

# Curso Android para *Iniciantes*



Lana Mesquita  
Cristiane Mayara  
Brunno Melo

# Contatos



- Lana Mesquita  
[lanabeatriz.mesquita@gmail.com](mailto:lanabeatriz.mesquita@gmail.com)
- Cristiane Mayara  
[cris.mayara@gmail.com](mailto:cris.mayara@gmail.com)
- Brunno Melo  
[brunnomelo@gmail.com](mailto:brunnomelo@gmail.com)
- Grupo Facebook:  
<https://www.facebook.com/groups/CursoAndroidCAET2014>

# Ementa de hoje

1. Introdução ao Android
2. Fontes de Estudo
3. Versões Android
4. Configurando o Ambiente
  - Configurando o Emulador
5. Prática: Hello World
6. Estrutura de um projeto Android (Prática)
  - Android Manifest
  - pasta res
  - R.java
7. LogCat
8. Toast
9. Utilizando o adb



# Introdução

- Android é um sistema operacional voltado para dispositivos móveis
- SO utilizado em smartphones, smart TVs, wear gadgets, tablets, Google Glass
- Sistema operacional baseado no kernel do linux
- Possui caráter open source, sob licença Apache



# Introdução

- Desenvolvido pela OHA - Open Handset Alliance

- Aliança entre varias empresas: Google, HTC, LG, Samsung, Motorola..

- Objetivo: criar padrões abertos para telefonia móvel, uma plataforma padrão que engloba todas as novas tendências de mercado.



open  
handset  
alliance

# Open Handset Alliance(OHA)

## Mobile Operators



open handset alliance

## Handset Manufacturers



## Semiconductor Companies



## Software Companies



## Commercialization Companies



# Características da Plataforma

- Framework de Aplicação permitindo reuso de componentes
- **Máquina Virtual Dalvik** otimizada para dispositivos móveis
- Gráficos Otimizados por meio de uma biblioteca de gráficos 2D e gráficos 3D; o Baseados na especificação **OpenGL ES (1.0-3.0)**



# Características da Plataforma

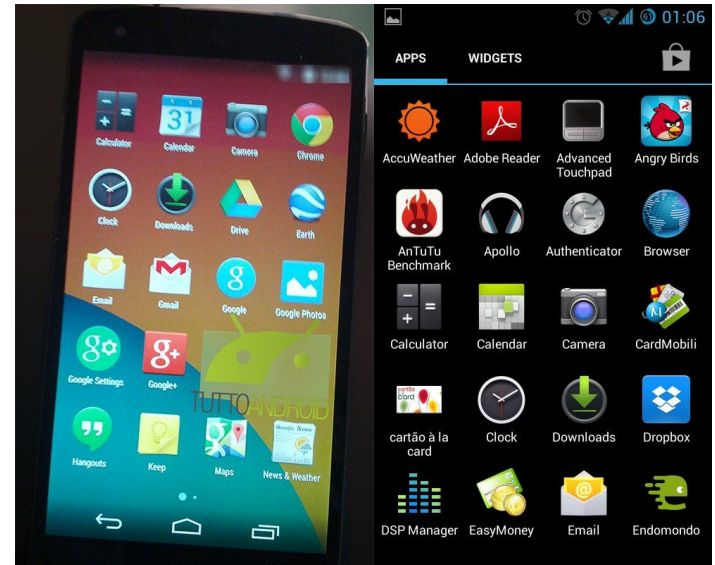
- SQLite, SGBD nativo para armazenamento de dados em formato de estruturas relacionais
- Suporte para mídias de áudio (formatos MP3, AAC, AMR), vídeo (MPEG4 e H.264) e Imagens (formatos JPG, PNG, GIF)
- Telefonia GSM (dependente de hardware)
- Bluetooth, EDGE, 3G, e WiFi, Wifi-Direct





# Características da Plataforma

- Câmera, GPS, bússola, acelerômetro, magnetômetro (dependente de hardware), NFC
- Poderoso Ambiente de Desenvolvimento, incluindo um emulador de dispositivo
- Ferramentas para depuração, analisador de memória e performance; e um plugin para a IDE Eclipse

