



**BEM VINDOS!**

**VENTURUS<sup>4</sup>TECH**

Módulo II  
Introdução ao Android

VENTURUS<sup>4</sup>TECH

Com Paulo Matuki da Cunha  
[paulo.cunha@venturus.org.br](mailto:paulo.cunha@venturus.org.br)

# Link para esta apresentação

<https://goo.gl/FrJH3j>

# O que é o Android?



# O “Ecossistema” Android



Fabricantes



Desenvolvedores de Apps



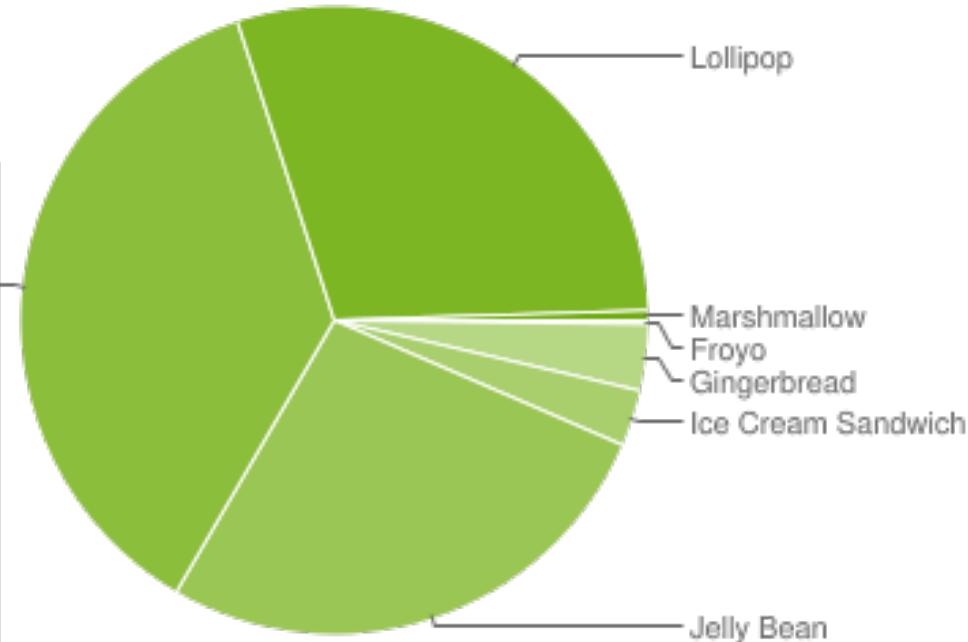
Você estará aqui!!!  
↑

# As “Sobremesas” (Versões do Android)



# Por que as Versões são importantes?

Version	Codename	API	Distribution
<a href="#">2.2</a>	Froyo	8	0.2%
<a href="#">2.3.3 - 2.3.7</a>	Gingerbread	10	3.4%
<a href="#">4.0.3 - 4.0.4</a>	Ice Cream Sandwich	15	2.9%
<a href="#">4.1.x</a>	Jelly Bean	16	10.0%
<a href="#">4.2.x</a>		17	13.0%
<a href="#">4.3</a>		18	3.9%
<a href="#">4.4</a>	KitKat	19	36.6%
<a href="#">5.0</a>	Lollipop	21	16.3%
<a href="#">5.1</a>		22	13.2%
<a href="#">6.0</a>	Marshmallow	23	0.5%



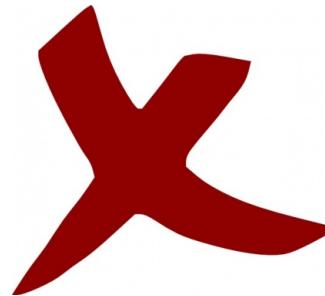
*Data collected during a 7-day period ending on December 7, 2015.*

# A linguagem oficial para apps nativos no Android



# Java – Tipagem Estática (Statically Typed)

```
boolean.isTrue = 4;  
  
int.isFalse = false;  
  
String.newString = new Object();
```



```
boolean.isTrue = false;  
  
int.isTrue = 4;  
  
String.newString = new String();
```



# Java - Orientada a Objetos

```
public class CoffeeCup implements Drinkable {  
  
    private int percentFull = 50;  
  
    private float temperature = 71.5f;  
  
    public boolean isEmpty() {  
        return percentFull == 0;  
    }  
  
    public void drink(int amountPercent) {  
        percentFull -= amountPercent;  
    }  
  
    public float getTemperature() {  
        return temperature;  
    }  
}
```

# Java – Classe ≠ Objeto

```
public class CoffeeCup implements Drinkable {  
  
    private int percentFull = 50;  
  
    private float temperature = 71.5f;  
  
    public boolean isEmpty() {  
        return percentFull == 0;  
    }  
  
    public void drink(int amountPercent) {  
        percentFull -= amountPercent;  
    }  
  
    public float getTemperature() {  
        return temperature;  
    }  
}
```

Classe

```
public class Table {  
  
    public void put2Coffee() {  
        CoffeeCup coffee1 = new CoffeeCup();  
  
        CoffeeCup coffee2 = new CoffeeCup();  
    }  
}
```

Objeto

# Java – Tipos de Dados

- **int**: Um valor inteiro, sem casas decimais, incluindo zero e números negativos
- **float**: Um valor de ponto flutuante com casas decimais ocupando até 32 bits. Não é recomendado para representar valores monetários (BigDecimal é o correto)
- **boolean**: Apenas os valores “true” ou “false”. Não podem ser usados 0 ou 1 como em C.
- **char**: Um caractere único como “a” ou “B” ou “%”.
- **String**: Um grupo de caracteres que juntos formam um texto como “Exemplo de String”
- Outros tipos: **byte, short e long**

Leitura completa sobre tipos: <http://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

# Java – Variáveis

```
String title = "Java Basics for Android";  
1           2       3           4           5
```

1. Tipo do dado da variável
2. Nome da variável
3. Operador de atribuição. Significa que o valor da direita será atribuido a variável na esquerda
4. O texto a ser atribuido a variável “title”

# Java – Métodos

```
public class View implements Drawable.Callback, KeyEvent.Callback,  
    AccessibilityEventSource {  
  
    (...)  
  
    public void setPadding(int left, int top, int right, int bottom) {  
        resetResolvedPaddingInternal();  
  
        mUserPaddingStart = UNDEFINED_PADDING;  
        mUserPaddingEnd = UNDEFINED_PADDING;  
  
        mUserPaddingLeftInitial = left;  
        mUserPaddingRightInitial = right;  
  
        mLeftPaddingDefined = true;  
        mRightPaddingDefined = true;  
  
        internalSetPadding(left, top, right, bottom);  
    }  
  
    (...)  
}
```

# Java – Métodos

```
// Em outro arquivo do projeto...

View myView = new View(this.getContext());

myView.setPadding(10, 30, 10, 30);
```

# Java – Interfaces

```
public class CoffeeCup implements Drinkable {  
  
    private int percentFull = 50;  
  
    private float temperature = 71.5f;  
  
    public boolean isEmpty() {  
        return percentFull == 0;  
    }  
  
    public void drink(int amountPercent) {  
        percentFull -= amountPercent;  
    }  
  
    public float getTemperature() {  
        return temperature;  
    }  
}
```



# Java – Interfaces

```
public class CoffeeCup implements Drinkable {  
  
    private int percentFull = 50;  
  
    private float temperature = 71.5f;  
  
    public boolean isEmpty() {  
        return percentFull == 0;  
    }  
  
    public void drink(int amountPercent) {  
        percentFull -= amountPercent;  
    }  
  
    public float getTemperature() {  
        return temperature;  
    }  
}  
  
public interface Drinkable {  
  
    public void drink(int amountPercent);  
}
```

# Java – Muito, muito mais!!!

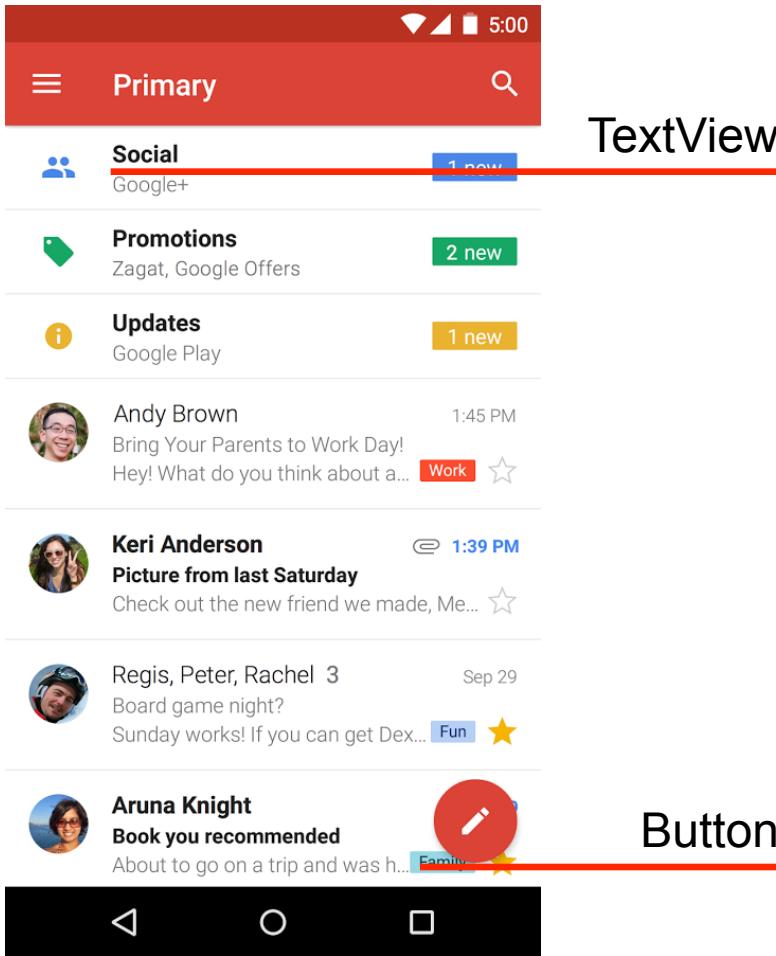
- Herança
- Polimorfismo
- Orientação a Objetos
- Controle de acesso
- Interfaces
- Threads



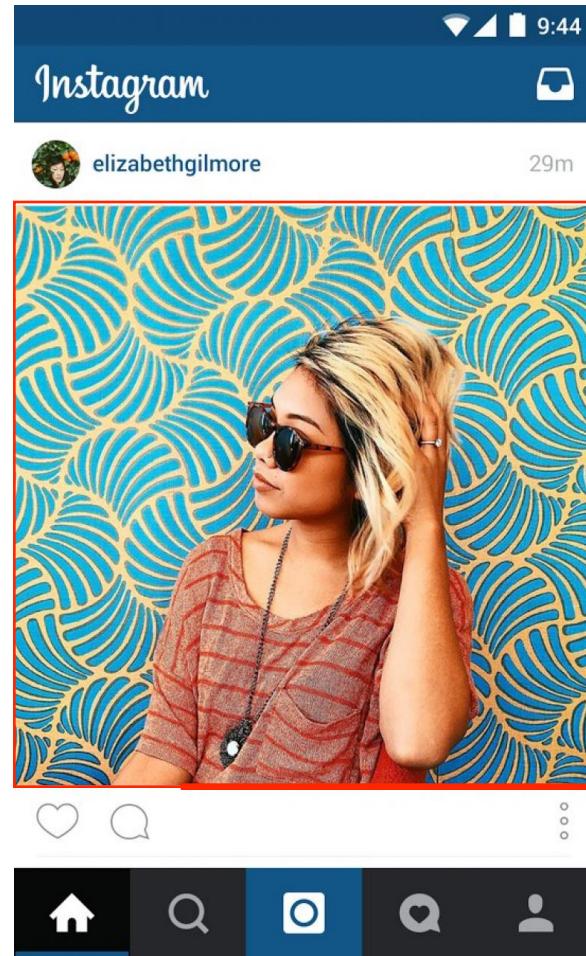
# Java – Referências Recomendadas

- **Java Basics for Android Development (treehouse - resumido)**
  - <http://blog.teamtreehouse.com/java-basics-for-android-development-part-1>
  - <http://blog.teamtreehouse.com/java-basics-for-android-development-part-2>
- **Java Tutorial for Complete Beginners (udemy - 16 horas de vídeo)**
  - <https://www.udemy.com/java-tutorial/>
- **Oracle – The Java Tutorials**
  - <http://docs.oracle.com/javase/tutorial/>

