



Conceitos Básicos

Disciplina: Linguagem de Programação III
Professor: Rafael Brazão

Android

- O Android consiste em uma nova plataforma de desenvolvimento para **aplicativos móveis**, baseada em um sistema operacional Linux, com diversas aplicações já instaladas e, ainda, um ambiente de desenvolvimento bastante poderoso, ousado e flexível;
- **Android, Inc.** foi fundada em **Palo Alto**, California em outubro de 2003 por Andy Rubin, Rich Miner, Nick Sears e Chris White, empresários no ramo da TI
- As primeiras intenções da empresa eram de desenvolver um sistema operacional avançado para **câmeras digitais**, quando se deram conta de que o mercado destes dispositivos não era grande o suficiente. Desviaram então seus esforços para produzir um sistema operacional móvel para ser rival aos **Symbian** e **Windows Mobile**
- Desde o início pensaram como um sistema operacional de **código aberto**
- O **Google** adquiriu Android Inc. em 17 de agosto de 2005

Open Handset Alliance

- É um grupo formado por gigantes do mercado de telefonia de celulares liberados pelo google. Entre alguns integrantes do grupo estão HTC, LG, Motorola, Samsung, Toshiba, Intel, Garmin dentre outras
- O objetivo do grupo era definir uma plataforma **única e aberta** para celulares para deixar os consumidores mais satisfeitos com o produto final
- O resultado desta união foi o **primeiro Android** comercial do mercado, rodando em um **HTC Dream**, lançado oficialmente em 22 de outubro de 2008



HTC Dream

Mais sobre o Android

- O Android é uma plataforma de desenvolvimento para aplicativos móveis e contém um sistema operacional baseado em **linux**
- Cada aplicativo no Android dispara um **novo processo** no sistema operacional e toda a segurança é baseada na segurança do linux
- Em 2007 ocorreu o lançamento da versão beta do primeiro **SDK** para Android
- Em 2014 o android já possuía cerca de 81% do mercado de celulares vendidos no mundo
- O Android teve várias versões com nome de **doces**:
 - Cupcake (1.5), Donut (1.6), Eclair (2.0), Froyo (2.2), Gingerbread (2.3), Honeycomb (3.0), Ice Cream Sandwich (4.0), Jelly Bean (4.1), KitKat (4.4), Lollipop (5.0), Marshmallow (6.0), Nougat (7.0)
- O Android é código aberto e distribuído sob licença Apache 2.0, o que quer dizer que você tem acesso aos códigos-fontes e também pode contribuir com o projeto

