

# **IF710**

# **Programação com Componentes**

**Leopoldo Teixeira**

[lmt@cin.ufpe.br](mailto:lmt@cin.ufpe.br) | @leopoldomt

# **Programação para Dispositivos Móveis: Android**

**Leopoldo Teixeira**

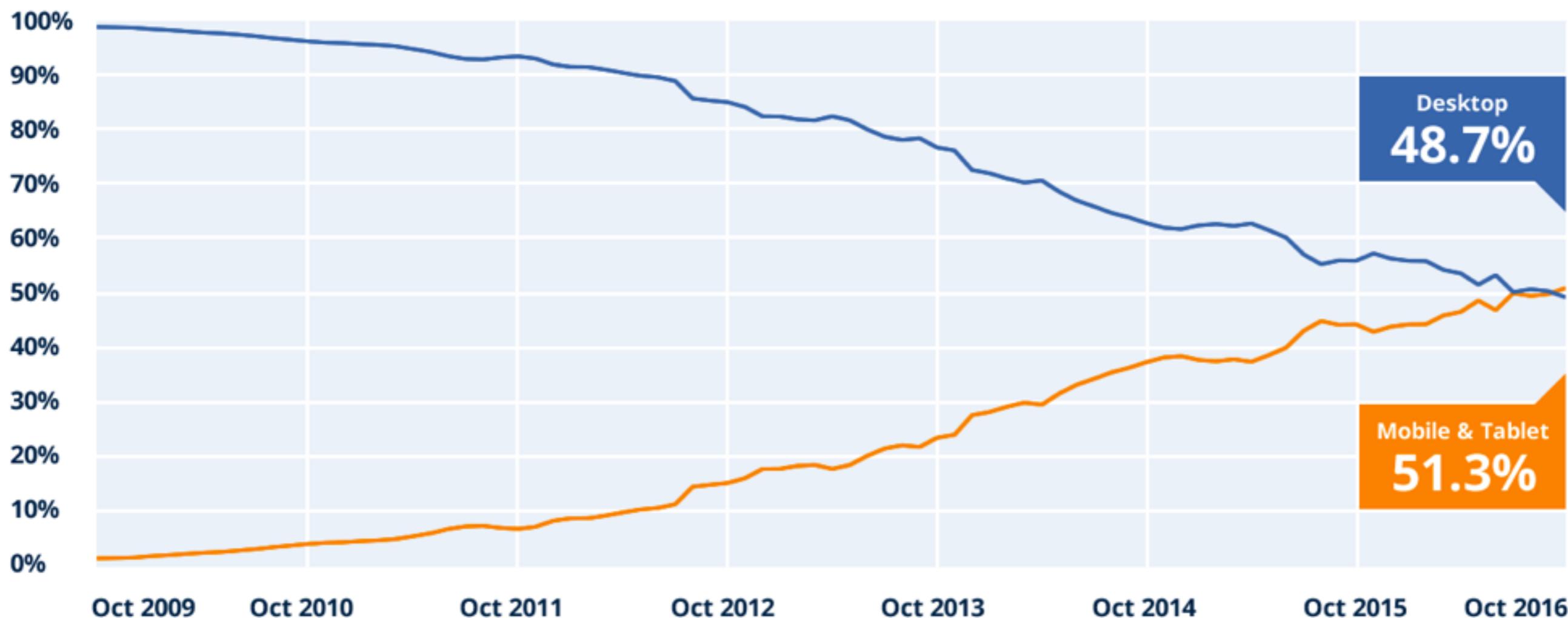
[lmt@cin.ufpe.br](mailto:lmt@cin.ufpe.br) | @leopoldomt



## Internet Usage Worldwide

October 2009 – October 2016

■ Desktop ■ Mobile & Tablet



## With which connected devices do people most often go online?

More often via smartphone than computer / tablet

[Hide](#)



59%

Equally via smartphone & computer / tablet

[Hide](#)



19%

More often via computer / tablet than smartphone

[Hide](#)



12%

Question asked:

How often do you use the Internet - through your web browser or apps - for personal reasons (i.e. all non business or work-related purposes) on a [computer / smartphone / tablet]?

Total Respondents: 550

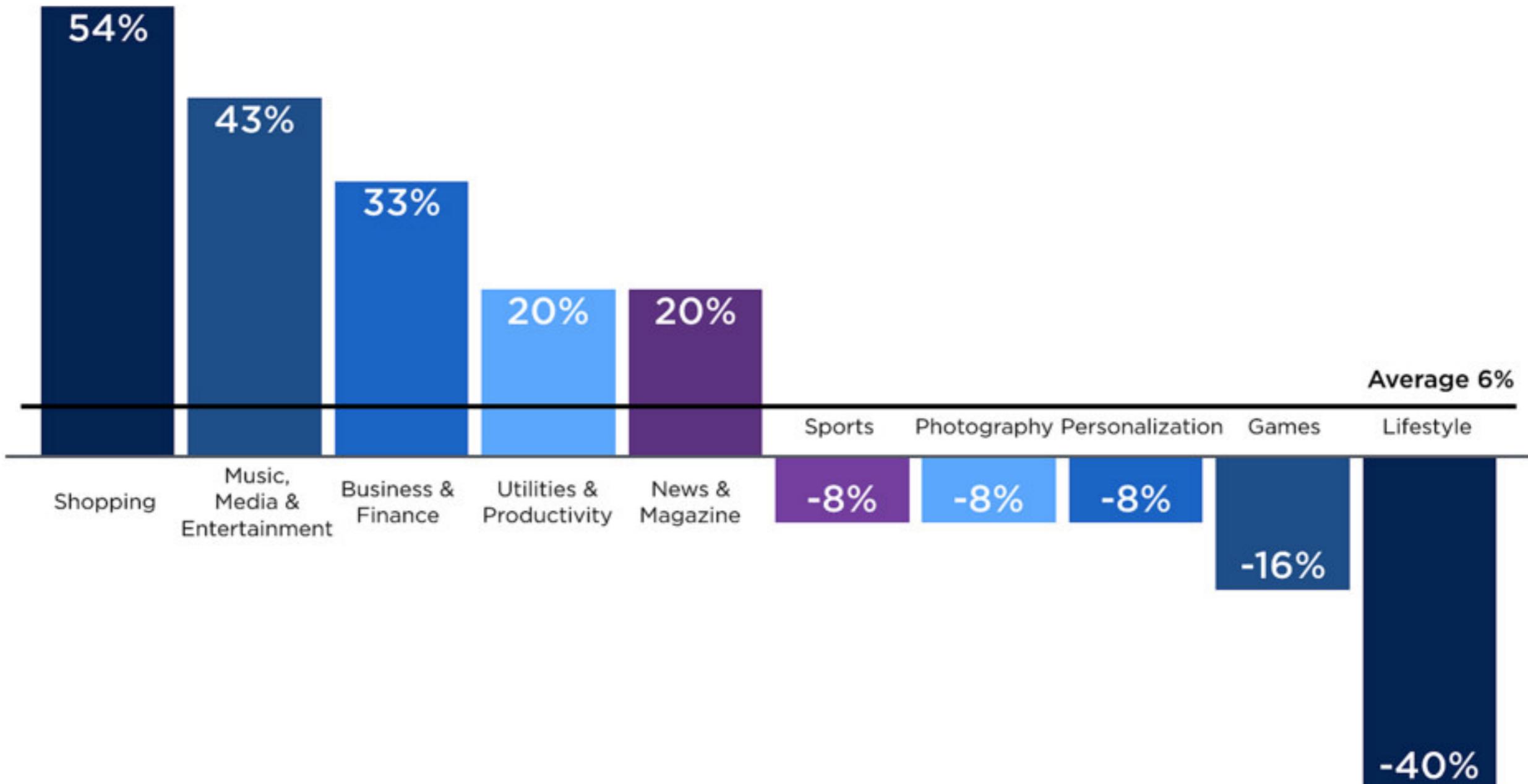
Base: Internet users (accessing via computer, tablet or smartphone)

Source: The Connected Consumer Survey 2016

See About section for more information on methodology.

