

Programação para Dispositivos Móveis

02 – Primeiro Projeto



Termos que iremos utilizar:

Activities: Seu projeto pode ter uma ou várias Activities. Activity é um container para interface e código fonte. Podemos pensar em Activity como uma página da aplicação

Intents: Sistema de Mensagem do Android. Composto de dois parâmetros (Action e Data). Intents são utilizados para iniciar activities bem como se comunicar com várias parte do sistema Android.

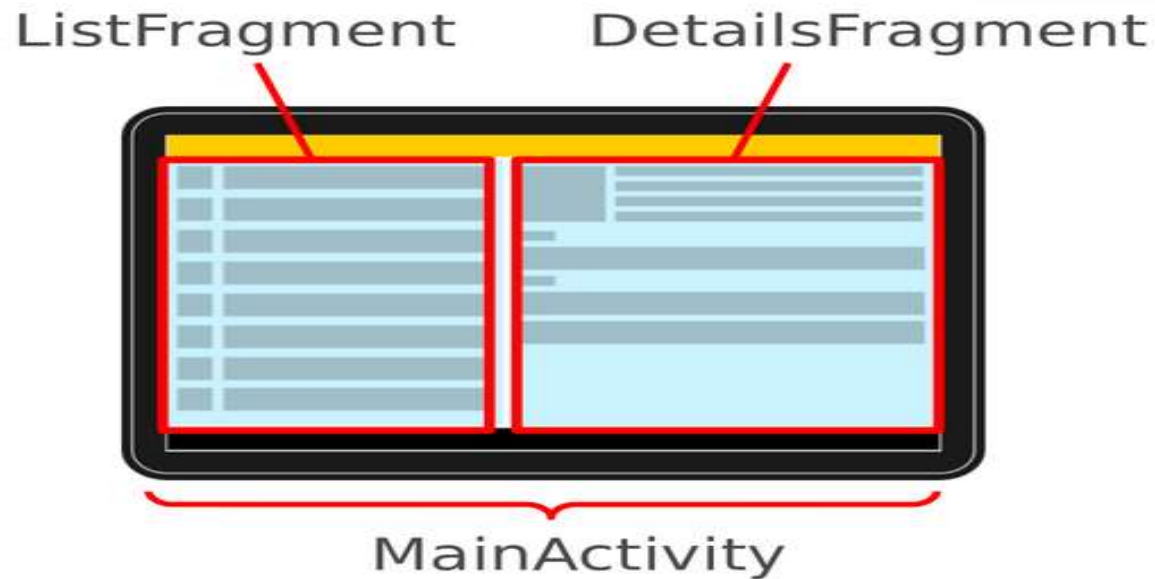
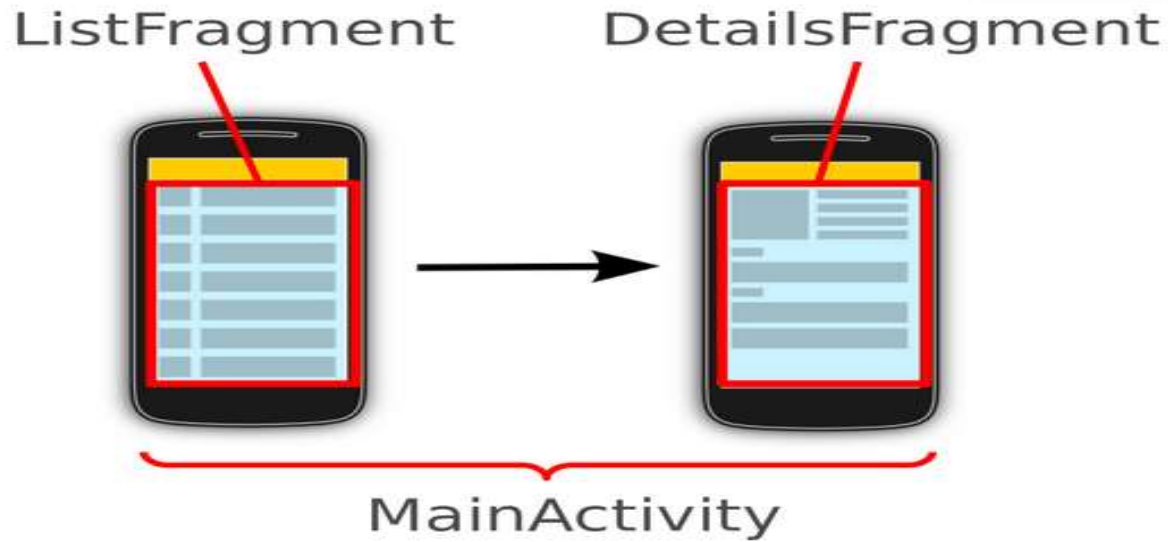
View: Elemento básico da interface do usuário, é uma área retangular na tela para desenho e tratamento de eventos.

