Android: *Tasks* 'Tareas' y el *Back Stack*

Carlos Alberto Gomez Ormachea
Jalasoft
Alberto.Gomez@Jalasoft.com

RESUMEN

Una aplicación android contiene diferentes *activities* 'actividades'. Cada actividad debería ser diseñada entorno a una acción especifica que puede realizar el usuario y que puede iniciar otras actividades. Por ejemplo, una aplicación de correo electrónico puede tener una actividad para mostrar una lista de los correos electrónicos nuevos. Cuando el usuario selecciona un correo, se abre una nueva actividad para ver ese correo.

Una actividad también puede iniciar actividades existentes en otras aplicaciones del dispositivo. Por ejemplo, si la aplicación quiere mandar un correo electrónico, se puede definir un *intent* 'intención' para que realice una acción *send* 'enviar' incluyendo datos, como la dirección de correo electrónico y el mensaje. En ese momento se abre una actividad de otra aplicación que va a gestionar este tipo de *intent* 'intención'. En este caso, el *intent* es para mandar un correo electrónico, por lo que se inicia una actividad *compose* 'componer' de correo electrónico de la aplicación (si varias actividades mantienen al mismo *Intent*, el sistema permite al usuario selecciona cual quiere utilizar). La actividad se reanuda el momento que se envia el email, causando un efecto que hace que el email aparente ser parte de la aplicación. El hecho que las actividades sean de distintas aplicaciones es manejado por Android y transparente al usuario, al guardarse ambas actividades en la misma tarea.

Palabras clave

Android, *Activity*, *Task*, *Backstack*, *Intent*.

1. INTRODUCCIÓN

Una tarea es una colección de actividades con las cuales interactua el usuario cuando quiere realizar un determinado trabajo. Las actividades son ordenadas en un *stack* 'pila' (también llamado *back stack*);

La pantalla Inicio es el lugar de inicio para la mayoría de las tareas del dispositivo. Cuando el usuario hace click en un icono, la tarea de la aplicación viene a un primer plano. Si no existe una tarea (la aplicación no se ha usado recientemente) se crea una nueva y la actividad *main* 'principal' de la aplicación se abre como la actividad raiz en la pila.

2. COMPORTAMIENTO DEL *BACKSTACK*

Cuando la actividad en curso inicia otra, la nueva actividad es empujada arriba de la pila y obtiene el focus. La actividad anterior permanece en la pila, pero está detenida. Cuando la actividad para, el sistema retiene el estado actual de la interfaz de usuario. Cuando el usuario hace click en la tecla *back* 'atrás', la actividad en curso salta desde arriba de la pila (la actividad se elimina) y la actividad previa se reanuda (se restaura el estado previo de su IU). Las actividades en la pila nunca se reorganizan, solo se meten o sacan de la pila - se meten en la pila cuando se inicia la actividad en curso y son sacadas cuando el usuario la abandona utilizando la tecla *back*. De esta manera, el *back stack* funciona mediante la estructura de objeto; "último en entrar, primero en salir".

Si el usuario sigue haciendo click en la tecla *back* 'atrás', cada actividad de la pila se saca, revelando la anterior, hasta que el usuario retorna a la pantalla de Inicio (o a cualquier actividad que estuviera ejecutándose en el comienzo de la tarea). Cuando todas las actividades son sacadas de la pila, la tarea deja de existir.

Una tarea es una unidad cohesiva que puede moverse a un "segundo plano" cuando el usuario comienza una nueva tarea o ir a la pantalla Inicio, a través de la tecla *home* 'inicio'. Mientras está en un segundo plano, todas las actividades en la tarea se paran, pero el *back stack* para la tarea se mantiene intacto. Una tarea puede retornar a un "primer plano" por lo que los usuarios pueden volver al lugar donde lo había dejado. Imaginese, por ejemplo, que la tarea en curso (Tarea A) tiene tres actividades en su pila - dos por debajo de la actividad en curso. El usuario hace click en la tecla *home*, e inicia una nueva aplicación. Cuando la pantalla de inicio aparece, la tarea A se sitúa en un segundo plano. Cuando se inicia la nueva aplicación, el sistema inicia una tarea para esa aplicación (Tarea B) con su propia pila de actividades. Después de interactuar con esa aplicación, el usuario retorna a la página de Inicio y selecciona la aplicación que originariamente había iniciado a la Tarea A. Ahora, la tarea A va a un primer plano, las tres actividades están intactas en su pila y se reanuda la actividad que está en la parte superior