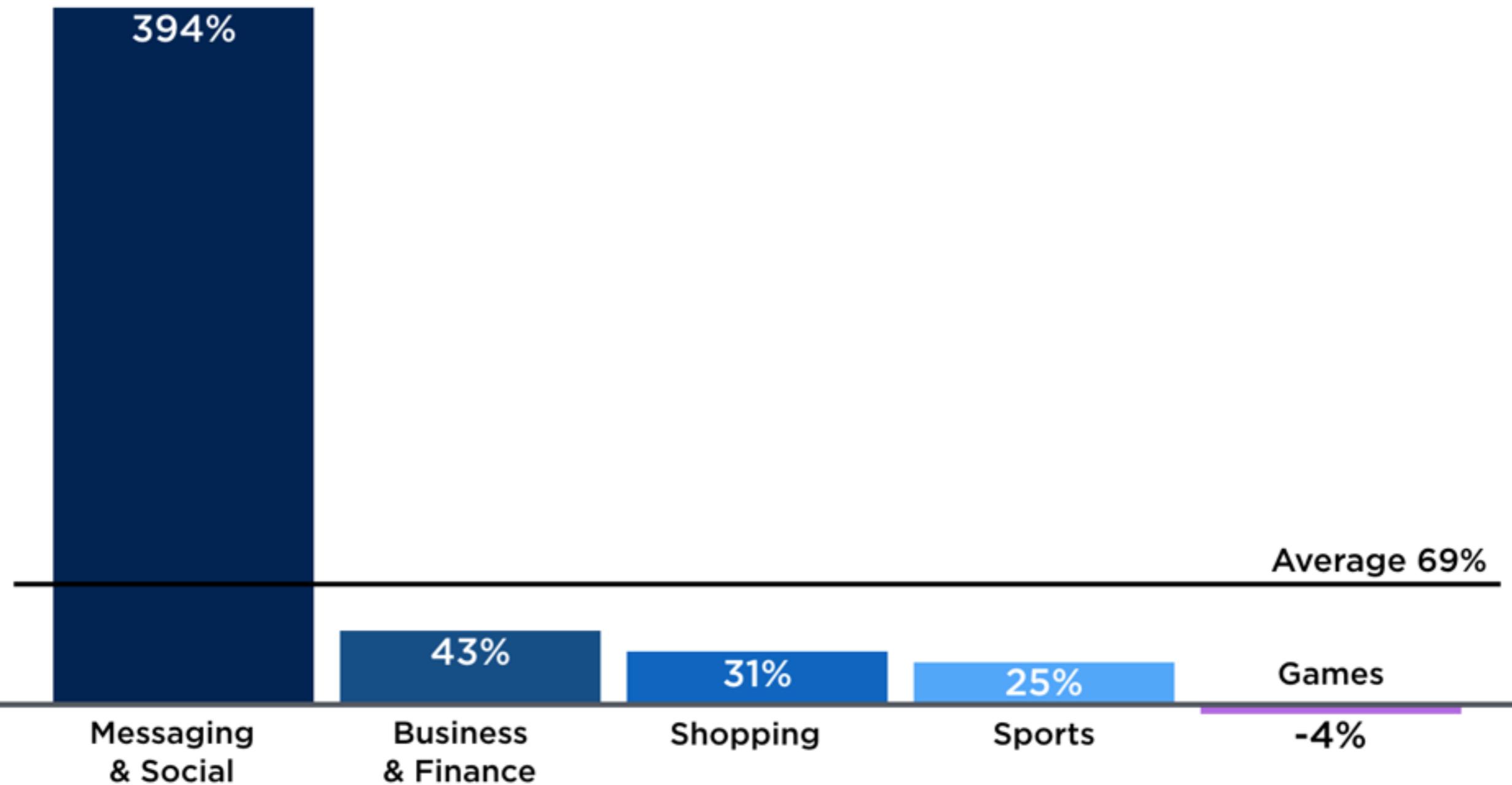
# Mobile App Usage Grows 6% Year-Over-Year

## Sessions



<http://flurrymobile.tumblr.com/post/169545749110/state-of-mobile-2017-mobile-stagnates>

# Mobile App Time Spent Grows 69% Year-Over-Year



<http://flurrymobile.tumblr.com/post/155761509355/on-their-tenth-anniversary-mobile-apps-start>

By MICHELLE CASTILLO / CBS NEWS / February 2, 2012, 5:41 PM

# Survey: 75 percent of Americans admit to using phone while in bathroom



Comment / [f Share](#) / [Tweet](#) / [Stumble](#) / [Email](#)



(CBS) Are you guilty of pulling out your phone during potty time? You're not alone: According to a new survey, about 75 percent of Americans admitted to using their mobile phones while in the bathroom.

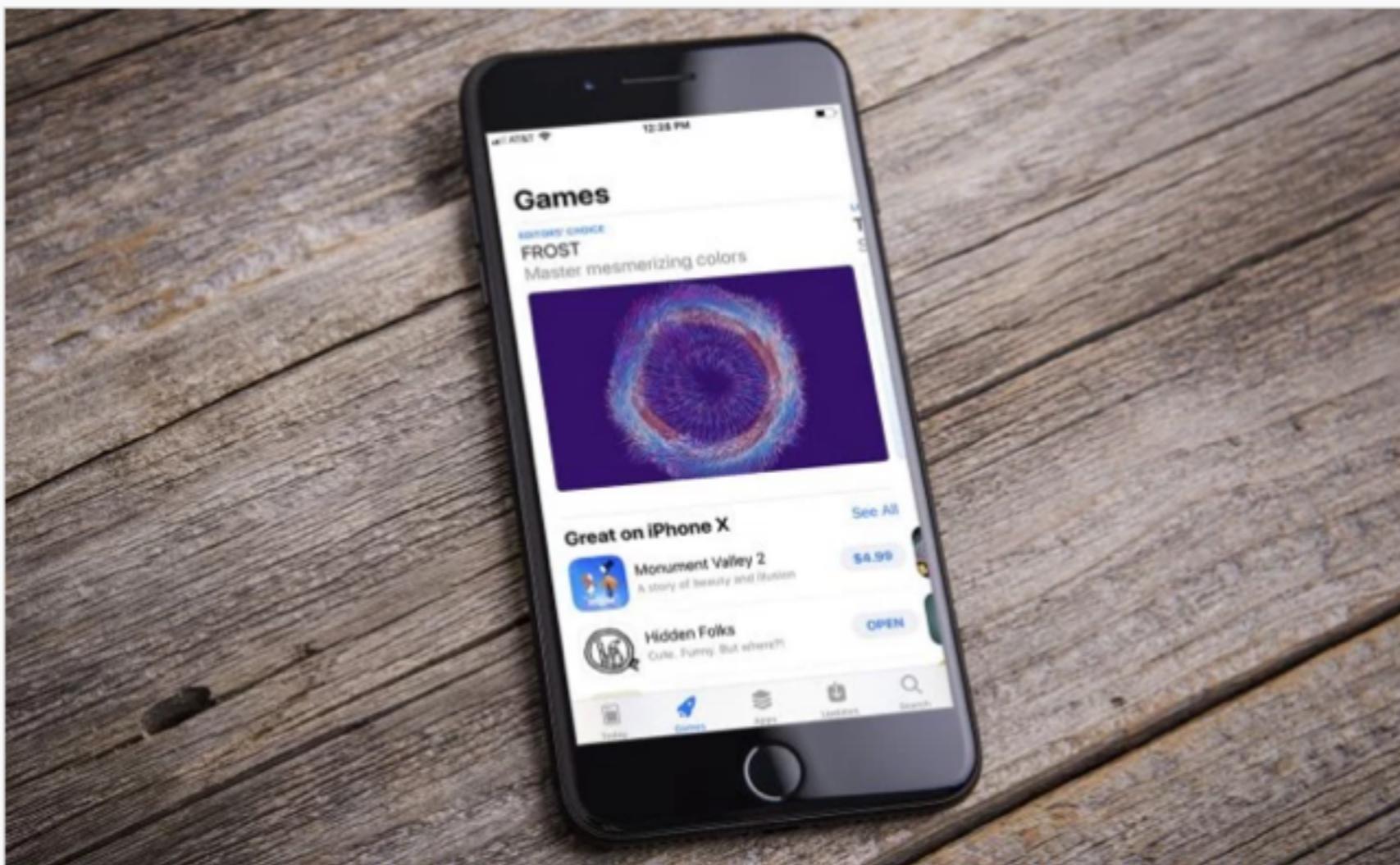
The survey, [called "IT in the Toilet,"](#) was conducted by marketing company 11Mark, who asked 1,000 Americans about their bathroom habits.

# App revenue climbed 35 percent to \$60 billion in 2017

Posted Jan 5, 2018 by **Sarah Perez** (@sarahintampa)



[Next Story](#)



Global app revenue climbed 35 percent in 2017 to reach nearly \$60 billion, according to a new [report](#) today from app intelligence firm Sensor Tower, which measured paid apps, subscriptions, and in-app purchases across both Apple's App Store and Google Play. However, Apple is the one pulling in the most revenue, the report found – at nearly double that of Google Play.

<https://techcrunch.com/2018/01/05/app-revenue-climbed-35-percent-to-60-billion-in-2017/>

