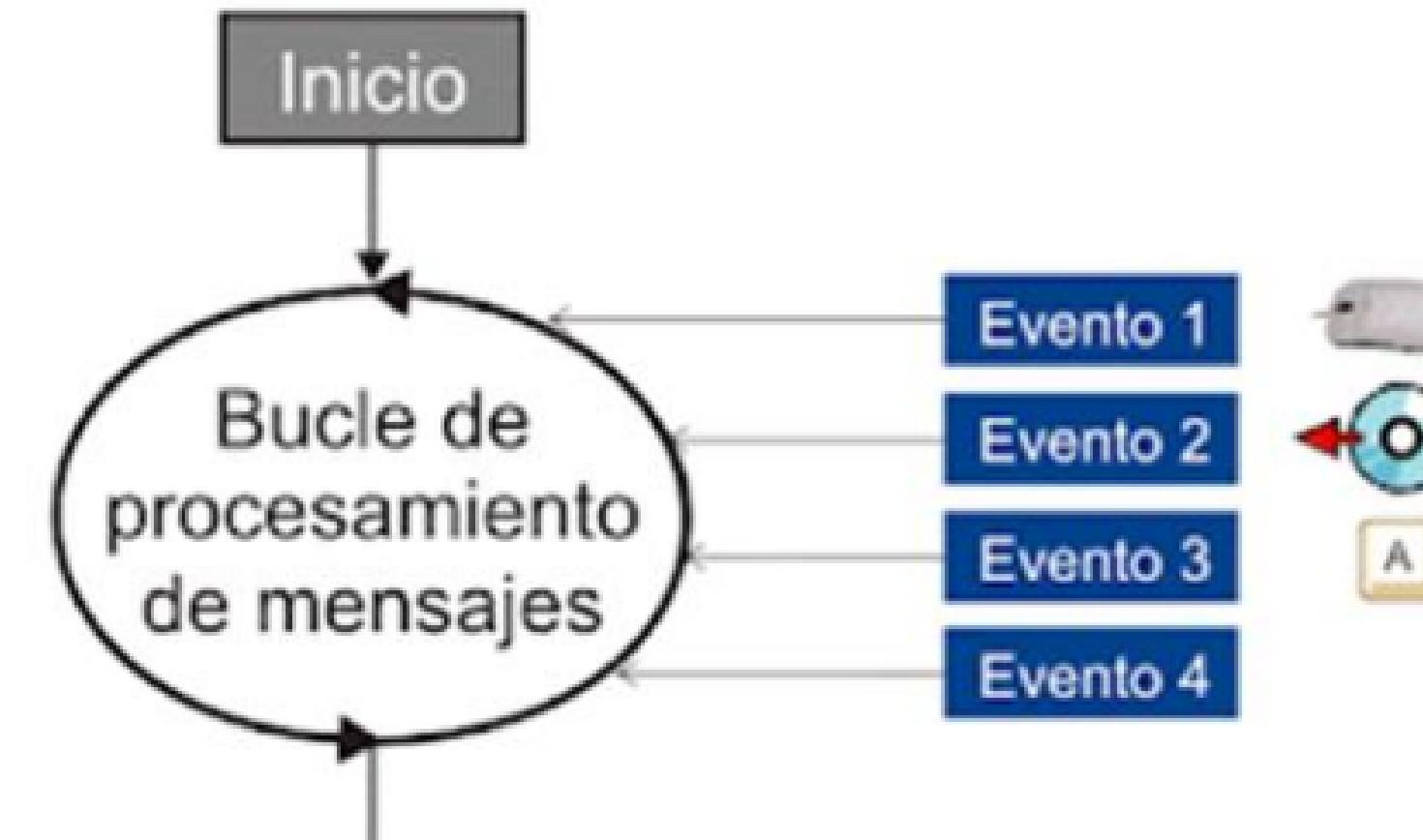


# **CICLO DE VIDA DE UNA APLICACION**

## Programación Orientada a Eventos

La programación dirigida por eventos es un paradigma de programación en el que tanto la estructura como la ejecución de los programas van determinados por los sucesos que ocurrán en el sistema, definidos por el usuario o que ellos mismos provoquen.

