

# Programação para dispositivos móveis

## Aula 1 - Introdução à computação móvel, a plataformas de desenvolvimento e à programação móvel Android

### INTRODUÇÃO

---



O Android é um sistema operacional para dispositivos móveis open source baseado em Linux, criado em 2003 para o mercado de mobile. Desde então, a demanda por profissionais desta área tem exigido, cada vez mais, uma melhor qualificação profissional.

Esta aula tem como tema central apresentar os conceitos de computação móvel, dispositivos móveis, bem como o Android e seus componentes essenciais a um aplicativo.

## OBJETIVOS

---



Descrever os principais conceitos referentes à tecnologia mobile;

Identificar os principais conceitos referentes ao sistema operacional Android;

Descrever os componentes básicos do sistema Android para a construção de um aplicativo.

## INTRODUÇÃO AO ANDROID

Você já deve ter notado que o crescimento das tecnologias da informação (TI) tem impactado profundamente nossas vidas, na medida em que tem modificado a forma como nos comunicamos, aprendemos e, porque não dizer, percebermos o mundo.

Precisamos cada vez mais de um maior volume, mais qualidade e velocidade no acesso às informações.

Neste “Admirável Mundo Novo” é fundamental o acesso às informações e serviços, independente de onde estejamos localizados e a qualquer momento, dando oportunidade ao aparecimento da computação móvel como um novo paradigma computacional.

Veja sua definição:



*“Computação móvel é a capacidade de ter acesso à informação, em qualquer lugar e a qualquer hora, onde existe total mobilidade do usuário.”*



## DISPOSITIVO MÓVEL

Um dispositivo móvel usado em larga escala, a partir de 1986, foi o Teletrim, cuja finalidade era a recepção de mensagens de texto (de poucas linhas) aos usuários do serviço. Era equivalente ao serviço de SMS (Short Message Service) provido pelas operadoras de telefonia nos dias de hoje.



Fonte da Imagem:

Um dispositivo móvel (*handheld*) é um computador de bolso, normalmente equipado com uma pequena tela (*output*) e um teclado em miniatura (*input*). Em alguns dispositivos móveis, o teclado está incorporado à tela, no que chamamos de dispositivo touchscreen, tal como nos tablets.

Existem diversas categorias de dispositivos de computação móvel. Entre os mais comuns estão:

- *Smartphone*;
- Tablet;
- PDA (*Personal Digital Assistant*);
- Celular;
- Console portátil;
- Coletor de dados;
- GPS (*Global Positioning System*).

## CONCEITO DE MOBILIDADE

O conceito base que impulsionou o desenvolvimento dos dispositivos móveis foi a mobilidade, que pode ser definida como:



*"A capacidade de poder ser movido fisicamente e/ou poder ser utilizado enquanto está em movimento".*



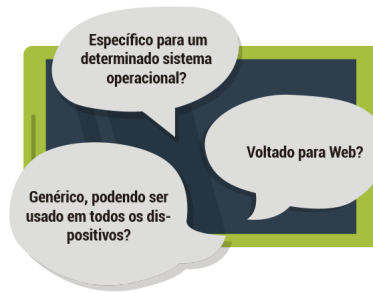
Para isso, os dispositivos móveis possuem determinadas características essenciais:



## APLICATIVOS

Com o crescimento do uso de dispositivos móveis, surgiram alguns sistemas operacionais e também uma crescente demanda por aplicativos.

## Mas que tipo de aplicativo deve-se construir?



## Atenção

, Antes de iniciarmos o desenvolvimento de uma aplicação para um dispositivo móvel é necessário definir o tipo de aplicação desejada.

Os possíveis tipos são:

---

### NATIVO

- Desenvolvidas especificamente para uma determinada plataforma móvel;
- Faz uso da linguagem de programação suportada pela plataforma e seu respectivo SDK (Software Development Kit);
- Normalmente, são instaladas através de uma loja de aplicativos, como, por exemplo, App Store e Google Play.

