

Estados de una aplicación: ciclo de vida

- **onCreate()**
Configurar la interfaz de usuario:
 - setContentView(R.layout.activity_main)
- **onStart()**
Iniciar la escucha de datos en tiempo real, como un LiveData o un Observer.
 - ViewModel.data.observe(this) { data -> print("Ha cambiado: \$data") }
- **onResume()**
Registrar receptores dinámicos relacionados con eventos de UI.
- **onPause()**
Guardar cambios en curso: datos temporales o persistentes como en SharedPreferences.
 - val preferences = getSharedPreferences("AppPreferences", MODE_PRIVATE)
- **onStop()**
Guardar datos más completos, como el estado de UI en caso de que el sistema finalice la actividad.
 - savedInstanceState?.putString("inputText", editText.text.toString())
- **onRestart()**
Reanudar estados interrumpidos y actualizar datos visibles o recursos que podrían haber cambiado mientras la actividad estaba detenida.
 - editText.setText("Bienvenido de nuevo")
- **onDestroy()**
Liberar completamente los recursos asignados: cerrar conexiones, anular registros de receptores, etc.
 - database?.close()

Nota: el interrogante (?) indica que la variable es nullable, es decir, que puede ser Null.