

# 1º Congresso da Escola Politécnica

**28 a 30/09/15**





# Desenvolvimento de Aplicativos Móveis para Android

Aluno Luiz Carlos D. de Jesus



# Desenvolvimento de Aplicativos Móveis para Android



- Introdução
- Mercado Mobile
- Sistemas Operacionais
- Oportunidades
- Wearables – Tecnologias Vestíveis
- Automóveis
- Smart TVs
- Internet das Coisas
- Casos de Sucesso
- Desenvolvimento Hibrido ou Nativo?
- Android Studio e Android SDK
- Google Play
- Ciclo de Vida das Aplicações
- Exemplos Práticos



# Introdução

- Os Aplicativos móveis já são uma realidade no nosso cotidiano.
- Quase um terço da população mundial está conectada através de smartphones.



# Mercado Mobile

- O Brasil possui a quinta maior base de smartphones do mundo, cerca de 90 milhões de aparelhos.
- Número maior que a população de países como França, Alemanha e Espanha.



# Mercado Mobile

- De acordo com a International Data Corporation (IDC), as vendas de smartphones e tablets vão continuar crescendo, atingindo US\$ 484 bilhões em 2015.
- O desenvolvimento empresarial de aplicativos móveis deverá dobrar.
- Os aplicativos devem continuar em crescimento mesmo não sendo na mesma velocidade do passado.
- Novas plataformas se desenvolverão a partir de agora.
- Uma grande parte delas deve se basear nas atuais tecnologias mobile.



# Sistemas Operacionais

- Android
- IOS
- Windos Phone
- Tizen
- Firefox OS
- Ubuntu Phone
- Blackberry OS



# Oportunidades

- Surge uma grande oportunidade para as marcas e empresas conquistarem e estarem cada vez mais próximas de seus clientes.
- Muitas marcas importantes já desenvolveram aplicativos para abrir mais um canal de relacionamento com seus clientes.



# Wearables – Tecnologias Vestíveis

- Atualmente relógios podem:
  - monitorando atividades físicas
  - realizando funções básicas dos smartphones.
  - Serem utilizadas como complemento ao smartphone.



# Wearables – Tecnologias Vestíveis

- Apesar da simplicidade e alto custo atual, esse é o inicio de uma transformação.
- Sistemas como android e ios já estão presentes em relógios.
- Esses são os primeiros dispositivos vestíveis – Wearables.



# Automóveis

- O Android Auto foi projetado tendo em mente a segurança.
- Com uma interface simples e intuitiva, controles integrados no volante e as novas e avançadas ações de voz.
- Foi projetado para minimizar as distrações para que você possa manter o foco na estrada.



# Automóveis

- Já está disponível na Austrália, Canadá, França, Alemanha, Irlanda, Itália, México, Nova Zelândia, Espanha, Reino Unido e Estados Unidos.
- Com o apoio das principais montadores.



Audi



Volkswagen



# Smart TVs

- Algumas Smart TVs já incluem o sistema operacional android.
- Smart TV Box
  - Transformam uma TV comum em uma Smart TV com android.



# Internet das Coisas

- Estamos vivendo uma fase de transição onde eletrodomésticos, carros e quase tudo estará conectado, facilitando o nosso dia a dia.



# Internet das Coisas

- Já temos geladeiras inteligentes como as da Samsung.
  - É possível navegar na internet em uma tela localizada na porta.
  - Utilizar alguns apps específicos, mas tudo ainda é muito fechado e precários.
- A humanidade daqui pra frente, usará sua criatividade para descobrir novas e revolucionárias formas de integrar essas tecnologias.
- Um novo mundo nos espera, com novas oportunidades.



# Casos de Sucesso



UBER



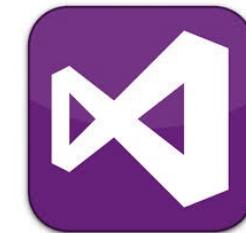
# Desenvolvimento Hibrido ou Nativo?

- Uma das primeiras dificuldades que existem no desenvolvimento mobile é escolher qual plataforma de desenvolvimento utilizar.
- Desenvolvimento Híbrido ou Nativo?



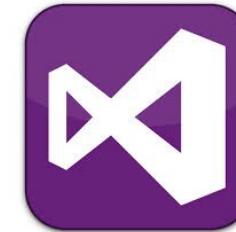
# Desenvolvimento Hibrido

- Combina linguagens já conhecidas das aplicações web como HTML, CSS e Javascript e funcionalidades nativas.
- Apresenta resultado satisfatório em algumas situações.
- Multiplataforma
  - Escreva uma vez e rode em vários sistemas operacionais.



# Desenvolvimento Hibrido vs Nativo

- Desenvolvimento Nativo
  - Utilizando uma linguagem própria para o desenvolvimento específico.
  - Java para Android
  - Objective C para iOS
  - C# para Windows Phone



# Desenvolvimento Hibrido vs Nativo

- Uma aplicação institucional, para apresentação de negócios e sem recursos avançados poderá usar o desenvolvimento híbrido.
- Em caso de aplicações mais complexas:
  - Acesso à webservices.
  - Geolocalização.
  - Notificações.
  - Recursos avançados do sistema.



# Desenvolvimento Hibrido vs Nativo

- A opção de desenvolvimento mais segura seria a nativa.
  - Pelo desempenho e robustez.
  - Documentação Oficial.
  - Recursos Exclusivos.
- Embora as plataformas hibridas avancem cada vez mais.
- Também é possível combinar uma interface híbrida com um back-end nativo.



# Limitações Phonegap (Híbrido)

	iOS iPhone / iPhone 3G	iOS iPhone 3GS and newer	Android	OS 4.6-4.7	OS 5.x	OS 6.0+	WebOS	Symbian	bada Bada
ACCELEROMETER	✓	✓	✓	✗	✓	✓	✓	✓	✓
CAMERA	✓	✓	✓	✗	✓	✓	✓	✓	✓
COMPASS	✗	✓	✓	✗	✗	✗	✗	✗	✓
CONTACTS	✓	✓	✓	✗	✓	✓	✗	✓	✓
FILE	✓	✓	✓	✗	✓	✓	✗	✗	✗
GEOLOCATION	✓	✓	✓	✓	✓	✓	✓	✓	✓
MEDIA	✓	✓	✓	✗	✗	✗	✗	✗	✗
NETWORK	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (ALERT)	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (SOUND)	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (VIBRATION)	✓	✓	✓	✓	✓	✓	✓	✓	✓
STORAGE	✓	✓	✓	✗	⚠	✓	✓	✓	✗



# Visual Studio

- Utiliza C# e .NET Framework, HTML e JavaScript ou C++
- Embora suporte outras plataformas, é a plataforma oficial apenas do Windows Phone utilizando o Windows Phone SDK.
- Desenvolvimento Hibrido (HTML / JavaScript)
  - Cordova
- Desenvolvimento Nativo ( .Net Framework)
  - Xamarin (Zamarin)
  - Xamarin IDE (MAC)



# Codename One

- Tecnologia Open Source
  - Planos Gratuitos e Pagos .
- Utiliza uma abordagem diferente da Hibrida para desenvolvimento multiplataforma.
- Nessa abordagem é desenvolvido um código Java nativo para android utilizando uma API própria em alternativa a oficial.

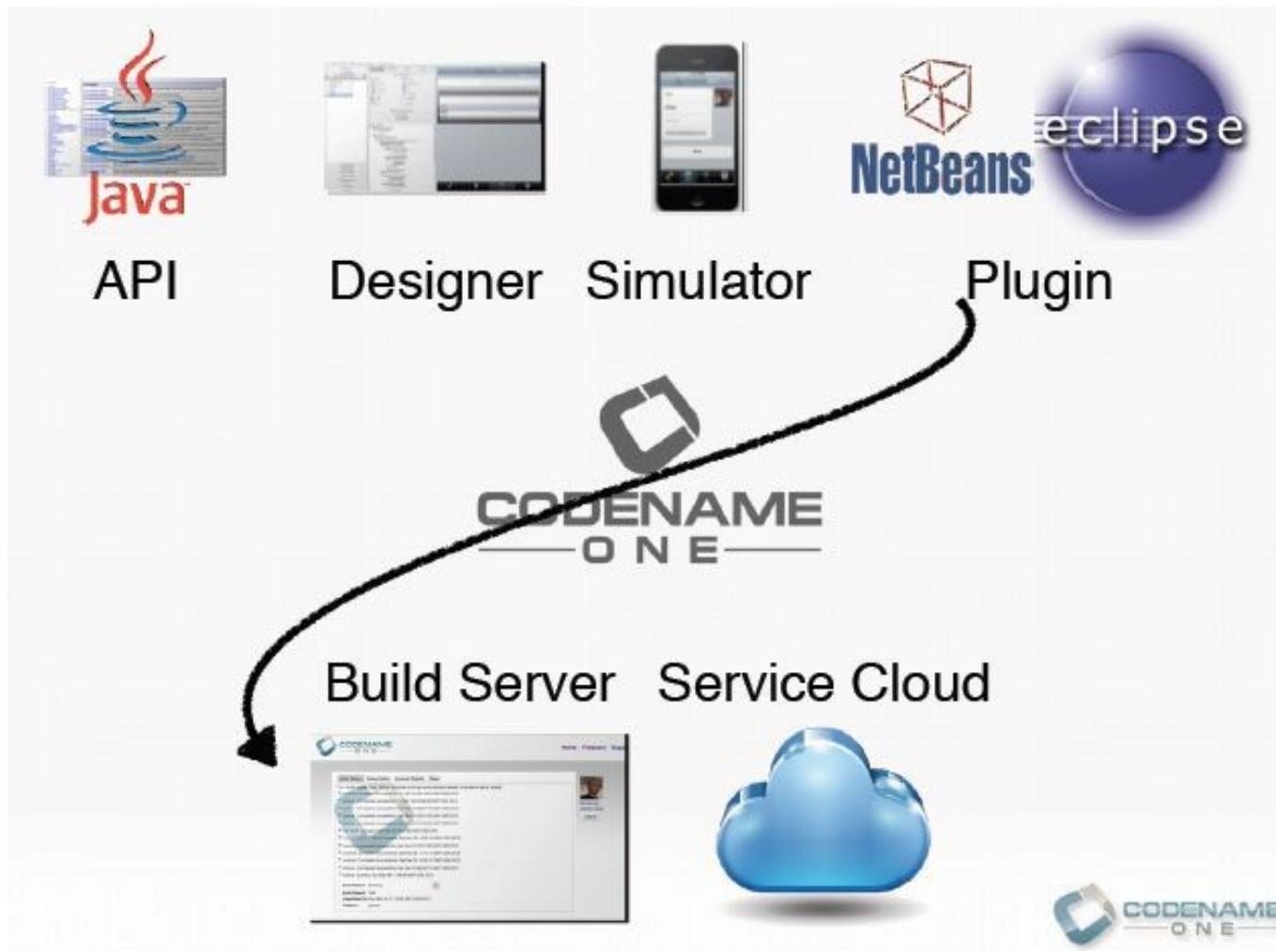


# Codename One

- É capaz de gerar código nativo automaticamente para outras plataformas:
  - Linguagem C para IOS.
  - Linguagem C# para Windows Phone
- Permite utilizar as IDEs NetBean, Eclipse ou IntelliJ
- Permite Acesso a Código Nativo.
  - Compensa funcionalidades ainda não implementadas em alguma plataforma.
- Também possibilita desenvolvimento desktop (Windows e Mac).



# Codename One



# Android SDK

- O Android SDK é o kit de desenvolvimento oficial para android.
- Distribuído inicialmente como um complemento ao eclipse.



# Android Studio

- Hoje o Android Studio se tornou a ide oficial, com o android sdk incluso.
- Ainda é possível utilizando o eclipse + android sdk.
- Pode-se desenvolver utilizando Windows, Linux ou MacOS.



# Android Studio

- A linguagem de desenvolvimento é Java.
- As interfaces são construídas através de marcações XML.
- Mesmo utilizando Java, desenvolver para Android exige conhecimento adicional.
- Documentação oficial:
  - <http://developer.android.com>

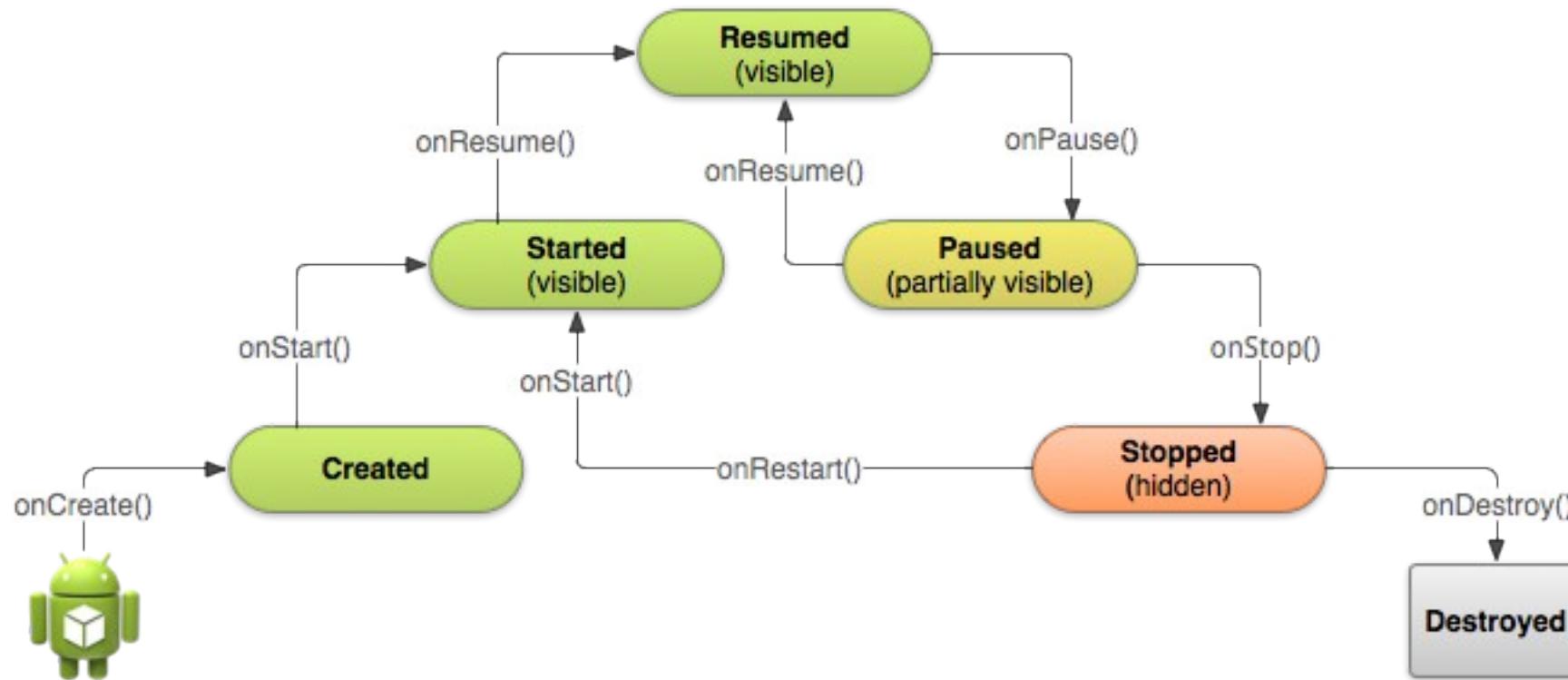


# Google Play

- Para se tornar um desenvolvedor licenciado pela Google e publicar aplicativos no Goole Play o valor é de U\$ 25.
- Além disso é cobrado a taxa de 30% do valor da venda do aplicativo ou conteúdo adicional pelo Google Play.



# Círculo de Vida das Aplicações



# Clico de Vida das Aplicações

- A classe Activity gerencia a interface com o usuário, ela quem recebe as requisições, trata e processa.
- Métodos:
  - **OnCreate()** - É a responsável por carregar os layouts XML e outras operações de inicialização. É executada apenas uma vez.
  - **onStart()** - É chamada após o onCreate() ou quando uma Activity que estava em background volta a ter foco.
  - **onResume()** - É chamada após o onStart(), ou quando uma Activity volta a ter foco.



# Clico de Vida das Aplicações

- A onStart() só é chamada quando a Activity não estava mais visível e volta a ter o foco, a onResume() é chamada nas “retomadas de foco”.
  - **onPause()** - É a primeira função a ser invocada quando a Activity perde o foco (isso ocorre quando uma nova Activity é iniciada).
  - **onStop()** - Só é chamada quando a Activity fica completamente encoberta por outra Activity.



# Clico de Vida das Aplicações

- **onDestroy()** - A última função a ser executada. Depois dela, a Activity é considerada “morta” – ou seja, não pode mais ser relançada. Se o usuário voltar a requisitar essa Activity, um novo objeto será construído.
- **onRestart()** - Chamada imediatamente antes da onStart(), quando uma Activity volta a ter o foco depois de estar em background.



# Exemplos Práticos

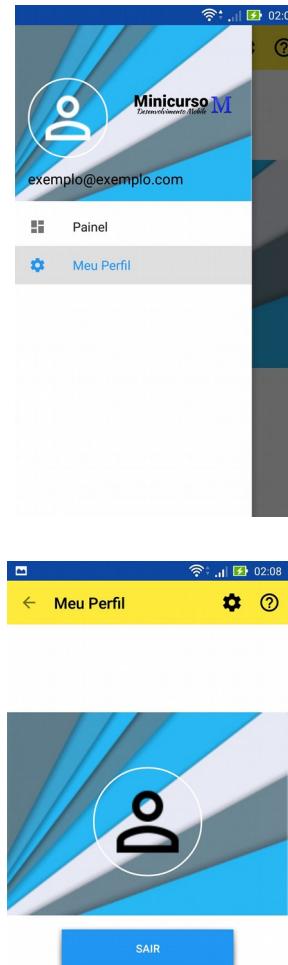


## Minicurso M

*Desenvolvimento Mobile*

Informações da Conta

Email	<input type="text"/>	
Senha	<input type="password"/>	
<input type="button" value="ENTRAR"/>		
<input type="button" value="ESQUECEU SUA SENHA?"/>		
<input type="button" value="CRIAR CONTA"/>		
Ainda não Implementado		



Informações da Conta

## Minicurso M

*Desenvolvimento Mobile*

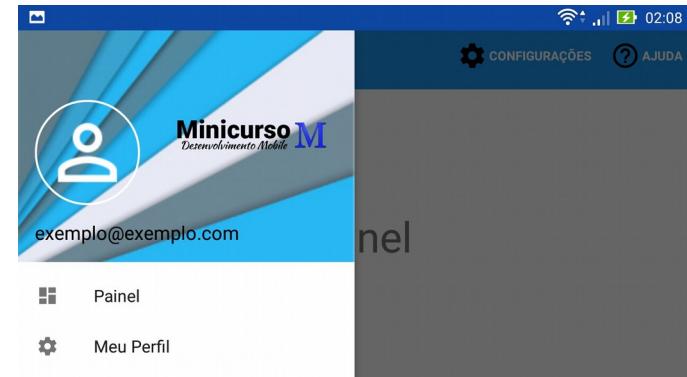
[ESQUECEU SUA SENHA?](#)

[CRIAR CONTA](#)



Email Obrigatório

[ENTRAR](#)

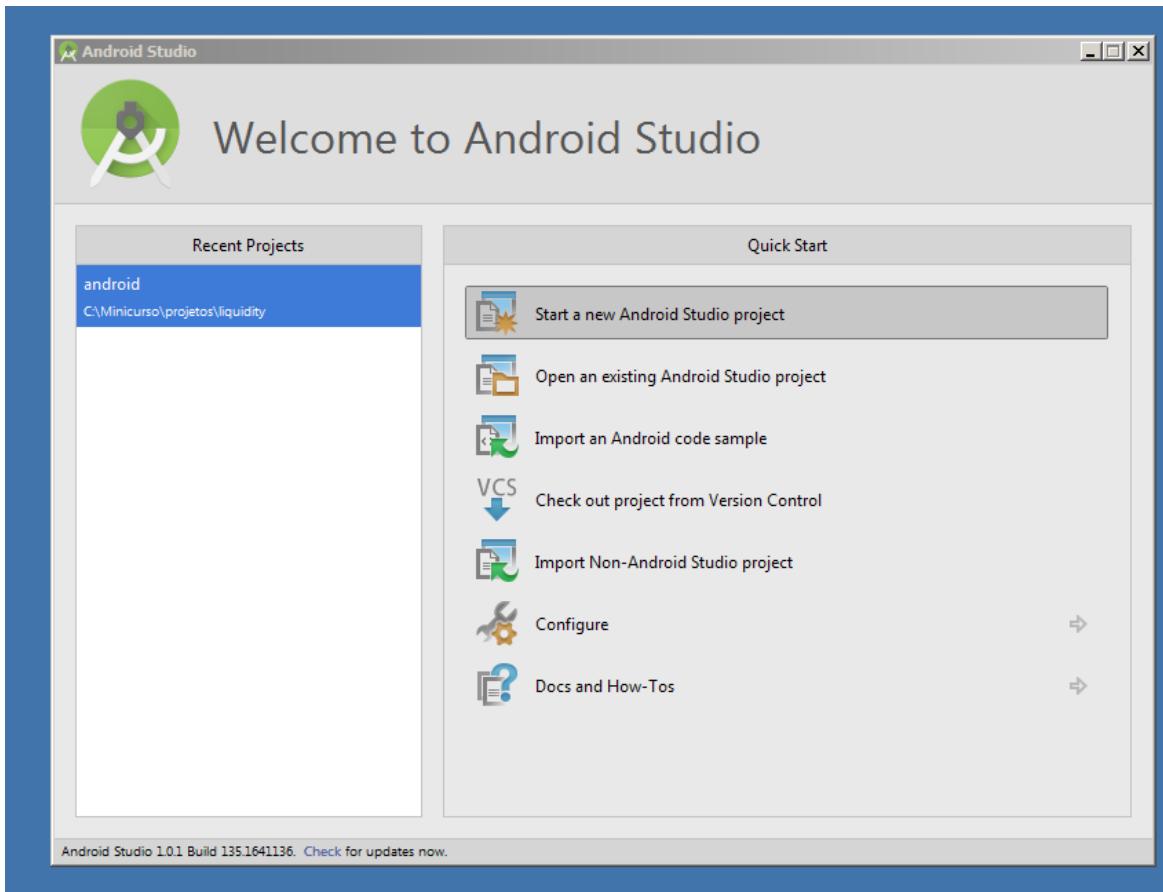


# Hello World

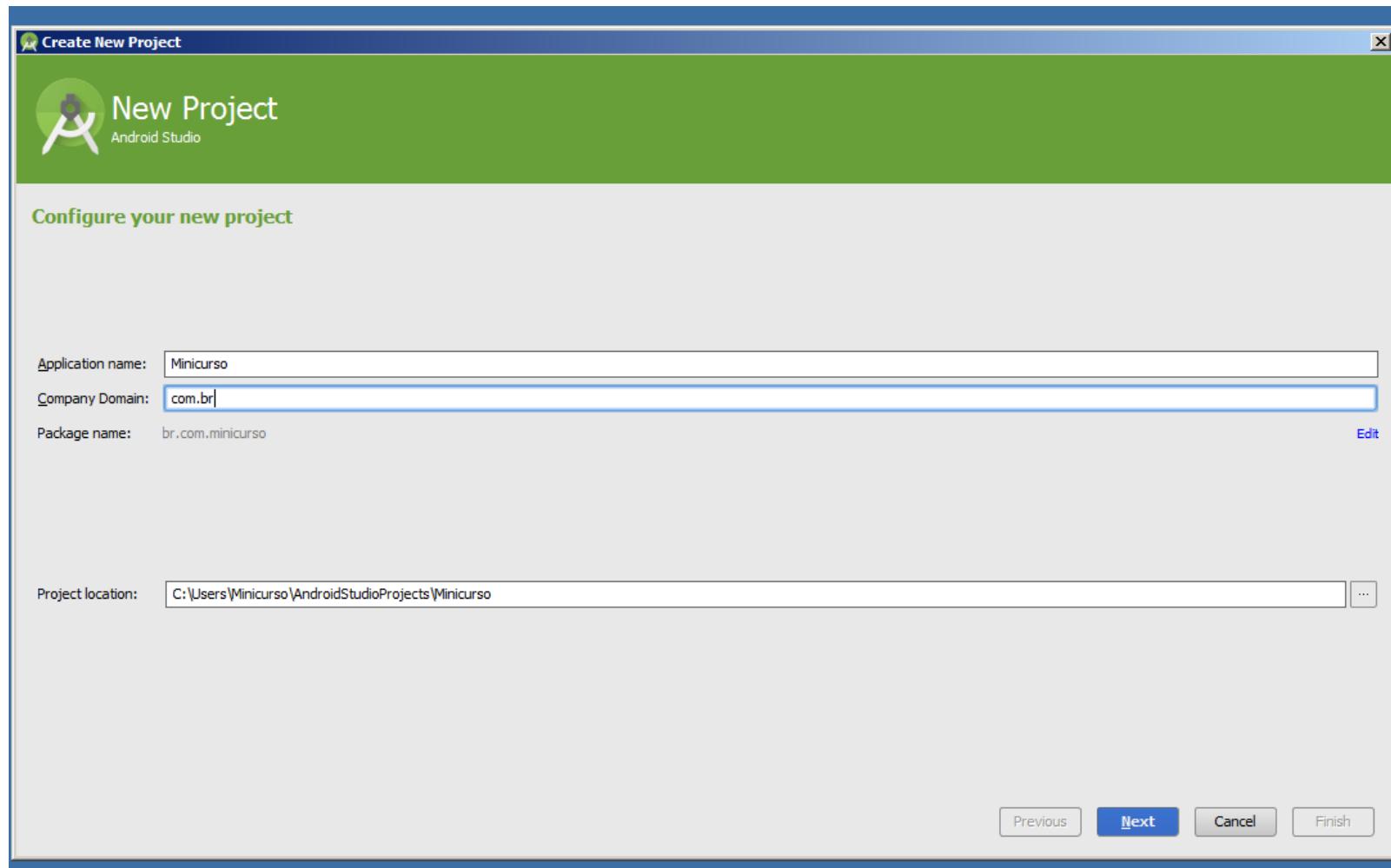


# Hello World

- Inicie o Android Studio e inicie um novo projeto



# Criar Projeto no Android Studio



# Criar Projeto no Android Studio

Create New Project

New Project

Android Studio

Select the form factors your app will run on

Different platforms require separate SDKs

Phone and Tablet

Minimum SDK API14: Android 4.0 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available. By targeting API14 and later, your app will run on approximately 87.9% of the devices that are active on the Google Play Store. [Help me choose.](#)