### WEBMOBILE

- Diferente das aplicações nativas, consiste em um site com um layout otimizado para plataforma móvel;
- Faz uso de linguagens web (Html, Css, Javascript);
- Pode ser usado por qualquer plataforma móvel.

### HÍBRIDA

- Consiste na combinação dos tipos nativo e WebMobile;
- Em geral, possui um navegador de internet customizado para o site do aplicativo;
- É desenvolvido para uma plataforma móvel específica;
- Tem se destacado nos últimos tempos.

### MULTIPLATAFORMA

- Faz uso de Framework para geração de aplicações móveis.

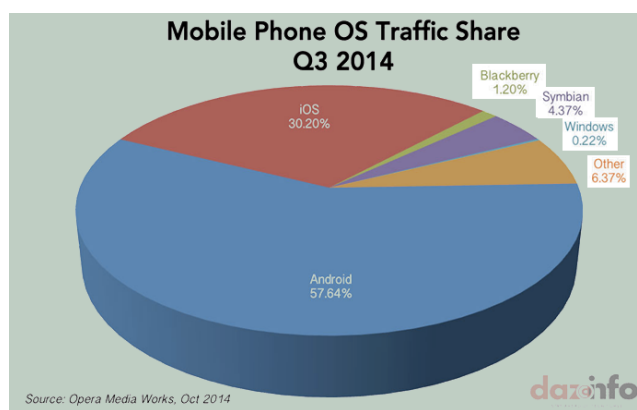
## PRINCIPAIS SISTEMAS OPERACIONAIS DE DISPOSITIVOS MÓVEIS

Existem vários sistemas operacionais para dispositivos móveis no mercado, merecendo destaque:



É público e notório que o Android tornou-se o sistema operacional mais usado em dispositivos móveis.

Veja a ilustração de um levantamento realizado no 3º bimestre de 2014:



## Sistema Operacional Android

Um pouco de história...

Android é um software open-source para dispositivos móveis, baseado no sistema operacional Linux.

Foi desenvolvido por uma empresa chamada Android Inc., situada na Califórnia-EHA, e fundada por Andy Rubin, Rick Miner, Nick Sears e Chris White, inicialmente para o desenvolvimento de aplicativos para celulares.

Em 2005, foi adquirida pela Google, que manteve Andy Rubin como colaborador no desenvolvimento da plataforma Android.

Em novembro de 2007, foi criada a Open Handset Alliance(OHA), formada por várias empresas do setor e capitaneada pela Google. A OHA fortaleceu o crescimento da plataforma Android.

Finalmente, em 2008, foi lançado o primeiro dispositivo móvel, usando o sistema operacional Android, batizado de HTC T-Mobile, cuja imagem é apresentada a seguir.

## PRINCIPAIS CARACTERÍSTICAS DO ANDROID

Handset Layout	• Oferece a possibilidade de desenvolvermos layouts de qualidade para dispositivos mais simples, assim como para smartphones e tablets. Suporta inclusive 2D ou 3D.
Conectividade	• Disponibiliza suporte às tecnologias de GSM / EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC e WiMAX.
Armazenamento	• É utilizado o SQLite para o armazenamento de dados.
Suporte de mídia	• Oferece suporte as tecnologias H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF e BMP.
Mensagens	• Oferece formas de envio de mensagens SMS (texto simples) e MMS (texto, imagem, áudio e vídeo).
Multi-touch	• Tem suporte nativo para multi-touch.
Multi-tasking	• Vários aplicativos podem ser executados simultaneamente.
Suporte Java	• As aplicações desenvolvidas nesta plataforma são implementadas na linguagem Java.
Market	• É uma loja de aplicativos (gratuitos e pagos) que permite instalar diretamente no dispositivo sem a necessidade de computador.

## MÁQUINA VIRTUAL ANDROID

# Assim como na linguagem Java, a plataforma Android possui sua máquina virtual própria, denominada Dalvik Virtual Machine (DVM).

Otimizada para consumir menos memória, ela difere da Java Virtual Machine (JVM) porque os arquivos .class são convertidos para o formato .dex (Dalvik Executable), que corresponde à aplicação Android compilada, e compactados em um arquivo com extensão .apk (Android Package File) que representa a aplicação final.

