

Universidade Federal do Paraná

Departamento de Informática

Ciência da Computação

Tópicos em Sistemas Embarcados

Android

Prof. Eduardo Todt

2016

1992



2008

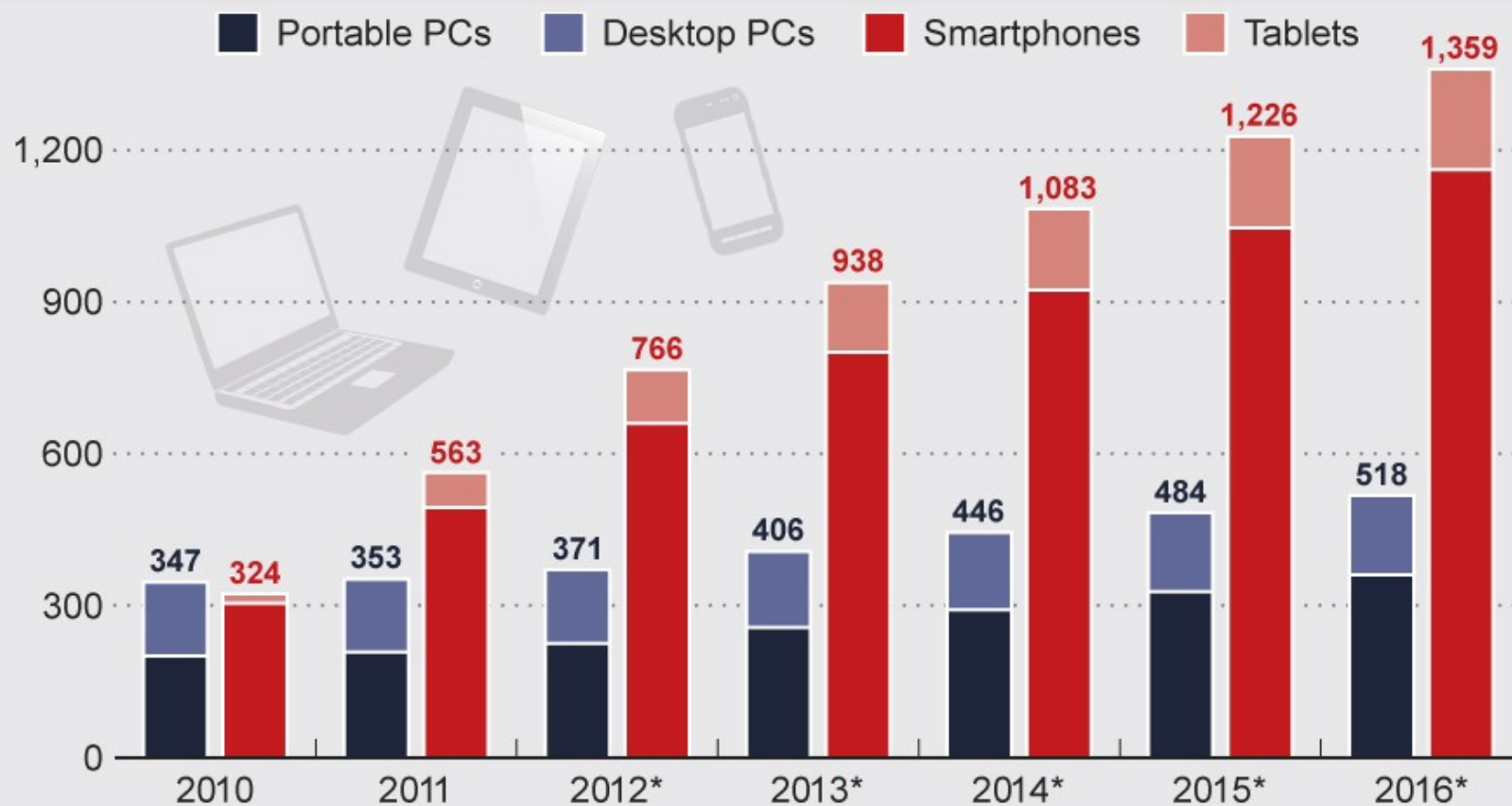


2016

# Produção

## The Post-PC Era Has Arrived

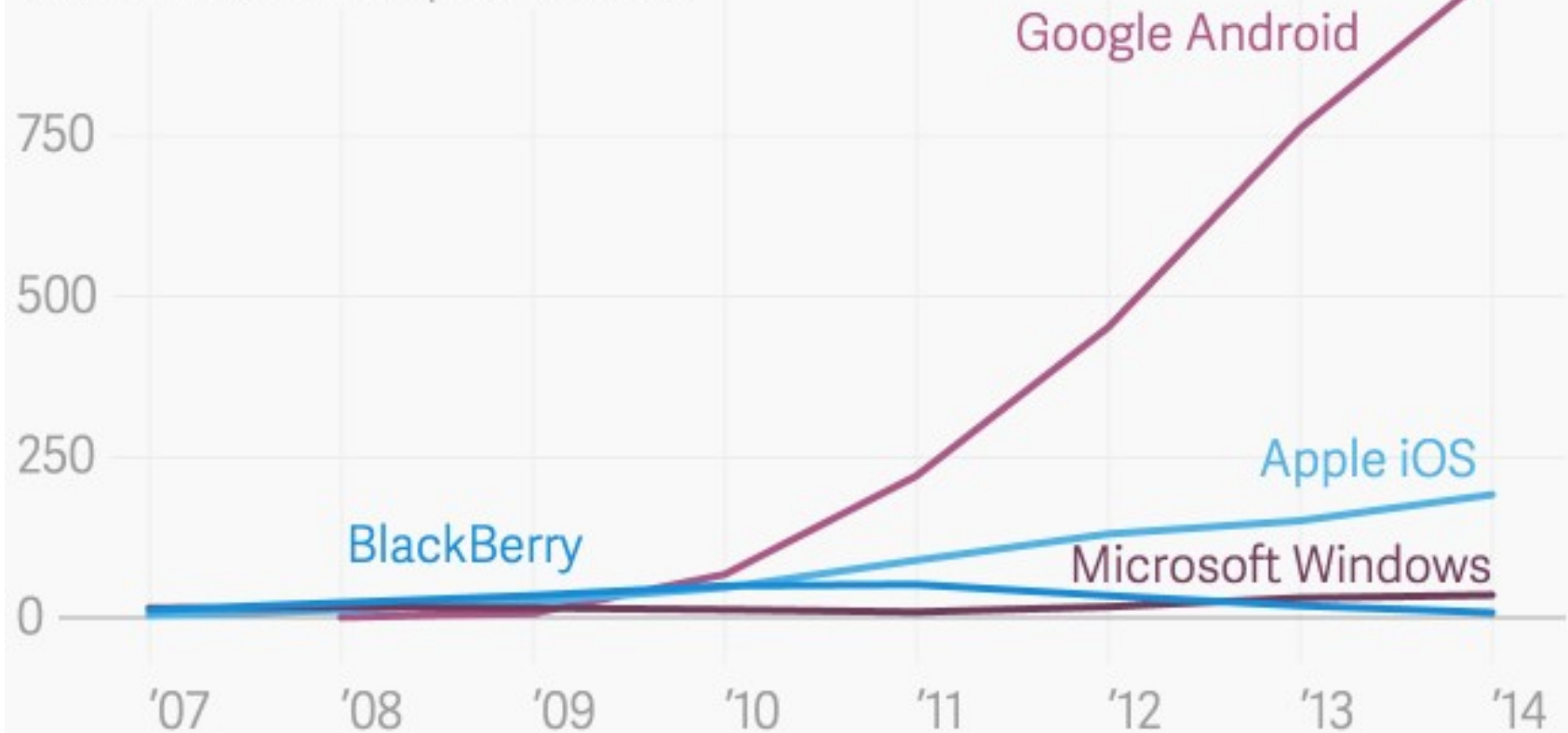
Global smartphone, tablet and PC shipments (in millions)



