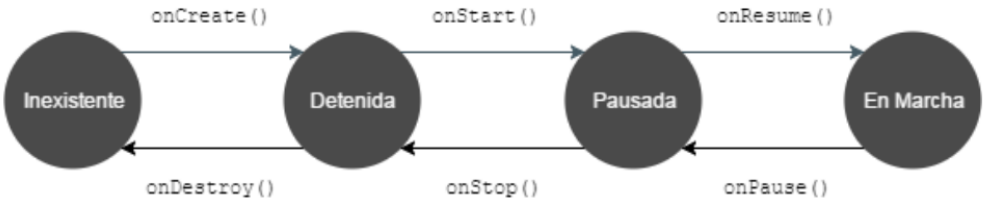
- **Inexistente**. Cuando no ocupa memoria, por lo que, evidentemente, no se puede interactuar con ella. Una Activity está pausada o parada, el sistema puede eliminarla de memoria matando el proceso o mandándole la señal de que finalice. Cuando vuelva a ser visible por el usuario, ésta debe ser completamente reiniciada.

- **Detenida**. Sí ocupa memoria, aunque no es visible y, por tanto, tampoco es accesible. La Activity deja de ser visible (por ejemplo, cuando se inicia otra Activity que ocupe la pantalla entera). Aun así sigue permaneciendo en memoria, aunque puede ser 'matada' por el sistema en condiciones de baja memoria.
- **Pausada**. Es visible, pero no se puede interactuar con ella. La Activity ha perdido el foco, pero sigue siendo visible (por ejemplo, cuando se inicia otra Activity que no ocupa la pantalla entera). Una Activity detenida sigue permaneciendo en memoria pero puede ser 'matada' o interrumpida por el sistema en condiciones de baja memoria.
- **Activa**. Está en primer plano del dispositivo y el usuario puede interrelacionarse con ella.

Estos estado suelen ir en sucesión creciente cuando la actividad se pone en marcha, y en sentido decreciente cuando desaparece y se destruye, por lo que el conjunto de estados en el que se puede encontrar una actividad es comúnmente denominado ciclo de vida.

Los cambios de estado pueden ser controlados a través de métodos, llamados por Android tipo callback.

Cuando un usuario navega por una app, sale de ella y vuelve a entrar, las instancias de Activity de la app pasan por diferentes estados de su ciclo de vida. La clase Activity proporciona una serie de devoluciones de llamada que permiten a la actividad saber que cambió un estado, es decir, que el sistema está creando, deteniendo o reanudando una actividad, o finalizando el proceso en el que se encuentra.



Para navegar por las transiciones entre las etapas del ciclo de vida de una actividad, la clase Activity proporciona un conjunto básico de seis devoluciones de llamadas: *onCreate()*, *onStart()*, *onResume()*, *onPause()*, *onStop()* y *onDestroy()*. El sistema invoca cada una de estas devoluciones de llamada cuando una operación entra en un nuevo estado.

1. **onCreate()**: es llamado cuando la actividad es invocada por primera vez. La tareas que se realiza son:
    - Se crean las vistas (el interfaz de usuario) con *setContentView()*
    - Se hace una reserva de memoria para los datos,
    - Se obtienen las referencias de view e instancias de componentes mediante *findViewById()*
    - Se inician las consultas iniciales en las fuentes de datos.
    - Se guardan las variables del entorno

2. **onStart()**: este método sucede al anterior. La actividad se hace visible para el usuario, pero aún no se puede interactuar con ella. Suele ser el lugar donde ubicar instrucciones asociadas al interfaz de usuario.