# Falando em Desenvolver Apps... Views!

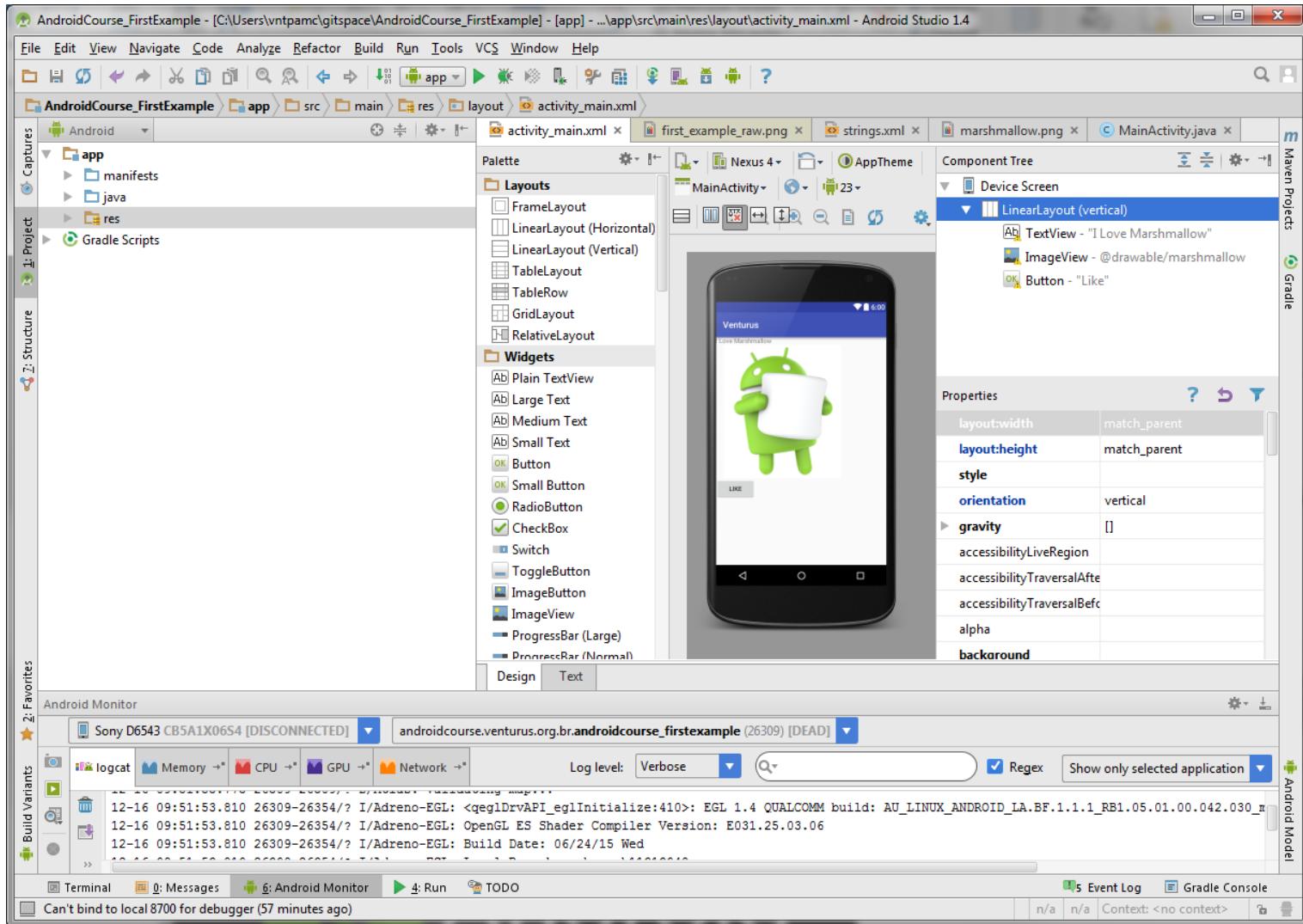


TextView



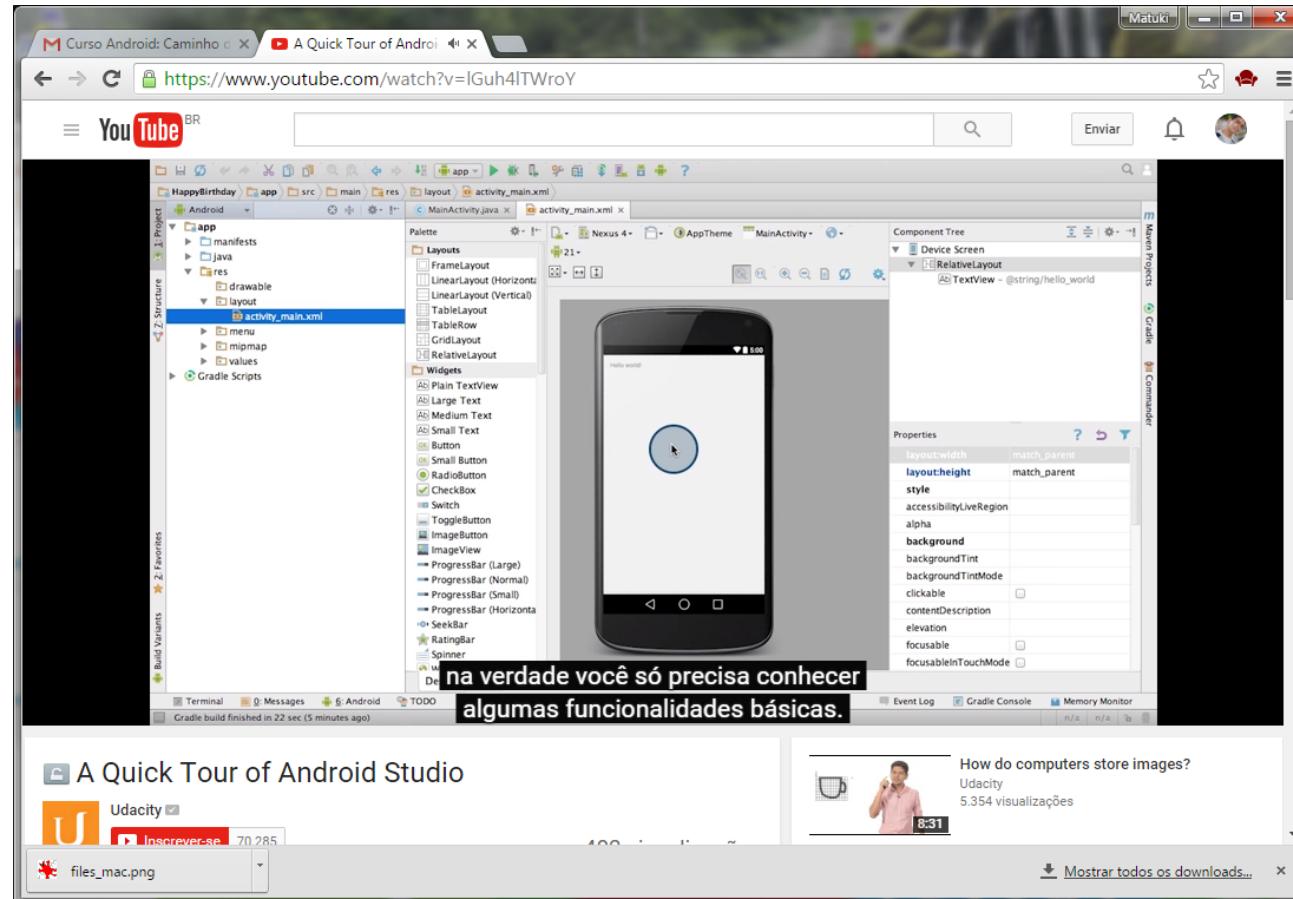
ImageView

# O Android Studio



# Android Studio - Vídeo de Introdução

<https://www.youtube.com/watch?v=lGuh4lTWroY>

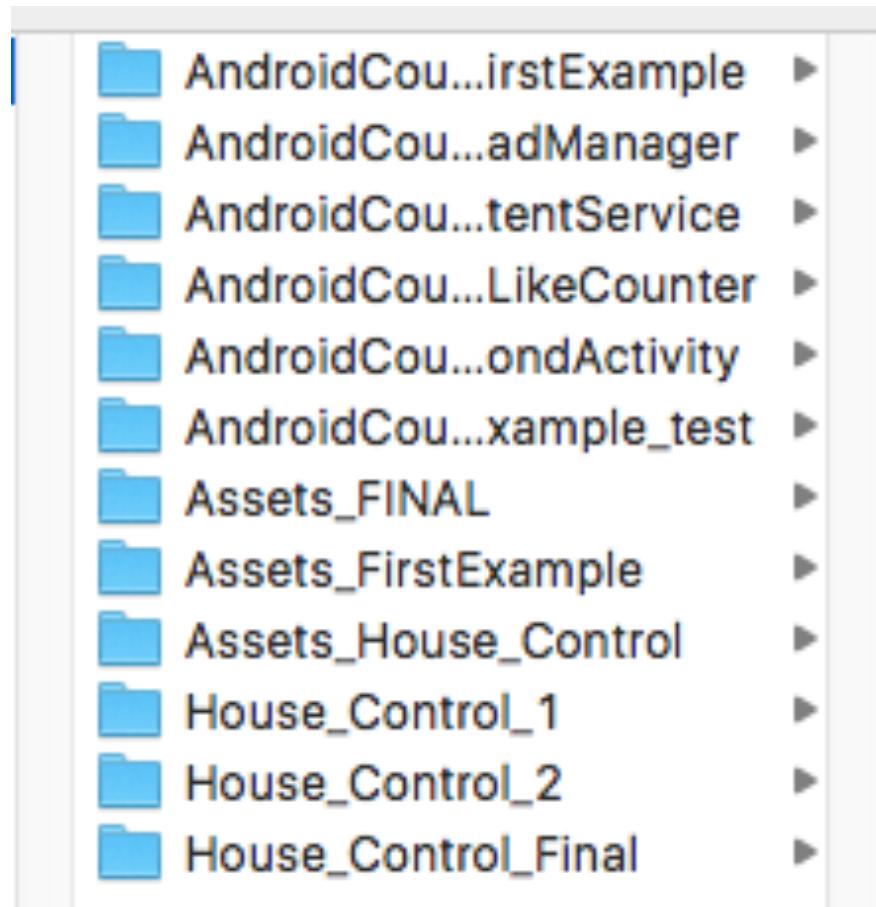


# Importando o Projeto (Git)

1. Abrir o Terminal
2. Navegar até /Users/vntlab
3. “mkdir gitspace” e depois “cd gitspace”
4. “mkdir Venturus4TechAndroid”
5. “cd Venturus4TechAndroid”
6. “git clone https://github.com/fti-venturus/venturus4tech\_android.git”

# Importando o Projeto

/Users/vntlab/projects/sources/android\_all\_projects



# Uma primeira tela!!

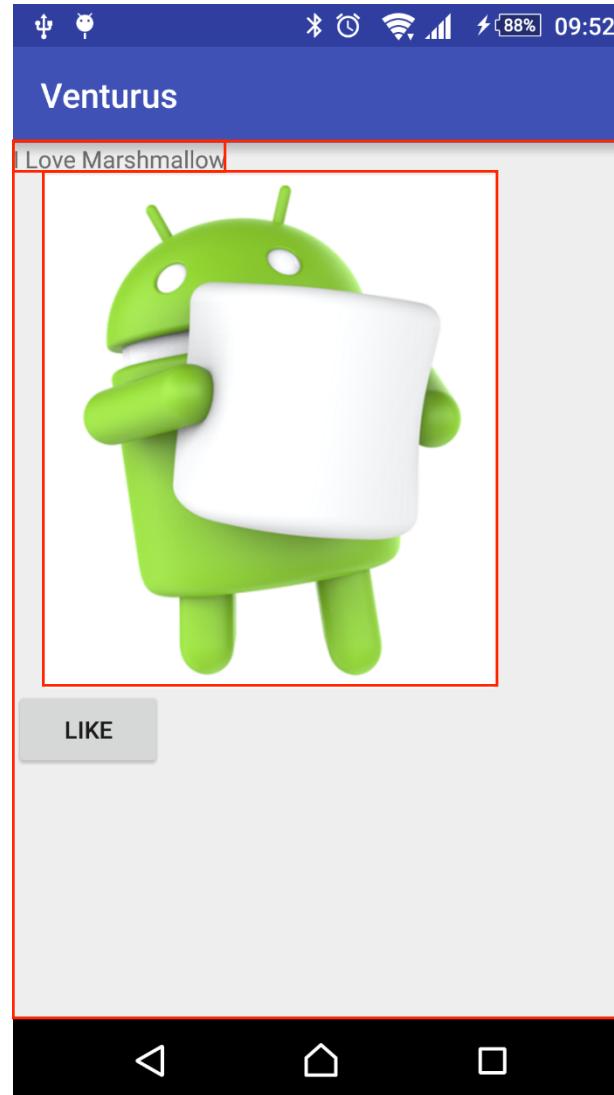
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I Love Marshmallow" />

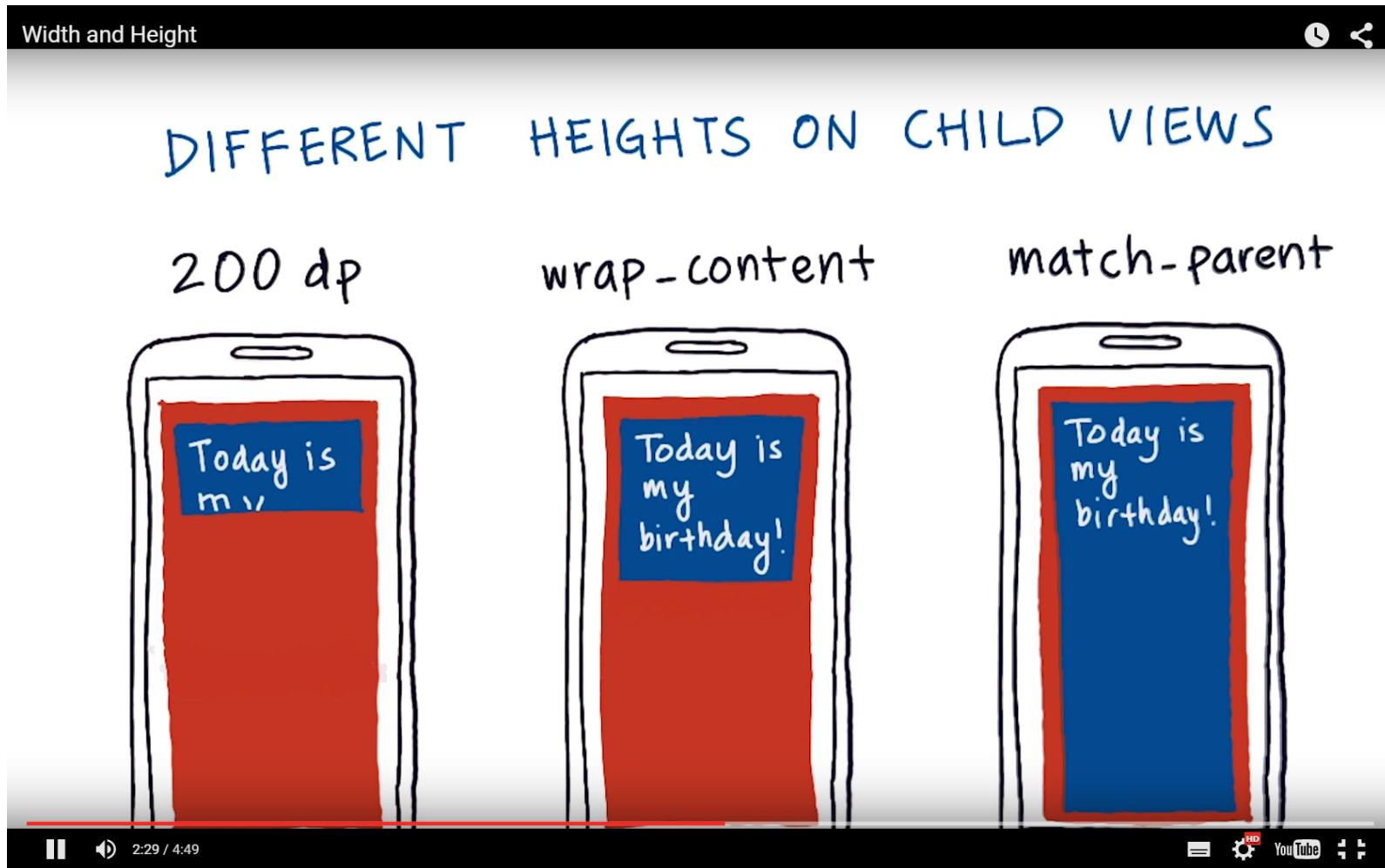
    <ImageView
        android:layout_width="300dp"
        android:layout_height="300dp"
        android:src="@drawable/marshmallow" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Like"/>

</LinearLayout>
```

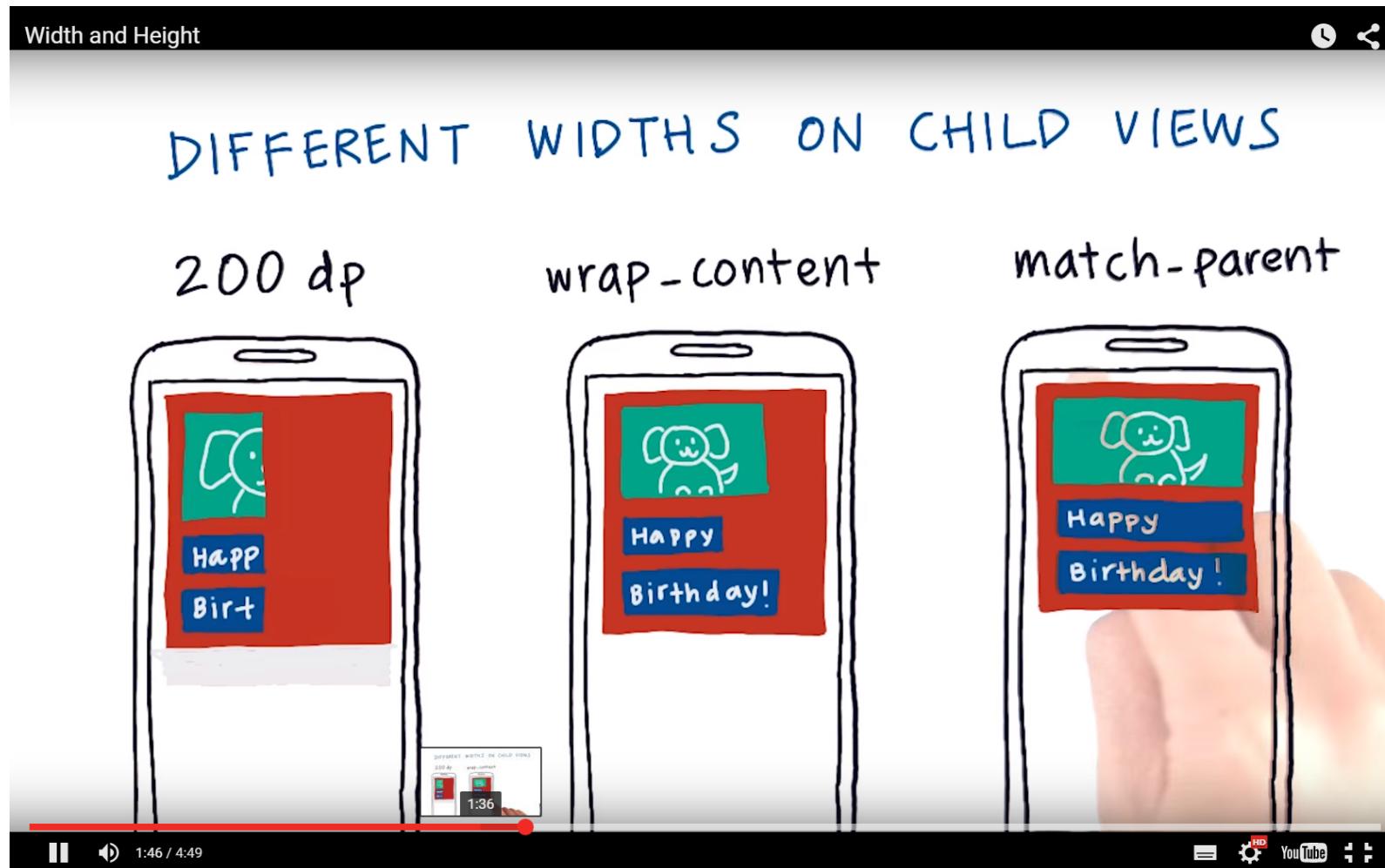


# Altura (android:layout\_height)



© Copyright: Udacity

# Largura(`android:layout_width`)



© Copyright: Udacity

# Um pouco melhor...

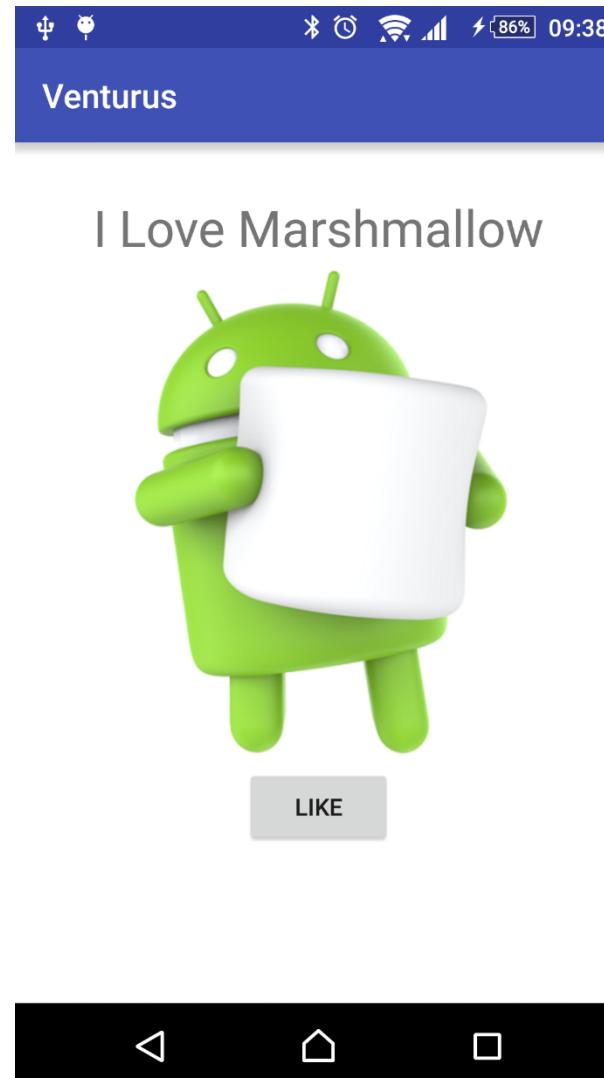
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    android:orientation="vertical"
    android:background="#ffffffff"
    android:paddingTop="30dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I Love Marshmallow"
        android:textSize="30sp" />

    <ImageView
        android:layout_width="300dp"
        android:layout_height="300dp"
        android:src="@drawable/marshmallow" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Like"/>

</LinearLayout>
```



# Pequenas mudanças – Faça você mesmo



I Love Marshmallow



LIKE



Antes



I Love Marshmallow

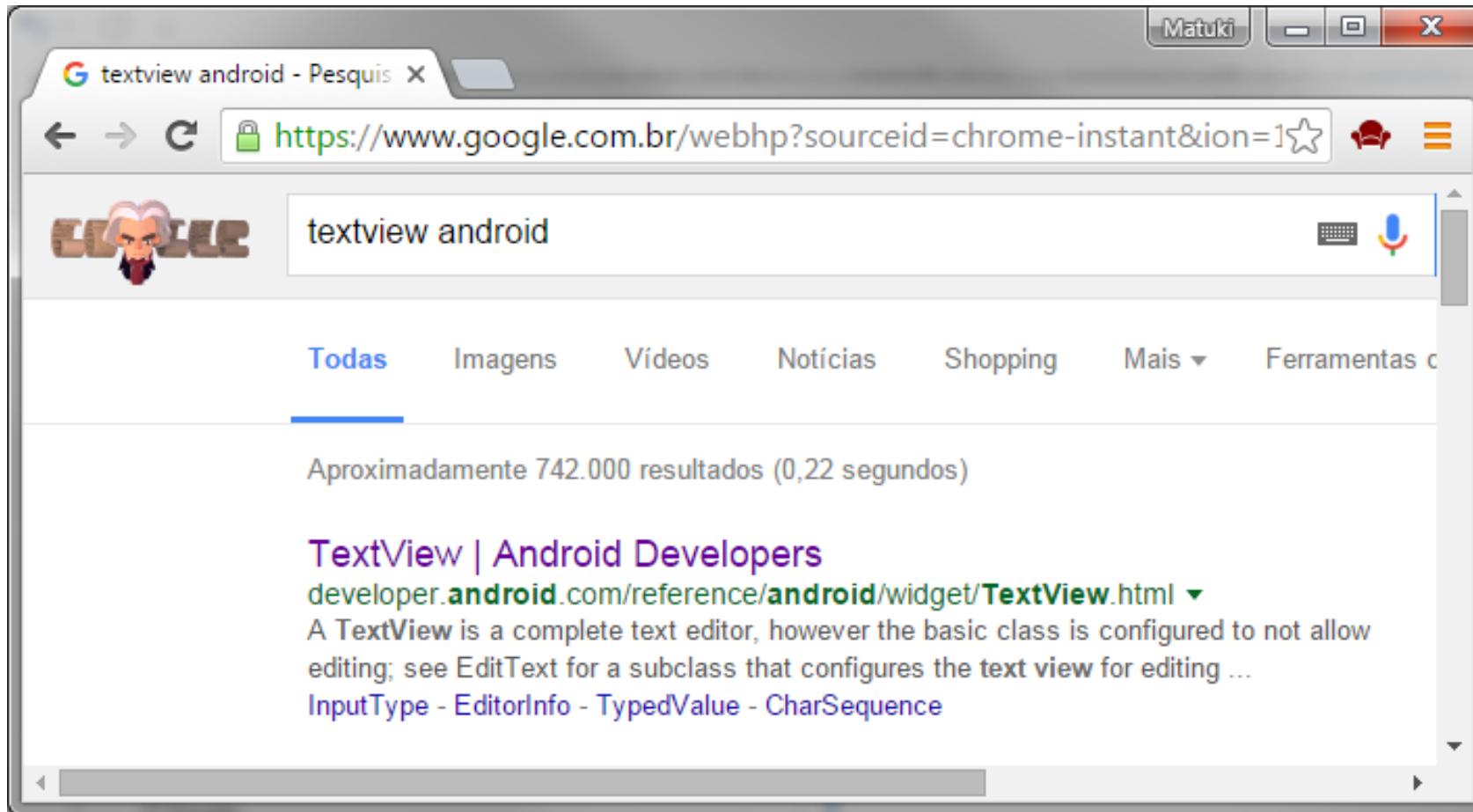


LIKE



Depois

# Dica: Procure no Google!



# Documentação (developer.android.com)

The screenshot shows a web browser window displaying the Android Developers documentation for the `TextView` class. The URL in the address bar is `developer.android.com/intl/pt-br/reference/android/widget/TextView.html`. The page is part of the "Reference" section under the "Develop" tab. The left sidebar lists various Java packages and classes, with `android.widget.TextView` selected. The main content area provides a detailed overview of the `TextView` class, including its inheritance from `View` and `Object`, and its implementation of `ViewTreeObserver.OnPreDrawListener`. It also lists known subclasses such as `AppCompatTextView` and `EditText`. A "Class Overview" section describes `TextView` as a text editor that can be edited or displayed. The API level listed is 23.

