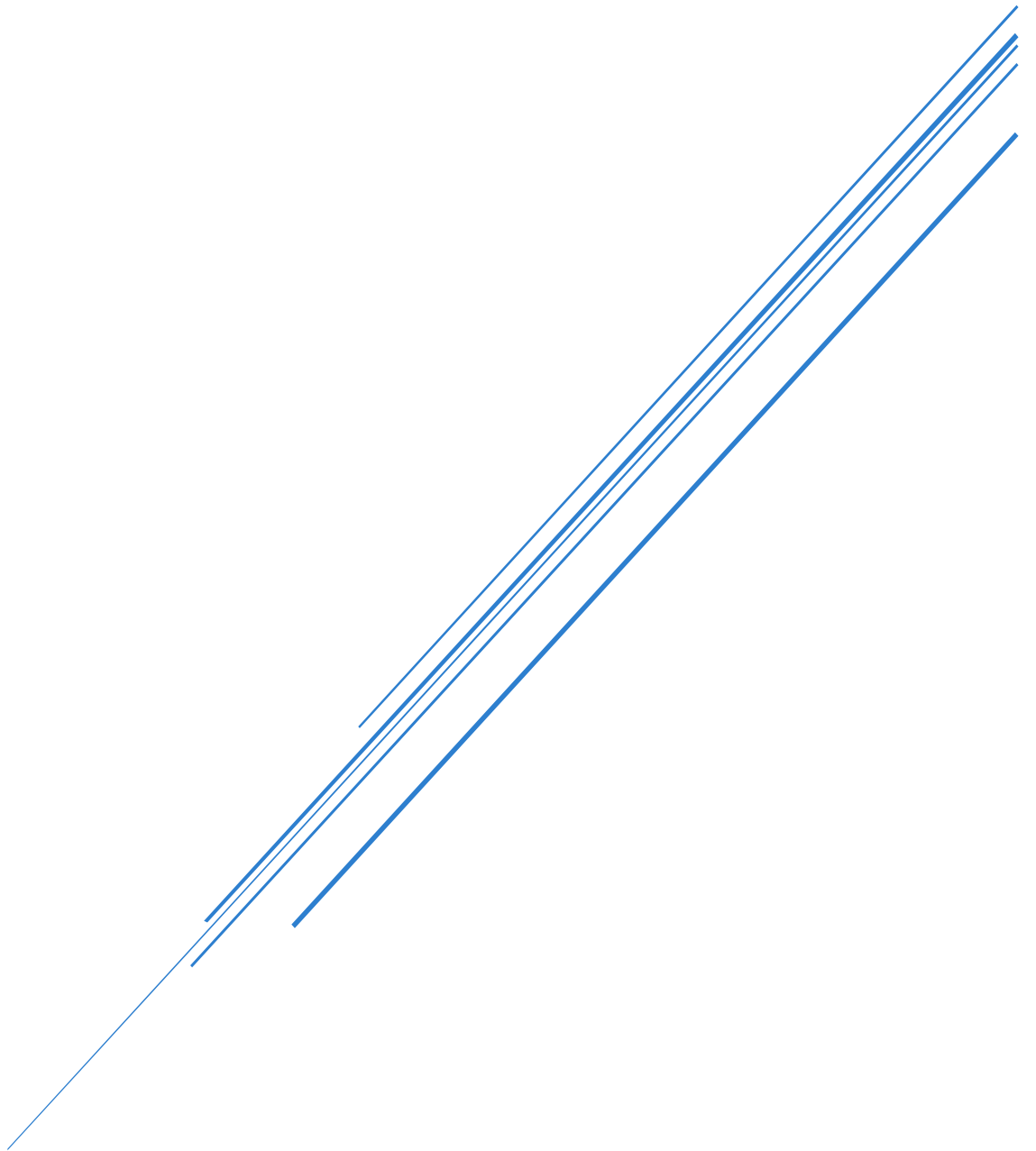


# ACTIVIDAD M4-01

Ciclo de vida de una activity



Jorge Ernesto Ceballos Guzman

1. Como siempre, se inicia con proyecto nuevo de Empty Views Activity
2. En la clase MainActivity se ponen los siguientes métodos del ciclo de vida de las actividades
  - a. onCreate

