

# Universidade da Beira Interior

## Departamento de Informática

[width=191pt]brasaoubi.JPG

**Grupo Nº 4 - 2024: *[Navegação e ciclo de Vida- Ensaio Teorico]***

Elaborado por:

**[Clausemen Nanro, Ivandra Bernardo, João Maravilhoso]**

Orientador:

**Professor**

October 25, 2025

# Ciclo de Vida de Atividades e Intents no Android

## 1 Ciclo de Vida da Atividade

O ciclo de vida de uma *Activity* no Android é fundamental para desenvolver aplicativos robustos. Uma *Activity* representa uma tela interativa, gerenciada por callbacks como `onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()` e `onDestroy()`. Cada callback permite ações específicas: `onCreate()` inicializa a interface e dados, `onStart()` torna a atividade visível, `onResume()` a coloca em primeiro plano, `onPause()` e `onStop()` liberam recursos, e `onDestroy()` finaliza a atividade. A preservação do estado, usando `onSaveInstanceState()` e `ViewModel`, garante a continuidade da experiência do usuário durante mudanças de configuração, como rotação de tela [1].

## 2 Intents no Android

Os *Intents* são estruturas que facilitam a comunicação entre componentes do Android, como *Activities*, *Services* e *BroadcastReceivers*. Eles descrevem operações a serem executadas, como iniciar uma *Activity* (`startActivity()`), enviar uma transmissão (`broadcastIntent()`) ou conectar a um serviço (`startService()`). Um *Intent* contém *action* (e.g., `ACTIONVIEW`),

### 2.1 Tipos de Intents

*Intents* explícitos especificam um componente (e.g., `SecondActivity`), enquanto *Intents* implícitos deixam o sistema escolher o componente com base em *action*, *type* e *category*. Por exemplo, um *Intent* explícito inicia uma *Activity* específica, enquanto um *Intent* implícito com `ACTION_DIAL` `tel:123abreodiscador`. A resolução de *Intents* é feita

## 3 Integração entre Ciclo de Vida e Intents

A interação entre *Activities* via *Intents* impacta o ciclo de vida. Quando uma *Activity* A inicia uma *Activity* B com `startActivity()`, a *Activity* A entra em `onPause()`, enquanto a *Activity* B passa por `onCreate()`, `onStart()` e `onResume()`. Se a *Activity* A ficar invisível, `onStop()` é chamado. Para *Intents* com resultado (`startActivityForResult()`), a *Activity* pai processa o resultado em `onActivityResult()`. Gerenciar esses callbacks garante transições suaves e preserva o estado [1].

## 4 Conclusão

Compreender o ciclo de vida de *Activities* e o uso de *Intents* é essencial para aplicativos Android robustos. Callbacks bem implementados e *intent-filters*

configurados adequadamente permitem gerenciar recursos, preservar estados e facilitar a comunicação entre componentes, alinhando-se às práticas recomendadas [3].

## References

- [1] Documentação do Android, "Ciclo de Vida da Atividade," <https://developer.android.com/guide/components/activities/activity-lifecycle>, acessado em 24 de outubro de 2025.
- [2] Documentação do Android, "Intents e Filtros de Intent," <https://developer.android.com/guide/components/intents-filters>, acessado em 24 de outubro de 2025.
- [3] Murphy, M. L., *The Busy Coder's Guide to Android Development*, CommonsWare, 2018.