TextView

extends View

implements ViewTreeObserver.OnPreDrawListener

java.lang.Object  
↳ android.view.View  
↳ android.widget.TextView

► Known Direct Subclasses

AppCompatTextView, Button, CheckedTextView, Chronometer, DigitalClock, EditText, RowHeaderView, TextClock

► Known Indirect Subclasses

AppCompatAutoCompleteTextView, AppCompatButton, AppCompatCheckBox, AppCompatCheckedTextView,  
AppCompatEditText, AppCompatMultiAutoCompleteTextView, AppCompatRadioButton, AutoCompleteTextView, CheckBox,  
CompoundButton, ExtractEditText, GuidedActionEditText, MultiAutoCompleteTextView, RadioButton, SearchEditText, Switch,  
SwitchCompat, ToggleButton

Class Overview

Displays text to the user and optionally allows them to edit it. A `TextView` is a complete text editor, however the basic class is configured to not allow editing; see `EditText` for a subclass that configures the text view for editing.

# android:padding VS android:layout\_margin

padding

VS

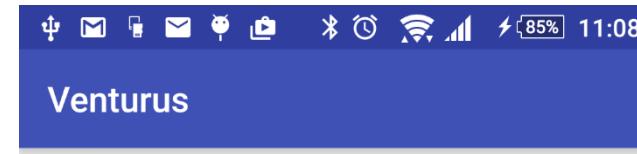
layout-margin



© Copyright: Udacity

# Resultado

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:background="#ffffffff"  
    android:gravity="center_horizontal"  
    android:orientation="vertical"  
    android:paddingTop="30dp"  
    tools:context=".MainActivity">  
  
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="I Love Marshmallow"  
    android:textColor="@color/colorAccent"  
    android:textSize="30sp" />  
  
<ImageView  
    android:layout_width="300dp"  
    android:layout_height="300dp"  
    android:layout_marginTop="30dp"  
    android:src="@drawable/marshmallow" />  
  
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="30dp"  
    android:text="Like" />  
  
</LinearLayout>
```



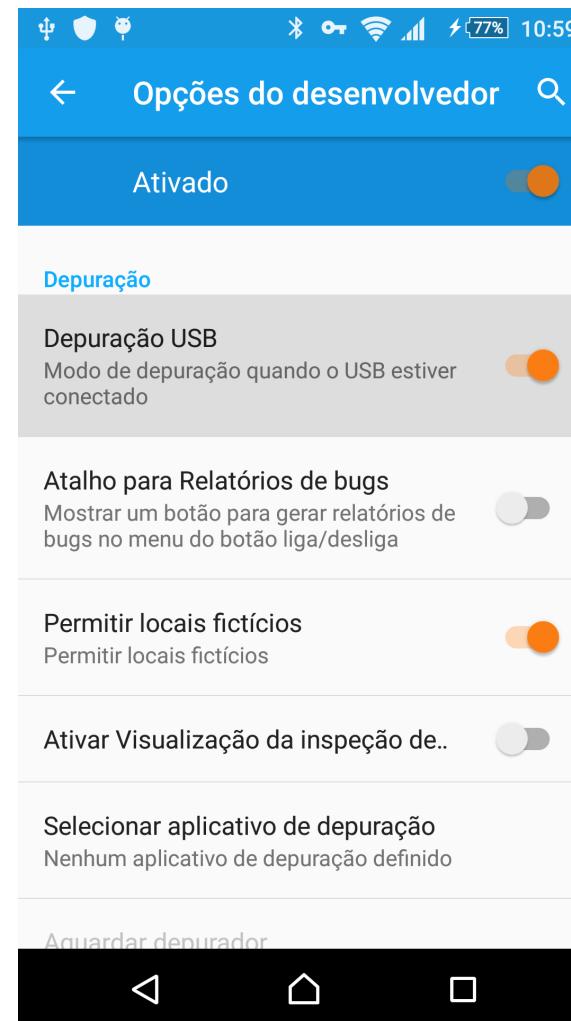
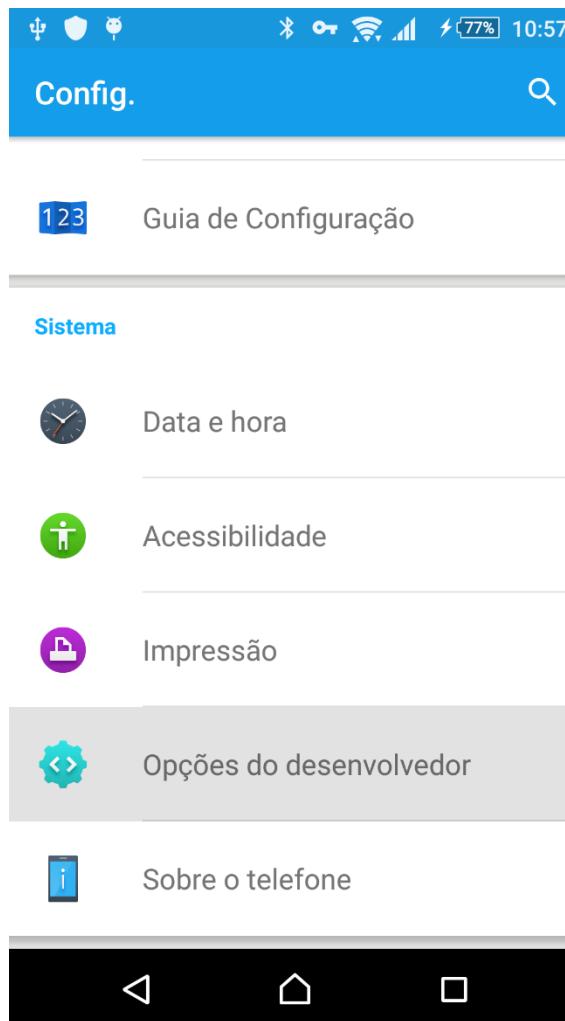
I Love Marshmallow



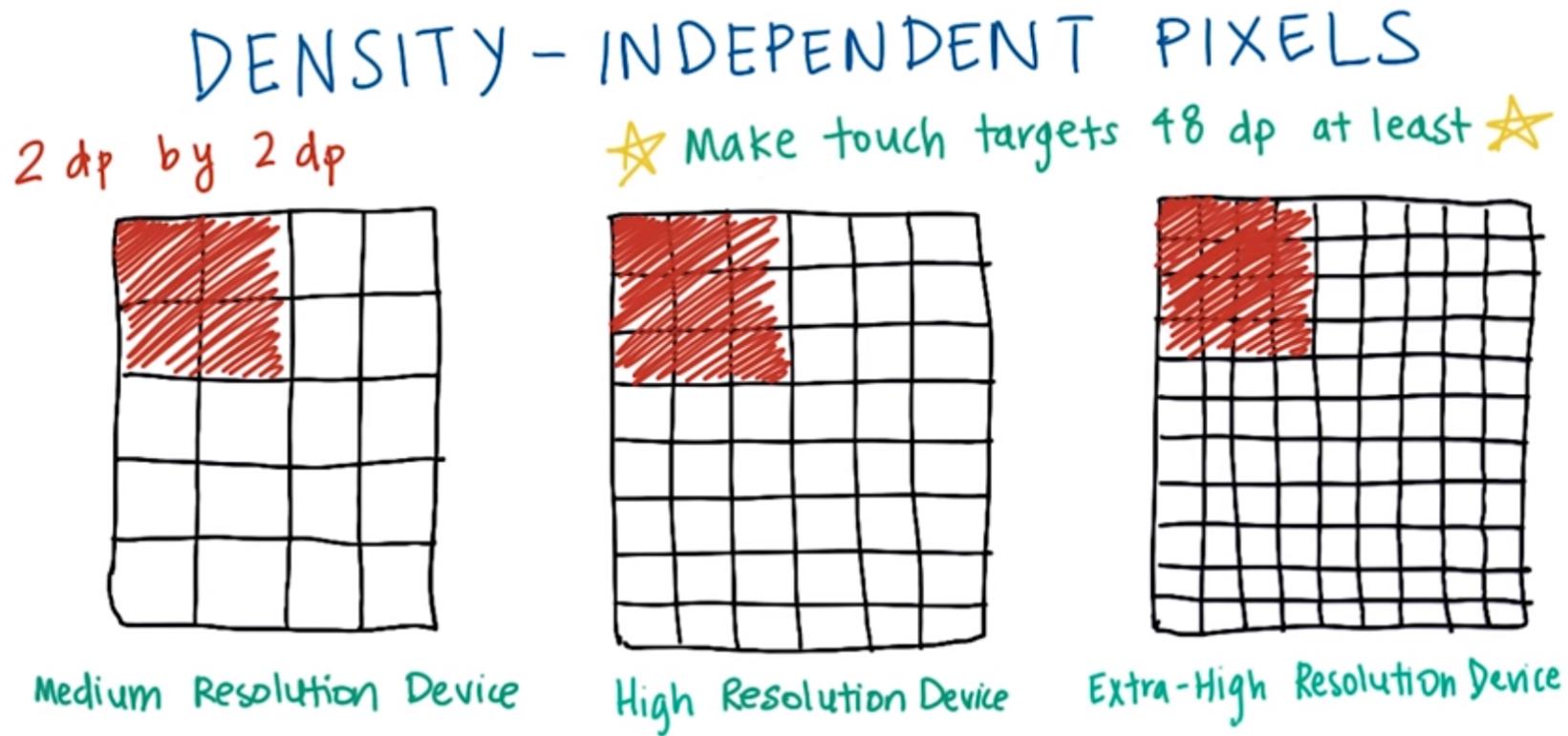
LIKE



# Rodando no Smartphone!

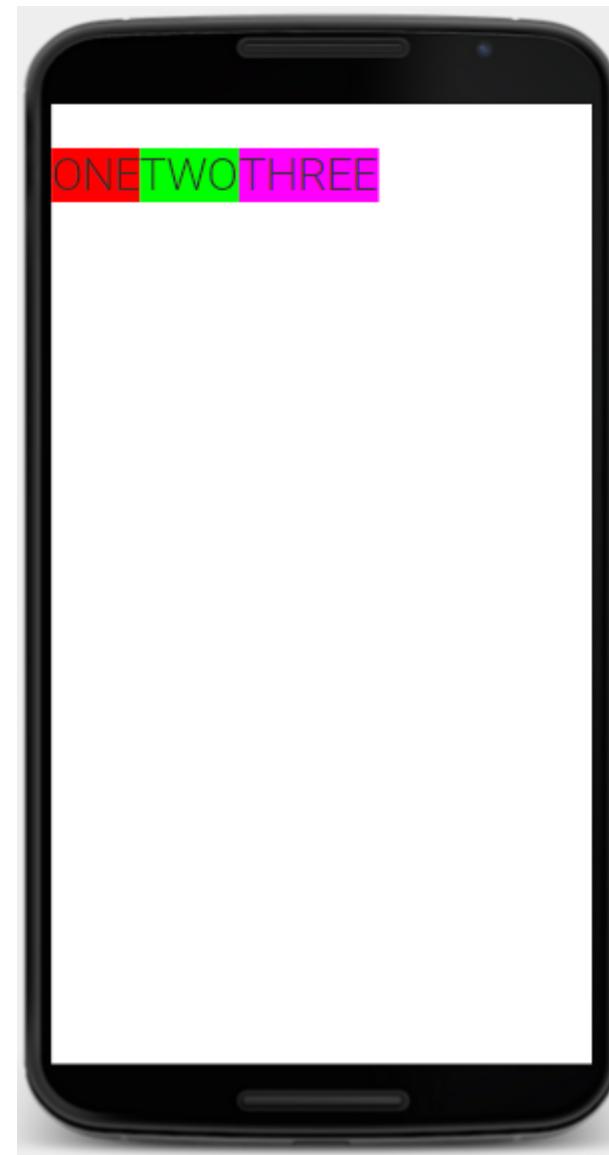


# Porque “dp” e não “cm” ou “pixels”?



# ViewGroups: LinearLayout

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="horizontal"  
    android:background="#ffffff"  
    android:paddingTop="30dp">  
  
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="ONE"  
    android:background="#ff0000"  
    android:textSize="30sp" />  
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="TWO"  
    android:background="#00ff00"  
    android:textSize="30sp" />  
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="THREE"  
    android:background="#ff00ff"  
    android:textSize="30sp" />  
  
</LinearLayout>
```



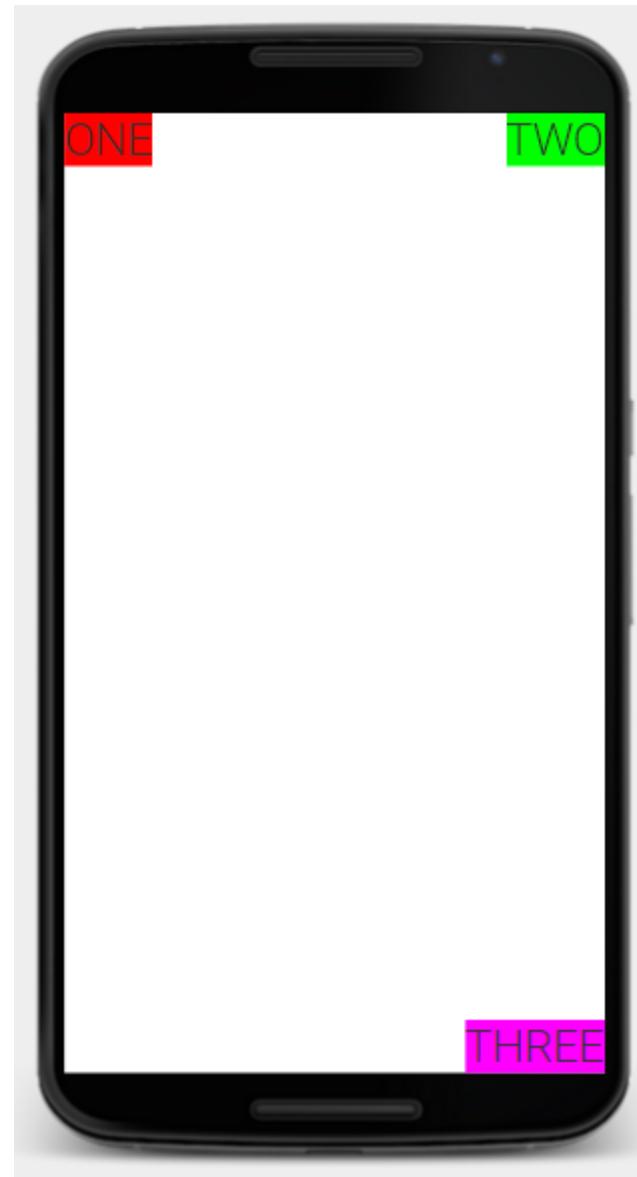
# ViewGroups: LinearLayout

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
    android:background="#ffffff"  
    android:paddingTop="30dp">  
  
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="ONE"  
    android:background="#ff0000"  
    android:textSize="30sp" />  
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="TWO"  
    android:background="#00ff00"  
    android:textSize="30sp" />  
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="THREE"  
    android:background="#ff00ff"  
    android:textSize="30sp" />  
  
</LinearLayout>
```



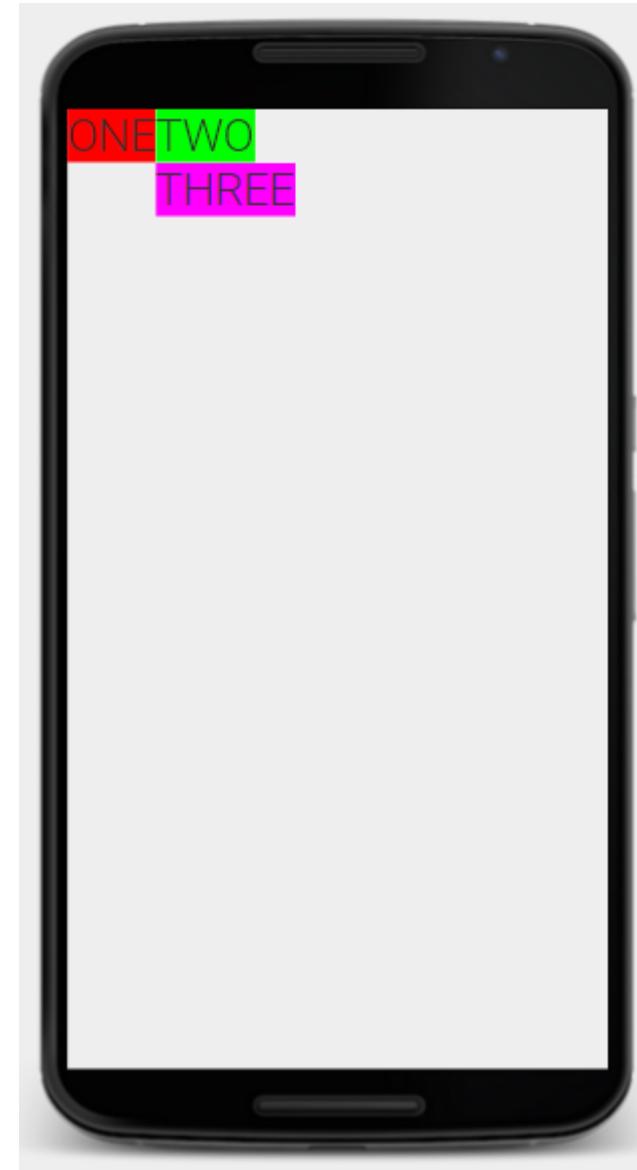
# ViewGroups: RelativeLayout

```
<RelativeLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:background="#ffffffff">  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="ONE"  
        android:background="#ff0000"  
        android:textSize="30sp" />  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="TWO"  
        android:layout_alignParentRight="true"  
        android:background="#00ff00"  
        android:textSize="30sp" />  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="THREE"  
        android:layout_alignParentBottom="true"  
        android:layout_alignParentRight="true"  
        android:background="#ff00ff"  
        android:textSize="30sp" />  
</RelativeLayout>
```

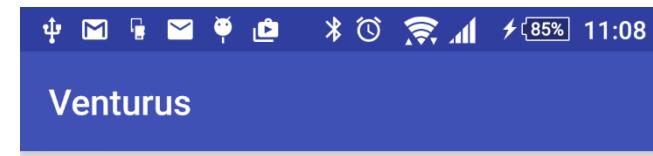


# ViewGroups: RelativeLayout

