

## Ciclo de vida de un Activity:

### **onCreate:**

Se deben realizar tareas que deberían ocurrir una sola vez en todo el ciclo de vida que tiene un activity. Cosas que solo queremos que ocurran una vez, por ejemplo el seteo de la vista (lo que se dibujara) en dicho activity.

```
setContentView(R.layout.main_activity);
```

Obtener los componentes que tenga esa vista, es decir los botones, imágenes, etc

```
mTextView = (TextView) findViewById(R.id.text_view);
```

También para crear y dejar inicializados nuestros atributos.

```
miContador = 0;
```

### **onStart:**

Luego de que el sistema invoca a este método, el activity estará visible para el usuario y el sistema empieza a prepararlo para que el usuario pueda interactuar con los componentes que tiene nuestra actividad.

No debemos realizar tareas pesadas como cosas vinculadas por ejemplo a una base de datos, ya que este método debe ocurrir de forma muy rápida, ya que necesitamos que el usuario pueda tener interacción lo más pronto posible.

### **onResume:**

Aquí la actividad ya está visible para el usuario y este puede interactuar con los componentes. Pueda apretar botones, llenar textos , etc.

Entonces se dice que a partir de este momento el usuario tiene capacidad de interacción, y quedará en este estado hasta que ocurra algún evento de pausa por ejemplo, que será detallado a continuación.

Al método OnResume al igual que al OnStart el sistema los invocará si por ejemplo minimizamos nuestra aplicación para ir al HOME del celular y luego la reanudamos. Al reanudarla el sistema invocará a OnStart y acto seguido al OnResume.

Por lo tanto debemos tener en cuenta que ambos métodos pueden ser invocado más de una vez (A diferencia del onCreate que se invoca solo una vez).

**onPause:**

A este método lo invoca el sistema si un evento interrumpe nuestra activity y perdemos el foco parcial de la misma. Como podría ser por ejemplo una llamada telefónica.

También se puede dar el caso que otra activity pase a ser la que este en primer plano, por lo que nuestra activity perdería el foco. En ese caso se invocará a este método ANTES de que se cree la nueva activity. Por esta razón no debemos hacer tareas pesadas y debe ser muy rápido este método, porque sino estaría demorando el inicio de la segunda actividad.

Por eso no se recomienda realizar tareas que requieran el salvado en una base de datos por ej.

Aquí podríamos tener el ejemplo de detener la música que nuestra activity haga sonar.

O liberar ciertos recursos que nuestra activity esté consumiendo y que no serán necesarios hasta que volvamos a ponerla en primer plano. Por ej desactivar el uso del GPS que afectaría a la batería del celular.

En algún lado debemos guardar información importante de la activity por ejemplo enviar información a una base de datos, y si nuestra aplicación perdió el foco, existe la posibilidad que nunca lo recupere, por lo que las operaciones costosas para salvar cosas debemos hacerlas en el onStop.

**onStop:**

A este método se llega si nuestra actividad perdió el foco por completo.

Aquí también es un buen momento para liberar recursos que no se necesitarán.

Puede ocurrir que nuestra activity que había perdido el foco, vuelva a tomar el primer plano, en ese caso el sistema invocará al método onStart, luego al método onResume y luego al onPause.

**onDestroy:**

A este método se lo invoca justo antes de que el sistema destruya nuestra activity.

Puede ser que sea destruida ya sea porque el sistema necesita liberar memoria o porque nosotros mismos invocamos un método llamado finish().

Aquí se terminan de liberar todos los recursos que no han sido liberados anteriormente por ejemplo en los métodos onPause o en el onStop. El sistema invoca a este método cuando apretamos el botón "Back" de android y nuestra actividad es destruida.