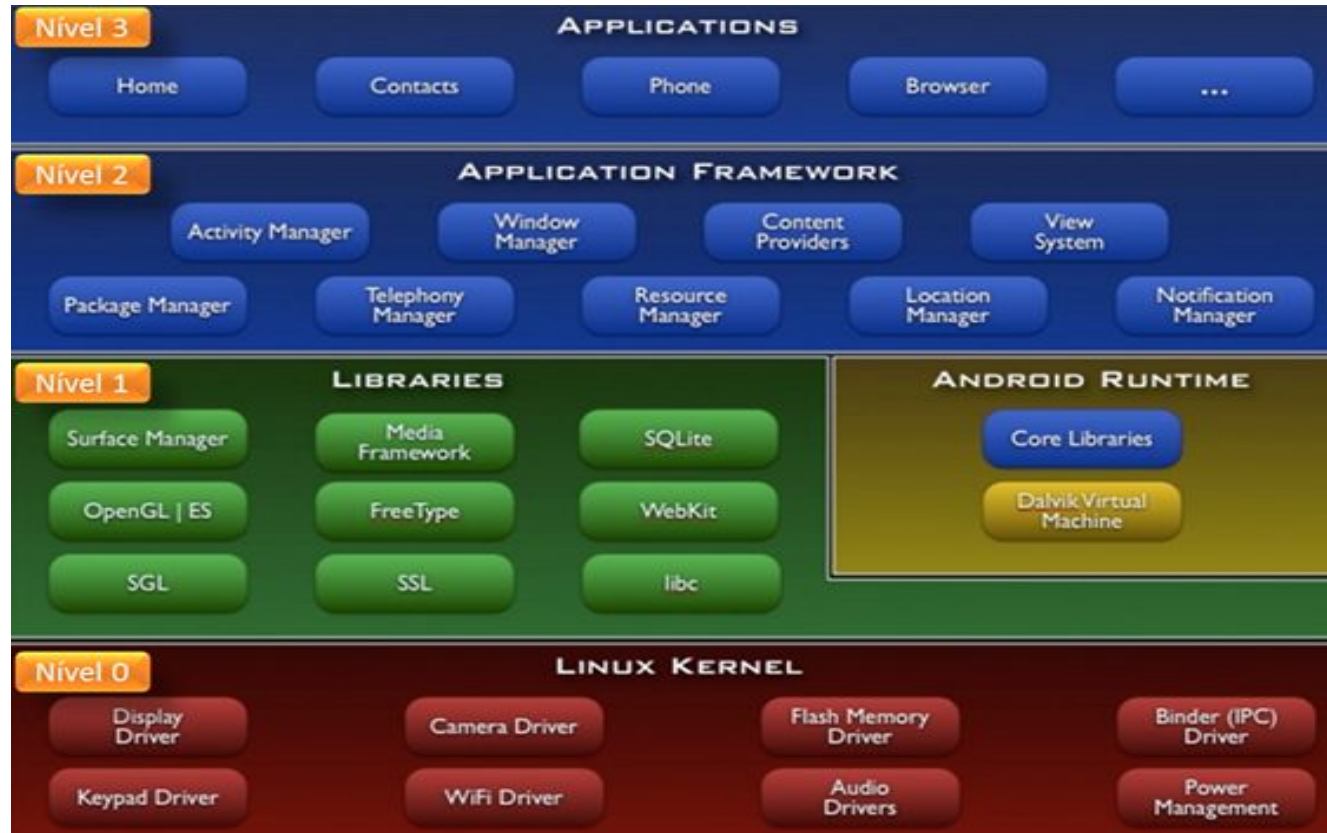


# Características herdadas do linux

- Sistema de arquivos
- Kernel, que faz a interface com o hardware, contendo os drivers para comunicação
- Servidores de terminais
- O próprio sistema gerencia a memória, os processos, threads, redes e drivers
- Segurança de arquivos e pastas



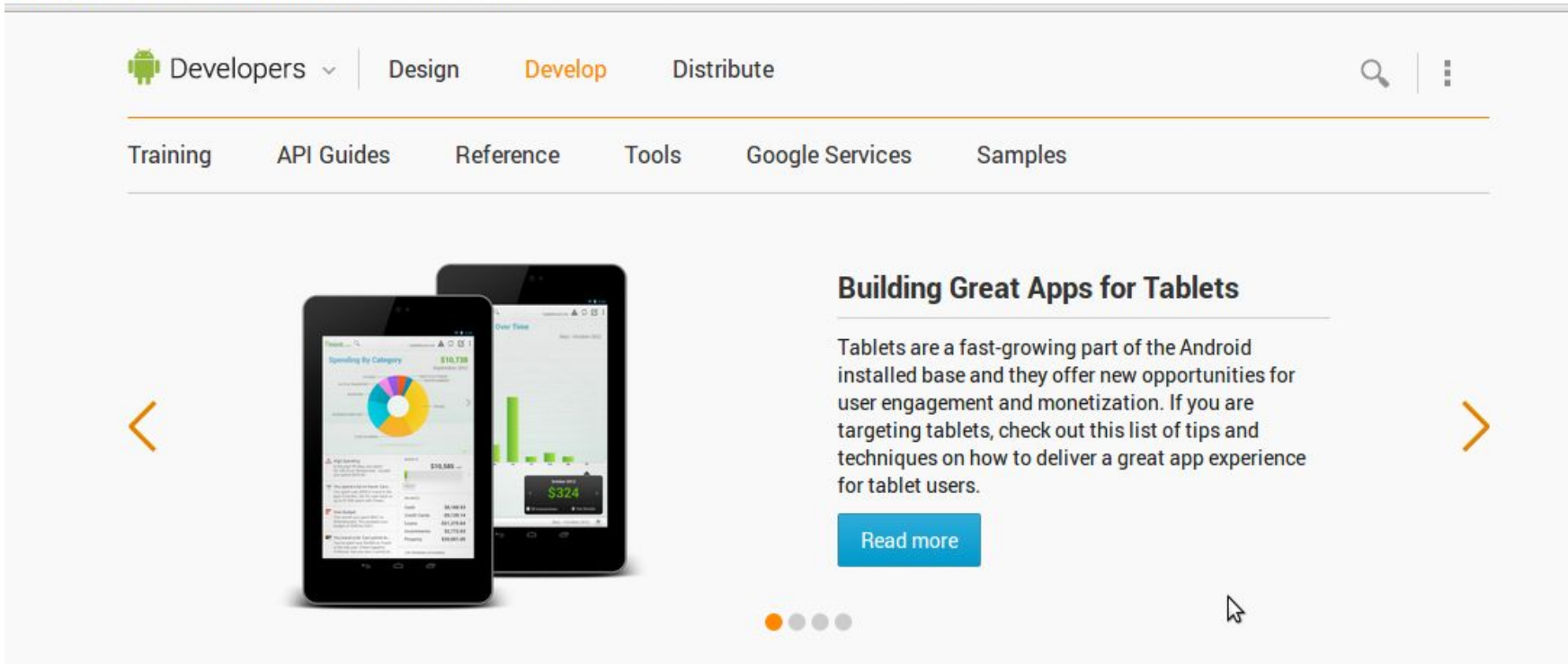
# Arquitetura Android



# Fontes de Pesquisa

- Programação Orientada à Google

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



The screenshot shows the Android Developer website. The top navigation bar includes the Android logo, 'Developers', 'Design', 'Develop' (highlighted in orange), and 'Distribute'. Below this is a secondary navigation bar with 'Training', 'API Guides', 'Reference', 'Tools', 'Google Services', and 'Samples'. The main content area features a large image of two tablets displaying financial data. To the right of the image is a section titled 'Building Great Apps for Tablets' with a paragraph of text and a 'Read more' button. Navigation arrows are visible on the left and right sides of the main content area.

Developers | Design | **Develop** | Distribute

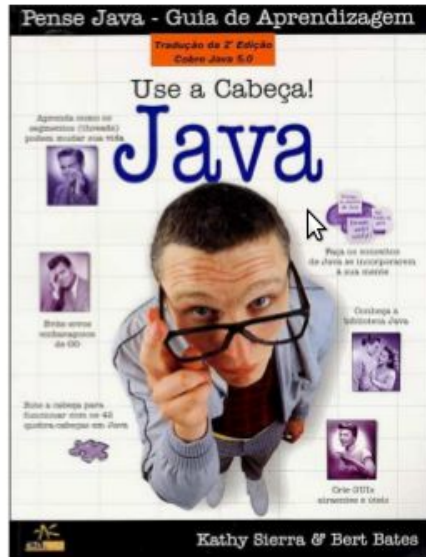
Training | API Guides | Reference | Tools | Google Services | Samples

## Building Great Apps for Tablets

Tablets are a fast-growing part of the Android installed base and they offer new opportunities for user engagement and monetization. If you are targeting tablets, check out this list of tips and techniques on how to deliver a great app experience for tablet users.

