

# PdAM-I

Programação de Aplicativos Mobile - I

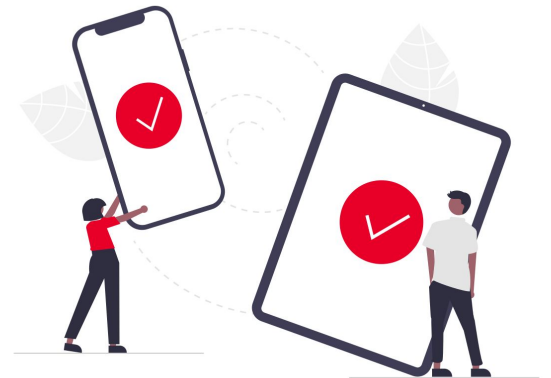
---

# Agenda

## 1. **Conceitos de Dispositivos móveis e o mercado**

- a. Introdução ao desenvolvimento mobile
- b. Dispositivos móveis e o mercado
- c. Definição de Dispositivos móveis
- d. Arquitetura de Sistemas

# Por que mobile?



# Utilização de dispositivos mobile vs desktop

Nos últimos anos a grande maioria dos acessos e utilização da internet se deu através de dispositivos **mobile**, sendo o principal o *smartphone*, conforme dados do IBGE.



Fonte: IBGE, Diretoria de Pesquisas, Coordenação de Trabalho e Rendimento, Pesquisa Nacional por Amostra de Domicílios Contínua 2018/2019.

# Utilização de dispositivos mobile vs desktop

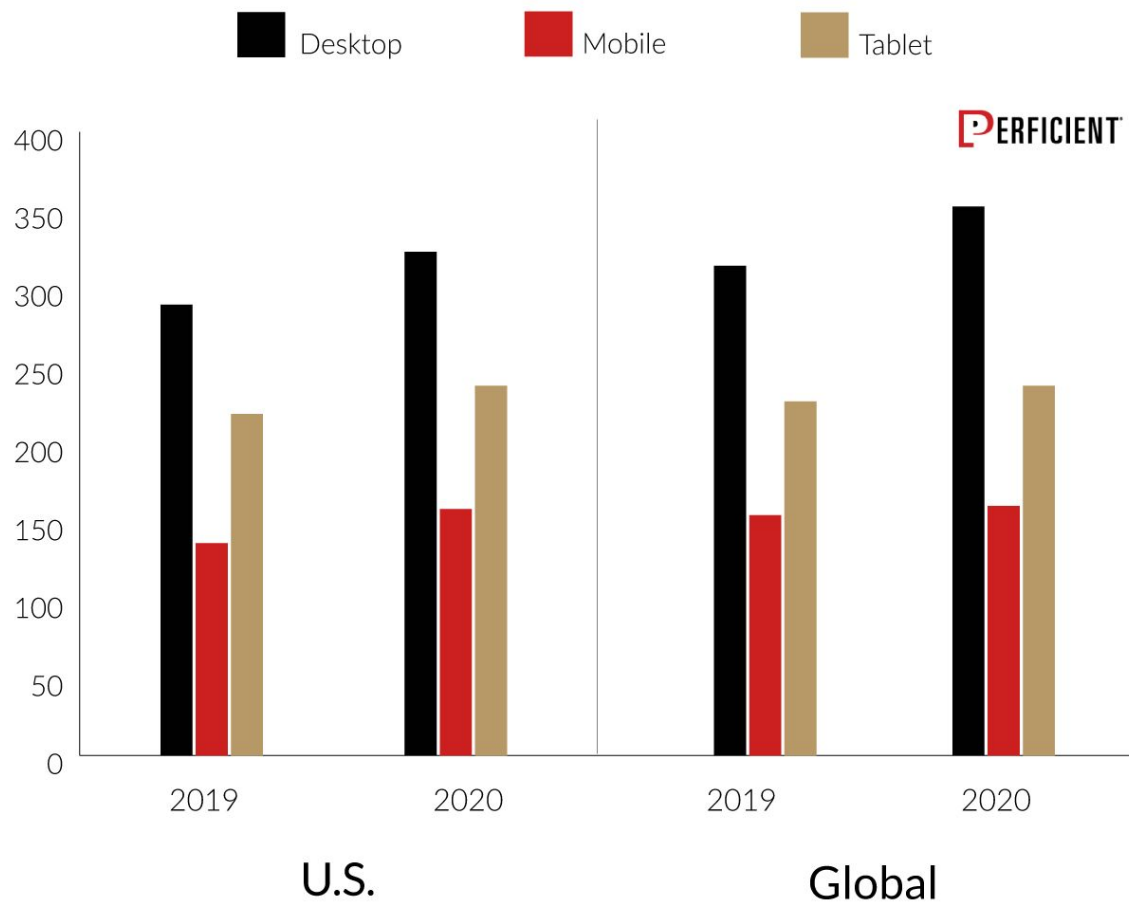
Podemos ver como a maior parte do fluxo de dados que passa pela internet, hoje, é dividido entre desktop e mobile.

