

Programação para Dispositivos Móveis



Activities

Google Android

Activity

Uma *activity* é uma tarefa, muito focada, do que um usuário pode fazer. Quase todas as atividades interagem com o usuário, então uma classe de atividade toma conta da criação de uma janela para você onde você poderá colocar todos os componentes de UI (User Interface) com o `setContentView(view)`.



A classe Activity é a mais importante no ciclo de vida de uma aplicação e a forma como as atividades são lançadas e unidas é parte fundamental para a plataforma de modelo de aplicação.



Google Android

```
void setContentView(int layoutResID)
```

É todo responsável por colocar um conteúdo de um recurso de layout, o recurso será inflado na activity.

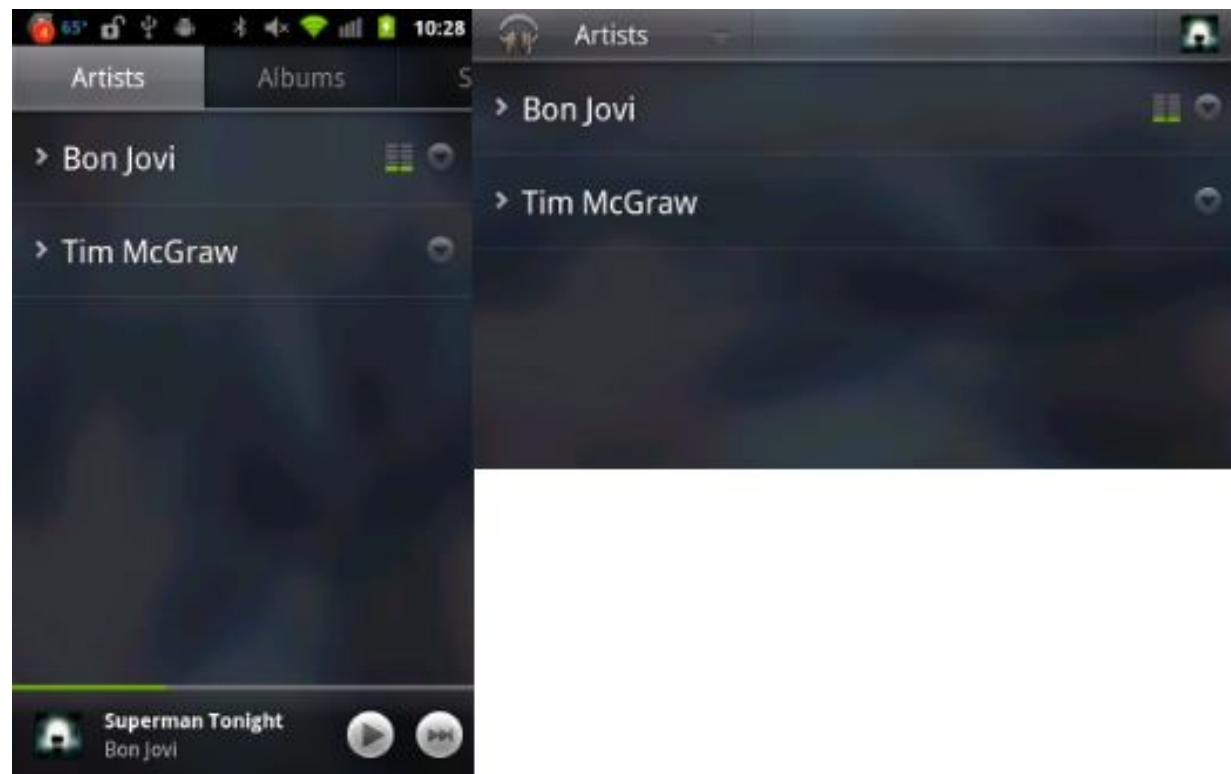


Google Android

Orientation e o Ciclo de Vida

✓ Portrait

✓ Landscape



Google Android

Ciclo de Vida

Activities são gerenciadas no sistema por uma *activity stack*.