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //Imprimir al log de Android el estado de la actividad
    Log.d( tag: "onCreate:", msg: "La actividad se ha creado");

    //Inicializar el elemento de texto
    txtvEstado = findViewById(R.id.txtvEstado);
    //Imprimir al layout el estado de la actividad
    agregarEstado("onCreate: La actividad se ha creado");

    //Inicializar el botón
    Button btnCambiarTexto = findViewById(R.id.btnCambiarTexto);

    //Definir el comportamiento del botón
    btnCambiarTexto.setOnClickListener( View v -> {
        agregarEstado("onClick: La actividad se ha creado");
    });

    //Inicializar el botón
    Button btnIrSegunda = findViewById(R.id.btnIrSegunda);
    //Definir el comportamiento del botón
    btnIrSegunda.setOnClickListener( View v -> {
        Intent intent = new Intent( packageContext: this, SecondaryActivity2.class);
        startActivity(intent);
    });
}
```

En este método, básicamente se inicializa a la actividad, para este ejemplo, se incluye un botón para cambiar el texto y para cambiar de actividad para probar distintas partes del ciclo, como el stop o el resume.

- Se imprime con un Log.d lo que se esta haciendo
- Se inicializa a los elementos de texto necesarios para mostrar el estado
- Con un método propio llamado agregarEstado se acumula el estado de la actividad en un texto mostrado en pantalla
- Se inicializan los botones que detonan diferentes estados, como el de cambiar de texto o ir a la otra actividad, y se les provee de su comportamiento mediante funciones lambda

b. onStart

```
protected void onStart() {  
    super.onStart();  
    Log.d( tag: "onStart:", msg: "La actividad ha comenzado");  
    agregarEstado("onStart: La actividad ha comenzado");  
}
```

Imprime en Log y acumula con el método agregarEstado el estado actual de la actividad.

c. onResume

```
protected void onResume() {  
    super.onResume();  
    Log.d( tag: "onResume:", msg: "La actividad ha continuado");  
    agregarEstado("onResume: La actividad ha continuado");  
}
```

Imprime en Log y acumula con el método agregarEstado el estado actual de la actividad.

d. onPause

```
protected void onPause() {  
    super.onPause();  
    Log.d( tag: "onPause:", msg: "La actividad ha pausado");  
    agregarEstado("onPause: La actividad ha pausado");  
}
```

Imprime en Log y acumula con el método agregarEstado el estado actual de la actividad.

e. onStop

```
protected void onStop() {  
    super.onStop();  
    Log.d( tag: "onStop:", msg: "La actividad se ha detenido");  
    agregarEstado("onStop: La actividad se ha detenido");  
}
```

Imprime en Log y acumula con el método agregarEstado el estado actual de la actividad.

f. onDestroy

```
protected void onDestroy() {  
    super.onDestroy();  
    Log.d( tag: "onDestroy:", msg: "La actividad se ha destruido");  
    agregarEstado("onDestroy: La actividad se ha destruido");  
}
```

Imprime en Log y acumula con el método agregarEstado el estado actual de la actividad.

3. Se modifica el Layout de la activity\_main para poder agregar los elementos necesarios para mostrar el ciclo de vida de la actividad, como los botones o el textview

a. TextView

```
<TextView  
    android:id="@+id/txvEstado"  
    android:layout_width="match_parent"  
    android:layout_height="200dp"  
    android:background="#FFFFFF"  
    android:text="Estados del ciclo de vida:\n"  
    android:textSize="16sp"  
    android:textColor="#000000"  
    android:padding="16dp"  
    android:gravity="start"  
    android:scrollbars="vertical" />
```

Permite mostrar y acumular los diferentes estados

b. Botón de cambiar de texto

```
<Button  
    android:id="@+id/btnCambiarTexto"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Cambiar Texto" />
```

c. Botón de ir a segunda actividad