Reforçando a nossa preocupação em desenvolver **sites** e **aplicações** para esse tipo de dispositivo.

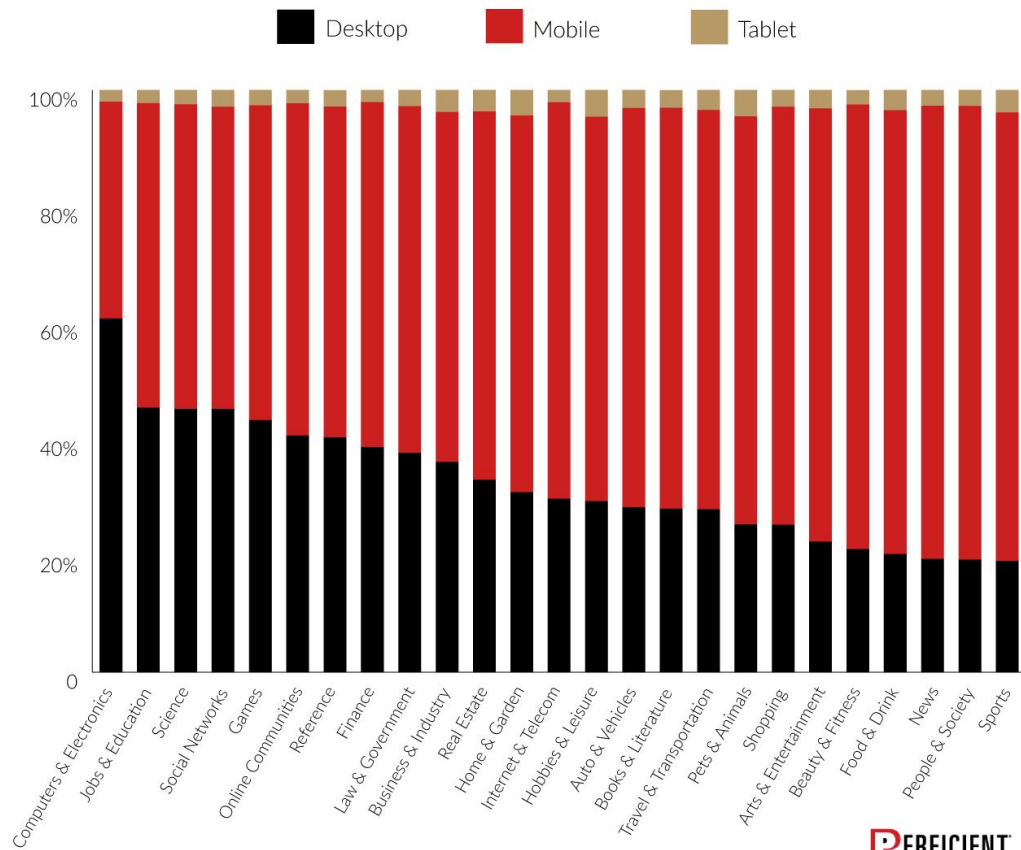
Desktop vs Mobile Internet Usage Statistics in 2020 (Global)



# US vs. Global Average Time on Site in Seconds



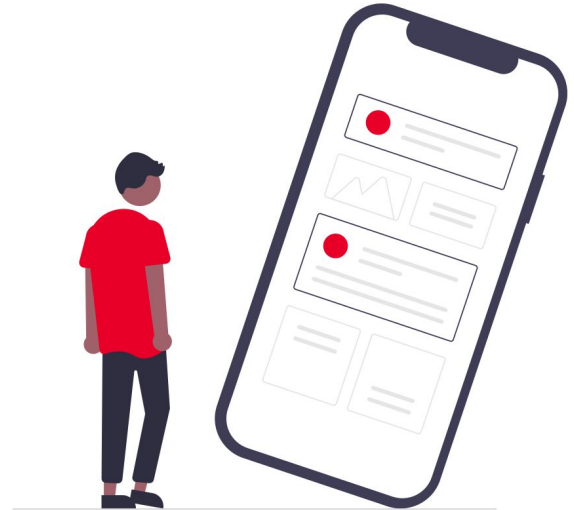
## Percentage of Global Visits by Device in 2020



# Utilização de dispositivos mobile vs desktop

Só de olhar essas estatísticas já entendemos que, hoje, **a principal porta de entrada** de um usuário para um serviço, produto ou portal, é através de um dispositivo mobile.

