

Ciclo de Vida

Ciclo de vida

A lo largo de una ejecución normal de una aplicación, sus Activities pueden estar en alguno de los siguientes estados:

- . **Activa**

Cuando el usuario ve la actividad y puede interactuar con ella desde la pantalla.
Cuando está la primera en la **pila de ejecución**.

- . **Pausada**

Cuando pasa a segundo plano pero aun está visible.
La actividad pausada pierde el foco de ejecución pero sigue visible.
Puede ser matada por el sistema si se necesita memoria.

- . **Parada**

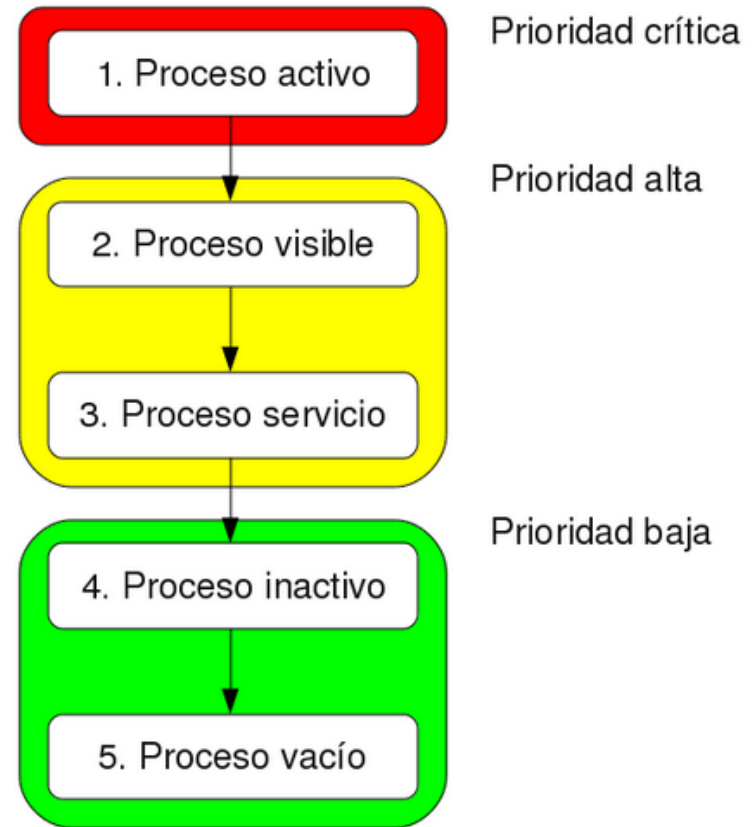
Cuando pasa a segundo plano y esta totalmente tapada por otra actividad.
Tambien puede ser matada por el sistema

- . **Destruida**

La actividad ya no está disponible, se han liberado todos sus recursos.

Ciclo de vida

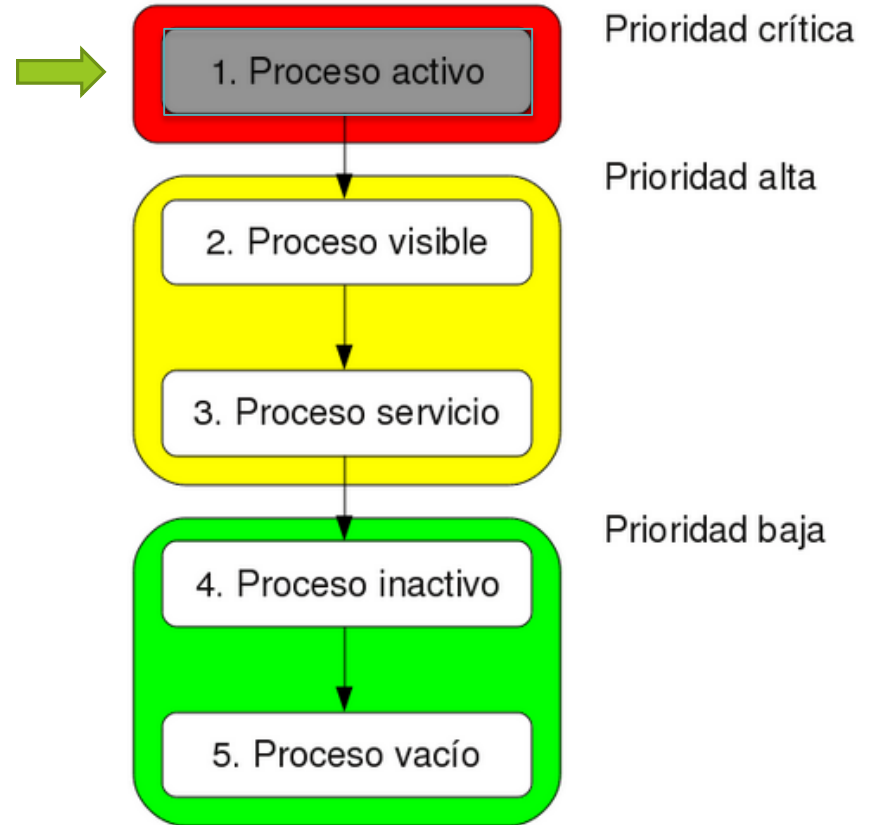
- Las aplicaciones Android no tienen casi control sobre su ciclo de vida.
- Android decidirá cuando una aplicación o actividad cambia de estado o muere
- Las aplicaciones tienen que estar atentas a estos cambios y preparadas para actuar en cada momento.
- Android intenta que el usuario siempre sea atendido por lo que la aplicación activa tendrá mas prioridad, si necesita recursos matará otras actividades no prioritarias.
- El orden de detención y eliminación de los procesos viene definido por la prioridad de los mismos.