[Read more](#)


# Fontes de Pesquisa



# Fontes de Pesquisa

pt.stackoverflow.com/questions/tagged/android

StackExchange ▾ registrar log-in tour ajuda ▾

 **stackoverflow** beta  
em Português

Perguntas Tags Tour Usuários [Faça uma pergunta](#)

Perguntas com tags info **mais recentes** frequentes votos ativo sem resposta

Android é um Sistema Operacional da Google para dispositivos móveis. Utilize esta tag para referenciar perguntas relacionadas à programação para Android e seja relevante apenas para este sistema operacional. Use a tag escrevendo [android-assunto] ao invés de [assunto] isoladamente.

[saiba mais...](#) [principais usuários](#) [sinônimos](#)

2  
votos


1  
resposta

23 visitas

## Aplicativo android fecha assim que carrega (NoClassDefFoundError)

Eu instalei o Android SDK tudo certinho, instalei os drivers USB do meu Motorola Moto G. Tá tudo certo, tanto que quando vou executar a aplicação diretamente do Eclipse, ele acha o meu dispositivo ...

[java](#) [android](#)

 **Andrey Luiz**  
24 ● 4

perguntada 15 horas atrás

270  
perguntas com tags  
[android](#) [about](#) »

Tags relacionadas

[java](#) × 121

[sqlite](#) × 15

[listview](#) × 14

[eclipse](#) × 14

[google-maps](#) × 13

[android-studio](#) × 9

[facebook](#) × 7

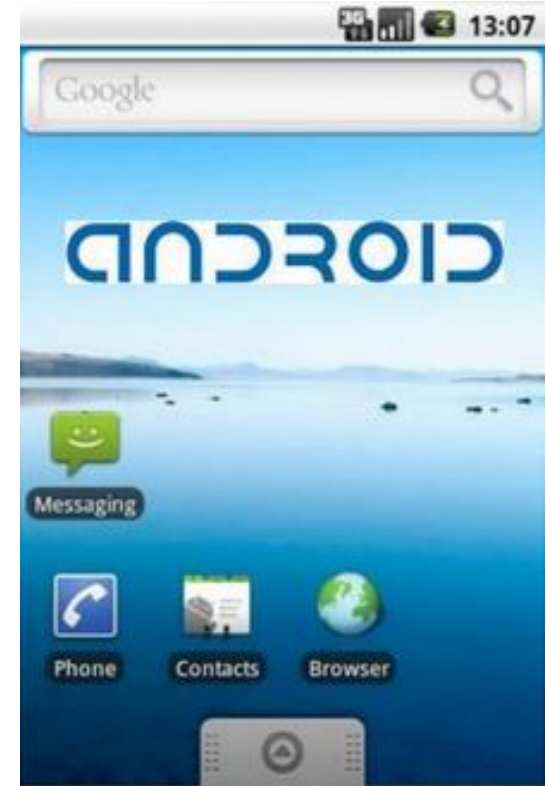


# Versões *Android*



# Versões *Android*

- Android era o nome de uma startup que desenvolvia o S.O.
- Comprada e financiada pela Google.
- Mantida pela Open Handset Alliance.
- Kernel Linux.
- Gratuito para o fabricante (Licença Apache 2)
- Integração com os serviços GOOGLE
- Google Play Services API
- Expectativa de 1 bilhão de usuários em 2014





Welcome ▾

KitKat ▾

Jelly Bean ▾

Ice Cream  
Sandwich ▾

Honeycomb ▾

Gingerbread ▾

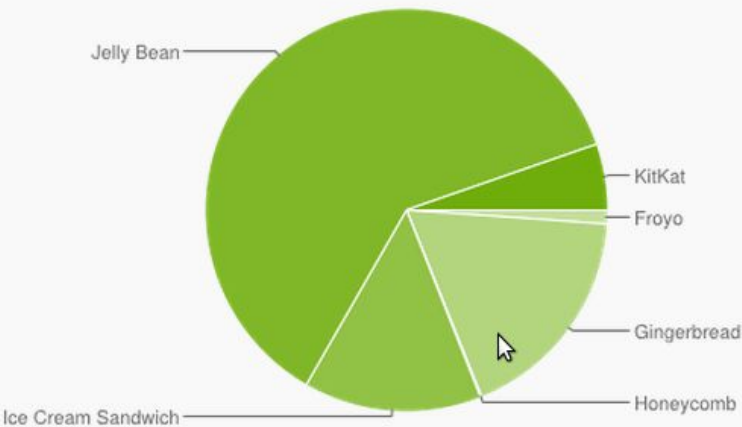
Dashboards

## Platform Versions

This section provides data about the relative number of devices running a given version of the Android platform.

For information about how to target your application to devices based on platform version, read [Supporting Different Platform Versions](#).

Version	Codename	API	Distribution
<a href="#">2.2</a>	Froyo	8	1.1%
<a href="#">2.3.3 - 2.3.7</a>	Gingerbread	10	17.8%
<a href="#">3.2</a>	Honeycomb	13	0.1%
<a href="#">4.0.3 - 4.0.4</a>	Ice Cream Sandwich	15	14.3%
<a href="#">4.1.x</a>	Jelly Bean	16	34.4%
<a href="#">4.2.x</a>		17	18.1%
<a href="#">4.3</a>		18	8.9%
<a href="#">4.4</a>	KitKat	19	5.3%



*Data collected during a 7-day period ending on April 1, 2014.  
Any versions with less than 0.1% distribution are not shown.*

**Note:** Because this data is gathered from the new Google Play Store app, which supports Android 2.2 and above, devices running older versions are not included. However, in August, 2013, versions older than Android 2.2 accounted for about 1% of devices that *checked in* to Google servers (not those that actually visited Google Play Store).