de la pila. En este punto, el usuario puede volver a la tarea B si va a *home* y selecciona el icono de la aplicación que empezó la tarea (o si hace click y mantiene la tecla *home* para ver las tareas recientes y selecciona una). Esto es un ejemplo de *multitasking* 'multitarea' en Android.

Dado que las actividades en el *back stack* nunca son reorganizadas, si la aplicación permite que los usuarios inicien una actividad concreta desde más de una actividad, se crea una nueva instancia de esa actividad y se mete en la pila (en vez de traer a la parte de arriba de la pila a una instancia anterior de la actividad). De esta manera, una actividad de la aplicación puede ser instanciada varias veces (incluso desde tareas diferentes). Así, si el usuario navega hacia atrás usando la tecla *back* 'atrás', aparecerá cada instancia de la actividad en el orden en el que fueron abiertas (cada una con su propio estado de IU). Sin embargo, se puede modificar este comportamiento si no se quiere instanciar una actividad más de una vez. Más adelante en la sección sobre Gestionar tareas se estudiará en profundidad.

Como resumen del comportamiento por defecto de las actividades y tareas tenemos:

- Cuando la actividad A inicia a la actividad B, la actividad A se detiene, pero el sistema mantiene su estado (como la posición del scroll y el texto introducido en los formularios). Si el usuario hace click en la tecla *back* durante la actividad B, la actividad A se reanuda con su estado restablecido.
- Cuando el usuario abandona una tarea haciendo click en la tecla *home*, la actividad en curso se detiene y su tarea se mueve a un segundo plano. El sistema retiene en la tarea el estado de cada una de las actividades. Si el usuario reanuda la tarea seleccionando el icono que inicia la tarea, la tarea se mueve a un primer plano y reanuda la actividad de la parte superior de la pila.
- Si el usuario hace click en la tecla *back*, la actividad en curso sale de la pila y es eliminada. Se reanuda la actividad anterior. Cuando una actividad es eliminada, el sistema no conserva el estado de la actividad.
- Las actividades pueden ser instanciadas varias veces, incluso desde otras tareas.

3. GESTIONAR TAREAS

La manera en la que *Android* gestiona las tareas y el *back stack*, - colocando todas las actividades iniciadas en la misma tarea y en una pila - "último en entrar, primero en salir"; funciona muy bien para la mayoría de las aplicaciones y no hay que preocuparse sobre como están asociadas las actividades con las tareas o como existen en el *back stack*. Sin embargo, se puede querer interrumpir el comportamiento normal. Quizás se necesite en la aplicación, que una actividad inicie una nueva tarea cuando se inicie (en vez de colocarse dentro de la tarea en curso); o, que cuando se inicie una actividad, se traiga hacia delante una instancia ya existente (en vez de crear una nueva instancia en la parte superior del *back stack*); o, que cuando el usuario deja la tarea, se eliminen todas las actividades exceptuando la actividad raiz.

Se puede hacer todo esto y más, con los atributos del elemento <*activity*> del archivo *manifest* 'manifesto' y con los *flags* 'banderas' en el *intent* que se pasan al método *startActivity()*.

A este efecto, los principales atributos <*activity*> que se pueden utilizar son:

- *taskAffinity*
- *launchMode*
- *allowTaskReparenting*
- *clearTaskOnLaunch*
- *alwaysRetainTaskState*
- *finishOnTaskLaunch*

Y los principales *flags* del *intent* que se pueden utilizar son:

- *FLAG_ACTIVITY_NEW_TASK*
- *FLAG_ACTIVITY_CLEAR_TOP*
- *FLAG_ACTIVITY_SINGLE_TOP*

En las siguientes secciones, vamos a ver como se pueden utilizar estos atributos del archivo *manifest* y los *flags* del *intent* para definir como las actividades son asociadas con las tareas y como se comportan en el *back stack*.