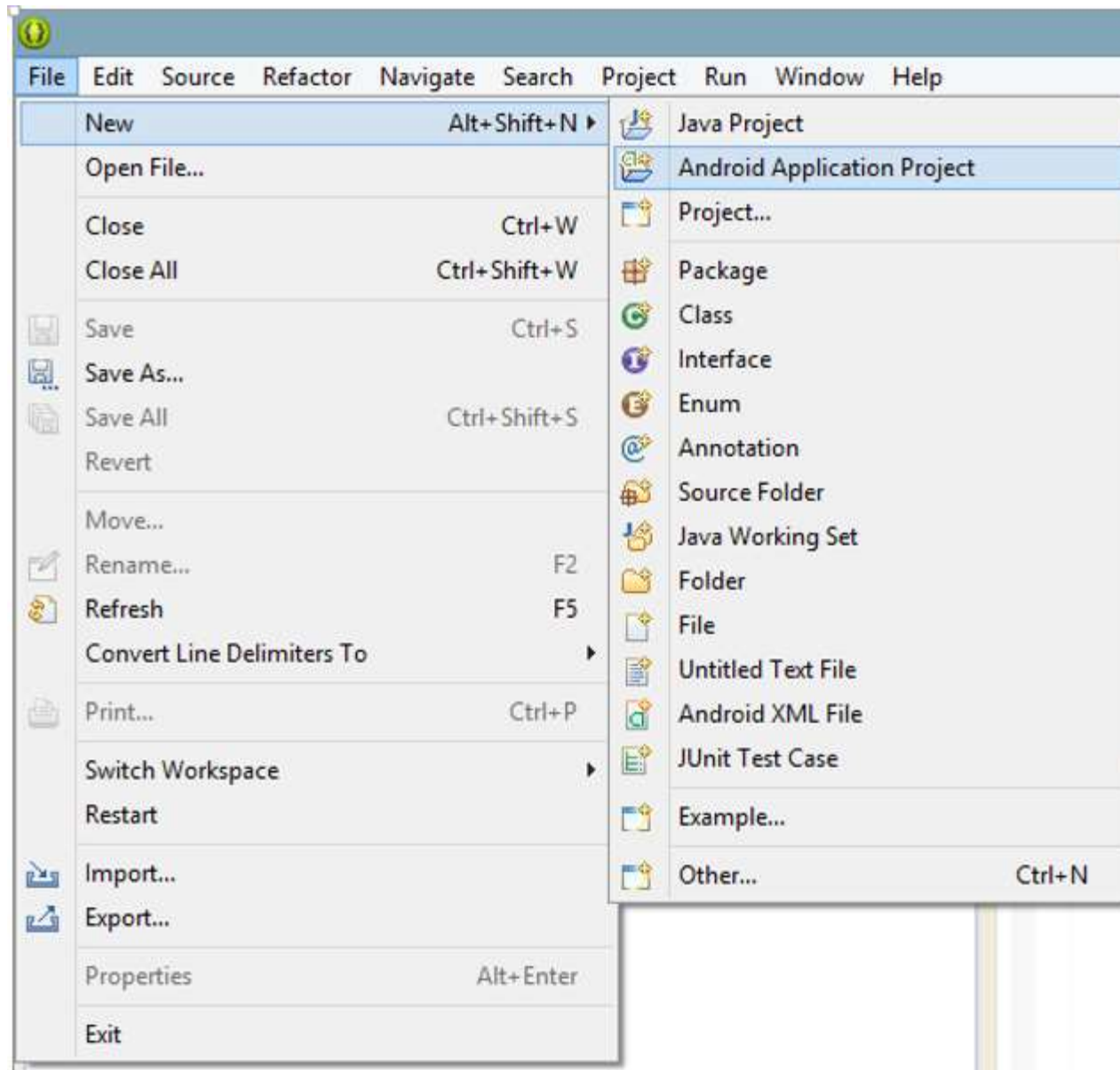
Fragments (3.0+): Cada página em uma aplicação Android é uma Activity, para que possamos trabalhar com várias Activities na mesma tela usamos fragments. Muito útil para aplicações multi-devices Tablet/Phone.

