

## APLICACIONES, ACTIVIDADES Y PROCESOS

- **Aplicación:** conjunto de actividades y servicios.
- **Actividad:** cada una de las pantallas de la aplicación.
  - Desde una actividad podemos lanzar nuevas actividades de mi aplicación o de otras aplicaciones.
  - El botón retroceder permite destruir la actividad actual y volver a la anterior.
- Cada aplicación se ejecuta en un **Proceso** Linux distinto.
  - De esta forma, se delega en Linux tareas tan vitales como seguridad o multiproceso.
  - Todas las actividades y servicios de una aplicación se ejecutan en el mismo proceso.

# CICLO DE VIDA DE LAS APLICACIONES

Procesos en memoria

Sistema

Inicio

Inicio



## Procesos en memoria

Sistema

Bandeja entrada

Inicio

Inicio



Correo



## Procesos en memoria

### Sistema

Navegador

Visor correo

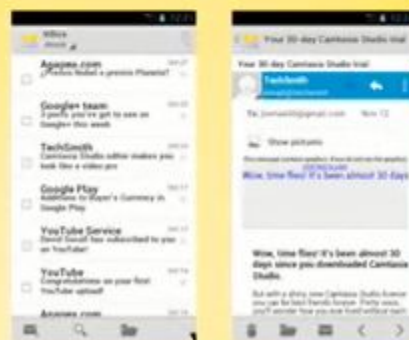
Bandeja entrada

Inicio

### Inicio



### Correo



### Navegador



## Procesos en memoria

### Sistema

Teléfono

Navegador

Visor correo

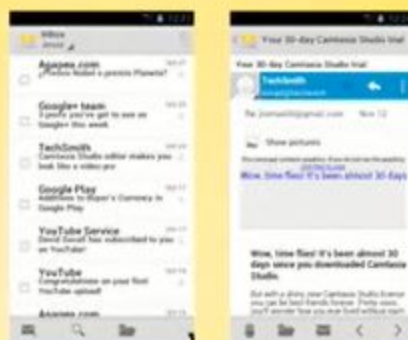
Bandeja entrada

Inicio

### Inicio



### Correo



### Navegador



## Procesos en memoria

### Sistema

Teléfono

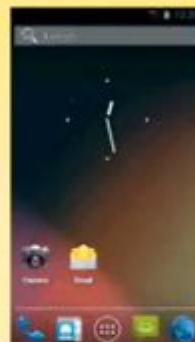
Navegador

Visor correo

Bandeja entrada

Inicio

### Inicio



### Teléfono



### Navegador





## Procesos en memoria

Sistema

Navegador

Visor correo

Bandeja entrada

Inicio

Inicio



Teléfono

Navegador



## Procesos en memoria

Sistema

Visor correo

Bandeja entrada

Inicio

Inicio



Teléfono

Navegador



## Procesos en memoria

Sistema

Visor correo

Bandeja entrada

Inicio

Inicio



Correo



Navegador

## Procesos en memoria

Sistema

Bandeja entrada

Inicio

Inicio



Correo



Navegador

## Procesos en memoria

Sistema

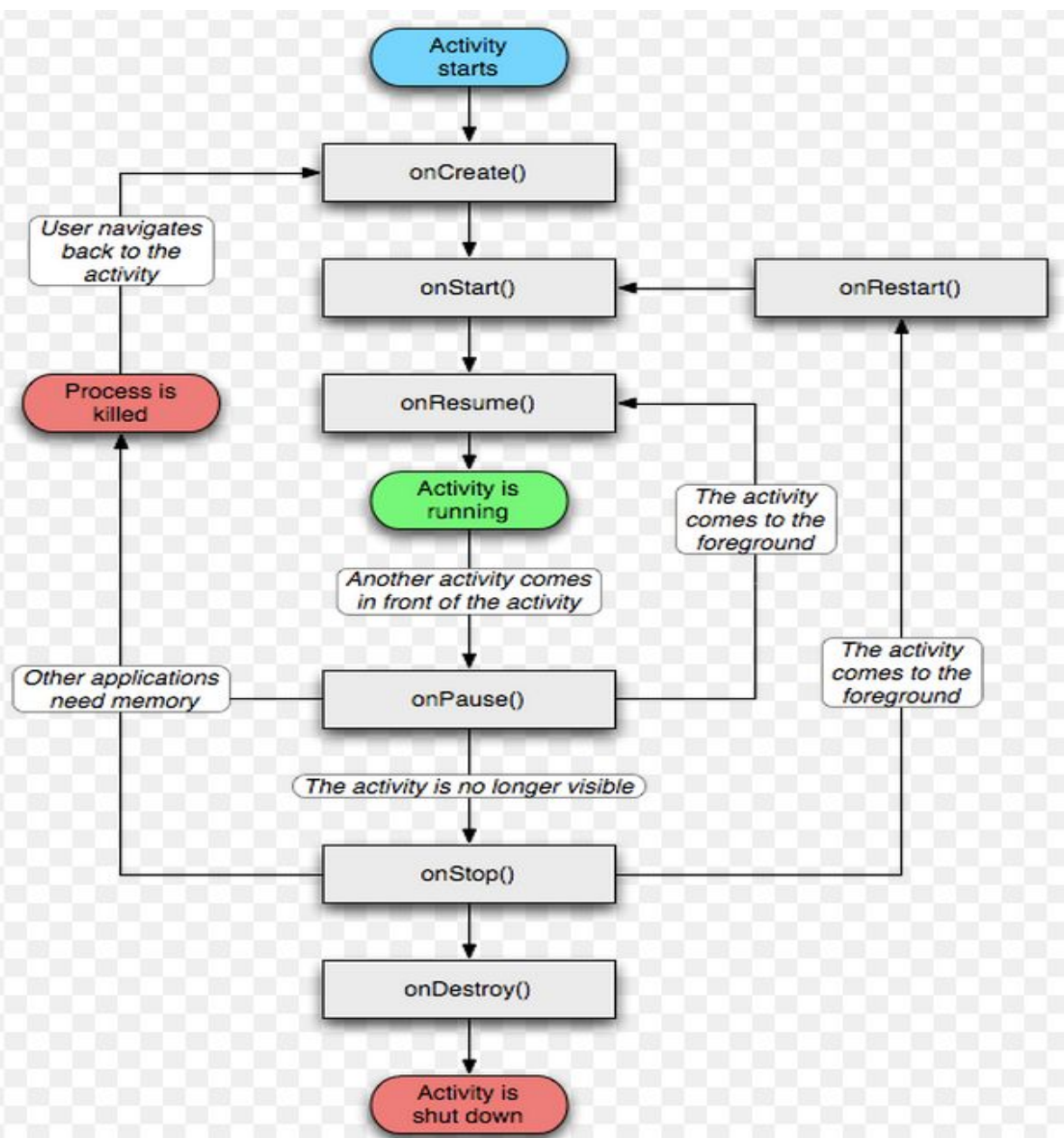
Inicio

Inicio



Correo

Navegador