4. DEFINIR MODO DE INICIO

Los modos de inicio te permiten definir como una nueva instancia de una actividad está asociada con la tarea en curso. Se pueden definir los diferentes modos de inicio de dos maneras:

Utilizando el archivo *manifest* 'manifesto'
Cuando se declara una actividad en el archivo *manifest*, se puede especificar como debe asociarse la actividad con las tareas cuando se inicia.

Utilizando *flags* de un *intent*
Cuando se llama al método *startActivity()*, se puede incluir un *flag* en el *Intent* que declara como (o si) la nueva actividad debería asociarse con la tarea en curso.

Así, si la Actividad A inicia la Actividad B, la Actividad B puede definir en su archivo *manifest* como debe asociarse con la tarea en curso y la Actividad A también puede pedir como la Actividad B se debería asociar con la tarea en curso. Si ambas actividades definen como la Actividad B debe asociarse con una tarea, entonces la petición de la Actividad A (definida en el *intent*) será tomada en cuenta antes que la petición de la Actividad B (definida en su archivo *manifest*).

5. UTILIZAR EL ARCHIVO *MANIFEST* 'MANIFIESTO'

Cuando se declara una actividad en el archivo *manifest*, se puede especificar como se debe asociar la actividad con una

tarea utilizando el atributo *launchMode* del elemento *<activity>*

El atributo *launchMode* especifica una instrucción de como la actividad debe ser iniciada en una tarea. Existen cuatro diferentes modos de inicio que se pueden asignar al atributo *launchMode*:

***"standard"* (modo por defecto).** Por defecto. El sistema crea una nueva instancia de la actividad en la tarea desde la cual se inició y le redirecciona el *intent*. La actividad puede ser instanciada varias veces, cada instancia puede pertenecer a tareas diferentes y una tarea puede tener varias instancias.

"singleTop". Si una instancia de la actividad ya existe en la parte superior de la tarea en curso, el sistema redirecciona el intent hasta esa instancia a través de una llamada a su método *onNewIntent()*, en vez de crear una nueva instancia de la actividad. La actividad puede ser instanciada varias veces, cada instancia puede pertenecer a diferentes tareas, y una tarea puede tener varias instancias (pero solo si la actividad en la parte superior de la *back stack* no es una instancia ya existente de la actividad).

Por ejemplo, una tarea del *back stack* consta de una actividad raiz A con actividades B, C y D encima (la pila es A-B-C-D; D encima). Llega un *intent* para una actividad de tipo D. Si D tiene el modo por defecto *"standard"*, se inicia una nueva instancia de la clase y la pila se convierte en A-B-C-D-D. Sin embargo, si el modo de inicio de D es *"singleTop"*, la instancia existente de D entrega el *intent* a través del método *onNewIntent()*, y dado que está en la parte superior de la pila, la pila permanece como A-B-C-D. Sin embargo, si un *intent* llega para una actividad de tipo B, se añade una nueva instancia de B a la pila, aún si el modo de inicio es *"singleTop"*.

"singleTask". El sistema crea una nueva tarea e instancia la actividad en la raiz de la nueva actividad. Sin embargo, si una instancia de la actividad ya existe en una tarea por separado, el sistema redirecciona el *intent* hacia la instancia ya existente a través de la llamada al método *onNewIntent()*, en vez de crear una nueva instancia. Sólo puede existir a la vez una instancia de la actividad.

"singleInstance". Igual que *"singleTask"*, excepto que el sistema no inicia ninguna otra actividad dentro de la tarea que guarda la instancia. La actividad siempre es el único miembro de su tarea; cualquier actividad iniciada por esta se abre en una tarea aparte.