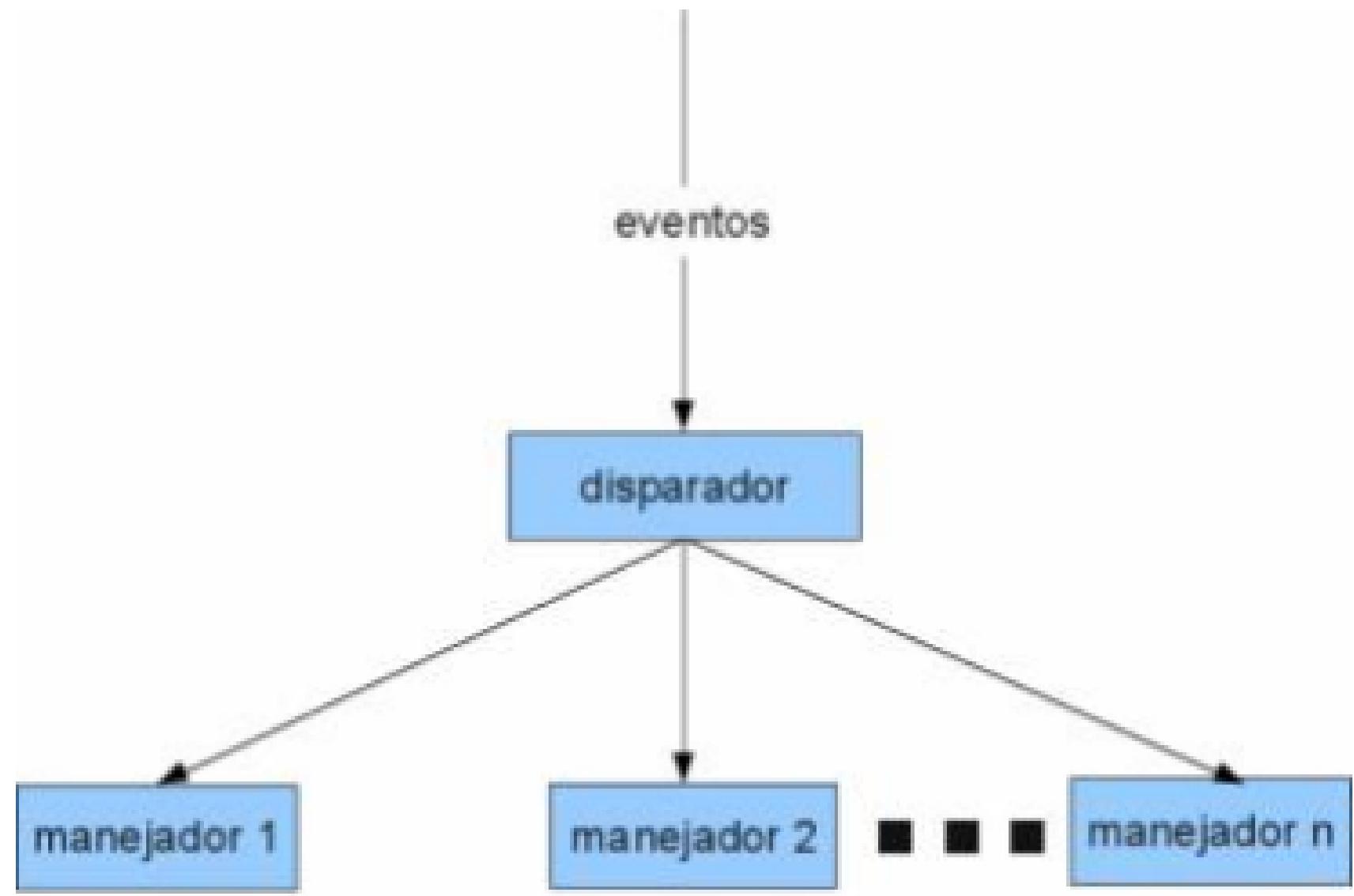
Las aplicaciones desarrolladas con programación dirigida por eventos implementan un bucle principal o main loop donde se ejecutan las dos secciones principales de la aplicación: El selector de eventos y el manejador de eventos.



Un flujo de datos invoca eventos.

Un disparador los envía a manejadores especializados, que se encargan de realizar operaciones sobre el flujo de datos

El trabajo del disparador es analizar los eventos y entonces enviarlos al manejador adecuado. El disparador tiene que procesar un flujo de eventos, así que su lógica debe incluir un bucle de eventos para poder enviar un evento a un manejador y volver a escuchar a la espera de nuevos eventos que disparar.



El creador de un programa dirigido por eventos debe definirlos en eventos que manejarán su programa y las acciones que se realizarán al producirse cada uno de ellos, lo que se conoce como el administrador de evento.

En la programación dirigida por eventos, al comenzarla ejecución del programa se llevarán a cabo las inicializaciones y demás código inicial y a continuación el programa quedará bloqueado hasta que se produzca algún evento.



## · El Administrador de Actividades

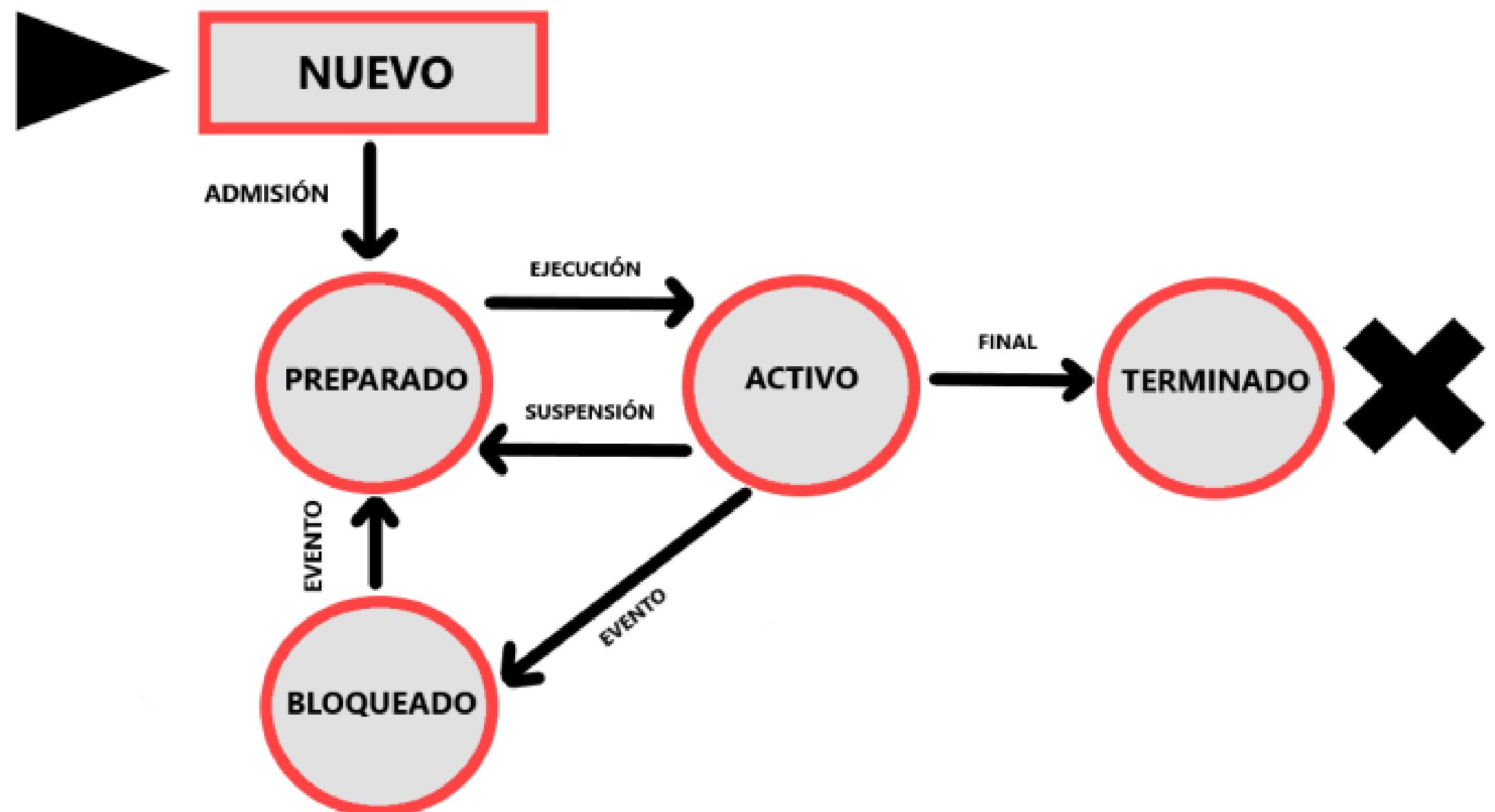
Cada vez que se inicia una actividad, el sistema operativo emplea una cantidad considerable de recursos.

