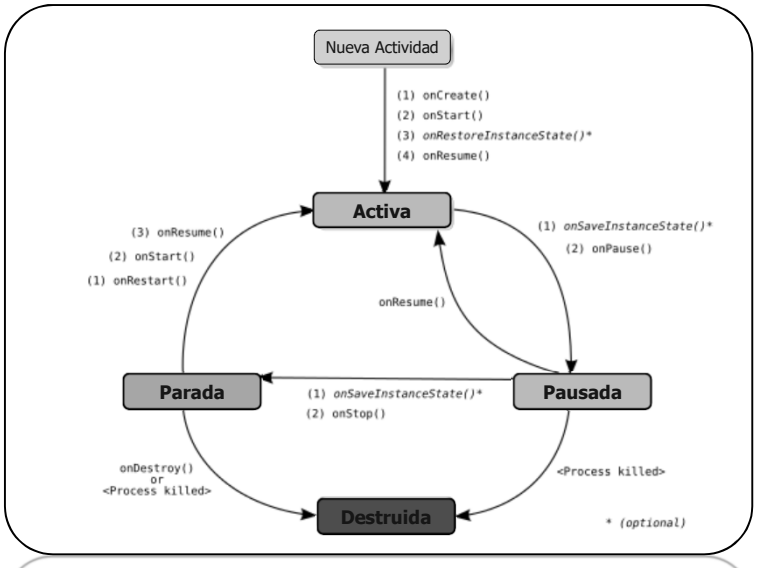
Ciclo de vida de una actividad (1)

A lo largo de la ejecución de una aplicación sus actividades pueden pasar por 4 estados:

- **Activa**
- **Pausada**
- **Parada**
- **Destruida**

42

Ciclo de vida de una actividad (2)

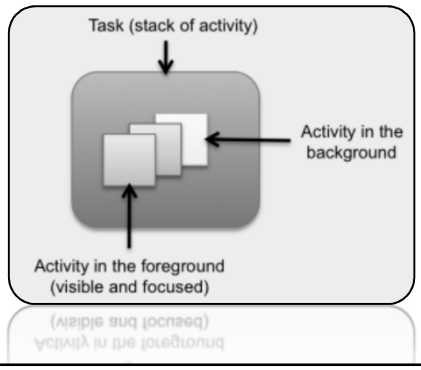


43

Ciclo de vida de una actividad (3)

- **Activa**:

La actividad se muestra en pantalla y tiene el foco, es decir, está disponible para que el usuario interactúe con ella.



44

Ciclo de vida de una actividad (4)

- **Pausada**:

La actividad ha pasado a segundo plano, pero aún está visible. Otra actividad se ha colocado sobre ella pero no la tapa por completo porque es transparente o mas pequeña.

La información de la actividad se mantiene y si vuelve al estado de "activa" todas las propiedades mantendrán sus valores.

La actividad puede ser "matada" por el sistema si se necesita liberar recursos.

45

Ciclo de vida de una actividad (5)

- **Parada**:

La actividad ha pasado a segundo plano y además está totalmente tapada por la nueva actividad.

La información de la actividad también se mantiene para poder devolverla a su estado si vuelve a activarse.

La actividad también puede ser "matada" por el sistema si se necesitan más recursos.

46

Ciclo de vida de una actividad (6)

- **Destruida**:

La actividad ya no está disponible, todos sus recursos se han liberado y si vuelve a ser llamada tendría que comenzar un nuevo ciclo de vida.

Si posee información importante deberá invocar a algún mecanismo de persistencia de los datos antes de ser destruida.

47

Ciclo de vida de una actividad (7)

El ciclo de vida de la actividad se gestiona mediante los siguientes métodos:

- ***onCreate()***
- ***onStart()***
- ***onResume()***
- ***onPause()***
- ***onStop()***
- ***onRestart()***
- ***onDestroy()***
- ***onSaveInstanceState()***
- ***onRestoreInstanceState()***

48

Ciclo de vida de una actividad (8)

- ***onCreate()***

Se llama al crear la actividad y se utiliza para preparar la interfaz gráfica de la pantalla y enlazar los datos con sus métodos de visualización.

Recibirá como parámetro un objeto *Bundle* con el estado anterior de la actividad en forma de pares clave-valor. Si el estado no se hubiera guardado el *Bundle* será nulo y no se podrá reestablecer el estado.

49

Ciclo de vida de una actividad (9)

- ***onStart()***

Se ejecuta a continuación de *onCreate()* y permite iniciar de forma efectiva la actividad pasando a ser visible.

- ***onResume()***

Se ejecuta antes de que el usuario pueda interactuar con la actividad y se encarga de ejecutar todos los tratamientos necesarios para el funcionamiento de la actividad, iniciar variables y listeners.

50

Ciclo de vida de una actividad (10)

Tras estos 3 métodos la actividad ya está lista para recibir la interacción del usuario.

Si otra actividad pasa a primer plano, la actividad en ejecución se pausará guardando previamente su estado.

- ***onSaveInstanceState()***

Se ejecuta antes de pausar una actividad y sirve para guardar su estado (objeto *Bundle*) y así poder volver a él cuando la actividad vuelva al estado "Activo".

51

Ciclo de vida de una actividad (11)

- ***onPause()***

Permite detener los tratamientos que no son necesarios cuando la actividad pasa a segundo plano y de esta manera liberar recursos para la actividad que la sustituye.

Si tras pausar la actividad, ésta vuelve a primer plano se ejecutará *onResume()*.

De lo contrario, se llamará a *onStop()* y la actividad quedará oculta.

52

Ciclo de vida de una actividad (12)

- ***onStop()***

Se ejecuta tras *onPause()* cuando la nueva actividad oculta por completo a la anterior y detendrá las tareas que no se pararon en la transición anterior.

Una vez parada la actividad puede volver a activarse mediante sucesivas llamadas a *onRestart()*, *onStart()* y *onResume()* o finalizar por completo mediante *onDestroy()*.

53

Ciclo de vida de una actividad (13)

- ***onRestart()***

Se ejecuta cuando tras estar parada, una actividad vuelve a estar activa, siguiéndole los métodos *onStart()* y *onResume()*.

Tras en método *onStart()* puede ejecutarse el método *onRestoreInstanceState()* para recuperar el estado de la actividad que se había guardado en *onSaveInstanceState()*.

54

Ciclo de vida de una actividad (14)

- ***onRestoreInstanceState()***

Puede ejecutarse tras *onStart()* y permitirá recoger el objeto *Bundle* con el estado anterior de la actividad en forma de pares clave-valor.

- ***onDestroy()***

Se llama antes de destruir una actividad y supone la liberación de todos sus recursos y la eliminación de todos sus datos asociados. Puede ser necesario implementar algún mecanismo de persistencia.

55