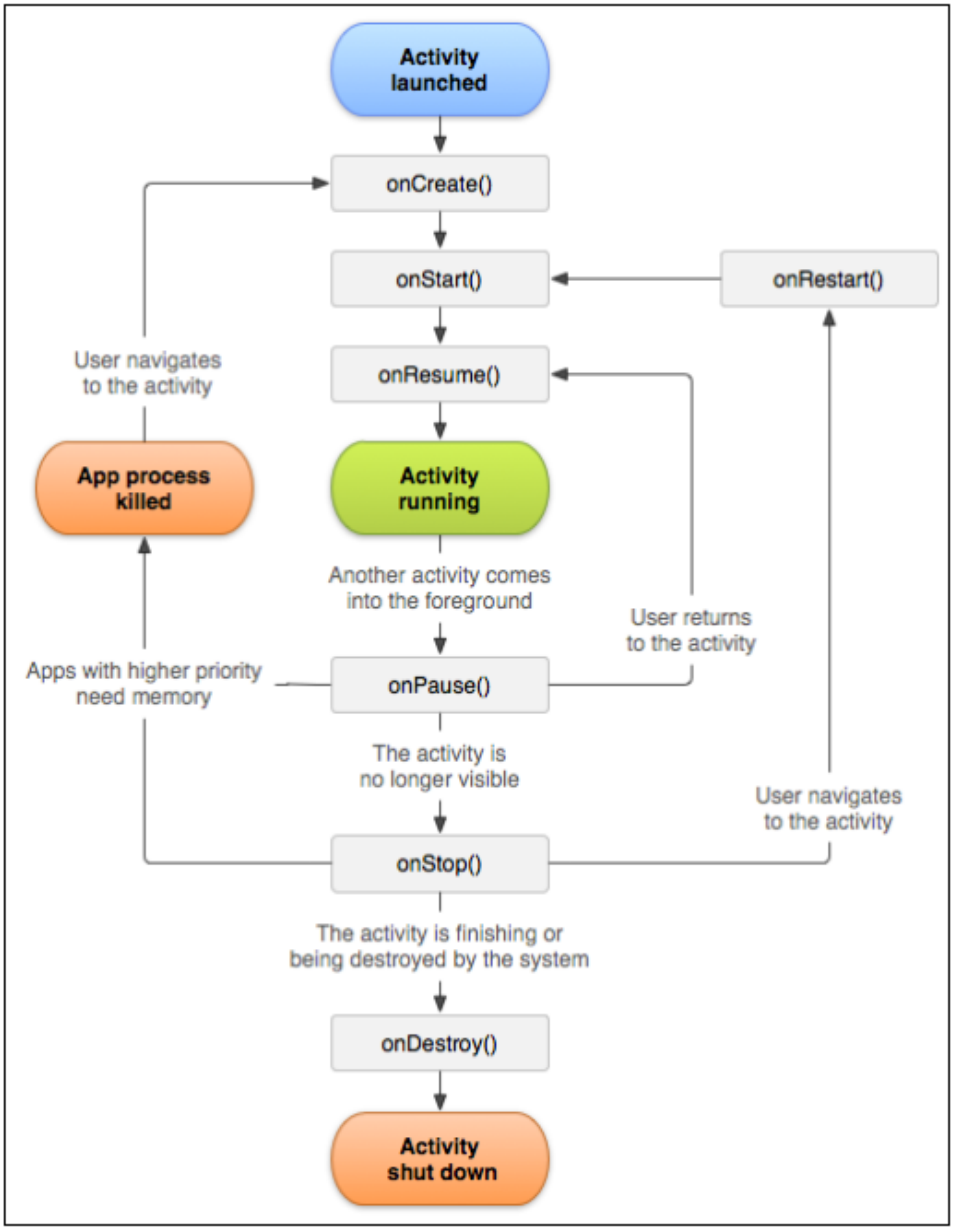
3. **onResume()**: un método que es llamado cuando la actividad puede interactuar con el usuario. Sucede al anterior y precede al onPause(). En este periodo se suelen activar sensores, cámara, ejecutar animaciones, actualizar información...

4. **onPause()**: ocurre cuando nuestra actividad pasa a un segundo plano, bien porque está en proceso de destrucción o porque otra actividad pasa a ocupar el primer plano. Es cuando se detienen procesos, animaciones... En este periodo la actividad es parcialmente visible. Puede ocurrir que vuelva a primer plano onResume() o que sea parada onStop(), haciéndose invisible para el usuario.

5. **onStop()**: este método es invocado cuando la actividad ya no es visible al usuario y otra actividad ha pasado a primer plano. Se pausan animaciones, determinados servicios se detienen (GPS) y se minimizan los recursos consumidos por la actividad, aunque la actividad aún está en memoria. Si queremos reactivarla, se utiliza onRestart(), volviendo a primer plano, o onDestroy(), si queremos eliminarla definitivamente.

6. **onRestart()**: llamado cuando una actividad parada vuelve a primer plano. Siempre es seguido de un onStart().

7. **onDestroy()**: es llamado cuando la actividad es definitivamente destruida y eliminada de memoria. En su interior se suelen limpiar y liberar recursos, por ejemplo, la cámara.



Se puede decir que el ciclo de vida completo de una actividad transcurre entre la llamada a onCreate() y la llamada a onDestroy().

El ciclo de vida visible de una actividad es aquel en que el usuario puede ver la actividad en pantalla y que transcurre entre la llamada a onStart() y la llamada a onStop().

Por último, el ciclo de vida en primer plano de una actividad transcurre entre la llamada a onResume() y la llamada a onPause(). Es el momento que la actividad tiene el foco en el dispositivo y se encuentra por encima de todas las demás actividades que están activas en memoria.

Referencias

https://developer.android.com/guide/components/activities/intro-activities?hl=es-419