

Introdução ao desenvolvimento Android

Felipe Ribeiro Rodrigues Magalhães

felipe_rrm@hotmail.com

felipe.rrm@gmail.com

<https://github.com/FelipeRRM>

Por que desenvolver para Android?

► Abrangência:

Period	Android	iOS	Windows Phone	BlackBerry OS	Others
2015Q2	82.8%	13.9%	2.6%	0.3%	0.4%

Fonte: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>

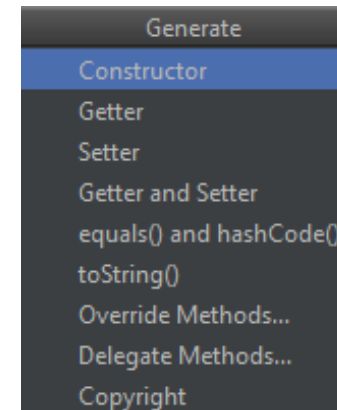
► Custo:

iOS : 100 dólares por ano + um Mac para desenvolver (R\$4000?)

Android: Taxa única de 25 dólares (pode desenvolver em Windows, MacOS e Linux)

Android Studio

- ▶ IDE baseada no IntelliJ IDEA Community Edition
- ▶ Alguns atalhos:
 - ▶ Auto Complete: Ctrl + Barra de Espaço
 - ▶ Menuzinho Bacana: Alt + Insert
 - ▶ Pesquisar por utilização de um método: Ctrl + F7
 - ▶ Verificar parâmetros esperados por um método: Ctrl + P



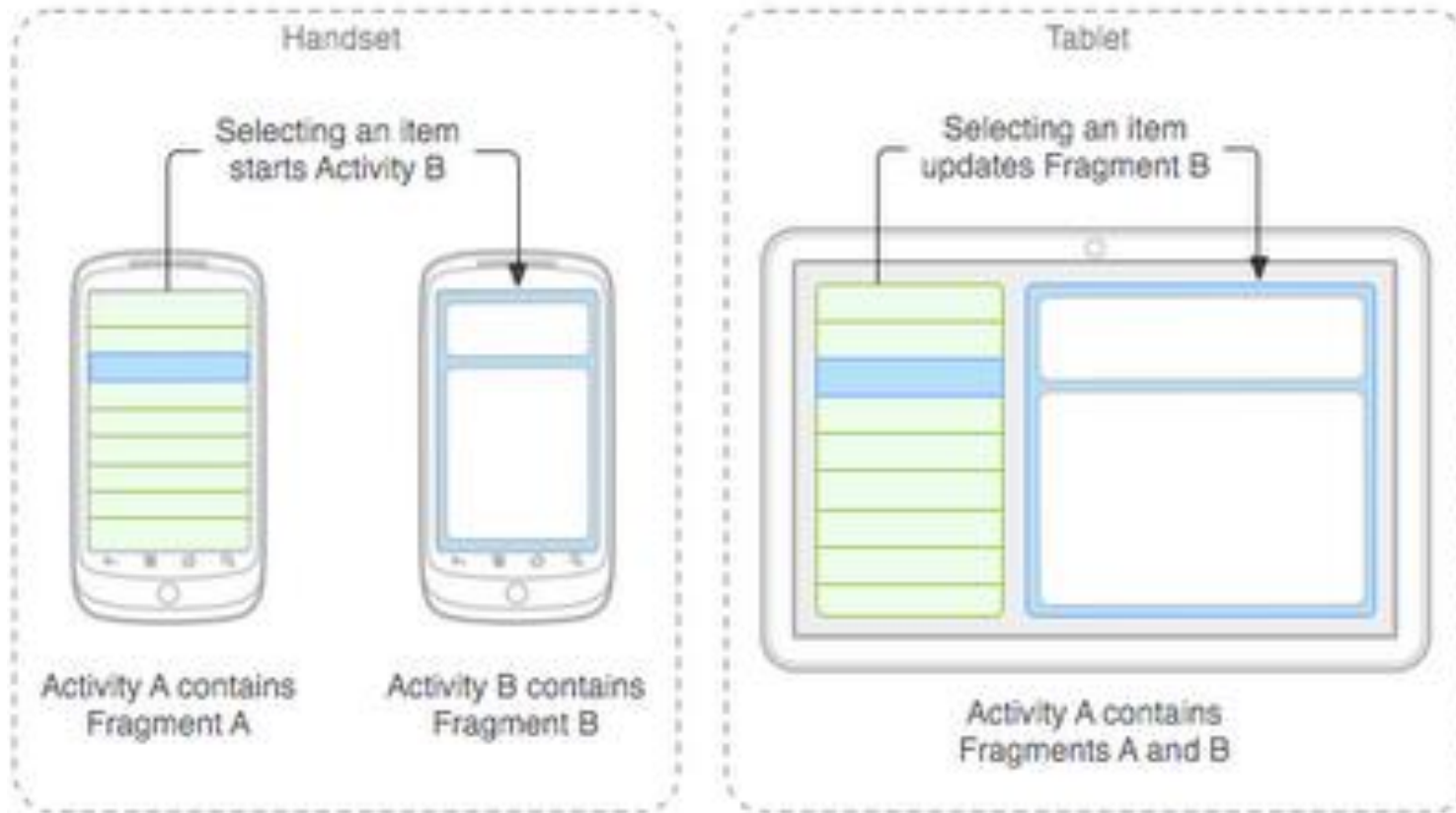
Qual versão escolher?