```
<RelativeLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
    <TextView  
        android:id="@+id/one"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="ONE"  
        android:background="#ff0000"  
        android:textSize="30sp" />  
    <TextView  
        android:id="@+id/two"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="TWO"  
        android:layout_toRightOf="@+id/one"  
        android:background="#00ff00"  
        android:textSize="30sp" />  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="THREE"  
        android:layout_toRightOf="@+id/one"  
        android:layout_below="@+id/two"  
        android:background="#ff00ff"  
        android:textSize="30sp" />  
</RelativeLayout>
```



# E aquele botão de “Like”? Não faz nada?



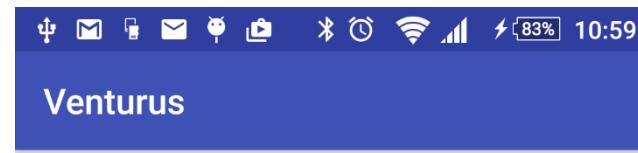
I Love Marshmallow



LIKE



# Contador de Likes



I Love Marshmallow



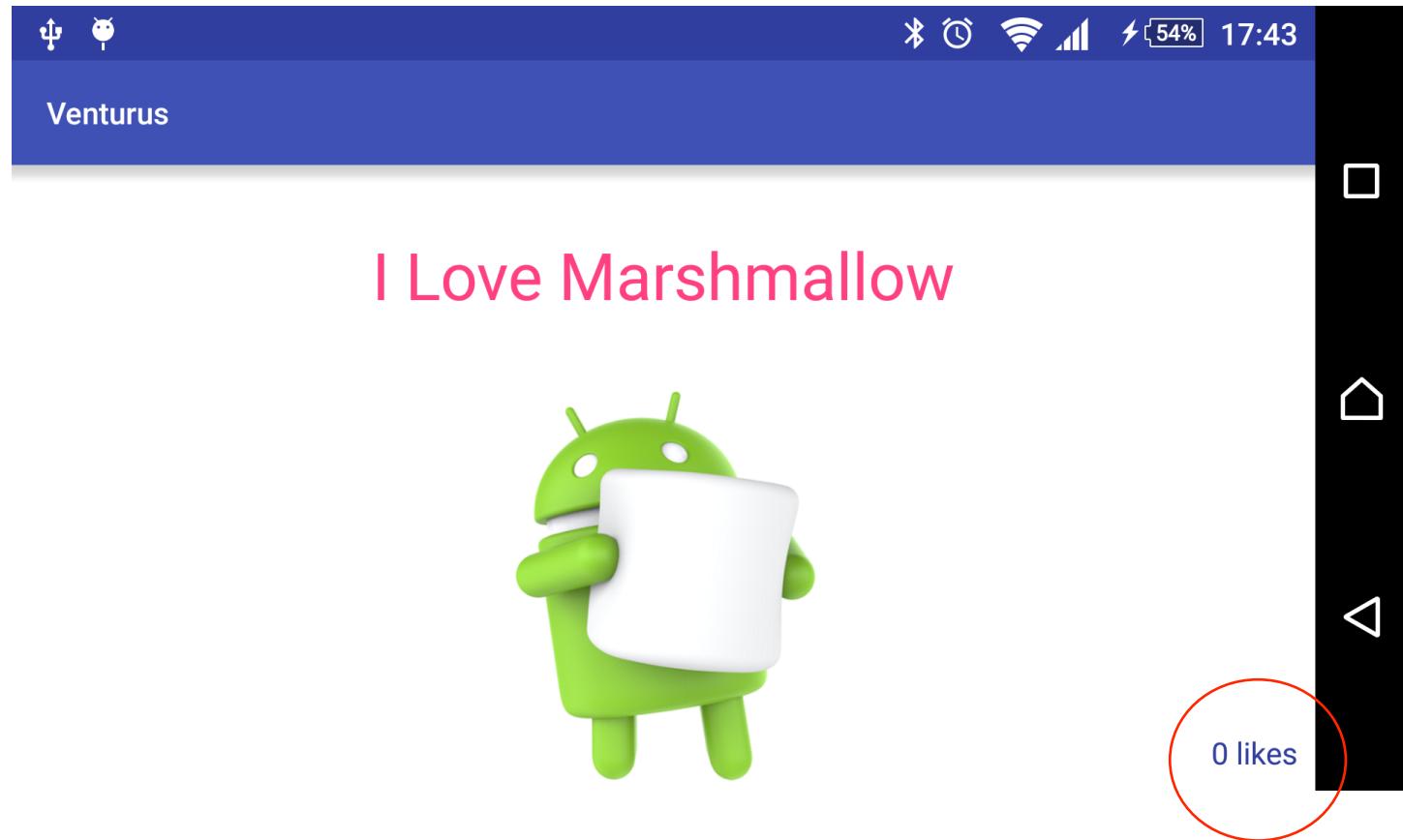
LIKE

10 likes



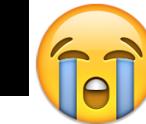
VENTURUS4TECH  
JANEIRO/16

# Rotação = Volta para 0 Likes?

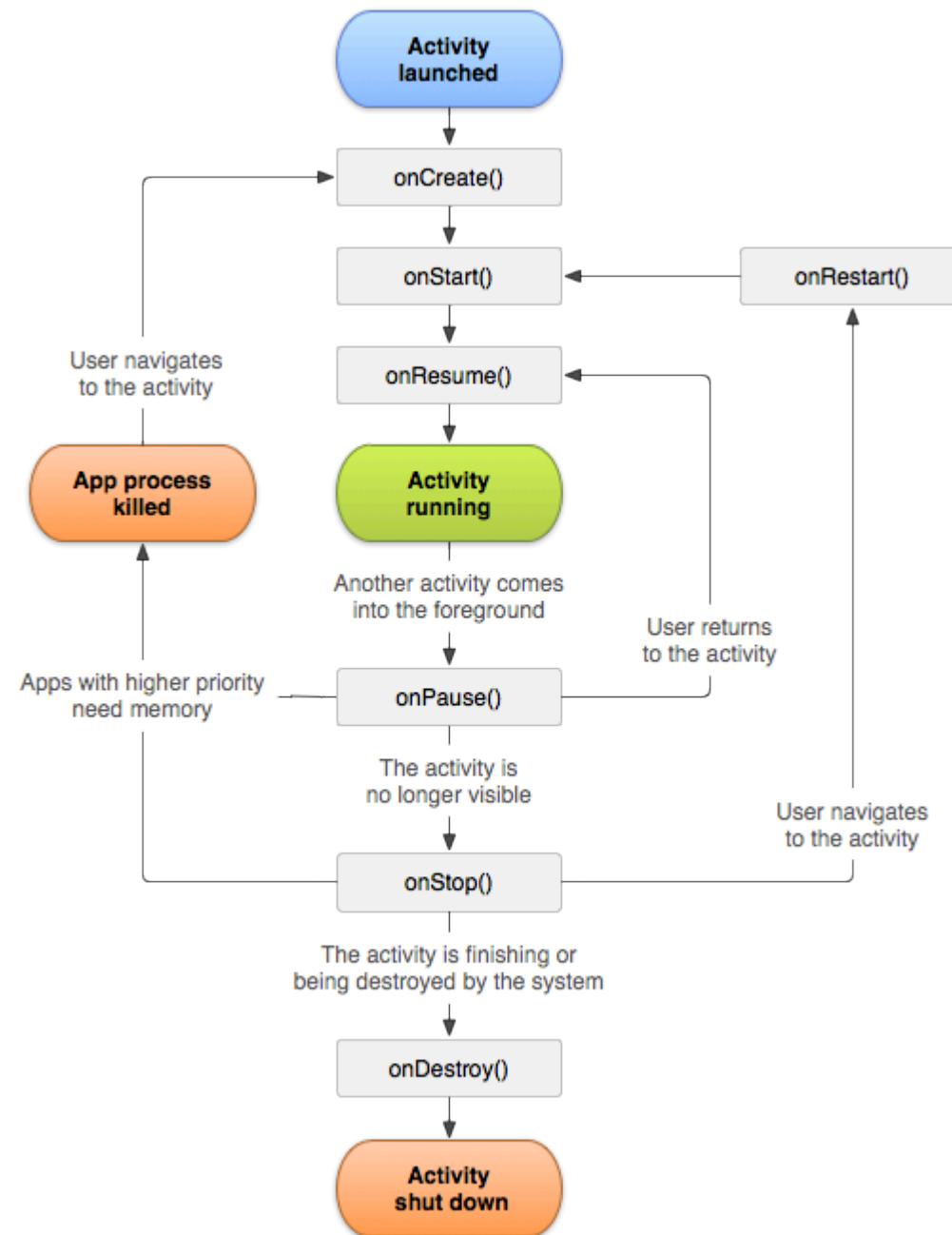


Cadê o botão?

0 likes



# Activity – Ciclo de Vida



# Métodos de call-back – Ciclo de vida

```
public class Activity extends ApplicationContext {  
  
    protected void onCreate(Bundle savedInstanceState);  
  
    protected void onStart();  
  
    protected void onRestart();  
  
    protected void onResume();  
  
    protected void onPause();  
  
    protected void onStop();  
  
    protected void onDestroy();  
}
```

# Configuration Changes

*“(...) a configuration change (such as a change in screen orientation, language, input devices, etc) **will cause your current activity to be destroyed**, going through the normal activity lifecycle process of `onPause()`, `onStop()`, and `onDestroy()` as appropriate.*

*“(...) once `onDestroy()` is called in that instance **then a new instance of the activity will be created**, with whatever `savedInstanceState` the previous instance had generated from `onSaveInstanceState(Bundle)`. ”*

Fonte:

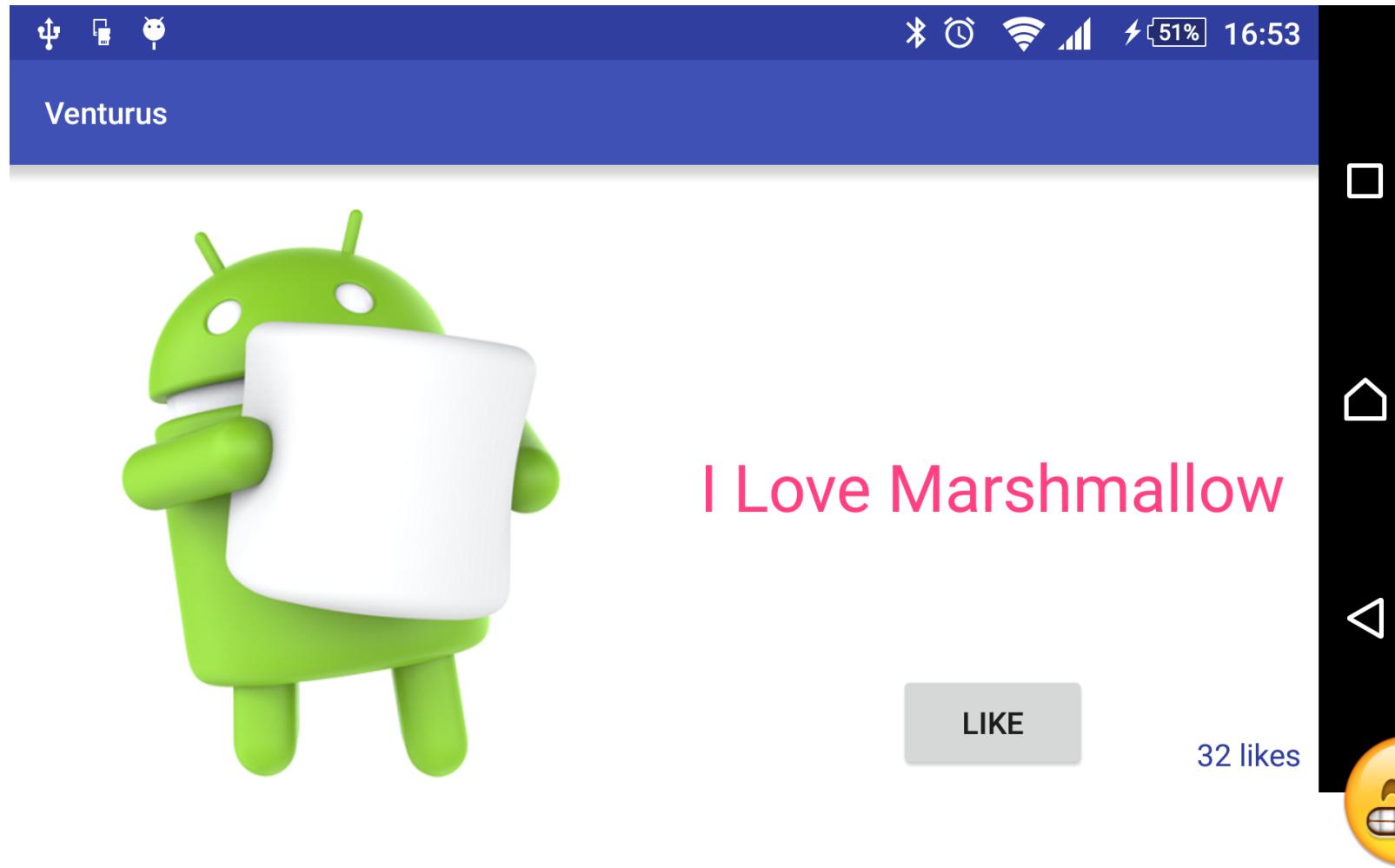
<http://developer.android.com/intl/pt-br/reference/android/app/Activity.html>

# Tarefa 1: Persistir o Estado da Activity

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    mPrefs = getSharedPreferences("arquivo_preferencias", Context.MODE_PRIVATE);  
  
    mLikeCounter = mPrefs.getInt("num_likes", 0);  
  
    (...)  
}  
  
@Override  
protected void onPause() {  
    super.onPause();  
    SharedPreferences.Editor editor = mPrefs.edit();  
    editor.putInt("num_likes", mLikeCounter);  
    editor.apply();  
}
```

<http://developer.android.com/intl/pt-br/reference/android/app/Activity.html#SavingPersistentState>

## Tarefa 2: Criar um layout “paisagem”

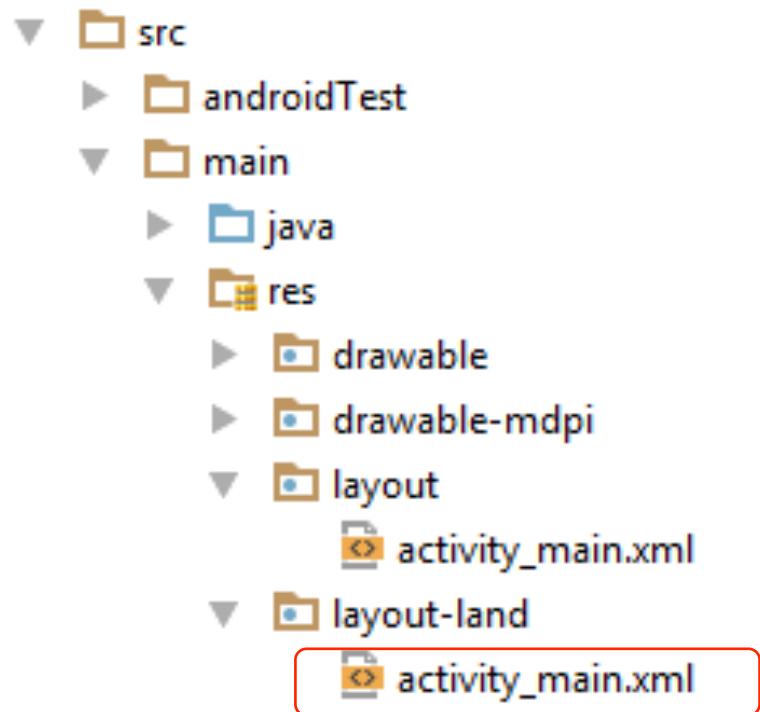


VENTURUS4TECH  
JANEIRO/16

# Qualificadores – Orientação da Tela

Screen characteristic	Qualifier	Description
Orientation	land	Resources for screens in the landscape orientation (wide aspect ratio).
	port	Resources for screens in the portrait orientation (tall aspect ratio).

## Tarefa 2: Criar um layout “paisagem”



# Tela de 3.2" – QVGA (320 x 240 pixels)



Cadê o botão? (parte 2)

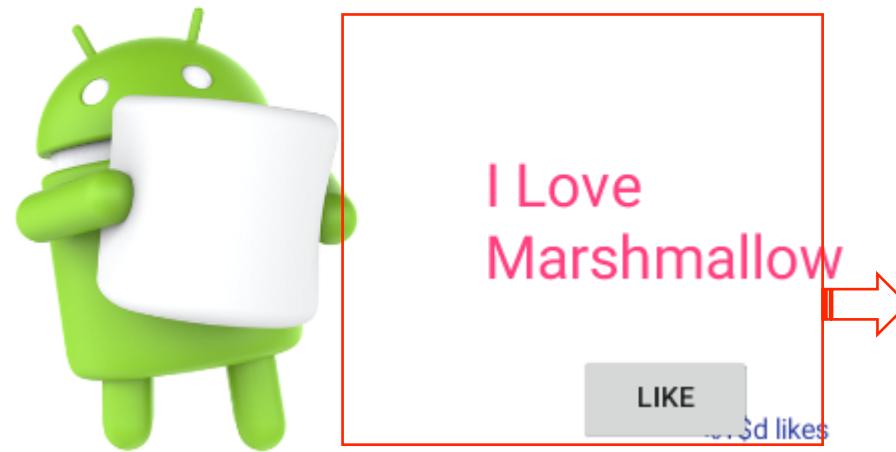


Imagen muito grande para a tela!  
“Empurrando” os componentes.

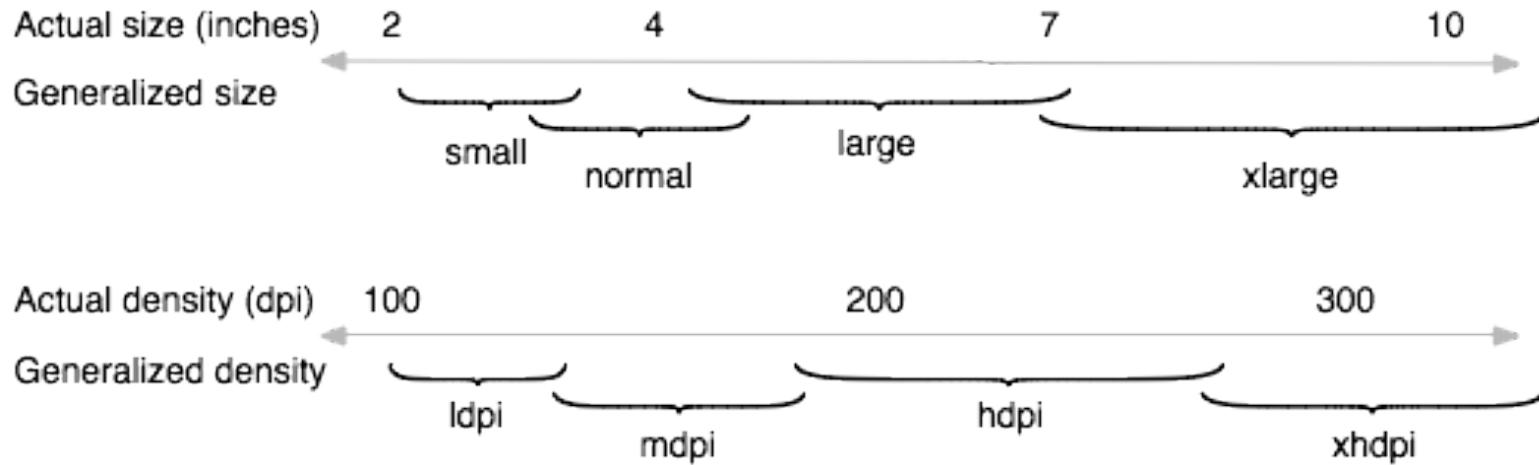


# Qualificadores – Tamanho da Tela

Screen characteristic	Qualifier	Description
Size	small	Resources for small size screens.
	normal	Resources for normal size screens. (This is the baseline size.)
	large	Resources for large size screens.
	xlarge	Resources for extra-large size screens.

