

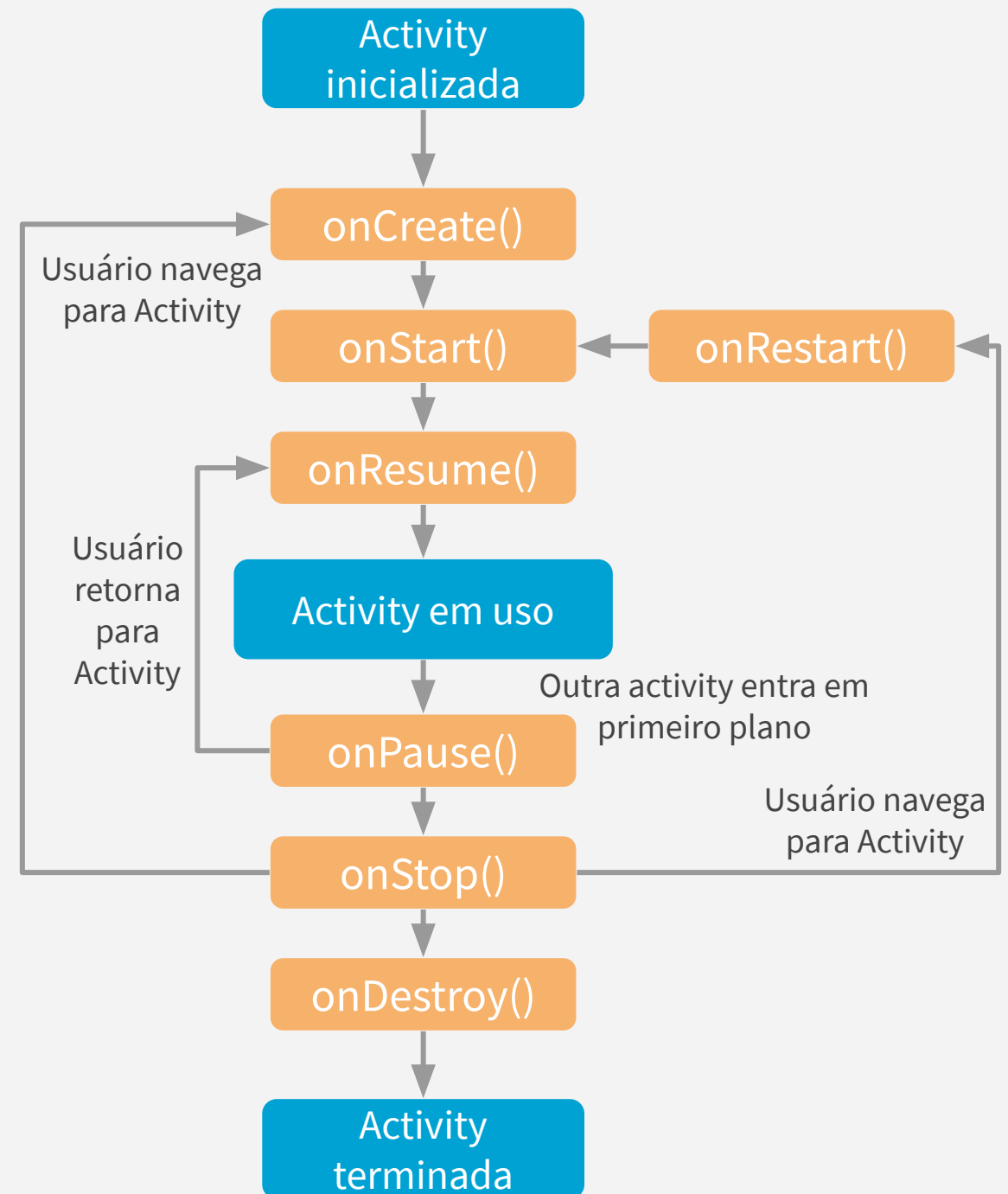
Ciclo de vida de Activity

Conceito

Uma Activity representa uma tela com interface gráfica capaz de promover algum tipo de interação com o usuário. Uma aplicação Android pode ser composta de diversas activities para fornecer um conjunto de funcionalidades para o usuário.