# Mercado

Android rules the world—by volume, at least

1,000 million smartphone sales



Quartz | qz.com

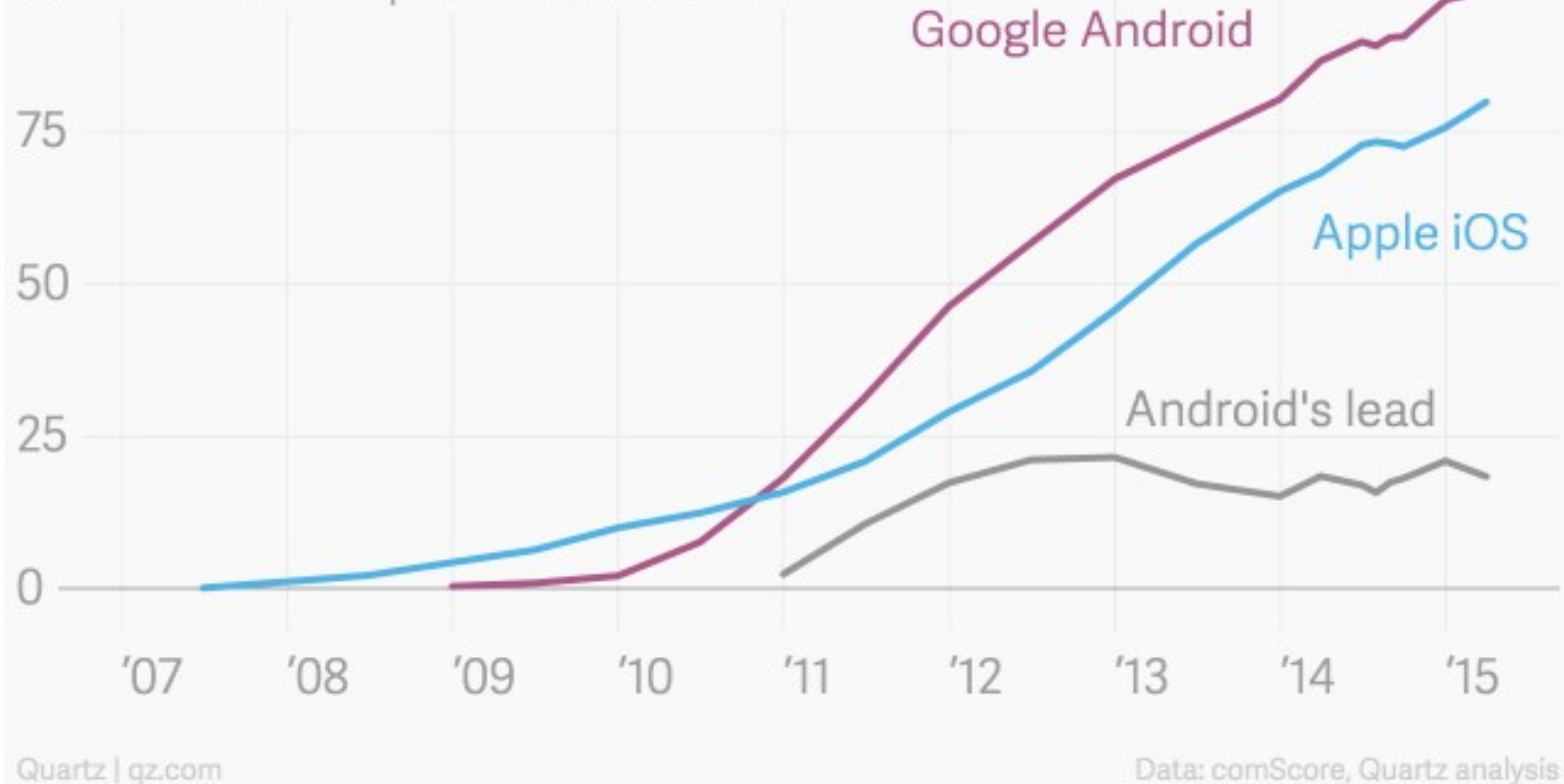
Data: Gartner

<http://qz.com/407776/>

# Mercado

Android is maintaining—but not increasing—its lead in the US

100 million smartphone subscribers



<http://qz.com/407776/>



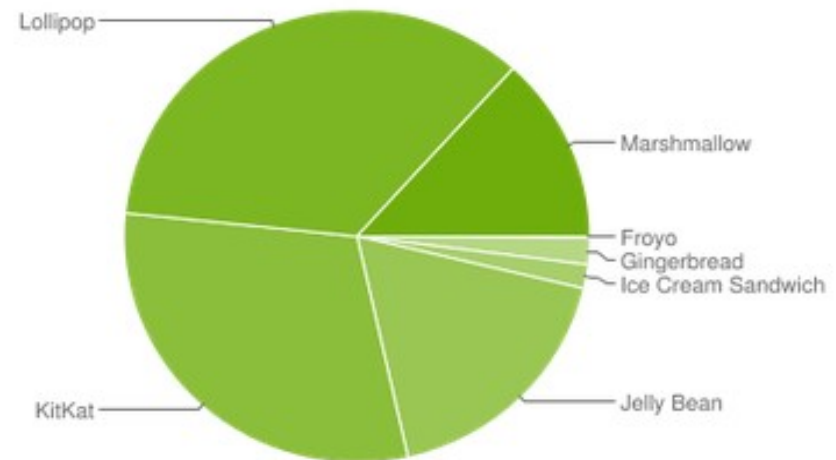
# Android vs iOS



<http://www.forbes.com/sites/chuckjones/2014/04/07>

# Versões

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.7%
4.1.x	Jelly Bean	16	6.4%
4.2.x		17	8.8%
4.3		18	2.6%
4.4	KitKat	19	30.1%
5.0	Lollipop	21	14.3%
5.1		22	20.8%
6.0	Marshmallow	23	13.3%



2008:  
Version BASE