Ciclo de vida

Procesos Activos

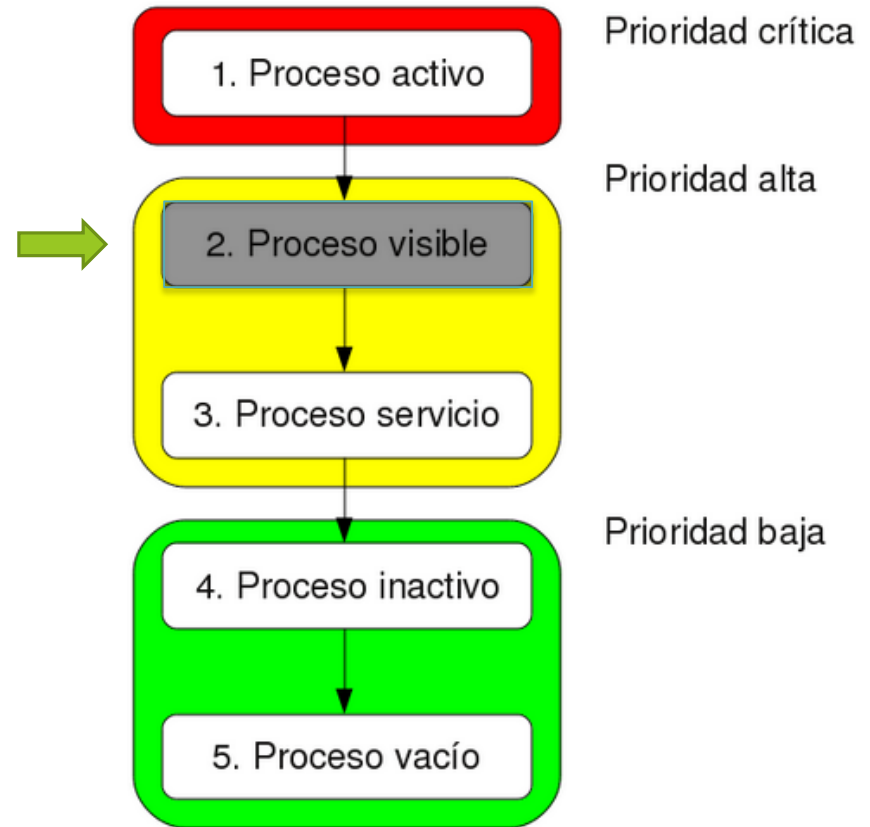
- Procesos que contienen aplicaciones que están interactuando con el usuario.
- Android intenta que su latencia sea mínima liberando recursos.
- Sólo son detenidos como último recurso.



Ciclo de vida

Procesos Visibles

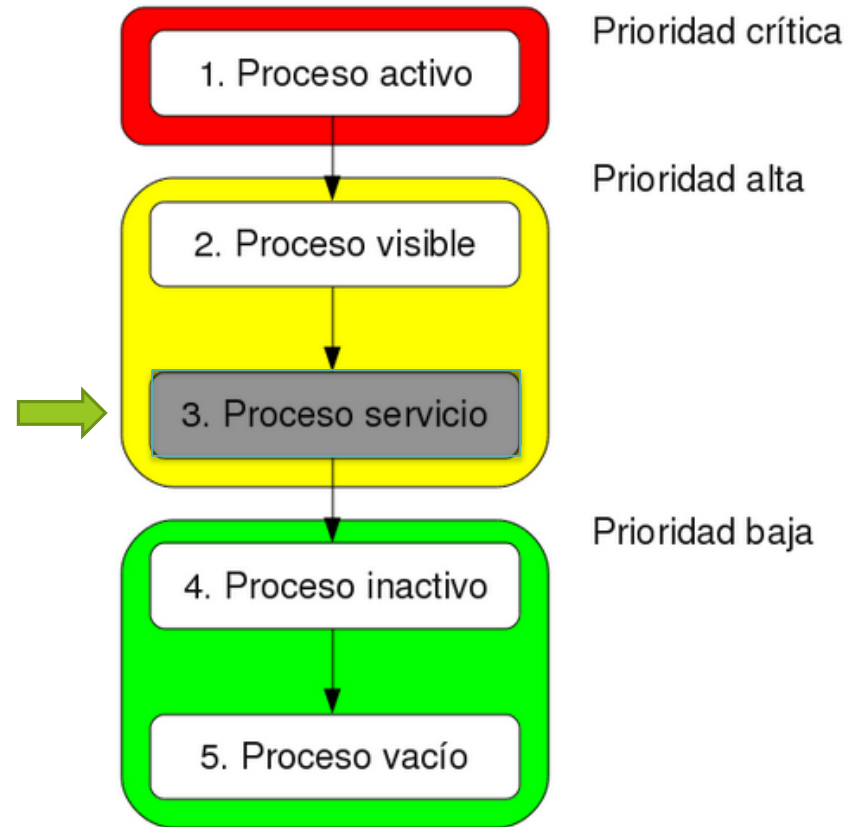
- Procesos visibles pero inactivos.
- Porque sus correspondientes aplicaciones se están mostrando detrás de otras o porque no están respondiendo a ninguna entrada del usuario.
- Esto sucede cuando una Actividad se encuentra parcialmente oculta por otra actividad (transparente o que no ocupa toda la pantalla).
- Estos procesos son detenidos tan solo bajo condiciones extremas.