Getting Started ^

Building Your First App ▾

Adding the Action Bar ▾

Supporting Different Devices ^

Supporting Different Languages

Supporting Different Screens

[Supporting Different Platform Versions](#)

Managing the Activity Lifecycle ▾

Building a Dynamic UI with Fragments ▾

Saving Data ▾

Interacting with Other Apps ▾

Building Apps with Content Sharing ▾

Building Apps with Multimedia ▾

Building Apps with

## Supporting Different Platform Versions

While the latest versions of Android often provide great APIs for your app, you should continue to support older versions of Android until more devices get updated. This lesson shows you how to take advantage of the latest APIs while continuing to support older versions as well.

The dashboard for [Platform Versions](#) is updated regularly to show the distribution of active devices running each version of Android, based on the number of devices that visit the Google Play Store. Generally, it's a good practice to support about 90% of the active devices, while targeting your app to the latest version.

**Tip:** In order to provide the best features and functionality across several Android versions, you should use the [Android Support Library](#) in your app, which allows you to use several recent platform APIs on older versions.

### Specify Minimum and Target API Levels

The [AndroidManifest.xml](#) file describes details about your app and identifies which versions of Android it supports. Specifically, the `minSdkVersion` and `targetSdkVersion` attributes for the `<uses-sdk>` element identify the lowest API level with which your app is compatible and the highest API level against which you've designed and tested your app.

< [PREVIOUS](#)

[NEXT](#) >

#### THIS LESSON TEACHES YOU TO

1. [Specify Minimum and Target API Levels](#)
2. [Check System Version at Runtime](#)
3. [Use Platform Styles and Themes](#)

#### YOU SHOULD ALSO READ

- [Android API Levels](#)
- [Android Support Library](#)

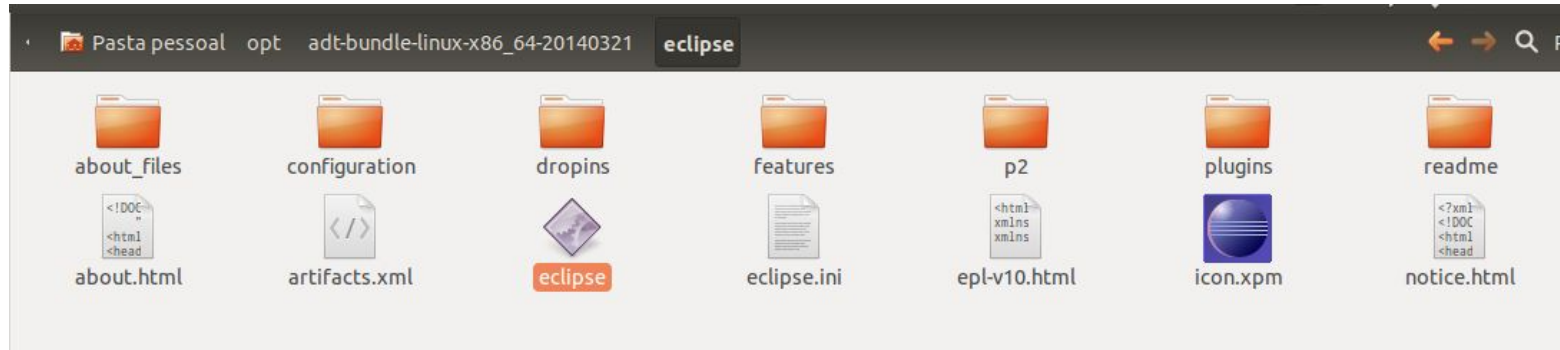
# Configurando o Ambiente

1. Faça o download do SDK:

<http://developer.android.com/sdk/>

2. Descompacte o arquivo;

3. Abra o Eclipse



# Configurando o Emulador

- ## 1. Eclipse: Menu Window -> Android Virtual Device Manager

- ## 2. New

**Edit Android Virtual Device (AVD)**

AVD Name:

Device:

Target:

CPU/ABI:

Keyboard: ☒ Hardware keyboard present

Skin:

Front Camera:

Back Camera:

Memory Options: RAM:  VM Heap:

Internal Storage:

SD Card: ☒ Size:    
☐ File:

Emulation Options: ☐ Snapshot ☐ Use Host GPU

☐ Override the existing AVD with the same name

## Se tiver problemas acessar via terminal:

```
cd ~/opt/adt-bundle-linux-x86_64-20140321/sdk/tools
./emulator-arm -avd Teste
```

3. Esperar carregar. Pode demorar muitos minutos! Tenha fé...

# Configurando o Ambiente

4. Eclipse: Menu Window -> Android SDK Manager

5. Marque as opções:

<input type="checkbox"/>	Android 4.4.2 (API 19)	
<input checked="" type="checkbox"/>	Documentation for Android SDK	19
<input checked="" type="checkbox"/>	SDK Platform	19

<input type="checkbox"/>	Android 4.3 (API 18)	
<input checked="" type="checkbox"/>	SDK Platform	18
<input checked="" type="checkbox"/>	Samples for SDK	18
<input type="checkbox"/>	ARM EABI v7a System Image	18



<input type="checkbox"/>	Android 2.3.3 (API 10)	
<input checked="" type="checkbox"/>	SDK Platform	10
<input type="checkbox"/>	Samples for SDK	10
<input type="checkbox"/>	Intel x86 Atom System Image	10
<input type="checkbox"/>	Google APIs	10
<input type="checkbox"/>	Android 2.2 (API 8)	
<input checked="" type="checkbox"/>	SDK Platform	8
<input type="checkbox"/>	Samples for SDK	8
<input type="checkbox"/>	Google APIs	8