**HOW EFFECTIVE ARE MOBILE SEARCHES?**

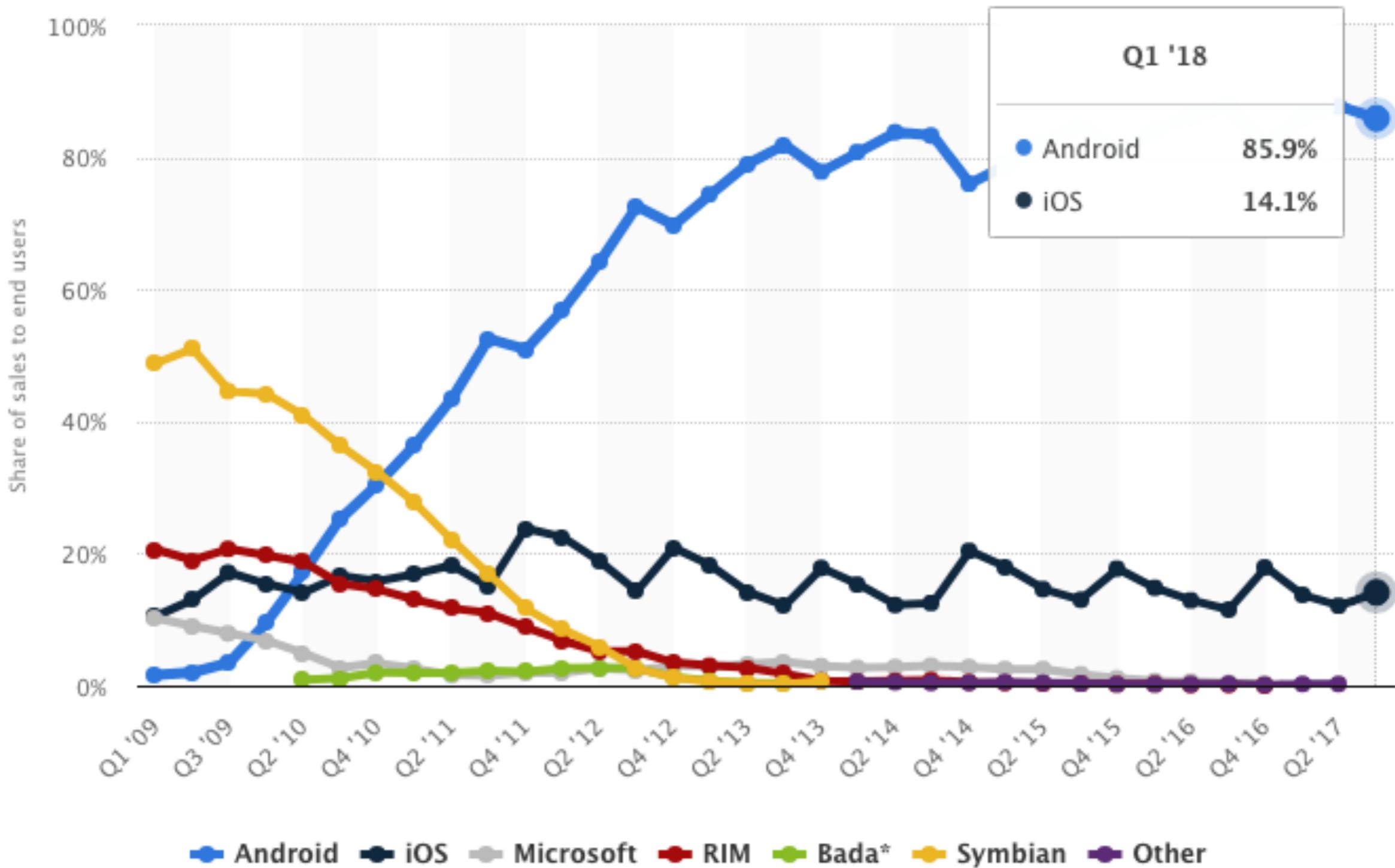
**70%**

**OF MOBILE SEARCHES LEAD TO  
ACTION ON WEBSITES WITHIN 1 HOUR**

# Mas, Android?

- ~1 bilhão de dispositivos no mercado
- ~81% do mercado de smartphones
- ~52% do mercado de dispositivos móveis
- ~1.5 milhão de ativações diariamente
- crescimento de download de apps: 50% (13-14)
- oportunidades para exercitar conceitos como segurança, privacidade, heterogeneidade

<http://expandedramblings.com/index.php/android-statistics>



# Number of Android Applications

Current number of Android apps on Google Play:

**3,038,609**



[View more »](#)

# Ratings of apps on Google Play

Total number of apps with a rating:

**1,686,532 (55.5%)**

Average number of stars for all apps:

**4.2** ★★★★☆

Number of apps with less than 3 ratings:

**1,352,077 (44.5%)**

# Most downloaded new apps

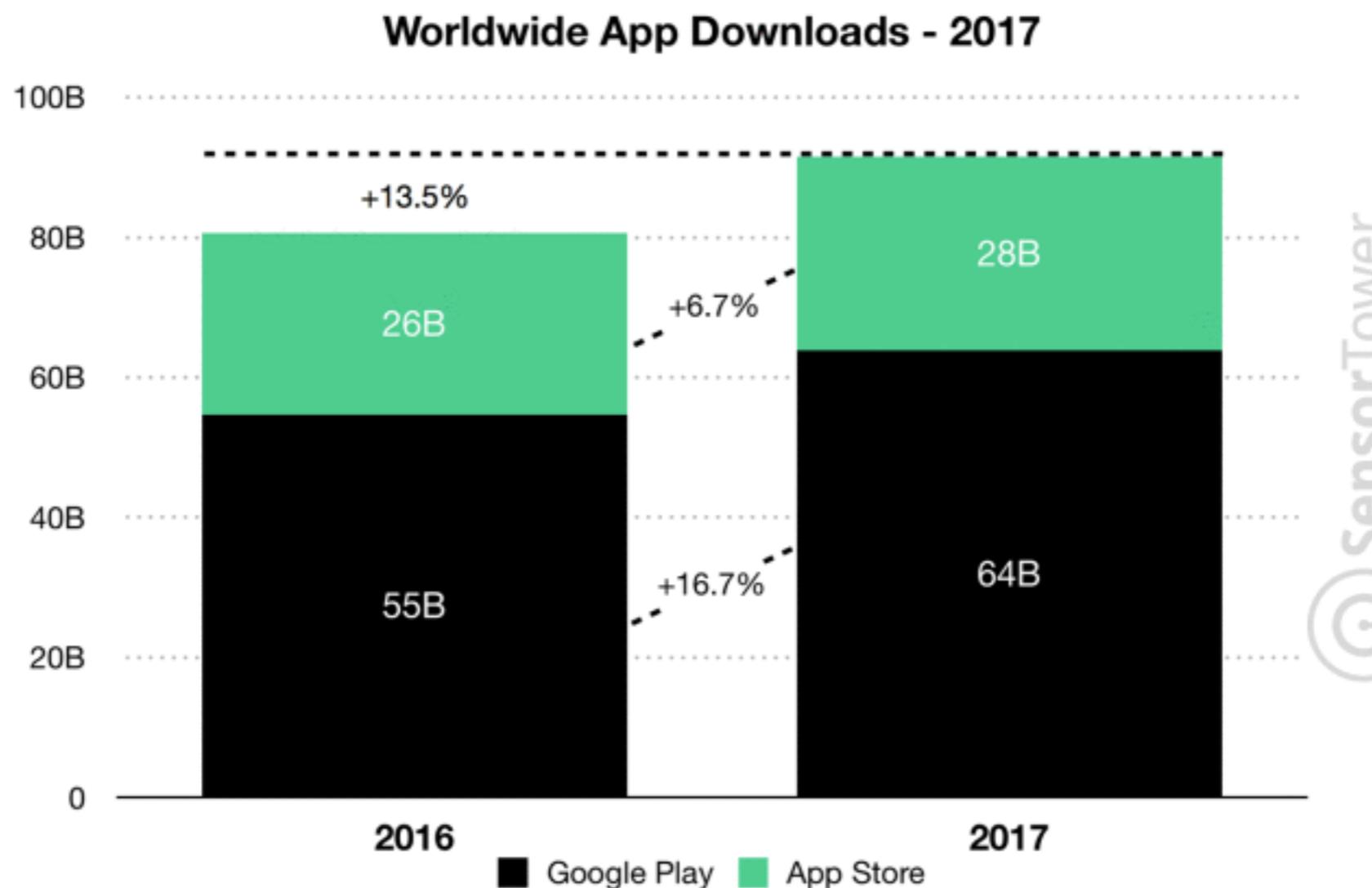
#	App	Rating	Installs
1	うたバス by KDDI株式会社	2.5	1M+
2	Sword Man - Monster Hunter by GMS Adventure	4.3	1M+
3	VIMAGE - cinemagraph creator & li... by vimage	4.7	500K+
4	Moto Face Unlock by Motorola Mobility LLC.	4.3	500K+
5	Werewolf Online by Philipp Eichhorn	4.7	500K+

[View more »](#)

# Latest Android SDK version

Android SDK version	Market share	Change in the last 30 days
8.0-8.1 (Oreo)	13.6 %	↑ 25%