Ciclo de vida

Procesos asociados a servicios en ejecución

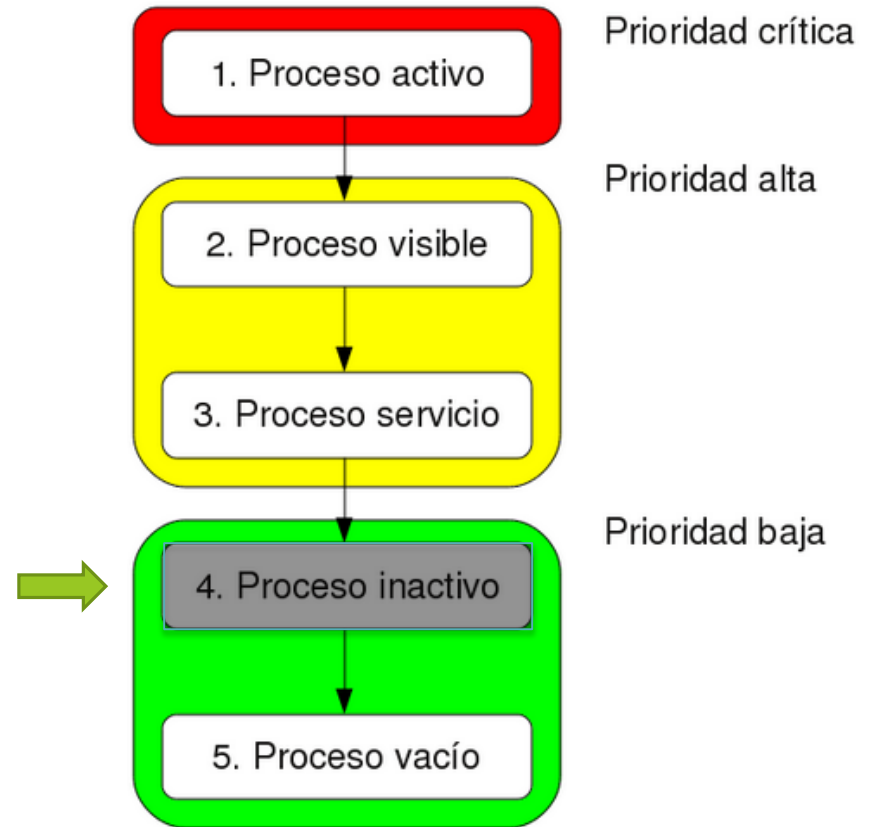
- Procesos Activos pero no visibles
- Los servicios permiten que exista procesamiento sin necesidad de que exista una interfaz de usuario visible.
- Debido a que estos servicios no interactúan directamente con el usuario, reciben una prioridad ligeramente inferior a la de los procesos visibles.
- Sin embargo se siguen considerando procesos activos y no serán detenidos a menos que sea estrictamente necesario.



Ciclo de vida

Procesos inactivos

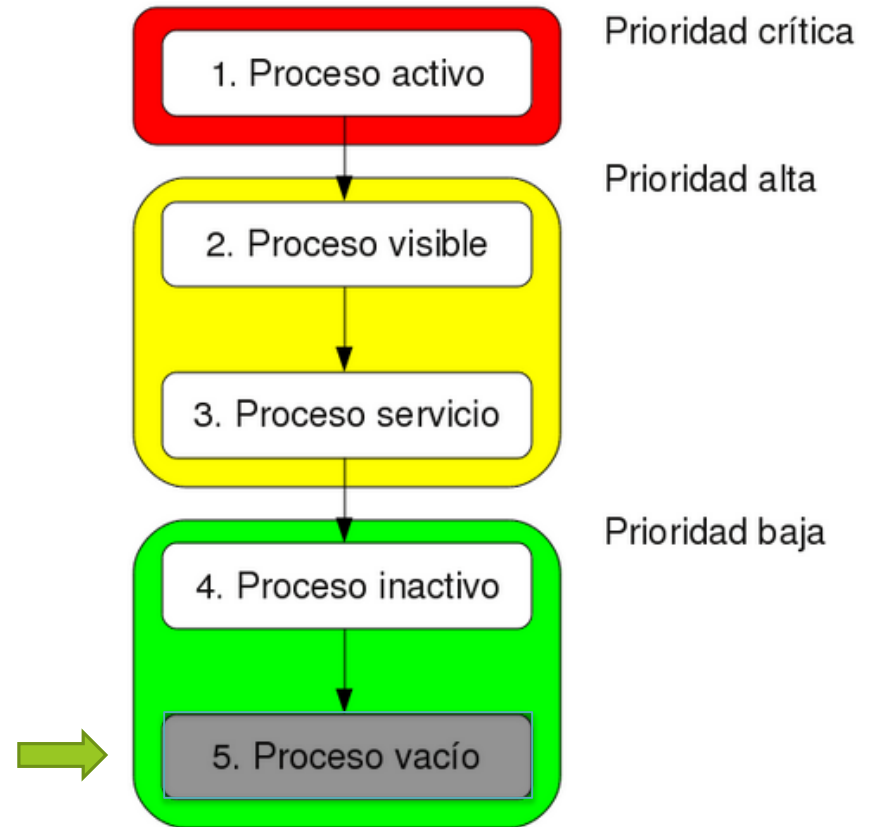
- Procesos que albergan actividades que ni son visibles ni se encuentran realizando un procesamiento, ni ejecutando ningún servicio.
- El orden en el que se detendrán estos procesos vendrá determinado por el tiempo que éstos llevan inactivos desde la última vez que fueron visibles.



Ciclo de vida

Procesos vacíos

- Son el resultado del intento de Android de retener aplicaciones en memoria una vez que éstas han terminado.
- Con esto consigue lanzar de nuevo la aplicación en menos tiempo.



Ciclo de vida

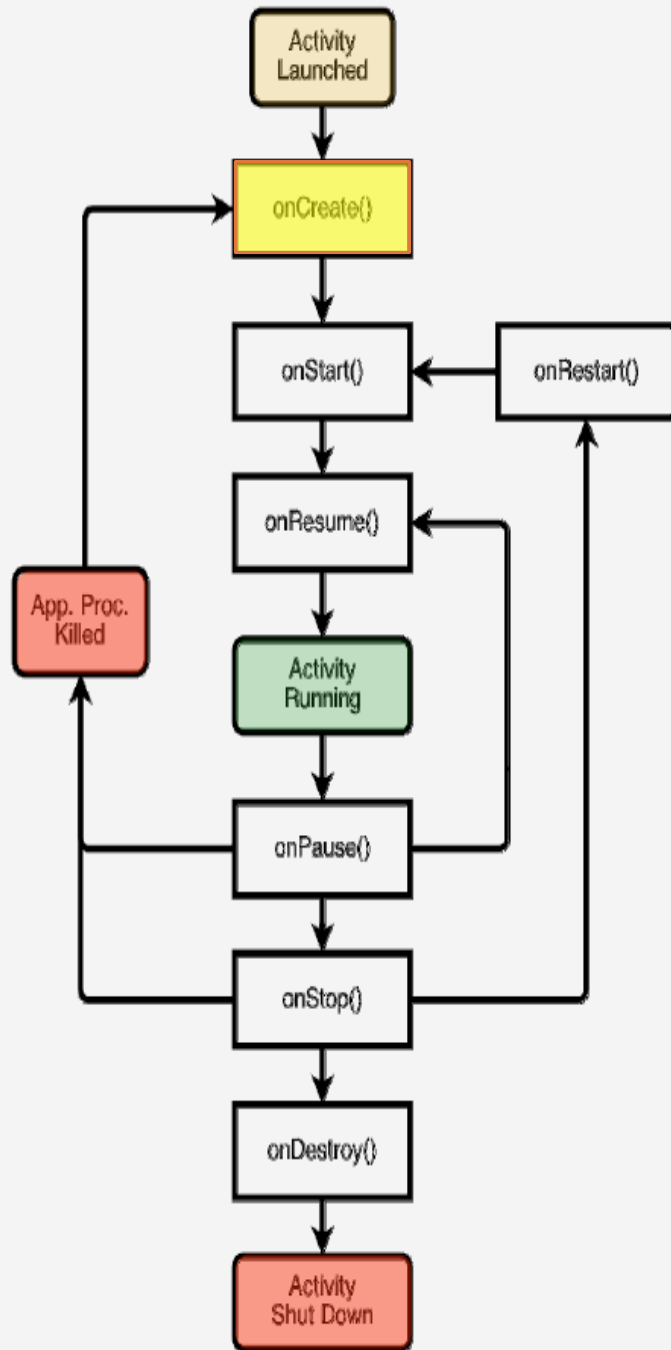
[Actividades Android](#) – [Ciclo de vida](#)