Version	Codename	API	Distribution
2.2	Froyo	8	0.2%
2.3.3 - 2.3.7	Gingerbread	10	3.8%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	3.4%
4.1.x	Jelly Bean	16	11.4%
4.2.x		17	14.5%
4.3		18	4.3%
4.4	KitKat	19	38.9%
5.0	Lollipop	21	15.6%
5.1		22	7.9%

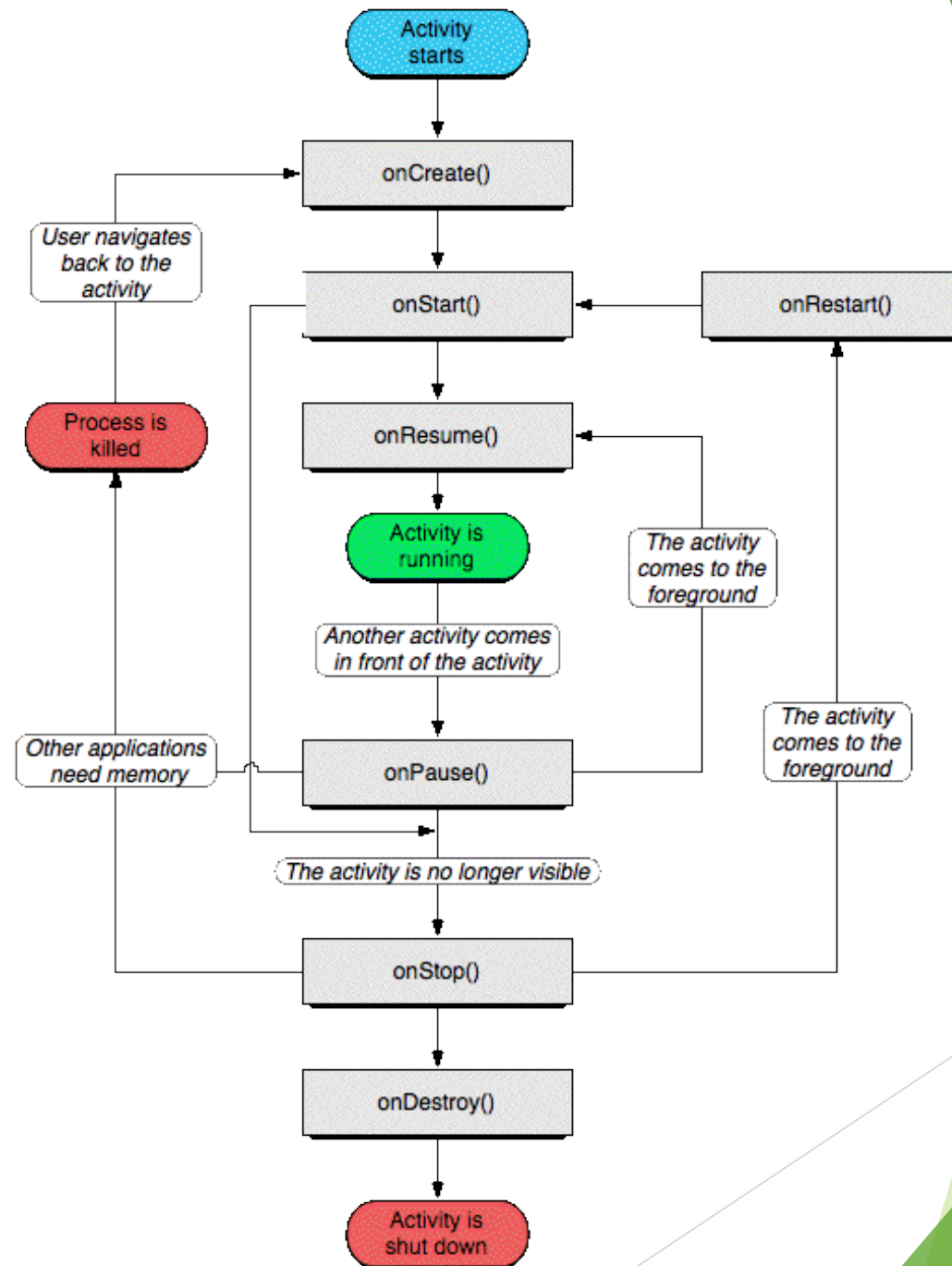
Estrutura do Projeto

- ▶ Pasta java: Código fonte em java (duh?) do projeto
- ▶ Pasta res: Arquivos diversos utilizados no projeto vão nessa pasta, alguns principais:
 - ▶ res\drawable: Arquivos de imagens (pode ser png, jpg, etc ou arquivos XML)
 - ▶ res\layout: Arquivos XML de layout utilizado pelos fragments e activities
 - ▶ res\values: Arquivos XML definindo strings, estilo e cores da aplicação
- ▶ AndroidManifest.xml: Arquivo que define quais são as activities do seu app, assim como as permissões de sistema necessárias.