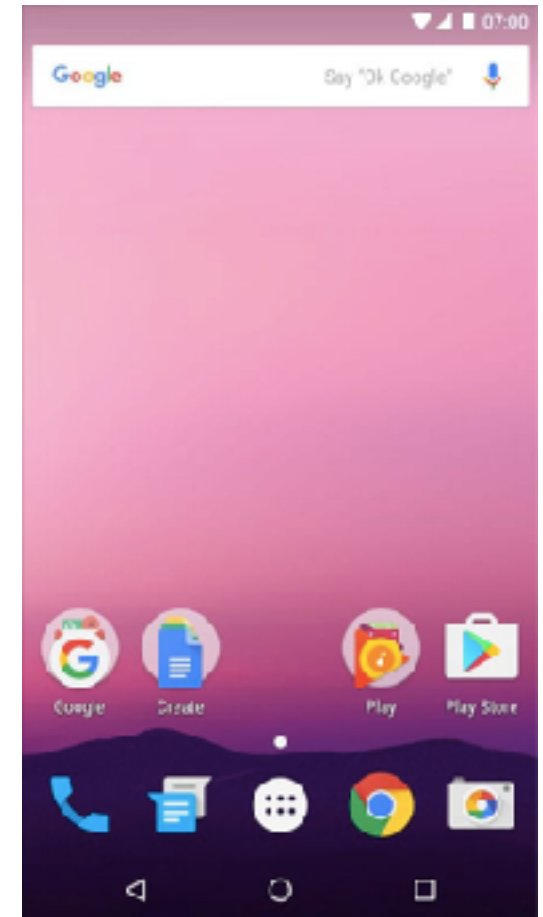
Mais sobre o Android



1.0

1.5

1.6



7.0

Outros dispositivos com Android



Video games



Android Wear



Android Auto

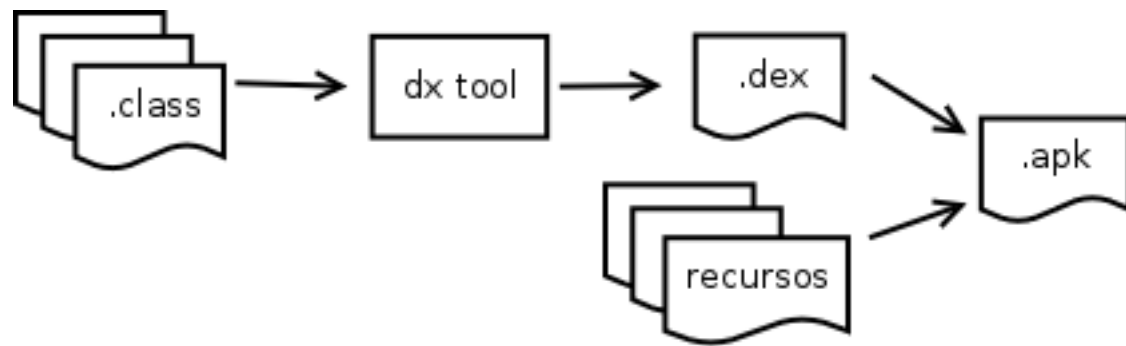
Google Play

- Para auxiliar a distribuição das aplicações do Android, além da divulgação da plataforma, foi criado o site **Android Market**
- Em 2012 a google substituiu o Android Market pela **Google Play**
 - <http://play.google.com>
- Além de **aplicativos**, a Google Play passou a oferecer livros, filmes e musicas
- Para publicar uma aplicação o desenvolvedor precisa fazer um cadastro, pagar uma **taxa de 25 dólares** e concordar com os termos de uso. Depois disto, ele estará habilitado a fazer o upload de aplicativos na Google Play
- Existem valores diferentes a serem pagos dependendo se o aplicativo for pago ou gratuito

O funcionamento do android

- Até a versão 4.4 do Android, as aplicações implementadas utilizando a linguagem Java eram executadas por meio de uma **máquina virtual**, baseada em registradores, e otimizada para consumir pouca memória, chamada **Dalvik**
- Ao contrário da máquina virtual Java que executa bytecodes, a Dalvik utiliza arquivos no formato **.dex (Dalvik Executable)** gerados a partir de classes Java compiladas
- Depois de criado, o arquivo .dex e todos os recursos utilizados na aplicação, como imagens e ícones, são adicionados em um arquivo **.apk**, que é o aplicativo propriamente dito, capaz de ser instalado em um dispositivo.
- No Android 5.0, foi introduzido o novo **Android Runtime (ART)**, que faz uso da compilação ahead-of-time (AOT), permitindo que, durante a instalação, o aplicativo seja compilado e otimizado para o dispositivo alvo

O funcionamento do android



Processo de geração do aplicativo

Configuração e uso do ambiente de desenvolvimento

- A primeira ferramenta popular para o desenvolvimento do Android foi o **eclipse** em conjunto com o plugin **ADT** (Android Developer Tools)
- Em 2014 o Google lançou a versão 1.0 da IDE **Android Studio** baseada na IDE **IntelliJ**
- Existe o site oficial com a documentação onde a ferramenta pode ser baixada:
 - <https://developer.android.com/studio>
- O Android Studio é uma ferramenta multiplataforma e atualmente estamos na versão 2.2 desta IDE
- Nesta disciplina utilizaremos o Android Studio como **padrão**
- **Construção da aplicação HelloAndroid**

Componentes de uma aplicação

- **Activity:** representa uma tela com interface gráfica capaz de promover algum tipo de interação com o usuário. Uma aplicação Android pode ser composta de diversas **activities** para fornecer um conjunto de funcionalidades para o usuário;
- **Services:** são componentes executados em segundo plano e que não dispõem de interface gráfica. Seu objetivo principal é realizar tarefas que podem consumir muito tempo para executar, sem comprometer a interação do usuário com alguma activity;
- **Content Providers:** são componentes que permitem o acesso e a modificação de dados armazenados em um banco de dados **SQLite** local, de arquivos armazenados no próprio dispositivo ou mesmo dados armazenados na web;
- **Broadcasts Receivers:** são componentes capazes de responder a eventos propagados pelo sistema operacional Android, por exemplo o nível baixo da bateria, ou eventos originados por uma aplicação, como o recebimento de uma nova mensagem de texto.