# Configurando o Ambiente

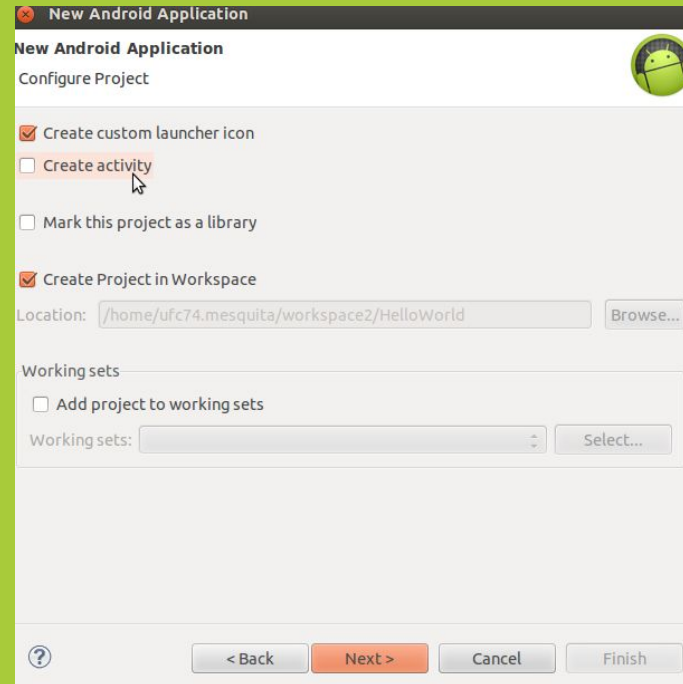
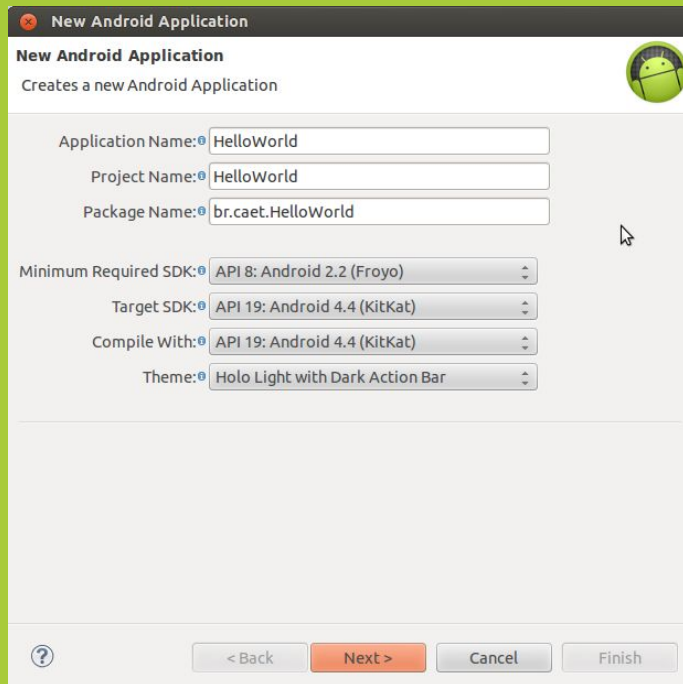
6. Instale os pacotes selecionados. Ao término da instalação verifique se os pacotes foram instalados com sucesso.

7. Pronto!



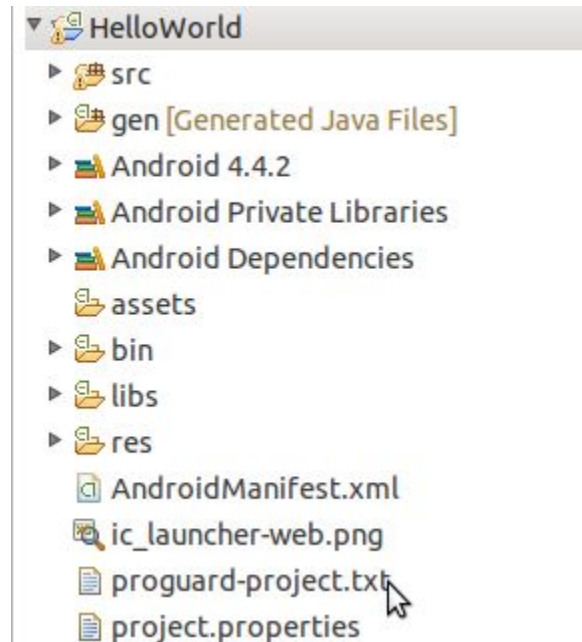
# Hello World

## 1. File -> New -> Android Project ou Android Application



<https://developer.android.com/training/basics/firstapp/creating-project.html>

# Estrutura de um projeto Android





# Estrutura de um projeto Android

## Android Manifest

- Define o **pacote java** que identifica unicamente a aplicação
- Declara as **permissões necessárias** para a aplicação interagir com a API do android e com outras aplicações
- Lista as **bibliotecas** as quais a aplicação deve estar conectada
- Define os **componentes da aplicação** - Activities, Broadcast Receivers and Content Providers.
- Nomeia as classes que implementam cada componente e suas capacidades
  - Isso permite que o sistema execute as classes de maneira correta

# Estrutura de um projeto Android

## Android Manifest

Abra o arquivo "AndroidManifest.xml", dentro da raiz do projeto.

- **package**: muito importante! Sua aplicação é conhecida pelo nome do pacote;
- **android:versionCode** e **android:versionName**: número da versão da aplicação. Primeiro em forma numérica e depois em forma String;

Dentro do tag **<application>** temos atributos importantes:

- **android:icon**: identificação do ícone da aplicação, de acordo com os nomes de imagens dentro da classe "R";
- **android:label**: título da aplicação. Todas as activities que não especificarem este atributo, herdarão o valor deste;

# Estrutura de um projeto Android

## Android Manifest

- Pode conter:

<activity>

<service>

<receiver>

<provider>

<uses-library>

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="br.caet.HelloWorld"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="19" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <activity
            android:name="br.caet.HelloWorld.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Manifest Application Permissions Instrumentation AndroidManifest.xml

# Estrutura de um projeto Android

## Android Manifest

Ao se criar uma activity, deve-se inserir tags `<activity>` manualmente (nunca esquecer) :

- `android:name`: o nome da classe (será concatenado ao string especificado no atributo "package", do tag `<manifest>`);
- Pode conter
  - `<intent-filter>`
  - `<meta-data>`

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="br.caet.HelloWorld"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="19" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <activity
            android:name="br.caet.HelloWorld.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Manifest Application Permissions Instrumentation AndroidManifest.xml

# Estrutura de um projeto Android

## Android Manifest

- Adicionando **permissões** a aplicação
  - Solicita ao usuário, uma permissão que a aplicação necessita para operar corretamente

```
<uses-permission android:name="android.permission.CAMERA" />
```

```
<uses-permission android:name="android.permission.INTERNET" />
```

```
<uses-permission
```

```
android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

- Obs: Para efeitos de filtro no play store, devemos adicionar o elemento uses-feature que esta relacionado a uma funcionalidade, por exemplo

