



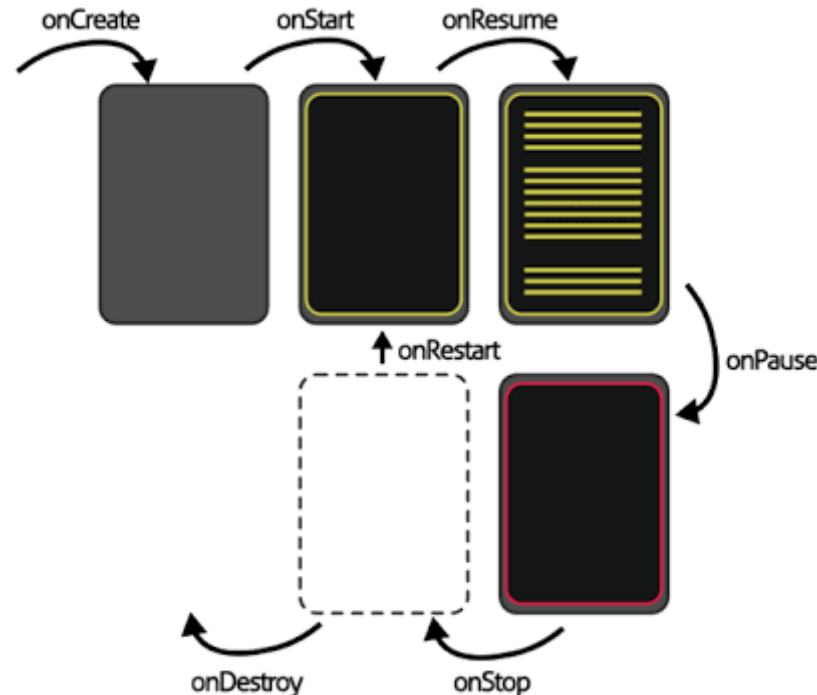
Universidad
Nacional de
Cajamarca
"Norte de la Universidad Peruana"

PROGRAMACIÓN APLICADA III

CICLO DE VIDA DE UNA ACTIVIDAD

I. Concepto

El ciclo de vida de una actividad proporciona una serie de devoluciones de llamada que permiten a la actividad saber que cambió un estado, es decir, que el sistema está creando, deteniendo o reanudando una actividad, o finalizando el proceso en el que se encuentra.