6. UTILIZAR *FLAGS* EN LOS *INTENT*

Cuando se inicia una actividad, se puede modificar la asociación por defecto entre una actividad y su tarea mediante la inclusión de *flags* en el *intent* que quieras pasarle al método *startActivity()*. Los *flags* que se pueden utilizar para modificar el comportamiento por defecto son:

FLAG_ACTIVITY_NEW_TASK
Inicia la actividad en una nueva tarea. Si una tarea ya se está ejecutando para la actividad que se está iniciando, la tarea se trae a un primer plano con su último estado reanudado y la actividad recibe el nuevo intent a través del método *onNewIntent()*. Esto consigue el mismo comportamiento que el valor del *launchMode* de *"singleTask"*, del que hemos hablado en la sección anterior.

FLAG_ACTIVITY_SINGLE_TOP
Si la actividad que se inicia es la actividad en curso (la que está en la parte superior del *back stack*), entonces la instancia existente recibe una llamada al método *onNewIntent()*, en vez de crear una nueva instancia de la actividad. Esto consigue el mismo comportamiento que el valor del *launchMode* de *"singleTask"*, del que hemos hablado en la sección anterior.

FLAG_ACTIVITY_CLEAR_TOP
Si la actividad que se inicia ya se está ejecutando en la tarea en curso, en vez de iniciar una nueva instancia de la actividad, todas las actividades encima suya son destruidas y este *intent* es entregado a la instancia reanudada de la actividad (que ahora está en la parte superior) a través del método *onNewIntent()*).

No hay un valor del atributo *launchMode* que consigue este comportamiento.

FLAG_ACTIVITY_CLEAR_TOP se utiliza normalmente junto con *FLAG_ACTIVITY_NEW_TASK*. Cuando se usan juntos, estos *flags* son una manera de colocar una actividad ya existente dentro de otra tarea en una posición donde puede responder al *intent*.

7. LIMPIAR EL *BACKSTACK*

Si el usuario abandona una tarea por mucho tiempo, el sistema limpia todas las actividades de la tarea, excepto la actividad raiz. Cuando el usuario retorna a la tarea, solo se restaura la actividad raiz. El sistema se comporta de esta manera, porque después de un tiempo determinado, los usuarios normalmente han abandonado lo que estaban haciendo y vuelven a la tarea para iniciar algo nuevo.

Existen algunos atributos de las actividades que se pueden utilizar para modificar este comportamiento:

alwaysRetainTaskState
Si en la actividad raiz de la tarea este atributo fue establecido en "true" , el comportamiento por defecto ahora descrito no ocurre. La tarea mantiene todas las actividades en su pila por un largo periodo de tiempo.

clearTaskOnLaunch
Si en la actividad raiz de la tarea este atributo fue establecido en "true", la pila elimina todo excepto la actividad raiz cada vez que el usuario abandona la tarea y retorna a ella. Dicho de otra manera, ocurre lo contrario que en *alwaysRetainTaskState*. El usuario siempre retorna a la tarea en su estado inicial, aunque haya abandonado la tarea solo por un instante.

finishOnTaskLaunch
Este atributo es como *clearTaskOnLaunch*, pero trabaja en una sola actividad, no en una tarea completa. También puede hacer que una actividad se vaya, incluyendo la actividad raiz. Cuando fue establecida en "true", la actividad forma parte de la tarea sólo durante la sesión en curso. Si

el usuario se va y vuelve a la tarea, ya no existirá.

8. REFERENCIAS

[1] Android Tasks and Activity backstack! - http://mobisynth.wordpress.com/2009/08/10/android-tasks-and-activity-backstack/.

[2] Tasks and Back Stack | Android - http://developer.android.com/guide/components/tasks-and-back-stack.html.

[3] Manipulating Android Tasks and Back stack - http://www.slideshare.net/RanNachmany/manipulating-android-tasks-and-back-stack.

[4] Navigation with Back and Up - http://stuff.mit.edu:8001/afs/sipb/project/android/docs/design/patterns/navigation.html.

[5] <activity> - http://stuff.mit.edu:8001/afs/sipb/project/android/docs/guide/topics/manifest/activity-element.html.