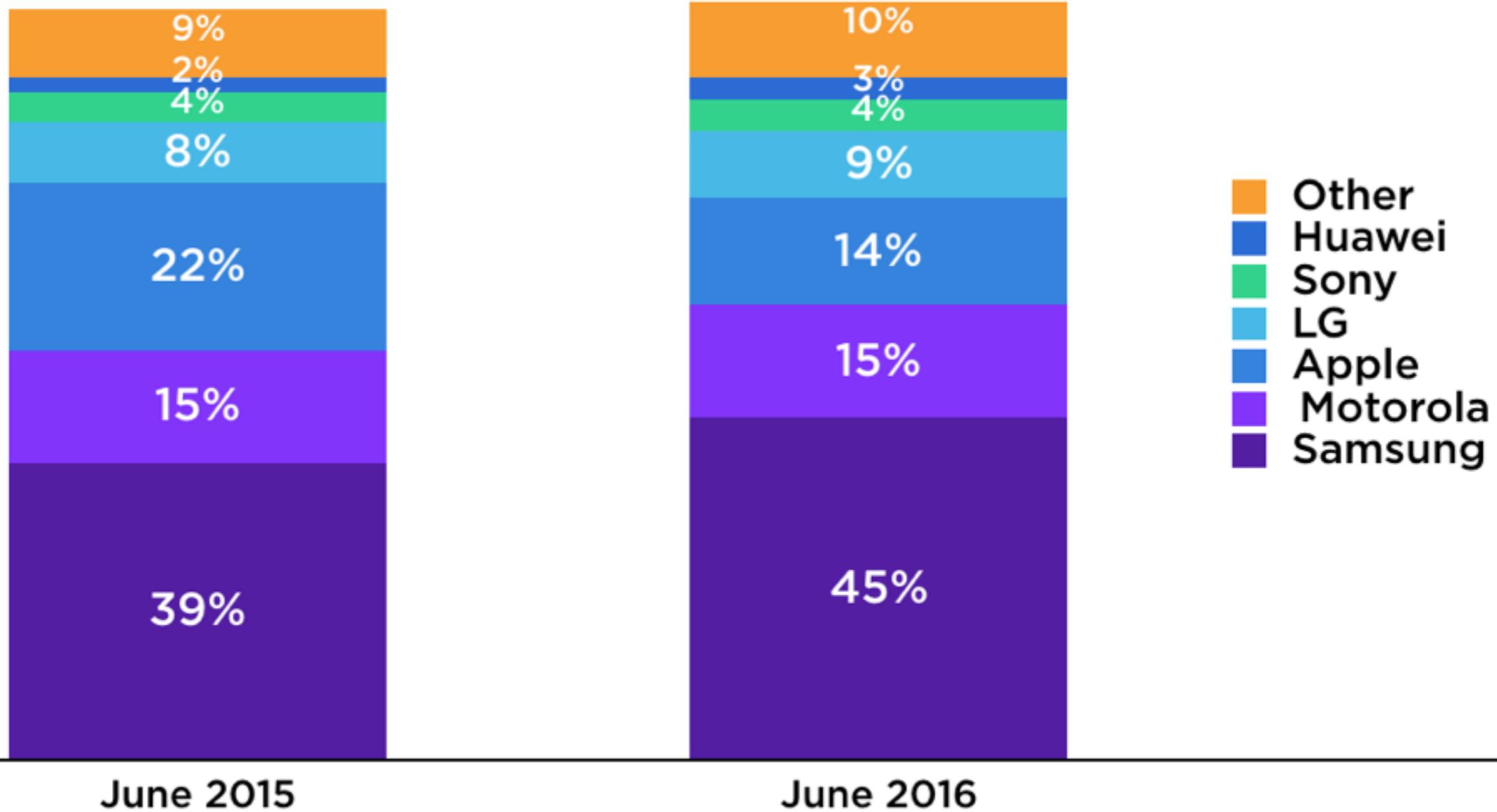
[View more »](#)



SensorTower Data That Drives App Growth

[sensortower.com](http://sensortower.com)

# SMARTPHONE MANUFACTURER DISTRIBUTION IN LATIN AMERICA

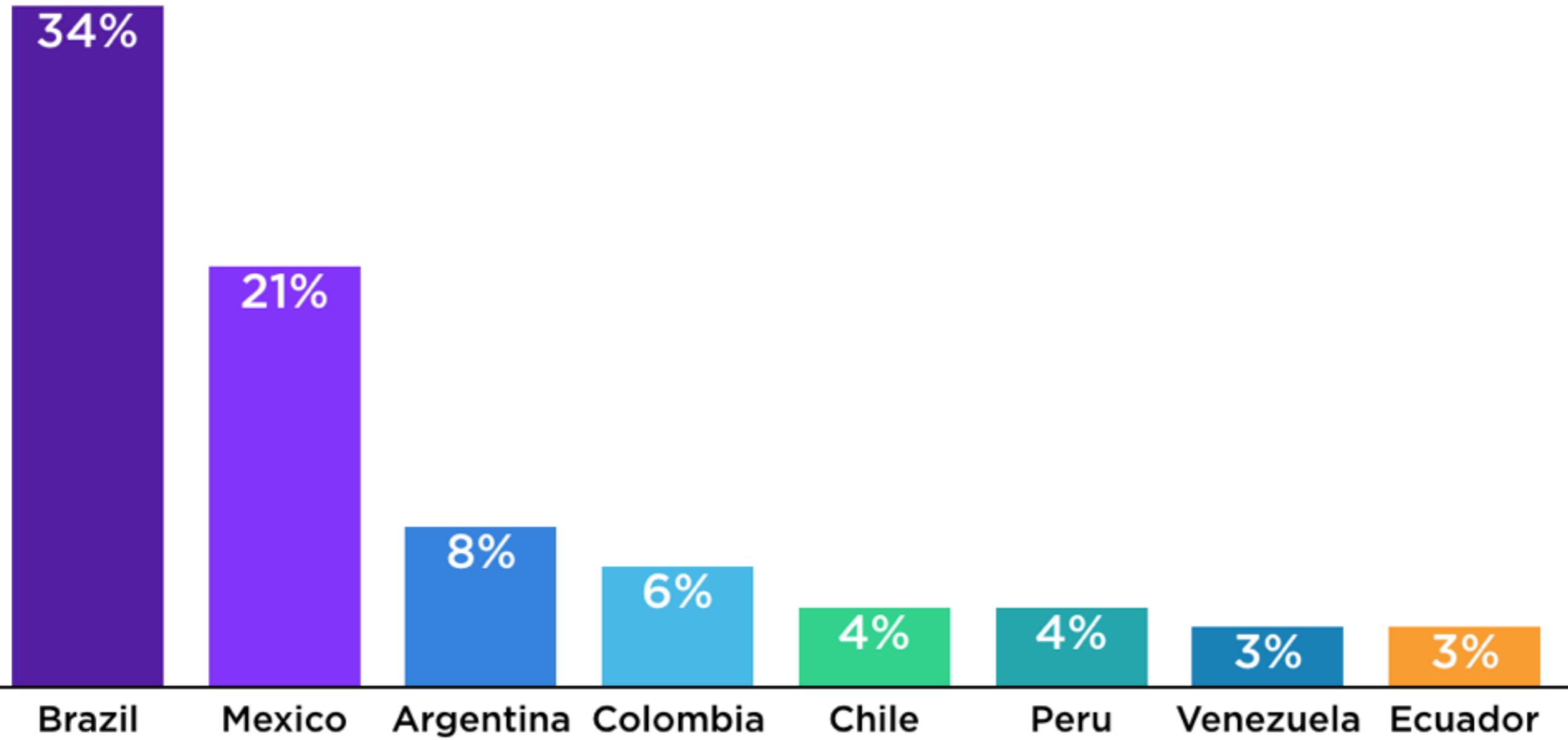


June 2015

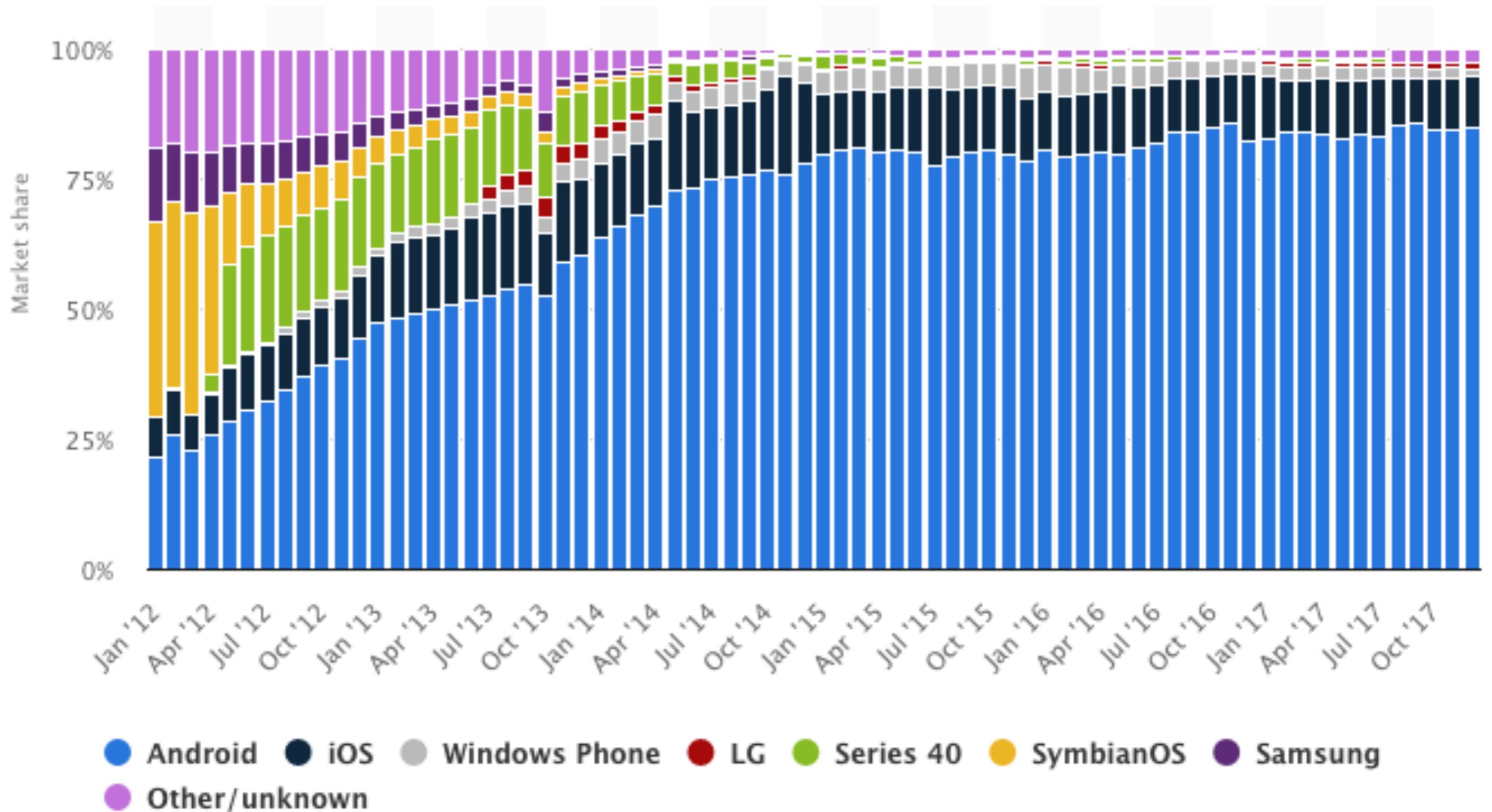
June 2016

<http://flurrymobile.tumblr.com/post/148496359837/latin-american-app-users-up-for-grabs-by-android>

# DISTRIBUTION OF SESSIONS BY COUNTRY IN LATIN AMERICA



<http://flurrymobile.tumblr.com/post/148496359837/latin-american-app-users-up-for-grabs-by-android>



<https://www.statista.com/statistics/262167/market-share-held-by-mobile-operating-systems-in-brazil/>

# 95,5% dos smartphones vendidos no Brasil são Androids

O resto é iPhone



Por [Paulo Higa](#)  
15 semanas atrás

NEWS 123

A nova extensão do Tecnoblog está pronta.