Esse usuário gasta cada vez **menos tempo** em sites e apps e possui uma **infinidade** de outras coisas para fazer, sendo assim a principal preocupação hoje em dia é com a **experiência do usuário**.





# Desenvolvimento mobile

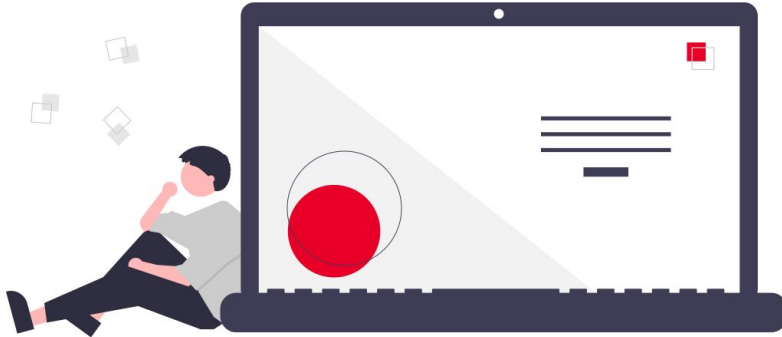
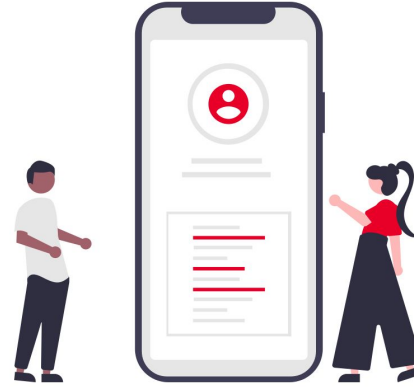
É desenvolver **pensando** nas especificidades de equipamentos mobile, sendo desde a sua utilização pelo usuário, como as necessidades de cada dispositivo, sendo então importante o entendimento das **arquiteturas** utilizadas por dispositivos móveis.

É **programar** para equipamentos como smartphones e tablets, visando uma melhor integração com o sistema operacional dos dispositivos móveis e seus inúmeros **recursos**.

# Desenvolvimento mobile

## Aplicativo (App)

Software desenvolvido em linguagem de programação nativa ao sistema operacional hospedeiro que é **instalado** no equipamento.



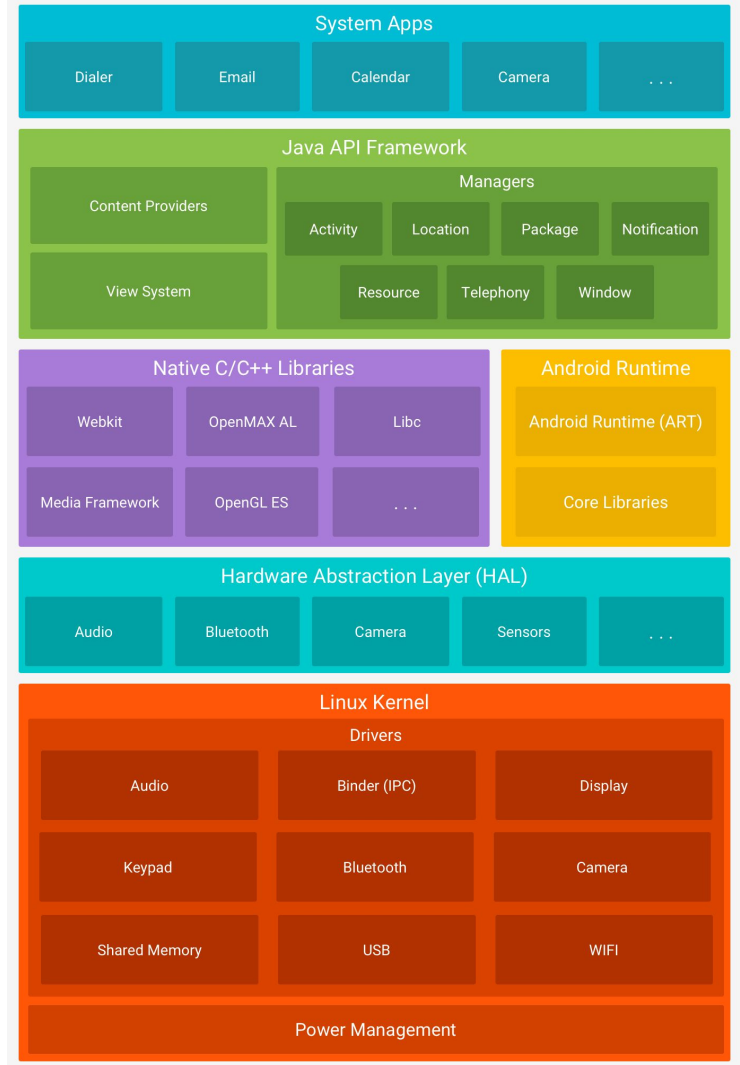
## Site:

Pedaço de código hospedado em um **servidor** que é acessado via *browser* pela internet

# Sistemas operacionais móveis - Android

O Android, sistema operacional da google é um sistema operacional de código aberto, desenvolvido a partir de um **kernel linux** e atualmente o sistema operacional mais popular em dispositivos móveis.

# Arquitetura básica do Android



# Arquitetura básica do Android

Cada app Android é ativado na própria sandbox de segurança, protegido pelos seguintes recursos de segurança do Android:

- O sistema operacional Android é um sistema Linux multiusuário em que cada aplicativo é um usuário diferente.
- Por padrão, o sistema atribui a cada aplicativo um código de usuário do Linux exclusivo (o código é usado somente pelo sistema e é desconhecido para o aplicativo). O sistema define permissões para todos os arquivos em um aplicativo, de modo que somente o código de usuário atribuído àquele aplicativo pode acessá-los.
- Cada processo tem a própria máquina virtual (VM), portanto, o código de um aplicativo é executado isoladamente de outros aplicativos.
- Por padrão, cada aplicativo é executado no próprio processo do Linux. O Android inicia o processo quando é preciso executar algum componente do aplicativo. Em seguida, encerra-o quando não mais é necessário ou quando o sistema precisa recuperar memória para outros aplicativos.

# Arquitetura básica do Android

O sistema Android implementa o ***princípio do privilégio mínimo***.

Ou seja, cada aplicativo, por padrão, tem acesso somente aos componentes necessários para a execução do seu trabalho e nada mais.

# Estrutura básica do App

Componentes de aplicativo são os blocos de construção de um app Android. Cada componente é um ponto de entrada por onde o sistema ou o usuário podem entrar no aplicativo. Alguns componentes dependem de outros.

Há quatro tipos diferentes de componentes de aplicativo:

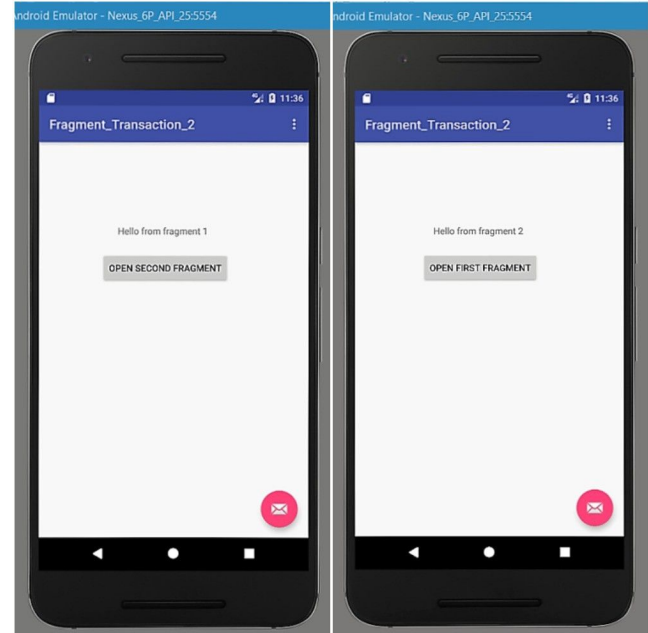
- Atividades
- Serviços
- Broadcast receivers
- Provedores de conteúdo

Cada tipo tem uma finalidade distinta e tem um ciclo de vida específico que define a forma como o componente é criado e destruído.

# Activity

Uma *atividade* é o ponto de entrada para a interação com o usuário.