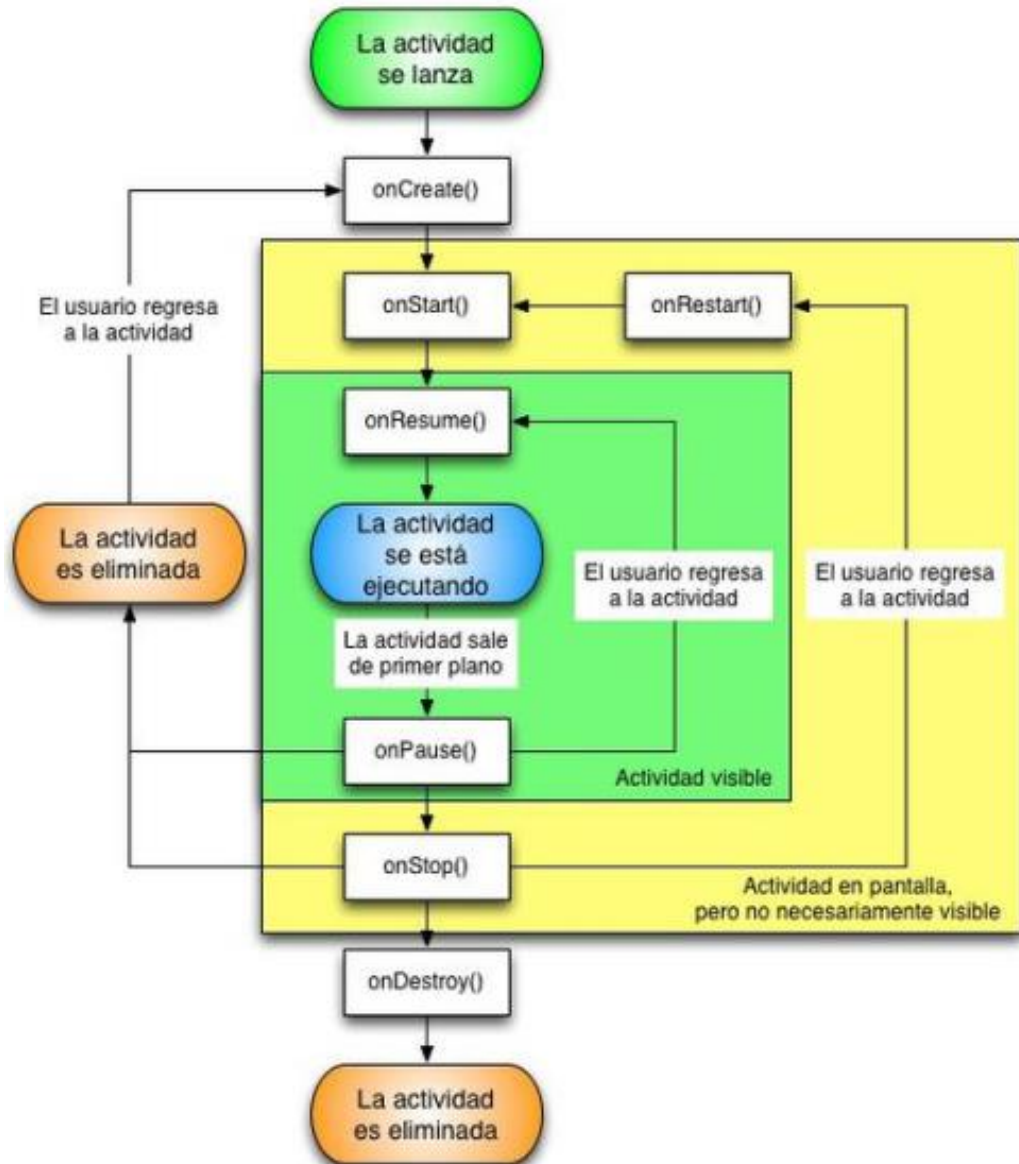


II. Representación visual



Para navegar por las transiciones entre las etapas del ciclo de vida de una actividad, la clase Activity proporciona un conjunto básico de seis devoluciones.

En algunos casos, este desmantelamiento es solo parcial; la actividad todavía reside en la memoria (por ejemplo, cuando el usuario cambia a otra app) y aún puede volver al primer plano.



III. Métodos de ciclo de vida

Actividad.

onCreate():

Se ejecuta la lógica de arranque básica de la aplicación que debe ocurrir una sola vez en toda la vida de la actividad. Por ejemplo, tu implementación de onCreate() podría vincular datos a listas.

Este método recibe el parámetro savedInstanceState, que es un objeto Bundle que contiene el estado ya guardado de la actividad. Si la actividad nunca existió, el valor del objeto Bundle es nulo.

```
@Override
public void onCreate(Bundle savedInstanceState) {
    // call the super class onCreate to complete the creation of activity like
    // the view hierarchy
    super.onCreate(savedInstanceState);

    // recovering the instance state
    if (savedInstanceState != null) {
        gameState = savedInstanceState.getString(GAME_STATE_KEY);
    }

    // set the user interface layout for this activity
    // the layout file is defined in the project res/layout/main_activity.xml file
    setContentView(R.layout.main_activity);

    // initialize member TextView so we can manipulate it later
    textView = (TextView) findViewById(R.id.text_view);
}
```

III. Métodos de ciclo de vida

Actividad.



onStart():

La llamada onStart() hace que el usuario pueda ver la actividad mientras la app se prepara para que esta entre en primer plano y se convierta en interactiva.

onResume():

Cuando la actividad entra en el estado onResume, pasa al primer plano y, a continuación, el sistema invoca la devolución de llamada onResume(). Este es el estado en el que la app interactúa con el usuario. La app permanece en este estado hasta que ocurre algún evento que la quita de foco.

onPause():

El sistema llama a este método a modo de primera indicación de que el usuario está abandonando tu actividad (aunque no siempre significa que está finalizando la actividad); esto indica que la actividad ya no está en primer plano (aunque puede seguir siendo visible si el usuario está en el modo multiventana).

III. Métodos de ciclo de vida

Actividad.



onStop():

Cuando el usuario ya no puede ver tu actividad, significa que ha entrado en el estado Stopped, y el sistema invoca la devolución de llamada onStop(). Esto puede ocurrir, por ejemplo, cuando una actividad recién lanzada cubre toda la pantalla. El sistema también puede llamar a onStop() cuando haya terminado la actividad y esté a punto de finalizar.

onDestroy():

Se llama a onDestroy() antes de que finalice la actividad. El sistema invoca esta devolución de llamada por los siguientes motivos:

- 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).

IV. Estados de una Actividad



Una actividad en Android puede estar en uno de estos cuatro estados:

Activa (Running): La actividad está encima de la pila, lo que quiere decir que es visible y tiene el foco.

Visible (Paused): La actividad es visible pero no tiene el foco. Se alcanza este estado cuando pasa a activa otra actividad con alguna parte transparente o que no ocupa toda la pantalla. Cuando una actividad está tapada por completo, pasa a estar parada.

Parada (Stopped): Cuando la actividad no es visible. El programador debe guardar el estado de la interfaz de usuario, las preferencias, etc.

Destruída (Destroyed): Cuando la actividad termina, al invocarse el método `finish()`, o cuando es matada por el sistema.

Práctica



Abrir el proyecto anterior(AppApiLista) en la actividad de facultad implementar los métodos correspondiente al ciclo de vida de una actividad y presentar los mensajes de los estados en un TextView y posteriormente en un ListView.