Loaders: Provê uma maneira de ler os dados em segundo plano, sem interferir no desempenho da interface do usuário

Action Bar: Barra que sempre fica no topo da tela







New Android Application

ⓘ A project with that name already exists in the workspace

Application Name:

Project Name:

Package Name:

Minimum Required SDK:

Target SDK:

Compile With:

Theme:

💡 Choose the base theme to use for the application

New Android Application

Configure Project

☒ Create custom launcher icon

☒ Create activity

☐ Mark this project as a library

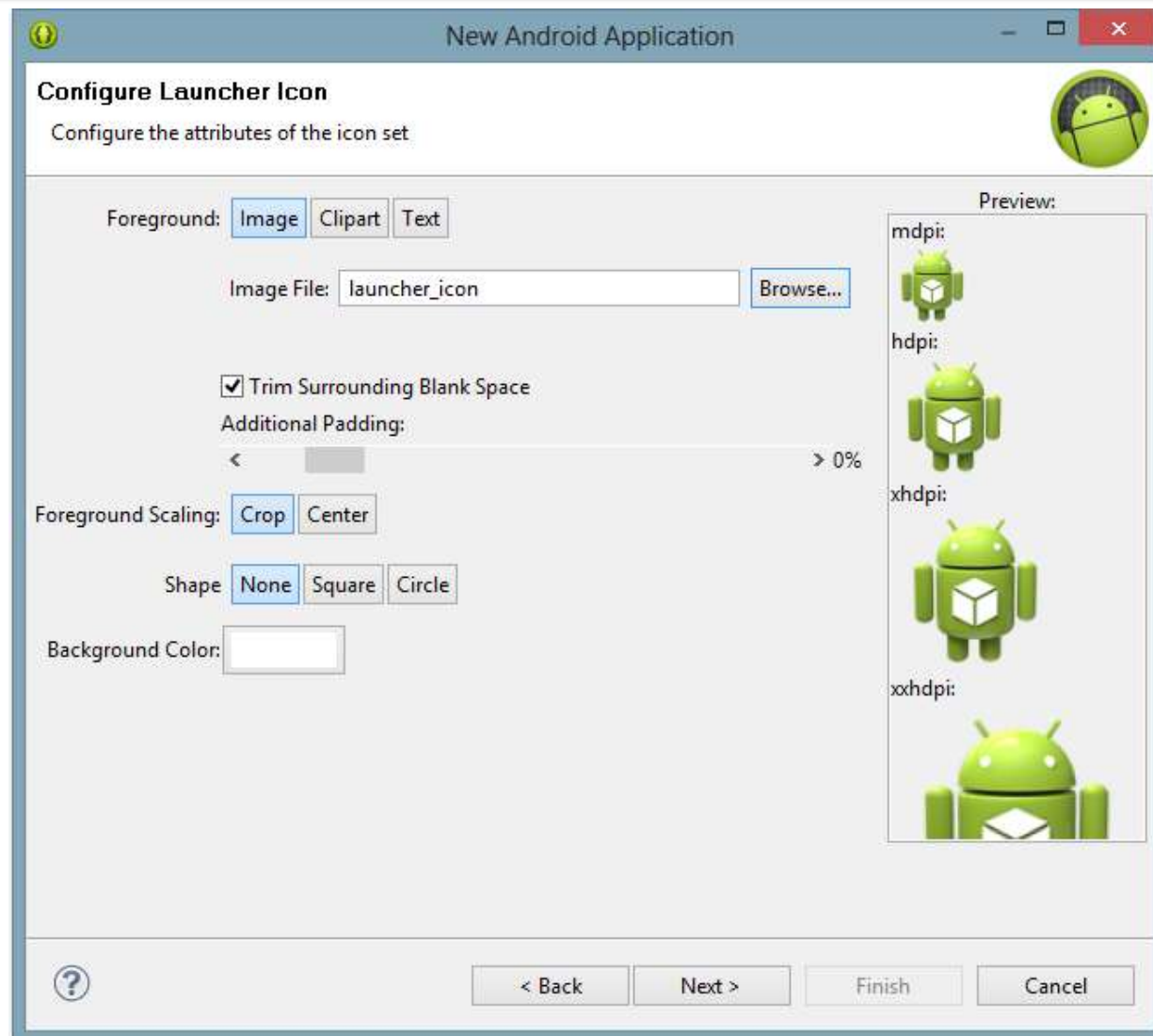
☒ Create Project in Workspace

Location:

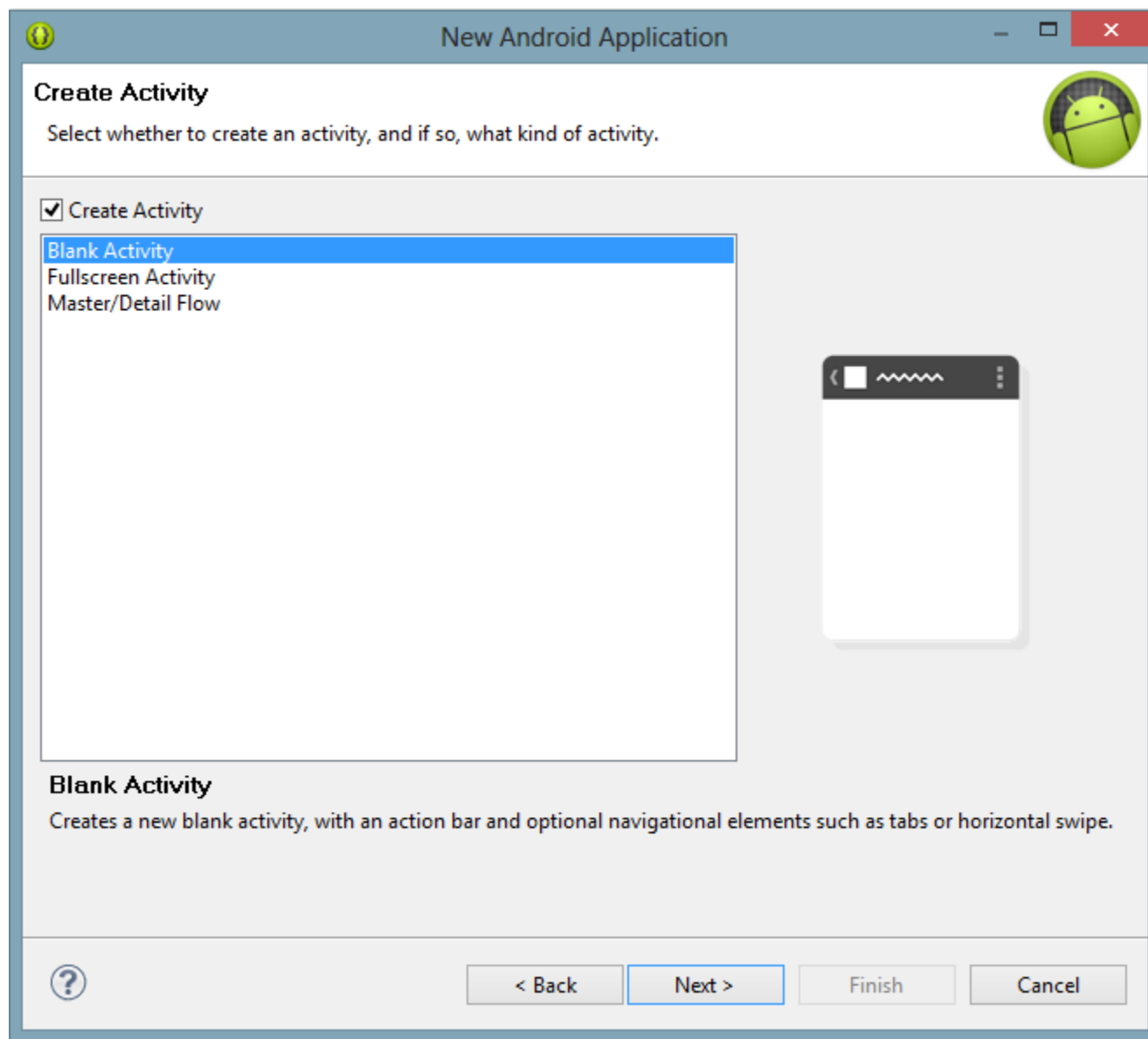
Working sets


☐ Add project to working sets


Working sets:



DICA: Ferramentas Úteis para customização de ícones e barras: <http://j.mp/androidassetstudio>
(Android Assets Studio)



New Android Application—□×

Blank Activity

Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.

Activity Name⑥