**onCreate(Bundle):** Se llama en la creación de la actividad. 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 información de estado de la actividad (en una instancia de la clase **Bundle**), por si se reanuda desde una actividad que ha sido destruida y vuelta a crear.

**onStart():** Nos indica que la actividad está a punto de ser mostrada al usuario.

**onResume():** Se llama cuando la actividad va a comenzar a interactuar con el usuario. Es un buen lugar para lanzar las animaciones y la música.

**onPause():** Indica que la actividad está a punto de ser lanzada a segundo plano, normalmente porque otra actividad es lanzada. Es el lugar adecuado para detener animaciones, música o almacenar los datos que estaban en edición.

**onStop():** La actividad ya no va a ser visible para el usuario. Ojo si hay muy poca memoria, es posible que la actividad se destruya sin llamar a este método.

**onRestart():** Indica que la actividad va a volver a ser representada después de haber pasado por onStop().

**onDestroy():** Se llama antes de que la actividad sea totalmente destruida. Por ejemplo, cuando el usuario pulsa el botón de volver o cuando se llama al método finish(). Ojo si hay muy poca memoria, es posible que la actividad se destruya sin llamar a este método.

- Los métodos de Activity que nos interesan son onCreate, onResume y onPause.
- **Ejercicio.** Haz que aparezca un mensaje la crear la actividad, al dibujarse en pantalla, cuando pase a segundo plano y cuando se destruya (Log o Toast)



```

public class LifeCycleTest extends Activity {
    StringBuilder builder = new StringBuilder();
    TextView textView;

    private void log(String text) {
        Log.d("LifeCycleTest", text);
        builder.append(text);
        builder.append('\n');
        textView.setText(builder.toString());
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        textView = new TextView(this);
        textView.setText(builder.toString());
        setContentView(textView);
        log("created");
    }

    @Override
    protected void onResume() {
        super.onResume();
        log("resumed");
    }

    @Override
    protected void onPause() {
        super.onPause();
        log("paused");

        if (isFinishing()) {
            log("finishing");
        }
    }
}

```