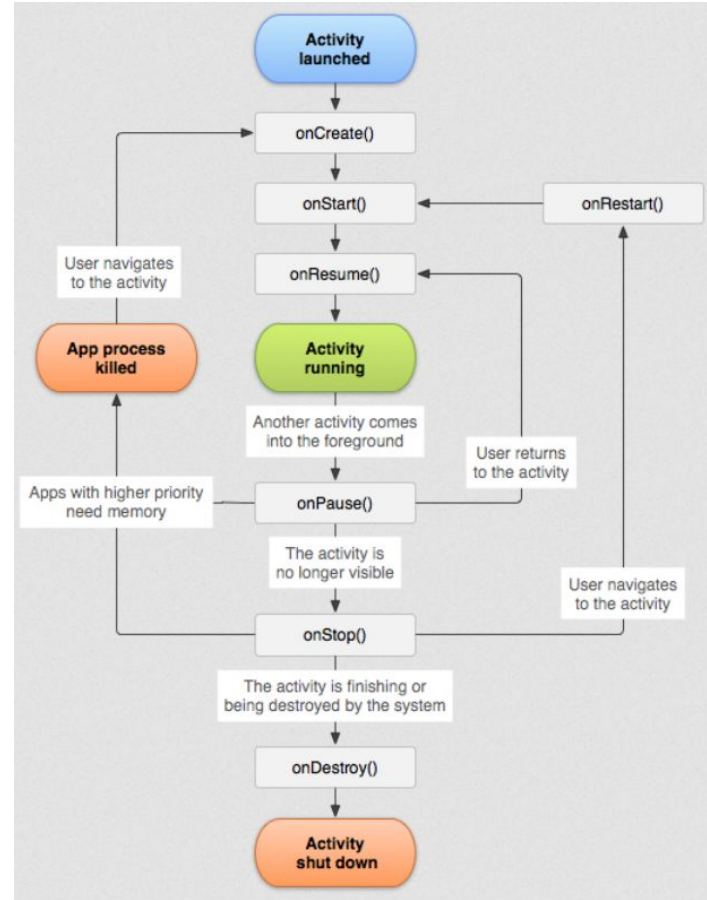
Ela representa uma tela única com uma interface do usuário.





# Ciclo de vida de uma activity

Para navegar entre as fases do ciclo de vida da atividade, a classe “Activity” fornece um conjunto principal de seis callbacks: onCreate(), onStart(), onResume(), onPause(), onStop() e onDestroy().



# Service

O *serviço* é um ponto de entrada para manter um aplicativo em execução no segundo plano, seja qual for o motivo. É um componente executado em segundo plano para realizar operações de execução longa ou trabalho para processos remotos. Serviços não apresentam uma interface do usuário. Por exemplo, um serviço pode tocar música em segundo plano enquanto o usuário está em um aplicativo diferente ou buscar dados na rede sem bloquear a interação do usuário com uma atividade.

# Broadcast Receiver

O *broadcast receiver* é um componente que faz o sistema entregar eventos ao aplicativo fora de fluxo de usuários comum. Isso permite que o aplicativo responda a anúncios de transmissão por todo o sistema. Como os broadcast receivers são mais uma entrada bem definida no aplicativo, o sistema consegue entregar transmissões até a aplicativos que não estejam em execução no momento. Por exemplo, um aplicativo pode programar um alarme para postar uma notificação que avise o usuário sobre um evento futuro.

# Provedores de Conteúdo

*Provedores de conteúdo* gerenciam um conjunto compartilhado de dados do aplicativo que você pode armazenar nos sistemas de arquivos, em banco de dados SQLite, na Web ou em qualquer local de armazenamento persistente acessível ao seu aplicativo. Por meio do provedor de conteúdo, outros aplicativos podem consultar ou até modificar os dados, caso o provedor de conteúdo permita. Por exemplo, o sistema Android oferece um provedor de conteúdo que gerencia os dados de contato do usuário.

# Referências

Perficient - Pesquisa sobre utilização de plataformas Desktop x Mobile

<https://www.perficient.com/insights/research-hub/mobile-vs-desktop-usage>

StatCounter

<https://gs.statcounter.com/os-market-share/mobile/worldwide>

IBGE - PNAD Informativo sobre utilização de diferentes dispositivos.

[https://biblioteca.ibge.gov.br/visualizacao/livros/liv101794\\_informativo.pdf](https://biblioteca.ibge.gov.br/visualizacao/livros/liv101794_informativo.pdf)

# Referências

Diretrizes para desenvolvimento Android

<https://developer.android.com/about?hl=pt-br>

StatCounter

<https://gs.statcounter.com/os-market-share/mobile/worldwide>

IBGE - PNAD Informativo sobre utilização de diferentes dispositivos.

[https://biblioteca.ibge.gov.br/visualizacao/livros/liv101794\\_informativo.pdf](https://biblioteca.ibge.gov.br/visualizacao/livros/liv101794_informativo.pdf)