```
<uses-feature android:name="android.hardware.CAMERA" android:required="true">
```

# Estrutura de um projeto Android

## pasta res

res/

Contem vários sub-pastas para [app resources](#). Por exemplo:

drawable/

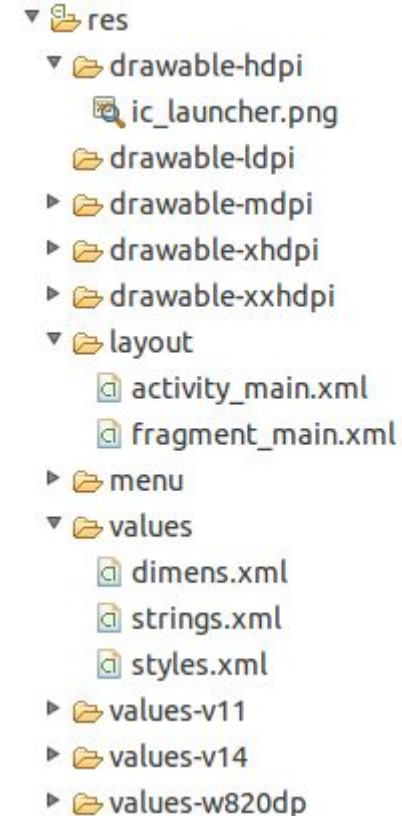
Arquivos de imagem

layout/

Arquivos xml compilados em layout

values/

Arquivos xml compilados na classe R acessados através do identificador gerado



# Estrutura de um projeto Android

## pasta res

### anim/

Arquivos xml compilados em objetos animados

### color/

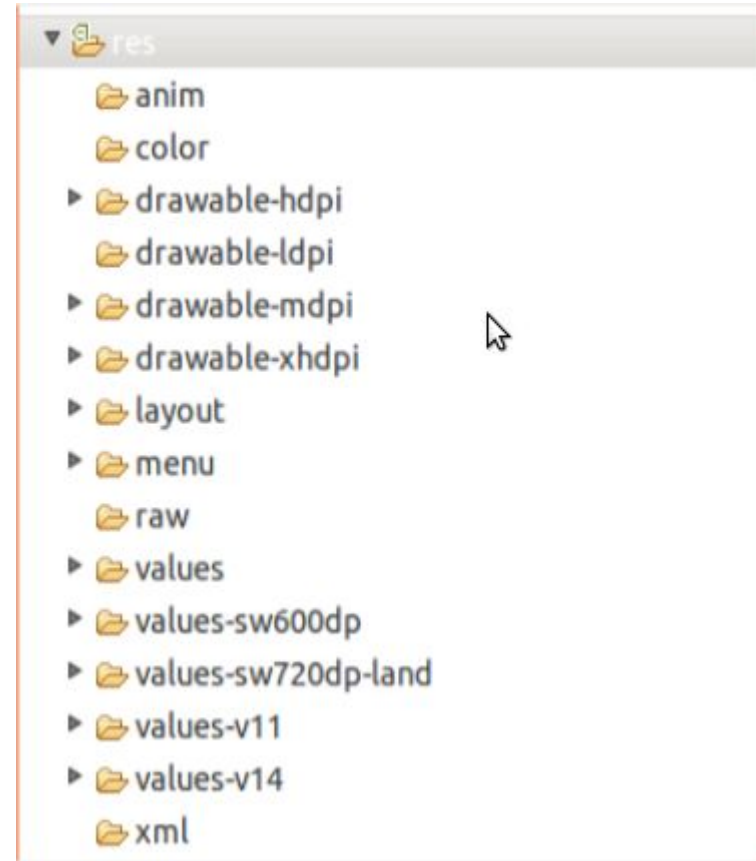
Arquivos xml que descrevem cores

### menu/

Arquivos xml que definem menus

### xml/

Arquivos xml que definem recursos em geral, como tela de preferência

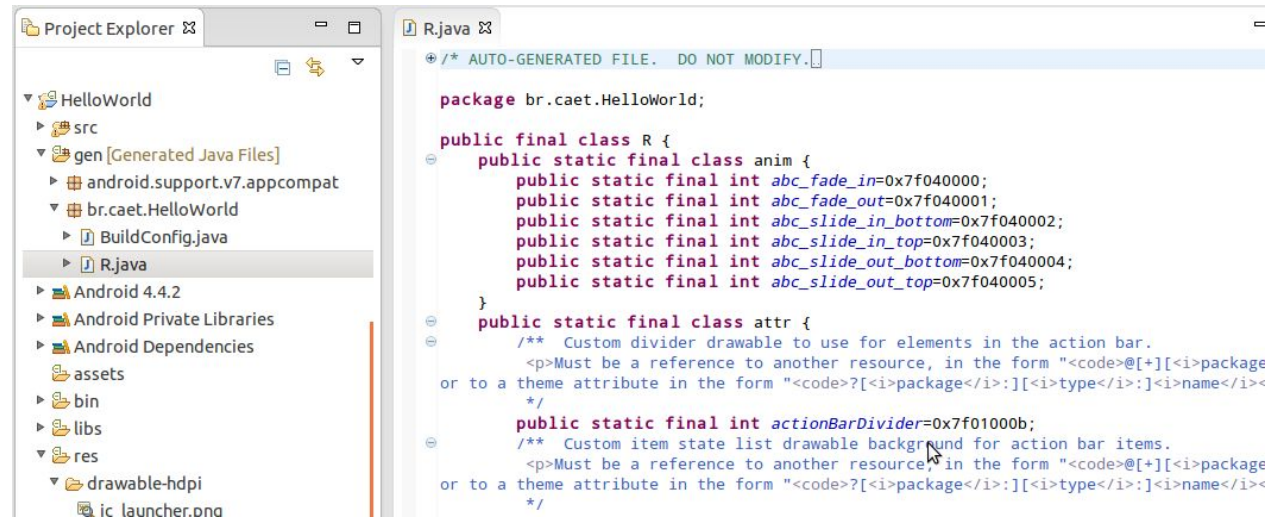


# Estrutura de um projeto Android

## R.java

- Gerada automaticamente pelo Eclipse
- Contém constantes para acessar os diversos recursos do projeto
- Nunca deve ser alterada manualmente
- Se ela não estiver no projeto existem problemas no seu projeto

Se visualmente não há problemas, podem haver problemas no AndroidManifest ou na pasta res





# Hello World

2. Criar uma Activity: Botão direito no nome do projeto -> New -> Class

**New Java Class**

Java Class  
Create a new Java class.

Source folder: HelloWorld/src Browse...

Package: br.caet.helloworld Browse...

☐ Enclosing type: Browse...

Name: MainActivity

Modifiers: ☒ public ☐ default ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass: android.app.Activity Browse...

Interfaces: Add...  
Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)  
☐ Constructors from superclass  
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))  
☐ Generate comments

? Cancel Finish



# Hello World

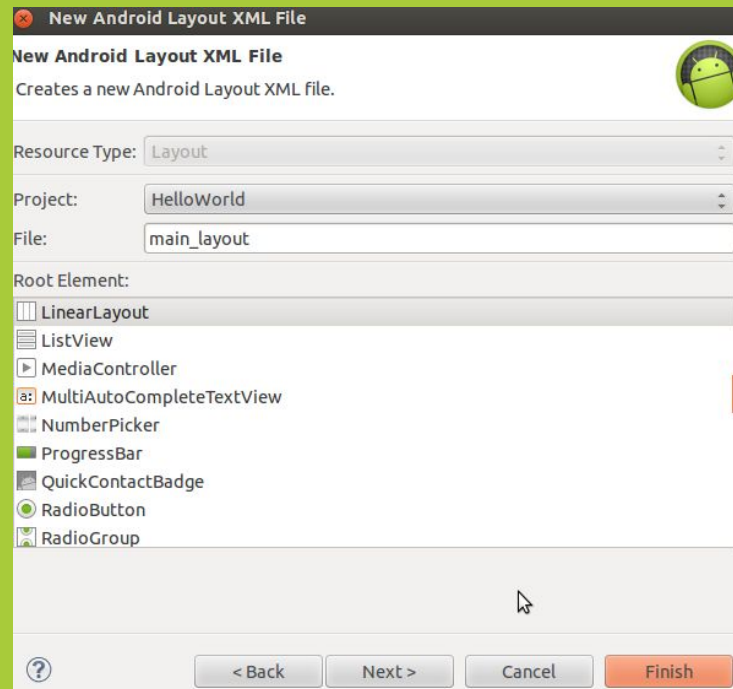
## 3. Sobrescrever métodos de Activity