*Data collected during a 7-day period ending on July 11, 2016.*

*Any versions with less than 0.1% distribution are not shown.*

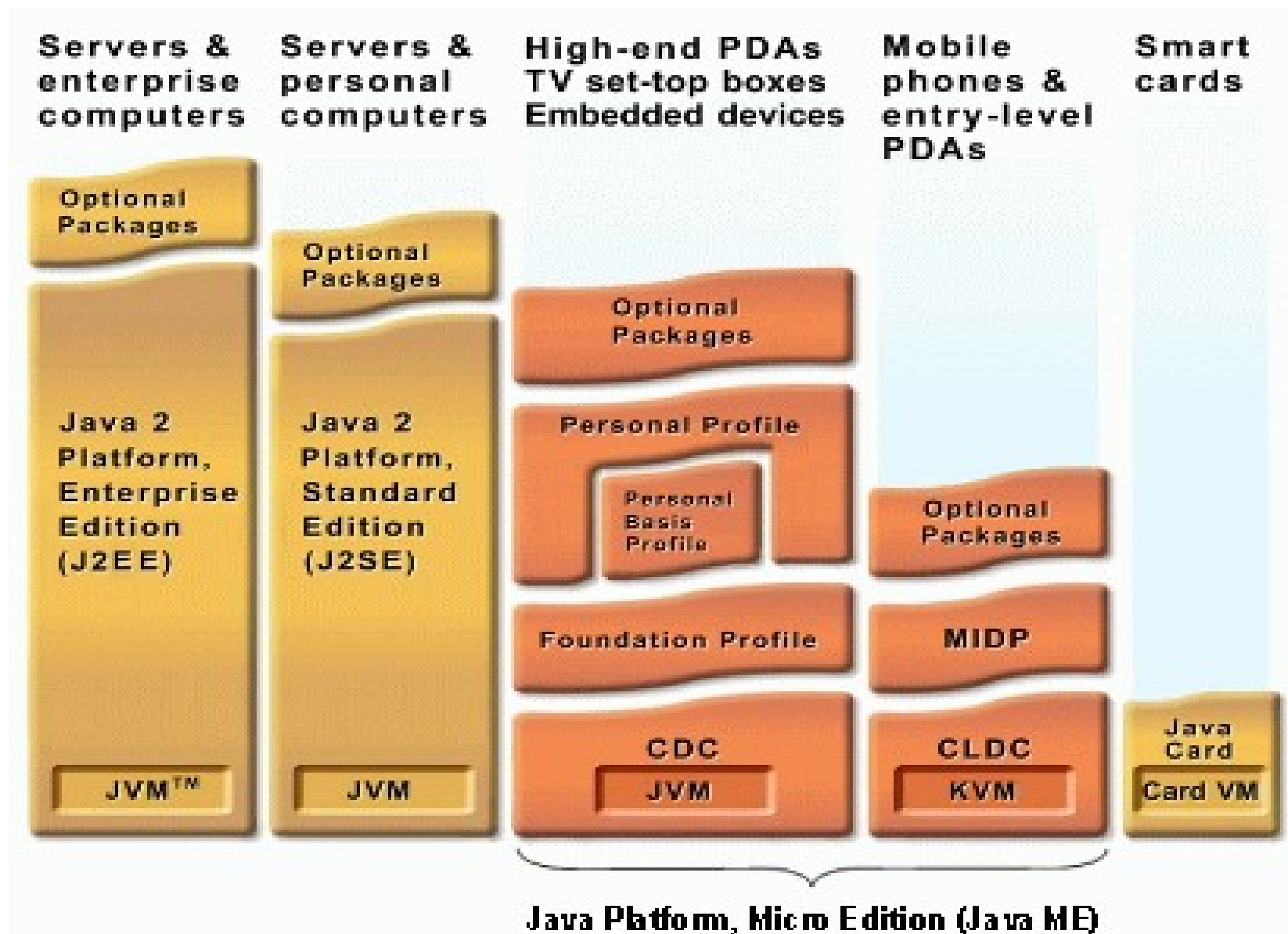
<https://developer.android.com/about/dashboards/index.html>

# JavaME

- parte da Java platform, Micro Edition (Java ME), antes Java 2 Platform ME (J2ME)
  - configuração =  
JRE (Java run-Time Environment) + Classes (core)
    - Connected Limited Device Configuration (CLDC)
  - Mobile Information Device Profile (MIDP)
    - i/o, GUI, LCDUI, storage, messaging, personal info management, file connection, ...
- JavaME é uma plataforma baseada em Java para dispositivos móveis, é uma VM. Android é uma plataforma completa.



