

## Introducción a las Tareas y Back Stack

¡Felicitaciones por llegar hasta aquí! Vas avanzando muy bien. A continuación te presentamos una breve **introducción a las tareas en Android y el Back Stack**.

**Una tarea es una colección de actividades** con las que los usuarios interactúan cuando realizan un determinado trabajo. Las actividades se organizan en una pila (back stack), en el orden en que se abre cada actividad. Por ejemplo, una aplicación de correo electrónico podría tener una actividad para mostrar una lista de mensajes nuevos. Cuando el usuario selecciona un mensaje, se abre una nueva actividad para ver ese mensaje. Esta nueva actividad se agrega al back stack. Si el usuario presiona el botón Atrás, esa nueva actividad finaliza y es removida de la pila.

Cuando las aplicaciones se ejecutan simultáneamente en un entorno de múltiples ventanas, compatible con Android 7.0 (API nivel 24) y superior, **el sistema administra las tareas por separado para cada ventana y cada una puede tener múltiples tareas. Lo mismo ocurre con las aplicaciones de Android** que se ejecutan en Chromebooks: el sistema gestiona las tareas, o grupos de tareas, por ventana.

**La interfaz de inicio del dispositivo es el punto de partida para la mayoría de las tareas.** Cuando el usuario hace clic en un ícono en el launcher (iniciador de aplicaciones), o en un acceso directo en la interfaz de inicio, la tarea de esa aplicación pasa a primer plano. Si no existe una tarea para la aplicación (la aplicación no se ha utilizado recientemente), se crea una nueva tarea y la actividad "principal" (Main activity) para dicha aplicación, se abre como la actividad raíz en la pila.

**Cuando la actividad actual comienza otra, la nueva actividad, se coloca en la parte superior de la pila y se enfoca.** La actividad anterior permanece en la pila, pero se detiene. Cuando se detiene una actividad, el sistema retiene el estado actual de su interfaz de usuario. Cuando el usuario presiona el botón Atrás, la actividad actual se abre desde la parte superior de la pila (la actividad se destruye) y se reanuda la actividad anterior (se restaura el estado anterior de su IU). Las actividades en la pila nunca se reorganizan, sólo se empujan y se sacan de la pila. Se empujan a la pila, cuando la actividad actual las inicia y se quitan cuando el usuario la abandona usando el botón

Atrás. Como tal, el back stack funciona como una estructura de tipo LIFO (last in, first out) "último en entrar, primero en salir". La Figura 1 visualiza este comportamiento con una línea de tiempo que muestra el progreso entre las actividades junto con el back stack actual en cada momento.

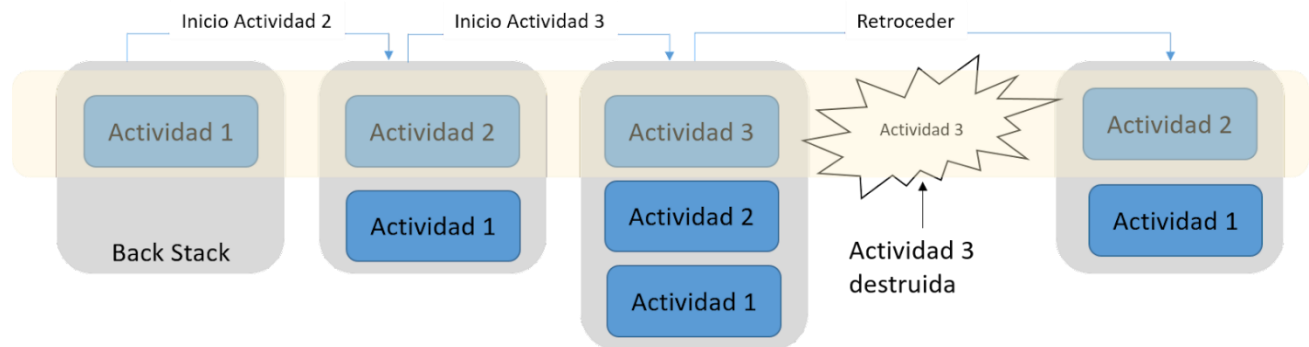


Figura 1. Representación de actividades y tareas en el back stack.