Compreender o funcionamento e as transições é importante para o desenvolvimento e implementação da lógica da aplicação. Por exemplo, ao trabalhar com um aplicativo de vídeo, pode ser um requisito pausar o vídeo se o dispositivo receber uma ligação ou se outro aplicativo for aberto simultaneamente. Dentro do ciclo de vida da Activity, é possível declarar o comportamento quando o usuário deixa ou retorna a activity, de maneira a atingir esses requisitos.

A activity possui um ciclo de vida específico. Quando um usuário abre o aplicativo, troca de tela ou deixa o aplicativo em segundo plano, existe uma transição entre estados descritos abaixo. Em outras palavras, as transições no ciclo de vida permitem a execução de códigos e funcionalidades apropriados para cada estado da activity.



Callbacks no ciclo de vida

Sempre que uma activity muda de estado, o Android faz a chamada de um método (callback) correspondente.

onCreate()

Quando um aplicativo é aberto, o Android sabe qual a página inicial que deve ser criada, uma vez que está definida no AndroidManifest.xml. Com isso, o método onCreate() da activity é invocado.

O método onCreate, embora não seja obrigatório, é sempre implementado por padrão na criação do projeto ou na criação de outras Activities. É criado por padrão por ser o melhor local para definição do layout, identificação dos elementos e inicialização de variáveis que serão usadas durante para implementação. Além disso, popular uma lista, fazer uma chamada à API ou acesso ao banco de dados são exemplos do que pode ser feito dentro deste método.

onStart()

Esse método faz com que a activity seja visível para o usuário, enquanto o aplicativo prepara para a activity entrar em funcionamento e se tornar interativa.

O método onStart() é executado rapidamente e não é recomendado para processamentos pesados. Assim que esse callback é terminado, a activity entra no modo resumido.

onResume()

Quando a activity entrar no estado resumido, ela entra em funcionamento e está pronta para interação com o usuário. Assim que o aplicativo entra nesse estado, permanecerá assim até que algo aconteça e tire o foco do aplicativo. Por exemplo, uma chamada telefônica, um alarme ou a tela do aplicativo sendo bloqueada ou desligada.

Se a activity retorna ao estado Resumido, o sistema chama onResume() novamente e não onStart(), uma vez que ela já foi iniciada e já está em memória. Por esse motivo, é necessário implementar onResume() para reinicializar componentes descartados no onPause().

Este método é chamado todas as vezes que uma activity se torna visível para o usuário, inclusive a primeira vez após a chamada do método onCreate() e onStart().

onPause()

Quando qualquer um dos eventos descritos no método onResume ocorre, o Android faz a chamada do callback onPause(). Esse método é chamado como o primeiro indicativo de que o usuário está saindo da activity. Este método deve ser usado para pausar animações e músicas que não devem continuar quando a tela está pausada.

Por exemplo, se o aplicativo utiliza a câmera, o método onPause() é um ótimo local para desalocar o recurso da câmera para ser alocada posteriormente no método onResume().

A execução desse método é muito breve e não necessariamente dispõe de muito tempo para performar ações de salvar dados. Por esse motivo, esse método não deve ser usado para salvar dados da aplicação, chamadas a APIs ou banco de dados. Tais ações devem ser feitas no método onStop() descrito abaixo.

onStop()

Quando a activity não está mais visível para o usuário, porém ainda permanece alocada na memória o sistema invoca o callback onStop().

É neste método que todos os recursos alocados na criação devem ser descartados. O método também é recomendado quando se deseja realizar alguma operação que use relativamente a CPU, como por exemplo, salvar dados no banco de dados ou uma chamada a uma API.

A partir deste estado, a activity volta a interagir com o usuário ou é finalizada e removida da memória. Se a activity voltar, o sistema invoca onStart(). Se a activity terminar de ser executada, o sistema chamará onDestroy().

onDestroy()

Método chamado antes da activity ser destruída. Esta é a chamada final que a Activity recebe. Este método libera todos os recursos alocados que não tenham sido descartados previamente durante os outros métodos.

Fontes

1. Developer Android. **Understand the Activity Lifecycle**. Disponível em <<https://developer.android.com/guide/components/activities/activity-lifecycle>>. Acesso em 22 de fevereiro de 2020.