# Plataforma Java

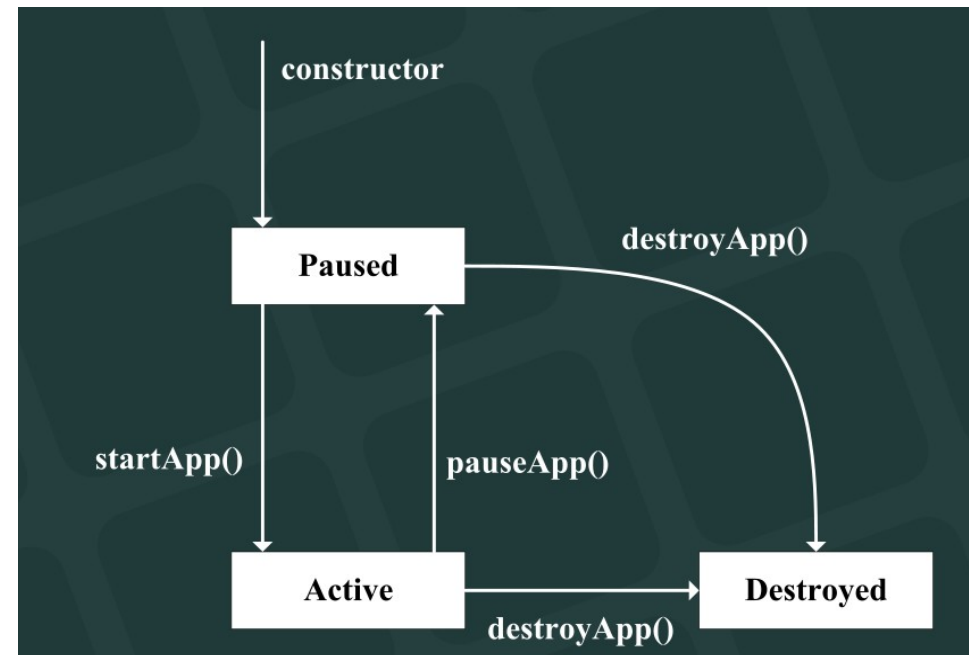


# MIDlet

- aplicações MIDP são MIDlets
- todo MIDlet é uma instância de `javax.microedition.midlet.MIDlet`
  - sem argumento construtor
  - implementa métodos lifecycle
  - pode ser downloaded
  - executada no host

# MIDlet: Ciclo de vida

- Application Manager controla instalação e execução de MIDlets
  - Start: constructor + startApp
- MIDlet
  - se coloca em pausa notifyPaused()
  - se destrói notifyDestroyed()
  - resumeRequest()
- AppManager
  - Pode PauseApp() e destroyApp()



# Package em JavaME

- MIDlets + Classes + Ressources + Manifest Information => Java Archive (jar); no Android corresponde ao .apk
  - Manifest: descreve conteúdo (versões do CLDC, MIDP, name, version, vendor)
- Java Application Descriptor (jad)
  - descreve um ou mais MIDlets Java
    - MIDlet-<n>, id do MIDlet no arquivo jad
    - MIDlet-Name, nome do MIDlet
    - MIDlet-Jar-URL, de onde pode ser instalada
    - MIDlet-Jar-Size, tamanho do download
    - MIDlet-Version
  - ainda, mesma informação do manifest

# Instalação MIDlet suite (JavaME)

- Direct
  - cabo, IR, bluetooth
- Over-The-air (OTA)
  - The client device sends an HTTP GET request to the server for the given URL.
  - The server sends an HTTP response with the suite's JAD file as the message body.
  - The client verifies the HTTP response and extracts the suite's MIDlet-Jar-File and MIDlet-Jar-Size attributes.
  - The device sends an HTTP GET request for the JAR file.
  - The server sends an HTTP response with the JAR file as the message body.
  - The device verifies the message and the JAR file.
  - The device asks the user to verify installation
  -
- Remoção é simples, MIDlet é uma entidade self contained
- Veja <http://www.oracle.com/technetwork/systems/wtoolkit-155632.html>

# Mas o Android?

- Compatibility Definition Document – CDD
- Compatibility Test Suite - CTS
- Linux kernel
- Bibliotecas C/C++, como SQLite, WebKit, OpenGL, media manager
- Run time para executar aplicações Android: Dalvik Virtual Machine e bibliotecas core
- Application framework, disponibilizando os serviços para camada de aplicação (window manager, location manager, databases, telephony, sensors)
- User interface framework
- Aplicações core pré instaladas
- SDK