```
<Button
    android:id="@+id/btnIrSegunda"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Ir a otra Activity"
    android:layout_marginTop="16dp" />
```

4. Se crea una segunda actividad de Empty Views Activity para poder probar los diferentes métodos del ciclo de vida de las actividades

```
© MainActivity.java    © SecondaryActivity2.java x
1  package com.example.actividad_m4_01;
2
3  import android.os.Bundle;
4  import androidx.appcompat.app.AppCompatActivity;
5
6  ▶ </> public class SecondaryActivity2 extends AppCompatActivity {
7
8      @Override
9  Ⓢ protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         //Configuración de la vista de la actividad 2
12         setContentView(R.layout.activity_secondary2);
13     }
14 }
```

5. Se modifica el layout de la nueva actividad creada

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:orientation="vertical"
6      android:gravity="center"
7      android:padding="16dp"
8      android:background="#FFF3E0">
9
10     <TextView
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:text="Segunda actividad"
14         android:textSize="24sp"
15         android:textColor="#FF5722" />
16 </LinearLayout>

```

## 6. Evidencias del ciclo de vida de las actividades en Log

Logcat Logcat x +

Xiaomi 2310FPCA4G (I7T07X4TWGTWSD9T) Android package:mine

Time	Level	Class	Message
2025-05-16 22:03:43.922	INFO	CompatChangeReporter	com.example.actividad_m4_01
2025-05-16 22:03:43.982	INFO	actividad_m4_01	com.example.actividad_m4_01
2025-05-16 22:03:44.005	INFO	CompatChangeReporter	com.example.actividad_m4_01
2025-05-16 22:03:44.156	INFO	onCreate	com.example.actividad_m4_01
2025-05-16 22:03:44.196	INFO	onStart	com.example.actividad_m4_01
2025-05-16 22:03:44.207	INFO	libc	com.example.actividad_m4_01
2025-05-16 22:03:44.211	INFO	SecurityManager	com.example.actividad_m4_01

Logcat Logcat x +

Xiaomi 2310FPCA4G (I7T07X4TWGTWSD9T) Android package:mine

Time	Level	Class	Message
2025-05-16 22:05:21.702	INFO	VRI[MainActivity]	com.example.actividad_m4_01
2025-05-16 22:05:21.724	INFO	HWUI	com.example.actividad_m4_01
2025-05-16 22:05:21.733	INFO	BLASTBufferQueue	com.example.actividad_m4_01
2025-05-16 22:05:21.733	INFO	BufferQueueConsumer	com.example.actividad_m4_01
2025-05-16 22:05:21.738	INFO	onStop	com.example.actividad_m4_01
2025-05-16 22:05:21.756	INFO	ActivityThread	com.example.actividad_m4_01

Logcat Logcat x +

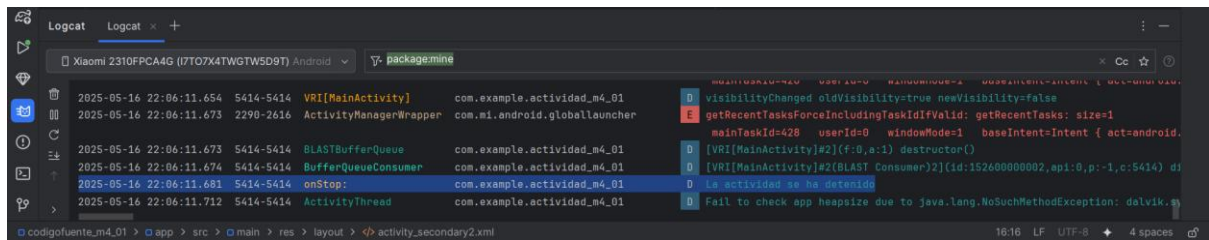
Xiaomi 2310FPCA4G (I7T07X4TWGTWSD9T) Android package:mine

Time	Level	Class	Message
2025-05-16 22:05:44.654	INFO	MIUIInput	com.example.actividad_m4_01
2025-05-16 22:05:44.670	INFO	MIUIInput	com.example.actividad_m4_01
2025-05-16 22:05:44.677	INFO	MIUIInput	com.example.actividad_m4_01
2025-05-16 22:05:44.703	INFO	onStart	com.example.actividad_m4_01
2025-05-16 22:05:44.713	INFO	SecurityManager	com.example.actividad_m4_01
2025-05-16 22:05:44.715	INFO	onResume	com.example.actividad_m4_01
2025-05-16 22:05:44.740	INFO	BufferQueueConsumer	com.example.actividad_m4_01