- ▶ build.gradle: Contém diversas configurações do projeto e suas dependências.

Activity vs Fragment:

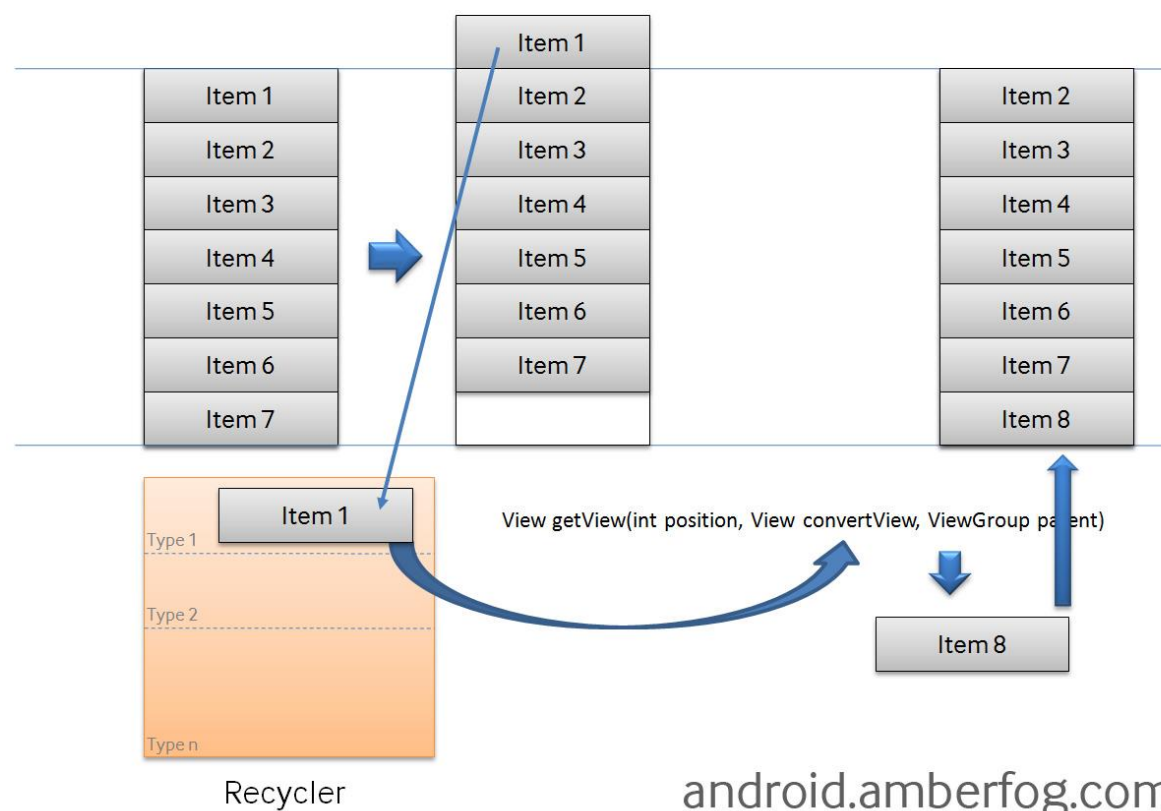


Activity Lifecycle:



Recycler View

- Melhor maneira de fazer listas no Android



Recycler View

Precisamos de:

- ▶ Um RecyclerView no layout da Activity, nome da View: `android.support.v7.widget.RecyclerView`
- ▶ Um conjunto de itens, sendo que cada item contém as informações de cada linha
- ▶ Um layout para representar como cada linha será exibida
- ▶ Um adapter (responsável por fazer a reciclagem de views)
- ▶ Um ViewHolder (mantém referência das Views do layout que criamos para cada linha)
- ▶ Um LayoutManager, que define como as diversas views serão exibidas no RecyclerView

Picasso

- Biblioteca muito boa para carregar imagens da rede de maneira eficiente:
compile 'com.squareup.picasso:picasso:2.5.2' nas dependências do build.gradle