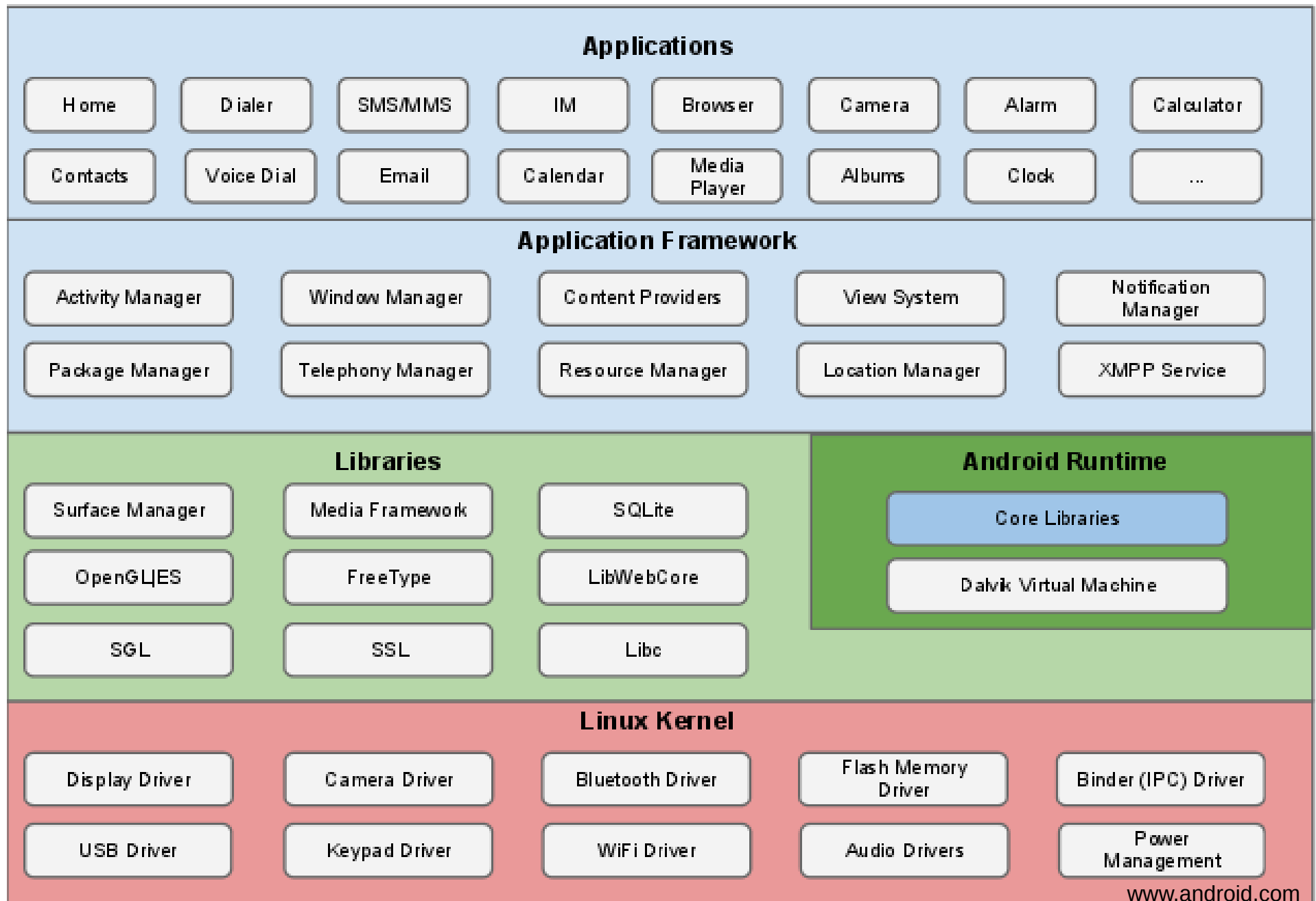
# Aplicações Nativas Android

- Parte do Android Open Source Project (AOSP)
  - Cliente email
  - SMS
  - PIM (personal information management): calendario e contatos
  - WebKit
  - Music player e picture gallery
  - Camera e video
  - Calculadora
  - Tela home
  - Relógio com alarme