[Baixe Agora!](#)



MacKeeper

**Download  
Cleaner  
for Mac**

Award-winning  
System  
Utility. Clean  
Mac files  
Right Now!



O **Android** continua mais que consolidado nos smartphones vendidos no Brasil, de acordo com os números divulgados nesta sexta-feira (18) pela consultoria IDC. 95,5% dos aparelhos comercializados entre julho e setembro de 2016 rodavam o sistema operacional do **Google**. Os outros 4,5% eram iPhones.

Segundo a IDC, o mercado brasileiro de celulares voltou a crescer depois de cinco trimestres em queda. Foram 12,556 milhões de aparelhos vendidos no período, sendo 11,1 milhões de smartphones (88,6% do total) e 1,4 milhão de celulares básicos — aliás, **esses dispositivos simples têm crescido mais que os smartphones**. É um aumento de 7,2% nas **vendas** em comparação com o mesmo trimestre de 2015.

## Mais Populares



Servidores da Amazon caem  
(e levam o resto da internet...)

<https://tecnoblog.net/203749/android-ios-market-share-brasil-3t-2016/>

- SEM ALUGUEL E SEM TAXA DE ADESÃO
- CHIP E PLANO DE DADOS INCLUSOS
- VEM COM CARTÃO PRÉ-PAGO GRÁTIS
- NÃO PRECISA DE CONTA BANCÁRIA

DÉBITO E CRÉDITO



CRÉDITO



REFEIÇÃO



(Foto: Reprodução)

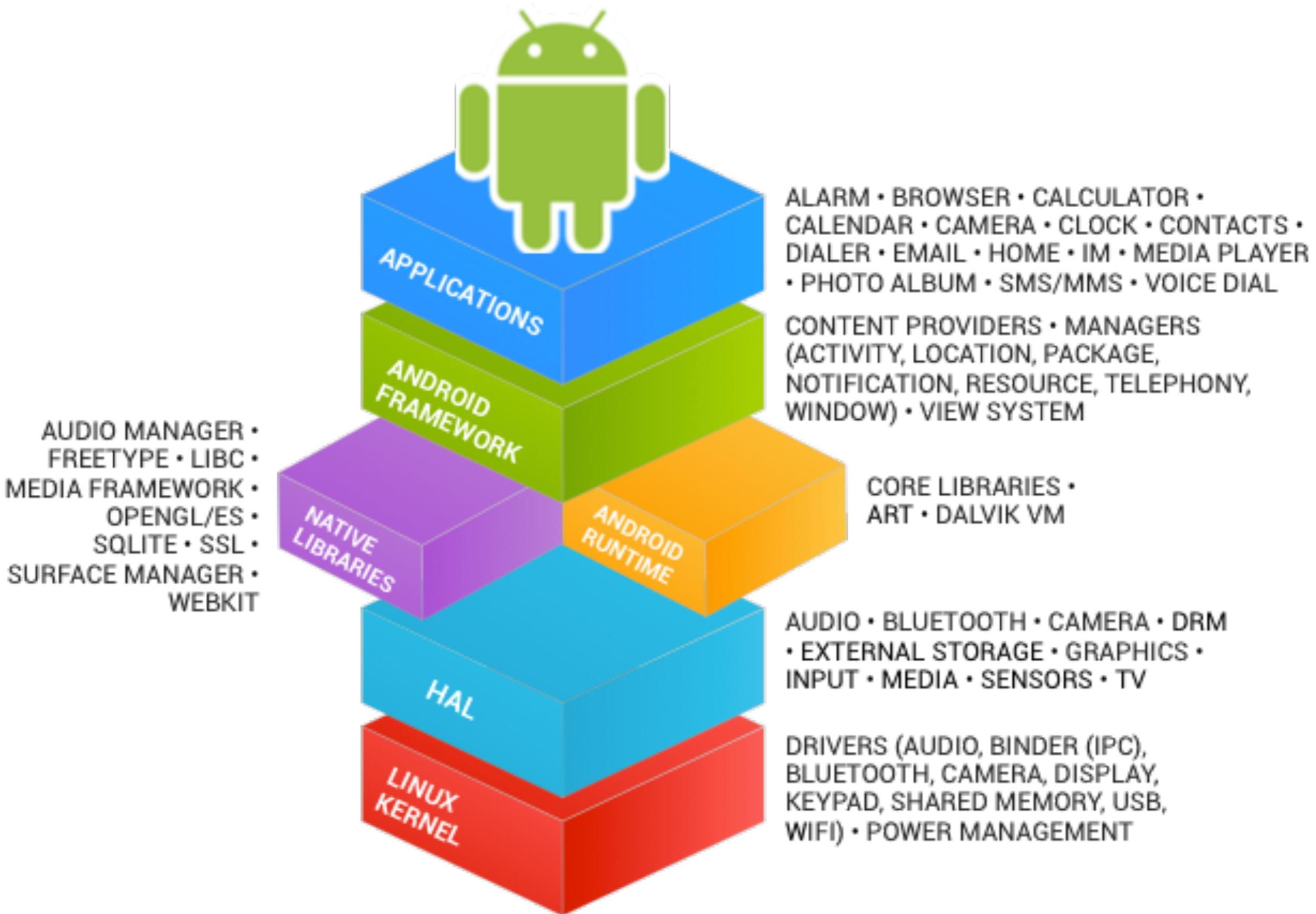
## Android cresce no Brasil e aumenta distância para iOS e Windows Phone

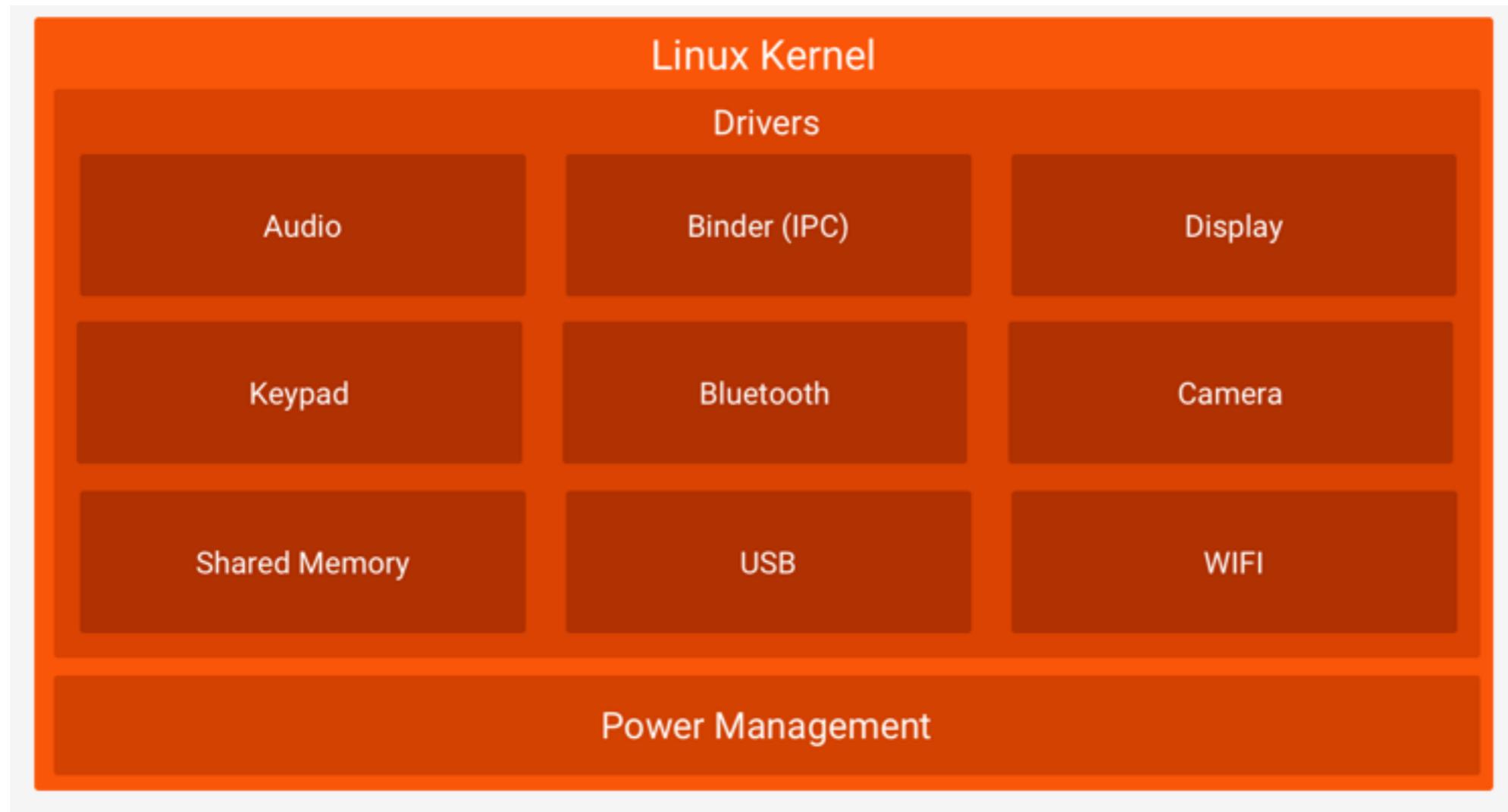
 LUCAS CARVALHO  03/05/2017 17H50 ANDROID :: IOS :: SISTEMAS OPERACIONAIS

O que é Android?