TV

Minimum SDK API 21: Android 5.0 (Lollipop)

Wear

Minimum SDK API 21: Android 5.0 (Lollipop)

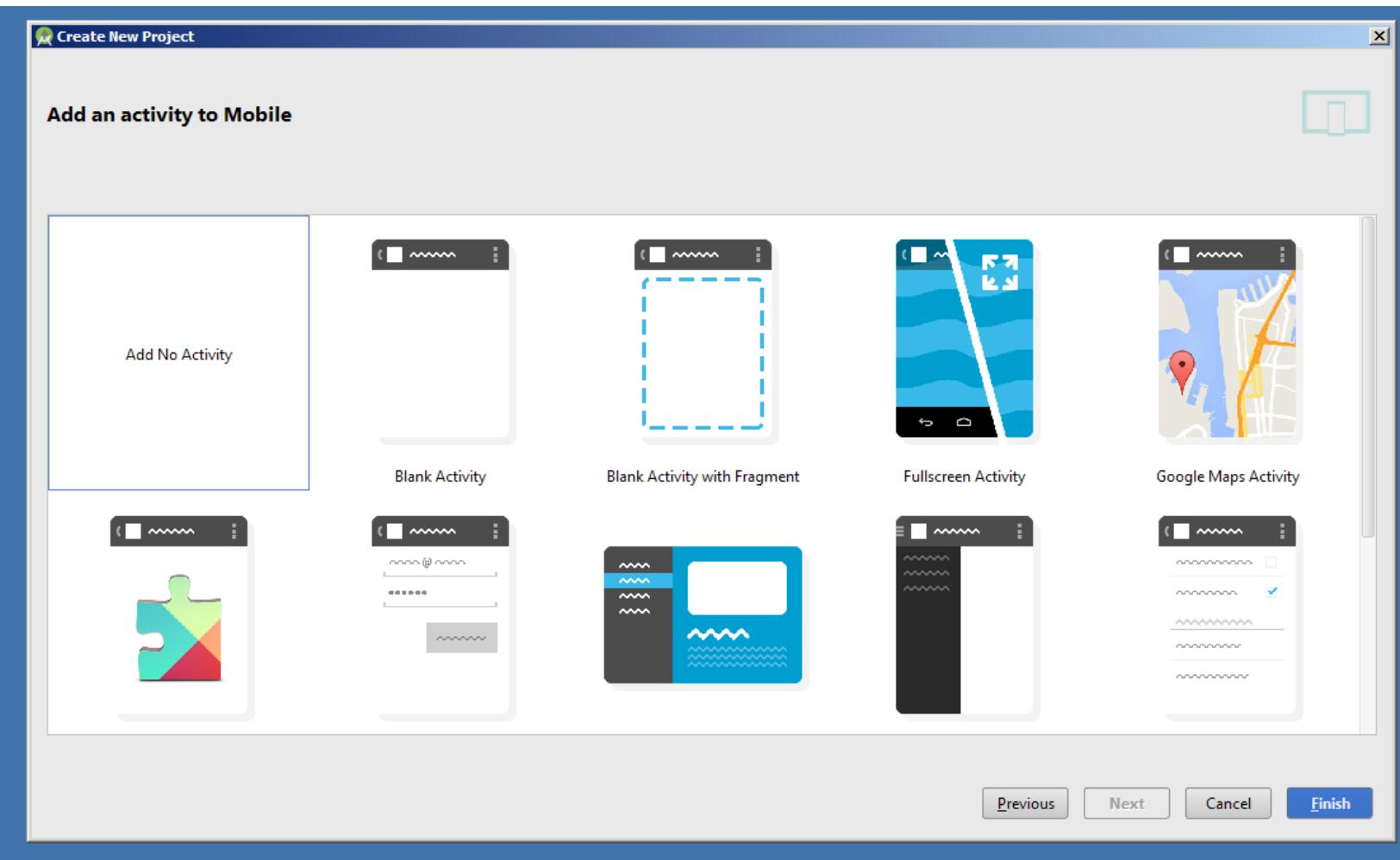
Glass (Not Installed)

Minimum SDK

Previous Next Cancel Finish

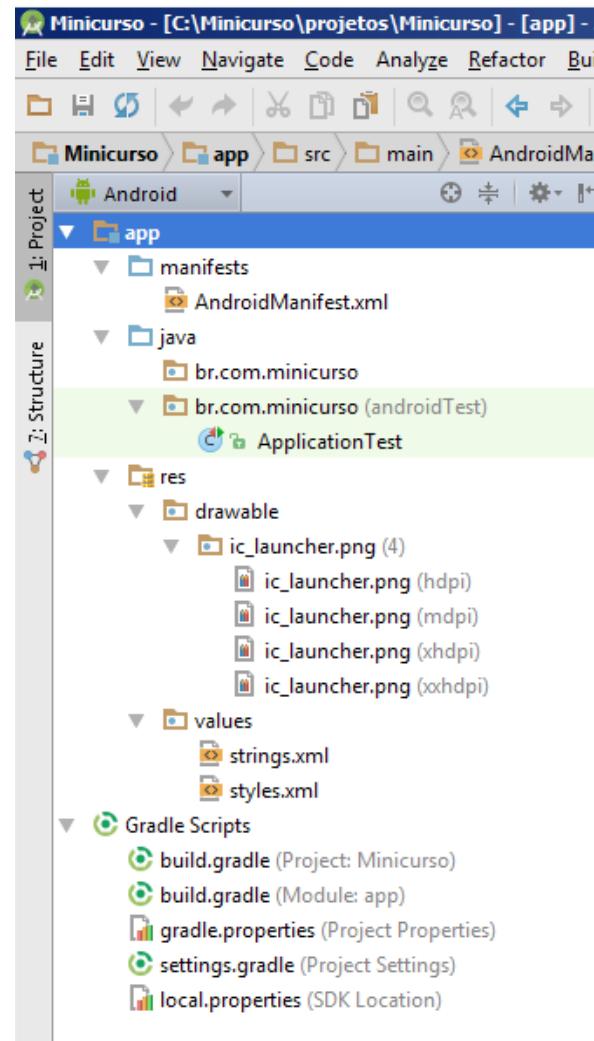


# Criar Projeto no Android Studio



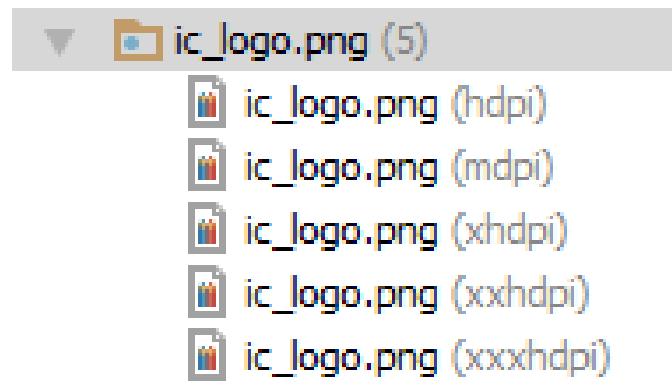
# Estrutura Básica de um Aplicativo

- **AndroidManifest.xml**
- **java**
  - br.com.minicurso
- **res**
  - drawable
  - values
- **build.gradle**



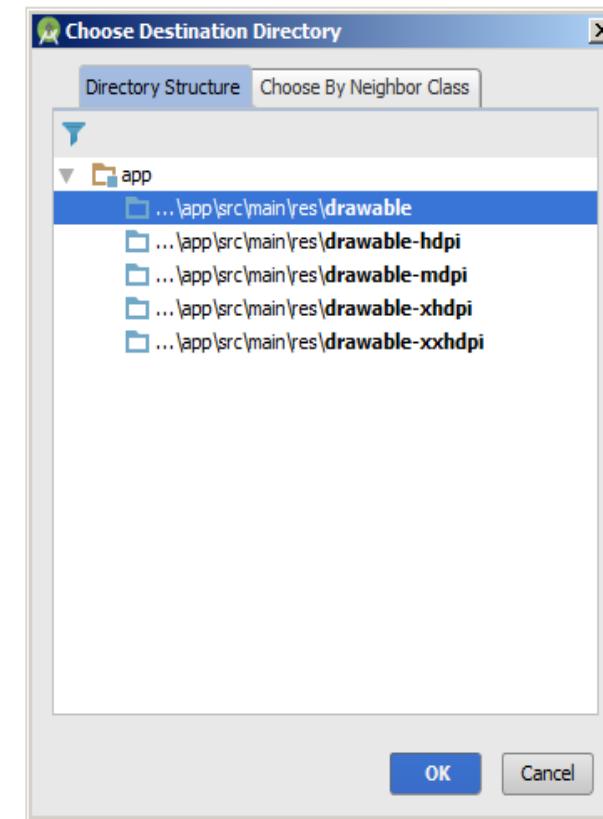
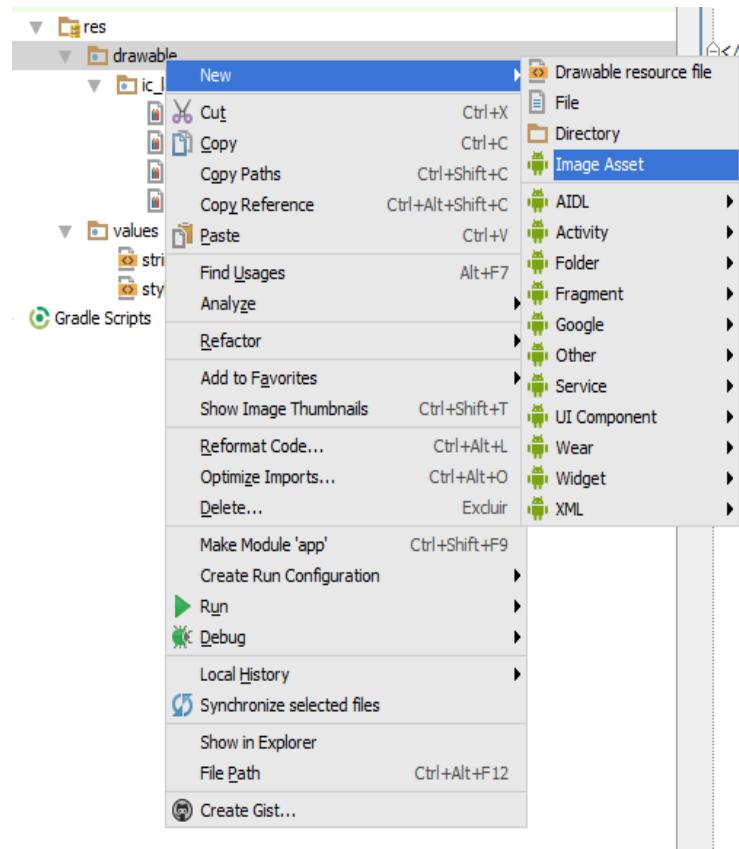
# Image Asset

- O android utiliza um conjunto de versões diferentes da mesma imagem.
- Cada versão tem uma resoluções diferente.
- Determinada imagem é utilizada dependendo do tamanho da tela.
- hdpi=72x72, mdpi=48x48, xhdpi=96x96, xxhdpi=144x144, xxhdpi=192x192



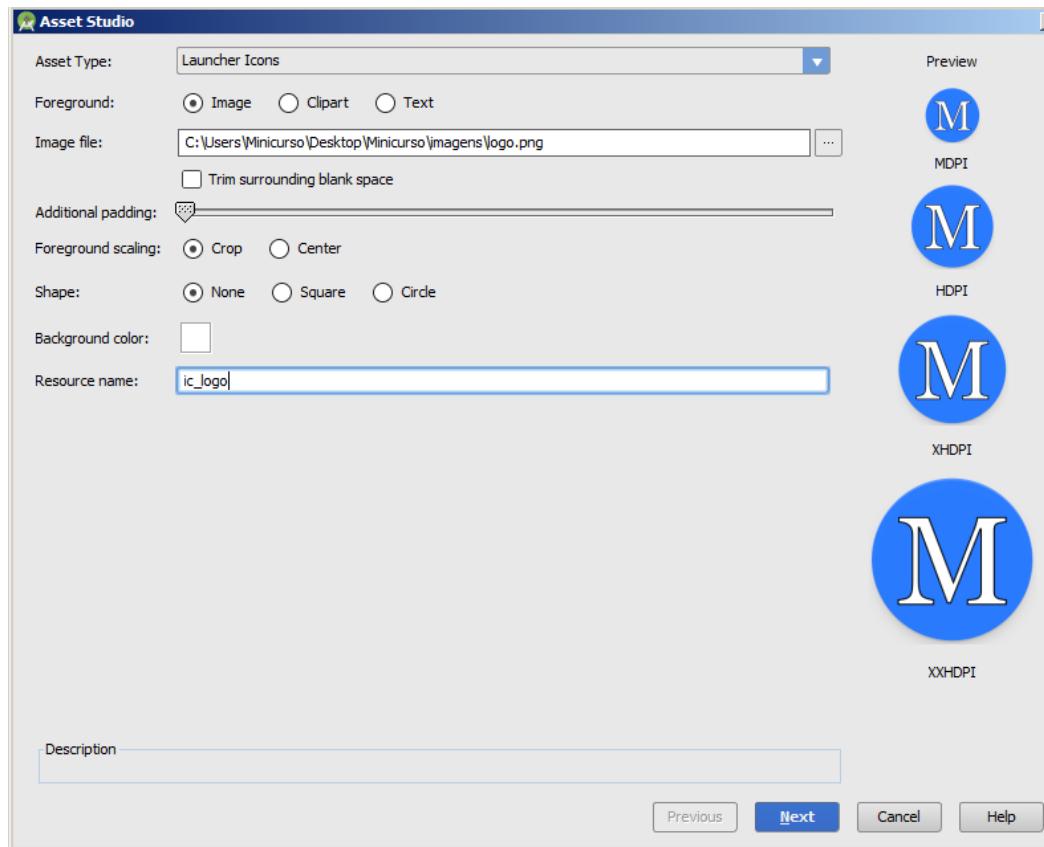
# Image Asset

- Para simplificar a criação desses conjuntos existe uma ferramenta de redimensionamento automático inclusa na IDE.

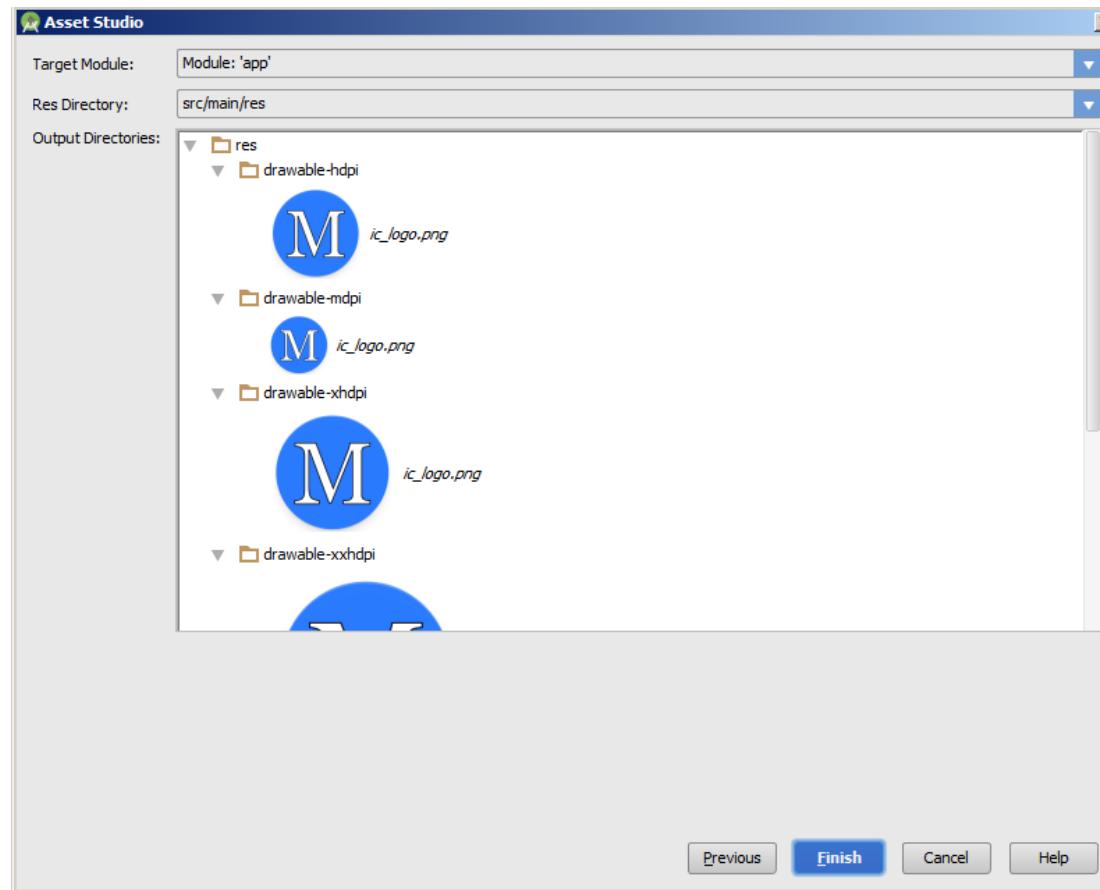


# Image Asset

- Para simplificar a criação desses conjuntos existe uma ferramenta de redimensionamento automático inclusa na IDE.

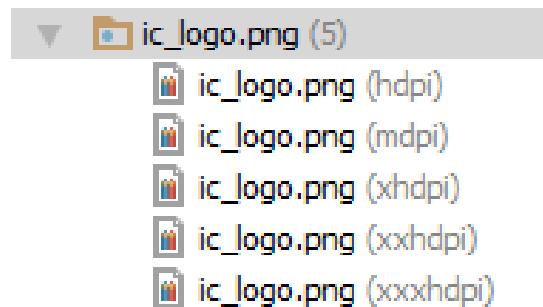


# Image Asset



# Image Asset

- Esse é o resultado dentro da ide:

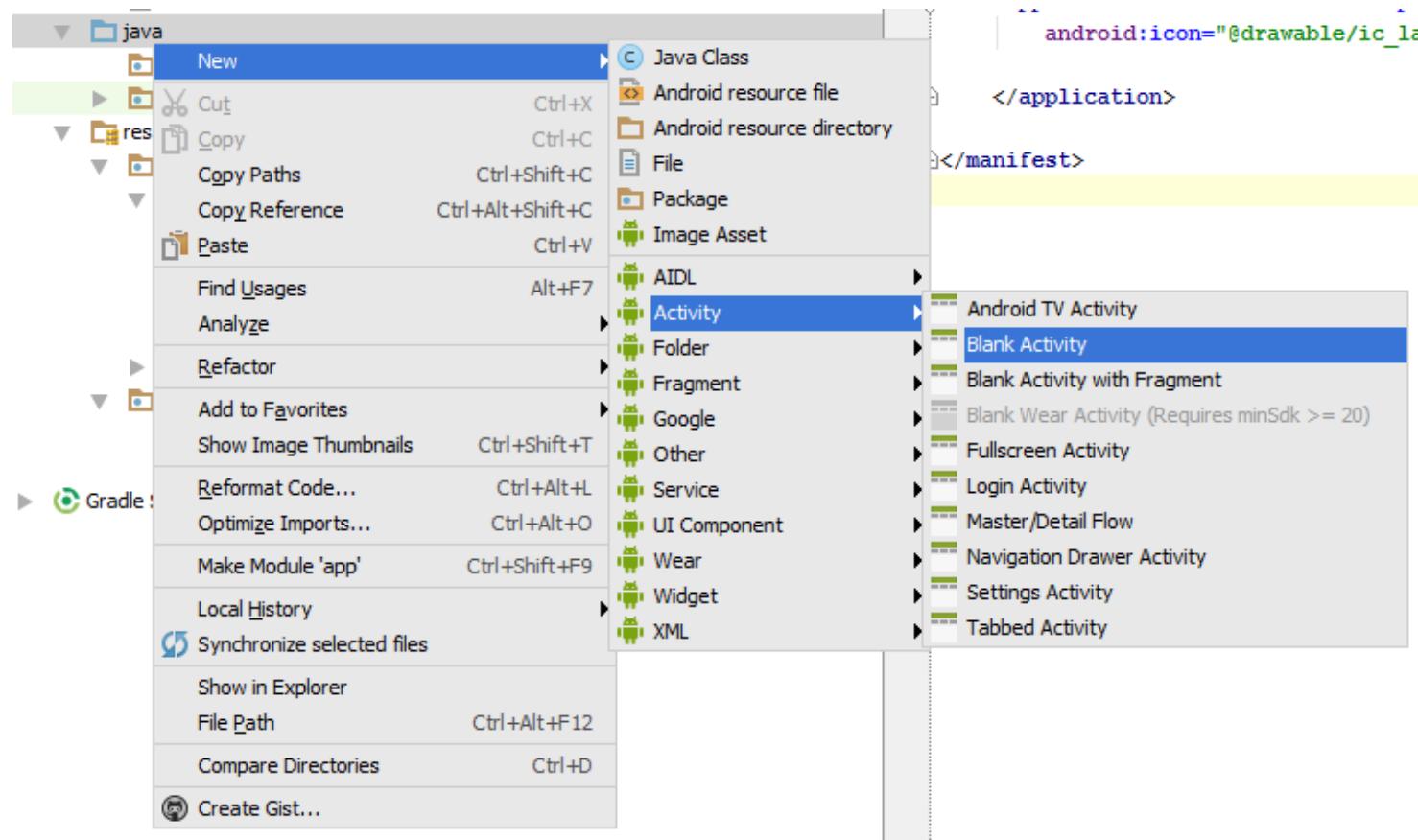


- No sistema de arquivos são criadas 5 pastas diferentes, cada uma com uma versão da imagem de tamanho diferente



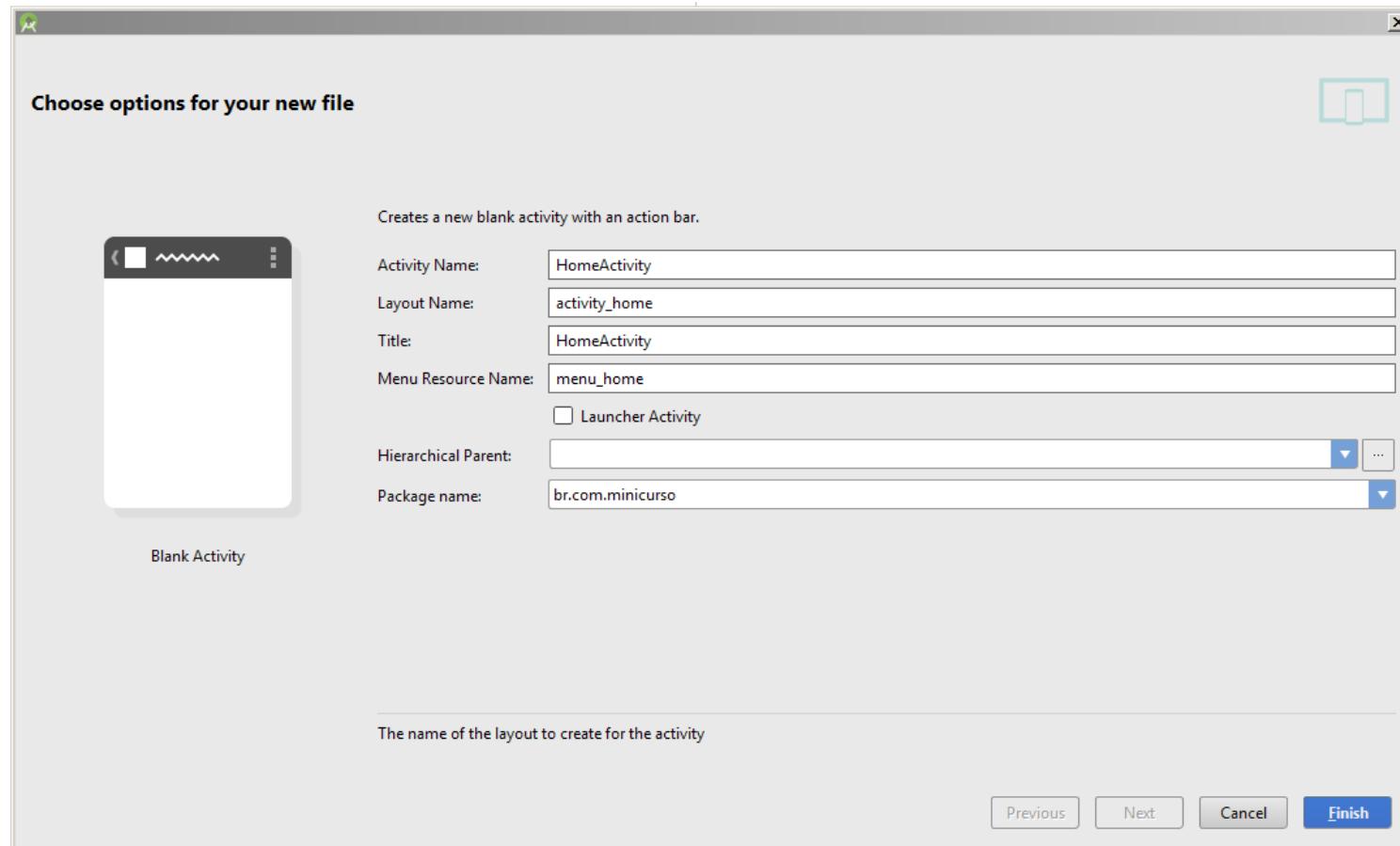
# Home Activity

- Vamos definir o nome da primeira activity como HomeActivity.



# Home Activity

- Vamos definir o nome da primeira activity como HomeActivity.



# Home Activity

- Exemplo de Activity:

```
package br.com.minicurso;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.MenuItem;

public class HomeActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_home, menu);
        return true;
    }
}
```



# Home Activity

- Exemplo de Activity

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

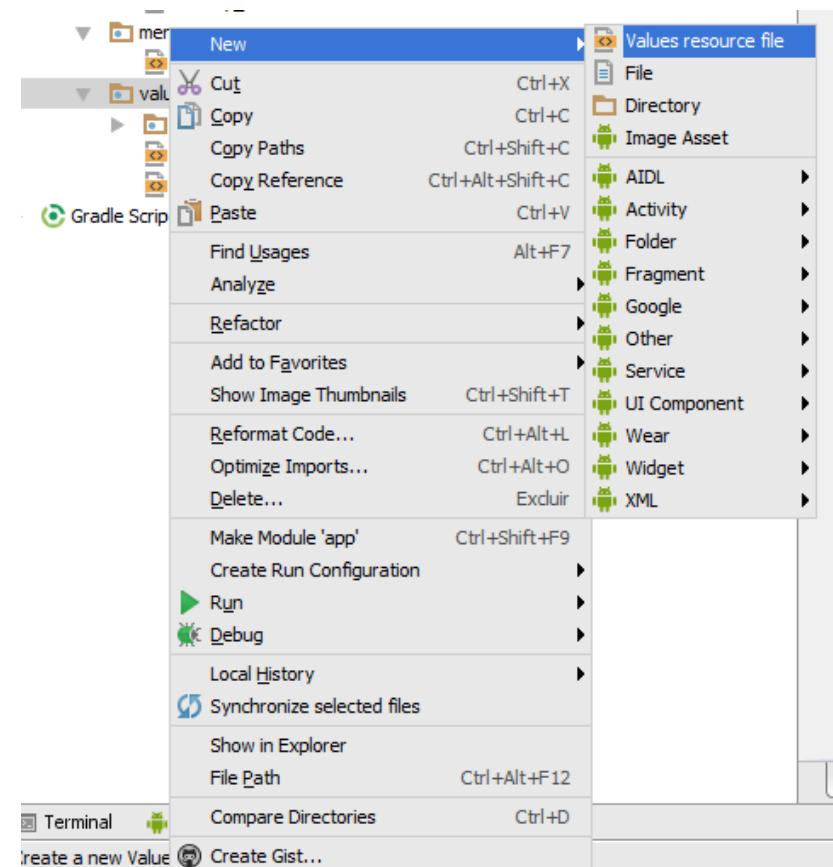
    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}
```



# Value Resource File

- Um resource file é um arquivo .xml utilizado pelo android.
- Deve estar obrigatoriamente dentro da pasta values.
- Crie um resource file chamado themes e um chamado colors.



# Value Resource File

New Resource File

File name: themes

Source set: main

Directory name: values

Available qualifiers:

- Country Code
- Network Code
- Language
- Region
- Layout Direction
- Smallest Screen Width
- Screen Width
- Screen Height
- Size
- Ratio
- Orientation
- UI Mode
- Night Mode
- Density

Chosen qualifiers:

>>

<<

Nothing to show

OK Cancel



# Value Resource File

- Cole o código a seguir arquivo colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <!-- Material Design Color -->

    <color name="Blue_500">#2196F3</color>
    <color name="Blue_900">#0D47A1</color>
    <color name="Blue_100">#BBDEFB</color>
    <color name="Roxo_500">#9C27B0</color>
    <color name="Roxo_800">#6A1B9A</color>
    <color name="Roxo_100">#E1BEE7</color>
    <color name="Green_500">#4CAF50</color>
    <color name="Green_900">#1B5E20</color>
    <color name="Green_100">#C8E6C9</color>
    <color name="Yellow_500">#FFEB3B</color>
    <color name="Yellow_A700">#FFD600</color>
    <color name="Yellow_100">#FFF9C4</color>
    <color name="Red_500">#F44336</color>
    <color name="Red_900">#B71C1C</color>
    <color name="Red_200">#EF9A9A</color>
    <color name="Black">#000000</color>
    <color name="White">#FFFFFF</color>
    <color name="Grey_500">#9E9E9E</color>
    <color name="Grey_300">#E0E0E0</color>
    <color name="Grey_800">#424242</color>
```



# Value Resource File

```
<color name="Brown_500">#795548</color>
<color name="Brown_200">#BCAAA4</color>
<color name="Brown_900">#3E2723</color>
<color name="Purple_500">#9C27B0</color>
<color name="Purple_100">#E1BEE7</color>
<color name="Purple_800">#6A1B9A</color>
<color name="PurpleA200">#E040FB</color>
<color name="Orange500">#FF9800</color>
<color name="Orange100">#FFE0B2</color>
<color name="Orange900">#E65100</color>
<color name="DeepOrange500">#FF5722</color>
<color name="DeepOrange100">#FFCCBC</color>
<color name="DeepOrange900">#BF360C</color>
<color name="PinkA200">#FF4081</color>
<color name="Transparent">#00000000</color>

</resources>
```



# Value Resource File

- Cole o código a seguir arquivo themes.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!--
    Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
-->

<resources xmlns:tools="http://schemas.android.com/tools">

    <style name="Theme.AppCompat.Light.NoActionBar.Minicurso"
parent="Theme.AppCompat.Light.NoActionBar">
        <item name="android:windowBackground">@color/White</item>
    </style>

</resources>
```



# AndroidManifest.xml

- Cole o código a seguir arquivo AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!--
    Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
-->
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="br.com.minicurso"
    android:versionName="0.1" >

    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="21" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_logo"
        android:label="@string/app_name" >
```



# AndroidManifest.xml

```
<activity
    android:name=".HomeActivity"
    android:icon="@mipmap/ic_logo"
    android:label="@string/app_name"

    android:theme="@style/Theme.AppCompat.Light.NoActionBar.Minicurso" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category
                android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

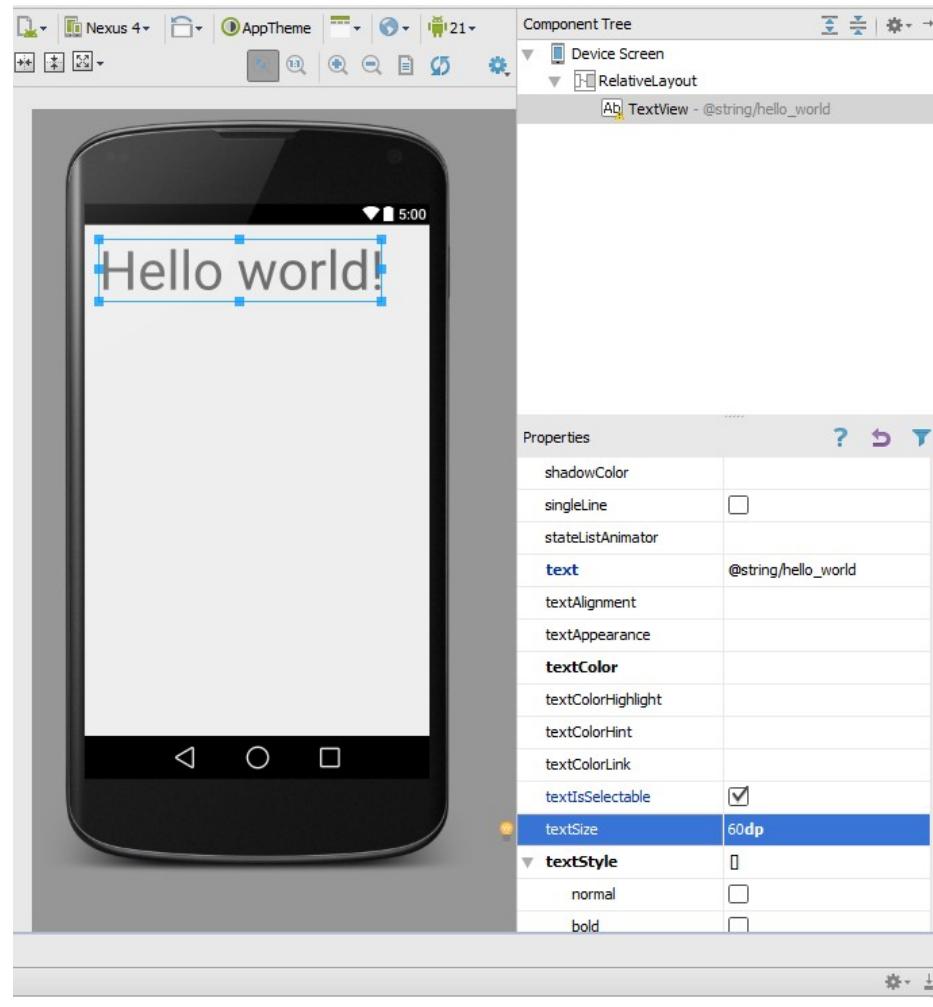
</application>

</manifest>
```



# Layouts (activity\_home)

- Conhecendo o modo design para criação de interfaces



# Layouts

- Resultado gerado:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context="br.com.minicurso.HomeActivity">

    <TextView android:text="@string/hello_world"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="60dp"
        android:textIsSelectable="true" />

</RelativeLayout>
```



# Virtual Device

- Para executar sua aplicação será necessário um dispositivo virtual ou dispositivo real conectado a porta USB do computador.
- Para esse exercício utilizaremos um dispositivo virtual.



# Virtual Device

AVD Manager

## Your Virtual Devices



Virtual devices allow you to test your application without having to own the physical devices.

Create a virtual device

To prioritize which devices to test your application on, visit the [Android Dashboards](#), where you can get up-to-date information on which devices are active in the Android and Google Play ecosystem.

OK Cancel



# Virtual Device

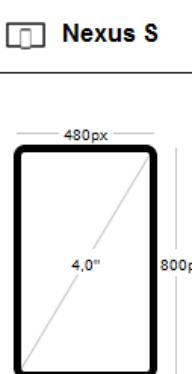
**Virtual Device Configuration**

### Select Hardware

Choose a device definition

Category	Name	Size	Resolution	Density
Phone	Nexus S	4,0"	480x800	hdpi
Tablet	Nexus One	3,7"	480x800	hdpi
Wear	Nexus 6	5,96"	1440x2560	560dpi
TV	Nexus 5	4,95"	1080x1920	xxhdpi
	Nexus 4	4,7"	768x1280	xhdpi
	Galaxy Nexus	4,65"	720x1280	xhdpi
	5,4" FWVGA	5,4"	480x854	mdpi
	5,1" WVGA	5,1"	480x800	mdpi
	4,7" WXGA	4,7"	720x1280	xhdpi

**Nexus S**



Size: normal  
Ratio: long  
Density: hdpi

New Hardware Profile Import Hardware Profiles 

Clone Device... Previous Next Cancel Finish



# Virtual Device

**Virtual Device Configuration**

**Android Virtual Device (AVD)**

Verify Configuration

AVD Name: Nexus S API 21

Nexus S: 4,0" 480x800 hdpi

Lollipop: Google APIs (Google Inc.) x86

Startup size and orientation: Scale: Auto

Emulated Performance:

- Use Host GPU
- Store a snapshot for faster startup

You can either use Host GPU or Snapshots

Show Advanced Settings

Use Host GPU

This enables the emulator graphics to run faster by using your computer's graphics card for OpenGL ES graphics rendering. (Recommended for better emulator experience)

Previous Next Cancel Finish

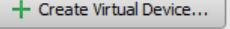


# Virtual Device

AVD Manager

Your Virtual Devices  
Android Studio

Type	Name	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Nexus S API 21	480 × 800: hdpi	21	Google APIs	x86	650 MB	

 Create Virtual Device... 

 OK  Cancel

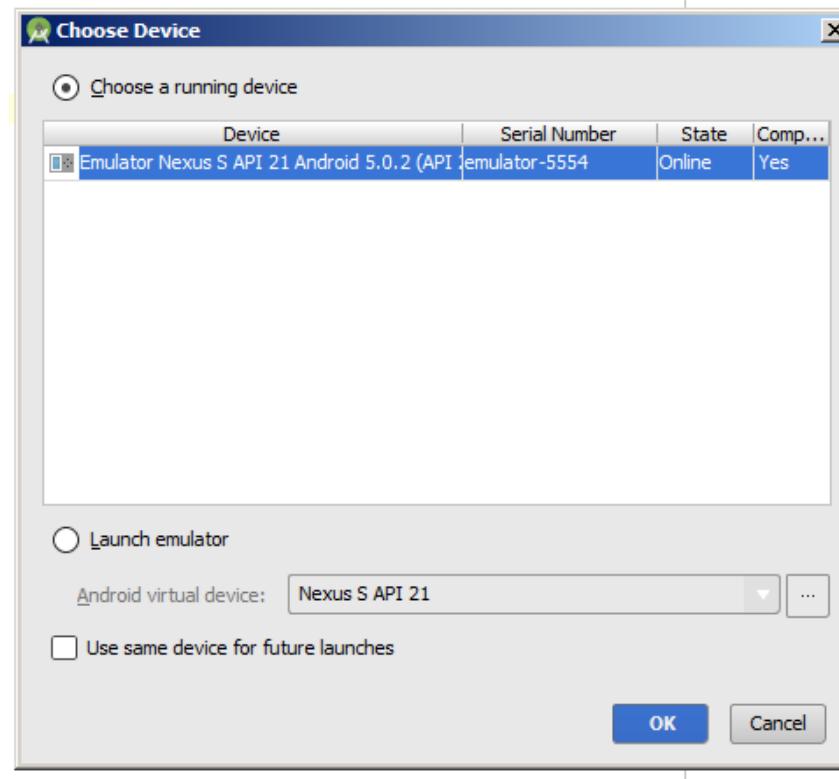


# Iniciar Virtual Device



# Executar Projeto

- Assim que o emulador iniciar, pode demorar um pouco
- Clique no botão executar 



# Aplicativo Hello World!



# Login Interno no Dispositivo



# Material Design



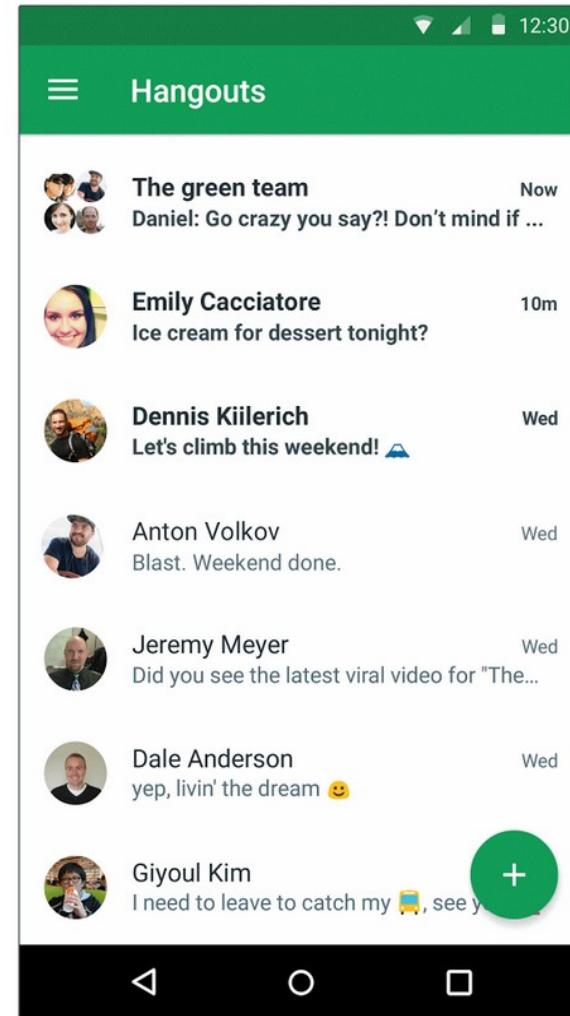
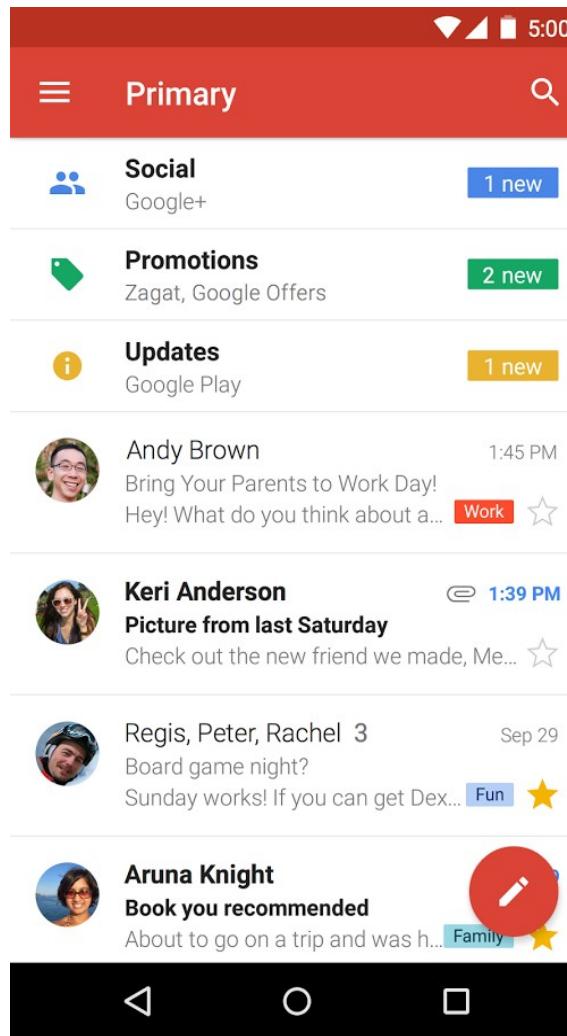
# Material Design

- Material Design é o novo estilo visual da Google.
- Interface oficial do Android Lollipop (API 21) e seus aplicativos.
- Pode ser utilizado nas versões anteriores através das bibliotecas de suporte.
- promete unificar o visual do Android com o dos aplicativos do Chrome OS e com as páginas de internet.

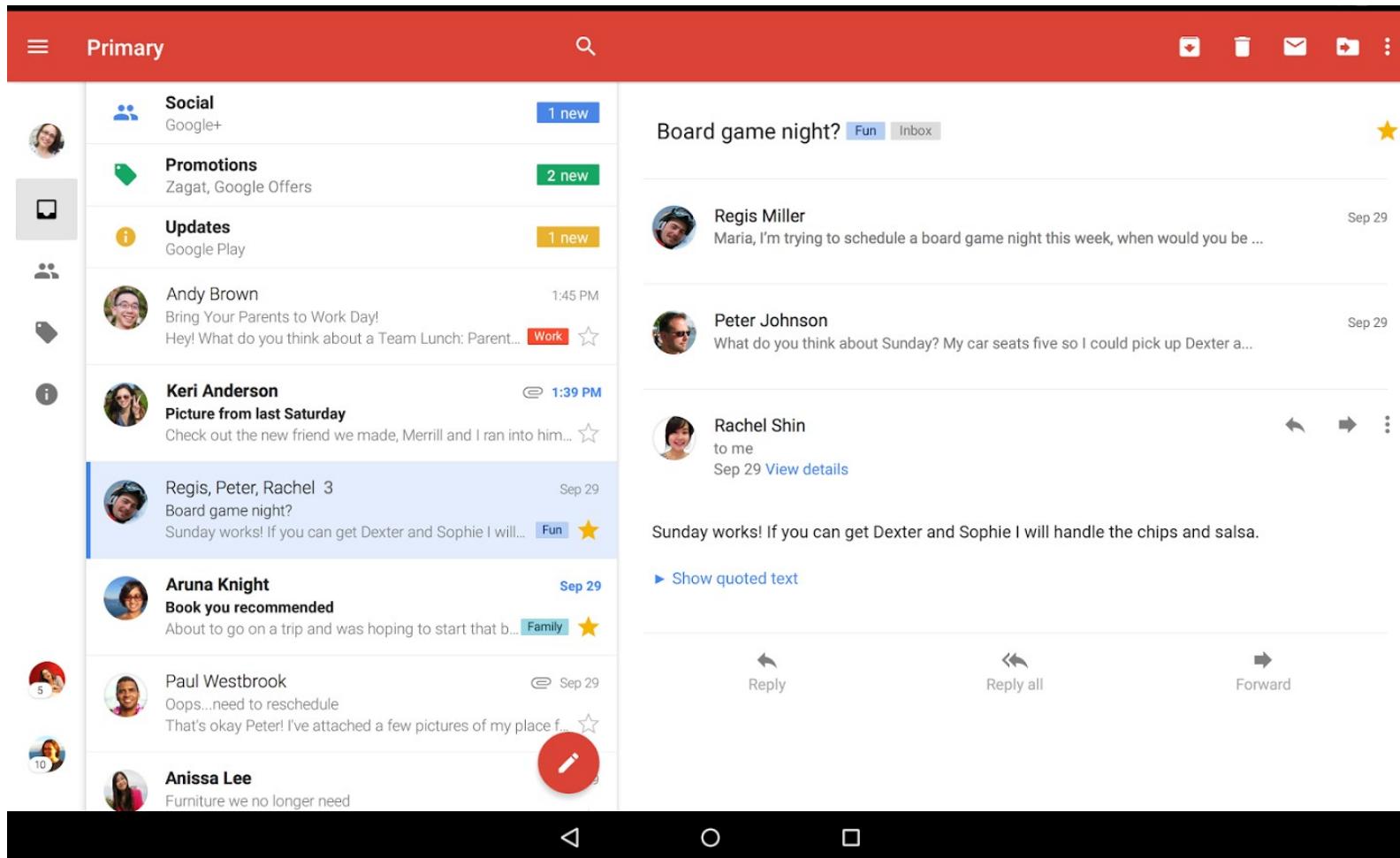


# Material Design

- Gmail
- Hangouts



# Material Design



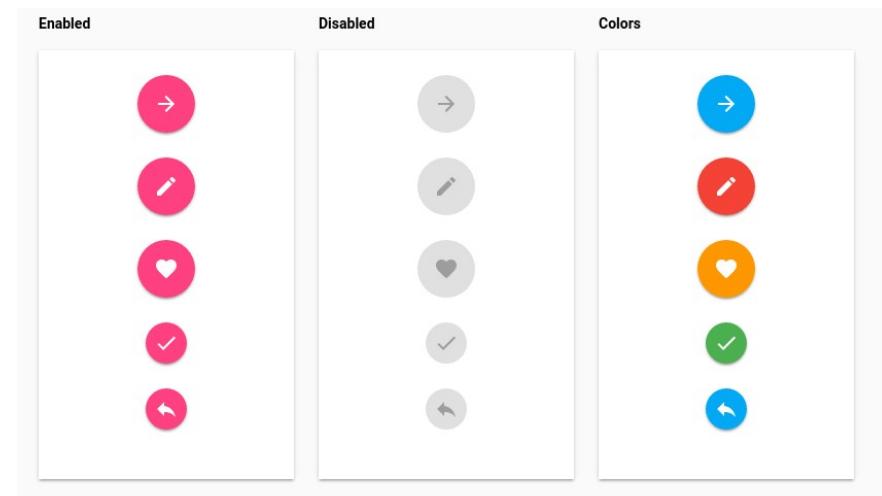
The screenshot shows the Gmail app interface on a smartphone. The top navigation bar is red, featuring the word "Primary" and a search icon. Below the navigation bar is a sidebar with various icons: Social (Google+), Promotions (Zagat, Google Offers), Updates (Google Play), Andy Brown (Bring Your Parents to Work Day), Keri Anderson (Picture from last Saturday), Regis, Peter, Rachel (Board game night), Aruna Knight (Book you recommended), Paul Westbrook (Oops...need to reschedule), and Anissa Lee (Furniture we no longer need). The main area displays an email thread from Regis Miller. The subject line is "Board game night?". The message body reads: "Maria, I'm trying to schedule a board game night this week, when would you be ...". The timestamp is Sep 29. Below this is another message from Peter Johnson: "What do you think about Sunday? My car seats five so I could pick up Dexter a...". The timestamp is also Sep 29. A reply from Rachel Shin is shown: "Sunday works! If you can get Dexter and Sophie I will handle the chips and salsa." The timestamp is Sep 29. At the bottom of the screen are standard Android navigation buttons: back, home, and recent apps.



# Polymer

- O Projeto Polymer é um Conjunto de Web Componentes que seguem o padrão Material Design.

```
<paper-fab icon="add"></paper-fab>  
  
<paper-fab mini  
icon="favorite"></paper-fab>  
  
<paper-fab src="star.png"></paper-  
fab>
```



# Ícones

- Utilizaremos os ícones que seguem o padrão material design.



- A lista completa está disponível em:
  - <https://www.google.com/design/icons/>



# Ícones

- Esses ícones não fazem parte das bibliotecas das versões anteriores a API 21.
- Devem ser inclusos no projeto para manter compatibilidade com versões anteriores do android.
- Alguns ícones já foram preparados para serem adicionados a esse projeto.



# Ícones

- Copie e cole essas pastas no diretórios de resources do projeto.
- app/src/main/res/



mipmap-hdpi



mipmap-mdpi



mipmap-xhdpi



mipmap-xxhdpi



mipmap-xxxhdpi

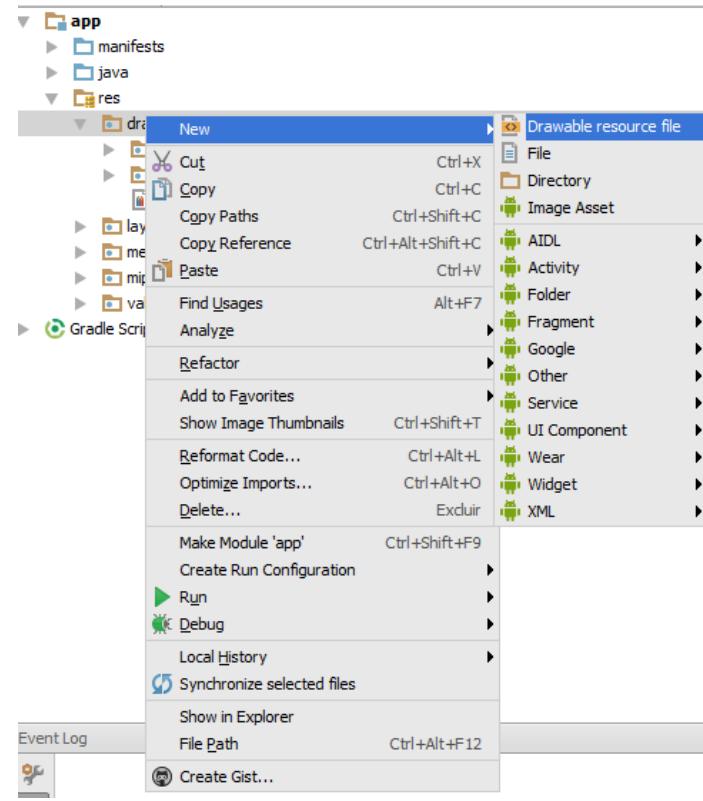
▼  mipmap

-  .directory (2)
-  ic\_account\_balance\_blue.png (5)
-  ic\_account\_balance\_white.png (5)
-  ic\_account\_circle\_white.png (5)
-  ic\_add\_black.png (5)
-  ic\_add\_cirde\_black.png (5)
-  ic\_add\_cirde\_white.png (5)
-  ic\_add\_shopping\_cart\_black.png (5)
-  ic\_add\_shopping\_cart\_white.png (5)
-  ic\_add\_white.png (5)
-  ic\_assessment\_black.png (5)
-  ic\_assessment\_colors.png (5)
-  ic\_assessment\_white.png (5)
-  ic\_assignment\_black.png (5)
-  ic\_assignment\_ind\_black.png (5)
-  ic\_assignment\_ind\_white.png (5)
-  ic\_assignment\_white.png (5)
-  ic\_attach\_money\_black.png (5)
-  ic\_attach\_money\_green.png (5)
-  ic\_attach\_money\_red.png (5)
-  ic\_dear\_black.png (5)
-  ic\_dear\_white.png (5)
-  ic\_credit\_card\_black.png (5)
-  ic\_credit\_card\_blue.png (5)

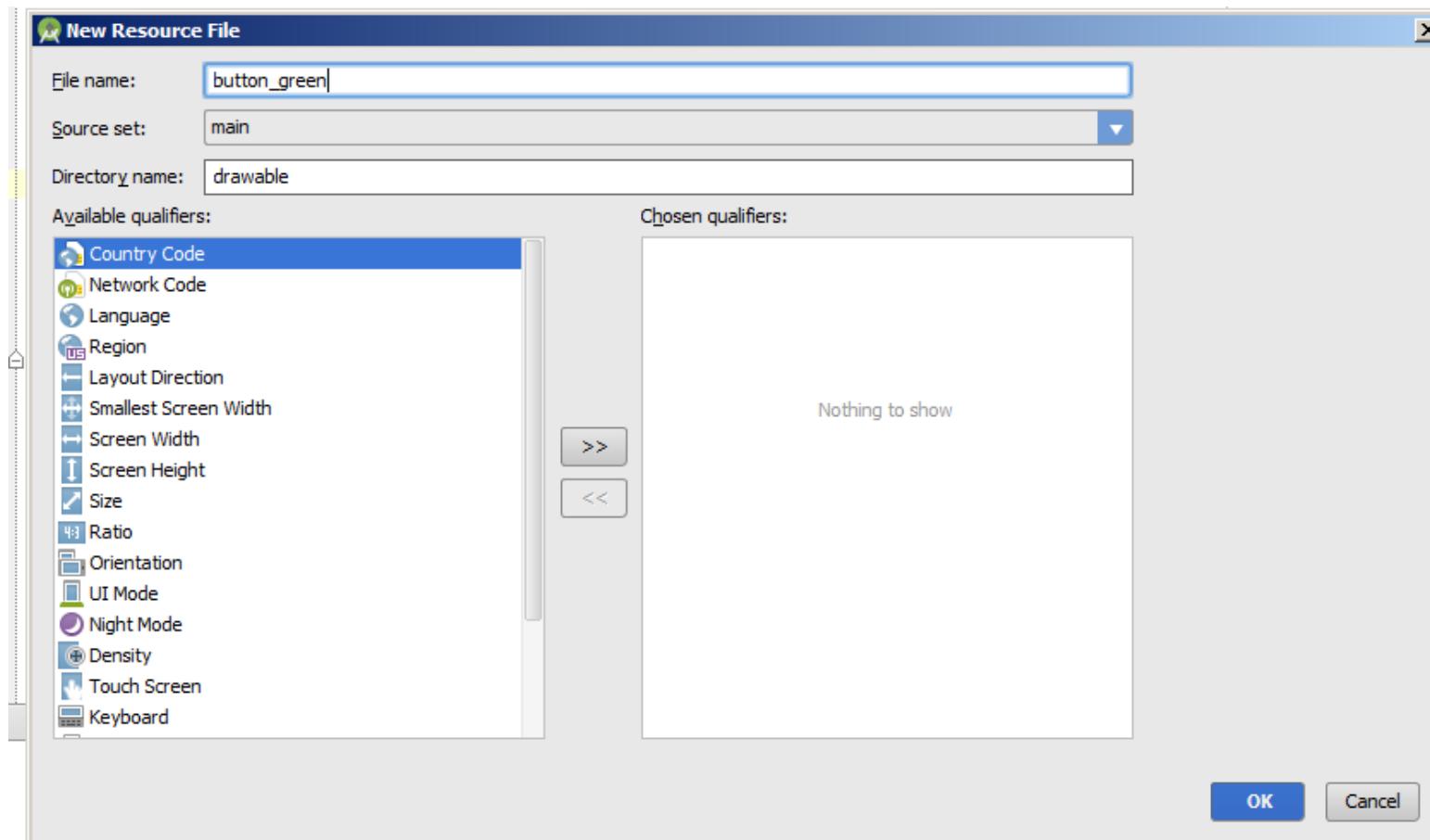


# Botões

- Crie um resource chamado button\_green e um chamdo button\_blue



# Botões



# Botões

- Conteúdo de button\_green.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!--
    Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
-->
<selector xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:drawable="@color/Green_500"
        android:state_activated="false"
        android:state_pressed="false"/>

    <item
        android:drawable="@color/Green_900"
        android:state_pressed="true"/>

    <item
        android:drawable="@color/Green_900"
        android:state_activated="true"/>

</selector>
```

 ENTRAR ENTRAR

# Botões

- Conteúdo de button\_blue.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!--
    Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
-->
<selector xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:drawable="@color/Blue_500"
        android:state_activated="false"
        android:state_pressed="false"/>

    <item
        android:drawable="@color/Blue_900"
        android:state_pressed="true"/>

    <item
        android:drawable="@color/Blue_900"
        android:state_activated="true"/>

</selector>
```

ESQUECEU SUA SENHA?

ESQUECEU SUA SENHA?



# Imagen

- Copie a imagem minicurso.png da pasta imagens para app/src/main/res/drawable

**Minicurso**   
*Desenvolvimento Mobile*



# Strings

- Adicione as seguintes strings

```
<string name="login">Entrar</string>
<string name="informationAccount">Informações da Conta</string>
<string name="email">Email</string>
<string name="password">Senha</string>
<string name="forgoutYourPassword">Esqueceu Sua senha?</string>
<string name="signup">Criar Conta</string>
```



# Ferramentas de Build

- No mundo Java as ferramentas de build (gerenciamento de dependências) mais conhecidas e usadas são o Maven e o Ant



**maven**

- O Gradle é a ferramenta de build padrão para desenvolvimento Android.



# Ferramentas de Build

- Exemplo de dependência pelo gradle:

```
dependencies
```

```
{  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    compile 'com.android.support:appcompat-v7:23.0.1'  
}
```

- Exemplo de dependência pelo maven:

```
<dependency>  
    <groupId>org.primefaces</groupId>  
    <artifactId>primefaces</artifactId>  
    <version>5.2</version>  
</dependency>
```



# Gradle

- Adicione a dependência ao gradle:

```
dependencies
```

```
{  
    compile 'com.android.support:design:23.0.1'  
}
```



# Temas

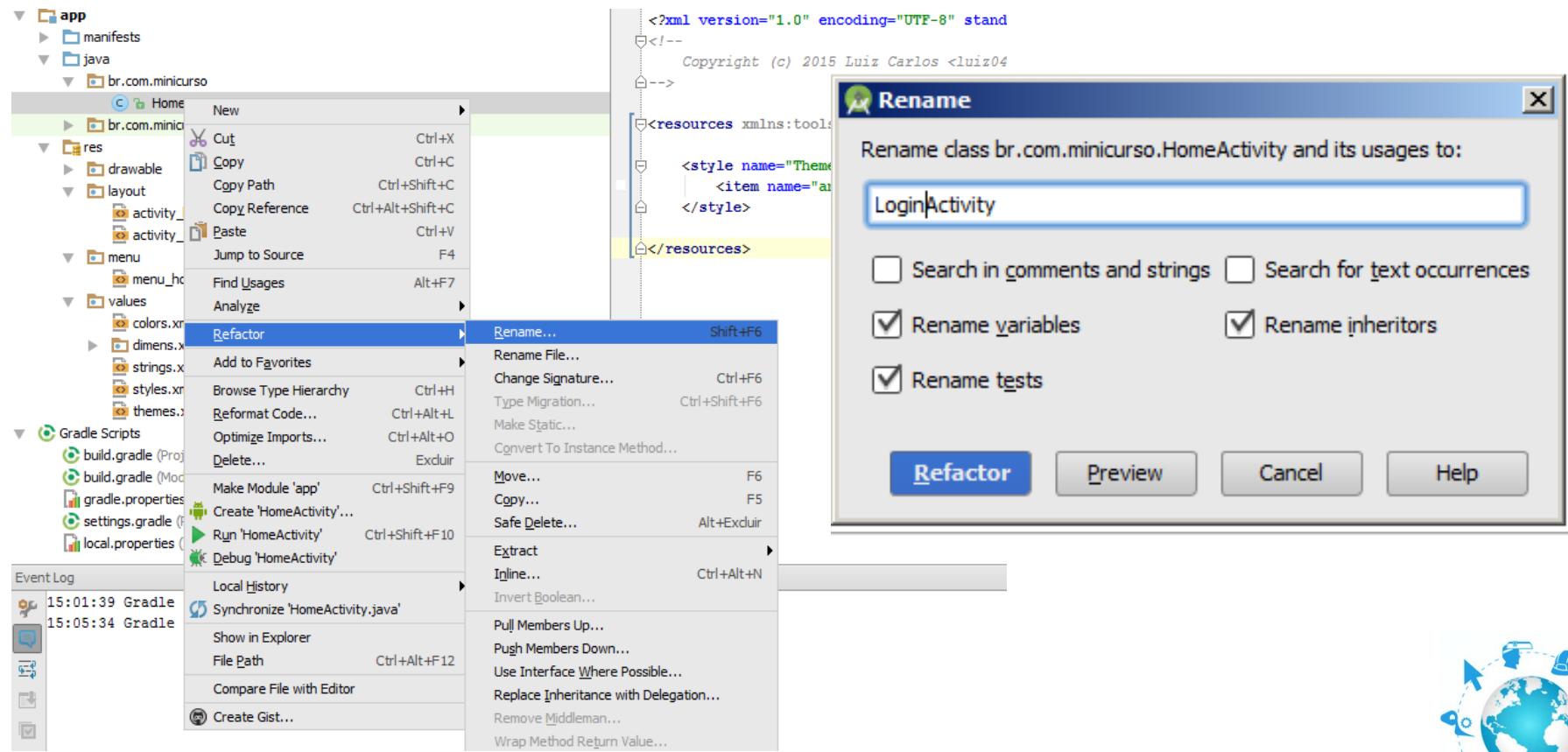
- Adicione o seguinte estilo no arquivo themes.xml:

```
<style name="Theme.AppCompat.Light.NoActionBar.Minicurso.Blue"
parent="Theme.AppCompat.Light.NoActionBar.Minicurso">
    <item name="colorPrimary">@color/Blue_500</item>
    <item name="colorPrimaryDark">@color/Blue_900</item>
    <item name="colorAccent">@color/Blue_500</item>
    <item name="android:windowBackground">@color/Blue_900</item>
</style>
```



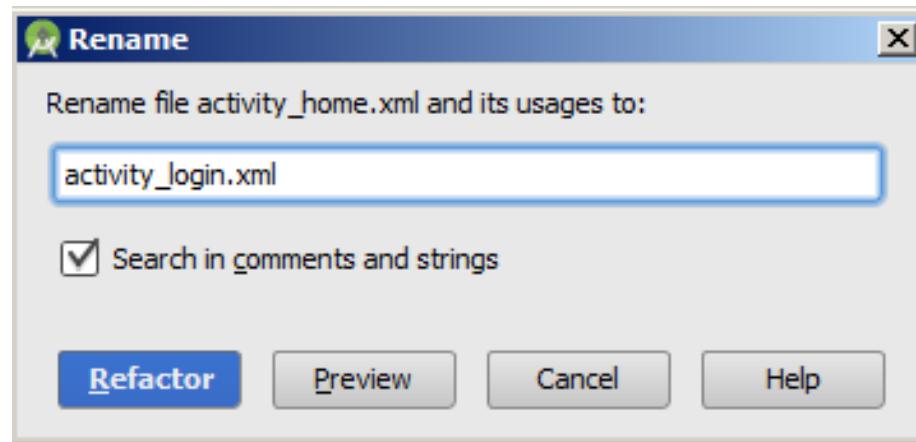
# LoginActivity

- Renomeie a classe HomeActivity para LoginActivity



# Layout

- Renomeie o arquivo activity\_home.xml para activity\_login.xml



# Temas

- Altere o tema no arquivo AndroidManifest.xml:

De:

```
<activity
    android:name=".LoginActivity"
    android:theme="@style/Theme.AppCompat.Light.NoActionBar.Minicurs
o" >
</activity>
```

Para:

```
<activity
    android:name=".LoginActivity"
    android:theme="@style/Theme.AppCompat.Light.NoActionBar.Minicurs
o.Blue" >
</activity>
```



# Estilos

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!--
    Copyright (c) 2015 Luiz Carlos <lui04nl@gmail.com>
-->

<resources>
    <style name="Main">
        <item name="android:background">@color/White</item>
        <item name="android:layout_width">fill_parent</item>
        <item name="android:layout_height">fill_parent</item>
        <item name="android:orientation">vertical</item>
        <item name="android:gravity">center_horizontal|center_vertical</item>
    </style>

    <style name="Background">
        <item name="android:layout_height">wrap_content</item>
        <item name="android:layout_width">fill_parent</item>
        <item name="android:orientation">vertical</item>
    </style>
```



# Estilos

```
<style name="Background.Land" parent="Background">
    <item name="android:gravity">center_vertical|center_horizontal|center</item>
    <item name="android:orientation">horizontal</item>
</style>

<style name="Logo">
    <item name="android:layout_marginTop">10dp</item>
    <item name="android:layout_marginBottom">0dp</item>
    <item name="android:layout_gravity">center</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:layout_width">wrap_content</item>
    <item name="android:orientation">horizontal</item>
    <item name="android:weightSum">1</item>
</style>

<style name="Logo.Land" parent="Logo">
    <item name="android:layout_marginLeft">10dp</item>
    <item name="android:layout_marginRight">20dp</item>
    <item name="android:orientation">vertical</item>
</style>

<style name="Img">
    <item name="android:layout_width">wrap_content</item>
    <item name="android:layout_height">match_parent</item>
    <item name="android:gravity">center</item>
    <item name="android:layout_gravity">center_vertical</item>
</style>
```



# Estilos

```
<style name="Img.Minicurso" parent="Img">
    <item name="android:layout_alignParentLeft">true</item>
    <item name="android:layout_centerVertical">true</item>
    <item name="android:layout_width">300dp</item>
    <item name="android:layout_height">84dp</item>
    <item name="android:src">@drawable/minicurso</item>
</style>

<style name="Img.Minicurso.Land" parent="Img.Minicurso">
    <item name="android:layout_width">250dp</item>
    <item name="android:layout_height">70dp</item>
</style>

<style name="Context">
    <item name="android:layout_height">wrap_content</item>
    <item name="android:layout_width">fill_parent</item>
    <item name="android:orientation">vertical</item>
    <item name="android:layout_weight">1</item>
</style>

<style name="Area">
    <item name="android:gravity">center</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:layout_width">250dp</item>
    <item name="android:orientation">horizontal</item>
    <item name="android:textSize">15dp</item>
</style>
```



# Estilos

```
<style name="Area.Relative" parent="Area">
    <item name="android:gravity">left</item>
    <item name="android:inputType">text</item>
    <item name="android:layout_height">70dp</item>
    <item name="android:layout_marginLeft">10dp</item>
    <item name="android:layout_marginRight">10dp</item>
    <item name="android:layout_width">fill_parent</item>
</style>

<style name="Area.Relative.Center" parent="Area.Relative">
    <item name="android:layout_height">wrap_content</item>
    <item name="android:layout_gravity">center_horizontal</item>
    <item name="android:gravity">center</item>
</style>

<style name="TextInputLayout">
    <item name="android:layout_width">fill_parent</item>
    <item name="android:layout_height">60dp</item>
</style>

<style name="Text">
    <item name="android:gravity">center</item>
    <item name="android:layout_gravity">center_vertical|center_horizontal</item>
    <item name="android:layout_height">match_parent</item>
    <item name="android:layout_width">wrap_content</item>
    <item name="android:textSize">16dp</item>
    <item name="android:textColor">@color/Black</item>
</style>
```



# Estilos

```
<style name="Text.SubTitle" parent="Text">
    <item name="android:layout_marginBottom">10dp</item>
    <item name="android:textSize">16dp</item>
</style>

<style name="Text.SubTitle.InformationAccount" parent="Text.SubTitle">
    <item name="android:text">@string/informationAccount</item>
</style>

<style name="Area.Relative.EditArea" parent="Area.Relative">
    <item name="android:layout_height">60dp</item>
    <item name="android:layout_marginBottom">10dp</item>
</style>

<style name="Img.Email" parent="Img">
    <item name="android:layout_marginTop">10dp</item>
    <item name="android:src">@mipmap/ic_email_blue</item>
</style>

<style name="Area.Relative.Edit" parent="Area.Relative">
    <item name="android:layout_marginLeft">60dp</item>
</style>

<style name="Area.Relative.Edit.Email" parent="Area.Relative.Edit">
    <item name="android:hint">@string/email</item>
    <item name="android:inputType">textEmailAddress</item>
    <item name="android:layout_width">fill_parent</item>
    <item name="android:layout_height">50dp</item>
</style>
```



# Estilos

```
<style name="Img.Password" parent="Img">
    <item name="android:layout_marginTop">10dp</item>
    <item name="android:src">@mipmap/ic_https_yellow</item>
</style>

<style name="Area.Relative.Edit.Pasword" parent="Area.Relative.Edit">
    <item name="android:hint">@string/password</item>
    <item name="android:inputType">textPassword</item>
    <item name="android:layout_width">fill_parent</item>
    <item name="android:layout_height">50dp</item>
</style>

<style name="Shadow">
    <item
name="android:background">@drawable/abc_popup_background_mtrl_mult</item>
    <item name="android:layout_gravity">center</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:layout_width">wrap_content</item>
    <item name="android:orientation">vertical</item>
</style>

<style name="ButtonSignInSmall">
    <item name="android:padding">5dp</item>
    <item name="android:background">@drawable/button_green</item>
    <item name="android:layout_height">match_parent</item>
    <item name="android:layout_width">220dp</item>
    <item name="android:text">@string/login</item>
    <item name="android:textColor">@color/White</item>
</style>
```



# Estilos

```
<style name="ButtonSignUp" parent="ButtonSignInSmall">
    <item name="android:text">@string/signup</item>
</style>

<style name="ButtonForgotYourpassword" parent="ButtonSignInSmall">
    <item name="android:background">@drawable/button_blue</item>
    <item name="android:text">@string/forgoutYourPassword</item>
</style>

</resources>
```



# Layout

- Substitua o conteúdo do arquivo activity\_login.xml por:

```
<?xml version="1.0" encoding="utf-8"?>
<!--
    Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
-->

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    style="@style/Main">
    <LinearLayout
        style="@style/Background">
        <LinearLayout
            style="@style/Logo">
            <ImageView
                style="@style/Img.Minicurso"/>
        </LinearLayout>
        <LinearLayout
            style="@style/Context">
            <LinearLayout
                style="@style/Area.Relative.Center">
                <TextView
                    style="@style/Text.SubTitle.InformationAccount"/>
            </LinearLayout>
        </LinearLayout>
    </LinearLayout>
</LinearLayout>
```



# Layout

```
<RelativeLayout
    android:id="@+id/email"
    style="@style/Area.Relative.EditArea">
    <android.support.design.widget.TextInputLayout
        android:id="@+id/text_input_layout_email"
        android:layout_alignLeft="@+id/imageEmail"
        style="@style/TextInputLayout">
        <AutoCompleteTextView
            android:id="@+id/editEmail"
            style="@style/Area.Relative.Edit.Email"/>
    </android.support.design.widget.TextInputLayout>
    <ImageView
        style="@style/Img.Email"
        android:id="@+id/imageEmail" />
</RelativeLayout>
```



# Layout

```
<RelativeLayout
    android:id="@+id/password"
    style="@style/Area.Relative.EditArea">
    <android.support.design.widget.TextInputLayout
        android:id="@+id/text_input_layout_password"
        android:layout_alignLeft="@+id/imagePassword"
        style="@style/TextInputLayout">
        <AutoCompleteTextView
            android:id="@+id/editPassword"
            style="@style/Area.Relative.Edit.Pasword"/>
    </android.support.design.widget.TextInputLayout>
    <ImageView
        style="@style/Img.Password"
        android:id="@+id/imagePassword" />
</RelativeLayout>
```



# Layout

```
<LinearLayout
    style="@style/Shadow">
    <Button
        style="@style/ButtonSignInSmall"
        android:id="@+id/btn_signIn_small" />
</LinearLayout>
<LinearLayout
    style="@style/Shadow">
    <Button
        style="@style/ButtonForgotYourpassword"
        android:id="@+id/btn_forgotYourpassword" />
</LinearLayout>
<LinearLayout
    style="@style/Shadow">
    <Button
        style="@style/ButtonSignUp"
        android:id="@+id/btn_signUp" />
</LinearLayout>
</LinearLayout>
</LinearLayout>
```



# Resultado

Visualização Portrait

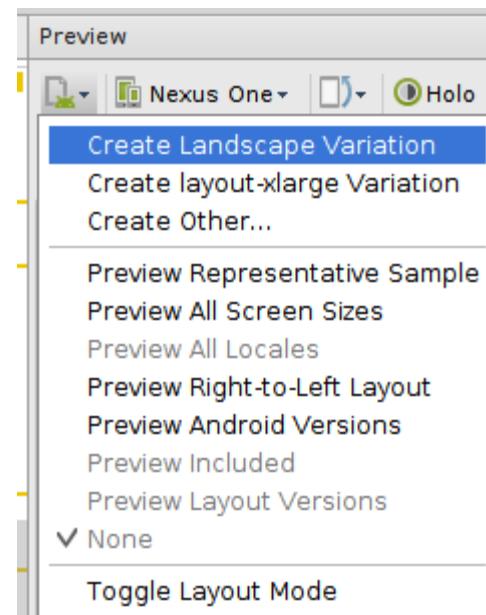


Visualização Landscap



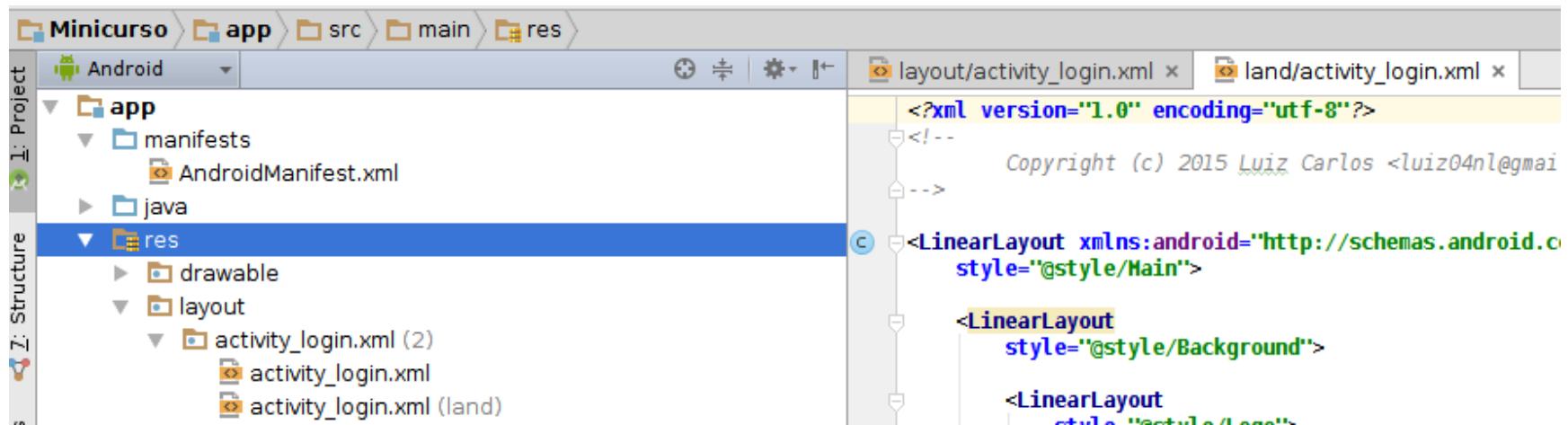
# Landscape Variation

- Para resolver esse problema criaremos uma variação do layout para telas landscape (deitadas).



# Landscape Variation

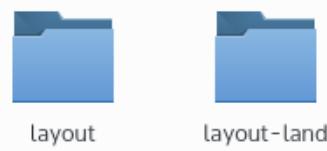
- Sera criado um nova arquivo de layout igual ao anterior, mas que devera ser alterado:



The screenshot shows the Android Studio interface. The Project tool window on the left displays the project structure under the 'app' module. The 'res' folder contains 'drawable' and 'layout' folders. Inside 'layout', there are two files: 'activity\_login.xml' (2) and 'activity\_login.xml (land)'. The right-hand editor pane shows the XML code for 'activity\_login.xml (land)'. The code includes standard XML declarations and three nested LinearLayout elements, each with specific style attributes.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- 
-->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    style="@style/Main">
    <LinearLayout
        style="@style/Background">
        <LinearLayout
            style="@style/Background">
    
```

- No sistemas de arquivos teremos a seguinte situação:



# Landscape Variation

- Substitua o conteúdo do arquivo activity\_login.xml (land) por:

```
<?xml version="1.0" encoding="utf-8"?>
<!--
    Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
-->

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    style="@style/Main">

    <LinearLayout
        style="@style/Background.Land">
```



# Landscape Variation

```
<LinearLayout
    style="@style/Logo.Land">

    <ImageView
        style="@style/Img.Minicurso.Land"/>
    <LinearLayout
        style="@style/Shadow">
        <Button
            style="@style/ButtonForgotYourpassword"
            android:id="@+id/btn_forgotYourpassword" />
    </LinearLayout>
    <LinearLayout
        style="@style/Shadow">
        <Button
            style="@style/ButtonSignUp"
            android:id="@+id/btn_signUp" />
    </LinearLayout>

</LinearLayout>
```



# Landscape Variation

```
<LinearLayout
    style="@style/Context">
    <LinearLayout
        style="@style/Area.Relative.Center">
        <TextView
            style="@style/Text.SubTitle.InformationAccount"/>
    </LinearLayout>
    <RelativeLayout
        android:id="@+id/email"
        style="@style/Area.Relative.EditArea">
        <android.support.design.widget.TextInputLayout
            android:id="@+id/text_input_layout_email"
            android:layout_alignLeft="@+id/imageEmail"
            style="@style/TextInputLayout">
            <AutoCompleteTextView
                android:id="@+id/editEmail"
                style="@style/Area.Relative.Edit.Email"/>
        </android.support.design.widget.TextInputLayout>
        <ImageView
            style="@style/Img.Email"
            android:id="@+id/imageEmail" />
    </RelativeLayout>
```



# Landscape Variation

```
<RelativeLayout
    android:id="@+id/password"
    style="@style/Area.Relative.EditArea">
    <android.support.design.widget.TextInputLayout
        android:id="@+id/text_input_layout_password"
        android:layout_alignLeft="@+id/imagePassword"
        style="@style/TextInputLayout">
        <AutoCompleteTextView
            android:id="@+id/editPassword"
            style="@style/Area.Relative.Edit.Pasword"/>
    </android.support.design.widget.TextInputLayout>
    <ImageView
        style="@style/Img.Password"
        android:id="@+id/imagePassword" />
</RelativeLayout>
<LinearLayout
    style="@style/Shadow">
    <Button
        style="@style/ButtonSignInSmall"
        android:id="@+id/btn_signIn_small" />
</LinearLayout>
</LinearLayout>
</LinearLayout>
</LinearLayout>
```

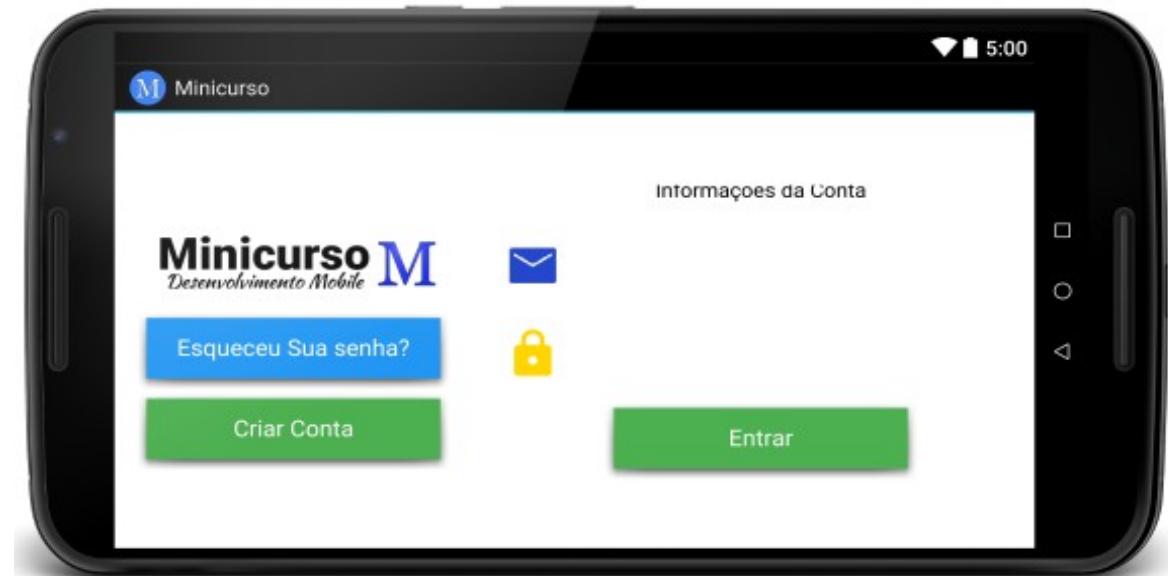


# Resultado Final

Visualização Portrait:



Visualização Landscap:



# Validação de Campos

- Modificaremos a atividade de login para validar campos obrigatórios.
- Substitua a classe ActivityLogin pelo conteúdo abaixo:

```
/*
Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
*/
package br.com.minicurso;

import android.os.*;
import android.support.v7.app.*;
import android.view.*;
import android.widget.*;

public class LoginActivity extends ActionBarActivity
{
    protected Button btnSignIn, btnSignUp, btnRecuperarpassword;
    protected AutoCompleteTextView txtEmail, txtPassword;
    protected String Email, Password;
```



# Validação de Campos

```
@Override
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);
    this.txtEmail = (AutoCompleteTextView) findViewById(R.id.editEmail);
    this.txtPassword = (AutoCompleteTextView) findViewById(R.id.editPassword);
    this.btnSignIn = (Button) findViewById(R.id.btn_signIn_small);

    this.btnSignIn.setOnClickListener(new View.OnClickListener()
    {
        public void onClick(View v)
        {
            Email = txtEmail.getText().toString();
            Password = txtPassword.getText().toString();

            if ( Email.isEmpty() )
            {
                txtEmail.setError(getResources().getString(R.string.empty_email));
            }
            if ( Password.isEmpty() )
            {

txtPassword.setError(getResources().getString(R.string.empty_password));
            }
        }
    });
}
```



# Validação de Campos

```
this.btnRecuperarpassword = (Button) findViewById(R.id.btn_forgotYourpassword);
this.btnRecuperarpassword.setOnClickListener(new View.OnClickListener()
{
    public void onClick(View v)
    {
        Toast.makeText(getApplicationContext(), "Ainda não Implementado",
Toast.LENGTH_SHORT).show();
    }
});
this.btnSignUp = (Button) findViewById(R.id.btn_signUp);
this.btnSignUp.setOnClickListener(new View.OnClickListener()
{
    public void onClick(View v)
    {
        Toast.makeText(getApplicationContext(), "Ainda não Implementado",
Toast.LENGTH_SHORT).show();
    }
});
```



# Validação de Campos

```
@Override  
public boolean onCreateOptionsMenu(Menu menu)  
{  
    return true;  
}  
@Override  
public boolean onOptionsItemSelected(MenuItem item)  
{  
    return super.onOptionsItemSelected(item);  
}
```



# Validação de Campos

- Por fim adicionar as novas Strings

```
<string name="erro">Erro</string>
<string name="empty_email">Email Obrigatório</string>
<string name="empty_password">Senha Obrigatória</string>
```



# Validação de Campos

- Resultado Final:



## Minicurso M

Desenvolvimento Mobile

### Informações da Conta

Email  !

Senha  !

**ENTRAR**

**ESQUECEU SUA SENHA?**

**criar conta**

Ainda não Implementado



## Minicurso M

Desenvolvimento Mobile

### Informações da Conta

Email  !

Senha  !

**ENTRAR**



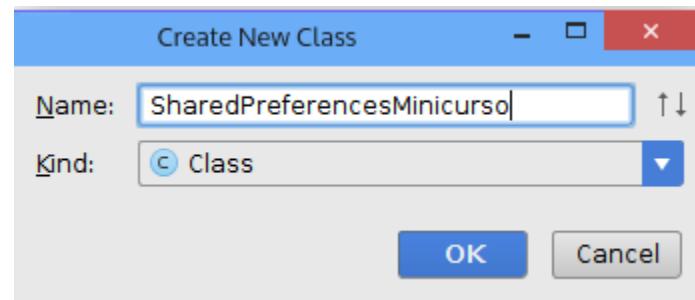
# Shared Preferences

- O SharedPreferences é uma das opções de armazenamento de dados que o Android prove para suas aplicações.
- Recurso úteis para armazenar e compartilhar com segurança pequenas informações entre as atividades do aplicativo.
- Como por exemplo o nome ou e-mail do usuário.
- Por padrão somente o aplicativo tem acesso a essas informações.



# Shared Preferences

- Crie a classe SharedPreferencesMinicurso, que sera utilizada futuramente



# Shared Preferences

- Substitua seu conteúdo por:

```
/*
Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
*/
package br.com.minicurso;

import android.app.*;
import android.content.*;

public class SharedPreferencesMinicurso
{
    protected Activity context;
    protected String UserName;
    public static final String PREFS_NAME = "MinicursoPrefsFile";

    public SharedPreferencesMinicurso(Activity context)
    {
        this.context = context;
    }
}
```



# Shared Preferences

- Substitua seu conteúdo por:

```
public void setUserName(String UserName)
{
    SharedPreferences settings = this.context.getSharedPreferences(PREFS_NAME, 0);
    SharedPreferences.Editor editor = settings.edit();
    editor.putString("UserName", UserName);
    editor.commit();
}

public String getUserName()
{
    SharedPreferences settings = this.context.getSharedPreferences(PREFS_NAME, 0);
    this.UserName = settings.getString("UserName", UserName);
    return this.UserName;
}
```



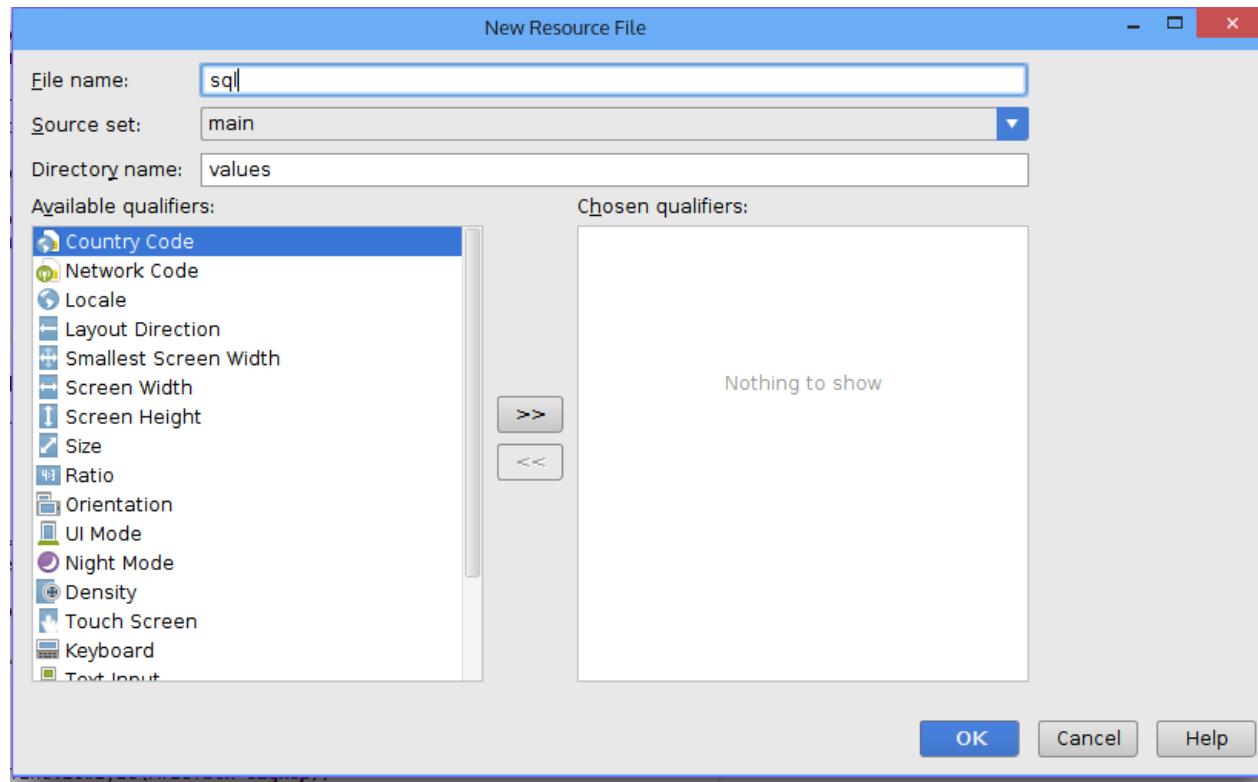
# Persistência SQL

- Inicialmente criaremos um pequeno banco de dados SQL local.
- O android utiliza nativamente o SQLite para esse fim.



# Persistência SQL

- Crie o resource sql.xml, onde adicionaremos os sqls locais da aplicação



# Persistência SQL

- Cole o conteúdo a seguir no arquivo sql.xml:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>-->
<resources>
    <string name="SQLiteHelper_onCreate" translatable="false">

        create table tbpais
        (
            cdpais integer not null,
            nmpais varchar unique not null,
            imgbandeira varchar,
            dtincreg timestamp default current_timestamp,
            dtaltreg timestamp default current_timestamp,
            constraint pk_tbpais_cdpais primary key(cdpais)
        );
        create table tbestado
        (
            cdestado integer not null,
            cdpais integer not null,
            nmestado varchar unique not null,
            sgestado char(2),
            dtincreg timestamp default current_timestamp,
            dtaltreg timestamp default current_timestamp,
            constraint pk_tbestado_cdestado primary key(cdestado),
            FOREIGN KEY (cdpais) REFERENCES TBPAIS(cdpais)
        );
    </string>
</resources>
```



# Persistência SQL

```
create table tbcidade
(
    cdcidade integer not null,
    cdestado integer not null,
    nmcidade varchar not null,
    dtinreg timestamp default current_timestamp,
    dtaltreg timestamp default current_timestamp,
    constraint pk_tbcidade_cdcidade primary key(cdcidade),
    FOREIGN KEY (cdestado) REFERENCES TBESTADO(cdestado)
);
create table tbbairro
(
    cdbairro integer not null,
    cdcidade integer not null,
    nmbairro varchar not null,
    dtinreg timestamp default current_timestamp,
    dtaltreg timestamp default current_timestamp,
    constraint pk_tbbairro_cdbairro primary key(cdbairro),
    FOREIGN KEY (cdcidade) REFERENCES tbcidade(cdcidade)
);
```



# Persistência SQL

```
create table tbrua
(
    cdrua integer not null,
    cdbairro integer not null,
    nmrua varchar not null,
    dtincreg timestamp default current_timestamp,
    dtaltreg timestamp default current_timestamp,
    constraint pk_tbrua_cdrua primary key(cdrua),
    FOREIGN KEY (cdbairro) REFERENCES tbbairro(cdbairro)
);
create table tbresidencia
(
    cdresidencia integer not null,
    cdrua integer,
    nuresidencia int,
    cep int,
    dtincreg timestamp default current_timestamp,
    dtaltreg timestamp default current_timestamp,
    constraint pk_tbresidencia_cdresidencia primary key(cdresidencia),
    FOREIGN KEY (cdrua) REFERENCES tbrua(cdrua)
);
```



# Persistência SQL

```
create table tbpessoa
(
    cdpessoa integer not null,
    cdresidencia integer,
    nmpessoa varchar not null,
    imgpessoa varchar,
    dtincreg timestamp default current_timestamp,
    dtaltreg timestamp default current_timestamp,
    constraint pk_tbpessoa_cdpessoa primary key(cdpessoa),
    FOREIGN KEY (cdresidencia) REFERENCES tbresidencia(cdresidencia)
);
create table tbusuario
(
    cdusuario integer not null,
    cdpessoa integer not null,
    email varchar not null,
    senha varchar,
    dtincreg timestamp default current_timestamp,
    dtaltreg timestamp default current_timestamp,
    constraint pk_tbusuario_cdusuario primary key(cdusuario),
    FOREIGN KEY (cdpessoa) REFERENCES tbpessoa(cdpessoa)
);
```



# Persistência SQL

```
Insert into tbpais (cdpais,nmpais,imgbandeira) Values (1,\'Brasil\',null);
Insert into tbestado (cdestado,cdpais,nmestado,sgestado) Values
(2,1,\'Alagoas\',\'AL\');
Insert into tbcidade (cdcidade,cdestado,nmcidade) Values (115,2,\'São Miguel dos
Milagres\');
Insert into tbrua (cdrua,cdbairro,nmrua) Values (1,1,\'Altar de Deus\');
Insert into tbresidencia (cdresidencia,cdrua,nuresidencia,cep) Values
(1,1,115,\'35172001\');
Insert into tbpessoa (cdpessoa,cdresidencia,nmpessoa,imgpessoa) Values
(1,1,\'Exemplo\',null);
Insert into tbusuario (cdusuario,cdpessoa,email,senha) Values
(1,1,\'exemplo@exemplo.com\',\'8EB90EC152BD30F4A53F15BF805783DC\');

</string>

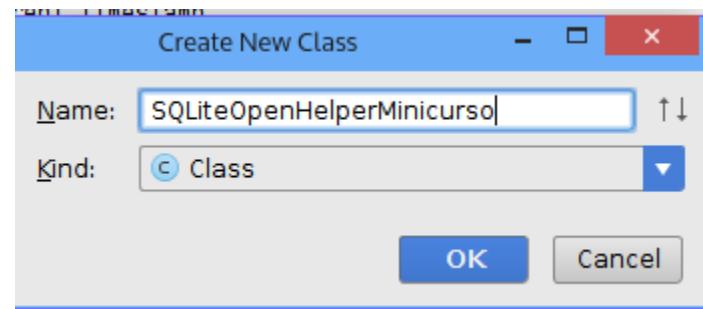
<string name="SQLiteHelper_onUpgrade" translatable="false"></string>

</resources>
```



# Persistência SQL

- Crie a classe SQLiteOpenHelperMinicurso



# Persistência SQL

- Cole o conteúdo a seguir na classe SQLiteOpenHelperMinicurso:

```
/*
Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
*/

package br.com.minicurso;

import android.content.*;
import android.database.*;
import android.database.sqlite.*;
import android.util.*;

public class SQLiteOpenHelperMinicurso extends SQLiteOpenHelper
{
    protected static int DATABASE_VERSION = 1;
    protected static String DATABASE_NAME = "minicurso.sqlite.db";
    protected static String LOG_TAG = "minicurso.sqlite";
    protected Context contexto;

    public SQLiteOpenHelperMinicurso(Context context)
    {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
        this.contexto = context;
    }
}
```



# Persistência SQL

```
public void onCreate(SQLiteDatabase db)
{
    String[] sql = contexto.getString(R.string.SQLiteHelper_onCreate).split(";");
    db.beginTransaction();
    try
    {
        ExecutarComandosSQL(db, sql);
        db.setTransactionSuccessful();
    }
    catch (SQLException e)
    {
        Log.e("Erro Interno", e.toString());
    }
    finally
    {
        db.endTransaction();
    }
}
```



# Persistência SQL

```
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
{
    Log.w(LOG_TAG, "Atualizando a base de dados da versão " + oldVersion + " para " + newVersion + ", que destruirá todos os dados antigos");
    String[] sql =
contexto.getString(R.string.SQLiteHelper_onUpgrade).split(";");
    db.beginTransaction();
    try
    {
        ExecutarComandosSQL(db, sql);
        db.setTransactionSuccessful();
    }
    catch (SQLException e)
    {
        Log.e("Erro Interno", e.toString());
        throw e;
    }
    finally
    {
        db.endTransaction();
    }
onCreate(db);
}
```



# Persistência SQL

```
public void onDowngrade(SQLiteDatabase db, int oldVersion, int newVersion)
{
    onUpgrade(db, oldVersion, newVersion);
}
private void ExecutarComandosSQL(SQLiteDatabase db, String[] sql)
{
    for (String s : sql)
    {
        if (s.trim().length() > 0)
        {
            db.execSQL(s);
        }
    }
}
```



# Persistência SQL

```
public String getData(String tabela, String[] projection, String campo, String valor)
{
    SQLiteDatabase db = getReadableDatabase();
    String consulta = "nf";
    String selection = campo + " LIKE ?";
    String[] selectionArgs = {String.valueOf(valor)};
    String sortOrder = campo + " DESC";
    Cursor c = db.query(
        tabela,
        projection,
        selection,
        selectionArgs,null,null,sortOrder
    );
    if (c.moveToFirst())
    {
        do
        {
            c.moveToFirst();
            consulta = c.getString(c.getColumnIndex(projection[0]));
        }
        while (c.moveToNext());
    }
    c.close();
    db.close();
    return consulta;
}
```



# Persistência SQL

- Adicionar na classe LoginActivity:

```
import java.io.*;
import java.security.*;
```

- Substituir o evento do botão btnSignIn por:

```
this.btnSignIn = (Button) findViewById(R.id.btn_signIn_small);
this.btnSignIn.setOnClickListener(new View.OnClickListener()
{
    public void onClick(View v)
    {
        if (ValidaLogin())
        {
            Toast.makeText(getApplicationContext(), "Validação de Login Bem Sucedida",
Toast.LENGTH_SHORT).show();
        }
    }
});
```



# Persistência SQL

- Adicionar na classe LoginActivity os métodos:

```
public boolean ValidaLogin()
{
    Email = txtEmail.getText().toString();
    Password = txtPassword.getText().toString();
    if ( Email.isEmpty() )
    {
        txtEmail.setError(getResources().getString(R.string.empty_email));
        return false;
    }
    if ( Password.isEmpty() )
    {
        txtPassword.setError(getResources().getString(R.string.empty_password));
        return false;
    }

    final SQLiteOpenHelperMinicurso db = new SQLiteOpenHelperMinicurso(getApplicationContext());
    String retornoConsulta, tabelaConsulta, campoConhecido, valorConhecido;
    tabelaConsulta = "tbusuario";
    String[] retornoEsperado = {"senha"};
    campoConhecido = "Email";
    valorConhecido = Email;
    retornoConsulta = db.getData(tabelaConsulta, retornoEsperado, campoConhecido,
    valorConhecido);
```



# Persistência SQL

```
String PasswordHash = PasswordHash(this.Password);
if ( !retornoConsulta.equals(PasswordHash) )
{
    txtEmail.setError(getResources().getString(R.string.email_password_invalid));
    txtPassword.setError(getResources().getString(R.string.email_password_invalid));
    return false;
}
return true;
}
public String PasswordHash(String password)
{
    MessageDigest algorithm = null;
    try
    {
        algorithm = MessageDigest.getInstance("MD5");
    }
    catch (NoSuchAlgorithmException e)
    {
        e.printStackTrace();
    }
    byte messageDigest[] = new byte[0];
```



# Persistência SQL

```
try
{
    messageDigest = algorithm.digest(password.getBytes("UTF-8"));
}
catch (UnsupportedEncodingException e)
{
    e.printStackTrace();
}
StringBuilder hexString = new StringBuilder();
for (byte b : messageDigest)
{
    hexString.append(String.format("%02X", 0xFF & b));
}
return hexString.toString();
}
```



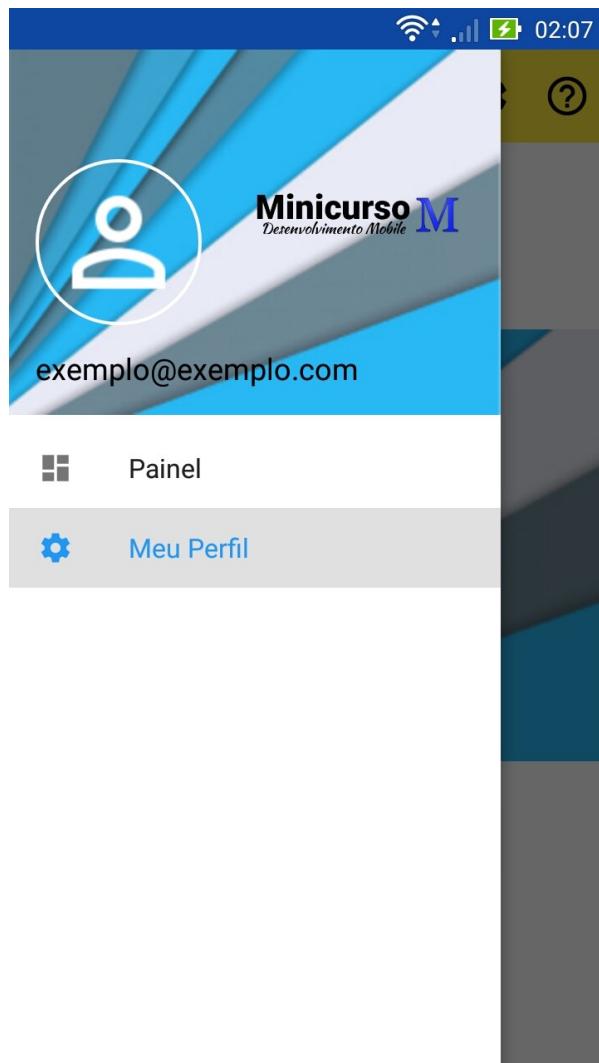
# Persistência SQL

- Adicionar novas strings:

```
<string name="email_password_invalid">Email ou Senha Incorreto(s)</string>
```

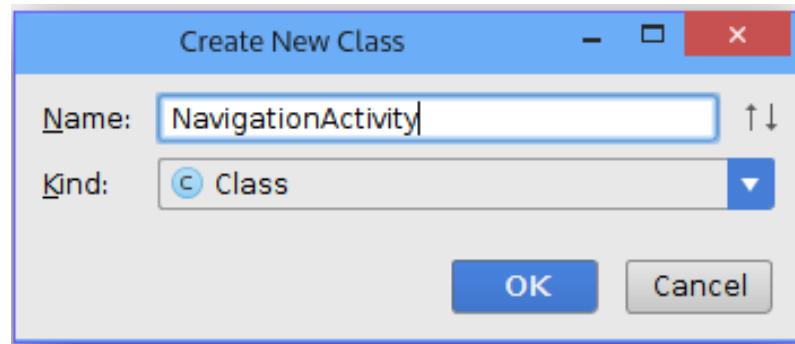


# Navigation Drawer



# Navigation Drawer

- Crie a classe NavigationActivity



# Navigation Drawer

- Substitua o conteúdo da classe NavigationActivity por:

```
/*
Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
*/
package br.com.minicurso;

import android.app.*;
import android.content.*;
import android.content.res.*;
import android.graphics.drawable.*;
import android.os.*;
import android.support.design.widget.*;
import android.support.v4.widget.*;
import android.support.v7.app.ActionBar;
import android.support.v7.widget.Toolbar;
import android.support.v7.app.*;
import android.view.*;
import android.widget.*;
```



# Navigation Drawer

```
public class NavigationActivity extends AppCompatActivity
{
    protected String activityName;
    protected Toolbar toolbar_dark, toolbar_light;
    protected DrawerLayout dlDrawer;
    protected String option;
    protected int i;
    protected Class fragmentClass;
    protected Fragment fragment = null;
    protected ActionBar actionBar;
    protected TextView textViewUser;
    protected Menu menu;
    protected ActionBarDrawerToggle drawerToggle;
    protected String userName;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_navigation);
        this.activityName = getClass().getName();
        this.dlDrawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        drawerToggle = setupDrawerToggle();
        this.dlDrawer.setDrawerListener(drawerToggle);
    }
}
```



# Navigation Drawer

```
final SharedPreferencesMinicurso sharedPreferencesMinicurso = new
SharedPreferencesMinicurso(this);
this.userName = sharedPreferencesMinicurso.getUserName();
this.textUser = (TextView) findViewById(R.id.textUser);
this.textUser.setText(userName);
Intent intent = getIntent();
Bundle params = intent.getExtras();

if(params!=null)
{
    this.option = params.getString("option");
    this.i = Integer.parseInt(option);
    setFragment(this.i);
}
else
{
    setFragment(0);
}

NavigationView nvDrawer = (NavigationView) findViewById(R.id.nvView);
setupDrawerContent(nvDrawer);
setFragmentManager();
}

private ActionBarDrawerToggle setupDrawerToggle()
{
    return new ActionBarDrawerToggle(this, dlDrawer, toolbar_dark,
R.string.drawer_open, R.string.drawer_close);
}
```



# Navigation Drawer

```
private void setupDrawerContent(NavigationView navigationView)
{
    navigationView.setNavigationItemSelectedListener(new
NavigationView.OnNavigationItemSelectedListener()
{
    public boolean onNavigationItemSelected(MenuItem menuItem)
    {
        selectDrawerItem(menuItem);
        return true;
    }
});
}

public void toolbarLight(Drawable color)
{
    this.toolbar_dark = (Toolbar) findViewById(R.id.toolbar_dark);
    this.toolbar_dark.setVisibility(View.GONE);
    this.toolbar_light = (Toolbar) findViewById(R.id.toolbar_light);
    this.toolbar_light.setVisibility(View.VISIBLE);
    this.toolbar_light.setTitleTextColor(getResources().getColor(R.color.Black));
    this.toolbar_light.setBackground(color);
    setSupportActionBar(this.toolbar_light);
    this.actionbar = getSupportActionBar();
    this.actionbar.setDisplayHomeAsUpEnabled(true);
}
```



# Navigation Drawer

```
public void toolbarDark(Drawable color)
{
    this.toolbar_light = (Toolbar) findViewById(R.id.toolbar_light);
    this.toolbar_light.setVisibility(View.GONE);
    this.toolbar_dark = (Toolbar) findViewById(R.id.toolbar_dark);
    this.toolbar_dark.setVisibility(View.VISIBLE);
    this.toolbar_dark.setTitleTextColor(getResources().getColor(R.color.White));
    this.toolbar_dark.setBackground(color);
    setSupportActionBar(this.toolbar_dark);
    this.actionbar = getSupportActionBar();
    this.actionbar.setDisplayHomeAsUpEnabled(true);
}

public void actionBarBlue()
{
    Drawable colorBlue = getResources().getDrawable(R.color.Blue_500);
    toolbarDark(colorBlue);
};

public void actionBarYellow()
{
    Drawable colorYellow = getResources().getDrawable(R.color.Yellow_500);
    toolbarLight(colorYellow);
};

public void actionBarRed()
{
    Drawable colorRed = getResources().getDrawable(R.color.Red_500);
    toolbarDark(colorRed);
};
```



# Navigation Drawer

```
public void actionBarGreen()
{
    Drawable colorGreen = getResources().getDrawable(R.color.Green_500);
    toolbarDark(colorGreen);
};

public void setFragment(int i)
{
    switch(i)
    {
        case 1:
            actionBarBlue();
            this.fragmentClass = FragmentDashboard.class;
            this.actionbar.setTitle(R.string.dashboard);
            break;
        case 2:
            actionBarYellow();
            this.fragmentClass = FragmentMyProfile.class;
            this.actionbar.setTitle(R.string.myProfile);
            break;
        default:
            actionBarBlue();
            this.fragmentClass = FragmentDashboard.class;
            this.actionbar.setTitle(R.string.dashboard);
    }
}
```



# Navigation Drawer

```
public void selectDrawerItem(MenuItem menuItem)
{
    switch(menuItem.getItemId())
    {
        case R.id.fragment_dashboard:
            setFragment(1);
            break;
        case R.id.fragment_myprofile:
            setFragment(2);
            break;
        default:
            setFragment(0);
    }
    setFragmentManager();
    menuItem.setChecked(true);
    this.dlDrawer.closeDrawers();
}

public void setFragmentManager()
{
    try{
        this.fragment = (Fragment) this.fragmentClass.newInstance();
    catch (Exception e){
        e.printStackTrace();
    FragmentManager fragmentManager = getFragmentManager();
    fragmentManager.beginTransaction().replace(R.id.flContent, this.fragment).commit();
}
}
```



# Navigation Drawer

```
@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    getMenuInflater().inflate(R.menu.menu_black_actions_drawer, menu);
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    switch (item.getItemId())
    {
        case R.id.action_settings:
        {
            Toast.makeText(this, R.string.settings, Toast.LENGTH_SHORT).show();
            setFragmentManager();
            return true;
        }
        case R.id.action_help:
        {
            Toast.makeText(this, R.string.help, Toast.LENGTH_SHORT).show();
            setFragmentManager();
            return true;
        }
        default:
            break;
    }
}
```



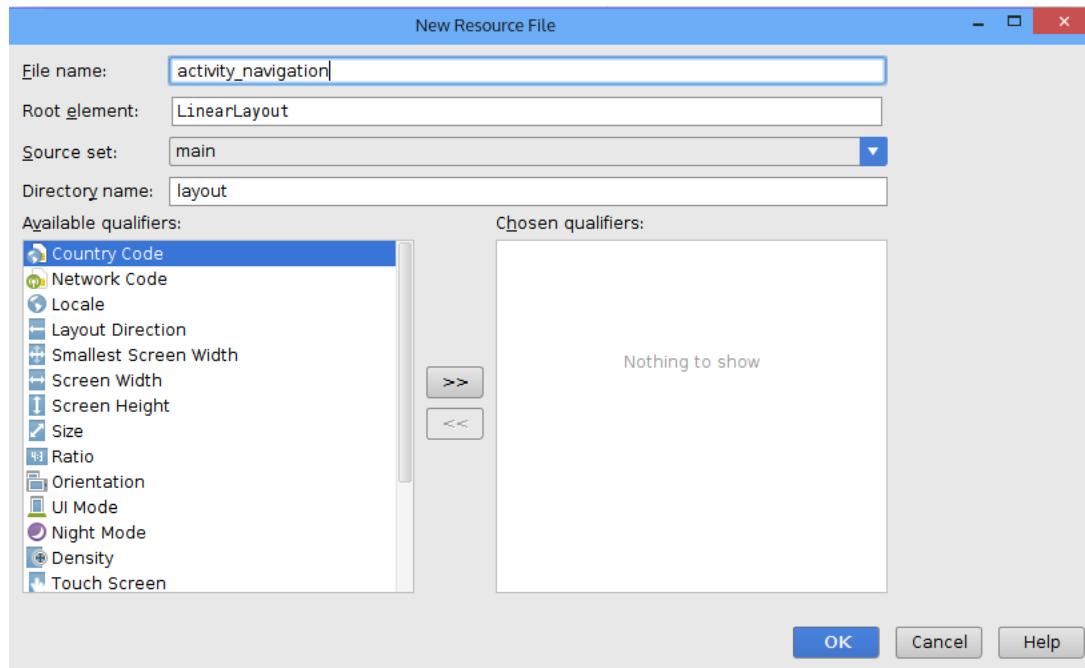
# Navigation Drawer

```
if (this.drawerToggle.onOptionsItemSelected(item))
{
    return true;
}
return super.onOptionsItemSelected(item);
}
@Override
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    this.drawerToggle.syncState();
}
@Override
public void onConfigurationChanged(Configuration newConfig)
{
    super.onConfigurationChanged(newConfig);
    this.drawerToggle.onConfigurationChanged(newConfig);
}
```



# Navigation Drawer

- Crie os layouts:activity\_navigation.xml, toolbar\_light.xml, toolbar\_dark.xml e navigation\_header.xml.
- Serão explicados mais a frente.



# Navigation Drawer

- Modifique o arquivo activity\_navigation.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!--
    Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
-->

<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/drawer_layout"
    style="@style/Main">

    <LinearLayout
        style="@style/BackgroundActionBar">
        <include
            layout="@layout/toolbar_dark"/>
        <include
            layout="@layout/toolbar_light"/>
        <LinearLayout
            style="@style/Context">
            <FrameLayout
                android:id="@+id/f1Content"
                style="@style/FrameLayout"/>
        </LinearLayout>
    </LinearLayout>
</android.support.v4.widget.DrawerLayout>
```



# Navigation Drawer

```
<android.support.design.widget.NavigationView
    style="@style/NavigationView"
    android:id="@+id/nvView"
    app:headerLayout="@layout/navigation_header"
    app:menu="@menu/menu_navigation_view"/>
</android.support.v4.widget.DrawerLayout>
```



# Navigation Drawer

- Modifique o arquivo toolbar\_light.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!--
    Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
-->
<android.support.v7.widget.Toolbar
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/toolbar_light"
    android:visibility="gone"
    style="@style/Toolbar"
    app:theme="@style/ThemeOverlay.AppCompat.Light">
</android.support.v7.widget.Toolbar>
```



# Navigation Drawer

- Modifique o arquivo toolbar\_dark.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!--
    Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
-->

<android.support.v7.widget.Toolbar
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/toolbar_dark"
    android:visibility="gone"
    style="@style/Toolbar"
    app:theme="@style/ThemeOverlay.AppCompat.Dark">
</android.support.v7.widget.Toolbar>
```



# Navigation Drawer

- Modifique o arquivo navigation\_header.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com> -->

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    style="@style/ActivityNavigation"
    android:background="@drawable/nav_drawer_header_view_b">
    <de.hdodenhof.circleimageview.CircleImageView
        android:layout_marginTop="50dp"
        android:id="@+id/profile_image"
        style="@style/Img.Circle.ProfileSmall"
        android:src="@mipmap/ic_perm_identity_white"
        app:border_width="2dp"
        app:border_color="@color/White"/>
    <ImageView
        android:id="@+id/imgButton"
        style="@style/ImgButton"
        android:layout_above="@+id/textUser"/>
    <TextView
        android:id="@+id/textUser"
        android:text="email@email.com"
        style="@style/NavigationText" />
</RelativeLayout>
```

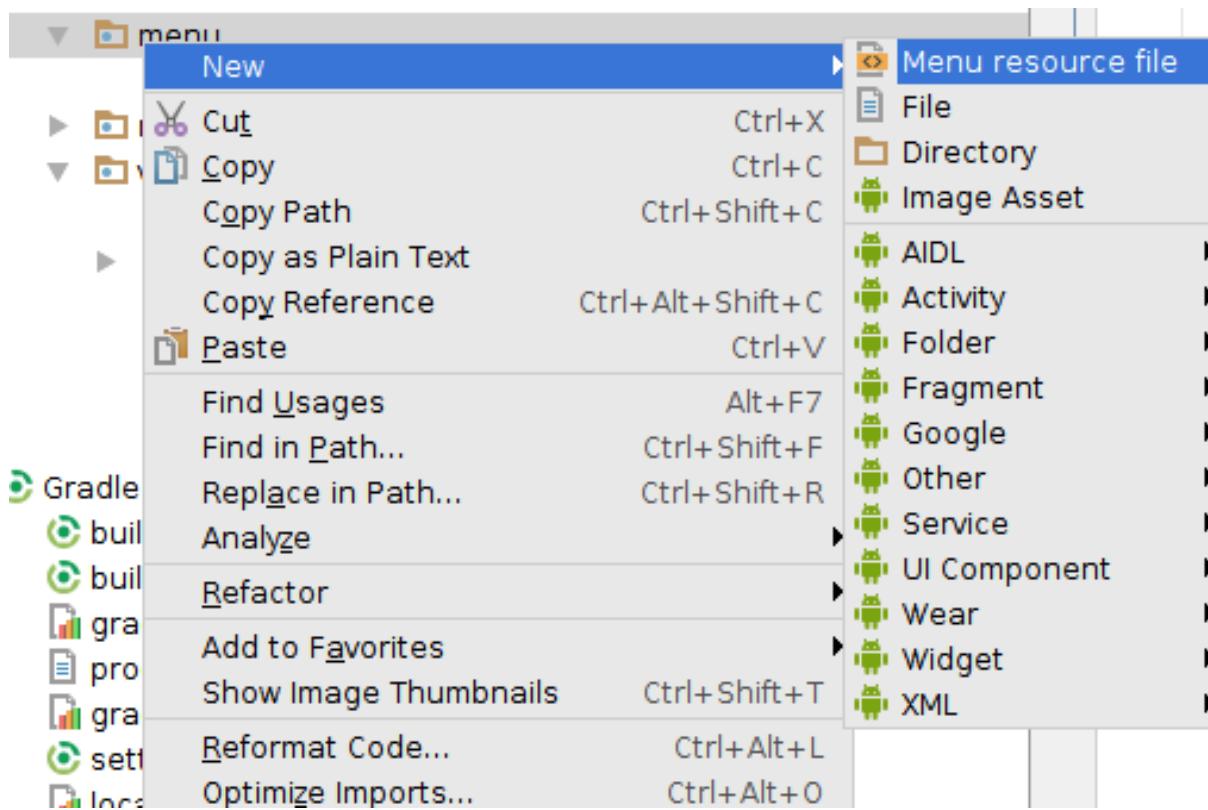


# Navigation Drawer

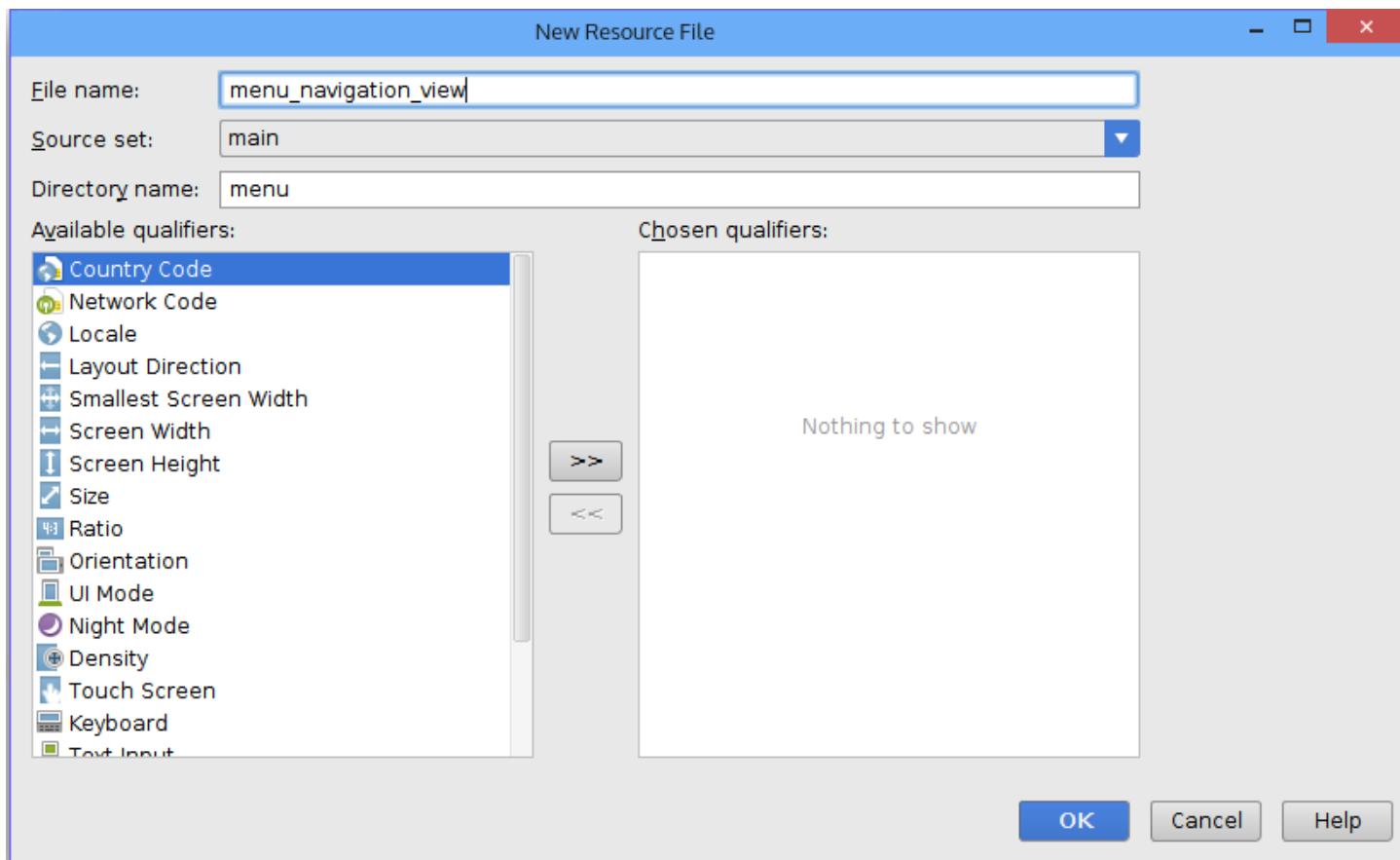
- Menu Resource file
- Crie três menus
  - menu\_navigation\_view:
  - menu\_black\_actions\_drawer



# Navigation Drawer



# Navigation Drawer



# Navigation Drawer

- Modifique o arquivo menu\_navigation\_view

```
<?xml version="1.0" encoding="utf-8"?>
<!--
    Copyright (c) 2015 Luiz Carlos <luiz04n1@gmail.com>
-->

<menu xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/menu">
    <group
        android:checkableBehavior="single">
        <item
            android:id="@+id/fragment_dashboard"
            android:icon="@mipmap/ic_dashboard_black"
            android:title="@string/dashboard" />
        <item
            android:id="@+id/fragment_myprofile"
            android:icon="@mipmap/ic_settings_black"
            android:title="@string/myProfile"/>
    </group>
</menu>
```



# Navigation Drawer

- Modifique o arquivo menu\_black\_actions\_drawer

```
<?xml version="1.0" encoding="utf-8"?>
<!--
    Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
-->

<menu
    android:id="@+id/menu"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:liquidity="http://schemas.android.com/apk/res-auto" >
    <item
        android:id="@+id/action_settings"
        android:icon="@mipmap/ic_settings_black"
        android:title="@string/settings"
        android:showAsAction="withText|ifRoom"
        liquidity:showAsAction="withText|ifRoom" />
    <item android:id="@+id/action_help"
        android:icon="@mipmap/ic_help_outline_black"
        android:title="@string/help"
        android:showAsAction="withText|ifRoom"
        liquidity:showAsAction="withText|ifRoom" />
</menu>
```



# Navigation Drawer

- Copiar a pasta drawable para src/main/res/



# Navigation Drawer

- Novas Strings

```
<string name="dashboard">Painel</string>
<string name="myProfile">Meu Perfil</string>
<string name="drawer_open" translatable="false">Open navigation
drawer</string>
<string name="drawer_close" translatable="false">Close navigation
drawer</string>
<string name="settings">Configurações</string>
<string name="help">Ajuda</string>
<string name="signout">Sair</string>
```



# Navigation Drawer

- Novos Estilos

```
<style name="BackgroundActionBar" parent="Background">
    <item name="android:layout_marginTop">0dp</item>
</style>
<style name="FrameLayout">
    <item name="android:layout_width">match_parent</item>
    <item name="android:layout_height">match_parent</item>
</style>
<style name="NavigationView">
    <item name="android:layout_width">wrap_content</item>
    <item name="android:layout_height">fill_parent</item>
    <item name="android:layout_gravity">left</item>
    <item name="android:background">@color/White</item>
</style>
<style name="ActivityNavigation">
    <item name="android:layout_width">match_parent</item>
    <item name="android:layout_height">220dp</item>
    <item name="android:background">?attr/colorPrimaryDark</item>
    <item name="android:padding">16dp</item>
    <item name="android:theme">@style/ThemeOverlay.AppCompat.Dark</item>
    <item name="android:orientation">vertical</item>
    <item name="android:gravity">bottom</item>
</style>
```



# Navigation Drawer

```
<style name="ImgButton">
    <item name="android:layout_marginLeft">120dp</item>
    <item name="android:src">@drawable/minicurso</item>
    <item name="android:layout_centerVertical">true</item>
    <item name="android:layout_width">wrap_content</item>
    <item name="android:layout_height">wrap_content</item>
</style>

<style name="NavigationText">
    <item name="android:layout_width">match_parent</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:textColor">@color/Black</item>
    <item name="android:textSize">18dp</item>
    <item name="android:layout_alignParentBottom">true</item>
    <item name="android:layout_alignParentLeft">true</item>
    <item name="android:layout_alignParentRight">true</item>
</style>
```



# Navigation Drawer

```
<style name="Toolbar">
    <item name="android:layout_width">match_parent</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:background">?attr/colorPrimary</item>
    <item name="android:fitsSystemWindows">true</item>
    <item name="android:minHeight">?attr/actionBarSize</item>
</style>
<style name="TitleActivity">
    <item name="android:layout_alignParentTop">true</item>
    <item name="android:layout_centerHorizontal">true</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:layout_width">wrap_content</item>
    <item name="android:textSize">40dp</item>
</style>
```



# Navigation Drawer

```
<style name="Img.Circle" parent="Img">
    <item name="android:layout_alignParentTop">true</item>
    <item name="android:layout_centerVertical">true</item>
    <item name="android:layout_alignParentLeft">true</item>
</style>
<style name="Img.Circle.ProfileSmall" parent="Img.Circle">
    <item name="android:layout_width">100dp</item>
    <item name="android:layout_height">100dp</item>
</style>
<style name="Img.Circle.ProfileBig" parent="Img.Circle">
    <item name="android:layout_width">140dp</item>
    <item name="android:layout_height">140dp</item>
</style>
<style name="Shadow.Land" parent="Shadow">
    <item name="android:layout_marginTop">10dp</item>
    <item name="android:orientation">horizontal</item>
    <item name="android:layout_centerHorizontal">true</item>
</style>
<style name="ButtonArea">
    <item name="android:layout_width">wrap_content</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:orientation">vertical</item>
</style>
```



# Navigation Drawer

```
<style name="ButtonSignOut" parent="ButtonSignInSmall">
    <item name="android:background">@drawable/button_blue</item>
    <item name="android:text">@string/signout</item>
</style>
<style name="LinearLayout">
    <item name="android:layout_width">wrap_content</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:layout_centerHorizontal">true</item>
    <item name="android:layout_marginBottom">50dp</item>
</style>
<style name="NavImage">
    <item name="android:layout_width">fill_parent</item>
    <item name="android:layout_height">260dp</item>
    <item name="android:background">@drawable/nav_drawer_header_view_b</item>
</style>
<style name="NavImage.Land" parent="NavImage">
    <item name="android:layout_height">180dp</item>
</style>
<style name="LinearLayout.Land" parent="LinearLayout">
    <item name="android:layout_marginBottom">20dp</item>
</style>
```



# Navigation Drawer

- Adicione ao AndroidManifest.xml

```
<activity
    android:name=".NavigationActivity"
    android:icon="@mipmap/ic_logo"
    android:label="@string/app_name"
    android:theme="@style/Theme.AppCompat.Light.NoActionBar.Minicurso.Blue" >
</activity>
```



# Adicionar Dependências

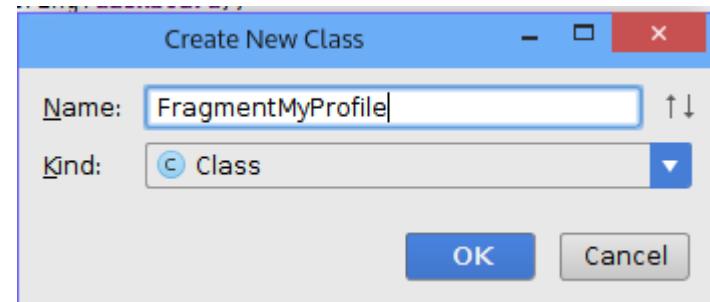
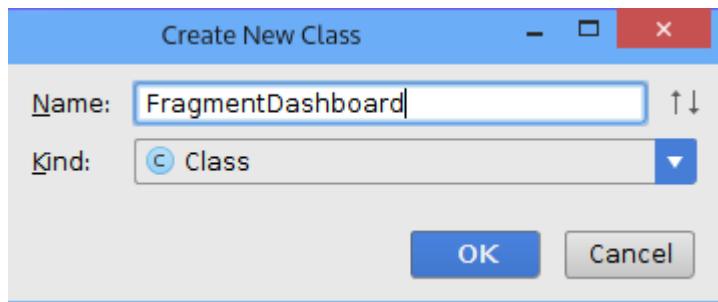
- Adicione ao build.gradle:

```
dependencies
{
    compile 'de.hdodenhof:circleimageview:1.3.0'
}
```



# Fragments

- Os fragmentos do android são partes de uma atividade que podem ser carregados pela atividade.
- Crie duas classes chamadas FragmentDashboard e FragmentMyProfile



# Fragments

- Modifique a clase FragmentDashboard

```
/*
Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
*/

package br.com.minicurso;
import android.app.*;
import android.os.*;
import android.view.*;

public class FragmentDashboard extends Fragment
{
    protected String userName;
    public FragmentDashboard() {}
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup accountiner,
                           Bundle savedInstanceState)
    {
        View rootView = inflater.inflate(R.layout.fragment_dashboard, accountiner, false);
        SharedPreferencesMinicurso userName = new
        SharedPreferencesMinicurso(getActivity());
        this.userName = userName.getUserName();
        return rootView;
    }
}
```



# Fragments

- Modifie a classe FragmentMyProfile

```
/*
Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
*/
package br.com.minicurso;
import android.app.*;
import android.content.*;
import android.os.*;
import android.view.*;
import android.widget.*;

public class FragmentMyProfile extends Fragment
{
    protected Button btnSignOut;
    public FragmentMyProfile(){}
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup accountiner, Bundle savedInstanceState)
    {
        final View rootView = inflater.inflate(R.layout.fragment_myprofile, accountiner,
false);
```



# Fragments

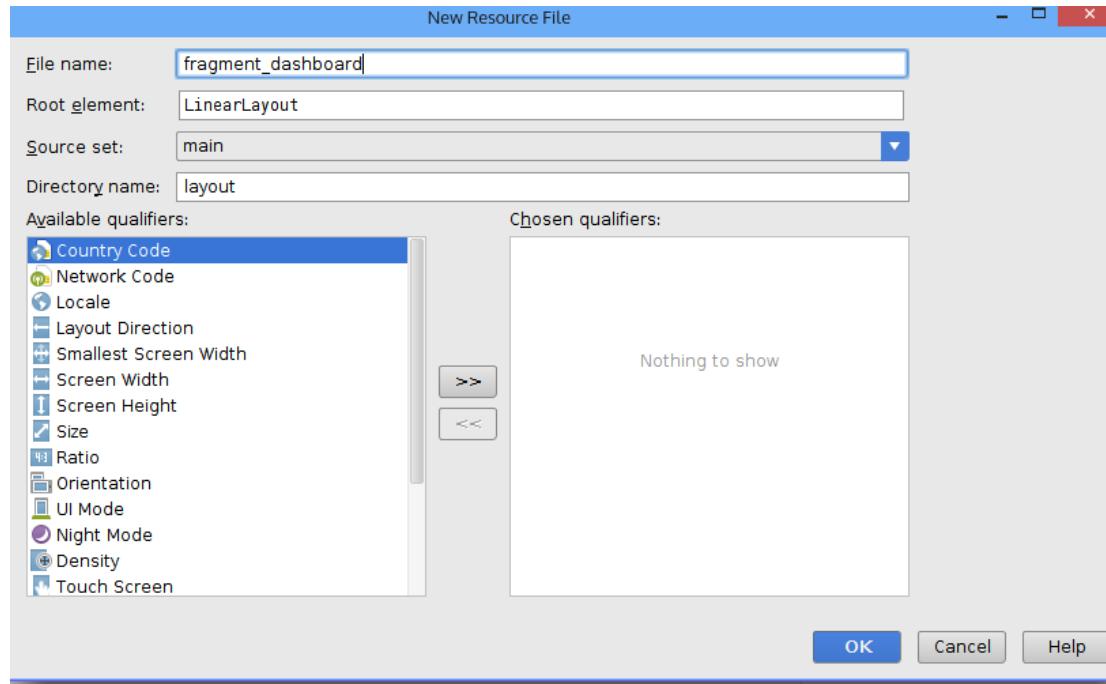
```
this.btnSignOut = (Button) rootView.findViewById(R.id.btn_sign_out);
this.btnSignOut.setOnClickListener(new View.OnClickListener()
{
    public void onClick(View v)
    {
        SignOut();
    }
});
return rootView;
}

public void SignOut()
{
    final SharedPreferencesMinicurso sharedpreferencesMinicurso = new
SharedPreferencesMinicurso(getActivity());
    sharedpreferencesMinicurso.setUserName(null);
    final Intent it = new Intent(getActivity(), LoginActivity.class);
    it.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
    startActivity(it);
    getActivity().overridePendingTransition(android.R.anim.slide_in_left,
android.R.anim.slide_out_right);
}
```



# Fragments

- Layouts para os fragmentos.
- Crie um layout chamado `fragment_dashboard`.
- E um chamado `fragment_myprofile`.



# Fragments

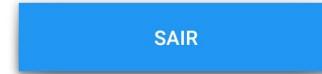
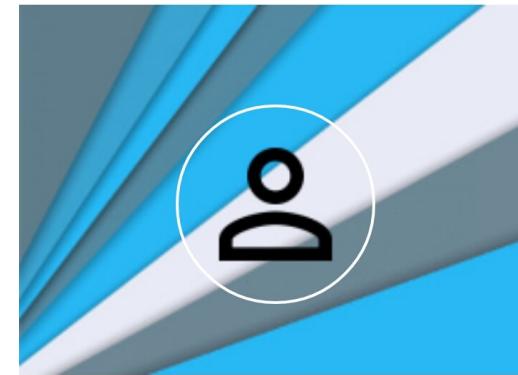
- Modifique o arquivo fragment\_dashboard.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!--
    Copyright (c) 2015 Luiz Carlos <luz04nl@gmail.com>
-->

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    style="@style/Main">
    <RelativeLayout
        style="@style/Background">
        <TextView
            android:id="@+id/titulo"
            android:text="@string/dashboard"
            style="@style/TitleActivity"/>
    </RelativeLayout>
</LinearLayout>
```



# Fragments



# Fragments

- Modifique o arquivo fragment\_myprofile.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!--
    Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
-->

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    style="@style/Main">

    <RelativeLayout
        style="@style/BackgroundActionBar">
        <ImageView
            android:id="@+id/imageView"
            style="@style/NavImage"/>
        <LinearLayout
            android:id="@+id/linearLayout"
            android:layout_alignBottom="@+id/imageView"
            style="@style/Linearlayout">
            <de.hdodenhofer.circleimageview.CircleImageView
                android:id="@+id/profile_image_myprofile"
                style="@style/Img.Circle.ProfileBig"
                android:src="@mipmap/ic_perm_identity_black"
                app:border_color="@color/White"
                app:border_width="2dp"/>
        </LinearLayout>
    </RelativeLayout>
</LinearLayout>
```



# Fragments

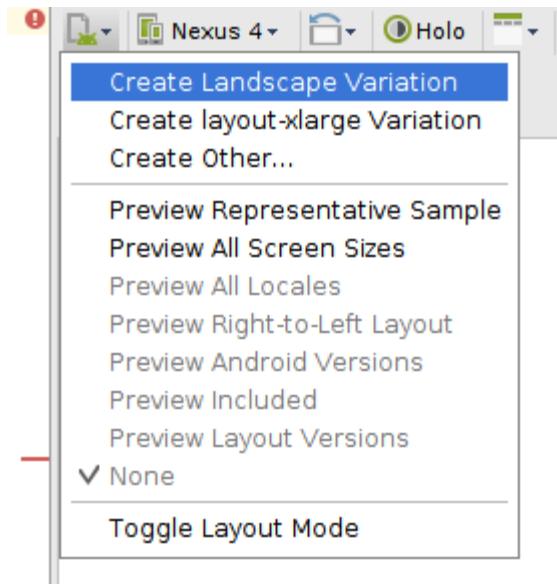
- Modifique o arquivo fragment\_myprofile

```
<LinearLayout
    android:id="@+id/linearLayout2"
    style="@style/Shadow_Land"
    android:layout_below="@+id/imageView">
    <LinearLayout
        style="@style/ButtonArea">
        <Button
            android:id="@+id/btn_sign_out"
            style="@style/ButtonSignOut" />
    </LinearLayout>
</LinearLayout>
</RelativeLayout>
</LinearLayout>
```



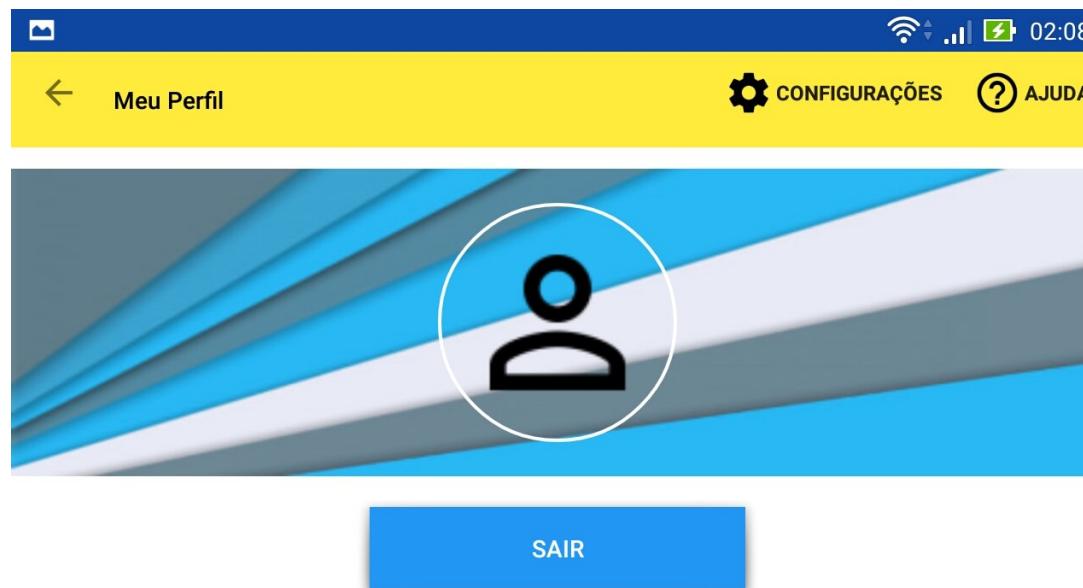
# Fragments

- Gere uma variação landscape para o layout fragment\_myprofile



# Fragments

- Modifique o arquivo fragment\_myprofile (land)



# Fragments

```
<?xml version="1.0" encoding="utf-8"?>
<!--
    Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
-->

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    style="@style/Main">
    <RelativeLayout
        style="@style/BackgroundActionBar">
        <ImageView
            android:id="@+id/imageView"
            style="@style/NavImage.Land"/>
        <LinearLayout
            android:id="@+id/linearLayout"
            android:layout_alignBottom="@+id/imageView"
            style="@style/LinearLayout.Land">
            <de.hdodenhof.circleimageview.CircleImageView
                android:id="@+id/profile_image_myprofile"
                style="@style/Img.Circle.ProfileBig"
                android:src="@mipmap/ic_perm_identity_black"
                app:border_color="@color/White"
                app:border_width="2dp"/>
        </LinearLayout>
    </RelativeLayout>
```



# Fragments

```
<LinearLayout  
    android:id="@+id/linearLayout2"  
    style="@style/Shadow_Land"  
    android:layout_below="@+id/imageView">  
    <LinearLayout  
        style="@style/ButtonArea">  
        <Button  
            android:id="@+id/btn_sign_out"  
            style="@style/ButtonSignOut" />  
    </LinearLayout>  
    </LinearLayout>  
    </RelativeLayout>  
</LinearLayout>
```



# Navigation Drawer

- Agora é necessário modificar a classe LoginActivity para redirecionar para a classe NavigationActivity caso o login seja bem sucedido
- Adicione:

```
import android.content.*;  
  
protected String userName;  
protected SharedPreferencesMinicurso sharedPreferencesMinicurso;
```

- apóis super.onCreate(savedInstanceState); adicione:

```
this.sharedPreferencesMinicurso = new SharedPreferencesMinicurso(this);  
this.userName = sharedPreferencesMinicurso.getUserName();  
if (userName != null)  
{  
    nextActivity();  
}  
else{}
```



# Navigation Drawer

- Agora é necessário modificar a classe LoginActivity para redirecionar para a classe NavigationActivity caso o login seja bem sucedido
- Recortar e cole o restante do método onCreate para dentro da condição else, pois só sera exibida caso o usuário ainda não tenha feito login.



# Navigation Drawer

- Substitua:

```
public void onClick(View v)
{
    if (ValidaLogin())
    {
        Toast.makeText(getApplicationContext(), "Validação de Login Bem
Sucedida", Toast.LENGTH_SHORT).show();
    }
}
```

- Por:

```
public void onClick(View v)
{
    if (ValidaLogin())
    {
        SharedPreferencesMinicurso.setUserName(Email);
        nextActivity();
    }
}
```



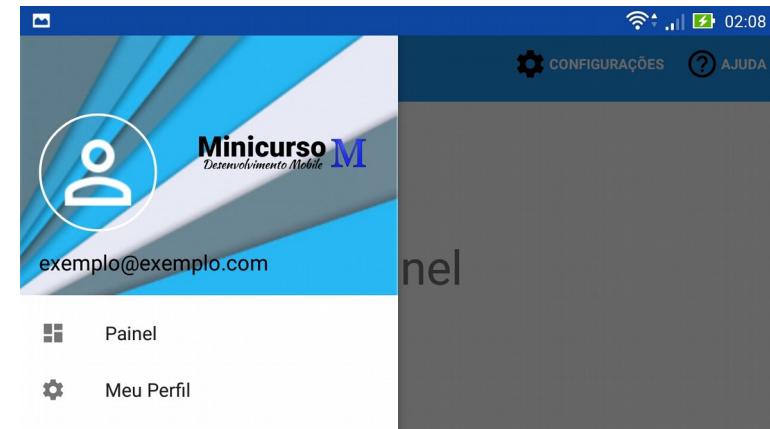
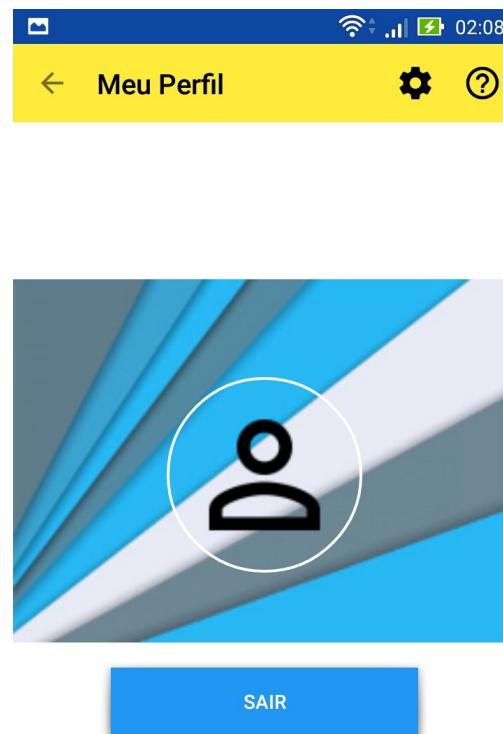
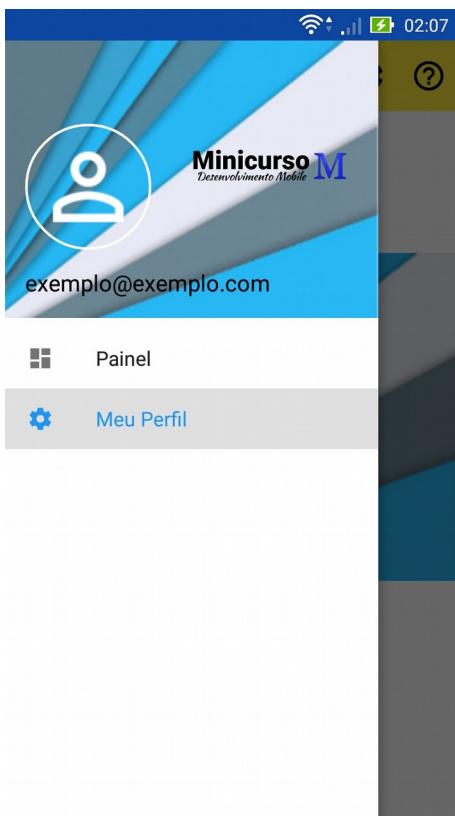
# Navigation Drawer

- E adicione o método:

```
public void nextActivity()
{
    final Intent it = new Intent(getApplicationContext(), NavigationActivity.class)
    final Bundle params = new Bundle();
    it.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
    params.putString("option", "1");
    it.putExtras(params);
    startActivity(it);
    overridePendingTransition(android.R.anim.slide_in_left,
    android.R.anim.slide_out_right);
}
```

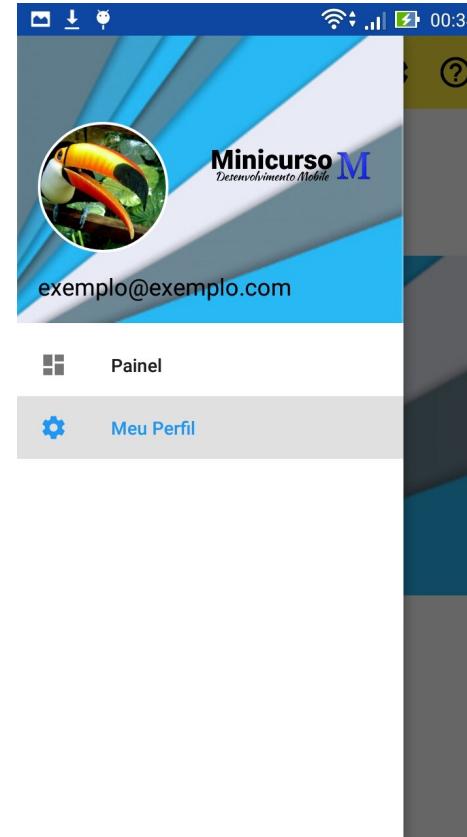
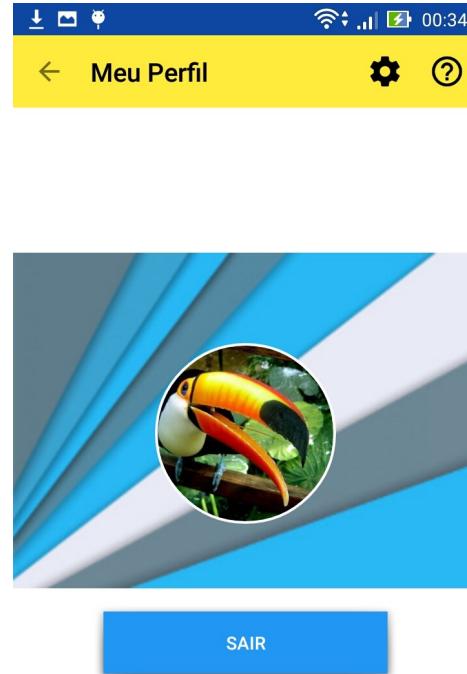
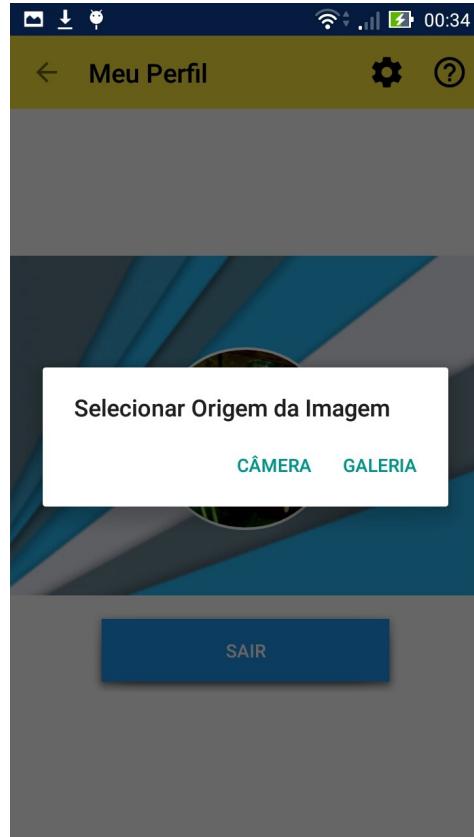


# Resultado Final



# Acesso à Câmera ou Galeria

- Vamos personalizar a imagem do usuário a partir de imagens da câmera ou galeria do android.



# Adicionando Permissões

- Para que nosso aplicativo possa acessar o sistemas de arquivos para salvar e ler a nova imagem de perfil.
- Ou para acessar a câmera do celular.
- É necessário que o usuário permita o acesso a esses recurso no momento da instalação.
- O arquivo AndroidManifest.xml deve conter as permissões que o aplicativo necessita.



# Adicionando Permissões

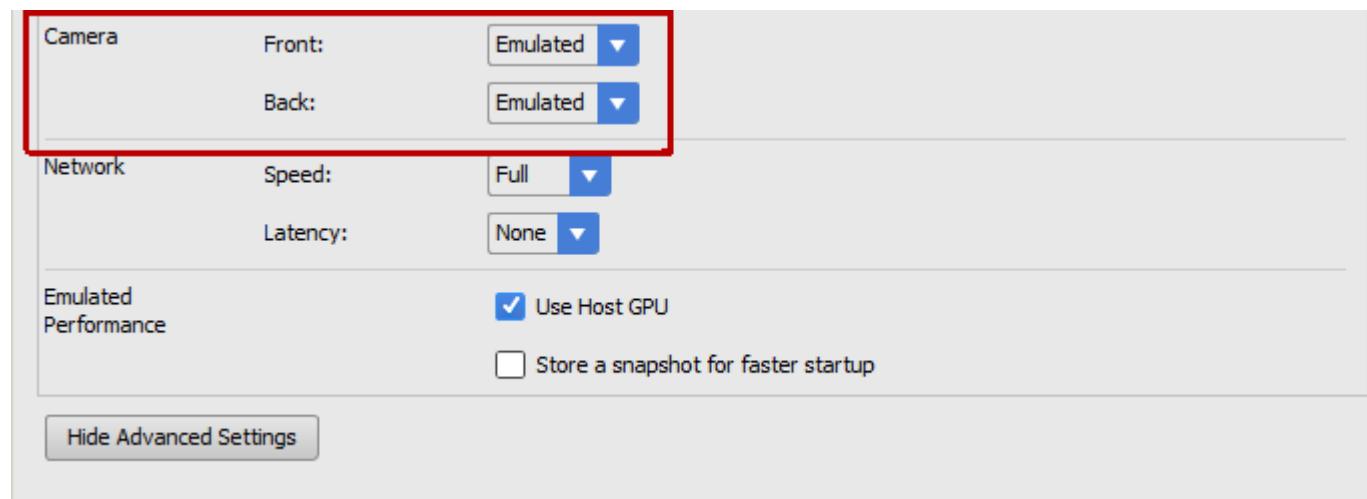
- Adicione ao arquivo AndroidManifest.xml após a tag <uses-sdk>:

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



# Modificando Virtual Device

- Adicione uma câmera virtual ao emulador:



# Atualizando Banco de Dados

- Adicione ao arquivo SQLiteOpenHelperMinicurso os métodos:

```
public boolean UpdateImgProfileUsuario( String Usuario, String imageString)
{
    String codPessoa, tabela, campo, valor;
    tabela = "tbusuario";
    String[] retornoUsuario = {"cdpessoa"};
    campo = "email";
    valor = Usuario;
    codPessoa = getData(tabela, retornoUsuario, campo, valor);

    ContentValues valuesPessoa = new ContentValues();
    valuesPessoa.put("imgpessoa", imageString);
    SQLiteDatabase dbwritePessoa = this.getWritableDatabase();
    dbwritePessoa.update("tbpessoa", valuesPessoa, "cdpessoa " + " = ? ",
        new String[]{codPessoa});
    dbwritePessoa.close();
    return true;
}
```



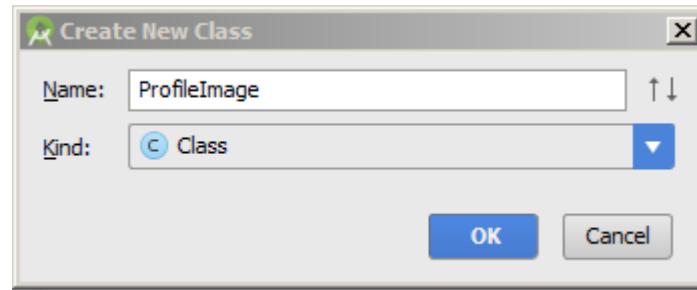
# Atualizando Banco de Dados

```
public String getDrawableProfileImageUsuario(String Email)
{
    SQLiteDatabase db = this.getReadableDatabase();
    final String SQL_QUERY = "SELECT imgpessoa FROM tbpessoa P INNER JOIN
tbusuario U " +
        " ON P.cdpessoa = U.cdpessoa and email = '" + Email + "'";
    Cursor c = db.rawQuery(SQL_QUERY, null);
    String imgPessoa = null;
    if (c.moveToFirst())
    {
        do
        {
            c.moveToFirst();
            imgPessoa = c.getString(c.getColumnIndex("imgpessoa"));
        }
        while (c.moveToNext());
    }
    c.close();
    db.close();
    return imgPessoa;
}
```



# ProfileImage

- A classe ProfileImage sera utiliza para abstrair os métodos responsáveis por salvar no sistema de arquivos a imagem proveniente da câmera ou galeria.
- Crie a classe ProfileImage:



# ProfileImage

- Adicione o seguinte conteúdo:

```
/*
Copyright (c) 2015 Luiz Carlos <luiz04nl@gmail.com>
*/
package br.com.minicurso;

import android.content.*;
import android.graphics.*;
import android.graphics.drawable.*;
import android.net.*;
import android.os.*;
import android.provider.*;
import android.util.*;
import java.io.*;
```



# ProfileImage

```
public class ProfileImage
{
    private static final String LOG_TAG = "LOG_TAG";
    protected static Drawable profileImage;
    protected String imageString;
    public boolean isExternalStorageWritable()
    {
        String state = Environment.getExternalStorageState();
        if (Environment.MEDIA_MOUNTED.equals(state))
        {
            return true;
        }
        return false;
    }
    public boolean isExternalStorageReadable()
    {
        String state = Environment.getExternalStorageState();
        if (Environment.MEDIA_MOUNTED.equals(state) ||
Environment.MEDIA_MOUNTED_READ_ONLY.equals(state))
        {
            return true;
        }
        return false;
    }
}
```



# ProfileImage

```
public File getAlbumStorageDir(String albumName)
{
    File file = new
File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES),
albumName);
    if (!file.mkdirs())
    {
        Log.e(LOG_TAG, "Directory not created");
    }
    return file;
}
public Uri getUriProfileImage(Intent data)
{
    Uri selectedImageUri = data.getData();
    return selectedImageUri;
}
public Bitmap getBitmapProfileImage(Intent data)
{
    Bundle extras = data.getExtras();
    Bitmap imageBitmap = (Bitmap) extras.get("data");
    return imageBitmap;
}
public Drawable getDrawableProfileImage()
{
    return profileImage;
}
```



# ProfileImage

```
public void setProfileImage(Drawable profileImage)
{
    this.profileImage = profileImage;
}
public Intent getIntentGaleria()
{
    Intent intent = new Intent();
    intent.setType("image/*");
    intent.setAction(Intent.ACTION_GET_CONTENT);
    return intent;
}
public Intent getIntentCamera()
{
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    return intent;
}
public void storeProfileImage(Drawable drawable, Context context, String user)
{
    setProfileImage(drawable);
    if (getDrawableProfileImage() != null)
    {
        if (isExternalStorageWritable())
        {
            this.imageString = "ic_profile_image_" + user + ".png";
            File fileOut = new File(getAlbumStorageDir("Minicurso"),
this.imageString );
```



# ProfileImage

```
Drawable drawableImgPessoa = getDrawableProfileImage();
Bitmap bm = Bitmap.createBitmap(140, 140, Bitmap.Config.ARGB_8888);
Canvas canvas = new Canvas(bm);
drawableImgPessoa.setBounds(0, 0, 140, 140);
drawableImgPessoa.draw(canvas);
FileOutputStream outStream = null;
try
{
    outStream = new FileOutputStream(fileOut);
}
catch (FileNotFoundException e)
{
    e.printStackTrace();
}
```



# ProfileImage

```
try
{
    outStream.flush();
    outStream.close();
}
catch (IOException e)
{
    e.printStackTrace();
}
SQLiteOpenHelperMinicurso db = new SQLiteOpenHelperMinicurso(context);
db.UpdateImgProfileUsuario(user, this.imageString);
}
```



# Strings

- Adicione as novas strings:

```
<string name="selectPicture">Selecionar Origem da Imagem</string>
<string name="gallery">Galeria</string>
<string name="camera">Câmera</string>
```



# Obter Imagem

- Modifique o FragmentMyProfile para obter a imagem da câmera ou galeria:
- Novas importações e atributos:

```
import android.graphics.drawable.*;  
  
protected android.support.v7.app.AlertDialog alertDialog;  
protected de.hdodenhof.circleimageview.CircleImageView profile_image_myprofile;
```

- No método onCreateView() adicione:

```
this.profile_image_myprofile = (de.hdodenhof.circleimageview.CircleImageView)  
rootView.findViewById(R.id.profile_image_myprofile);  
ProfileImage profileImage = new ProfileImage();  
  
if (profileImage.getDrawableProfileImage() != null)  
{  
    Drawable drawableImgPessoa = profileImage.getDrawableProfileImage();  
    this.profile_image_myprofile.setImageDrawable(drawableImgPessoa);  
}
```



# Obter Imagem

```
final android.support.v7.app.AlertDialog.Builder builder = new
android.support.v7.app.AlertDialog.Builder(getActivity());
this.profile_image_myprofile.setOnClickListener(new View.OnClickListener()
{
    public void onClick(View v)
    {
        builder.setTitle(R.string.selectPicture);
        builder.setPositiveButton(R.string.gallery, new
DialogInterface.OnClickListener()
        {
            public void onClick(DialogInterface arg0, int arg1)
            {
                ((NavigationActivity) getActivity()).getIntentGaleria();
            }
        });
        builder.setNegativeButton(R.string.camera, new
DialogInterface.OnClickListener()
        {
            public void onClick(DialogInterface arg0, int arg1)
            {
                ((NavigationActivity) getActivity()).getIntentCamera();
            }
        });
        alertDialog = builder.create();
        alertDialog.show();
    }
});
```



# NavigationActivity

- Por Ultimo modifique a classe NavigationActivity.
- Novas importações:

```
import java.io.*;
import android.net.*;
import android.graphics.*;
```

- Novos atributos:

```
public static final int SELECT_PICTURE = 1;
public static final int REQUEST_IMAGE_CAPTURE = 2;
protected de.hdodenhof.circleimageview.CircleImageView profile_image;
protected String stringImgPessoa;
protected String imageString;
```



# NavigationActivity

- Adicione ao método onCreate():

```
final ProfileImage profileImage = new ProfileImage();
this.profile_image = (de.hdodenhof.circleimageview.CircleImageView)
findViewById(R.id.profile_image);
final SQLiteOpenHelperMinicurso db = new SQLiteOpenHelperMinicurso(this);
if ( db.getDrawableProfileImageUsuario(this.userName) != null )
{
    this.stringImgPessoa = db.getDrawableProfileImageUsuario(this.userName);
    File file = profileImage.getAlbumStorageDir("Minicurso");
    if ( profileImage.isExternalStorageReadable() )
    {
        this.imageString = "ic_profile_image_" + userName + ".png";
        String pathName = file.getPath() + "/" + this.imageString;
        Drawable drawableImgPessoa = Drawable.createFromPath(pathName);
        profileImage.setProfileImage(drawableImgPessoa);
        this.profile_image.setImageDrawable(drawableImgPessoa);
    }
}
```



# NavigationActivity

- Adicione os novos métodos a classe:

```
public void getIntentGaleria()
{
    ProfileImage profileImage = new ProfileImage();
    Intent intent = profileImage.getIntentGaleria();
    startActivityForResult(Intent.createChooser(intent,
getResources().getString(R.string.selectPicture)), SELECT_PICTURE);
}

public void getIntentCamera()
{
    ProfileImage profileImage = new ProfileImage();
    Intent intent = profileImage.getIntentCamera();
    startActivityForResult(intent, REQUEST_IMAGE_CAPTURE);
}
```



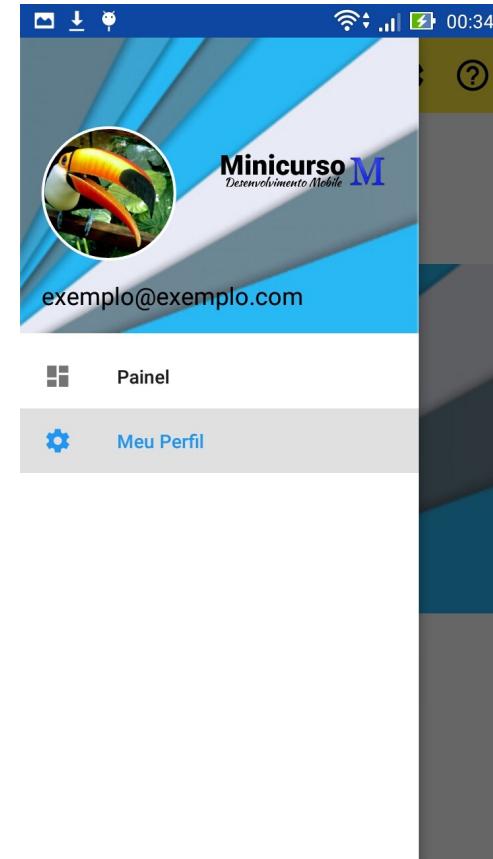
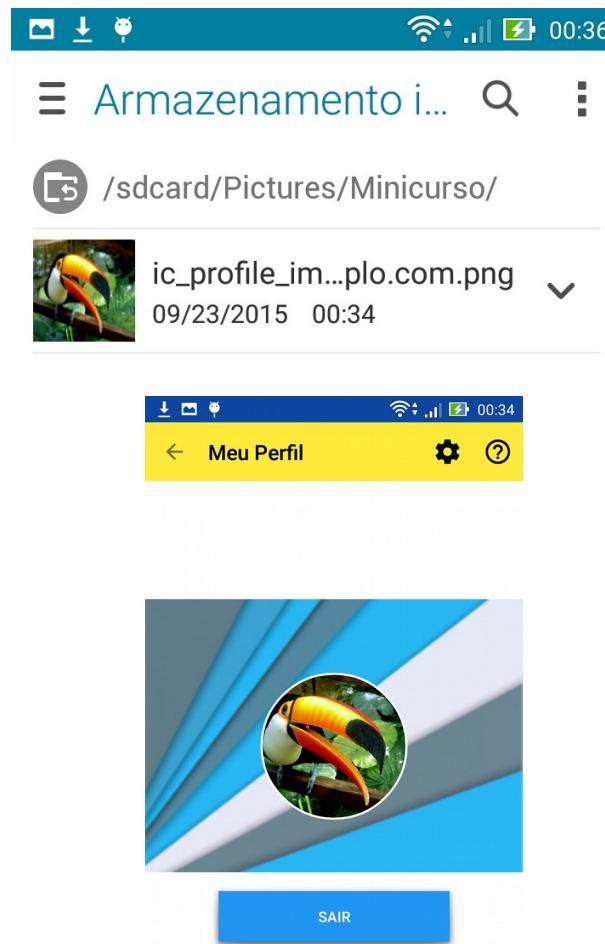
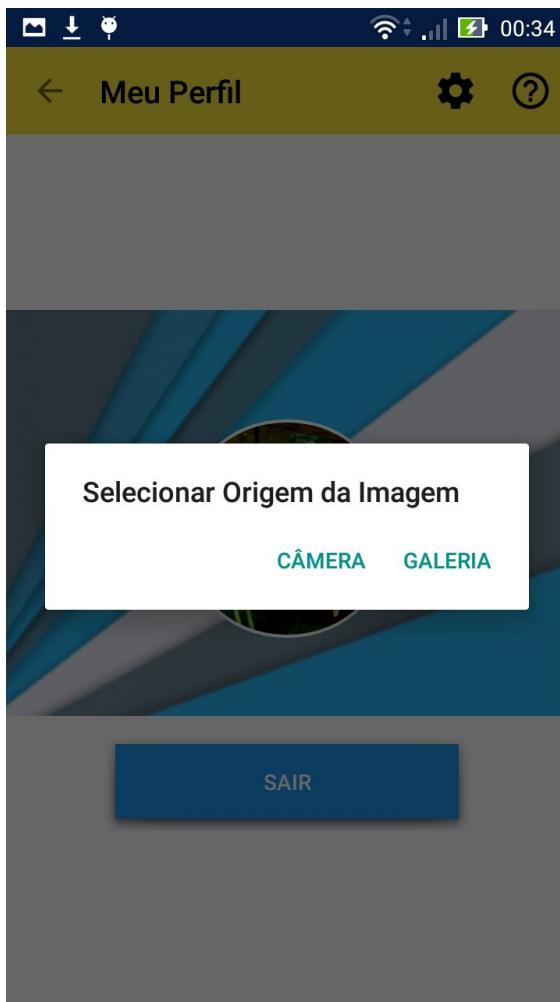
# NavigationActivity

- Adicione os novos métodos a classe:

```
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    this.profile_image = (de.hdodenhof.circleimageview.CircleImageView)
findViewById(R.id.profile_image);
    ProfileImage profileImage = new ProfileImage();
    if (requestCode == SELECT_PICTURE && resultCode == RESULT_OK)
    {
        Uri selectedImageUri = profileImage.getUriProfileImage(data);
        this.profile_image.setImageURI(selectedImageUri);
    }
    else if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK)
    {
        Bitmap imageBitmap = profileImage.getBitmapProfileImage(data);
        this.profile_image.setImageBitmap(imageBitmap);
    }
    Drawable drawable = this.profile_image.getDrawable();
    profileImage.storeProfileImage(drawable, this, this.userName);
    setFragmentManager();
}
```



# Resultado Final



# Código Fonte do Minicurso

- <https://github.com/luiz04nl/Minicurso-Android.git>
- Licença Apache v2



# Referências Bibliográficas

- Proxxima. <http://www.proxxima.com.br/>. Acesso em: 27 ago. 2015.
- Tudocelular. <http://www.tudocelular.com/>. Acesso em: 27 ago. 2015.
- Android . <https://www.android.com/auto/>. Acesso em: 27 ago. 2015.
- Startipi. <http://startipi.com.br/>. Acesso em: 27 ago. 2015.
- Nextecommerce. <http://nextecommerce.com.br/>. Acesso em: 27 ago. 2015.
- Codename One. <http://www.codenameone.comr/>. Acesso em: 28 ago. 2015.
- Devmedia . <http://www.devmedia.com.br/>. Acesso em: 28 ago. 2015.
- Tecmundo. <http://www.tecmundo.com.br/>. Acesso em: 31 ago. 2015.
- Thiengo. <http://www.thiengo.com.br/>. Acesso em: 8 set. 2015.