Layout Name⑥


Navigation Type⑥


MainActivity

activity_main

None



 The name of the activity class to create

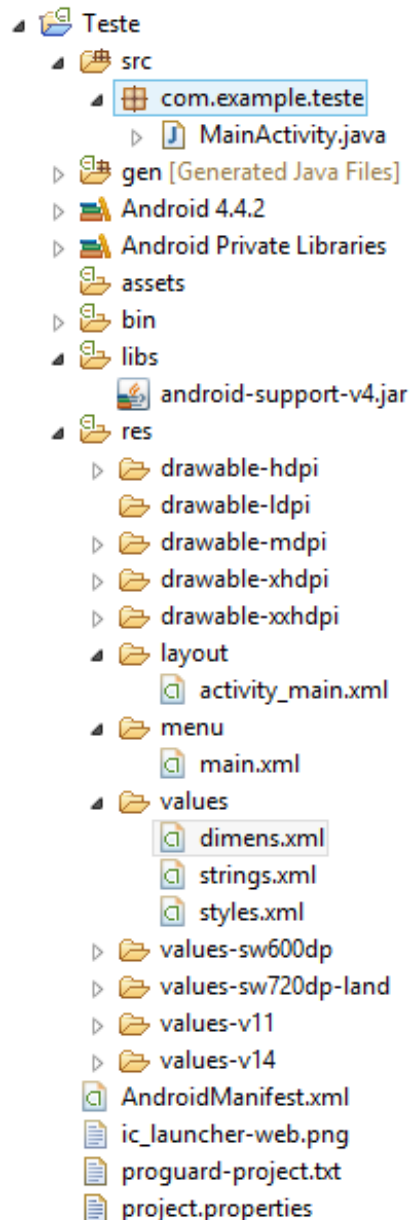


< Back

Next >

Finish

Cancel



src – contém o código fonte da aplicação. *.java

gen – contém R.java, um arquivo gerado e compilado que referencia todos os recursos encontrados no projeto.