Para empezar, se necesita crear un nuevo proceso, a continuación, asignar memoria a los diferentes objetos que deben aparecer en la interfaz gráfica y, por último, se representará en la pantalla. Lógicamente, todo este trabajo no puede desecharse sólo porque el usuario cambie de pantalla. Para evitarlo, se diseñó el Administrador de Actividades, un componente del sistema operativo diseñado para gestionar las diferentes actividades de las aplicaciones que se ejecuten en Android.

En Android, la aplicación creará sus actividades, pero será el sistema operativo quien decida cuándo terminan (y se liberan sus recursos).

# Estados del ciclo de vida

En general un proceso pasa por estos estados:



## Eventos que ocurren en las transiciones

<b>OnCreate()</b>	Sólo se ejecuta una vez en el ciclo de vida de la actividad, cuando se llama a la actividad por primera vez.
<b>onStart()</b>	Se ejecuta al mostrar la actividad en la pantalla. Siempre se ejecuta después de <code>onCreate()</code> o después de <code>onRestart()</code> , si la actividad ha estado parada.
<b>onResume()</b>	Se ejecuta cuando la actividad termina de cargarse y el usuario comienza a interactuar con ella.
<b>onPause()</b>	Cuando el sistema arranca una nueva actividad que va a centrar su atención, se ejecuta <code>onPause()</code> en la actividad actual.

## Eventos que ocurren en las transiciones

### onStop()

Se ejecuta cuando la actividad ya es invisible para el usuario porque otra actividad ha ocupado su lugar. Después de ejecutar este método pueden suceder tres cosas:

- Que la actividad vuelva a primer plano. Se ejecutará *onRestart()*.
- Que el sistema elimine la actividad para liberar memoria.
- Terminar la aplicación. Se ejecutará *onDestroy()*