```
package br.caet;  
  
import android.app.Activity;  
import android.os.Bundle;  
  
public class MainActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        // TODO Auto-generated method stub  
        super.onCreate(savedInstanceState);  
    }  
  
    @Override  
    protected void onDestroy() {  
        // TODO Auto-generated method stub  
        super.onDestroy();  
    }  
}
```



# Hello World

4. Criar Layout para a Main : botão direito no nome do projeto -> New -> Others -> Android -> Android XML Layout File



# Hello World

5. Criar um textView no layout criado.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="HelloWord" />

</LinearLayout>
```



# Hello World

## 6. Modificando o AndroidManifest:

```
package="br.caet.helloworld"

...

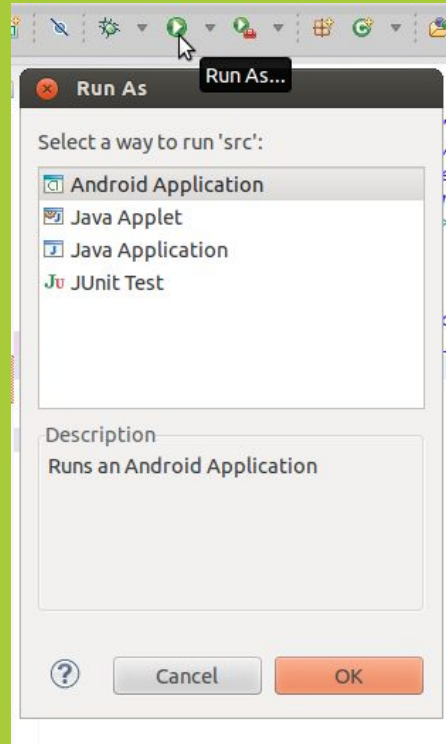
<activity
    android:name="br.caet.helloworld.MainActivity"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```



# Hello World

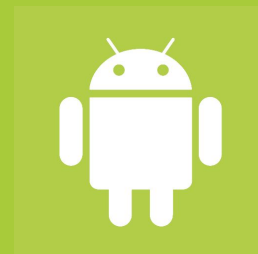
7. Rodar aplicativo!

Escolher o emulador ou  
o dispositivo!



# Hello World

Tudo isso pra:



# Hello World

Tudo isso pra:

Isso ->

:)



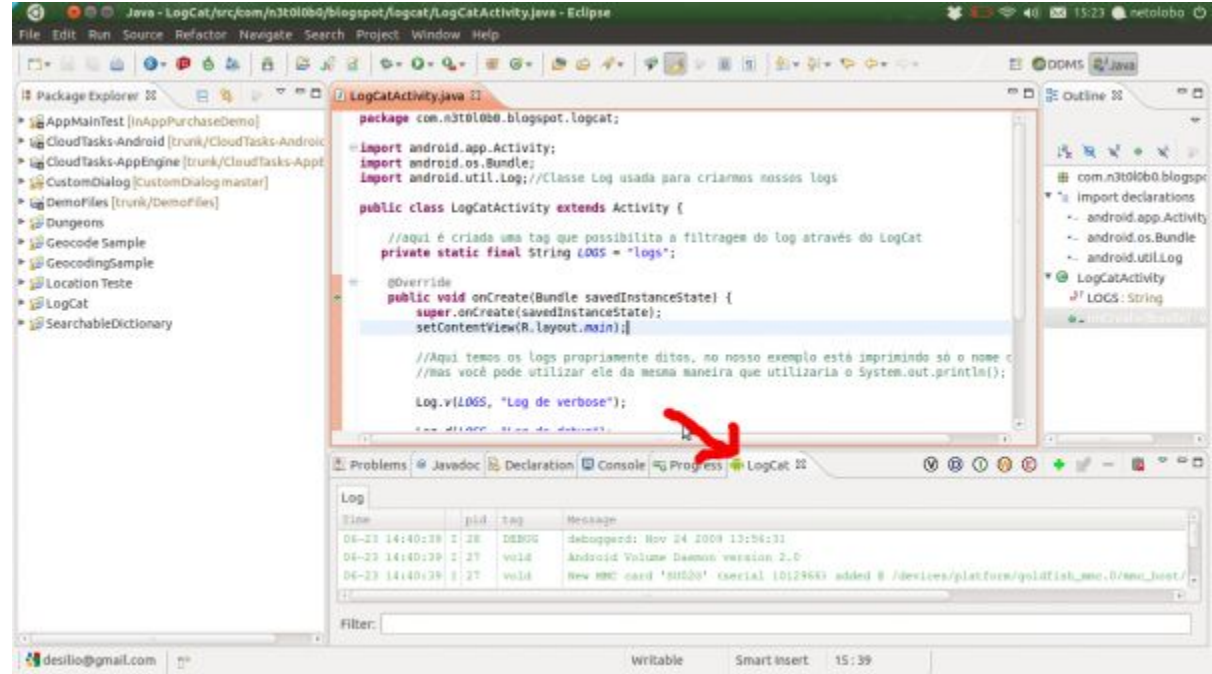


# LogCat

- é o `System.out.println()` do Android
- permite escrever mensagens no log do Android
- usa a classe `android.util.Log`

Nível	Sigla	Cor
Debug	D	azul
Verbose	V	preto
Info	I	verde
Warn	W	laranja
Error	E	vermelho

# LogCat



- Insira uma mensagem no log do HelloWorld no método onCreate.  
Log.i("teste\_info", "Teste do LogCat!");

# Toast

- Insira um toast no HelloWorld no método onCreate.

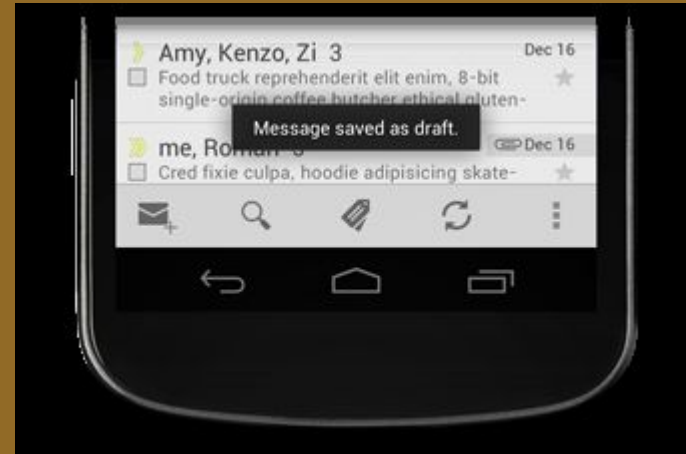
```
Context context = getApplicationContext();
```