# API Android

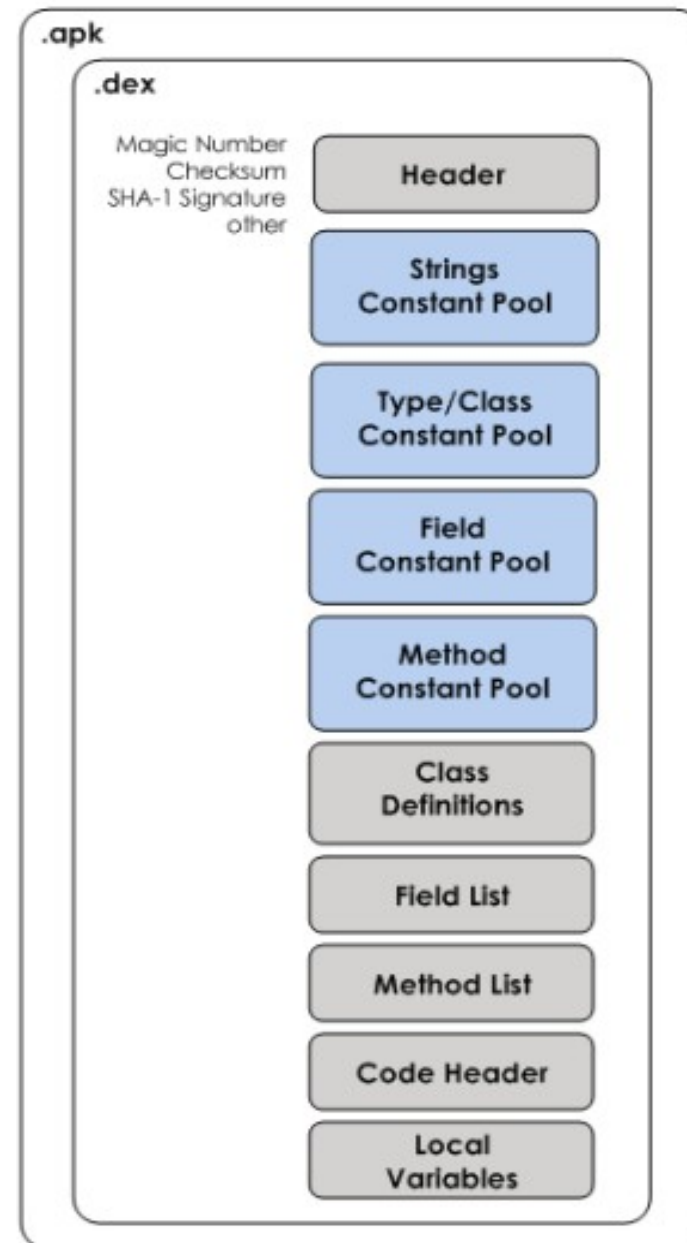
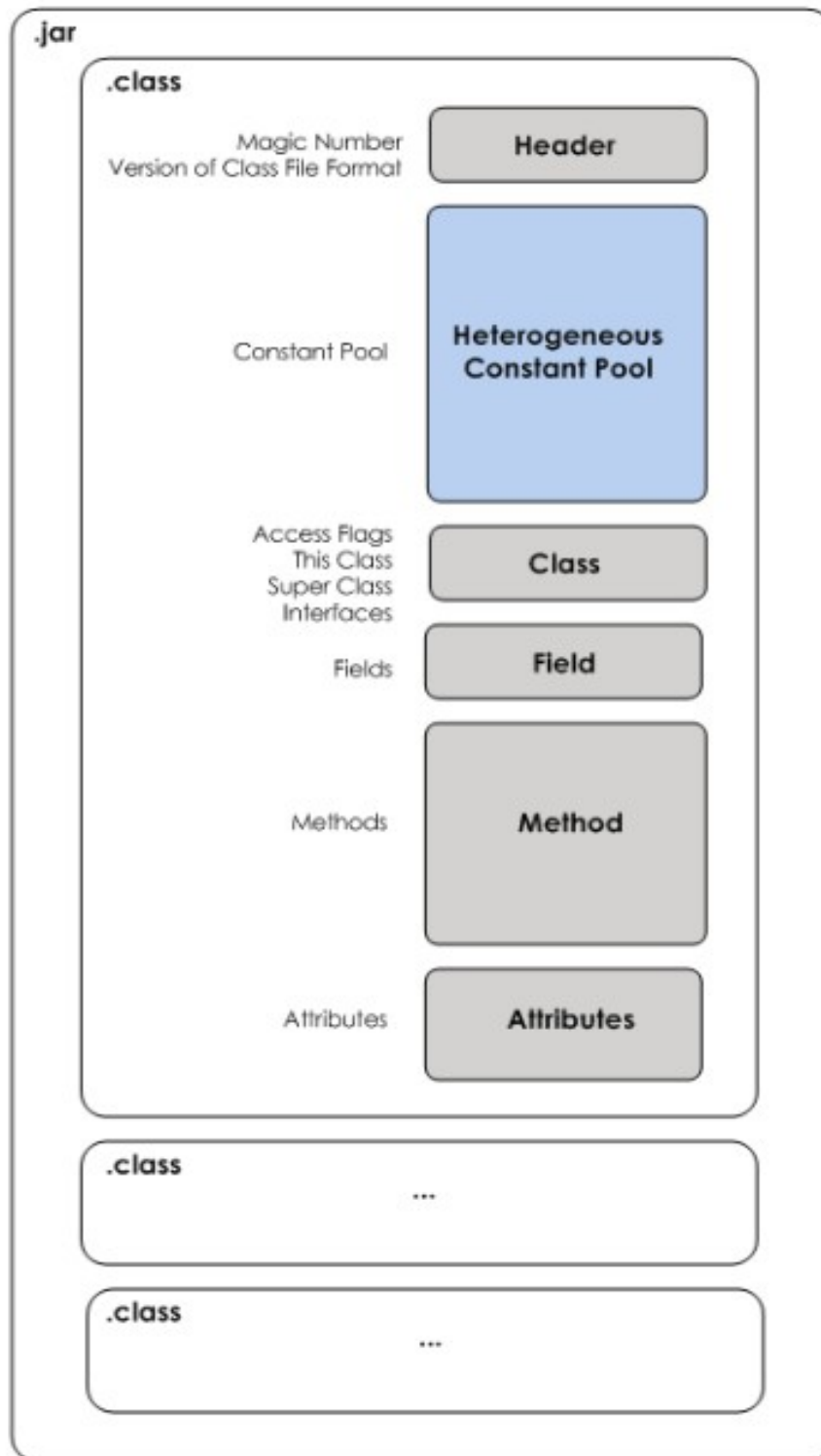
- GSM, EDGE, 3G, 4G, LTE
- Location-based services e mapas
- Wi-fi e peer-to-peer
- Multimidia, camera e microfone
- Audio, video
- Sensores: acelerometro, bussolas, barômetros
- Bluetooth, NFC
- IPC message passing
- Shared data, contatos, redes sociais, agenda
- Background services, aplicações e processos
- Home screen widgets
- HTML 5 WebKit
- Gráficos 2D e 3D, OpenGL
- Application framework orientado à reutilização de componentes

# Android Stack



# DALVIK

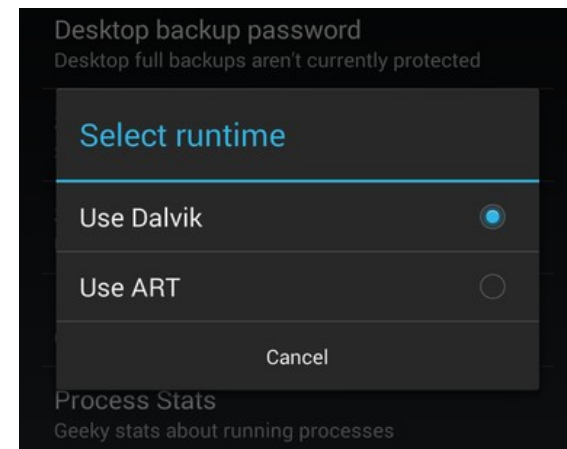
- Executa aplicações no Android
- Java (.java) → bytecode JVM (.class) → Dalvik bytecode (dex – Dalvik EXecutable)
  - .class um por classe; .dex contém múltiplas classes
  - .dex otimizado para reduzir memory footprint, por ex., constant pool compartilhada entre classes reduz até 50% code size
- arquitetura baseada em registradores (+ rápida que stack embora maior código, não implica em mais carga computacional significativa)
- threading e low-level memory suportados pelo Linux kernel
- cada aplicação Android executa seu próprio processo, com sua própria instância da Dalvik
- Android 2.2. recebeu JIT (Just In-Time) compiler para aumentar desempenho



dex\n009\0

# ART

- Android Run Time
- opção no 4.4 (KitKat), default no 5.0 (Lollipop)
- AOT – ahead of time compilation
  - compila na instalação com dex2aot no dispositivo em tempo de instalação
  - utiliza arquivo dex de entrada
- profiler melhorado (Traceview com sampling)
- debugger e diagnóstico melhorado, com maior informação de contexto em app crashes
- garbage collection melhorado
- app usa Java Native Interface (JNI) para executar código C/C++
- ELF Executable and Linkable File (padrão Unix)