Referência: [http://developer.android.com/intl/pt-br/guide/practices/screens\\_support.html](http://developer.android.com/intl/pt-br/guide/practices/screens_support.html)

# Qualificadores – “Buckets”



# Usando o dimens.xml

res\values\dimens.xml

```
<resources>
    (...)
    <dimen name="marshmallow_width">300dp</dimen>
    <dimen name="marshmallow_height">300dp</dimen>
</resources>
```

res\layout\activity\_main.xml || res\layout-land\activity\_main.xml

```
<ImageView
    android:id="@+id/marshmallow_image"
    android:layout_width="@dimen/marshmallow_width"
    android:layout_height="@dimen/marshmallow_height"
    android:layout_below="@id/marshmallow_text"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="30dp"
    android:src="@drawable/marshmallow" />
```

# Usando o dimens.xml

res\values-small\dimens.xml

```
<resources>
    <dimen name="marshmallow_width">150dp</dimen>
    <dimen name="marshmallow_height">150dp</dimen>
</resources>
```

res\layout\activity\_main.xml || res\layout-land\activity\_main.xml

```
<ImageView
    android:id="@+id/marshmallow_image"
    android:layout_width="@dimen/marshmallow_width"
    android:layout_height="@dimen/marshmallow_height"
    android:layout_below="@id/marshmallow_text"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="30dp"
    android:src="@drawable/marshmallow" />
```

# Tela de 3.2" – QVGA (320 x 240 pixels)



I Love Marshmallow



LIKE

2 likes



I Love Marshmallow

LIKE

2 likes



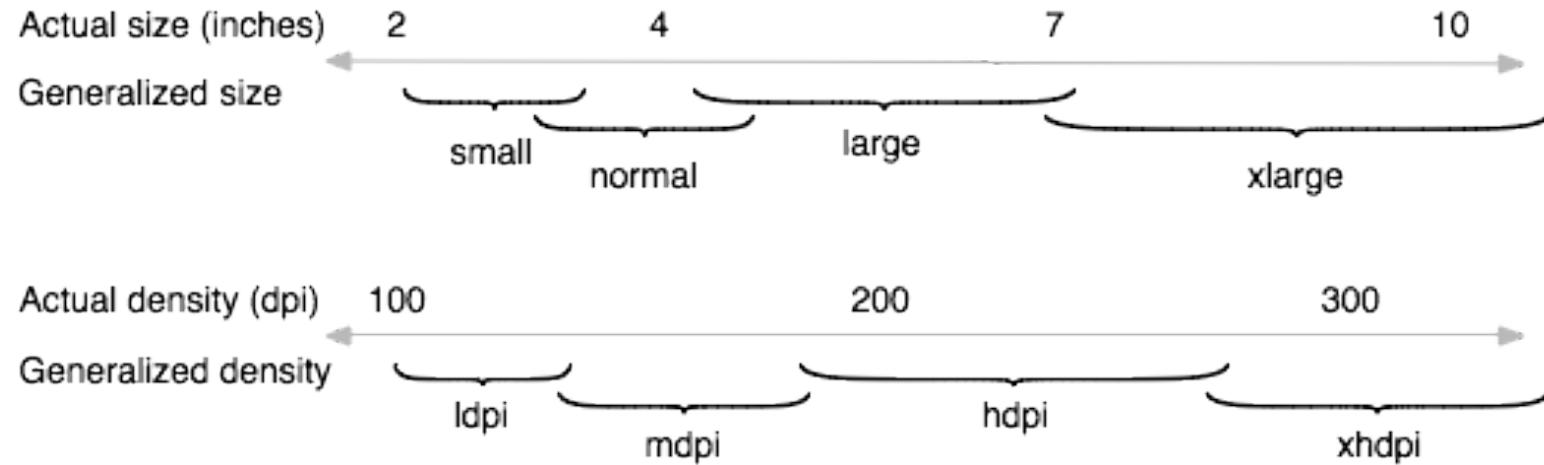
# Ainda na Tela de 3.2" – QVGA



# Qualificadores – Densidade de Pixels

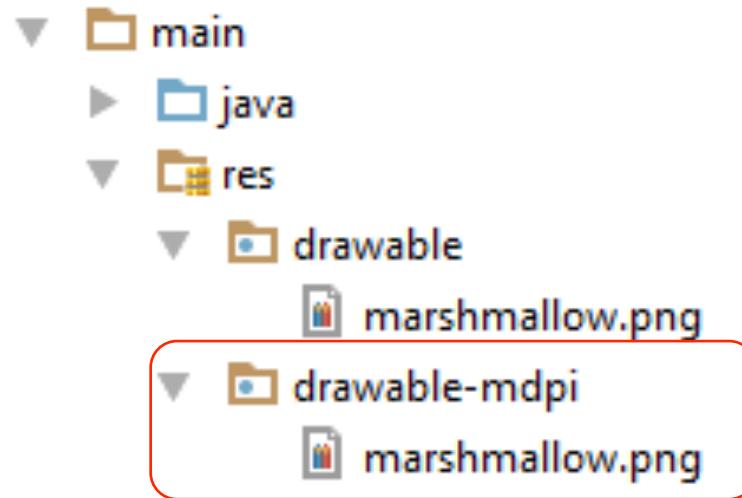
Screen characteristic	Qualifier	Description
Density	ldpi	Resources for low-density (ldpi) screens (~120dpi).
	mdpi	Resources for medium-density (mdpi) screens (~160dpi). (This is the baseline density.)
	hdpi	Resources for high-density (hdpi) screens (~240dpi).
	xhdpi	Resources for extra-high-density (xhdpi) screens (~320dpi).
	xxhdpi	Resources for extra-extra-high-density (xxhdpi) screens (~480dpi).
	xxxhdpi	Resources for extra-extra-extra-high-density (xxxhdpi) uses (~640dpi). Use this for the launcher icon only, see <a href="#">note</a> above.

# Qualificadores – “Buckets”

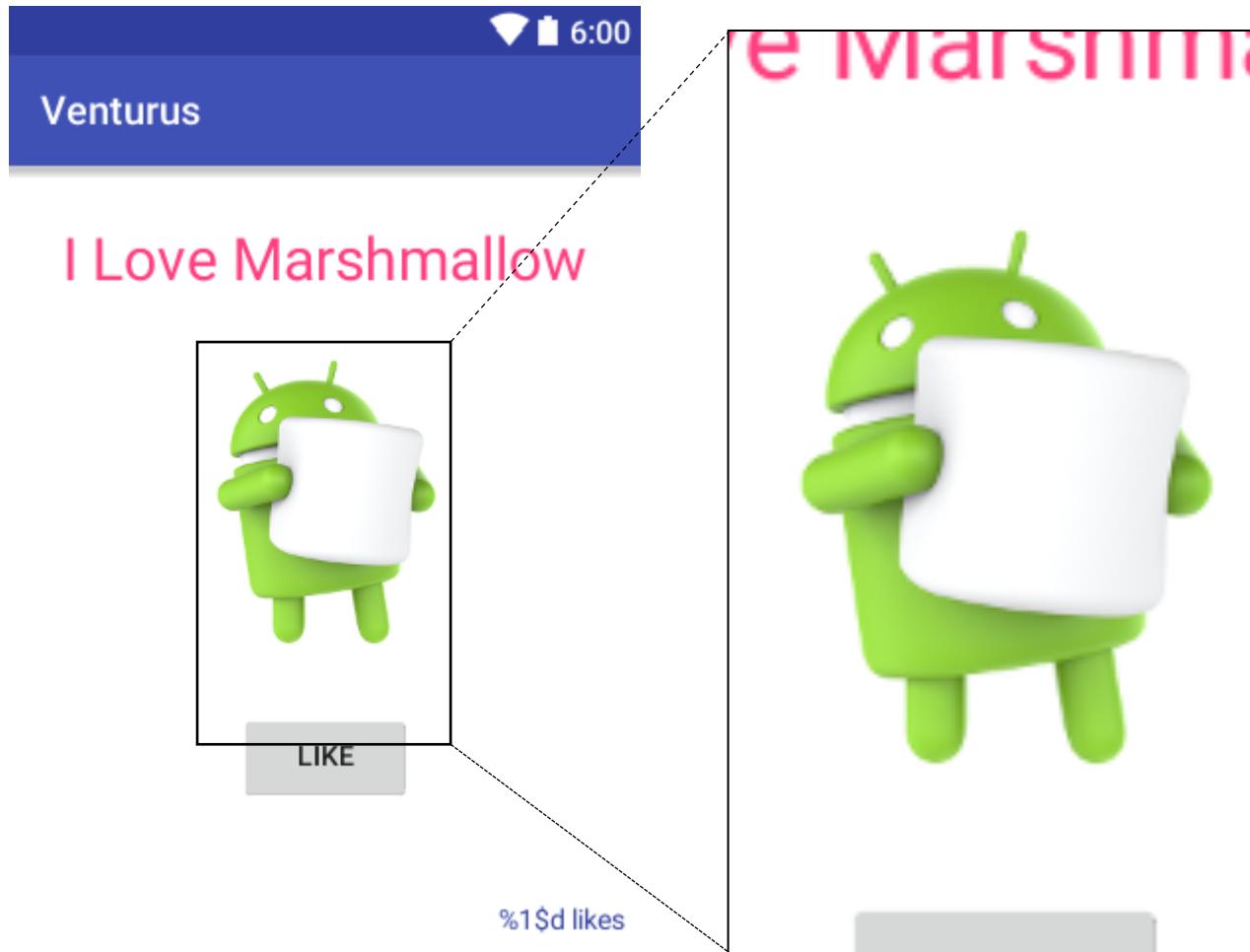


# Pasta drawable-<qualifier>

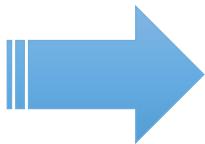
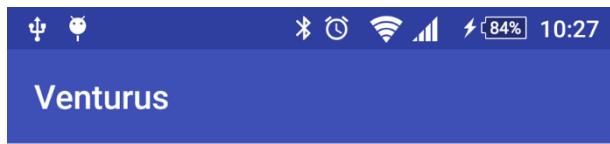
- ↳ 3.7" WVGA (480 × 800: hdpi)
- ↳ 4.0" WVGA (480 × 800: hdpi)
- ↳ 3.2" HVGA slider (ADP1) (320 × 480: mdpi)
- ↳ 10.1" WXGA (Tablet) (1280 × 800: mdpi)
- ↳ 4.7" WXGA (1280 × 720: xhdpi)
- ↳ 5.4" FWVGA (480 × 854: mdpi)
- ↳ 4.65" 720p (720 × 1280: xhdpi)
- ↳ 3.7" FWVGA slider (480 × 854: hdpi)
- ✓ 3.2" QVGA (ADP2) (320 × 480: mdpi)
- ↳ 2.7" QVGA slider (240 × 320: ldpi)
- ↳ 3.4" WQVGA (240 × 432: ldpi)
- ↳ 7.0" WSVGA (Tablet) (1024 × 600: mdpi)
- ↳ 3.3" WQVGA (240 × 400: ldpi)
- ↳ 5.1" WVGA (480 × 800: mdpi)
- ↳ 2.7" QVGA (240 × 320: ldpi)



# Tela de 3.2" – QVGA - Final



# Adicionando Uma Nova Tela (Activity)



# Opções do ImageView (android:scaleType)

Must be one of the following constant values.

Constant	Value	Description
<code>matrix</code>	0	Scale using the image matrix when drawing. See <a href="#">setImageMatrix(Matrix)</a> .
<code>fitXY</code>	1	Scale the image using <a href="#">FILL</a> .
<code>fitStart</code>	2	Scale the image using <a href="#">START</a> .
<code>fitCenter</code>	3	Scale the image using <a href="#">CENTER</a> .
<code>fitEnd</code>	4	Scale the image using <a href="#">END</a> .
<code>center</code>	5	Center the image in the view, but perform no scaling.
<code>centerCrop</code>	6	Scale the image uniformly (maintain the image's aspect ratio) so both dimensions (width and height) of the image will be equal to or larger than the corresponding dimension of the view (minus padding). The image is then centered in the view.
<code>centerInside</code>	7	Scale the image uniformly (maintain the image's aspect ratio) so that both dimensions (width and height) of the image will be equal to or less than the corresponding dimension of the view (minus padding). The image is then centered in the view.

This corresponds to the global attribute resource symbol [scaleType](#).

# Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="androidcourse.venturus.org.br.androidcourse_firstexample_likecounter">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Venturus"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

        <activity android:name=".PictureActivity"></activity>
    </application>
</manifest>
```

# Intent - Invocando a PictureActivity

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
  
    (...)  
  
    ImageView robotImageView = (ImageView)  
        findViewById(R.id.marshmallow_image);  
    robotImageView.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            Intent pictureIntent = new Intent(MainActivity.this, PictureActivity.class);  
  
            MainActivity.this.startActivity(pictureIntent);  
        }  
    });  
}
```

# Intent – Parâmetros Extra

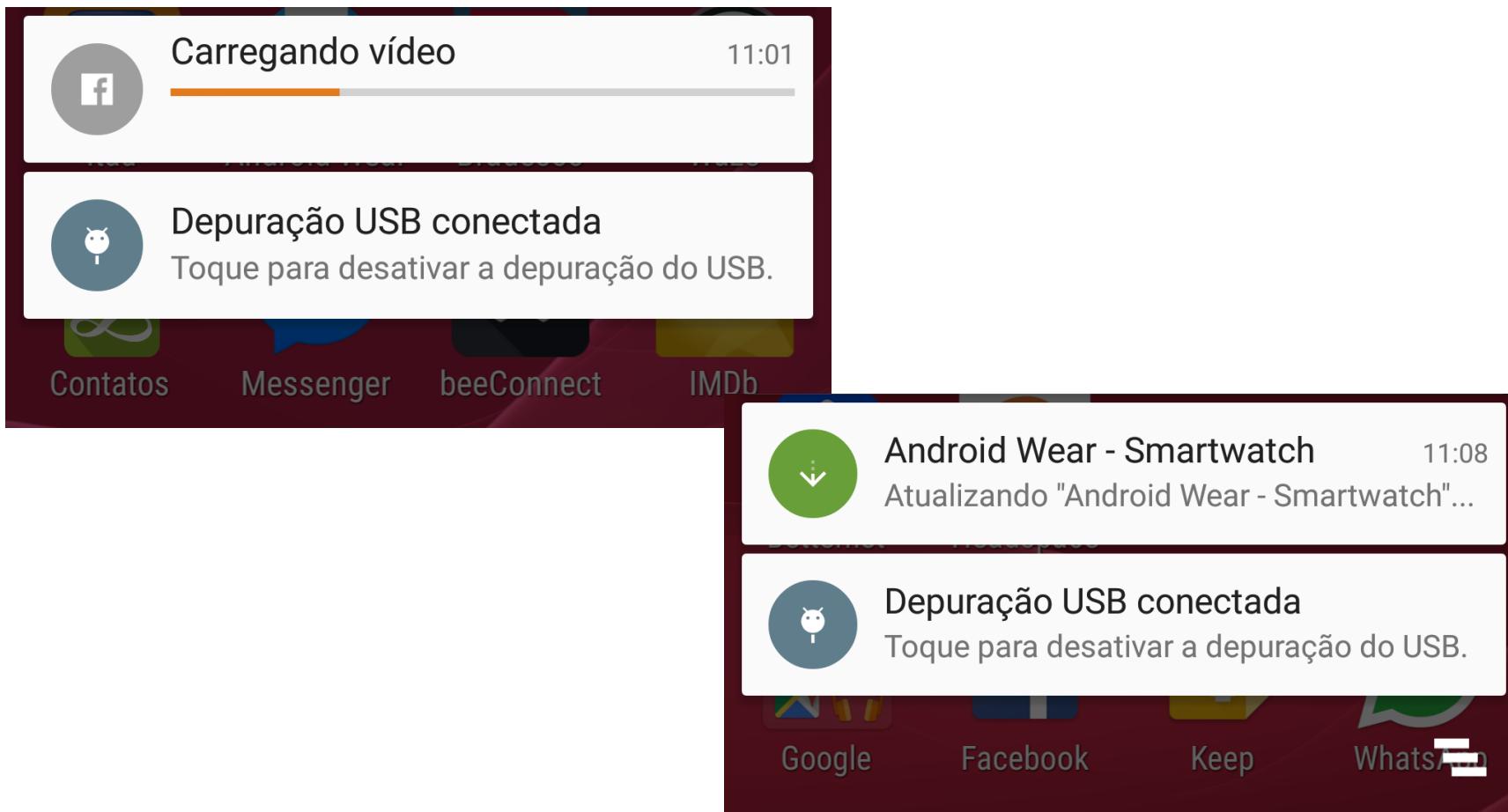
## Activity 1

```
Intent intent = new Intent(Activity1.this, Activity2.class);  
  
intent.putExtra("PARAM_1", 15);  
  
intent.putExtra("PARAM_2", false);  
  
intent.putExtra("PARAM_3", "Algum Texto");
```

## Activity 2

```
Intent intent = this.getIntent();  
  
int passedInt = intent.getIntExtra("PARAM_1", -1);  
  
boolean passedBool = intent.getBooleanExtra("PARAM_2", false);  
  
String passedString = intent.getStringExtra("PARAM_3");
```

# Services



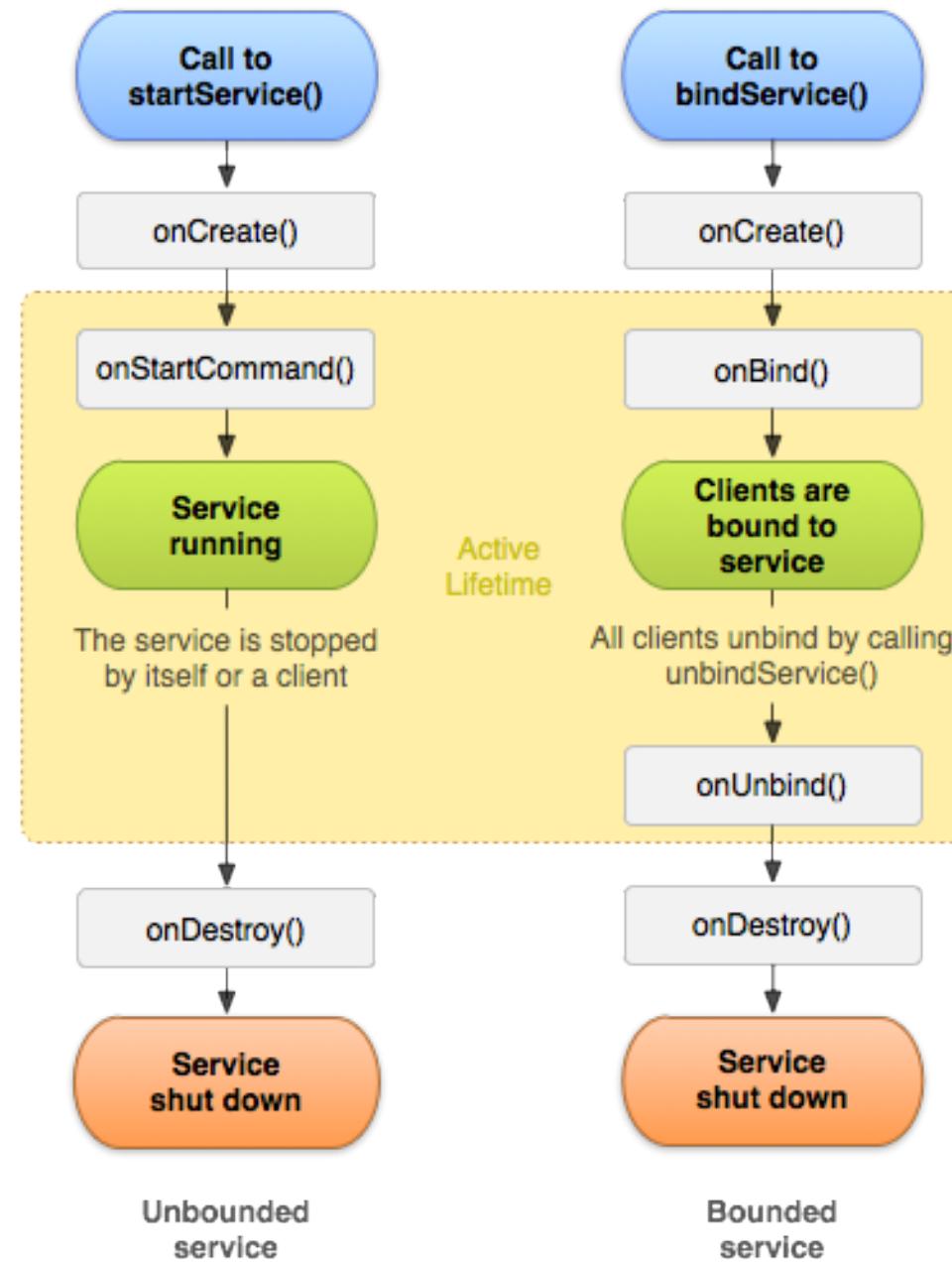
# Started e Bound Services

- Started: Um Service que você inicia do seu aplicativo explicitamente.

```
Intent intent = new Intent(this, HelloService.class);  
startService(intent);
```