onCreate(Bundle):

Al crear la Activity, Android llama a este callback de nuestra Activity.

Se utiliza para realizar todo tipo de inicializaciones, como la creación de la interfaz de usuario o la inicialización de estructuras de datos.

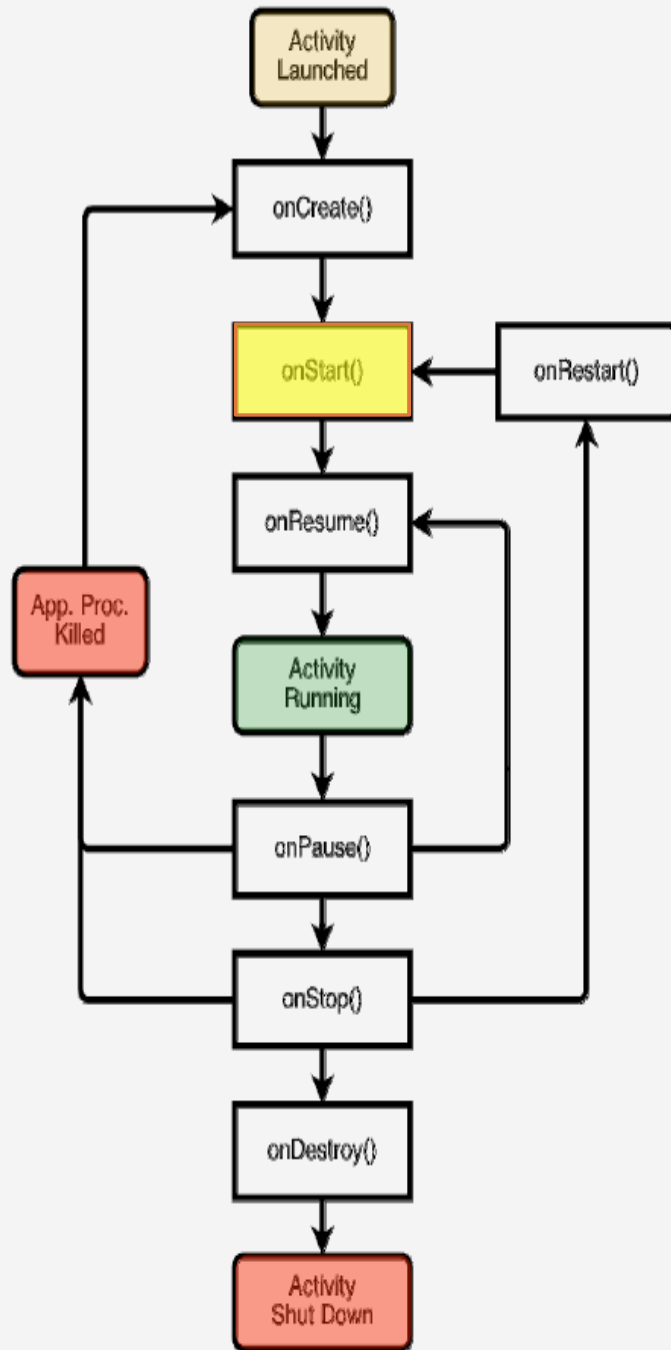
Puede recibir como parámetro, información de estado de la instancia (en una instancia de la clase **Bundle**), en caso de haber sido reanudada desde una actividad que fue destruida y vuelta a crear.