**Si el usuario continúa presionando el botón de Back (Atrás), cada actividad en la pila se abre para revelar la anterior, hasta que el usuario regrese a la pantalla de Inicio (o a la actividad que se estaba ejecutando cuando comenzó la tarea). Cuando todas las actividades se eliminan de la pila, la tarea ya no existe.**

**Una tarea es una unidad cohesiva que puede moverse al "fondo"** cuando los usuarios comienzan una nueva tarea o van a la pantalla de Inicio, a través del botón de Inicio. **Mientras está en segundo plano**, todas las actividades de la tarea se detienen, pero la pila de tareas para la tarea permanece intacta: la tarea simplemente ha perdido el foco mientras se lleva a cabo otra tarea, como se muestra en la figura 2.

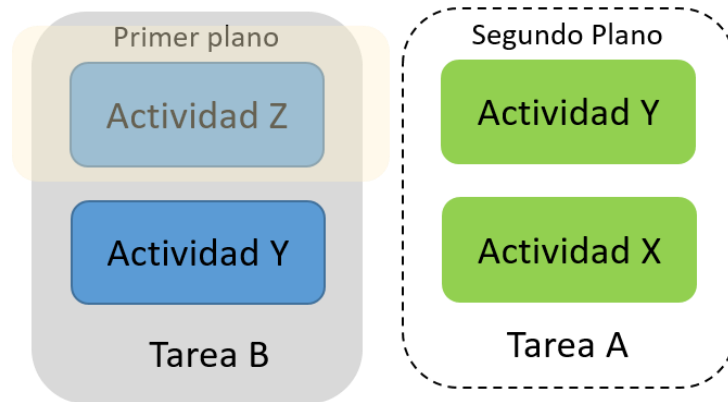


Figura 2. Dos tareas: la Tarea B recibe la interacción del usuario en primer plano, mientras que la Tarea A está en segundo plano, esperando que se reanude.

**Una tarea puede volver al " primer plano " para que los usuarios puedan continuar donde lo dejaron.** Supongamos, por ejemplo, que la tarea actual (Tarea A) tiene tres actividades en su pila, dos bajo la actividad actual. El usuario presiona el botón Inicio, luego inicia una nueva aplicación desde el launcher. Cuando aparece la pantalla de inicio, la tarea A pasa a 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 regresa a Inicio nuevamente y selecciona la aplicación que originalmente inició la Tarea A. Ahora, la Tarea A pasa a primer plano: las tres actividades en su pila están intactas y se reanuda la actividad en la parte superior de la pila. En este punto, el usuario también puede volver a la Tarea B yendo a Inicio y seleccionando el ícono de la aplicación que inició esa tarea (o seleccionando la tarea de la aplicación en la pantalla Recientes). Este es un ejemplo de multitarea en Android.

Debido a que las actividades en el back stack nunca se reorganizan, **si nuestra aplicación permite a los usuarios iniciar una actividad particular desde más de una actividad**, se crea una nueva instancia de esa actividad y se empuja a la pila (en lugar de traer cualquier instancia previa de la actividad al tope). Como tal, una actividad en su aplicación se puede instanciar varias veces (incluso de diferentes tareas), como se muestra en la figura 3. Como tal, si el usuario navega hacia atrás con el botón Atrás, cada instancia de la actividad se revela en el orden en que fueron abiertos (cada uno con

su propio estado de interfaz de usuario). Sin embargo, se puede modificar este comportamiento si no se desea que una actividad se instancie más de una vez.

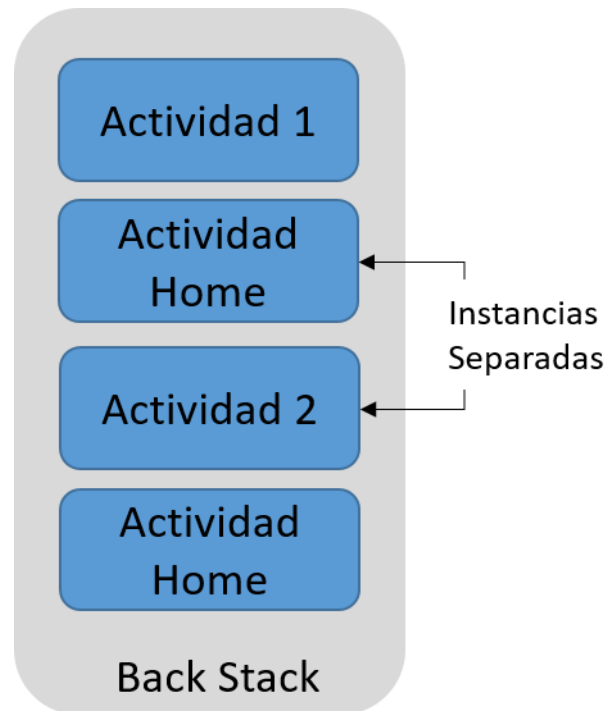


Figura 3. Una sola actividad se instancia varias veces.