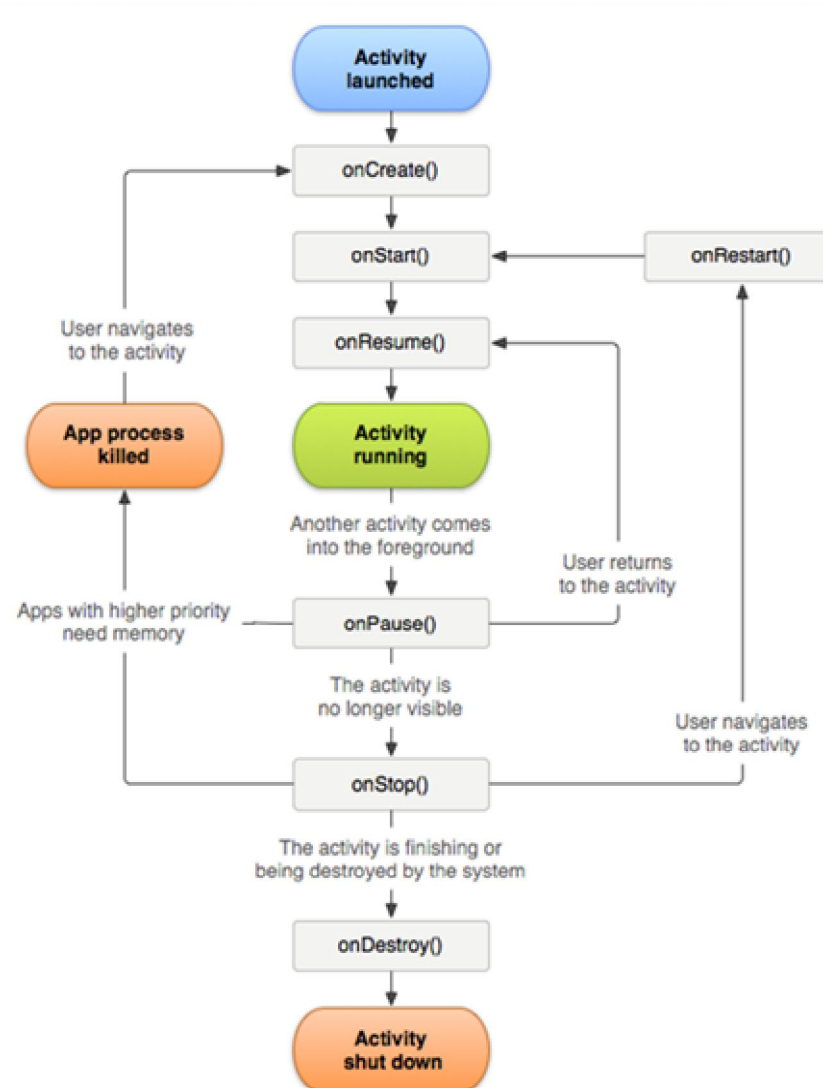
Uma Activity tem quatro estados essenciais.

- ✓ **modo active ou running:** atividade está sendo executada e está sendo mostrada na tela.
 - ✓ **modo paused:** atividade perdeu o foco mas ainda assim está visível (ou seja, uma nova atividade está sendo mostrada na tela mas não ocupando-a completamente - atividades em janelas flutuantes).
 - ✓ **modo stopped:** atividade é completamente obscurecida por outra atividade.
 - ✓ **modo killed:** Se uma atividade está em modo paused ou stopped, o sistema pode retirar a atividade da memória simplesmente pedindo a ela que seja finalizada ou simplesmente matando o seu processo.
-



Google Android

Ciclo de Vida



Google Android

A classe android.app.Activity

```
public class Activity extends ApplicationContext {  
  
    protected void onCreate(Bundle savedInstanceState);  
  
    protected void onStart();  
  
    protected void onRestart();  
  
    protected void onResume();  
  
    protected void onPause();  
  
    protected void onStop();  
  
    protected void onDestroy();  
  
}
```



Google Android

Method	Description	Killable?	Next
onCreate()	Chamado quando a atividade é criada. Aqui é quando você deve fazer todas as funções como: criar as views, linkar os dados às listas, etc.	Não	onStart()
onRestart()	Chamado após a atividade ser parada e antes de ser reiniciada. Sempre seguida por onStart()	Não	onStart()
onStart()	Chamado quando a atividade se torna visível ao usuário. Seguido pelo onResume() se a atividade roda na frente ou por onStop() se ela se torna invisível.	Não	onResume() or onStop()
onResume()	Chamado quando a atividade vai iniciar a interação com o usuário. Nesse ponto, sua atividade está no topo da pilha de atividades e quaisquer dados que sejam inseridos serão feitos aqui. Sempre seguido pelo onPause().	Não	onPause()
onPause()	Chamado quando o sistema está por resumir a atividade anterior. Tipicamente usado para persistir quaisquer mudanças ainda não efetivadas, parar animações e outras coisas que possam consumir a CPU, etc. Seguido por onResume() se a atividade retornar para a frente ou onStop() se ela se tornar invisível ao usuário.	Sim	onResume() or onStop()
onStop()	Chamado quando a atividade não mais estiver visível ao usuário, pois outra atividade foi resumida e está na frente desta. Isso pode acontecer porque outra atividade está sendo iniciada. Seguida pelo onRestart() se essa atividade está voltando para interagir com o usuário ou onDestroy() se a atividade estiver sendo encerrada.	Sim	onRestart() or onDestroy()
onDestroy()	A chamada final que você receberá antes que a atividade seja destruída ou finalizada. Isso pode acontecer porque a atividade está, de fato, sendo encerrada (alguém chamou finish() nela) ou porque o sistema está temporariamente destruindo a instância da atividade para aumentar o espaço na memória.	Sim	nothing



Exemplo Ciclo de Vida

- ✓ LifecycleActivity
 - ✓ onPause()
 - ✓ onResume()



LifeCycle Project



Google Android

Context

Interface que provê informações globais sobre o ambiente do aplicativo.

```
getApplicationContext();
```



Google Android



Context



//Obtendo Recursos da Aplicação

```
Context contexto = getApplicationContext();
```

```
String nomeAplicacao = contexto.getResources().getString(R.string.app_name);
```



Google Android

Application Context x Activity Context

- ✓ Application Context viverá enquanto a aplicação estiver viva, não há dependências do Ciclo de Vida da Activity;
- ✓ Não mantenha longas referências a contextos de activities, a referência deve durar apenas o seu próprio Ciclo de Vida;
- ✓ É sempre recomendado o uso de Application Context para evitar memory leaks.

<http://android-developers.blogspot.com.br/2009/01/avoiding-memory-leaks.html>