A partir do Android 4.4, a DVM foi substituída pela Android Runtime (ART) que apresenta um desempenho muito superior em relação à DVM. Uma das principais diferenças entre a DVM e a ART é a forma de compilação.



Na DVM, o processo de compilação é baseado em JIT (Just in time). Como próprio nome diz, somente a parcela do código necessária para execução é compilada naquele momento. É importante lembrar que, devido somente uma parte do código ser compilada, esse processo consome menos memória e menos espaço físico no dispositivo.



Já na ART, ele é totalmente compilado na instalação do aplicativo e isso ocorre somente uma vez. Com isso, o código é muito mais rápido em sua execução porque não precisa ser compilado muitas vezes. Além disso, por demandar menor recursos de CPU, consome menos bateria.

## PLATAFORMA DO ANDROID



*"No Android, uma versão do sistema operacional é conhecida como plataforma" (LECHETA, 2015, p.43).*



Cada versão possui um código identificador (número inteiro) denominado API Level, que corresponde à versão da plataforma Android.

É importante ressaltar que o valor da API Level será usado quando forem definidos os dispositivos alvos para seu aplicativo.

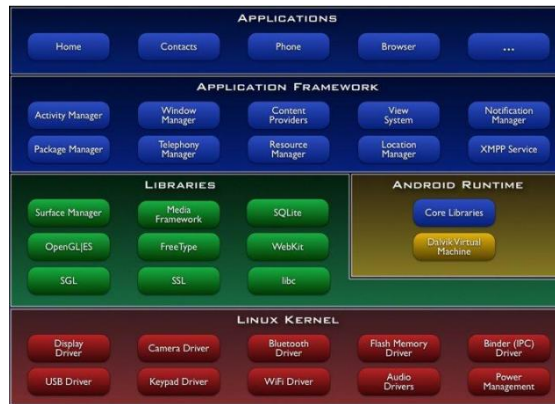
Clique [aqui \(glossário\)](#) e baixe a tabela que apresenta a relação de Api Level do Android, até o presente momento.

## ARQUITETURA ANDROID

Segundo a ilustração a seguir, observamos que a arquitetura do Android é agrupada em 3 níveis e é composta de 5 módulos:

- 1) Linux Kernel;
- 2) Libraries;
- 3) Android Runtime;
- 4) Application Framework;
- 5) Applications.





Veja maiores detalhes de cada módulo:

## Applications

É nesse módulo que se encontram os aplicativos instalados, como, por exemplo:

Cliente email;  
 Navegador;  
 Contatos;  
 Mapas;  
 Programas para SMS.

## Application Framework

Fornece vários serviços para aplicações na forma de classes Java.

Os desenvolvedores de aplicativos têm permissão para fazer uso desses serviços em suas aplicações.

Os principais serviços são:

- Activity Manager - Controla todos os aspectos do ciclo de vida da aplicação e da pilha de atividade;
- Content Provider - Permite que os aplicativos publiquem e compartilhem dados com outras aplicações;
- Resource Manager - Fornece acesso a recursos, como Strings, configurações de cores e layouts de interface do usuário;
- Notifications Manager - Permite que os aplicativos exibam alertas e notificações para o usuário;
- View System - Conjunto de views destinado a criar interfaces de usuário.

## Libraries

São as bibliotecas destinadas ao desenvolvimento Android.

Dentre elas, merecem destaque:

- Android.app - Fornece acesso à aplicação;
- Android.content - Facilita o acesso ao conteúdo, publicação e mensagens entre aplicativos e componentes dos aplicativos;
- Android.database - Usado para acessar os dados publicados por provedores de conteúdo e inclui classes de gerenciamento de banco de dados SQLite;
- Android.opengl - Uma interface Java para os gráficos OpenGL;
- Android.os - Fornece acesso aos serviços do sistema operacional padrão, incluindo mensagens, serviços do sistema e comunicação entre processos;
- Android.text - Usado para processar e manipular texto em uma tela do dispositivo;
- Android.view - Construção das interfaces com o usuário do aplicativo;
- Android.widget - Uma rica coleção de componentes de interface do usuário pré-construídos, tais como botões, etiquetas, exibições de lista, gerenciadores de layout, botões de rádio etc.