- Bound: Um Service que se “vincula” do seu aplicativo para receber eventos

```
Intent intent = new Intent(this, BTService.class);  
Messenger messenger = new Messenger(myHandler);  
intent.putExtra("MESSENGER", messenger);  
bindService(intent, myConnection,  
Context.BIND_AUTO_CREATE);
```



# IntentService

1. Inicia (onHandleIntent())
2. Executa as tarefas em segundo plano
3. Termina

Tudo isso independentemente de o app estar em primeiro plano (aberto para o usuário)

# Exemplo com Notifications

```
public class ToastIntentService extends IntentService {  
    (...)  
  
    @Override  
    protected void onHandleIntent(Intent workIntent) {  
        mNotificationManager =  
            (NotificationManager) getSystemService(NOTIFICATION_SERVICE);  
  
        showNotification("Estou aqui");  
  
        sleep();  
  
        showNotification("Ainda estou aqui");  
  
        sleep();  
  
        showNotification("Tchau!");  
  
        sleep();  
    }  
    (...)  
}
```

# IntentService no Manifest

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="androidcourse.venturus.org.br.androidcourse_firstexample_likecounter">

    <application (...)>
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

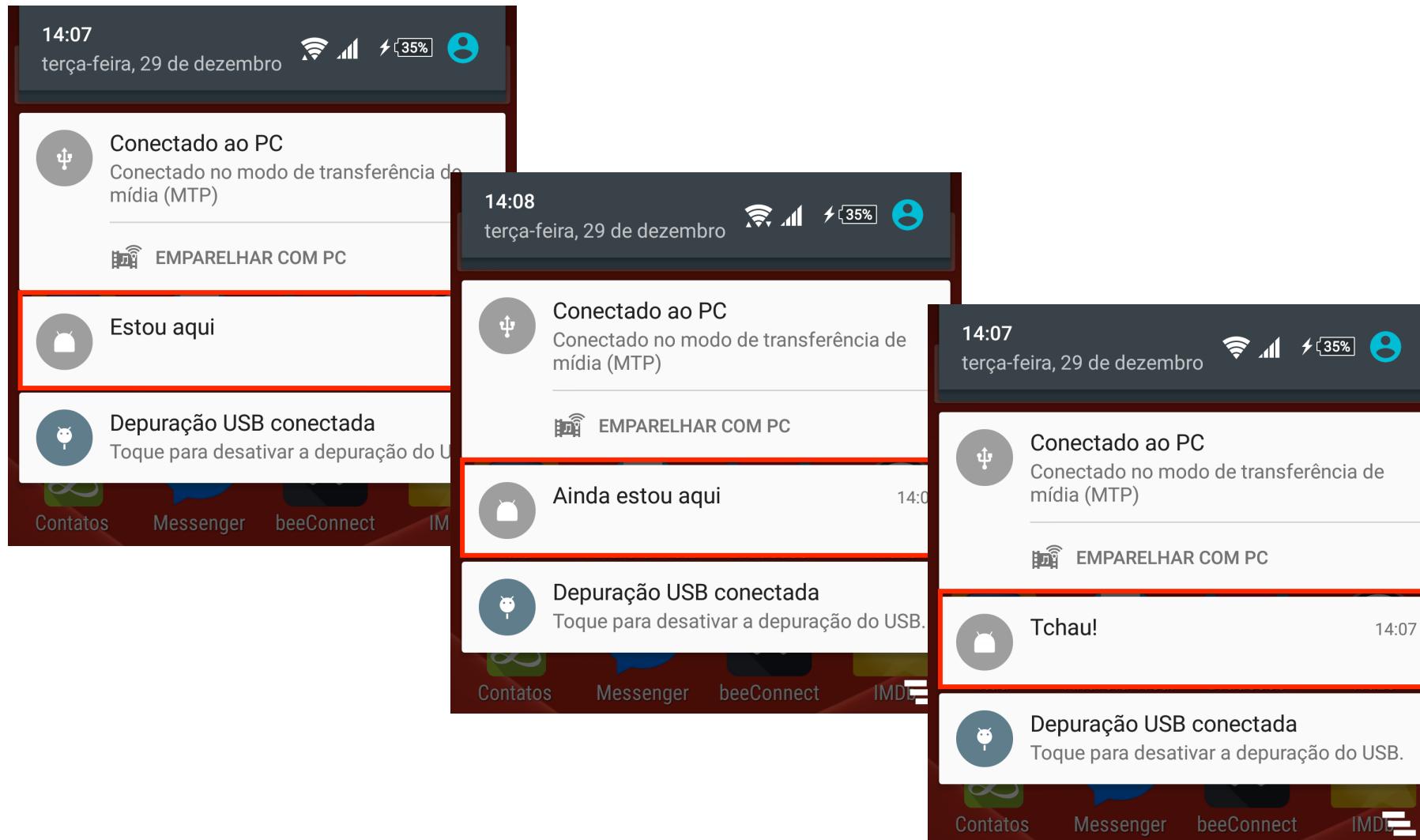
        <activity android:name=".PictureActivity"></activity>

        <service
            android:name=".ToastIntentService"
            android:exported="false"/>

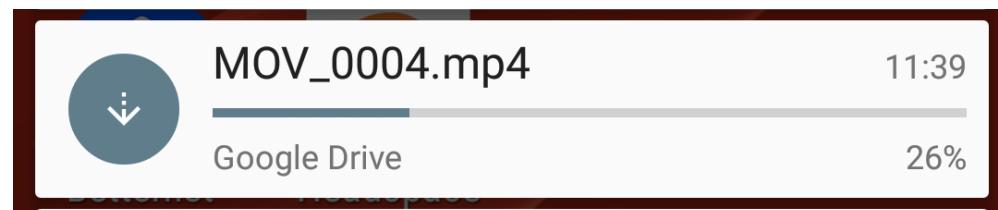
    </application>

</manifest>
```

# Notificações Mesmo com o App “fechado”



# Service de Sistema: DownloadManager



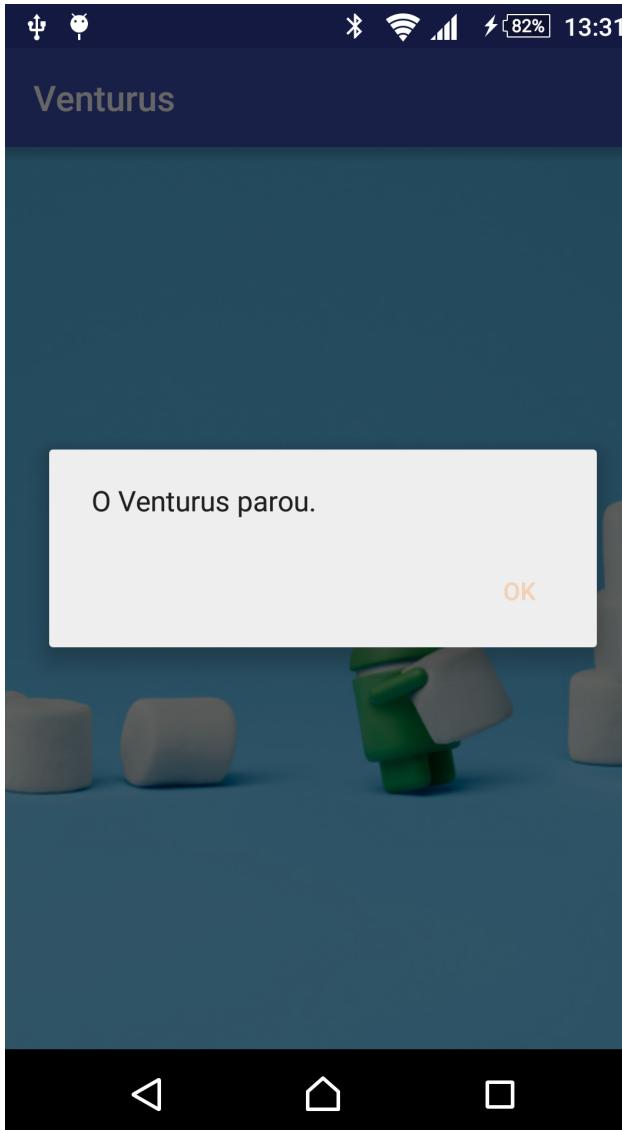
- Cria e atualiza a notificação acima
- Executa o download (HTTP)
- Faz multiplas tentativas caso falhe (retry)
- Continua um download se o celular for reiniciado

# Em PictureActivity > onCreate()

```
ImageView image = (ImageView) findViewById(R.id.marshmallow_big_image);

image.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        DownloadManager downloadManager=(DownloadManager)
getSystemService(DOWNLOAD_SERVICE);
        DownloadManager.Request request = new DownloadManager.Request(
            Uri.parse("https://goo.gl/RnPVhk"));
        request.setTitle(getResources().getString(R.string.download_text));
        downloadId = downloadManager.enqueue(request);
    }
});
```

# Crash!!!



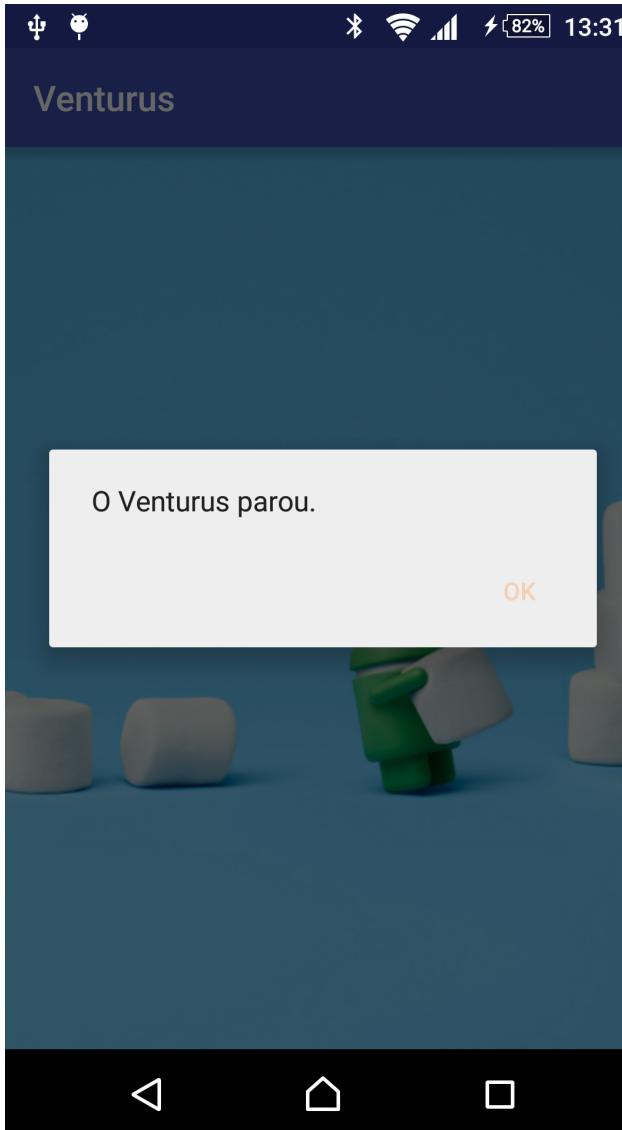
# Permissões

- Um aplicativo necessita declarar o uso de permissões para usar qualquer funcionalidade que possa impactar os dados ou a experiência do usuário.
- O uso das permissões é declarado no Manifest:
  - android.permission.INTERNET
  - android.permission.SEND\_SMS
  - android.permission.READ\_CONTACTS

# Declarando Permissões

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="androidcourse.venturus.org.br.androidcourse_firstexample_likecounter">  
  
    <uses-permission android:name="android.permission.INTERNET" />  
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
  
    <application  
        android:allowBackup="true"  
        (...)  
        </application>  
  
</manifest>
```

# Crash aínda!!!



# Permissões “Perigosas” – Android 6.0

- A partir do Android 6.0, apps com target SDK >= 23
- Lista de permissões “perigosas” – **WRITE\_EXTERNAL\_STORAGE** é uma.
- Deve ser requisitada a permissão **quando o usuário for utiliza-la**
- Código necessário para:
  1. Verificar se a permissão já foi concedida
  2. Requisitar a permissão caso não tenha sido concedida
  3. Mostrar mensagem explicativa sobre a permissão caso necessário
  4. Mostrar uma mensagem de erro caso o usuário não conceda a permissão

# Permissões “Perigosas” - Referências

Resumo sobre Permissões

<https://developer.android.com/training/permissions/index.html>

Lista de permissões **normal/dangerous**

<https://developer.android.com/guide/topics/security/permissions.html#normal-dangerous>

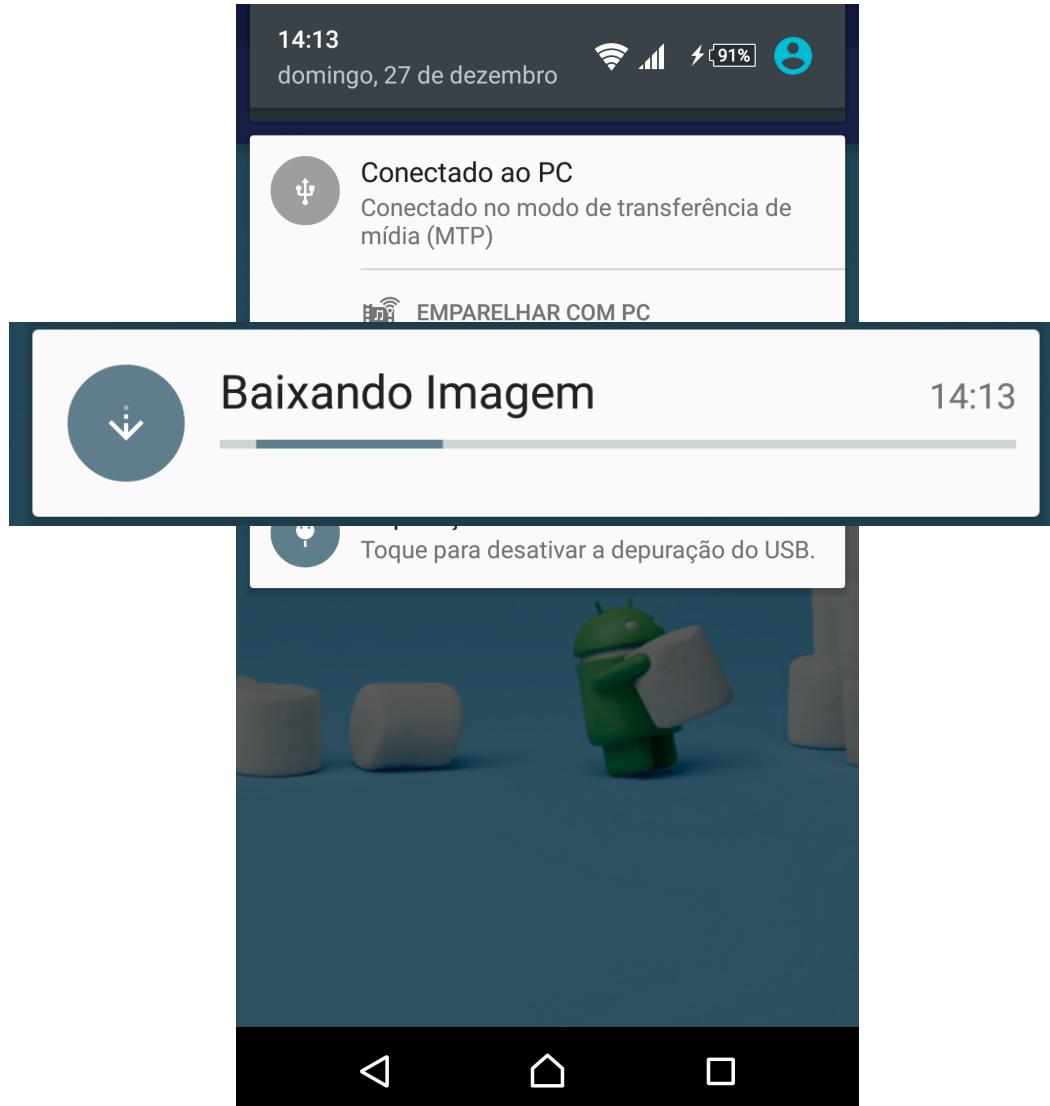
# Conferindo / Requisitando Permissão

```
private void checkPermissionAndDownload() {  
    // Here, thisActivity is the current activity  
    if (ContextCompat.checkSelfPermission(this,  
        Manifest.permission.WRITE_EXTERNAL_STORAGE)  
        != PackageManager.PERMISSION_GRANTED) {  
  
        // Should we show an explanation?  
        if (ActivityCompat.shouldShowRequestPermissionRationale(this,  
            Manifest.permission.WRITE_EXTERNAL_STORAGE)) {  
            Toast.makeText(this, R.string.permission_explanation_text,  
Toast.LENGTH_LONG).show();  
        }  
        ActivityCompat.requestPermissions(this,  
            new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE},  
            MY_PERMISSIONS_REQUEST_WRITE_EXTERNAL_STORAGE);  
    } else {  
        doDownload();  
    }  
}
```

# Tratando “Resposta” do Usuário

```
@Override  
public void onRequestPermissionsResult(int requestCode,  
                                      String permissions[], int[] grantResults) {  
    switch (requestCode) {  
        case MY_PERMISSIONS_REQUEST_WRITE_EXTERNAL_STORAGE: {  
            if (grantResults.length > 0  
                && grantResults[0] == PackageManager.PERMISSION_GRANTED) {  
                doDownload();  
            } else {  
                Toast.makeText(this, R.string.permission_denied_text,  
Toast.LENGTH_LONG).show();  
            }  
            return;  
        }  
    }  
}
```

# Baixando! Mas sem aviso quando acaba?



# Broadcast Receivers

- Recebem “avisos” do sistema ou do próprio aplicativo
- O aplicativo se “registra” para receber tipos específicos de “avisos” (broadcasts)
- Exemplos de broadcasts:
  - android.intent.action.BOOT\_COMPLETED
  - android.net.wifi.WIFI\_STATE\_CHANGED
  - android.intent.action.DOWNLOAD\_COMPLETE

# Criar classe extendendo BroadcastReceiver

```
public class DownloadCompleteReceiver extends BroadcastReceiver {

    private long mDownloadId = -1;

    public DownloadCompleteReceiver(long downloadId) {
        mDownloadId = downloadId;
    }

    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        if (DownloadManager.ACTION_DOWNLOAD_COMPLETE.equals(action)) {
            long downloadId = intent.getLongExtra(
                DownloadManager.EXTRA_DOWNLOAD_ID, 0);

            if (downloadId == mDownloadId) {
                Toast toast =
                    Toast.makeText(context, R.string.download_completed_text, Toast.LENGTH_LONG);
                toast.show();
            }
        }
    }
}
```