Para resumir el comportamiento predeterminado para actividades y tareas:

- **Cuando la Actividad A inicia la Actividad B**, la Actividad A se detiene, pero el sistema conserva su estado (como la posición de desplazamiento y el texto ingresado en los formularios). Si el usuario presiona el botón Atrás mientras está en la Actividad B, la Actividad A se reanuda con su estado restaurado.
- **Cuando el usuario deja una tarea presionando el botón Inicio**, la actividad actual se detiene y su tarea pasa a un segundo plano. El sistema retiene el estado de cada actividad en la tarea. Si el usuario luego reanuda la tarea seleccionando el ícono del iniciador que comenzó la tarea, la misma pasa a primer plano y reanuda la actividad en la parte superior de la pila.

- **Si el usuario presiona el botón Atrás**, la actividad actual se extrae de la pila y se destruye. Se reanuda la actividad anterior en la pila. Cuando se destruye una actividad, el sistema no retiene el estado de la actividad.

Las actividades se pueden instanciar varias veces, incluso desde otras tareas.

## Administración de las Tareas

La forma en que Android gestiona las tareas y el back stack, funciona muy bien para la mayoría de las aplicaciones y no deberíamos tener que preocuparnos sobre cómo se asocian nuestras actividades con las tareas o cómo existen en el back stack. Sin embargo, en ciertos casos, podemos decidir que queremos interrumpir el comportamiento normal. Quizás deseemos que una actividad en su aplicación comience una nueva tarea cuando se inicie (en lugar de ubicarse dentro de la tarea actual) o cuando comienza una actividad, deseemos presentar una instancia existente de la misma (en lugar de crear una nueva instancia en la parte superior del back stack) o , quizás, deseemos que en el back stack se borren todas las actividades excepto la actividad raíz cuando el usuario abandona la tarea.

## Definiendo los Modos de Lanzamiento (launch modes)

Los modos de inicio nos permiten definir cómo se asocia una nueva instancia de una actividad con la tarea actual. Se pueden definir diferentes modos de inicio de dos maneras:

### Usando el archivo de manifiesto

Cuando declara una actividad en su archivo de manifiesto, puede especificar cómo debe asociarse la actividad cuando se inicia, con las tareas.

### Usando banderas (flags) en el intento

Cuando llama a **startActivity()**, se puede incluir un indicador en el intento que declara como (o si) la nueva actividad debe asociarse con la tarea actual.

Como tal, si la Actividad A inicia la Actividad B, la Actividad B puede definir en su manifiesto cómo debe asociarse con la tarea actual (si es que lo hace) y la Actividad A también puede indicar cómo la Actividad B debe asociarse con la tarea actual. Si ambas actividades definen cómo la Actividad B debe asociarse con una tarea, la solicitud de la Actividad A (como se define en el intent) tendrá prioridad sobre la solicitud de la Actividad B (como se define en el manifiesto).

En este sentido, los principales atributos de **<activity>** que se pueden usar son:

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

Y las principales banderas que se pueden usar en el intent son:

- **FLAG\_ACTIVITY\_NEW\_TASK**
- **FLAG\_ACTIVITY\_CLEAR\_TOP**
- **FLAG\_ACTIVITY\_SINGLE\_TOP**

## Despejando el Back Stack

**Si el usuario deja una tarea durante mucho tiempo, el sistema borra todas las actividades de la tarea, excepto la actividad raíz.** Cuando el usuario vuelve a la tarea nuevamente, sólo se restaura la actividad raíz. El sistema se comporta de esta manera, porque después de un período prolongado de tiempo, los usuarios probablemente hayan abandonado lo que estaban haciendo antes y estén regresando a la tarea para comenzar algo nuevo.

Hay algunos **atributos de actividad** que se pueden usar para modificar este comportamiento:

- **alwaysRetainTaskState**
- **clearTaskOnLaunch**
- **finishOnTaskLaunch**

Gracias por permanecer con nosotros y disfrutar este espacio a través de las tareas en Android.