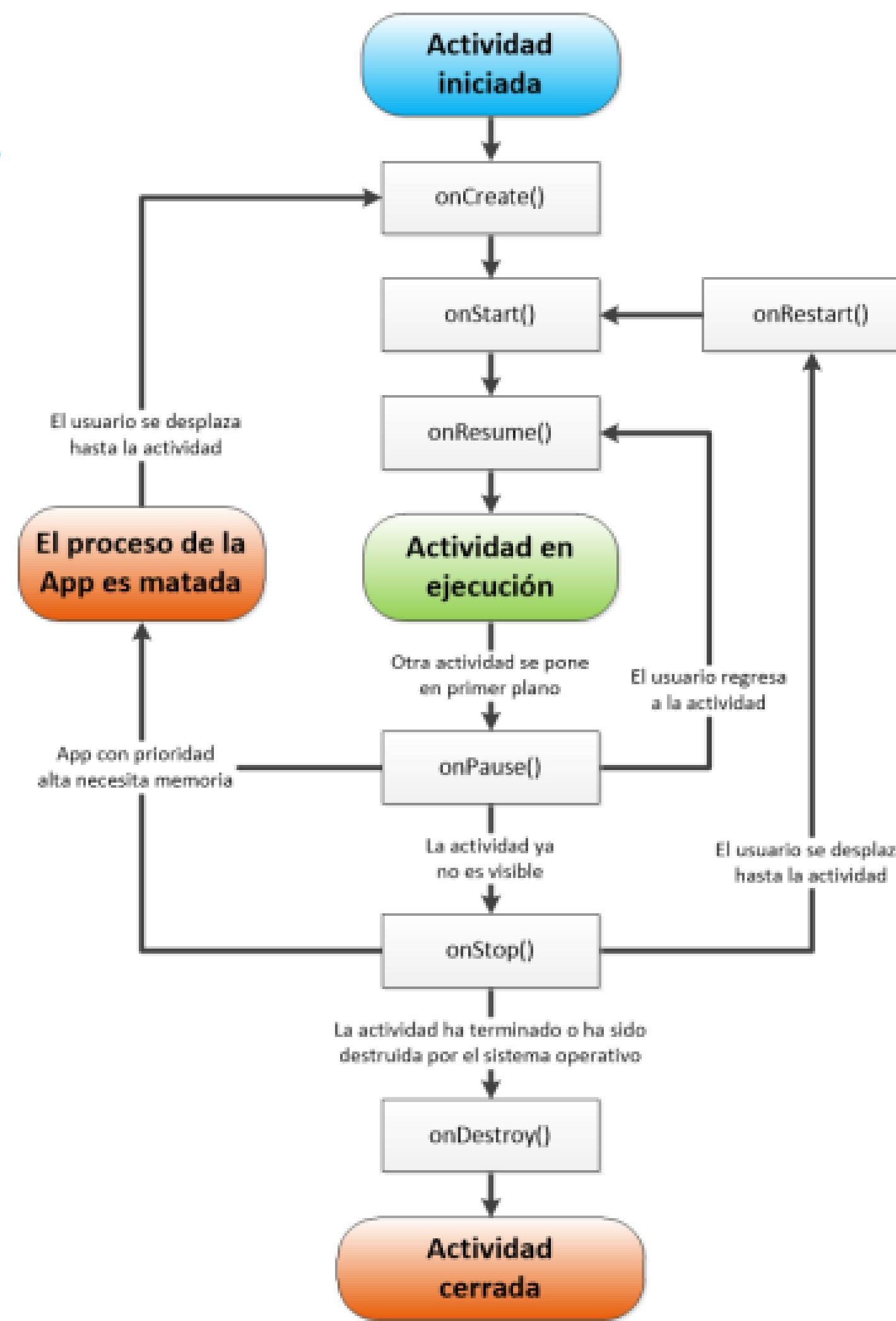
### onRestart()

Se ejecuta cuando vuelve a usarse la actividad después de haber estado parada. Se ejecutará entre un *onStop()* y un *onStart()*

### onDestroy()

Se ejecuta al final del ciclo de vida de la actividad. Después, se liberan todos los recursos que utiliza. Para volver a ejecutarla debería iniciarse un nuevo ciclo de vida. Puede ejecutarlo el sistema justo antes de eliminar la actividad, o hacerlo de forma explícita el propio usuario.

## Estados del ciclo de vida y eventos que ocurren en las transiciones



## Estados del ciclo de vida y eventos que ocurren en las transiciones