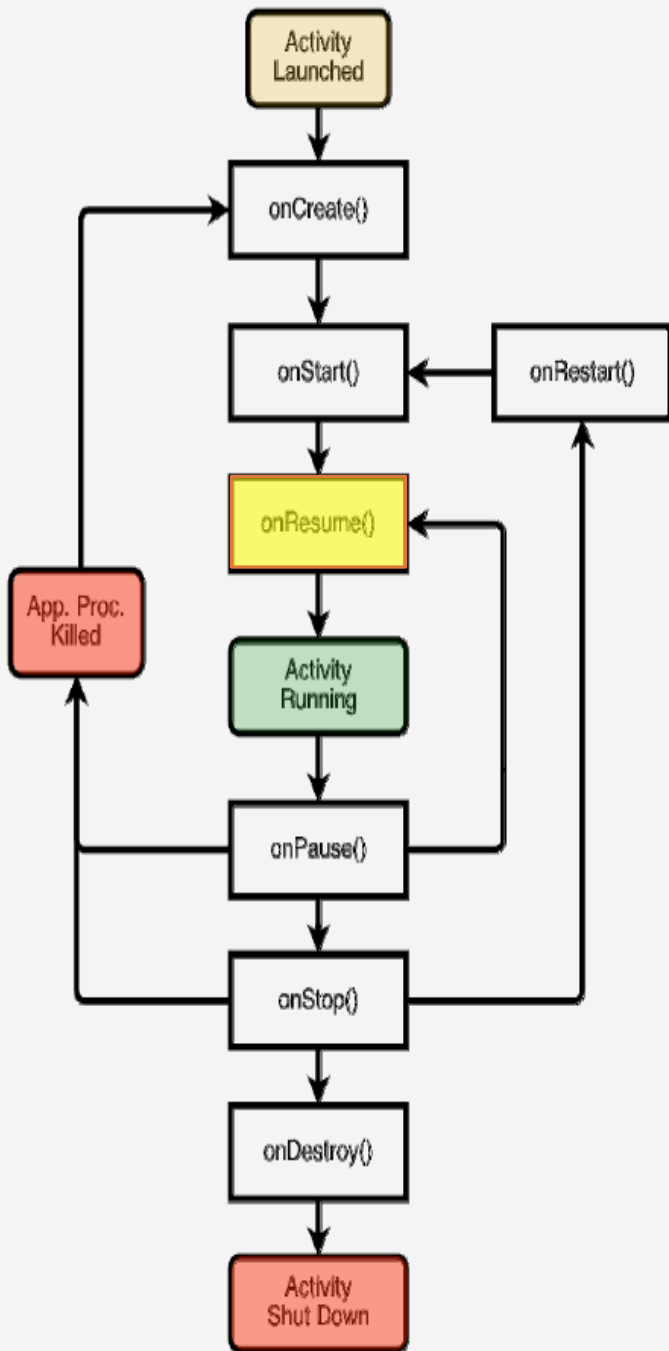


Ciclo de vida

onStart():

Este callback es llamado por Android la actividad está a punto de ser mostrada al usuario.





Ciclo de vida

onResume():

Es llamado por Android cuando la actividad va a comenzar a interactuar con el usuario.

En este punto:

- La Activity ha alcanzado la cima de la Back Stack
- Tiene redireccionada la entrada de usuario.

Es un buen lugar para lanzar las animaciones y la música.

Ciclo de vida

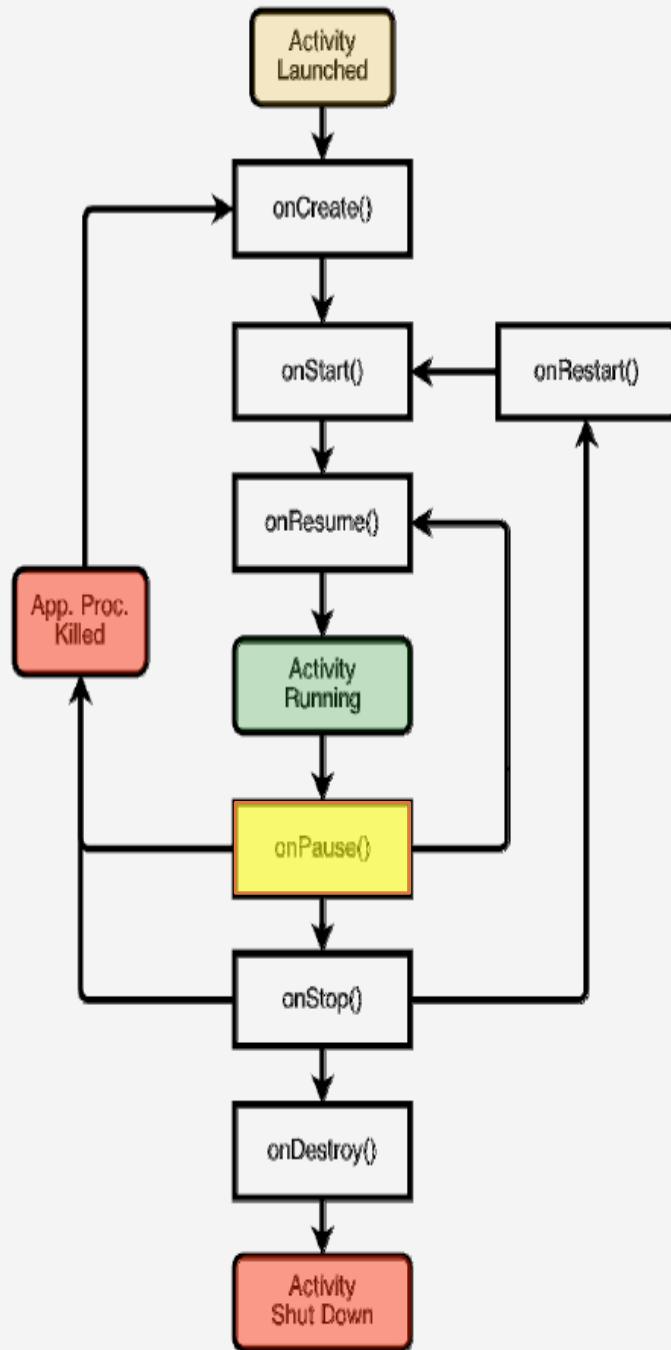
onPause():