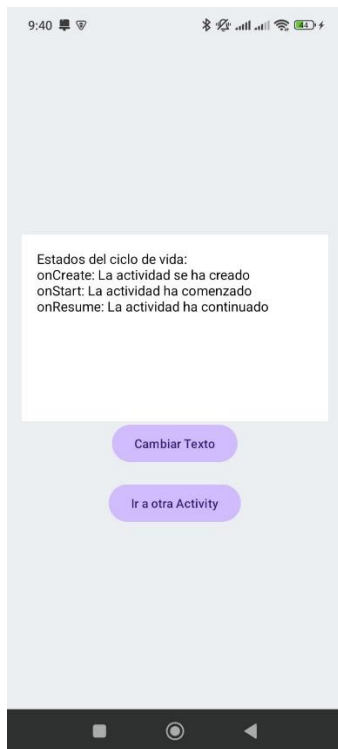
Logcat Logcat x +

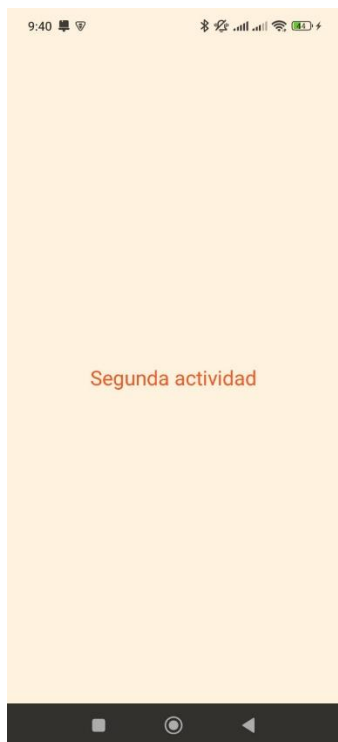
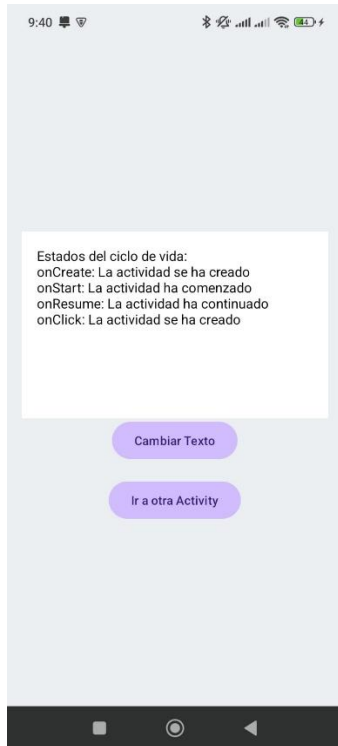
Xiaomi 2310FPCA4G (I7T07X4TWGTWSD9T) Android package:mine

Time	Level	Class	Message
2025-05-16 22:06:10.804	INFO	onPause	com.example.actividad_m4_01
2025-05-16 22:06:10.821	INFO	ActivityManagerWrapper	com.mi.android.globallauncher
2025-05-16 22:06:10.933	INFO	ActivityManagerWrapper	com.mi.android.globallauncher
2025-05-16 22:06:11.001	INFO	ActivityManagerWrapper	com.mi.android.globallauncher



## 7. Evidencias del ciclo de vida de las actividades en la interfaz del usuario`







9:40 4G



Estados del ciclo de vida:  
onCreate: La actividad se ha creado  
onStart: La actividad ha comenzado  
onResume: La actividad ha continuado  
onClick: La actividad se ha creado  
onPause: La actividad se ha pausado  
onStop: La actividad se ha detenido  
onStart: La actividad ha comenzado  
onResume: La actividad ha continuado

Cambiar Texto

Ir a otra Activity