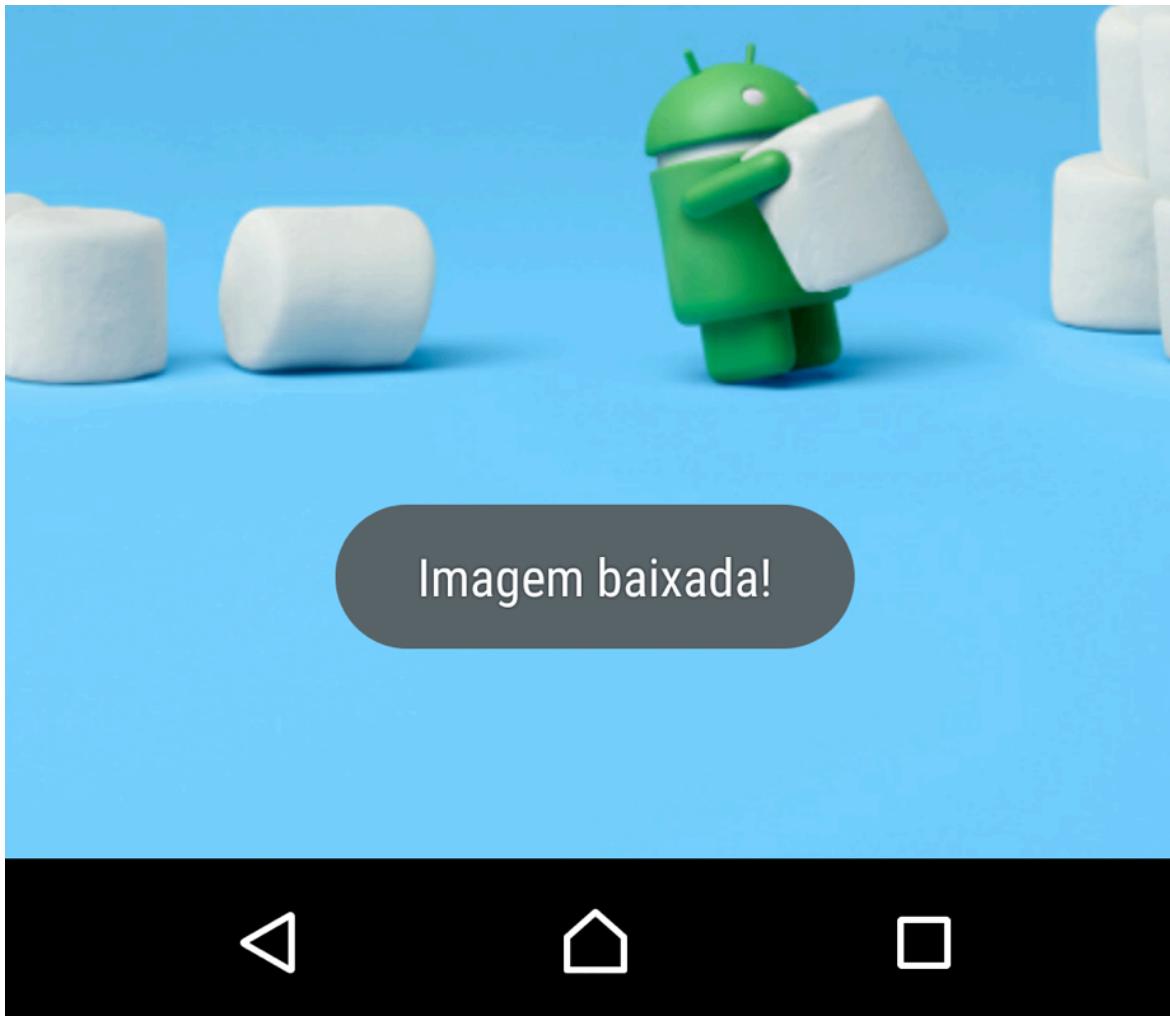
# Registrando o BroadcastReceiver

```
DownloadManager downloadManager = (DownloadManager)
getSystemService(DOWNLOAD_SERVICE);
DownloadManager.Request request = new DownloadManager.Request(
    Uri.parse("https://goo.gl/RnPVhk"));
request.setTitle(getResources().getString(R.string.download_text));

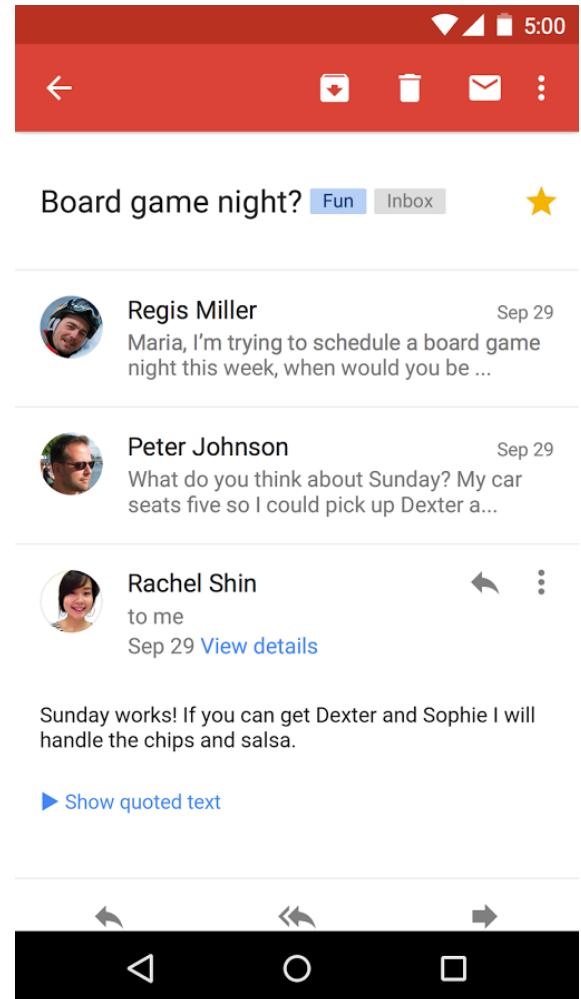
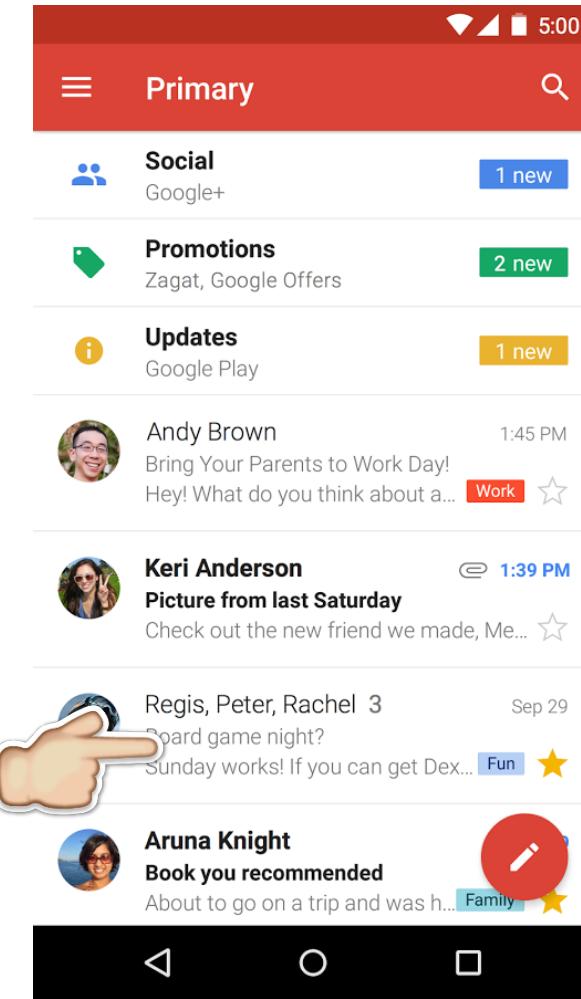
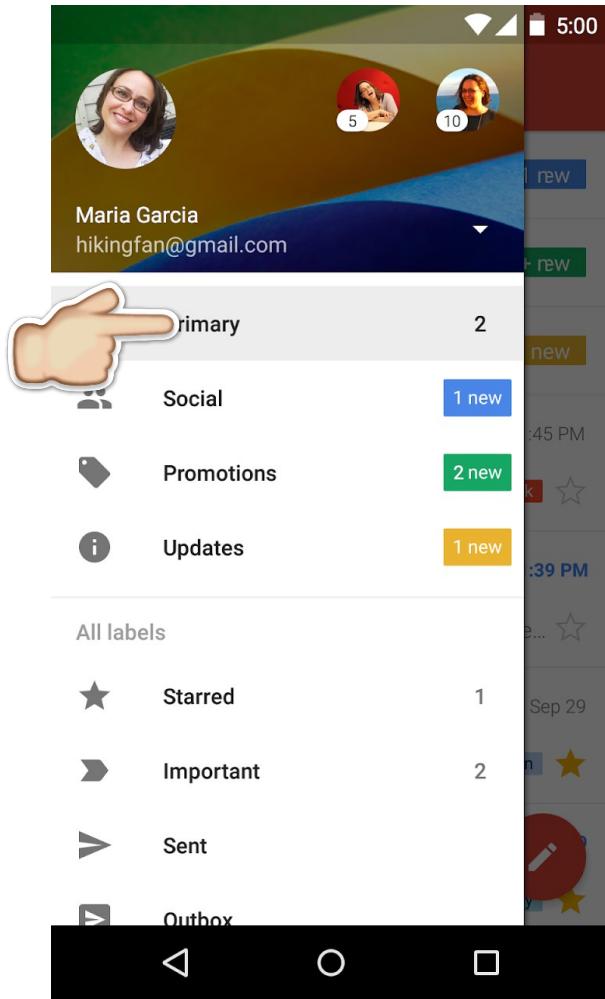
downloadId = downloadManager.enqueue(request);

registerReceiver(new DownloadCompleteReceiver(downloadId), new IntentFilter(
    DownloadManager.ACTION_DOWNLOAD_COMPLETE));
```

# Feito!!

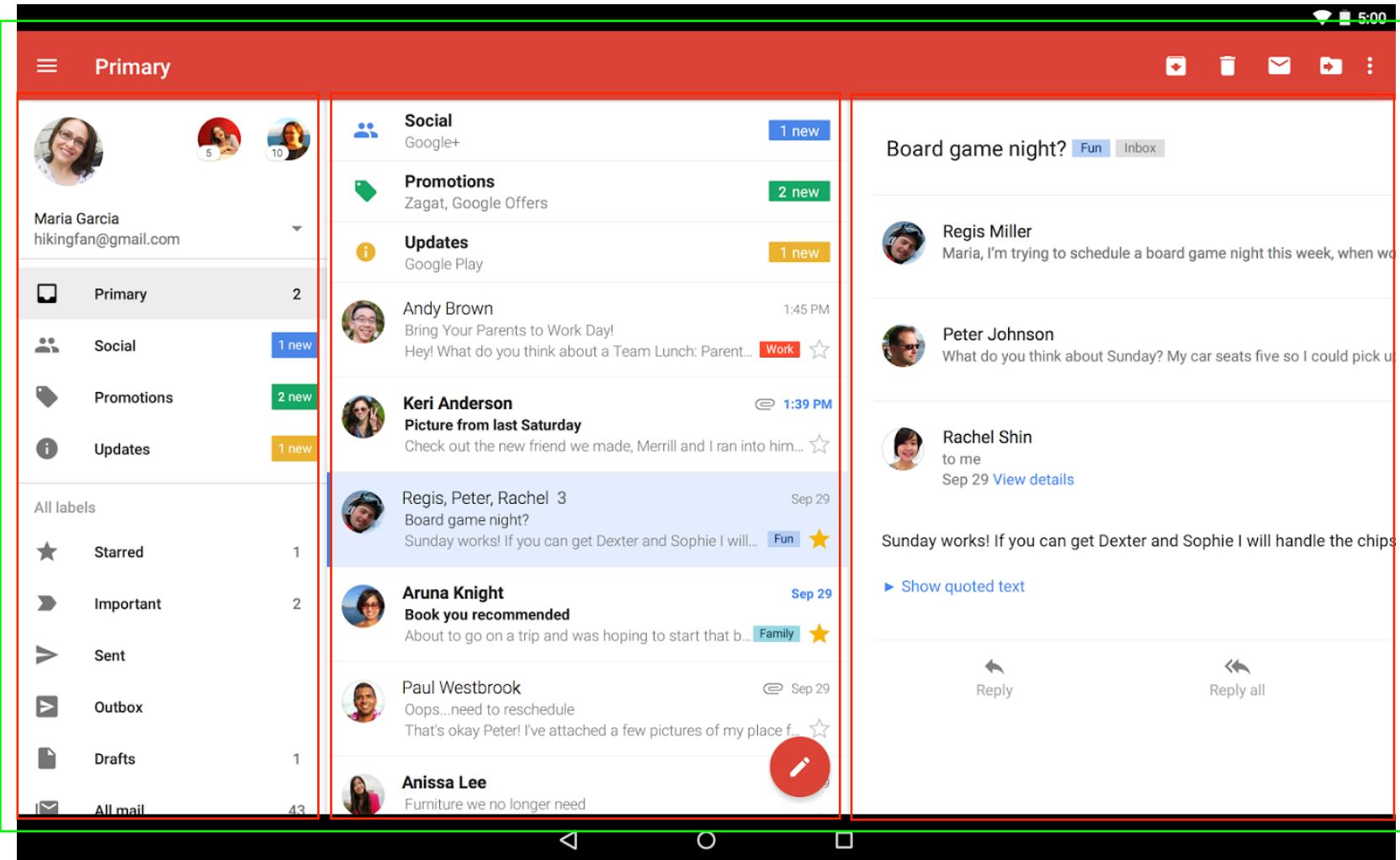


# Gmail no Smartphone



# Gmail no Tablet

Activity



Fragment 1

Fragment 2

Fragment 3

# Fragments – Classe Java

```
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.ViewGroup;

public class ArticleFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.article_view, container, false);
    }
}
```

Fonte: <http://developer.android.com/intl/pt-br/training/basics/fragments/creating.html>

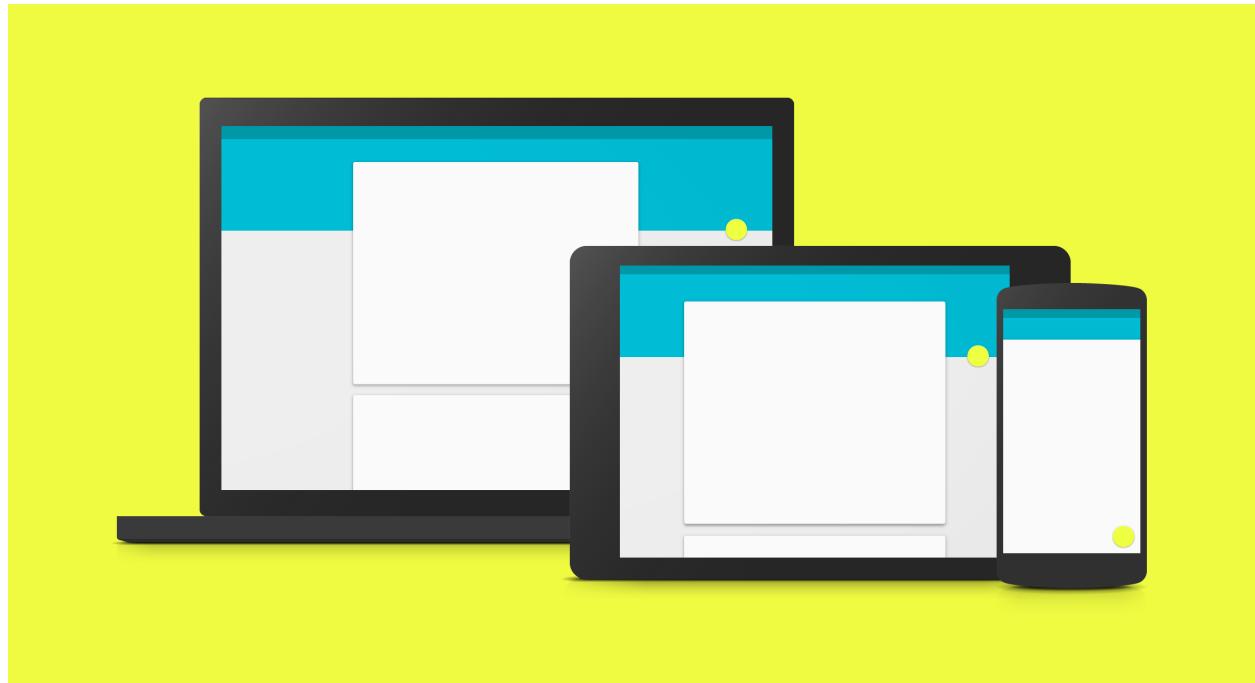
# Fragments – No Layout (XML)

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="horizontal"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent">  
  
    <fragment android:name="com.example.android.fragments.HeadlinesFragment"  
        android:id="@+id/headlines_fragment"  
        android:layout_weight="1"  
        android:layout_width="0dp"  
        android:layout_height="match_parent" />  
  
    <fragment android:name="com.example.android.fragments.ArticleFragment"  
        android:id="@+id/article_fragment"  
        android:layout_weight="2"  
        android:layout_width="0dp"  
        android:layout_height="match_parent" />  
  