## Android Runtime

Módulo que fornece a Dalvik Virtual Machine, que é uma espécie de Java Virtual Machine, especialmente projetado e otimizado para o Android.

## Linux kernel

É o nível de abstração do hardware do dispositivo que proporciona o gerenciamento de memória, de energia, de processos, de segurança, dentre outros;

Possui drivers essenciais do hardware.

## COMPONENTES DE UM APLICATIVO ANDROID

# Componentes são blocos de construção essenciais a um aplicativo Android.

O arquivo AndroidManifest.xml, que é único em cada aplicação, descreve cada componente da aplicação e como eles interagem.

Nele constam todas as configurações necessárias para executar a aplicação, como, por exemplo, o nome do pacote utilizado, o nome das classes de cada activity, as permissões que o aplicativo possui, qual a versão mínima da API Android, dentre outras configurações.



Os principais componentes que podem ser utilizadas dentro de uma aplicação Android são:

## Activity

É considerado o componente base de uma aplicação Android. Por isso, é o mais utilizado.

Consiste em uma classe gerenciadora de UI (Interface do usuário). Representa uma única tela do usuário.

Tanto o fluxo da aplicação como eventos de tela são de sua responsabilidade. Isso não significa que todo aplicativo precisa ter uma interface do usuário, mas, se tiver, precisará de, pelo menos, uma Activity.

**Exemplo:**

## Intent

Uma Intent (intenção) é uma descrição abstrata de uma operação a ser executada. Ela pode ser utilizada para iniciar uma Activity, "ativar" um broadcast, enviar uma mensagem para uma aplicação que roda em outro processo etc. Uma Intent faz parte da arquitetura do Android e é um conceito básico que deve ser dominado por todos que desejam programar para Android.

Sendo assim, em linguagem mais humana, uma Intent representaria uma "Mensagem", um pedido que é encaminhado ao sistema operacional. O sistema pegará a mensagem, verificará qual é a "Intenção da mensagem" e tomará uma decisão que pode ser desde abrir uma página na Web (um browser abrirá e a página será exibida), fazer uma chamada telefônica ou iniciar uma Activity.

## Service

Quando o aplicativo possuir um ciclo de vida mais longo, normalmente colocamos em um Service.

Conhecido por ser um componente de background, responsável pelo processamento em

segundo plano associado a uma aplicação.

Geralmente é utilizado para realizar tarefas de sincronização, como, por exemplo, sincronização de dados em segundo plano.

**Exemplo:**

## BroadCast Receiver

Lidam com a comunicação entre sistema operacional e aplicativos Android. São responsáveis por tratar eventos do sistema ou de outras aplicações.

Se um aplicativo precisar receber e responder a qualquer evento global, precisa registrar-se como um BroadCast Receiver.

**Exemplo:**

## Content Provider

Responsável pela gestão de dados e banco de dados.

Além de permitir o compartilhamento de dados entre aplicações, centraliza as rotinas de armazenamento e recuperação em um único local, provendo às aplicações dados.

Em outras palavras, podemos afirmar que corresponde a uma abstração de dados para seus clientes.

É importante lembrar que o *Content Provider* pode acessar qualquer forma de armazenamento de dados existente na plataforma Android, entre eles arquivos, banco de dados *SQLite* ou até mesmo mapas *hash* de memória, caso não precise persistir dados.

**Exemplo:**

## COMPONENTES ADICIONAIS:

---

### FRAGMENTS

- Permitem a modularização da interface do usuário;
- Representam uma parte da UI em uma atividade;
- Assim como o Activity, possuem seu próprio ciclo de vida e podem ser definidos como fragmento de uma tela em um aplicativo Android;
- São uma espécie de “subatividade” que pode ser reutilizada em diferentes atividades.