Intents

- Podem ser definidas como **mensagens** enviadas por um componente da sua aplicação (uma activity, por exemplo) para o Android, informando a intenção de **inicializar outro componente** da mesma aplicação ou de outra
- Através de **Intents**, é possível iniciar novas activities, como fazer uma busca e seleccionar um contato do telefone, abrir a aplicação de mapas com as coordenadas de localização do GPS, abrir uma página da web, tirar fotos utilizando a câmera etc., apenas reaproveitando funcionalidades já existentes, disponibilizadas pelos aplicativos instalados no aparelho
- Exemplo: criar uma intent para abrir uma página usando o navegador
 - `Uri uri = Uri.parse("http://www.android.com");`
 - `Intent intent = new Intent(Intent.ACTION_VIEW, uri);`
 - `startActivity(intent);`

Intents

- No exemplo anterior não informamos o nome da Activity que desejamos iniciar. Nesse caso temos uma **Intent implícita**, pois o android tentará descobrir a aplicação que será aberta ou irá perguntar para o usuário qual será esta aplicação
- Podemos indicar explicitamente qual a activity que será aberta, passando no construtor da Intent a classe correspondente à Activity que deve ser iniciada
 - `Intent intent = new Intent(this, OutraAtividade.class);`
 - `startActivity(intent);`
- Geralmente, as **Intent explícitas** são utilizadas apenas para interação entre componentes de uma mesma aplicação, já que é necessário conhecer o componente que deverá ser ativado; enquanto as implícitas são usadas para ativar componentes de outra aplicação

Intents

- Podemos colocar informações extras na Intent que serão usadas posteriormente pelo componente iniciado
- O código a seguir inicia a Activity de câmera do aparelho, informando o local e nome desejado para o armazenamento da imagem capturada:
 - `Uri uri = Uri.fromFile(
• new File("/sdcard/LivroDeAndroid/hello_camera.jpg"));`
 - `Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);`
 - `intent.putExtra(MediaStore.EXTRA_OUTPUT, uri);`
 - `startActivity(intent);`

Informações de uma Intent

- **Nome do componente:** é definido pelo nome completo da classe e o nome do pacote definido no AndroidManifest.xml que representam o componente que deve ser o encarregado de tratar a Intent
- **Ação:** é uma string que define o que deve ser realizado. Existem diversas ações genéricas no Android, disponibilizadas como constantes na classe Intent. Alguns exemplos de constantes são:
 - ACTION_CALL indica que uma chamada telefônica deve ser realizada
 - ACTION_VIEW indica que algum dado deve ser exibido para o usuário
 - ACTION_EDIT indica que se deseja editar alguma informação
 - ACTION_SENDTO indica que se deseja enviar alguma informação
- Enquanto a Intent declara **o que** deve ser **feito**, o componente que a recebe é o responsável por definir **como** a ação será **executada**. Ou seja, para uma mesma ação, podemos ter **comportamentos distintos**, quando ela for executada por diferentes componentes

Informações de uma Intent

- **Dados:** são representados por meio de uma URI e, a partir dela, a aplicação decide o que deve ser feito
 - Exemplo:
 - `Uri uri = Uri.parse("content://contacts/people/");`
 - `Intent intent = new Intent(Intent.ACTION_VIEW, uri);`
 - `startActivity(intent);`
- **Informações extras:** são quaisquer outros dados necessários para que o componente execute a ação apropriadamente. Elas podem ser informadas por meio dos extras da Intent
 - `intent.putExtra(MediaStore.EXTRA_OUTPUT, uri);`
- **Categoria:** representada apenas por uma string, serve como informação adicional para auxiliar o Android na escolha de qual componente é o mais adequado para receber a Intent