</LinearLayout>
```

Fonte: <http://developer.android.com/intl/pt-br/training/basics/fragments/creating.html>

# Material Design

*“Criar uma linguagem visual que sintetize princípios clássicos do bom design com a inovação e possibilidades da tecnologia.”*

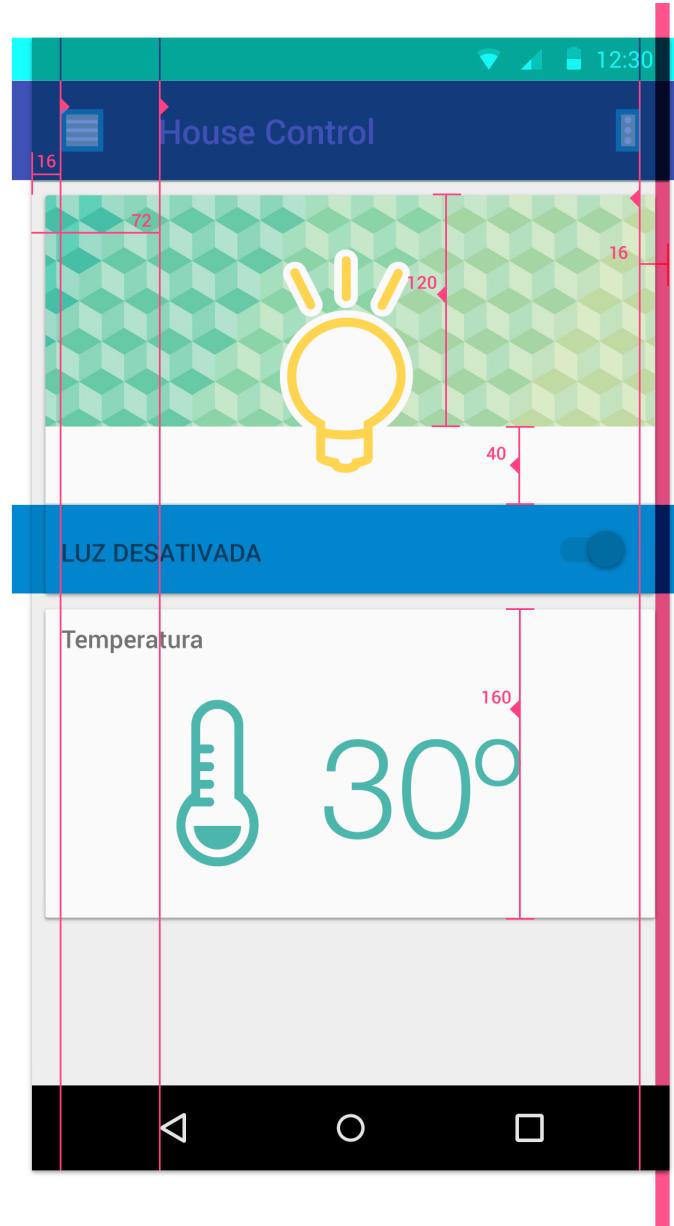
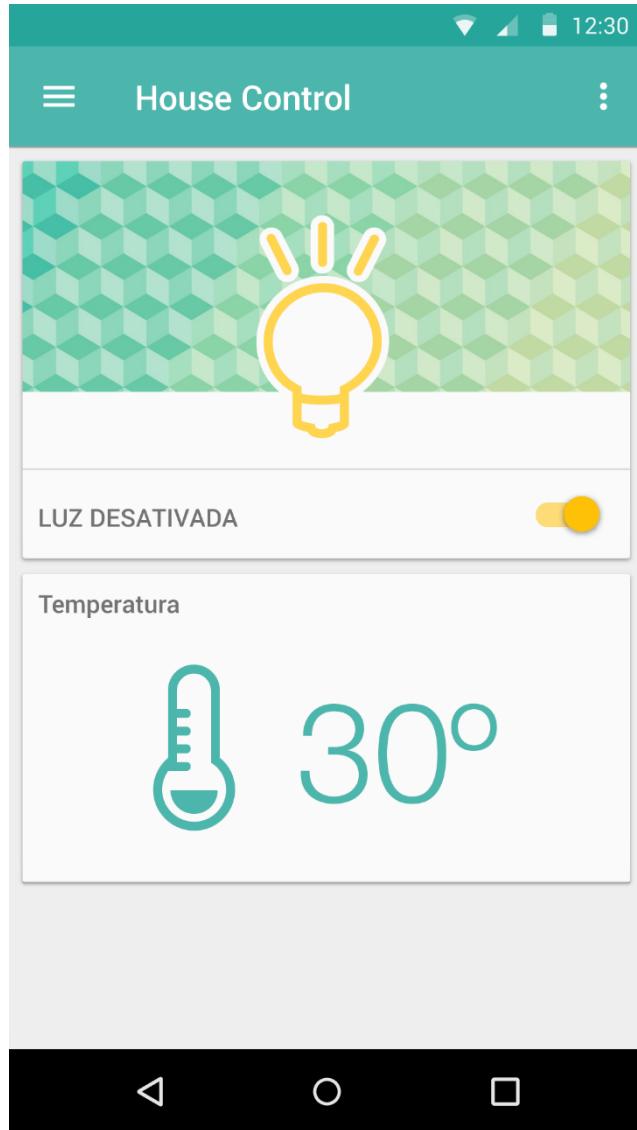


# Especificações do Material Design

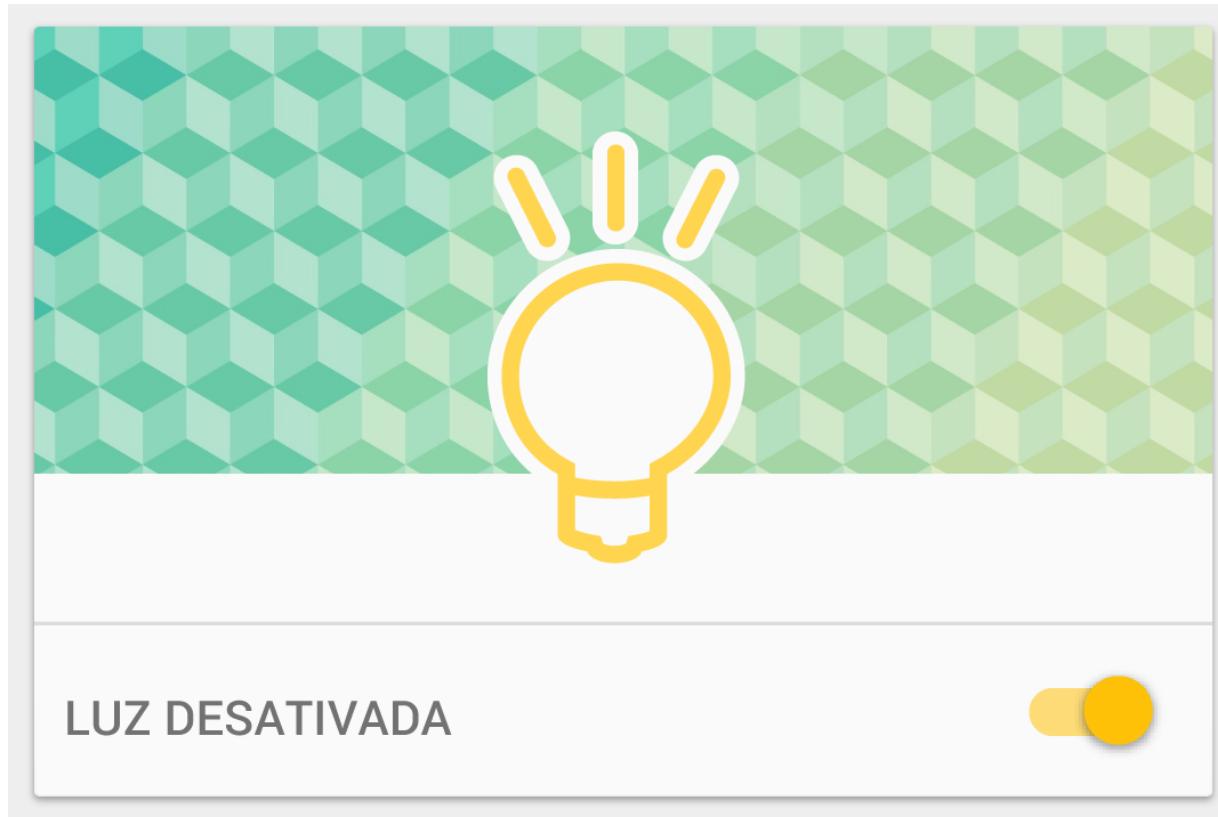
■ <http://www.google.com/design/spec/material-design/introduction.html#>

- Animações
- Estilo
- Layout
- Componentes
- Etc...

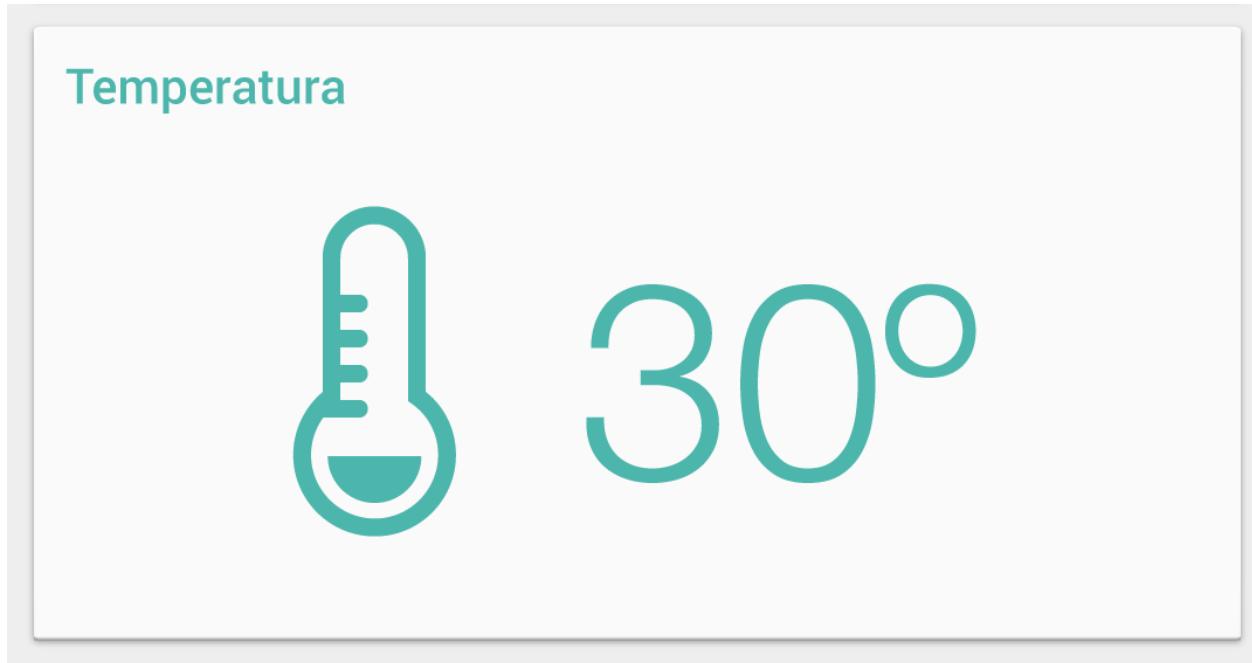
# Aplicativo House Control



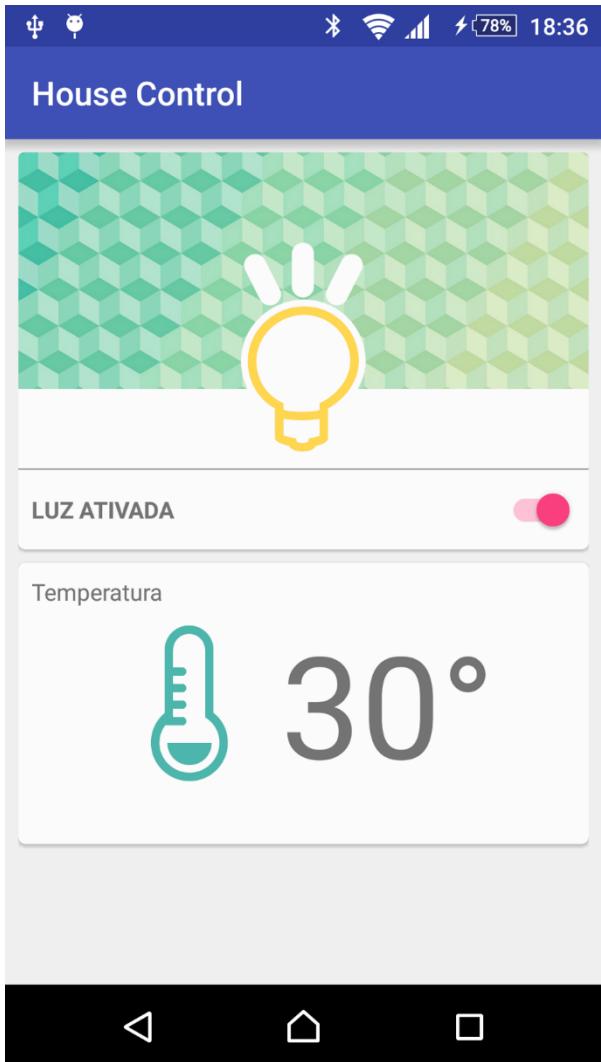
# Card Superior



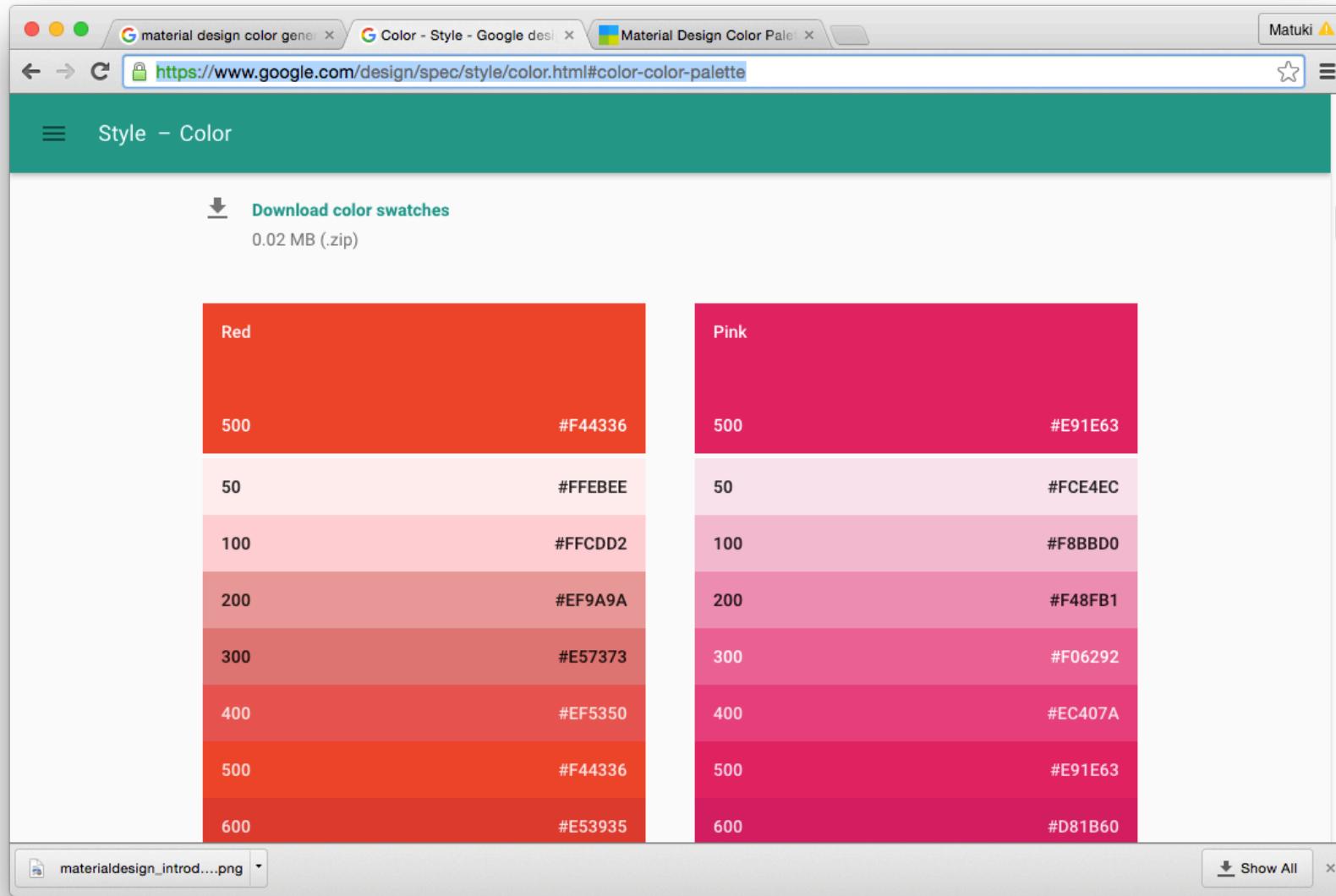
# Card Inferior



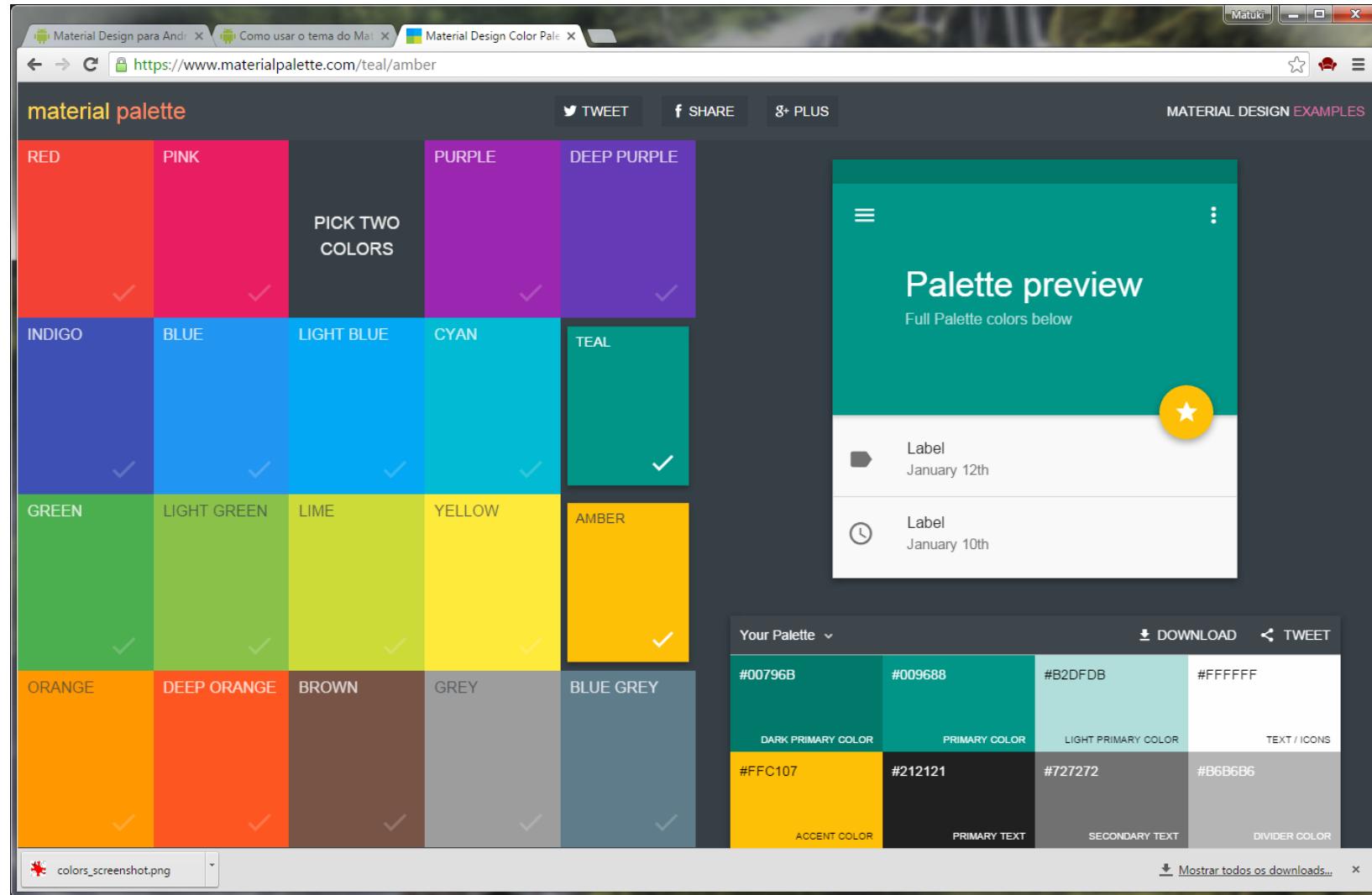
# Diferenças?



<https://www.google.com/design/spec/style/>



# <https://www.materialpalette.com/>



# colors.xml



The screenshot shows the contents of an XML file named "colors.xml". The file defines three color resources: "colorPrimary" (#009688), "colorPrimaryDark" (#00796B), and "colorAccent" (#FFC107). The code is written in XML, with tags in blue and attribute values in green.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#009688</color>
    <color name="colorPrimaryDark">#00796B</color>
    <color name="colorAccent">#FFC107</color>
</resources>
```

# Ajustes Finais

- Fixar a orientação do app no modo “retrato”
- Mudar fundo do card superior e texto de acordo com o switch (ligado/desligado)
- Adicionar o ícone (launcher) do aplicativo

# Fixando o app no modo “retrato” (portrait)

```
<activity android:name=".MainActivity"
    android:configChanges="orientation"
    android:screenOrientation="portrait">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

# “Ligando/Desligando” no Card Superior

```
public void onSwitchClicked(View view) {  
    Switch switchView = (Switch) view;  
  
    TextView switchText = (TextView) findViewById(R.id.lamp_switch_text);  
    ImageView cardBackground = (ImageView) findViewById(R.id.card_bg);  
    ImageView lightBulb = (ImageView) findViewById(R.id.lamp);  
  
    if (switchView.isChecked()) {  
        switchText.setText(R.string.switch_text_on);  
  
        cardBackground.setImageResource(R.drawable.bgon);  
  
        lightBulb.setImageResource(R.drawable.icon_luz_on);  
    } else {  
        switchText.setText(R.string.switch_text_off);  
  
        cardBackground.setImageResource(R.drawable.bgoff);  
  
        lightBulb.setImageResource(R.drawable.icon_luz_off);  
    }  
}
```

# Ícone “Launcher”

- Pastas em src/main/res/
  - /mipmap-hdpi
  - /mipmap-xhdpi
  - /mipmap-xxhdpi



# Obrigado!