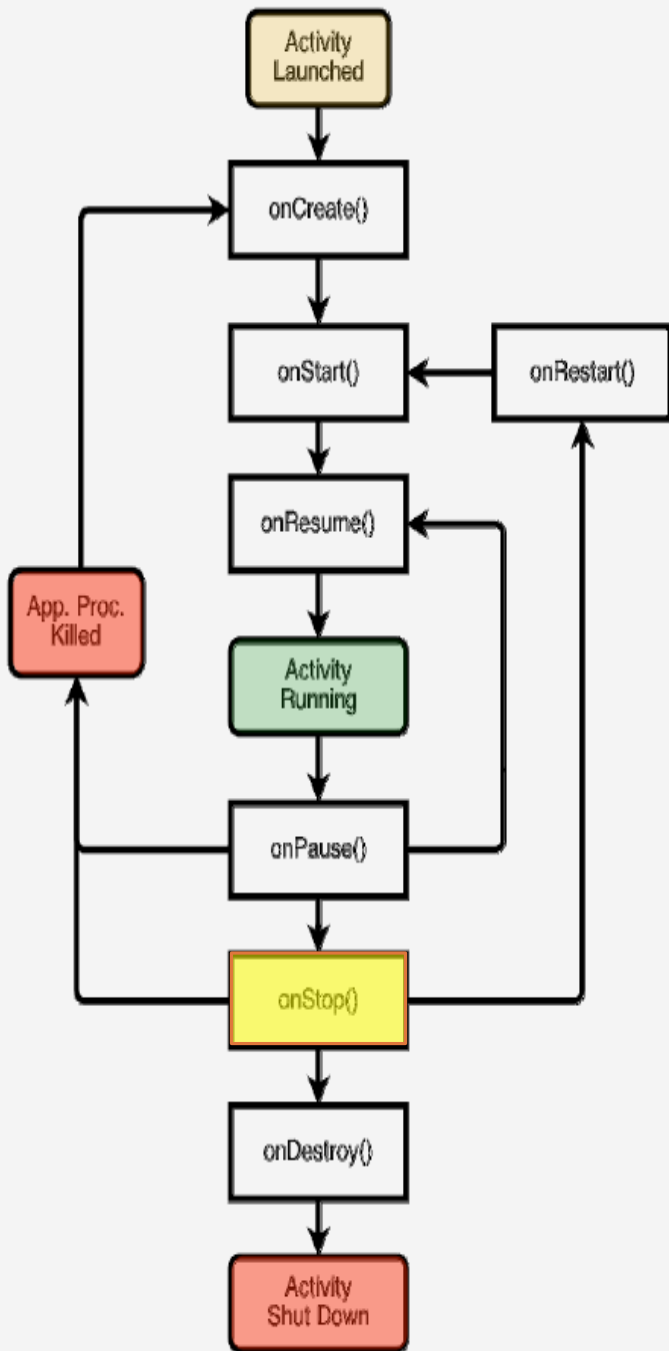
Es llamado por Android cuando la actividad está a punto de ser lanzada a segundo plano, normalmente porque otra aplicación es lanzada.

Es el lugar adecuado para:

- Detener animaciones, música
- Almacenar los datos que estaban en edición.

No debe demorarse su ejecución pues la siguiente Activity no va a continuar hasta que termine ésta.





Ciclo de vida

onStop():

Es llamado por Android cuando la actividad ya no va a ser visible para el usuario.

Puede ser porque la Activity es destruida por Android, o porque otra actividad ha sido “resumed” y la cubre totalmente.

Ojo si hay muy poca memoria, **es posible que la actividad se destruya sin llamar a este método**, por lo que **no es buen lugar para guardar lo que se estaba editando**.

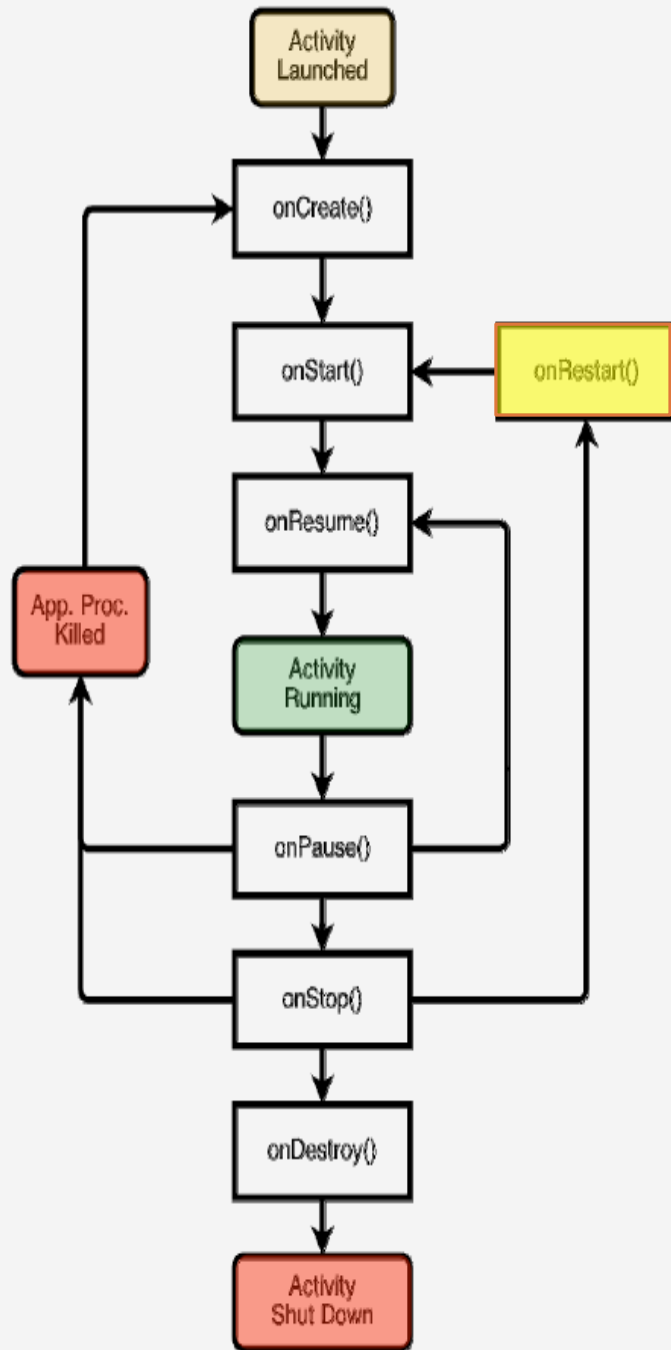
Desde este estado puede pasar que:

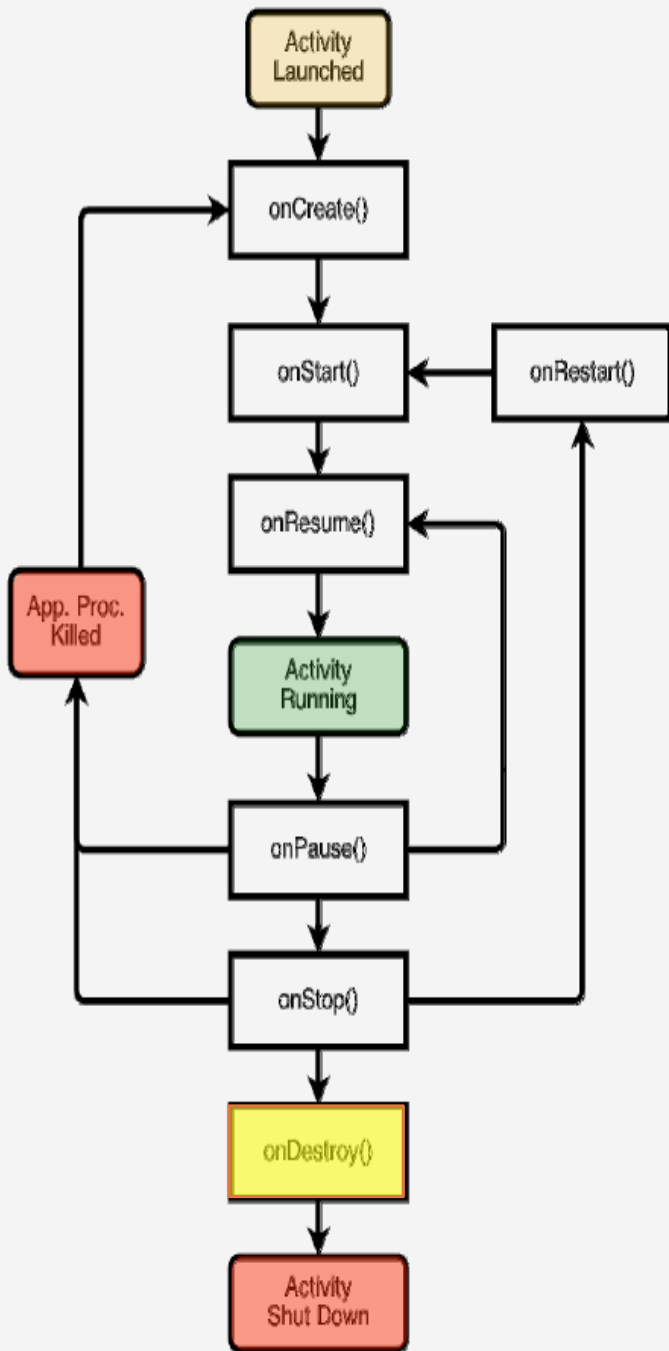
- La actividad termine y desaparezca
- La actividad vuelva a interactuar con el usuario

Ciclo de vida

onRestart():

Es llamado por Android cuando la actividad va a volver a ser representada después de haber pasado por **onStop()**.





Ciclo de vida

`onDestroy()`:

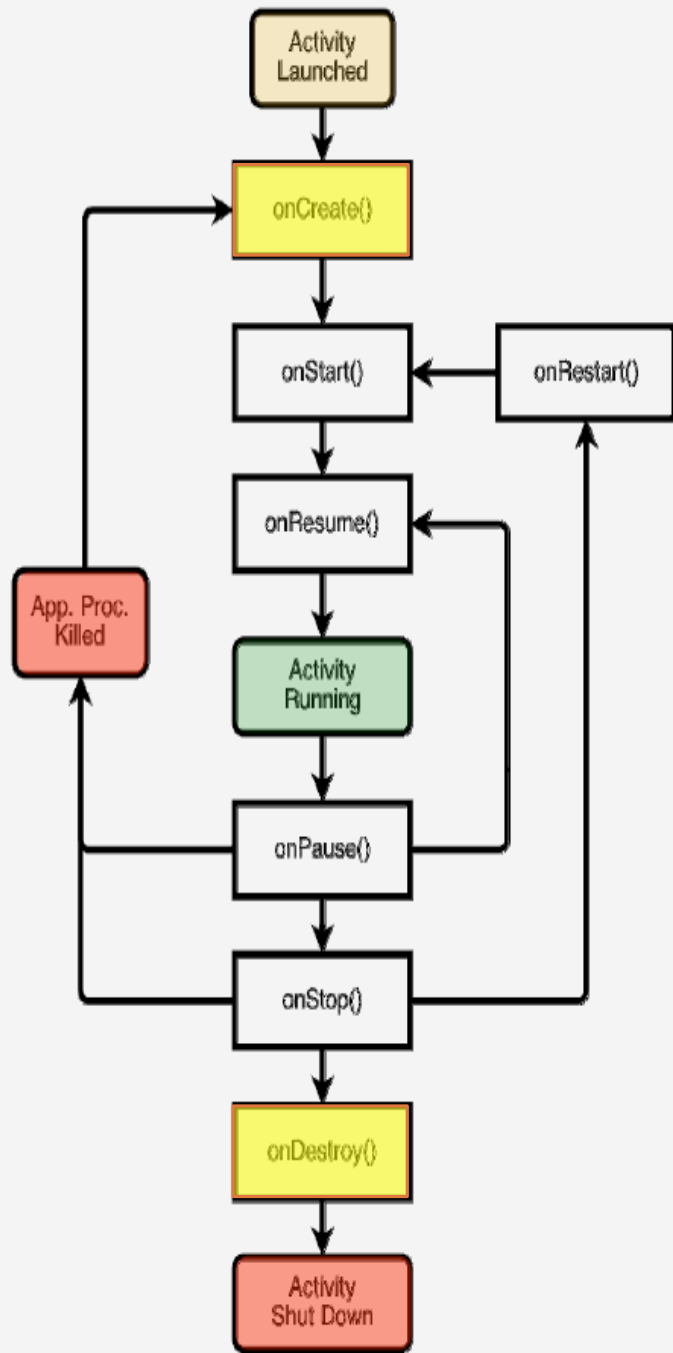
Es llamado por Android antes de que la actividad sea totalmente destruida porque:

- La actividad está terminando (debido a que el usuario la descarta por completo o a que se llama a `finish()`).
- El sistema está finalizando temporalmente la actividad debido a un cambio de configuración (como la rotación del dispositivo o el modo multiventana).

Se puede discernir ambos escenarios llamando a `isFinishing()`, esto lo haremos normalmente en el callback `onPause()`;

Ojo si hay muy poca memoria, es posible que la actividad se destruya sin llamar a este método.