Settings > dev options

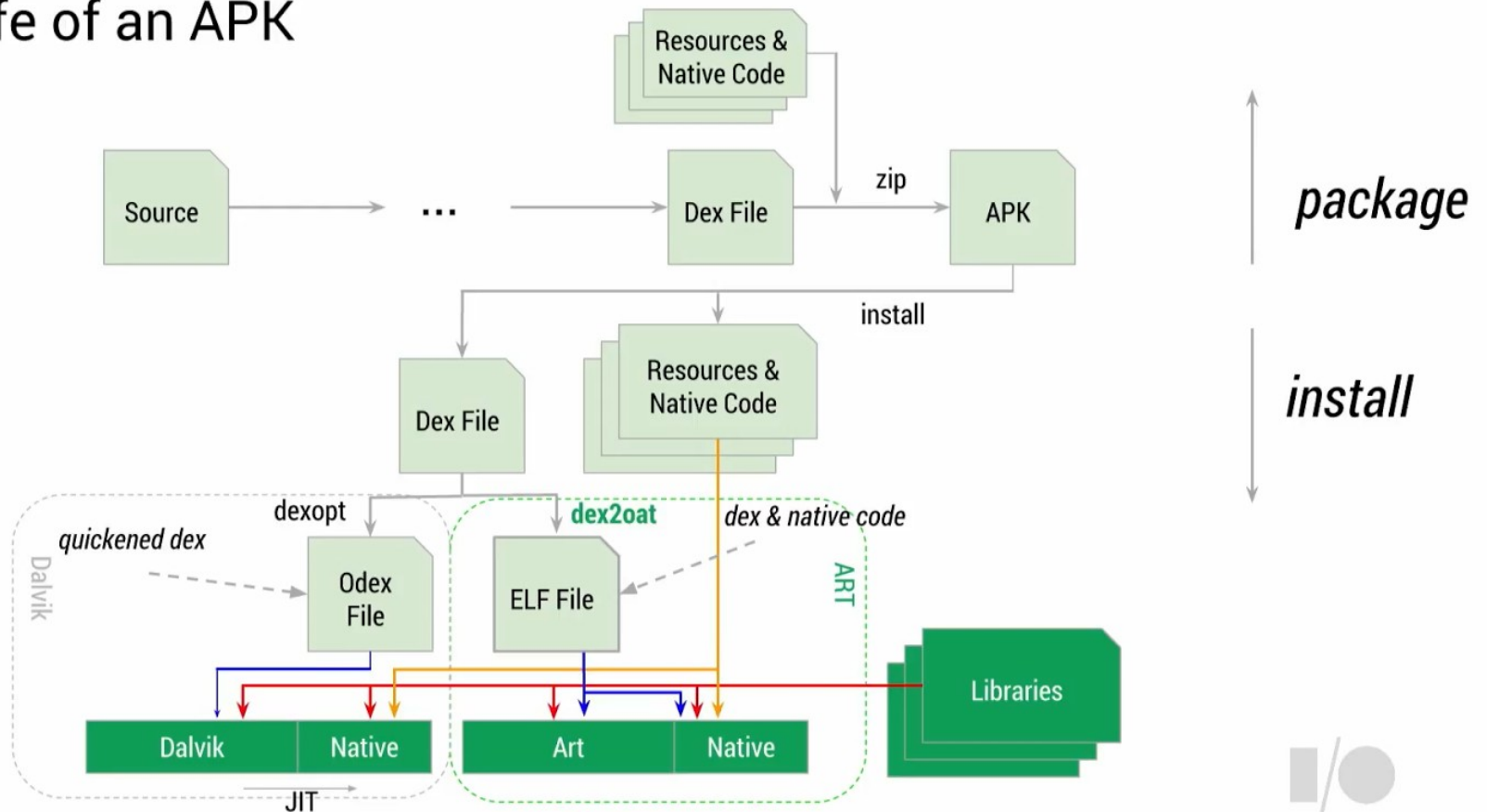


# ART e Dalvik

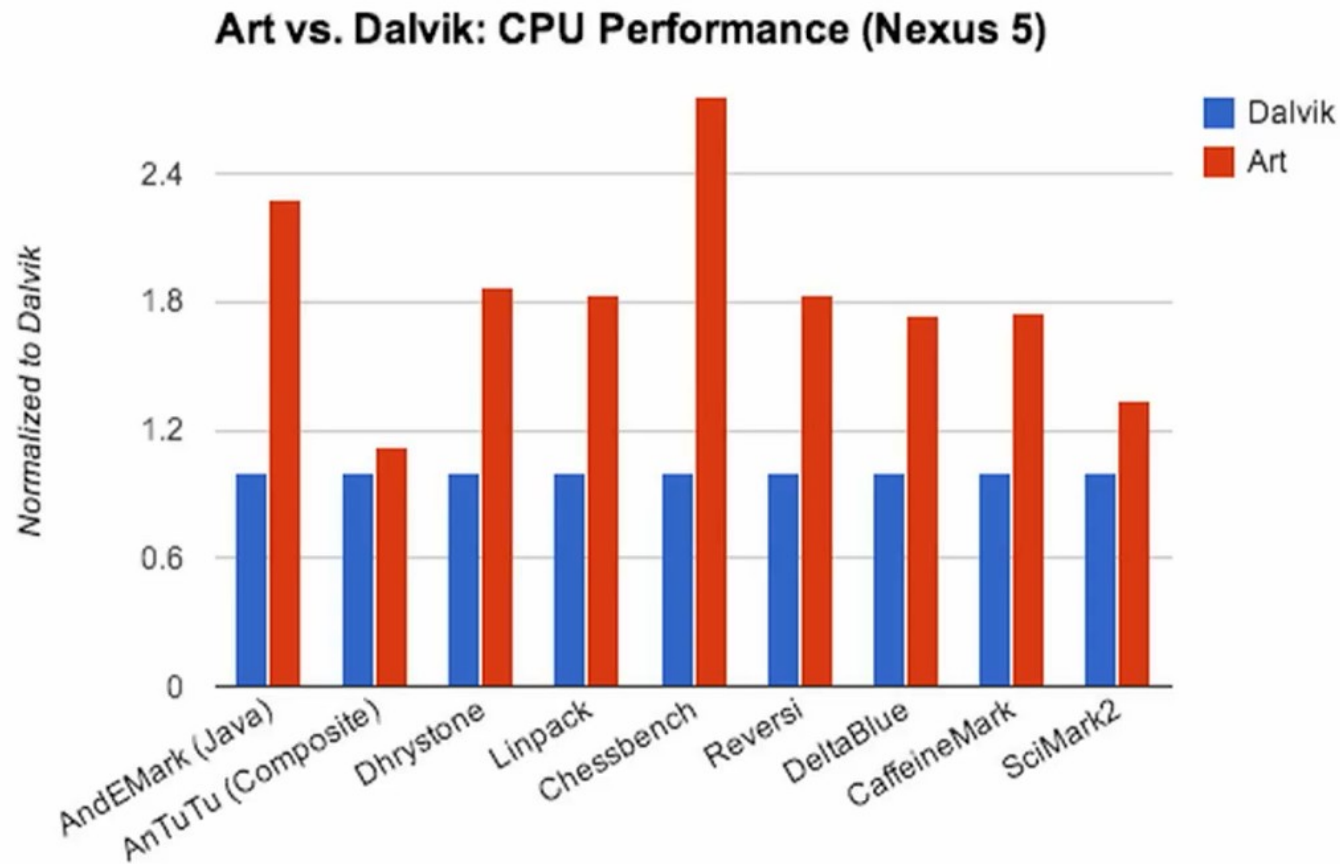
- ART e Dalvik são runtimes compatíveis executando Dex bytecode
- Apps desenvolvidas para Dalvik devem funcionar em ART
- Mas, algumas técnicas que funcionam em Dalvik não funcionam em ART ...
- Dalvik: Just In-Time (JIT)
- ART: Ahead-of-Time (AOT)

