

## Capítulo 6: Activity

Para comprender el tema de Activity es importante recordar que Java es la base para las aplicaciones nativas y una de las características principales de este lenguaje es el uso de clases, por lo tanto, cuando se crea una pantalla esta hereda de "AppCompatActivity" que es una clase predefinida por Android que será modificada según las necesidades del proyecto.

### Ciclo de vida.

Durante el desarrollo de una aplicación debe tenerse noción de todas las fases que atraviesa, estas fases son llamadas también estados y al pasar de un estado a otro se encuentran métodos predefinidos a los cuales se les pueden añadir instrucciones para que se ejecuten en ese preciso momento. Cabe destacar que, los cambios de estado suceden de forma rápida y algunos de ellos de forma casi imperceptible para el usuario.

**onCreate():** Se ejecuta al inicio de una actividad, este método por lógica solo se ejecuta una vez al momento de invocar a la actividad. En este se definen las variables y la interfaz de la actividad.

**onStart():** Momento donde la actividad se comienza a mostrar al usuario.

**onResume():** Momento en el que el usuario interactúa con la actividad, este es popularmente conocido como "corriendo" o "ejecutando".

**onPaused():** Se encuentra en pausa, no se pueden recibir datos de entrada del usuario, ni ejecutarse código, se encuentra semi-visible.

**onStop():** El método se encuentra completamente oculto para el usuario, es decir, se encuentra en el fondo.

**onRestart():** Este método se llama después del **onStop()** cuando la actividad actual se encuentra en proceso de volver a mostrarse al usuario.

**onDestroy():** Este método permite limpiar los recursos y destruye la actividad.

## Capítulo 7: Layouts

En el desarrollo web se emplea comúnmente HTML, sin embargo, en el caso de Android se utiliza XML, que tiene parecido a HTML puesto que de igual forma emplea etiquetas. Una de sus principales ventajas es que su estructura es rígida y estricta lo cual permite evitar errores en las aplicaciones.

Un layout puede ser entendido como un contenedor que permite asignar propiedades a los elementos que se colocaran dentro de los layouts.

### Tipos de layouts

Los tipos de layouts más empleados en el desarrollo de aplicaciones Android son:

**Linear layout:** El acomodo puede ser de dos formas: vertical u horizontal, si se selecciona vertical los elementos se colocarán de arriba hacia abajo, caso contrario, en horizontal se colocarán de izquierda a derecha.

**Relative Layout:** Los elementos se acomodan en una posición relativa a otro, es decir, los elementos se colocan tomando como referencia la posición de otro elemento.

**Frame layout:** Los elementos internos se acomodan en la parte superior izquierda, un error común es colocar varios elementos de este tipo dado que se quedaran montados unos sobre otros, impidiendo la visibilidad.