# A Plataforma Android

- Uma software stack para dispositivos móveis
- Diversas camadas, que vão desde serviços de baixo nível até os aplicativos utilizados pelo usuário
- SDK para criar aplicativos
- Ampla documentação disponível





- Camada de abstração entre hw e as camadas restantes, fornecendo serviços de SO genéricos
- Permissões, gerenciamento de memória e processos, IO de arquivos e rede

## Hardware Abstraction Layer (HAL)

Audio

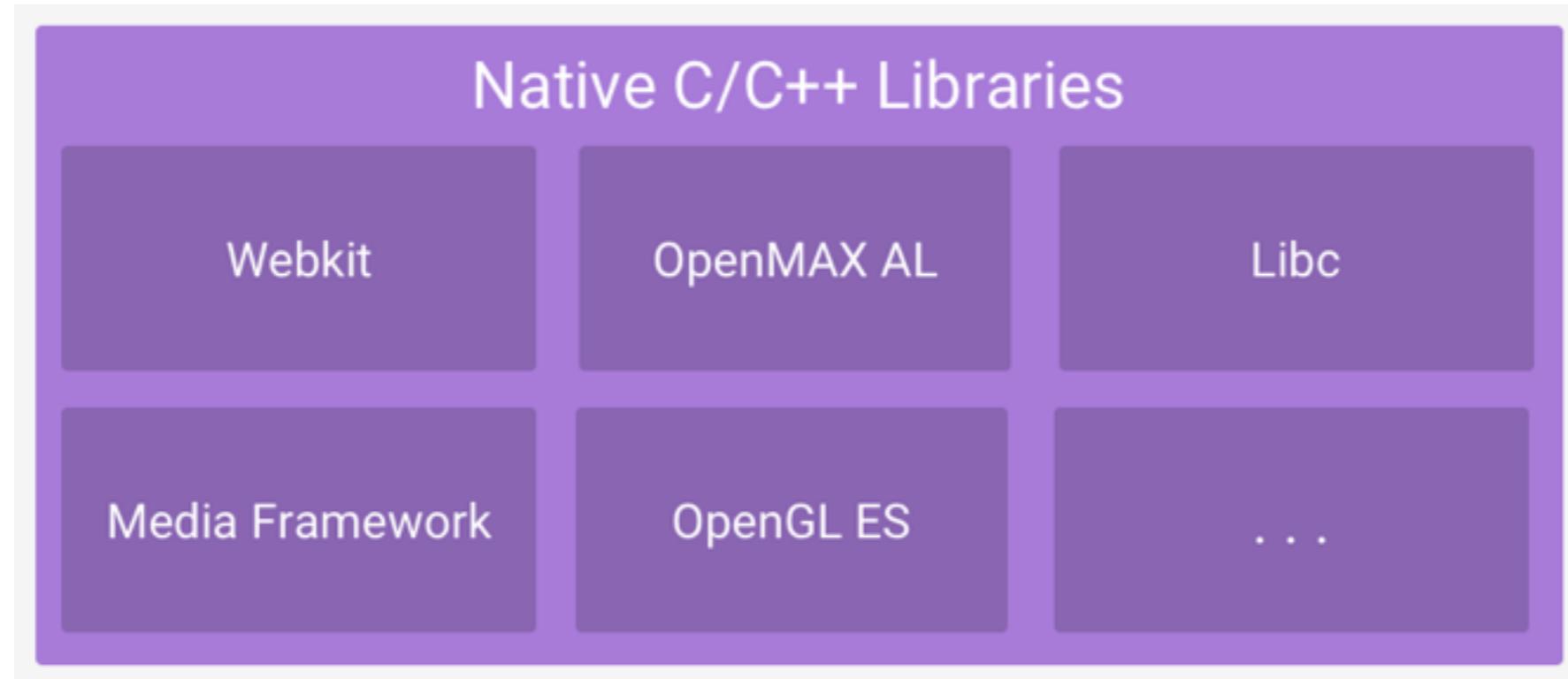
Bluetooth

Camera

Sensors

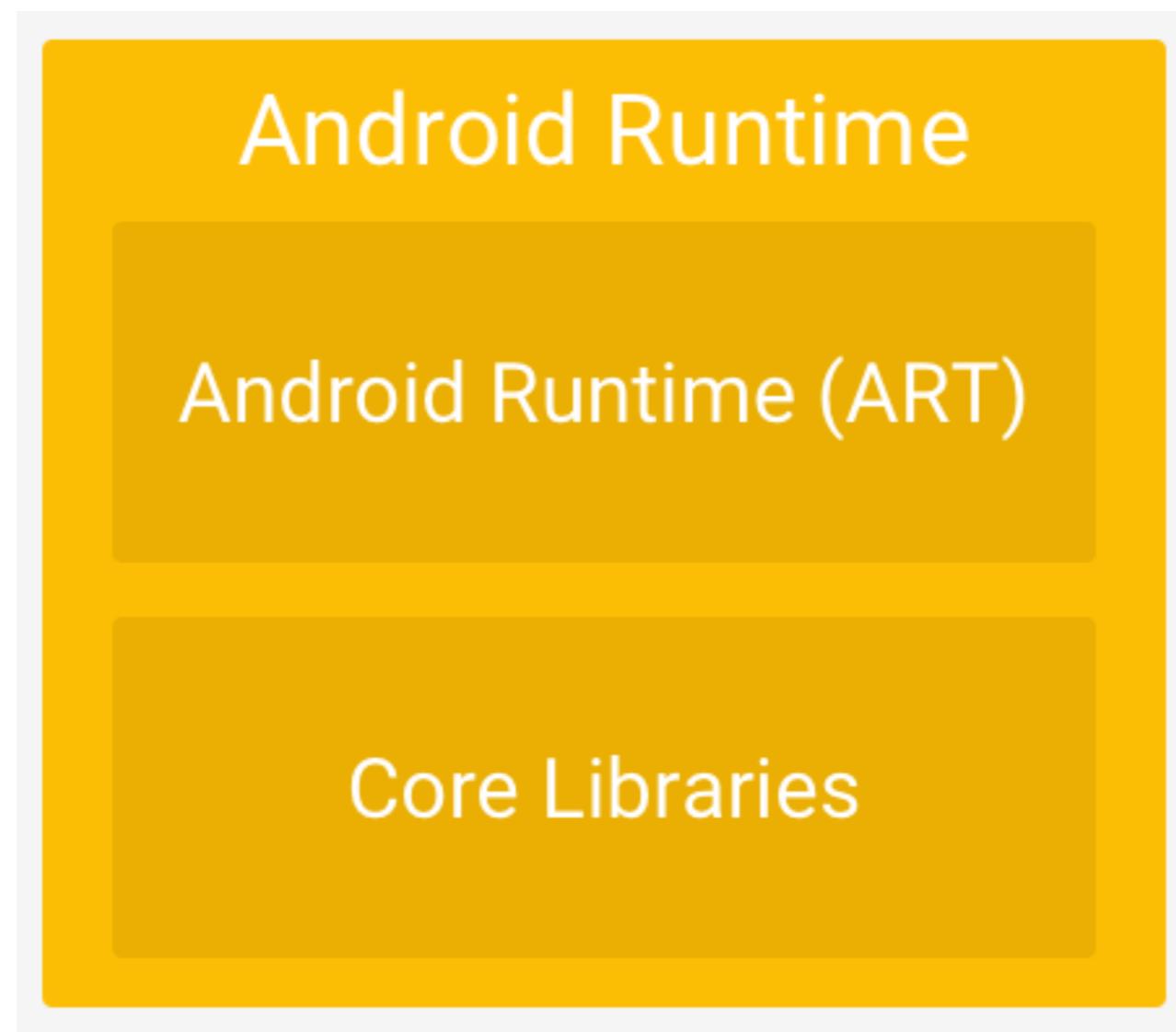
...