Intent Filters

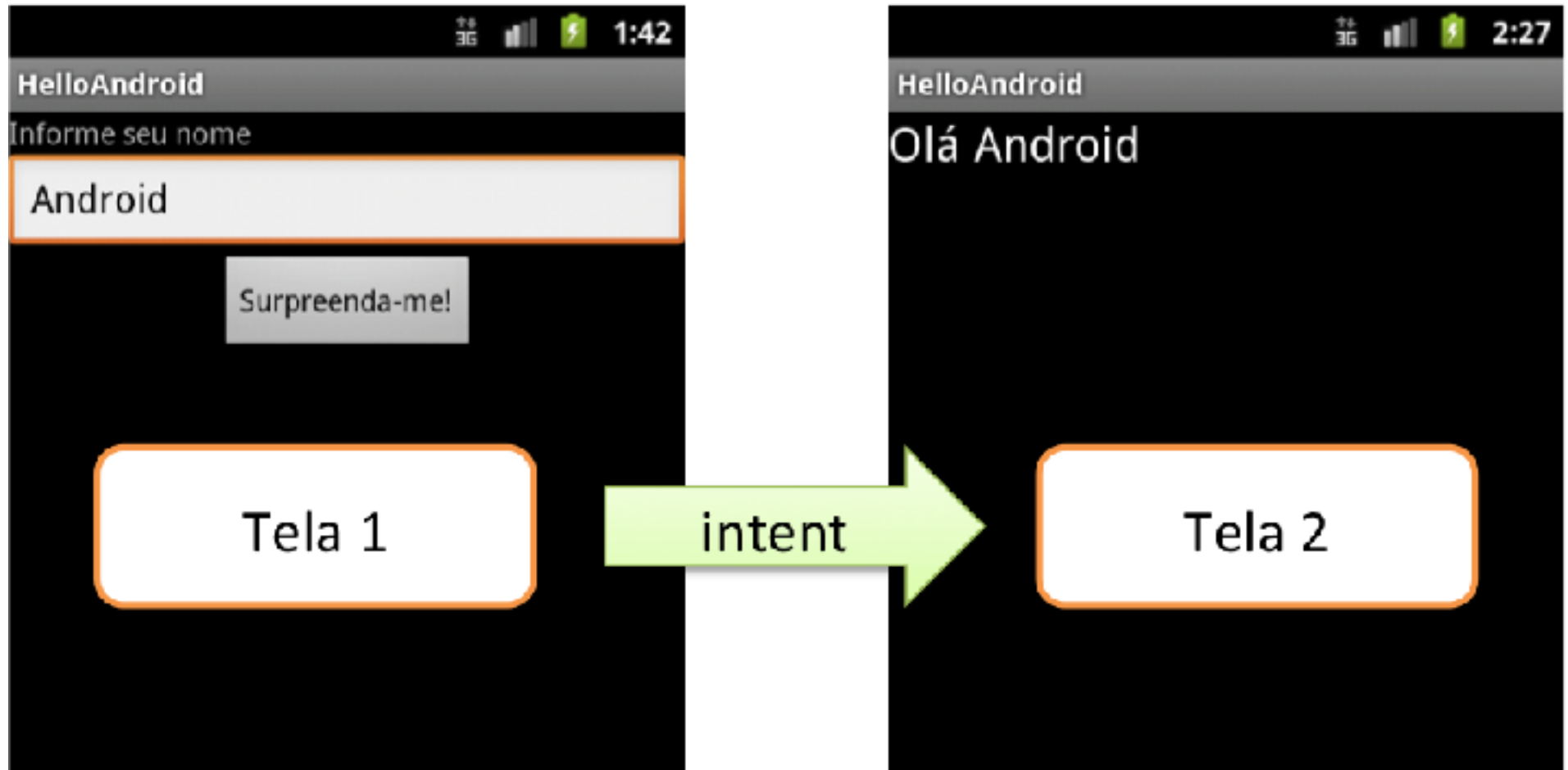
- A definição de quais ações um componente está apto a responder, bem como a quais categorias ele pertence e quais dados ele sabe tratar, é realizada por meio de Intent Filters, que são configurados no arquivo AndroidManifest.xml

```
<activity
    android:name=".MainActivity"
    android:label="@string/title_activity_main" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category
            android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

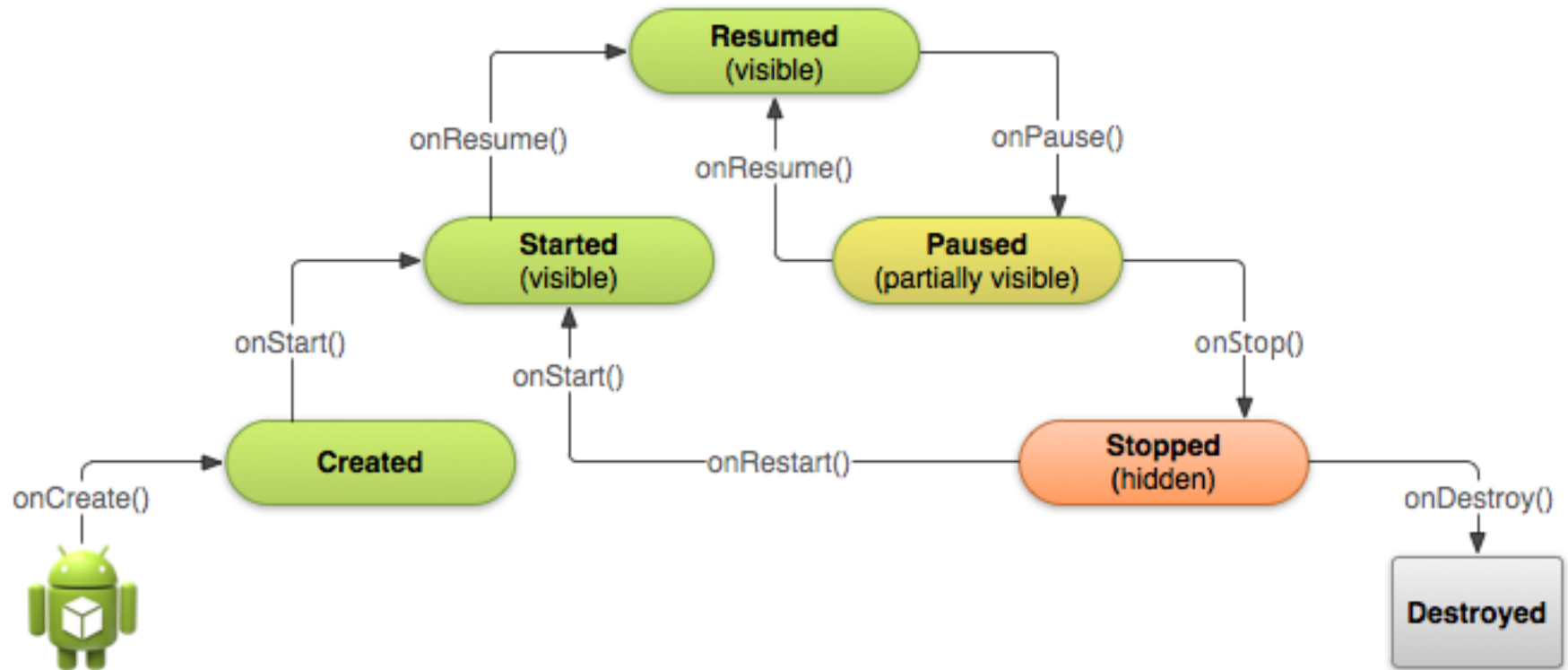
- Os intent filters podem ainda declarar, além da ação e da categoria, os tipos de dados com os quais o componente é capaz de lidar, como uma imagem, por exemplo:
 - <data android:mimeType="image/*" />



Construção da nossa primeira Intent



O ciclo de vida de uma Activity



O ciclo de vida de uma Activity

- Sempre que a Activity muda de estado, o Android aciona um método (callback) correspondente.
- Assim que o usuário inicia uma aplicação, o Android cria a atividade principal que está declarada no AndroidManifest.xml e invoca o seu método onCreate
- Em seguida, o Android invoca os métodos onStart e, logo após, o onResume.
- A Activity torna-se visível para o usuário no estado Started e assim permanece até os métodos onPause (visível parcialmente) ou onDestroy serem chamados.
- Quando ela está no estado Resumed, dizemos que ela está no foreground e pode realizar interação com o usuário.

O ciclo de vida de uma Activity

- Para descrever o comportamento da aplicação em qualquer dos estados que ela possuir devemos sobrescrever o respectivo método do estado
- Exemplo: onCreate

```
@Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // A activity está sendo criada
    }
```

Fim da aula

- Obrigado!