### VIEWS

- Elementos de interface do usuário que são desenhados na tela, como botões, quadros de texto e até mesmo gerenciadores de layouts;
- São a classe-pai de todos os componentes visuais.

### LAYOUTS

- Controlam o formato de tela e a aparência da interface do usuário, sendo possível serem criados com declaração de elementos XML ou através de programação Java.

## INTENTS

- Representam uma ação que a aplicação deseja executar. Isso se dá através de mensagem (Broadcast) enviada ao sistema operacional, que decidirá a realização ou não da ação, a partir do conteúdo dessa mensagem;
- É importante lembrar que se constituem uma intenção, como próprio nome diz. Não há certeza de que a aplicação será executada.

## MANIFEST

- Arquivo que possui informações essenciais referentes à configuração para execução da aplicação, como, por exemplo, nome do pacote utilizado, nome das classes de cada activity e outros;
- Deve estar na pasta raiz do projeto.

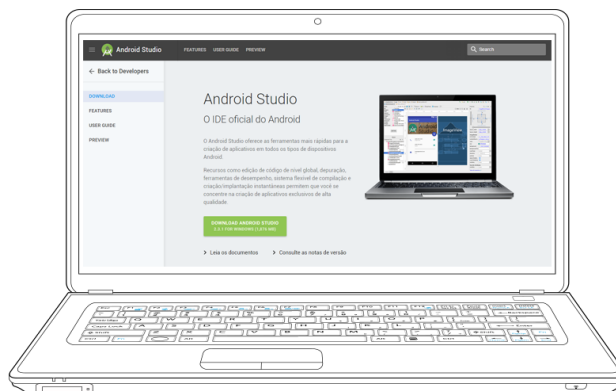
## PRINCIPAIS IDEs:

A IDE é um ambiente integrado, que acelera o desenvolvimento de aplicativos durante a fase de programação.

Abaixo os 3 IDEs mais populares do mercado, para a plataforma Android:

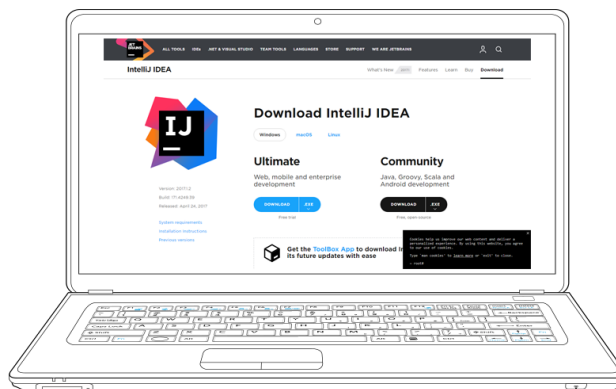
### Android Studio:

<https://developer.android.com/sdk/index.html>



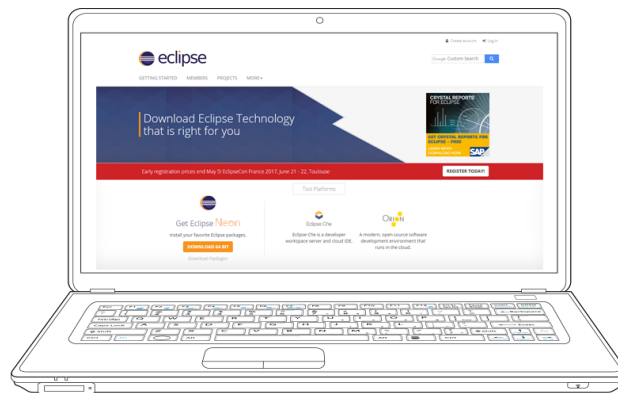
### IntelliJ:

<https://www.jetbrains.com/idea/download/>



### Eclipse:

<https://eclipse.org/downloads/>



## Atenção

, Em nossas aulas usaremos a IDE Android Studio.

## ATIVIDADE

---

A figura abaixo ilustra os principais componentes da plataforma de desenvolvimento Android. Explique cada um dos componentes demonstrados:



## Resposta Correta

# Glossário