```
CharSequence text = "Hello toast!";
```

```
int duration = Toast.LENGTH_SHORT;
```

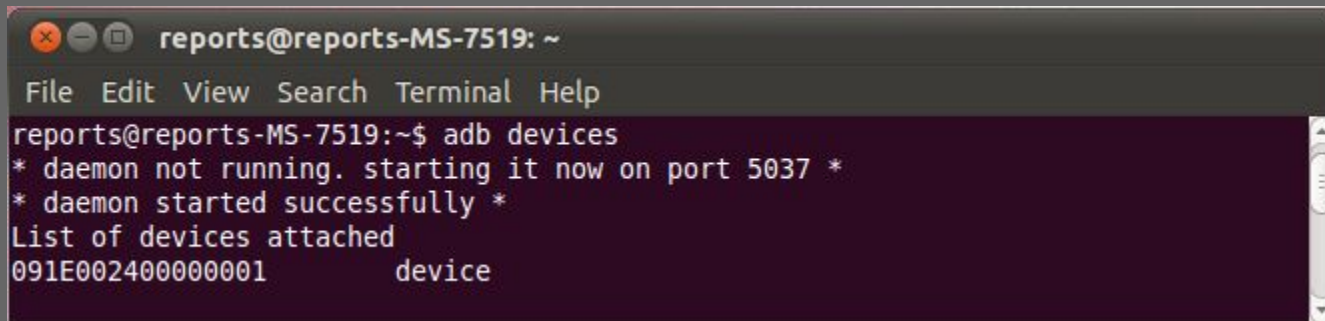
```
Toast toast = Toast.makeText(context, text,  
    duration);
```

```
toast.show();
```



# adb

- O adb (Android Debug Bridge) consiste em uma aplicação cliente-servidor que permite gerenciar o estado do dispositivo ou emulados. Alguns comandos em terminal:
  - adb push : enviar arquivos ao dispositivo
  - adb shell : entrar na estrutura de diretórios do emulador
  - adb devices : para verificar os dispositivos disponíveis

A screenshot of a terminal window titled 'reports@reports-MS-7519: ~'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal text shows the command 'reports@reports-MS-7519:~\$ adb devices' being executed. The output indicates that the adb daemon was not running and was started successfully on port 5037. It then lists the attached devices, showing one device with ID '091E0024000000001' and the label 'device'.

```
reports@reports-MS-7519: ~  
File Edit View Search Terminal Help  
reports@reports-MS-7519:~$ adb devices  
* daemon not running. starting it now on port 5037 *  
* daemon started successfully *  
List of devices attached  
091E0024000000001      device
```

# adb

- local: `.../adt-bundle-linux-x86_64-20140321/sdk/platform-tools`
- o acesso deve ser via Terminal (Ubuntu) e via Prompt de Comando (Windows)
- Dica: se o dispositivo não aparecer mate o processo e reinicie o adb utilizando os dois comandos abaixo.
  - `sudo adb kill-server`
  - `sudo adb start-server`
- local: `~/opt/adt-bundle-linux-x86_64-20140321/sdk/platform-tools`
- Dica (Ubuntu) : criar um link para adb
  - `sudo ln -s ~/opt/adt-bundle-linux-x86_64-20140321/sdk/platform-tools/adb`

# Hello World

Desafio:

