

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL-RIO-GRANDENSE
Campus Passo Fundo



Introdução à Plataforma Android

Guilherme A. Borges
Anubis G. M. Rossetto

Sumário

- ▶ Referências/Fontes
- ▶ Introdução
- ▶ Dicas
- ▶ Configuração Ambiente de Desenvolvimento
- ▶ Criando aplicações
- ▶ Exercícios

Referências

Referências

- ▶ <https://sites.google.com/site/rossettopf/pdm-6m1>
- ▶ <http://www.slideshare.net/joseberardo>
- ▶ <http://www.k19.com.br/downloads/apostilas/java/k19-k41-desenvolvimento-mobile-com-android>
- ▶ <http://developer.android.com/guide>

Fontes

- ▶ Códigos Fonte da Apresentação:

- ▶ <https://docs.google.com/file/d/0B746pjy4jQVAdC1ldkc5VDBlZWw/edit?usp=sharing>

- ▶ Lista de Exercícios:

- ▶ <https://docs.google.com/document/d/1VkuLJKioBuKbl8ptTLAvZtRjV63e03--9c3wjDZyBFU/edit?usp=sharing>

- ▶ Slides Apresentação Conceitual/Implementação

- ▶ Conceitual:

- ▶ <https://docs.google.com/file/d/0B746pjy4jQVAS0tjNzQzTS1VT2c/edit?usp=sharing>

- ▶ Implementação

Introdução

Plataforma Android

▶ Android

- ▶ Sistema operacional móvel que roda sobre o núcleo Linux
 - ▶ Projeto inicial da Google
 - ▶ Depois Open Handset Alliance (OHA)
 - ▶ Open Source
-
- ▶ SDK possui um conjunto de bibliotecas e API's de simples acesso e fácil entendimento.

Plataforma Android

▶ Open Handset Alliance (OHA)

- ▶ Aliança entre as empresas: Google, HTC, Dell, Intel, Motorola, Qualcomm, Texas Instruments, Samsung, LG, T-Mobile e Nvidia.
- ▶ Objetivo de Criar padrões abertos para telefonia móvel.
- ▶ Plataforma open source Android – SDK (Nov, 2007).
- ▶ <http://www.openhandsetalliance.com>

SDK Manager

- ▶ É necessário utilizar o Android SDK, pois ele possui:
 - ▶ Bibliotecas da API Android
 - ▶ Ferramentas necessárias para criar, testar e depurar apps para Android
 - ▶ Criação de Emuladores
- ▶ Link:
 - ▶ <http://developer.android.com/sdk/index.html>

Versões (Curiosidade)

- Nomes das versões do Android são doces:



Cupcake
Android 1.5



Donut
Android 1.6



Eclair
Android 2.0/2.1



Froyo
Android 2.2



Gingerbread
Android 2.3



Honeycomb
Android 3.0



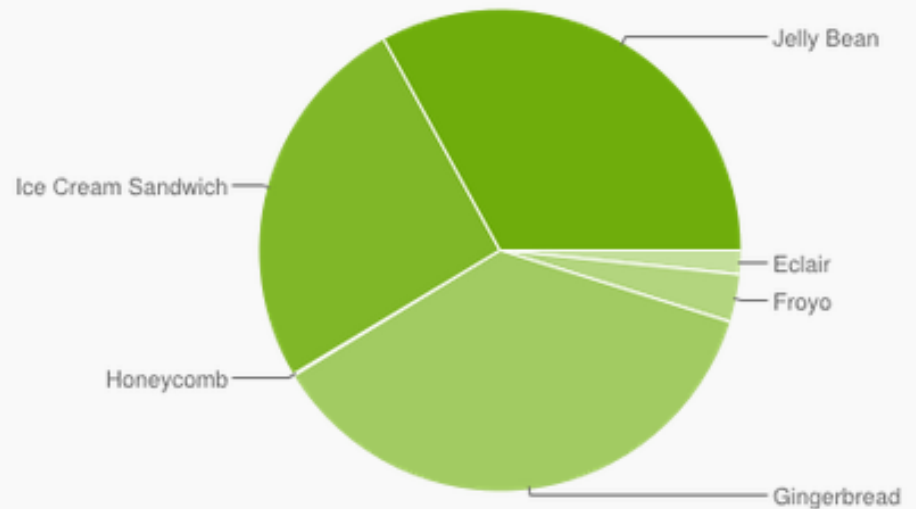
Ice Cream Sandwich
Android 4.0



Jelly Bean
Android 4.1

Versões

Version	Codename	API	Distribution
1.6	Donut	4	0.1%
2.1	Eclair	7	1.5%
2.2	Froyo	8	3.2%
2.3 - 2.3.2	Gingerbread	9	0.1%
2.3.3 - 2.3.7		10	36.4%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	25.6%
4.1.x	Jelly Bean	16	29.0%
4.2.x		17	4.0%



Data collected during a 14-day period ending on June 3, 2013.

► <http://developer.android.com/about/dashboards/index.html>

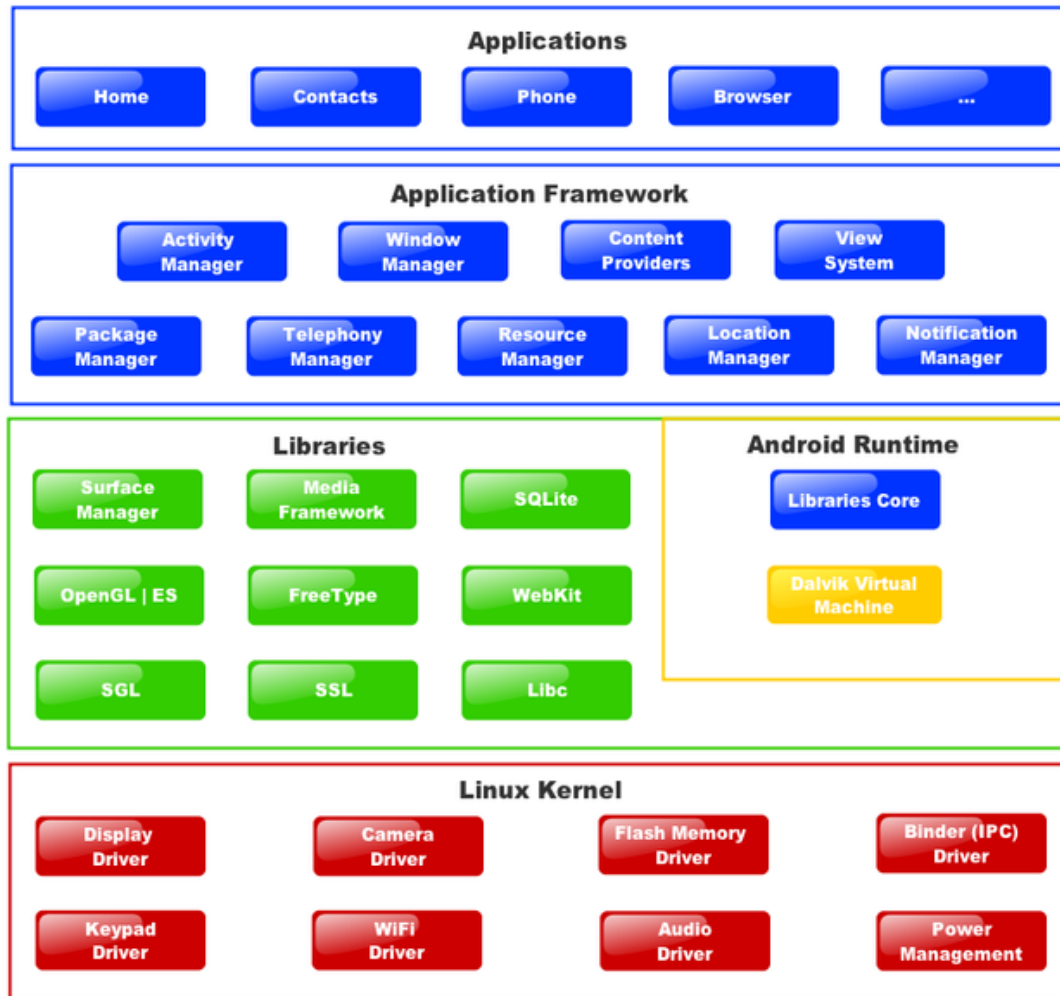
API Level

- ▶ Cada versão do Android possui uma API Level
 - ▶ É importante para referenciar as versões do Android
- ▶ A API Level é um valor inteiro que identifica uma versão do Android.
- ▶ A plataforma Android fornece uma estrutura de API que os aplicativos podem usar para interagir com o Sistema Android

A estrutura API consiste em:

- ▶ Conjunto de pacotes e classes
- ▶ Conjunto de elementos e atributos XML para declarar um manifest file (arquivo de configuração da aplicação no Android)
- ▶ Conjunto de elementos e atributos XML para declarar e acessar recursos
- ▶ Conjunto de intenções
- ▶ Conjunto de permissões que os aplicativos podem solicitar, bem como a autorização incluída no sistema
- ▶ Cada versão sucessiva da plataforma Android pode incluir atualizações e com a API Level é possível identificar sua utilização no mesmo

Arquitetura



Maquina Virtual

- ▶ Aplicações escritas em Java são compiladas em bytecodes Dalvik e executadas usando a Máquina Virtual Dalvik.
- ▶ Máquina Virtual Dalvik
 - ▶ Máquina virtual especializada desenvolvida para uso em dispositivos móveis
 - ▶ Baseada em registradores
 - ▶ Otimizada para utilizar pouca memória
 - ▶ Permite que múltiplas instâncias da MV rodem ao mesmo tempo, deixando para o SO o isolamento de processos, o gerenciamento de memória e o suporte a threading.

Principaux Classes

- ▶ Activity
- ▶ Intent
- ▶ View

Outras Classes/Funcionalidades

- ▶ BroadcastReceiver
- ▶ Notification
- ▶ Service
- ▶ AlarmManager
- ▶ Handler, Threads e AsyncTask
- ▶ Banco de dados (SQLite)
- ▶ Content Provider
- ▶ Mapas e GPS
- ▶ Http, sockets e Web Services
- ▶ SMS
- ▶ Google Cloud Messaging
- ▶ Reconhecimento de Gestos
- ▶ Sensores
- ▶ Bluetooth

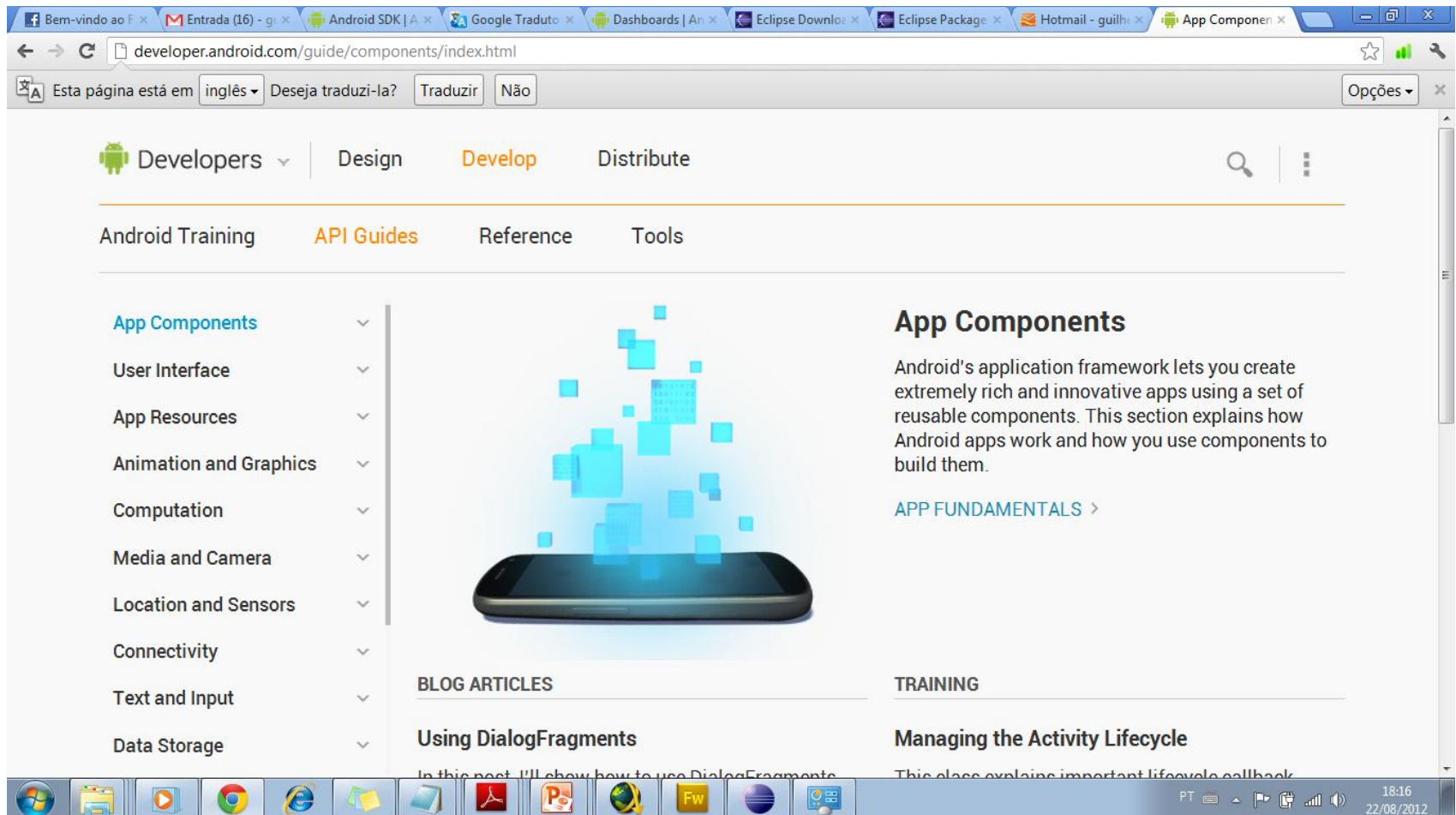




Dicas

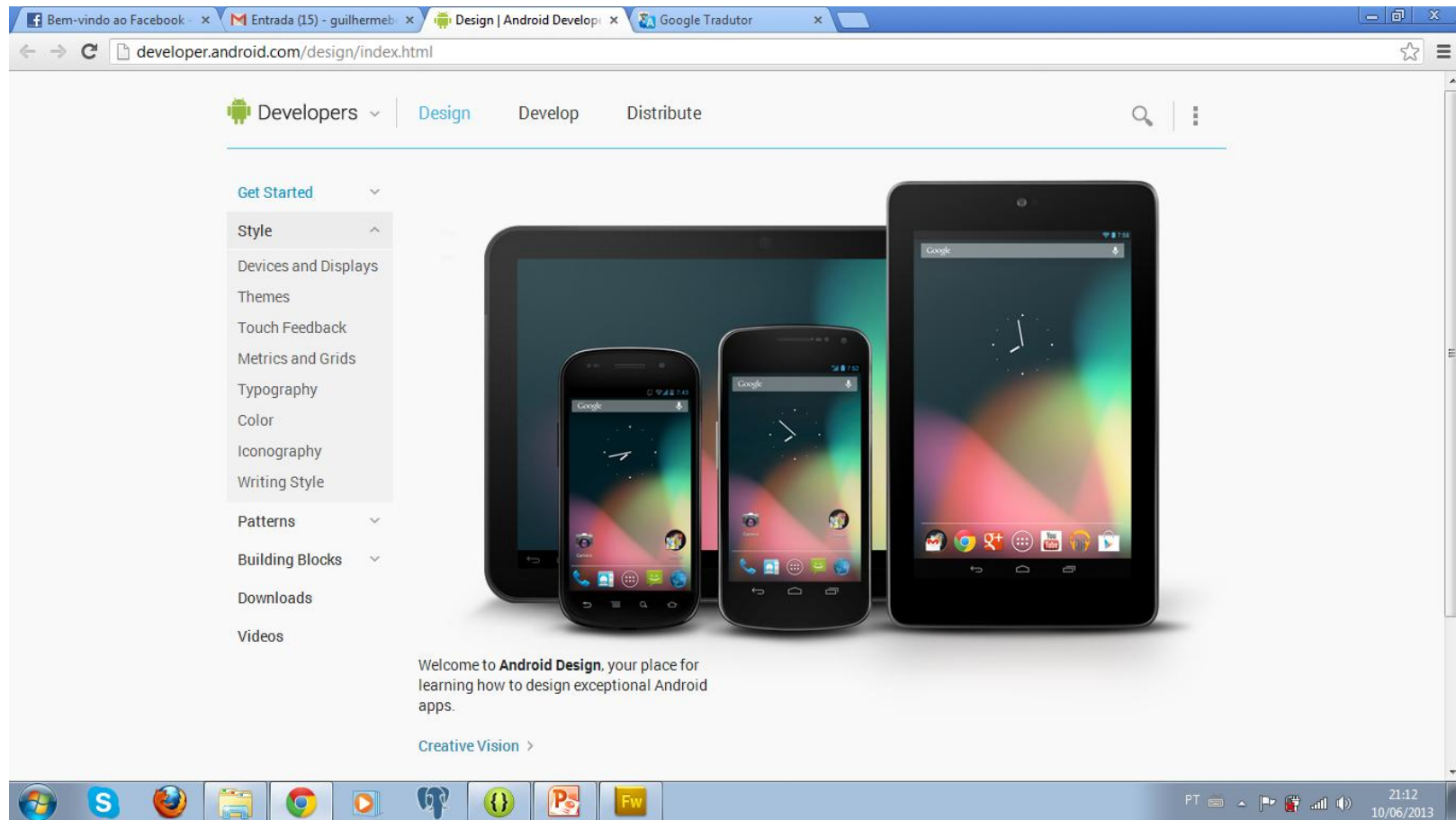
Referência On-line

► <http://developer.android.com/guide/>



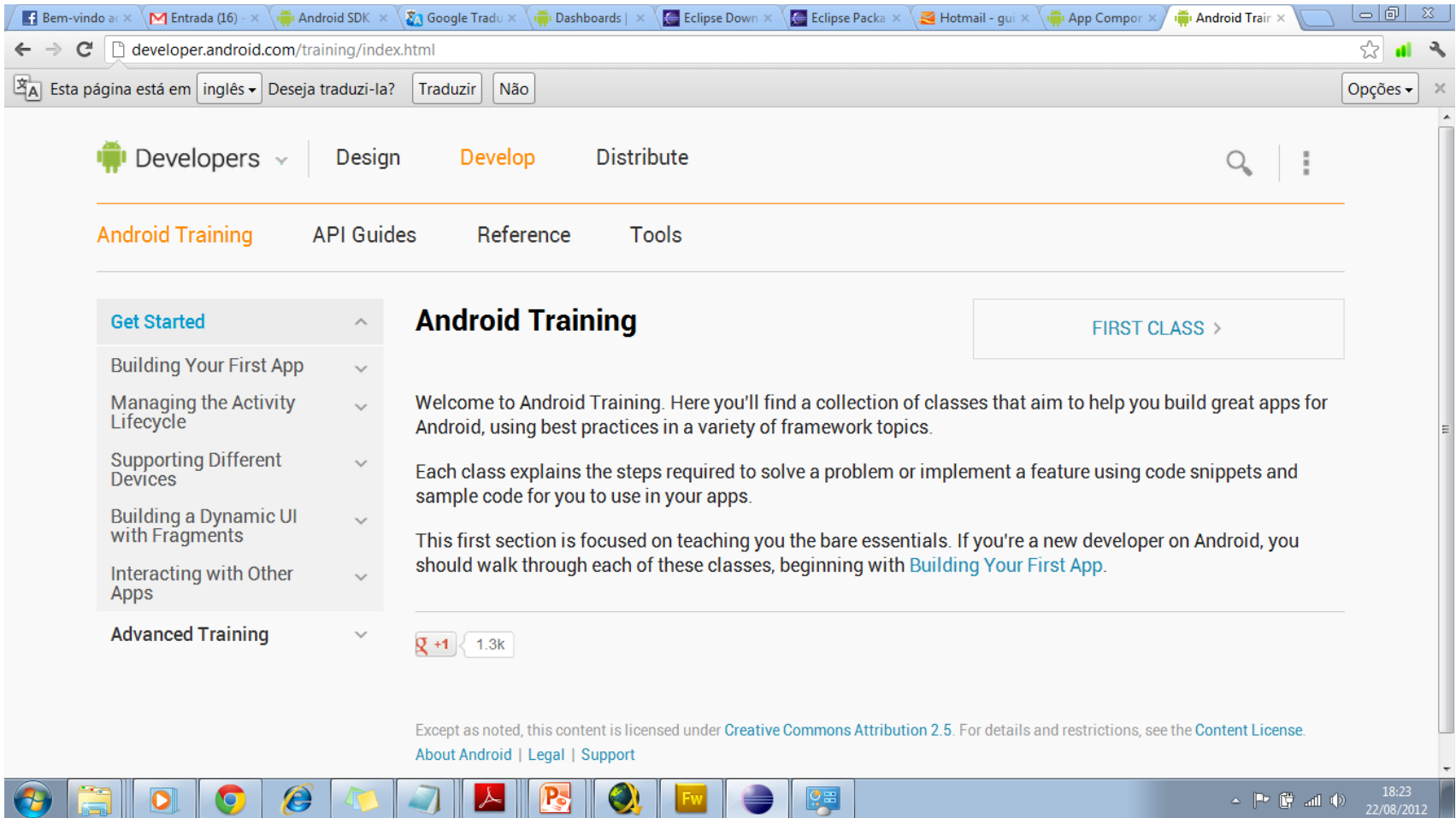
Referência para Estilo

- ▶ <http://developer.android.com/design/index.html>



Tutoriais

► <http://developer.android.com/training/index.html>



The screenshot shows a web browser window displaying the Android Developer Training page. The browser's address bar shows the URL `developer.android.com/training/index.html`. The page features a navigation bar with links for **Design**, **Develop** (highlighted), and **Distribute**. Below this, there are links for **Android Training**, **API Guides**, **Reference**, and **Tools**. The main content area is titled **Android Training** and includes a **FIRST CLASS >** button. The page describes the training classes, stating: "Welcome to Android Training. Here you'll find a collection of classes that aim to help you build great apps for Android, using best practices in a variety of framework topics. Each class explains the steps required to solve a problem or implement a feature using code snippets and sample code for you to use in your apps. This first section is focused on teaching you the bare essentials. If you're a new developer on Android, you should walk through each of these classes, beginning with [Building Your First App](#)." A sidebar on the left lists topics like **Get Started**, **Building Your First App**, **Managing the Activity Lifecycle**, **Supporting Different Devices**, **Building a Dynamic UI with Fragments**, **Interacting with Other Apps**, and **Advanced Training**. At the bottom, there is a Creative Commons Attribution 2.5 license notice and links for **About Android**, **Legal**, and **Support**.

Livros

Google Android para Tablets 2ª Ed. (2012)

<http://www.novateceditora.com.br/livros/android-tablets/>



Google Android 3ª Ed. (2013)

<http://novatec.com.br/livros/googleand3/>

Dicas

- ▶ Deixar o emulador sempre aberto (lento para carregar)
- ▶ Para virar tela
 - ▶ Desligar num lock e usar teclas 7 e 9
 - ▶ CTRL F11 / CTRL F12
- ▶ Para trabalhar com Google Maps
 - ▶ Criar projeto como Google API (não Android....)

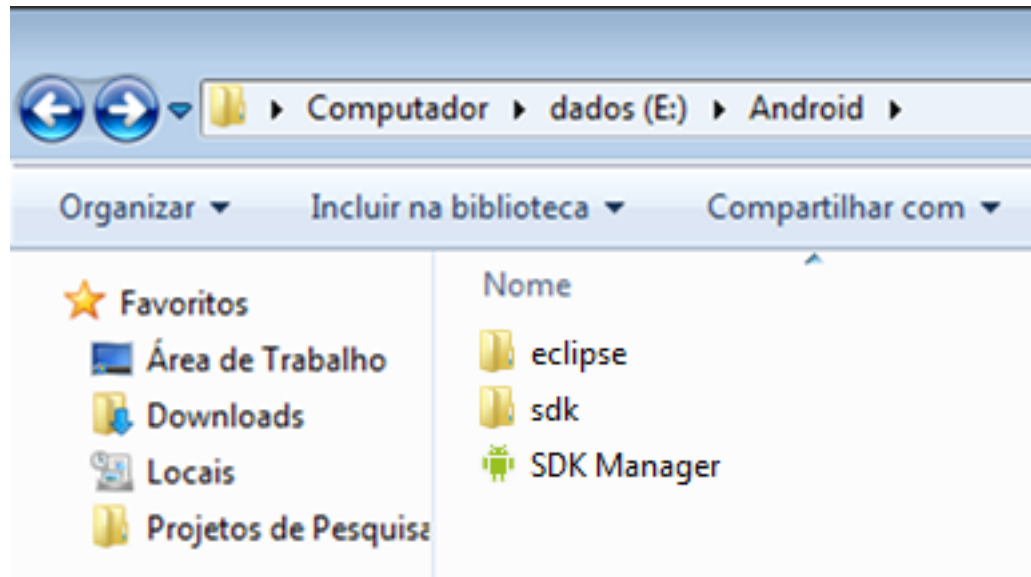
Ambiente de Desenvolvimento

Ambiente de Desenvolvimento

- ▶ SDK (Software Development Kit) do Android
 - ▶ API 17 – Android 4.2.2
 - ▶ API 17 – Google Maps Android 4.2.2 (Baixar)
- ▶ Eclipse IDE
 - ▶ 4.2 – Juno - <http://www.eclipse.org/juno/>
 - ▶ Android Developer Tools - <http://developer.android.com/sdk/index.html>
- ▶ Android Development Tools Plugin (ADT)
 - ▶ Plugin para o Eclipse IDE - Versão atual: 22.0.1
 - ▶ <http://developer.android.com/tools/sdk/eclipse-adt.html>

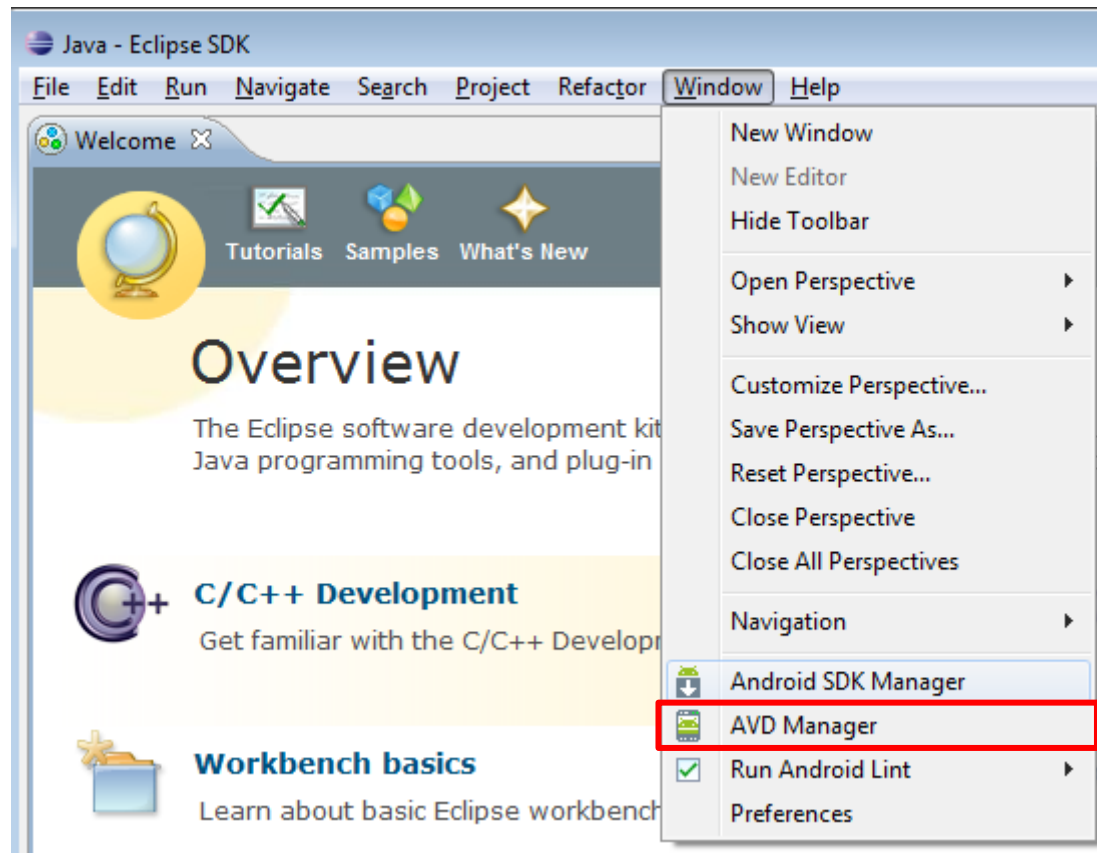
Sugestão (Para Android Developer Tools)

- ▶ Criar pasta C:\android
- ▶ Descompactar o Android Developer Tools

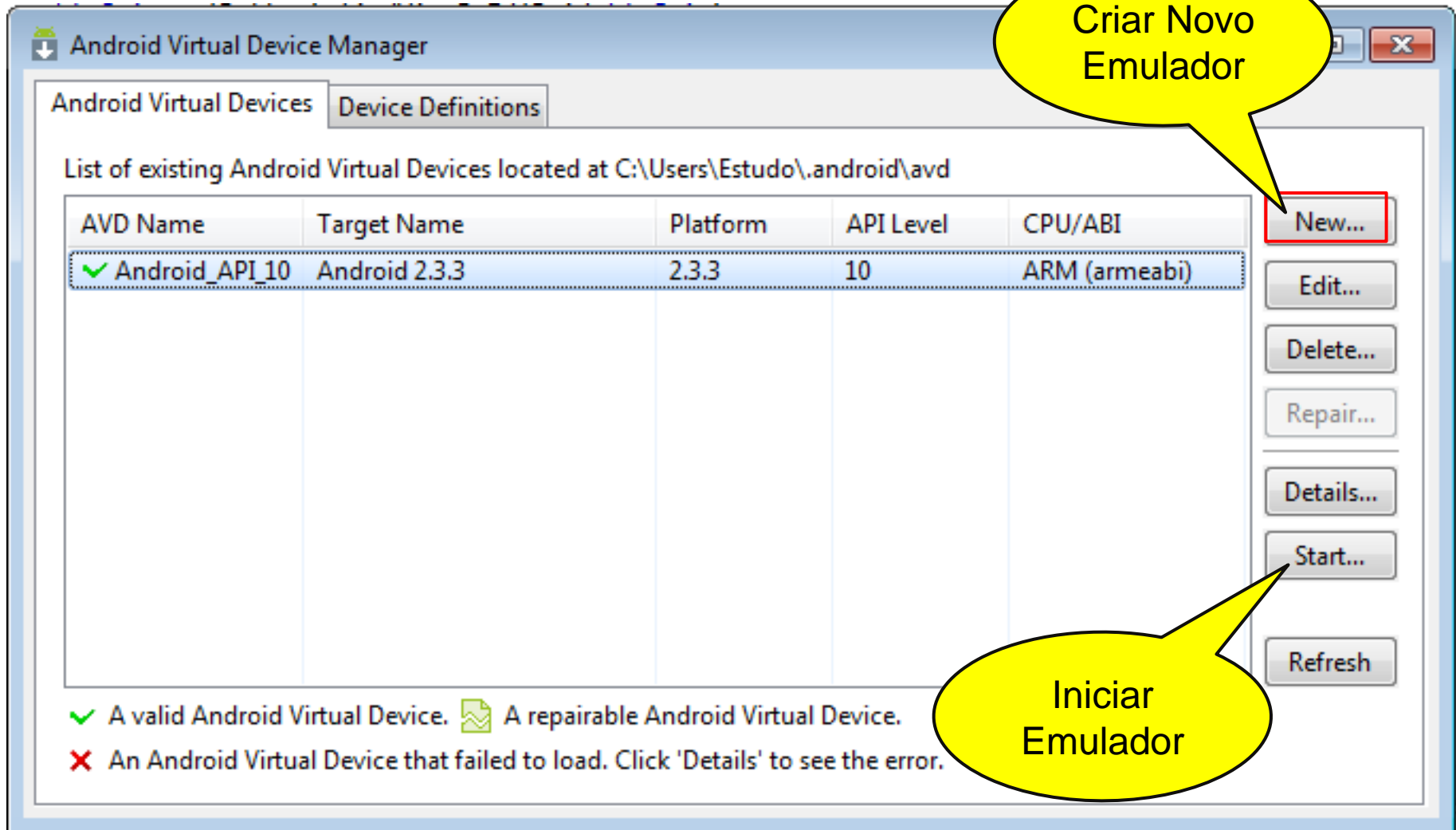


Configuração

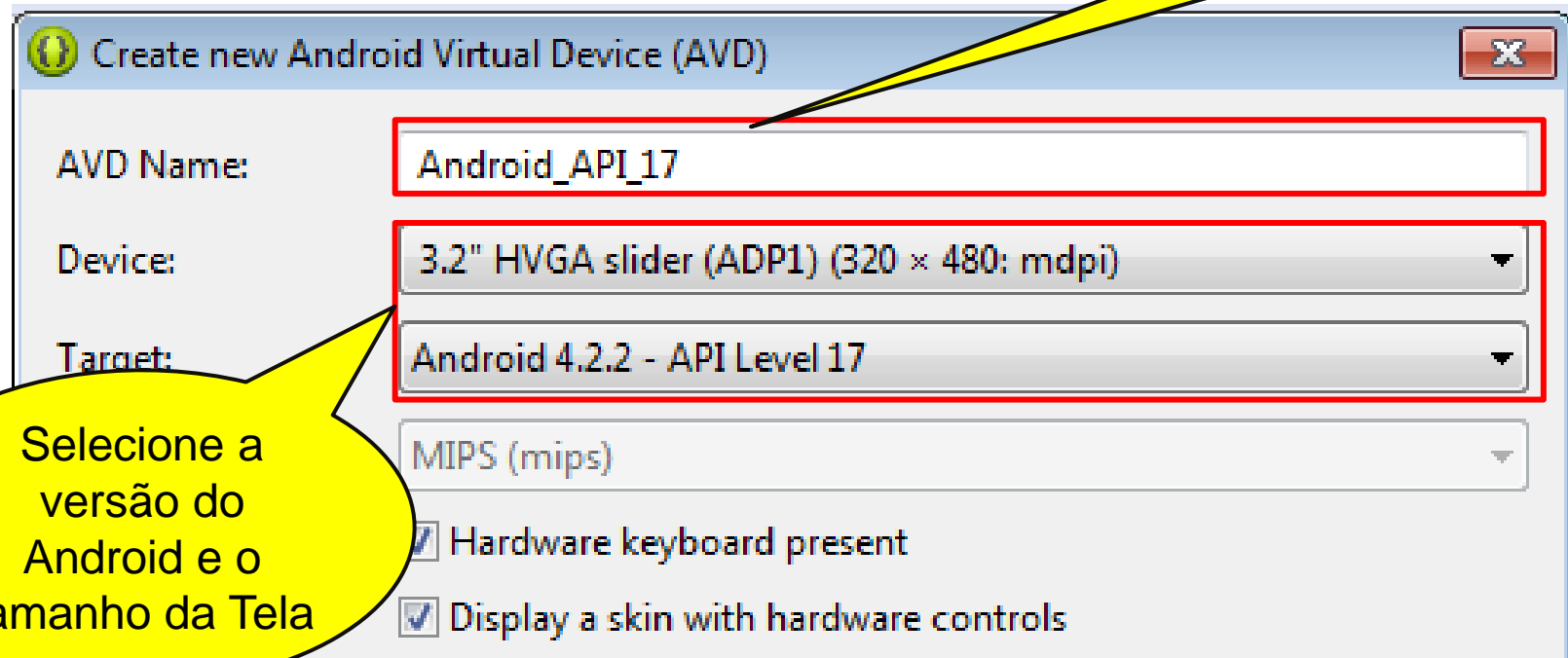
- Depois de Instalar o ADT Plugin, Executar pelo eclipse o Android SDK Manager



Emulador



Emulador



Emulador

Front Camera:

Back Camera:

Memory Options: RAM: VM Heap:

Internal Storage:

SD Card:

☒ Size:

☐ File:

Este recursos
servem para
salvar
informações no
emulador.

Emulador

▶ Android Virtual Devices (AVD)

- ▶ Usado para testar as aplicações
- ▶ Simula um dispositivo real
- ▶ É possível criar várias configurações de AVD para testar as aplicações.



Criando uma aplicação

Resumo

- ▶ Criar uma aplicação
- ▶ Estrutura dos Arquivos
- ▶ Layouts
 - ▶ Elementos/Componentes XML
- ▶ Classe R
 - ▶ Integração XML com Activity
- ▶ Classe Activity
 - ▶ Manipulando componentes da classe View pela Activity
 - ▶ Eventos
- ▶ Classe Intent
 - ▶ Chamadas com e sem retorno

Exemplos

- ▶ Exemplos serão desenvolvidos durante a explicação dos componentes:

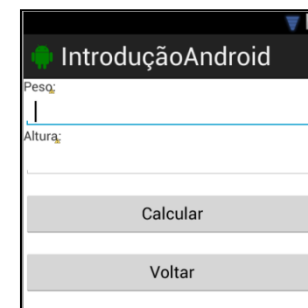
- ▶ Menu



- ▶ Abrir URL de navegador

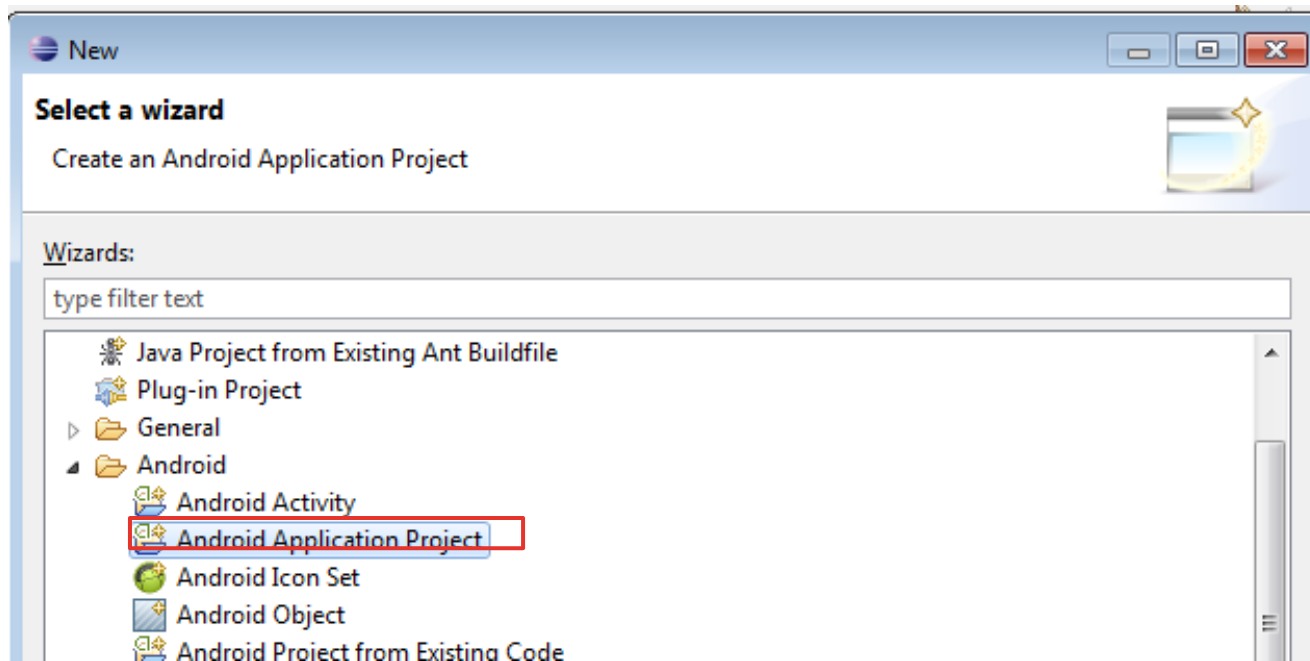


- ▶ Cálculo IMC



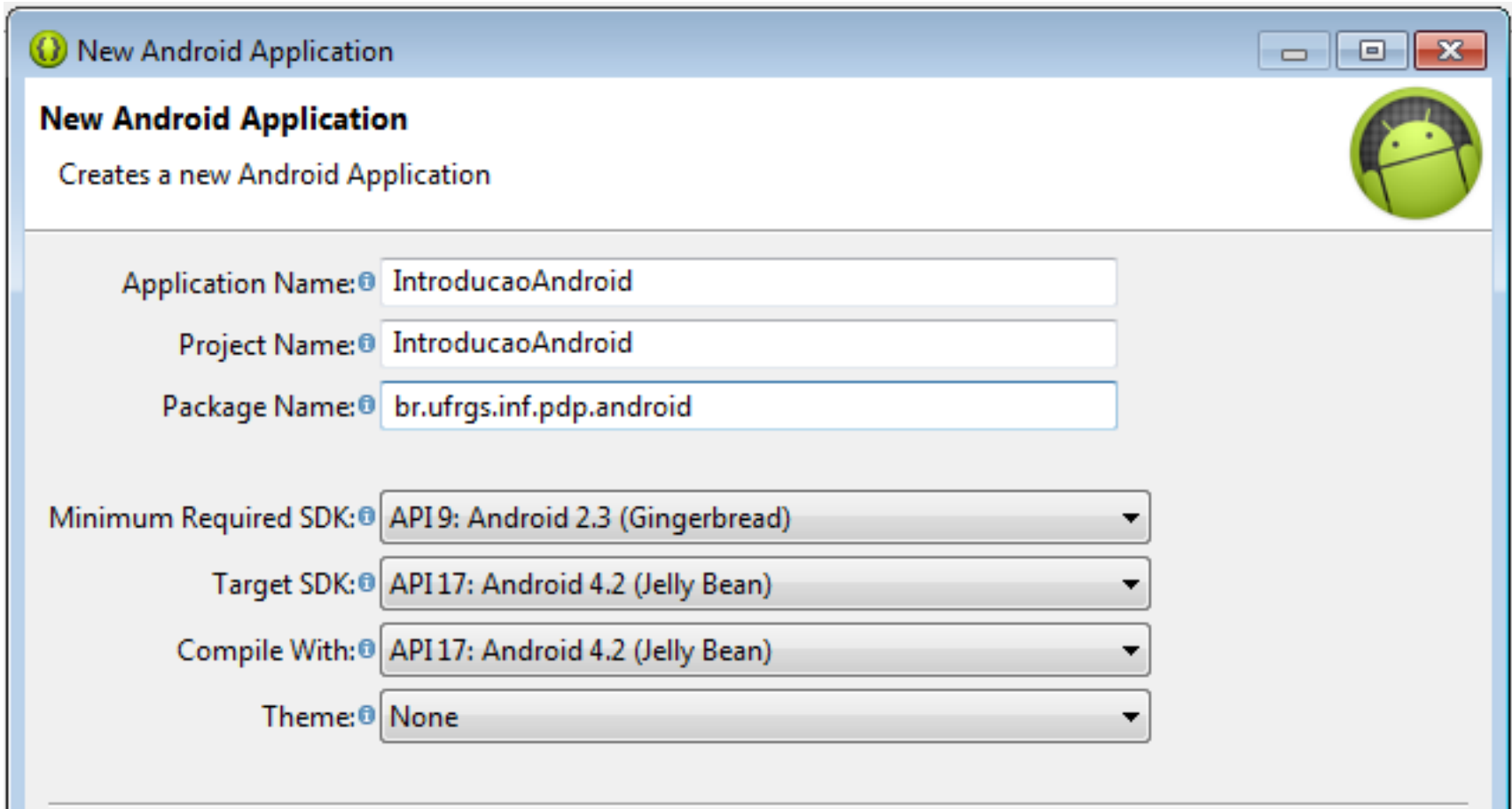
Criando uma aplicação

- ▶ Abrir a IDE eclipse
- ▶ Menu File → New → Other ou Menu File → New
- ▶ Selecionar Android Application Project



Criando uma aplicação

- Preencha as informações e clique em **Next >**.



New Android Application
Creates a new Android Application

Application Name:

Project Name:

Package Name:

Minimum Required SDK:

Target SDK:

Compile With:

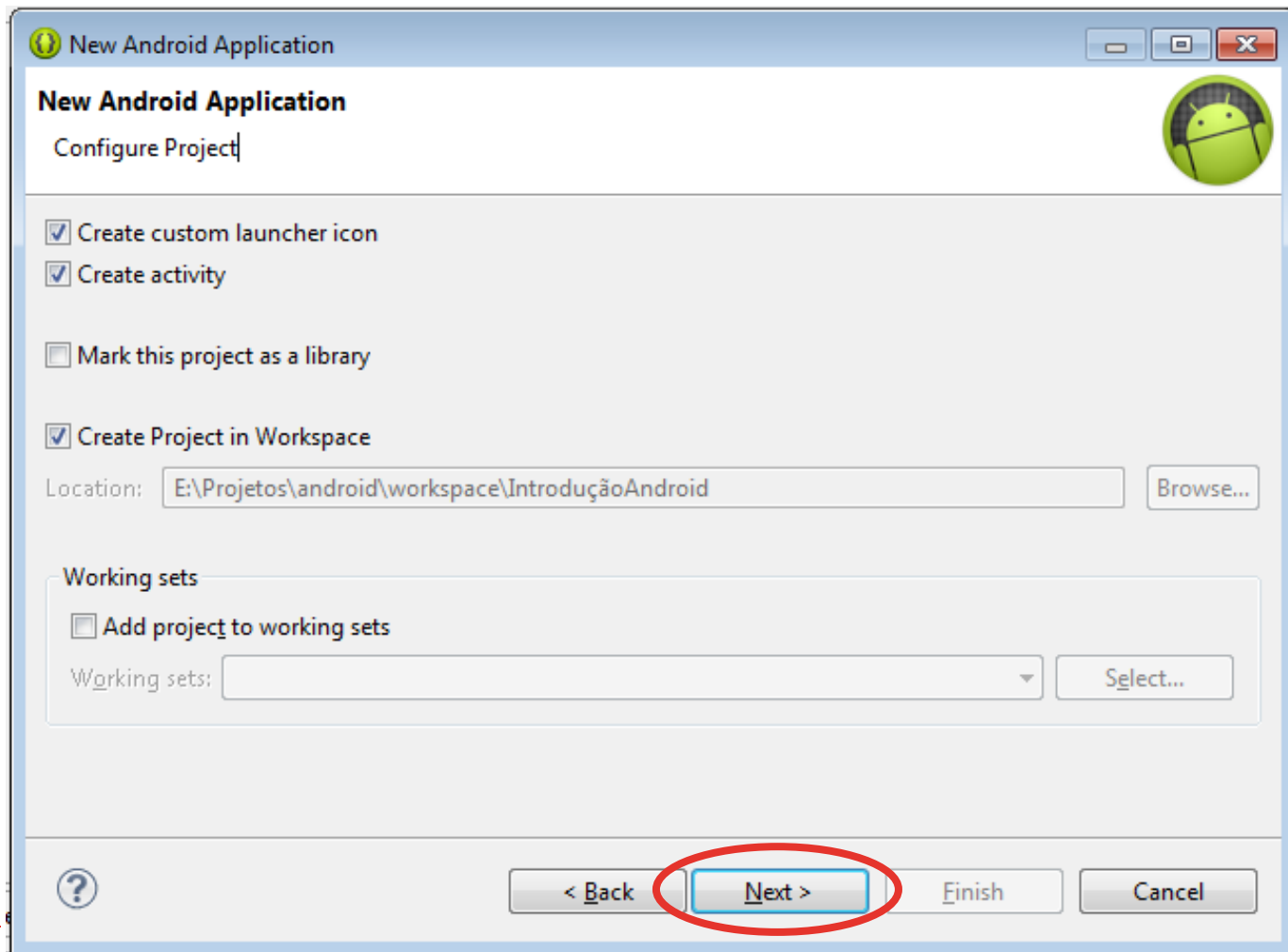
Theme:

Informações da tela de criação

- ▶ Application Name: nome da sua aplicação
- ▶ Project Name: nome do seu projeto
- ▶ Package Name: Nome do seu pacote
- ▶ Minimum Required SDK: Versão mínima requerida pela Aplicação
- ▶ Target SDK: Versão alvo da aplicação

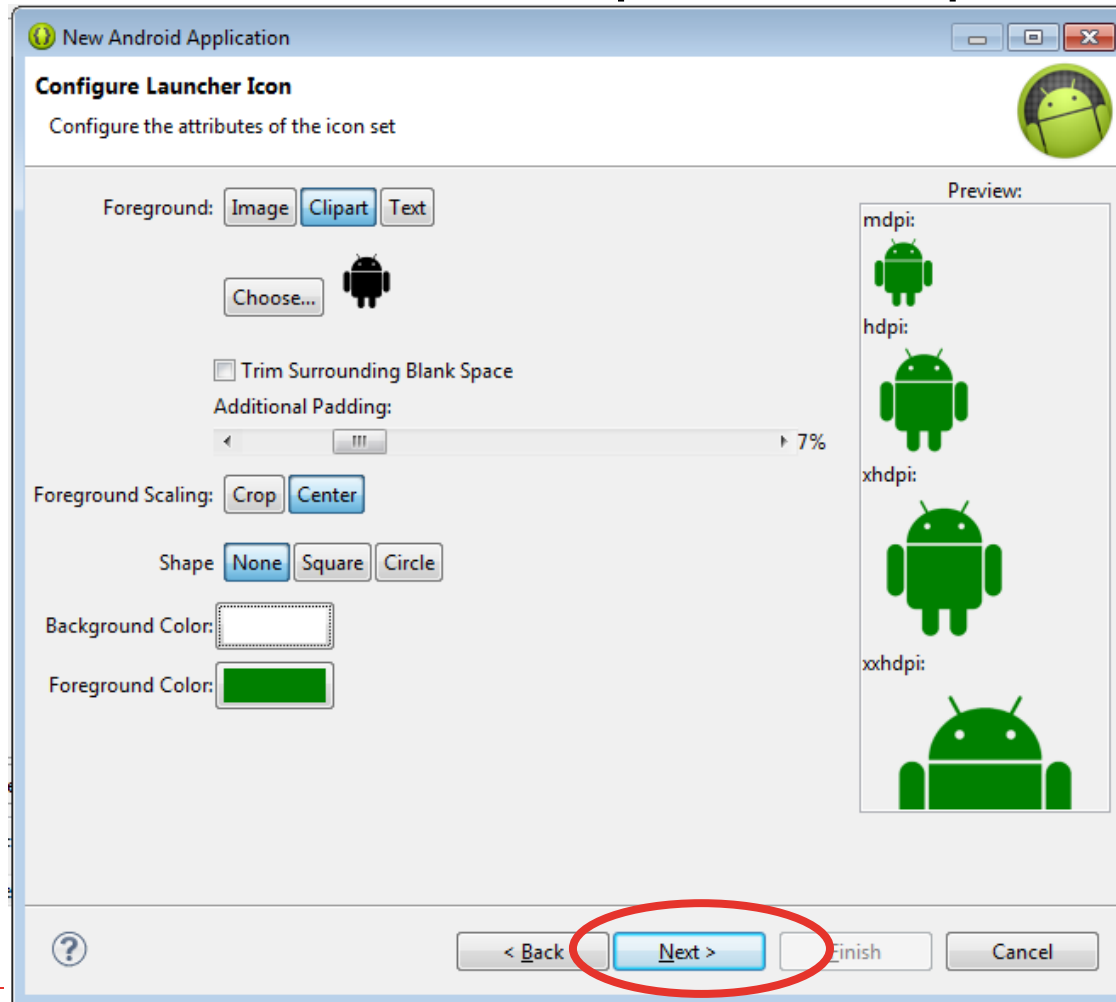
Criando uma Aplicação

- Deixe as informações Default e clique em **Next >**



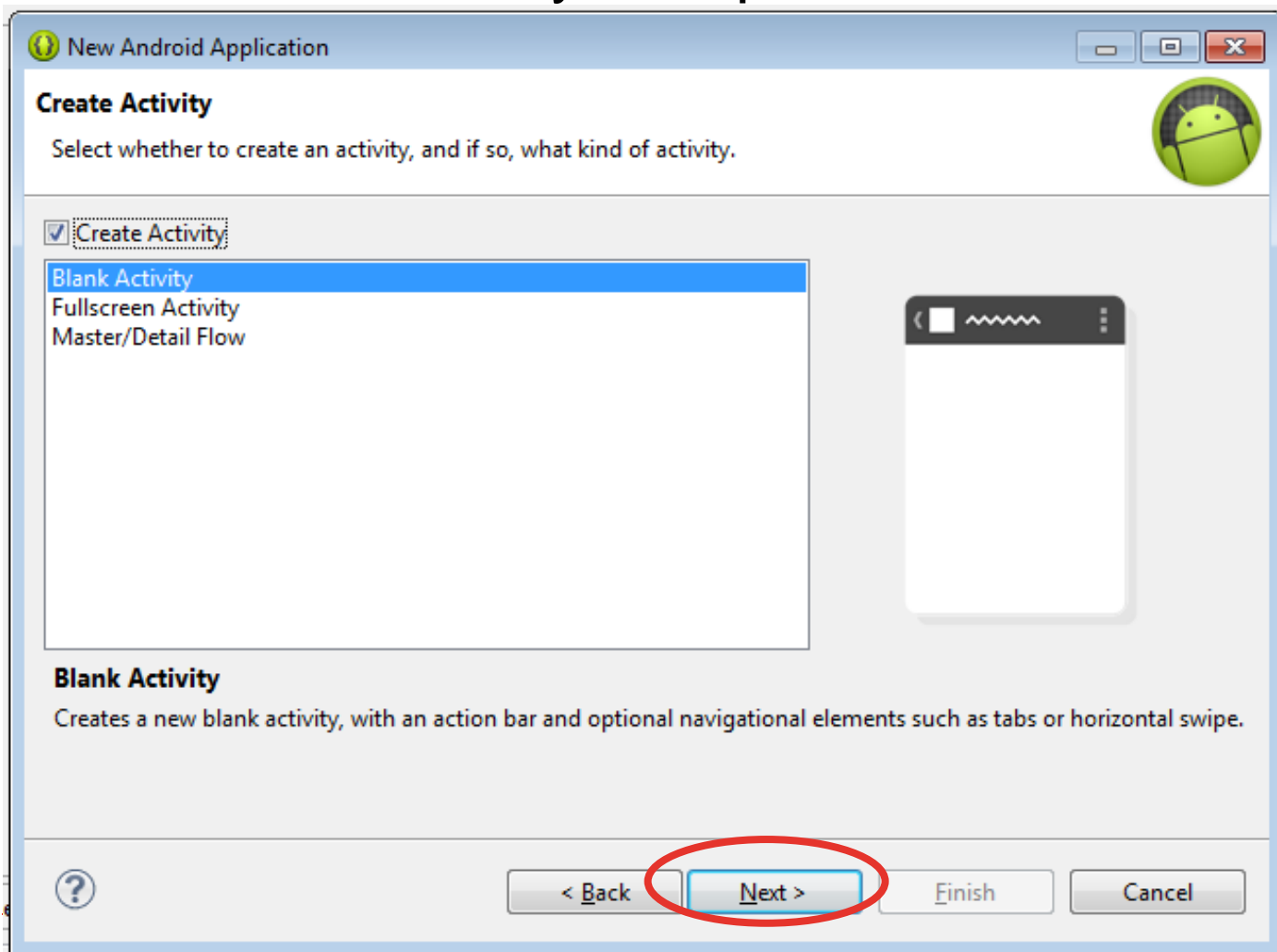
Criando uma Aplicação

- Customize o ícone como quiser e clique em **Next >**



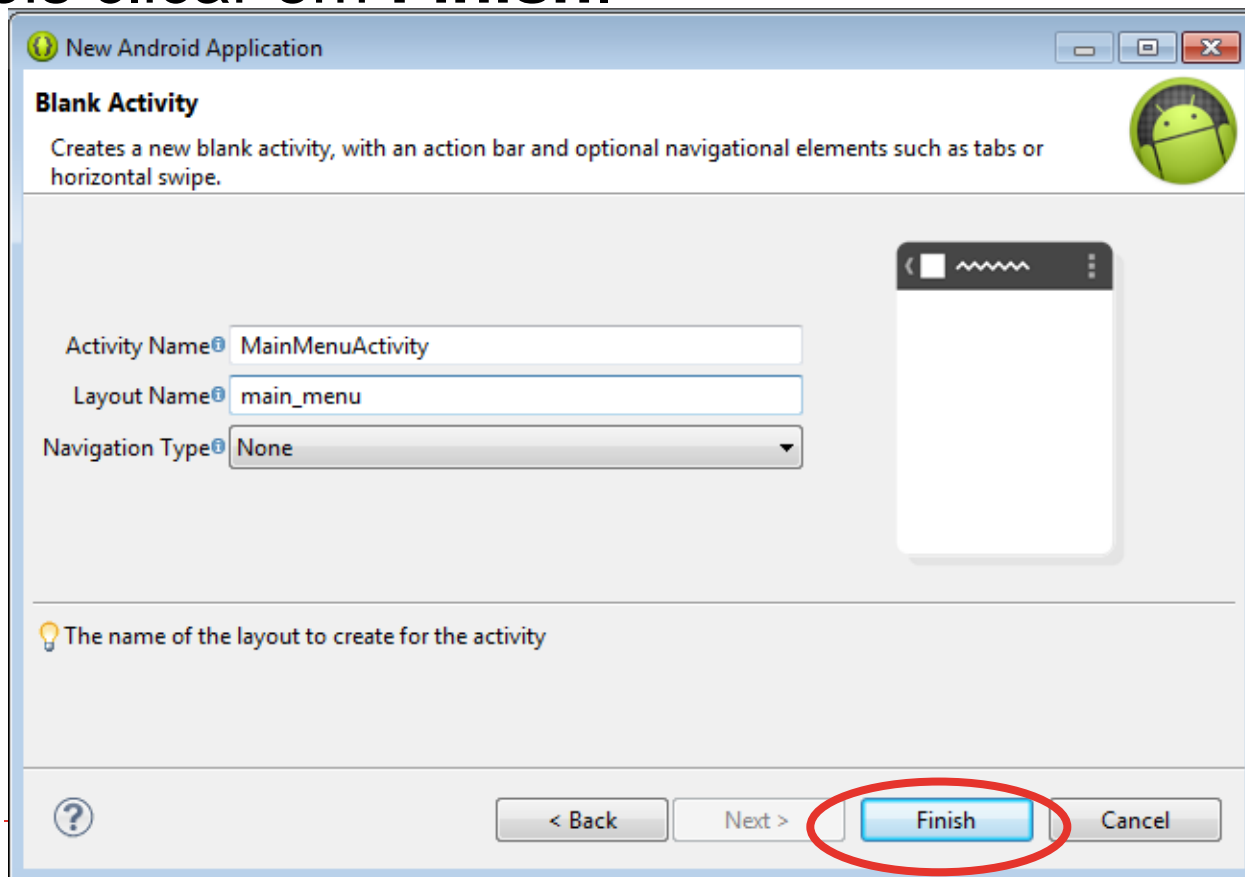
Criando uma Aplicação

- ▶ Selecione Blank Activity e clique em **Next >**

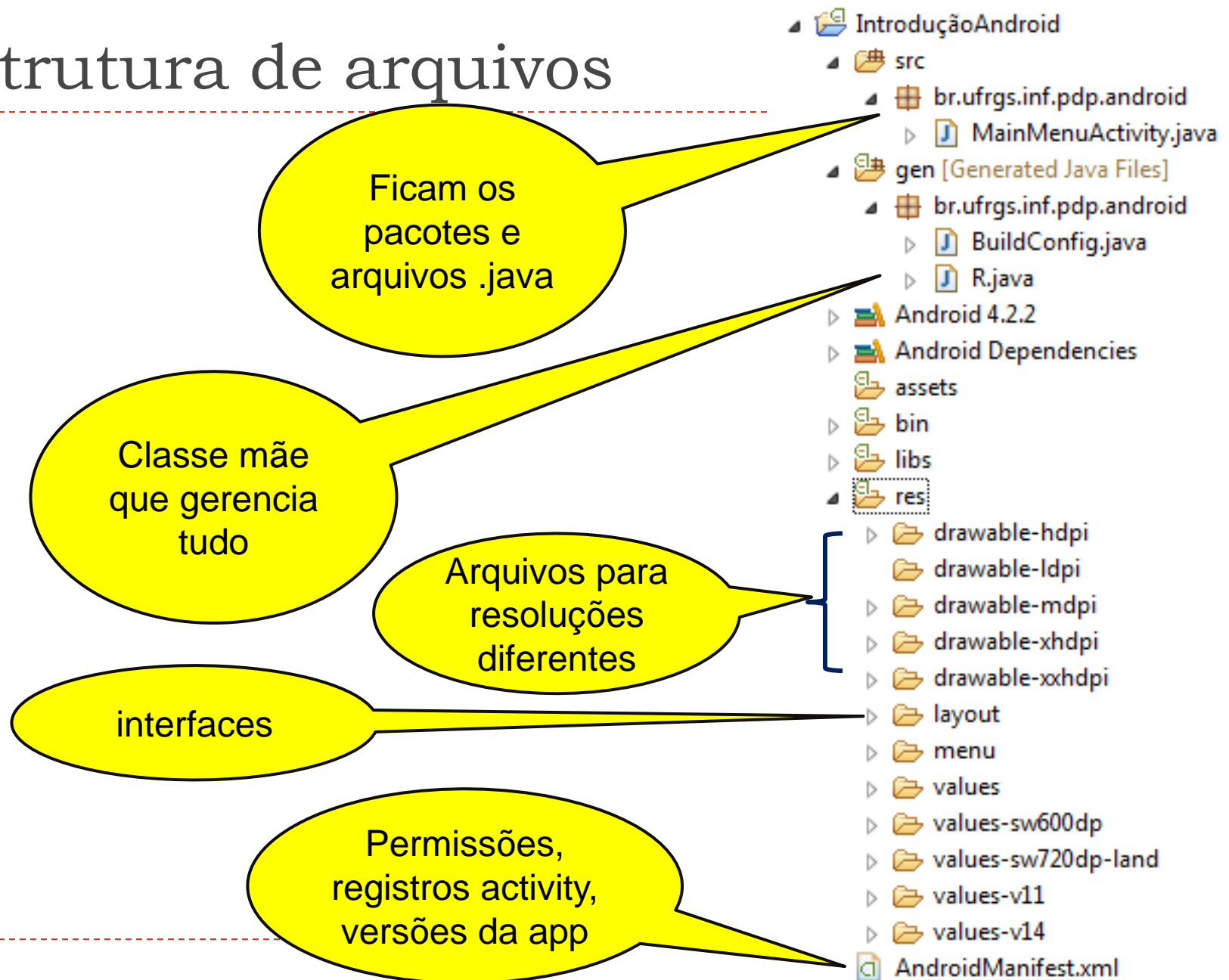


Criando uma aplicação

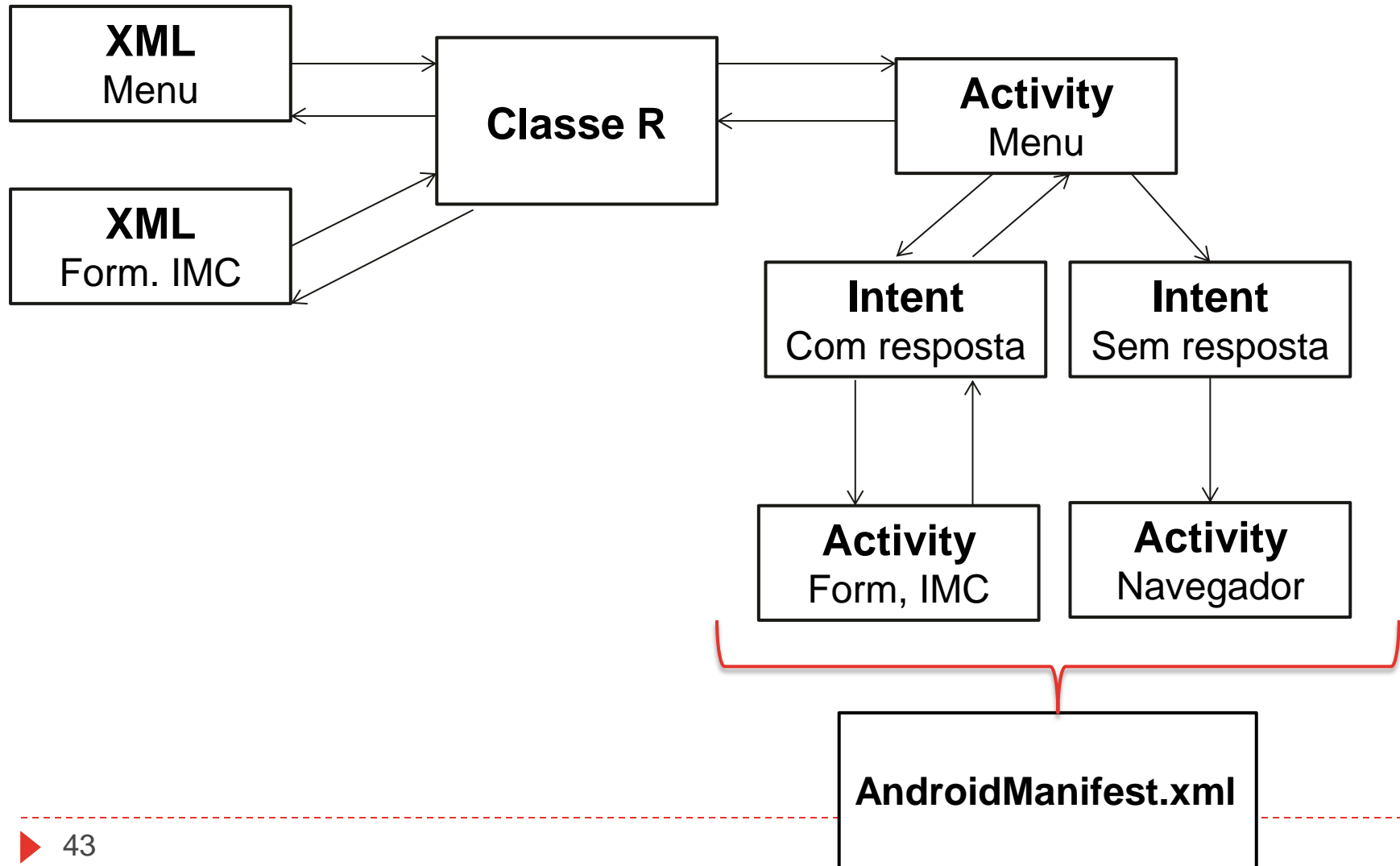
- ▶ Activity Name: MainMenuActivity
Layout Name: main_menu, Navigation Type: None
- ▶ Depois clicar em **Finish**.



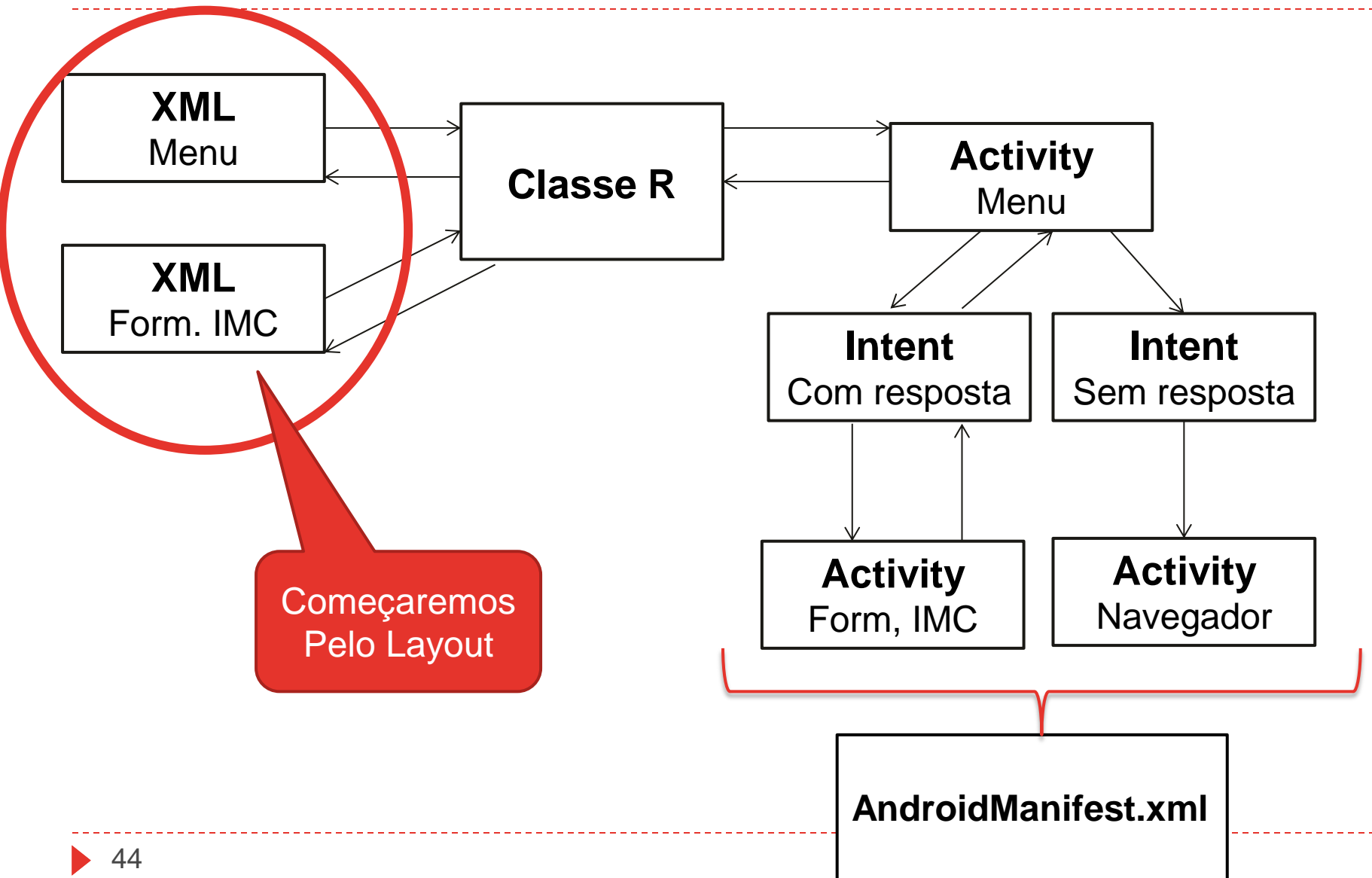
Estrutura de arquivos



Estrutura Conceitual



Estrutura Conceitual



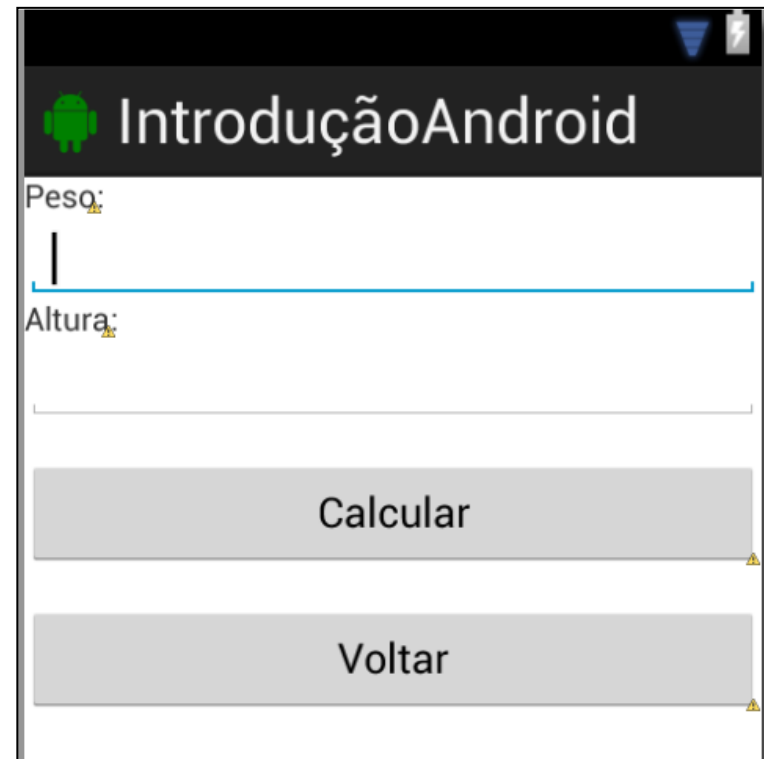
Exemplos

Construir as View:

1) menu principal;



2) Calculo de IMC.



Construir Tela Menu

Elemento Root: LinearLayout



Id: btnNavegador

Id: btnIMC

Id: btnSair

Elemento Horizontal: LinearLayout

Sem ID

Id: txtUltimoResultadoDeIMC



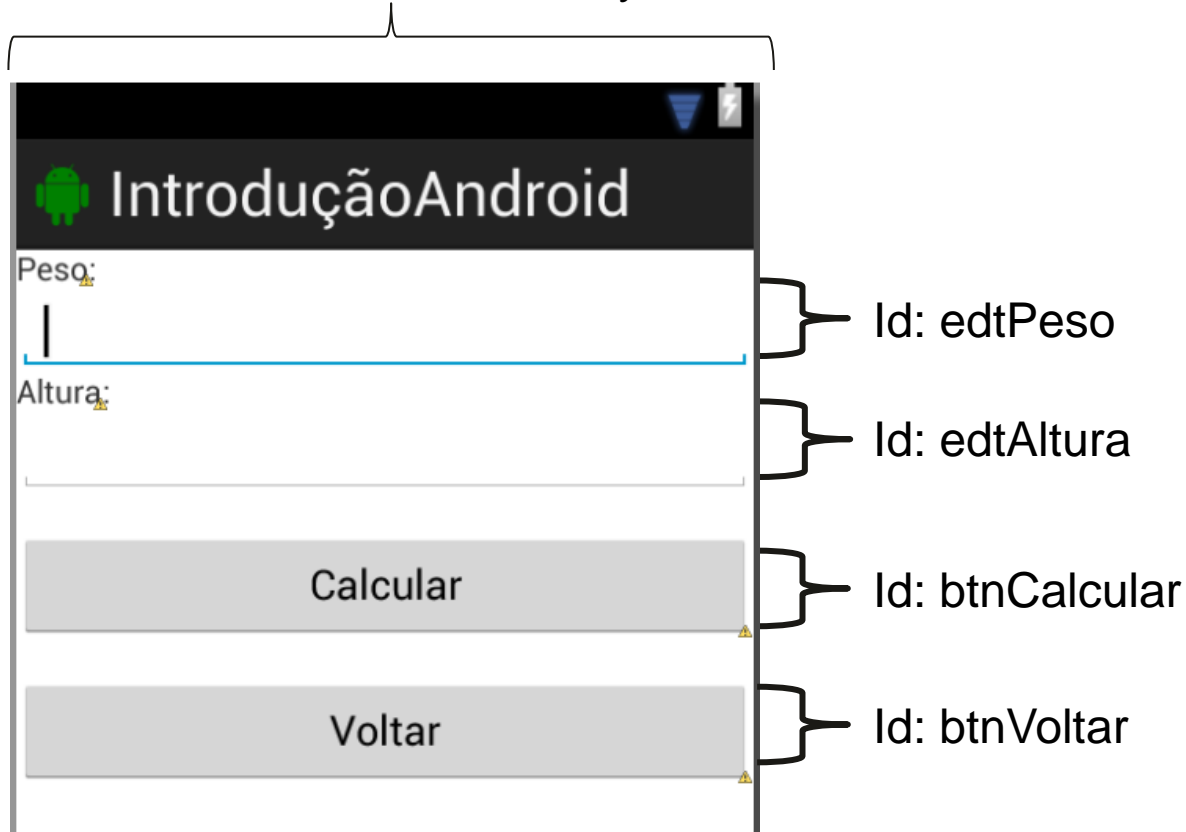
XML do menu

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <Button
        android:id="@+id/btnNavegador"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Abrir Navegador" />
    <Button
        android:id="@+id/btnIMC"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="IMC" />
    <Button
        android:id="@+id/btnSair"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Sair" />
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Último Resultado de IMC: " />
        <TextView
            android:id="@+id/txtUltimoResultadoDeIMC"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="0" />
    </LinearLayout>
</LinearLayout>
```



Construir: Tela do Formulário de IMC

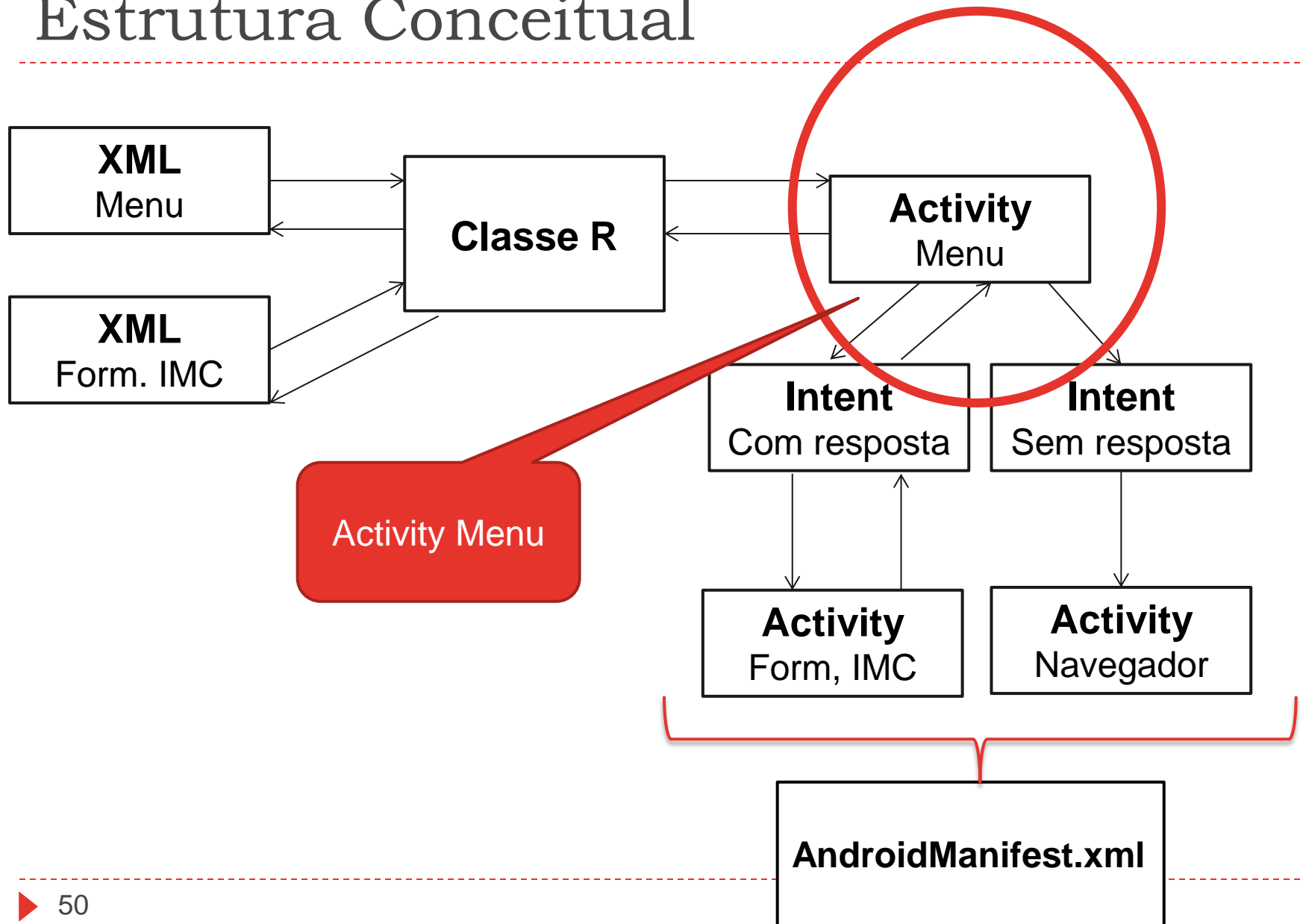
Elemento Root: LinearLayout



XML do Formulário de IMC

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Peso:" />
    <EditText
        android:id="@+id/edtPeso"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal" >
        <requestFocus />
    </EditText>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Altura:" />
    <EditText
        android:id="@+id/edtAltura"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal" />
    <Button
        android:id="@+id/btnCalcular"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Calcular" android:layout_marginTop="15dp"/>
    <Button
        android:id="@+id/btnVoltar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Voltar" android:layout_marginTop="15dp"/>
</LinearLayout>
```

Estrutura Conceitual



Método onCreate MainMenuActivity

- ▶ Adicionar os Objetos para integrar e manipular os elementos/componentes do layout main_menu.xml

```
public class MainMenuActivity extends Activity {  
  
    private Integer IMC_RC = 1;  
    private Button btnNavegador;  
    private Button btnIMC;  
    private Button btnSair;  
    private TextView txtUltimoResultadoDeIMC;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main_menu);  
  
        btnNavegador = (Button) findViewById(R.id.btnNavegador);  
        btnIMC = (Button) findViewById(R.id.btnIMC);  
        btnSair = (Button) findViewById(R.id.btnSair);  
        txtUltimoResultadoDeIMC = (TextView) findViewById(R.id.txtUltimoResultadoDeIMC);  
    }  
}
```

Método onCreate IMCActivity

- Criar Activity e adicionar os Objetos para integrar e manipular os componentes do layout imc.xml

```
public class IMCActivity extends Activity{

    private Double resultadoCalculo;
    private Button btnCalcular;
    private Button btnSair;
    private EditText edtPeso;
    private EditText edtAltura;

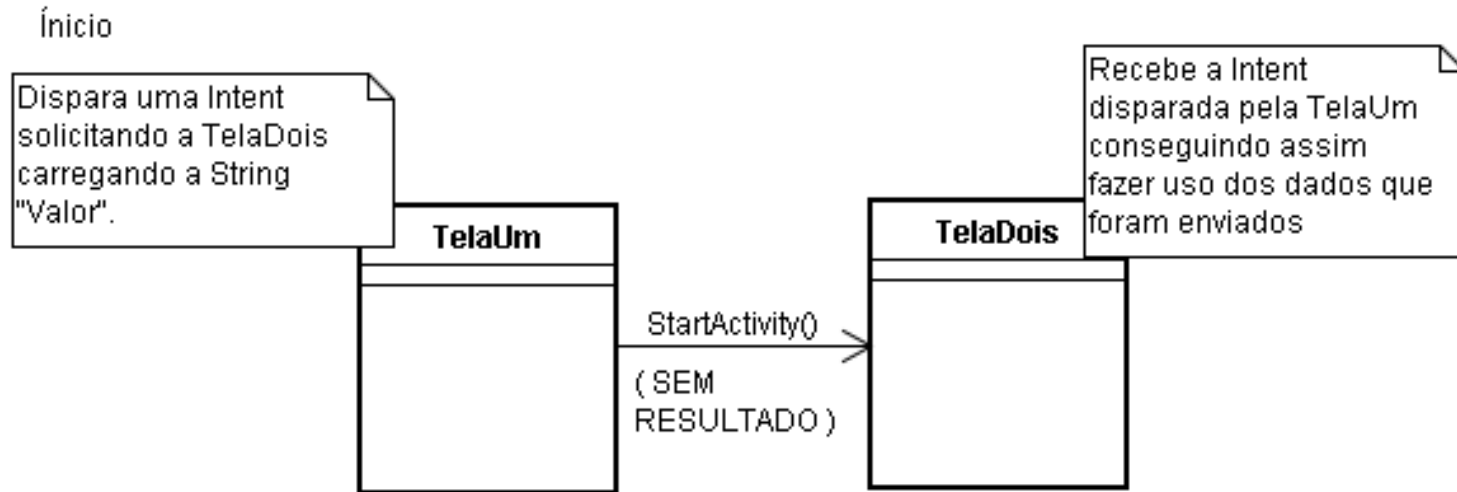
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.imc);

        this.resultadoCalculo = 0.0;
        this.btnCalcular = (Button) findViewById(R.id.btnCalcular);
        this.btnSair = (Button) findViewById(R.id.btnVoltar);
        this.edtAltura = (EditText) findViewById(R.id.edtAltura);
        this.edtPeso = (EditText) findViewById(R.id.edtPeso);
    }
}
```

Intent

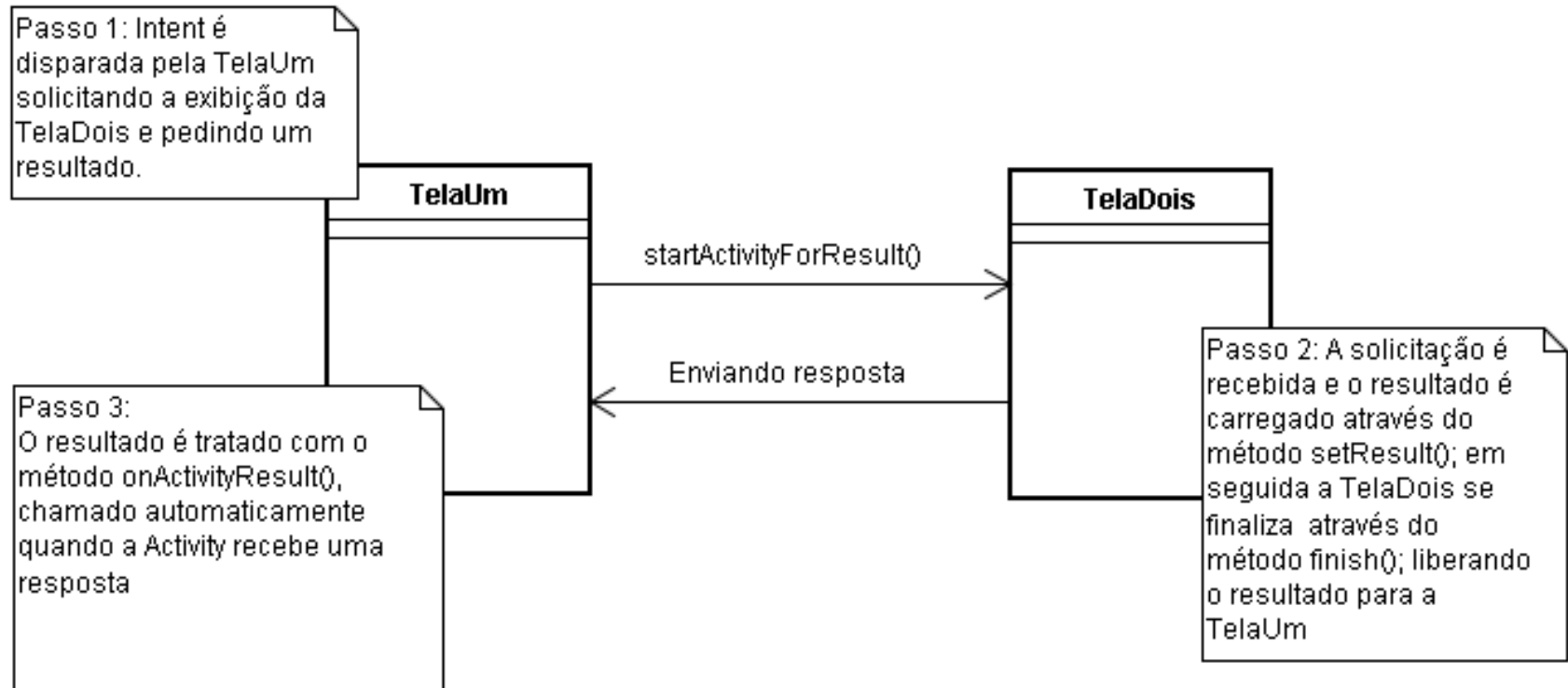
- ▶ Indica que há a intenção de executar outra Activity
- ▶ Pode somente executar uma Activity
 - ▶ `startActivity()`
- ▶ Pode executar uma Activity esperando sua resposta
 - ▶ `startActivityForResult()`

Trafegando dados usando o método `startActivity()`



- ▶ Quando se usa o `startActivity()`, o desenvolvedor está dizendo que a `Intent` vai ser disparada, mas que a `Activity` que a disparou não tem a necessidade de receber nenhum resultado da próxima tela que será exibida.

Trafegando dados usando o método `startActivityForResult()`.



Código de função de retorno

► Activity A:

```
public void acessarIMC() {
    Intent intent = new Intent(getBaseContext(), IMCActivity.class);
    startActivityForResult(intent, IMC_RC);
}

protected void onActivityResult(int requestCode, int resultCode,
    Intent intentRetornada) {
    if (requestCode == IMC_RC) {
        if (resultCode == RESULT_OK) {
            Double imc = intentRetornada.getDoubleExtra("imc", 0);
            txtUltimoResultadoDeIMC.setText(imc.toString());
            Log.d("Marcador res", imc.toString());
        }
    }
}
```

► Activity B:

```
btnSair.setOnClickListener(new View.OnClickListener() {

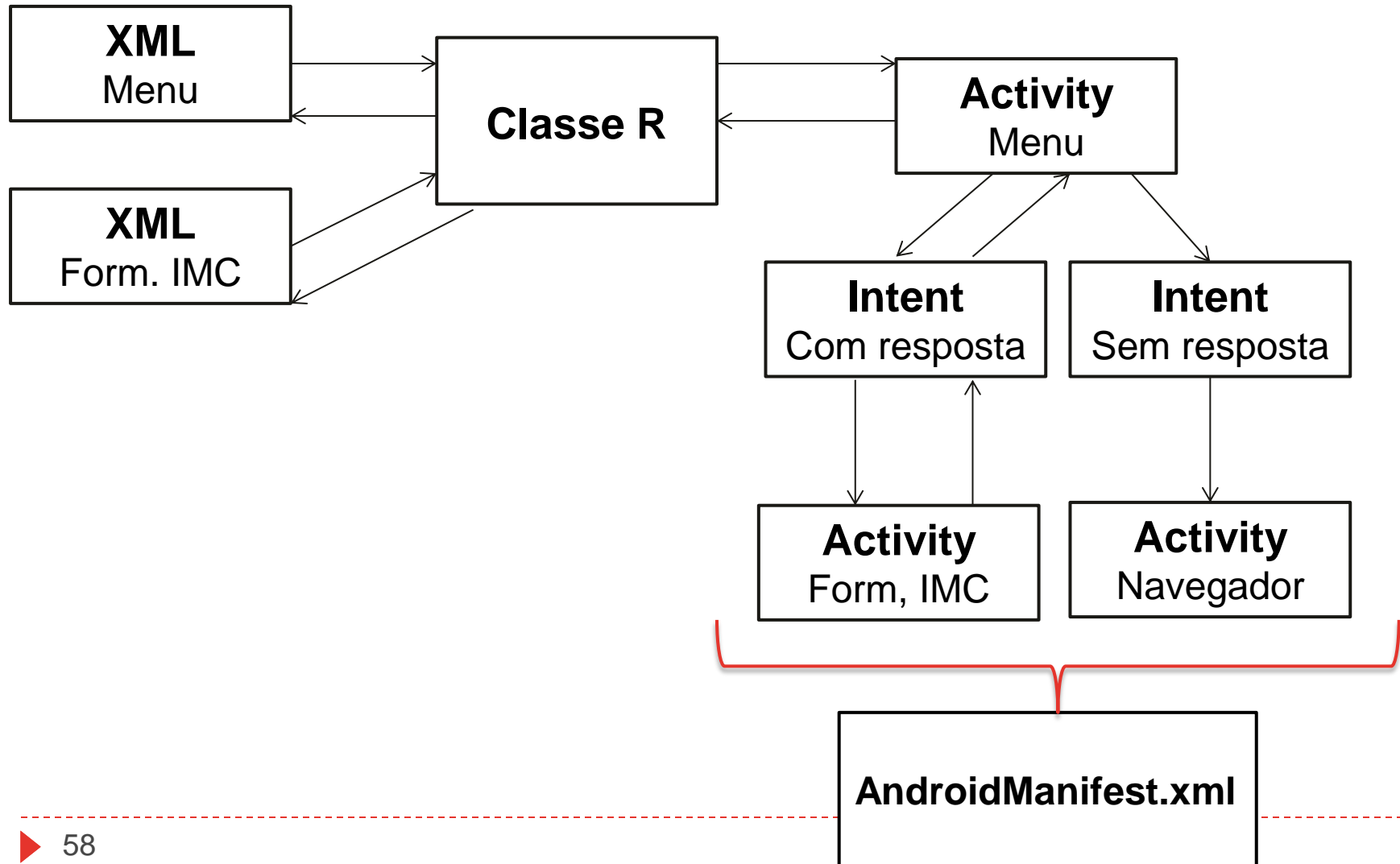
    @Override
    public void onClick(View v) {
        Intent it = new Intent();
        it.putExtra("imc", resultadoCalculo);
        setResult(RESULT_OK, it);
        finish();
    }
});
```



Navegando pelo Android - A Classe Intent

- ▶ Cenário: duas activities: uma para buscar dados e outra para listá-los.
- ▶ Na tela 1 uma série de componentes EditText para que o usuário possa fornecer as informações para a consulta:
- ▶ Os dados recuperados apenas poderão ser passados para a tela de lista caso uma **Intent** tenha os carregado através do método **putExtra()** e tenha sido disparada através do método **startActivity()** ou **startActivityForResult()**.

Estrutura Conceitual

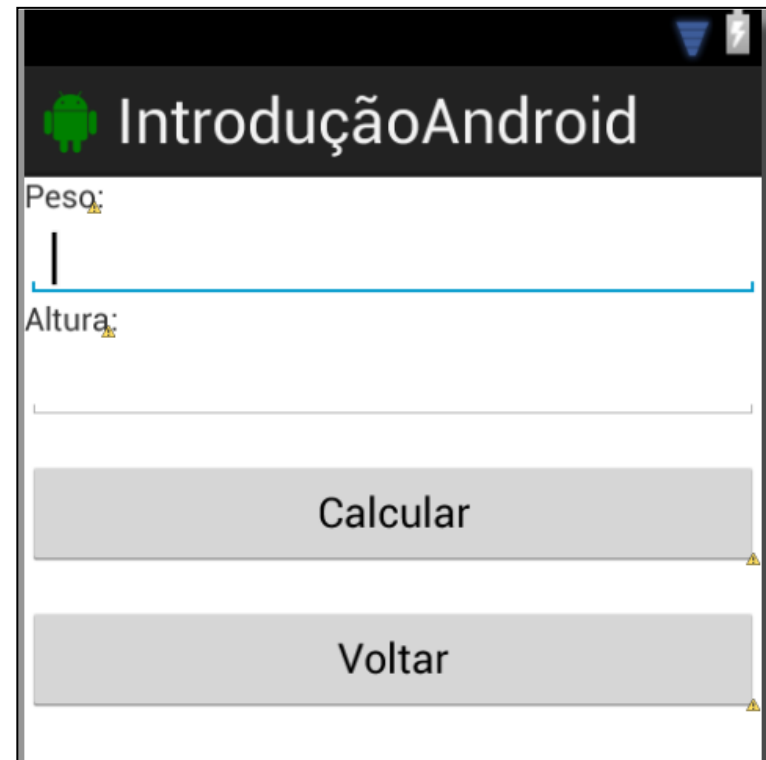


Telas

1) menu principal;



2) Calculo de IMC.



Adicionar Evento de Click nos botões

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main_menu);  
  
    btnNavegador = (Button) findViewById(R.id.btnNavegador);  
    btnIMC = (Button) findViewById(R.id.btnIMC);  
    btnSair = (Button) findViewById(R.id.btnSair);  
    txtUltimoResultadoDeIMC = (TextView) findViewById(R.id.txtUltimoResultadoDeIMC);  
  
    btnNavegador.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            abrirNavegador();  
        }  
    });  
    btnIMC.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View arg0) {  
            acessarIMC();  
        }  
    });  
    btnSair.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View arg0) {  
            finish();  
        }  
    });  
}
```

Funções dos Eventos Click Listener

- ▶ Funções dos eventos:
 - ▶ abrirNavegador()
 - ▶ acessarIMC()
- ▶ Função de retorno de resultado do cálculo
 - ▶ onActivityResult

Código das Funções

```
protected void onCreate(Bundle savedInstanceState) {  
  
    public void abrirNavegador() {  
        String endereco = "http://inf.ufrgs.br";  
        Uri uri = Uri.parse(endereco);  
        Intent intent = new Intent(Intent.ACTION_VIEW, uri);  
        startActivity(intent);  
    }  
  
    public void acessarIMC() {  
        Intent intent = new Intent(getBaseContext(), IMCAActivity.class);  
        startActivityForResult(intent, IMC_RC);  
    }  
  
    protected void onActivityResult(int requestCode, int resultCode,  
        Intent intentRetornada) {  
        if (requestCode == IMC_RC) {  
            if (resultCode == RESULT_OK) {  
                Double imc = intentRetornada.getDoubleExtra("imc", 0);  
                txtUltimoResultadoDeIMC.setText(imc.toString());  
                Log.d("Marcador res", imc.toString());  
            }  
        }  
    }  
}
```

Código IMCActivity

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.imc);

    this.resultadoCalculo = 0.0;
    this.btnCalcular = (Button) findViewById(R.id.btnCalcular);
    this.btnSair = (Button) findViewById(R.id.btnVoltar);
    this.edtAltura = (EditText) findViewById(R.id.edtAltura);
    this.edtPeso = (EditText) findViewById(R.id.edtPeso);
    btnCalcular.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Double altura = Double.parseDouble(edtAltura.getText().toString());
            Double peso = Double.parseDouble(edtPeso.getText().toString());
            resultadoCalculo = peso/(altura * altura);
            Toast.makeText(IMCActivity.this,
                "Seu imc é: "+String.valueOf(resultadoCalculo), Toast.LENGTH_LONG).show();
        }
    });
    btnSair.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent it = new Intent();
            it.putExtra("imc", resultadoCalculo);
            setResult(RESULT_OK,it);
            finish();
        }
    });
}
```

Executar...

- ▶ Executem...
- ▶ Erro?

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="br.ufrgs.inf.pdp.android"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk
        android:minSdkVersion="9"
        android:targetSdkVersion="17" />
    <uses-permission android:name="android.permission.INTERNET"/>
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="br.ufrgs.inf.pdp.android.MainMenuActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name="br.ufrgs.inf.pdp.android.IMCActivity"></activity>
    </application>
</manifest>
```

Adicionar
Permissão
de uso da
Internet

Adicionar a
nova Activity
criada

Executar...

- ▶ Executem...
- ▶ Erro?

Exercícios

Exercícios Para Entregar

Exercício 1

- ▶ 1) Dado um servidor de Socket, que efetua a soma de dois números inteiros baseado no protocolo na tabela abaixo, faça um cliente de Socket em Android e uma interface com dois EditText que conterão os números a serem somados e submetidos e exibir o resultado com um Toast ou em um TextView. A aplicação deve permitir fazer a soma de dois números diversas vezes. Ao sair deve-se enviar uma mensagem ao servidor afirmando que o cliente vai se desconectar.
- ▶ Detalhes:
 - ▶ <https://docs.google.com/document/d/1VkuLJKioBuKbl8ptTLAvZtRjV63e03--9c3wjDZyBFU/edit>

Exercício 2

- ▶ 2) Usar a API de Geo localização pela classe LocationManager e obter a latitude, longitude e altura atual, exibindo essas informações na tela (Seja com Toast ou com TextView).
- ▶ Detalhes:
 - ▶ <https://docs.google.com/document/d/1VkuLJKioBuKbl8ptTLAvZtRjV63e03--9c3wjDZyBFU/edit?usp=sharing>

Exercício 3

- ▶ 3) Deve-se implementar utilizando a biblioteca Google Maps v2 para Android um sistema onde busca a localização atual (latitude e longitude) através da classe LocationManager e exibe um ponto em um mapa com a latitude e a longitude atual.
- ▶ **Dica:** Faça funcionar primeiro somente o MAPA utilizando coordenadas fixas, depois adicionar o recurso para buscar a posição por Geo Localização.
- ▶ Detalhes:
 - ▶ <https://docs.google.com/document/d/1VkuLJKioBuKbl8ptTLAvZtRjV63e03--9c3wjDZyBFU/edit?usp=sharing>

Exercícios Extras (Entretenimento)

Exercício 1 - Entretenimento

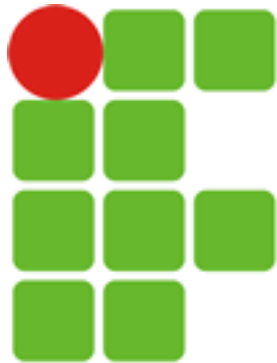
- ▶ 1) Criar um formulário de clientes com TableLayout, onde a primeira coluna contém TextViews e a segunda EditTexts. Também, para escolher o sexo utilizar RadioButton (dica: Radio Group.). Depois integrar com a Activity e após clicar em um botão exibir os dados com uma mensagem usando a Classe Toast na tela.
- ▶ OBS.: Exemplo de como utilizar RadioButton:
<http://developer.android.com/guide/topics/ui/controls/radiobutton.html>

Exercício 2 - Entretenimento

- ▶ 2) Criar uma lista utilizando o elemento ListView ou fazer uma lista com diversos Buttons. Ao clicar em um dos componentes (ou linha no caso do ListView) é passado por parâmetro o nome que está rotulado dentro do Button para uma segunda Activity. Na segunda Activity há um campo EditText e um Button para editar o nome. Quando submetido, o nome alterado é retornado e substitui o rótulo do Button que gerou o evento.
- ▶ Exemplo de como passar parâmetros para outra Activity:
<http://minhaexperienciacomandroid.blogspot.com.br/2011/11/passagem-de-parametros-entre-activities.html>
- ▶ Exemplo de ListView:
<http://zarelli.wordpress.com/2012/01/23/android-criando-um-listview-personalizado/>

Exercício 3 - Entretenimento

- ▶ 3) Criar um formulário com um componente Button dois TextView e dois componentes EditText, em que o primeiro EditText representa um valor inteiro x e o segundo um valor inteiro y. Ao submeter passar esses dados para uma nova Activity construída com um AbsoluteLayout que adiciona uma imagem (pode ser o ícone da aplicação) nas coordenadas x e y passados por parâmetro.
- ▶ OBS.: O valor de x e y não devem ser maiores do que o tamanho da tela.
- ▶ Exemplo de como passar parâmetros para outra Activity:
<http://minhaexperienciacomandroid.blogspot.com.br/2011/11/passagem-de-parametros-entre-activities.html>



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL-RIO-GRANDENSE
Campus Passo Fundo



Introdução à Plataforma Android

Contato: gaborges@inf.ufrgs.br