- Fornece interfaces padronizadas para expor hardware à camadas superiores
- Múltiplos módulos especificando componentes de hardware, como câmera, bluetooth
- Ao fazer uma chamada que acessa hardware, o sistema carrega o módulo correspondente



- Componentes “core” Android são construídos a partir de código nativo, que usam bibliotecas C/C++
- Algumas são expostas a desenvolvedores através do Android Application Framework, como OpenGL
- Aplicativos desenvolvidos com uso de C/C++ podem usar a NDK para acessar estas bibliotecas diretamente

# Android Runtime



## Core Libraries

- Bibliotecas
  - Fornecem maioria das funcionalidades disponíveis nas bibliotecas padrão Java
  - **java.\*** e **javax.\*** packages
  - pacotes específicos ao ciclo de vida das aplicações, **android.\***
  - **junit.\*** - testes

## Android Runtime (ART)

- Ambiente onde cada aplicação Android roda
- Cada aplicação roda seu próprio processo, com sua própria instância de ART
- Escrita para possibilitar a um dispositivo rodar múltiplas VMs eficientemente
- Executa arquivos no formato DEX (bytecode)
- Até Android 5.0 (API 21), Dalvik era padrão

## Java API Framework

Content Providers

View System

Managers

Activity

Location

Package

Notification

Resource

Telephony

Window

- APIs fornecem o fundamento para criação de aplicativos, ao permitir, de forma simplificada, o reuso de componentes e serviços do sistema
- Desenvolvedores têm acesso total às mesmas APIs utilizadas pelos core apps

## System Apps

Dialer

Email

Calendar

Camera

...

- Android fornece um conjunto de aplicativos:
  - Cliente de email, SMS, calendário, mapas, navegador, contatos, etc.
  - Estas aplicações não tem status especial, podem ser substituídas (exceto apps como **Settings**)
  - Fornecem funcionalidades que podem ser compartilhadas

# Android Apps

- Foco do curso: desenvolvimento de aplicações Android usando Kotlin
- A maioria das aplicações Android são escritas inteiramente em Java
  - É possível escrever parte ou tudo em C/C++
  - É possível criar web apps... etc...

# Atenção!

- Esta disciplina assume que todos já sabem o mínimo de programação!
- Se você ainda não sabe programar razoavelmente bem em Java/Kotlin, mas tem familiaridade com outras linguagens, você pode melhorar o seu conhecimento com muitos tutoriais e cursos disponíveis na web.
- Porém... Não é necessária experiência prévia com desenvolvimento para dispositivos móveis! :-)
- Também assumo que você está disposto a procurar, ler e aprender sobre o assunto a partir de documentação disponível.

Qual o primeiro programa  
Java que você escreveu?

# Componentes

- Aplicações Android não definem método main()
- Estendemos classes fornecidas por Android que definem componentes de aplicação
  - Adicionalmente, definimos metadados que informam detalhes destas classes à plataforma
- Android instancia e roda estes componentes na medida do necessário
- Cada componente tem um propósito próprio e APIs correspondentes

# Componentes Android

- Activities
- Services
- Content Providers
- Broadcast Receivers

# Activity

- Classe básica para interação com usuário, *análoga* a uma janela ou dialog em app desktop
- Representa uma tela com interface para o usuário
- Em geral, pode ser um ponto de entrada da aplicação
  - Exemplo: Email App
  - Em alguns dispositivos, já é possível lidar com mais de uma activity ao mesmo tempo

# Service

- Activities tem vida curta e podem ser encerradas a qualquer momento
- Services são pontos de entrada de propósito geral que permitem que um app continue rodando em background
- Componentes que foram projetados para continuar rodando independente de UI
- Por exemplo, Music Player app, live wallpaper, serviços de acessibilidade, entre outros...

# BroadcastReceiver

- O sistema e outros aplicativos enviam eventos de *broadcast* de tempos em tempos
- Escuta e responde a eventos que acontecem “fora” do fluxo regular do usuário no app
- Desempenham papel de subscriber no padrão publish/subscribe
- Exemplo: Messaging application

# ContentProviders

- Gerencia um conjunto compartilhado de dados do app que podem estar armazenados no sistema de arquivos, banco de dados SQLite, web, ou em outro local de persistência...
- Por meio deste componente, é definida uma camada de abstração que permite compartilhar dados entre aplicações
  - Interfaces similares à bancos de dados
  - Gerenciam *interprocess communication*
  - Permitem controlar acesso aos dados
- Exemplo: Browser, contatos

# Componentes

- Um aspecto particular do projeto do sistema Android é o fato de que qualquer app pode iniciar o componente de um outro app
- Por exemplo, se você deseja tirar uma foto...
- Quando o sistema inicia um componente, o processo do app é iniciado, caso não esteja rodando
- Cada app roda em seu próprio processo, quando você deseja iniciar outro componente, você solicita ao sistema...

# Apps e Packages

- Dado que você escreveu toda sua aplicação, as ferramentas de build de Android geram aplicação
- Formato APK é utilizado para distribuição
- Todo app tem um package name, também chamado de application id
  - nome válido de acordo com regras Java
  - único no dispositivo e na play store
- Ao criar um projeto declaramos o package name

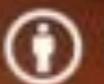
# Apps e Packages

- Dado que você escreveu toda sua aplicação, as ferramentas de build de Android geram aplicação
- Formato APK é utilizado para distribuição
- Todo app tem um package name, também chamado de application id
  - nome válido de acordo com regras Java
  - único no dispositivo e na play store
- Ao criar um projeto declaramos o package name

# Variabilidade e fragmentação



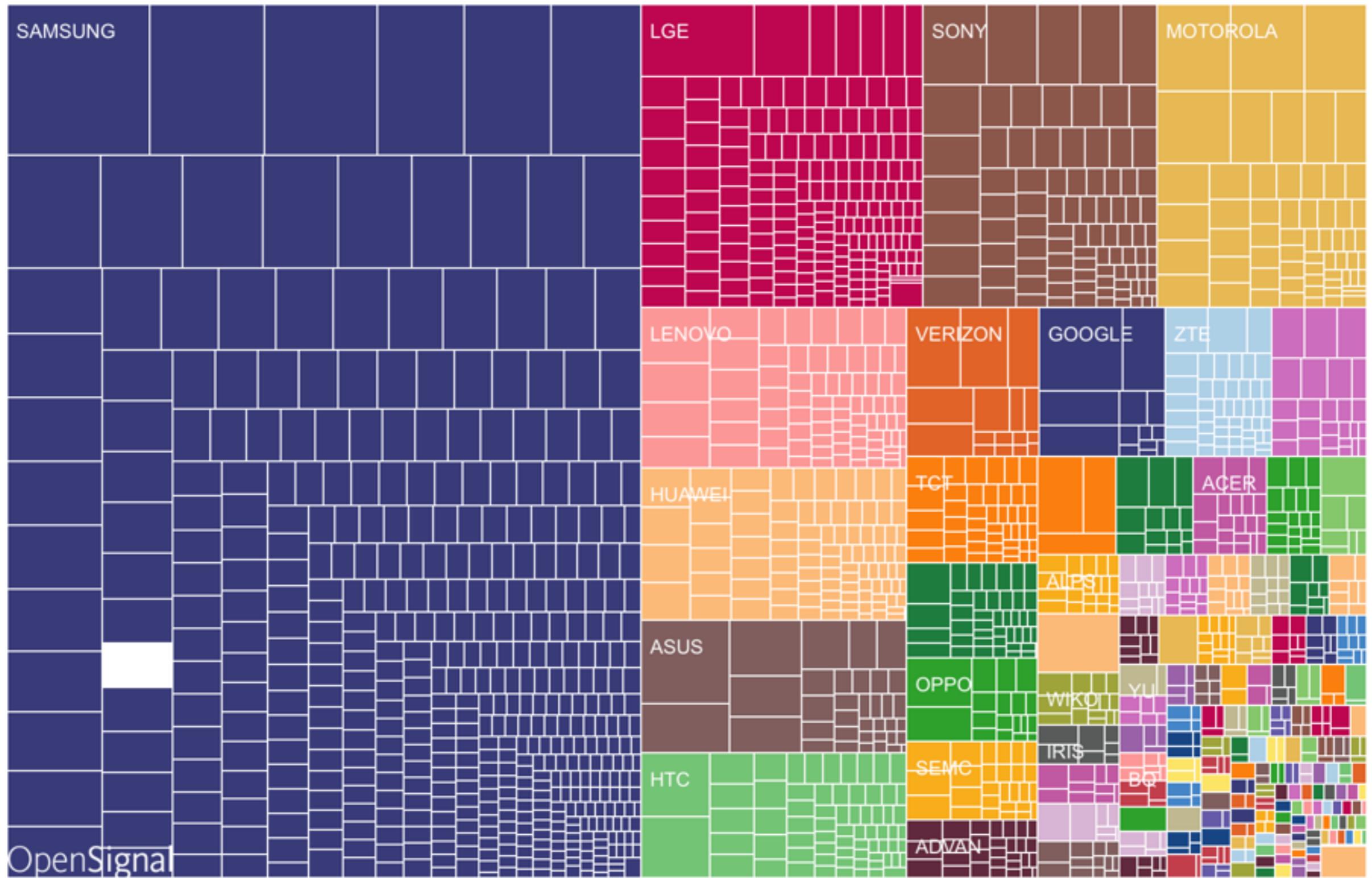
 animoca



Except where otherwise noted, this work is licensed under  
<http://creativecommons.org/licenses/by-nd/3.0/>