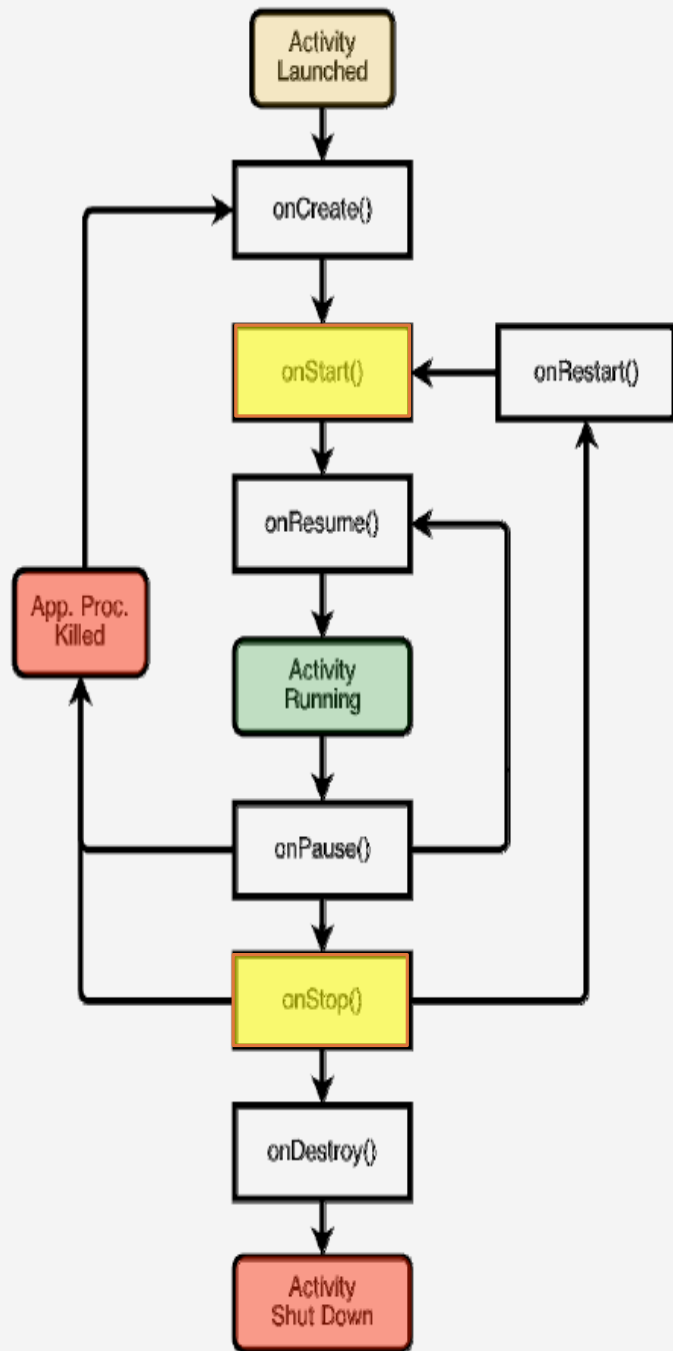
Es el último callback llamado por Android para nuestra activity.



Ciclo de vida

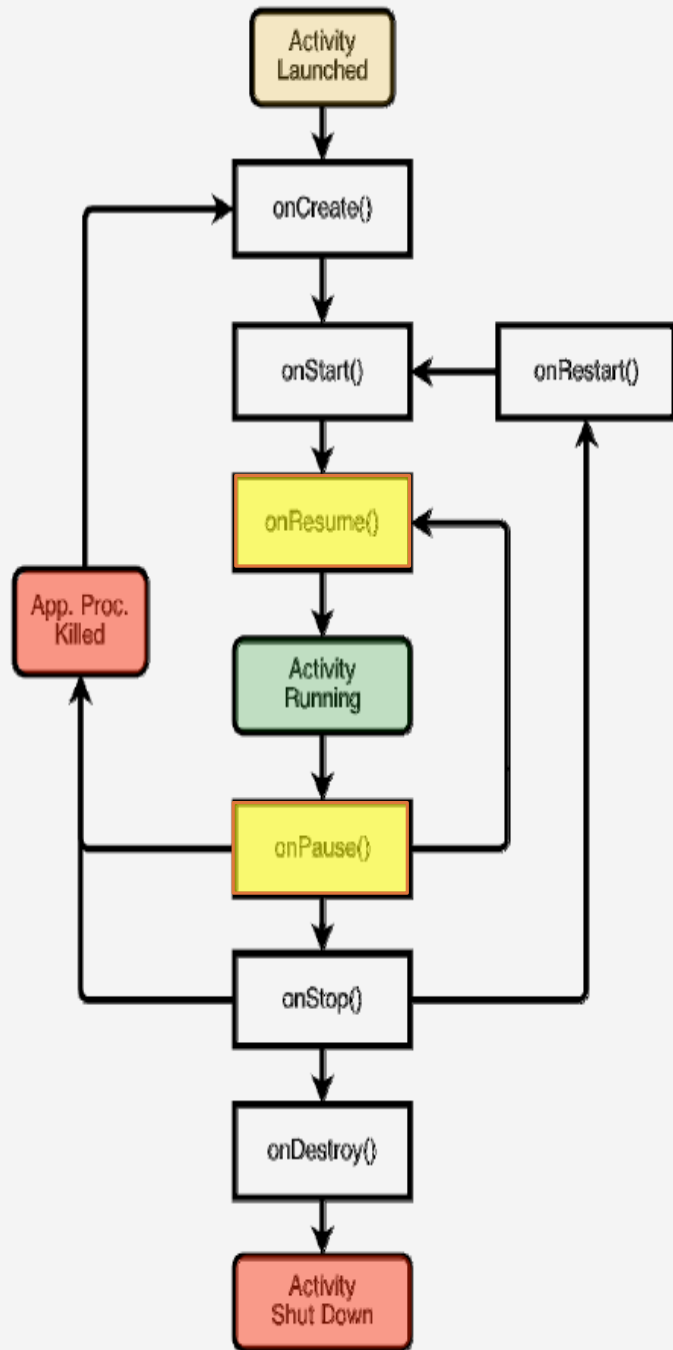
Lifetime Completo

Ciclo de vida



Lifetime Visible

Ciclo de vida



Lifetime Visible y en Foreground