# ART e Dalvik

## The life of an APK



# Performance Boosting Thing, realized



# C/C++

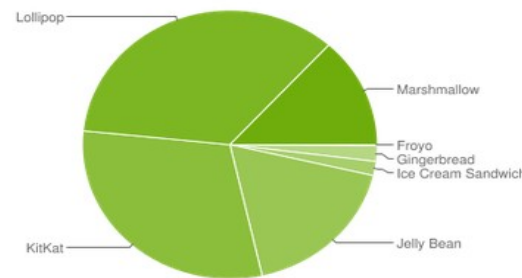
- Native Development Kit (NDK)
  - pode criar bibliotecas
  - <https://developer.android.com/tools/sdk/ndk>
  - pouca influência no desempenho, grande na complexidade ...
- Gradualmente incorporado no Android Studio

# O tema do API level

- Selecionando a versão mínima do SDK (API level) implica em selecionar o nível de backward compatibility que será suportado

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.7%
4.1.x	Jelly Bean	16	6.4%
4.2.x		17	8.8%
4.3		18	2.6%
4.4	KitKat	19	30.1%
5.0	Lollipop	21	14.3%
5.1		22	20.8%
6.0	Marshmallow	23	13.3%

*Data collected during a 7-day period ending on July 11, 2016.  
Any versions with less than 0.1% distribution are not shown.*



# Support Library Package

- Bibliotecas estáticas que habilitam uso de características recentes em dispositivos antigos (API level 4 e superiores)
- Cuidado com nomes iguais na *framework library* e na *support library*
- Em *developer.android.com* é recomendado o uso de support library
- <http://developer.android.com/tools/support-library/index.html>



# Limitações pelo hw

- Capacidade de processamento
- RAM limitada
- Armazenamento limitado
- Telas variadas, pequenas e baixa resolução (...)
- Custo de transferência de dados
- Conectividade intermitente, taxas baixas, alta latência
- Energia restrita

# Componentes de uma App Android

## Activities

- Camada de apresentação
- Uma tela com interface usuário
  - Ex: em uma email app, activity para mostrar lista de emails, activity para escrever email, activity para ler email
- UI construída por extensões de Activity class
- Fragments e Views para visualização

# Componentes de uma App Android

## Services

- Executam sem UI, long term ou sem interação com usuário
- Atualizam dados
- Disparam notificações
- Broadcast Intents (objeto msg para solicitar ação de outro componente app, ex. `startActivity()` e `startService()`)

# Componentes de uma App Android

## Content providers

- Armazenamento de dados compartilhável
- Gerenciam e persistem
- Interagem com databases SQL
- Compartilhamento dados entre limites app
- Pode oferecer a outras app, pode acessar dados de outras app
- Ex. Media e contatos

# Componentes de uma App Android

## Intents

- Message passing framework
- Start e stop activities e services
- Broadcast messages system wide ou para uma activity, service ou broadcast receiver específico
- Requerer ação em um dado particular

# Componentes de uma App Android

## Broadcast receiver

- Ouvintes de Intents
- Habilitam escutar Intents que atendem critérios
- Permitem aplicação reagir a algum Intent recebido
- Permite criar app event-driven



# Componentes de uma App Android

## Widgets

- Para home screen
- Tipo particular de Broadcast Receiver

## Notifications

- Alerta usuário para eventos sem interromper Activity corrente

# Instalação

- Android studio
  - Baixar bundle completo do site developer
  - Descompactar, verificar que tem android-studio e android-sdk-linux (ex. Em /opt/Android)
  - Verificar atualizações (Android 6 e Android N)
  - Criar dispositivo virtual, embora em geral a depuração em dispositivo real é mais rápida
  - Para depurar em dispositivo real é necessário debugging over USB estar habilitado:
    - <https://developer.android.com/studio/run/device.html>

# Google test-lab

<https://firebase.google.com/docs/test-lab/>



# Referências

- Exemplos, listagens, conceitos, imagens, obtidos:
  - livro Peter Meier (Professional Android 4 Application Development)
  - [www.developer.android.com](http://www.developer.android.com)
  - fórum [www.stackoverflow.com](http://www.stackoverflow.com)
  - outras fontes, quando indicado.