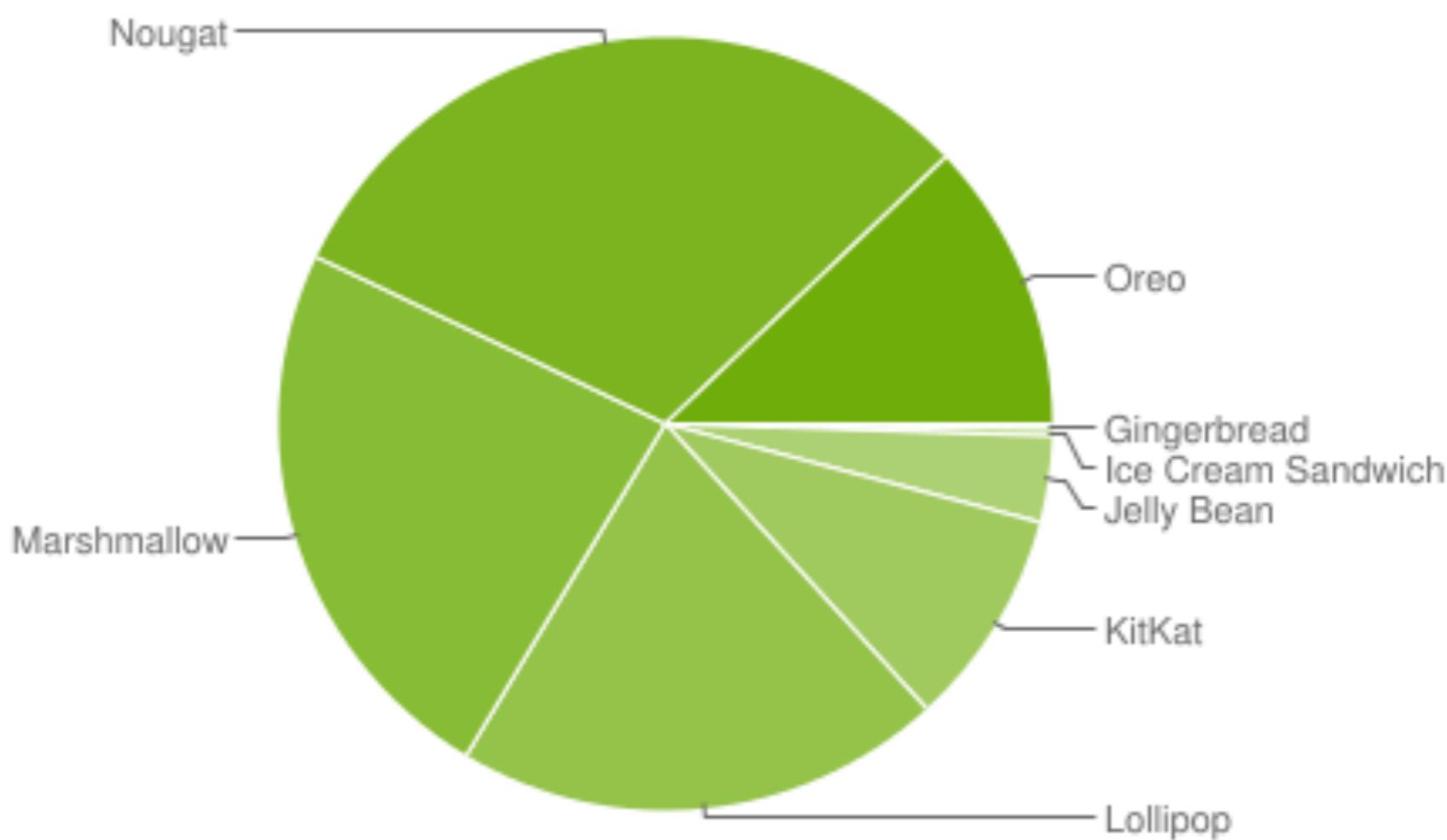
<http://phandroid.com/2012/05/12/this-is-androids-fragmentation-and-it-is-still-a-feature-opinion/>

## BRAND FRAGMENTATION



OpenSignal

<https://opensignal.com/reports/2015/08/android-fragmentation/>



*Dados coletados durante um período de 7 dias encerrado em 2018/7/23.  
Todas as versões com menos de 0,1% de distribuição não foram exibidas.*

android



Quais os desafios inerentes ao desenvolvimento de aplicações para dispositivos móveis?

# Memória Limitada

- Obviamente há exceções, mas na média...
- Desenvolvedores costumam utilizar dispositivos mais rápidos, novos e melhores que a média
- Android roda múltiplos apps (activities e services) em paralelo
- Idealmente, use o mínimo de memória possível

# CPU

- Mesmo em dispositivos *high-end*, CPUs são mais lentas
- Dispositivos Android *low-end* são abundantes no mercado, na média tem 1 ou 2 cores
- O sistema controla uso da CPU para aumentar duração da bateria, então nem sempre o poder de processamento está sendo totalmente utilizado

# Memória e Desempenho

- Estão intimamente ligados
- Quanto mais memória alocamos, mais esforço o dispositivo tem de fazer para seu app
- Maior necessidade de memória significa uma heap maior para seu app, menos espaço para outros
- Maior heap, pausas mais longas de GC
- Quanto mais esforço, maior consumo de bateria...

# Armazenamento

- Variam drasticamente entre dispositivos
- Uma aplicação de 500MB é pequena?
- Desempenho também é um problema, principalmente ao utilizarmos cartões SD e memória removível
  - desempenho também varia de acordo com fornecedor, chipset, classe de velocidade

# Conexões de Rede

- Conexões de dados intermitentes, com baixa largura de banda, alta latência
- Principais problemas associados:
  - Dependência de redes velozes
  - Sincronização frequente

em resumo...

# *Don't be evil*

- É fácil acharmos que a nossa aplicação é a mais importante
- Pode até ser, mas ainda será necessário usar outras aplicações para carregar
- Quanto mais recursos forem utilizados, mais o resto das aplicações no dispositivo sofre
- Quanto maior o processo, maior a necessidade do sistema de liberar espaço ao precisar de recursos, o que prejudicará a experiência de uso

# *Tragedy of the Commons*

- Aplicativos individualmente não esgotam os recursos do dispositivo
- Mas, o agregado de vários aplicativos que não se preocupam com recursos gera experiência terrível
- Por exemplo: sincronização frequente.
  - imagine um dispositivo com mais de 100 apps que fazem sincronização frequente
  - nunca dorme, gasta bateria mais rápido, etc...

vamos ao que interessa  
a vocês hoje...

# Objetivos Gerais da Disciplina

- Motivar, apresentar, exercitar e consolidar o desenvolvimento de aplicações para dispositivos móveis, utilizando a plataforma Android como estudo de caso
- Desenvolver vários aplicativos simples no decorrer da disciplina para exercitar diferentes conceitos da plataforma
- Desenvolver um aplicativo como projeto da disciplina
- Analisar o aplicativo desenvolvido de acordo com os tópicos comumente associados ao contexto *mobile*

# Resultados Esperados ao Final do Curso

- Compreender organização, padrões e mecanismos de programação da plataforma Android e estar apto a usá-los efetivamente para desenvolver sua própria aplicação;
- Usar ferramentas de desenvolvimento para criar, entender, depurar e otimizar aplicações Android;
- Compreender as características distintas e restrições que existem em dispositivos móveis e lidar com isto no contexto de aplicações Android;
- Estar apto a encontrar fontes adicionais de informação para entender e resolver problemas relacionados com desenvolvimento Android.

# Conteúdo Planejado

- Introdução à Plataforma Android
- Fundamentos de Aplicações Móveis
- Android Components
- Lidando com Threads
- Testes
- Análise de código
- Rede e consumo de dados
- Design Patterns
- Bateria
- Processamento
- Memory leaks
- Gerenciamento de dados
- ...

# Metodologia

- Aulas tradicionais
- Aulas práticas (*labs*)
- Acompanhamento de projetos (presencial e por slack)

# Bibliografia Básica

- **<http://developer.android.com>**
- **The Busy Coder's Guide to Android Development**  
(<https://commonsware.com/Android/>)
- **Android Programming: The Big Nerd Ranch Guide**  
(<https://www.bignerdranch.com/books/android-programming/>)
- **Kotlin for Android Developers**  
(<https://antonioleiva.com/kotlin-android-developers-book/>)
- **Kotlin in Action**  
(<https://www.manning.com/books/kotlin-in-action>)
- **Android Weekly**  
(<http://androidweekly.net>)

# Contato

- Site: **<http://if710.github.io>**  
*em constante construção*
- Slack: **<https://if710.slack.com/signup>**  
*Faça o login com seu email do CIn.*
- Google Classroom: **<https://classroom.google.com>**  
*Código: **9r6jm5**.*
- E-mail: **[lmt@cin.ufpe.br](mailto:lmt@cin.ufpe.br)**

# Avaliação

***sujeita à alteração...***

- Exercícios diversos (50%)
  - Criação e avaliação de pequenos apps
  - Criação de Tutoriais de bibliotecas
- Projeto (50%)
  - A nota do projeto comprehende não apenas a qualidade do resultado final, mas também acompanhamentos, respeito a prazos e datas de entrega, etc.

*as porcentagens dizem respeito à  
nota da disciplina como um todo*

# Para a próxima aula

- Navegar em developers.android.com e estudar!
- Se inscrever no Slack
- Se ainda não tem conta em Github, criar *vergonha na cara*
- Formar uma dupla e discutir ideias de projeto
- Instalar Android Studio e fazer o Hello World
- Fazer Kotlin koans (<https://kotlinlang.org/docs/tutorials/koans.html>)

# **Programação para Dispositivos Móveis**

**Leopoldo Teixeira**

[lmt@cin.ufpe.br](mailto:lmt@cin.ufpe.br) | @leopoldomt