NÃO MODIFICAR !!!!

Android 4.4.2 – contém a biblioteca de classes necessárias na aplicação

assets – contém os recursos da aplicação, como html, arquivos texto, databases, etc.

bin – gerado pelo compilador, contém os .class que gerarão o .apk (aplicação)

libs – bibliotecas

res – todos os recursos da aplicação

drawable – imagens, ícones, etc.

layout – xml de layout da activity

menu – itens do menu

values – mapeamento campo valor

AndroidManifest.xml – manifesto do Android.

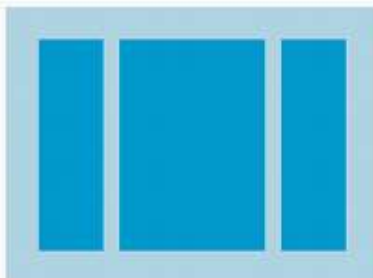
```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_turma" />

</RelativeLayout>
```

Layout

Linear Layout



A layout that organizes its children into a single horizontal or vertical row. It creates a scrollbar if the length of the window exceeds the length of the screen.

Relative Layout



Enables you to specify the location of child objects relative to each other (child A to the left of child B) or to the parent (aligned to the top of the parent).

Web View

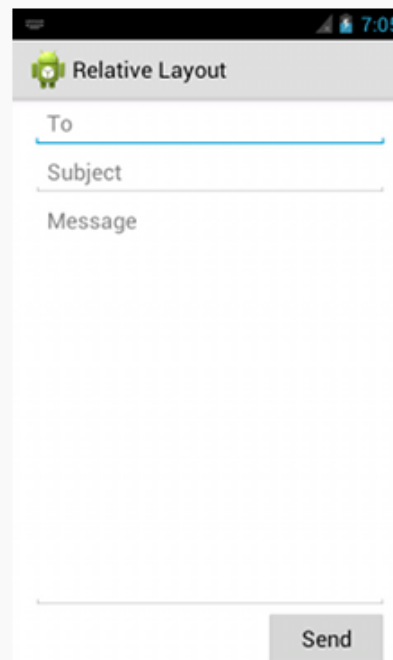


Displays web pages.

LinearLayout : É um grupo de visualização que alinha todos os filhos em uma única direção (vertical ou horizontal). Essa orientação pode ser especificada no layout com o atributo `android:orientation`

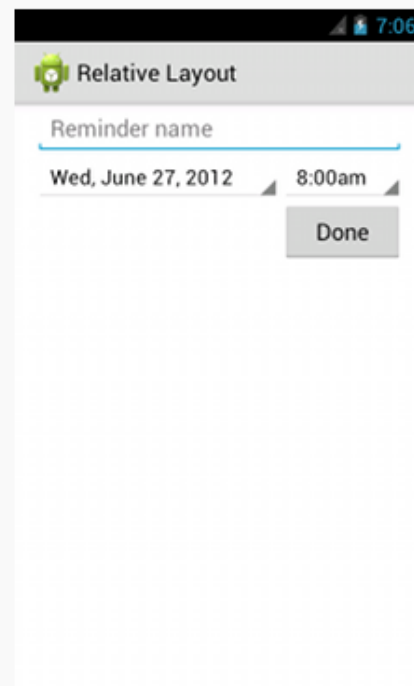
Example

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:orientation="vertical" >
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/to" />
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/subject" />
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="top"
        android:hint="@string/message" />
    <Button
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="@string/send" />
</LinearLayout>
```



RelativeLayout: A posição dos filhos são relativos a algum elemento.

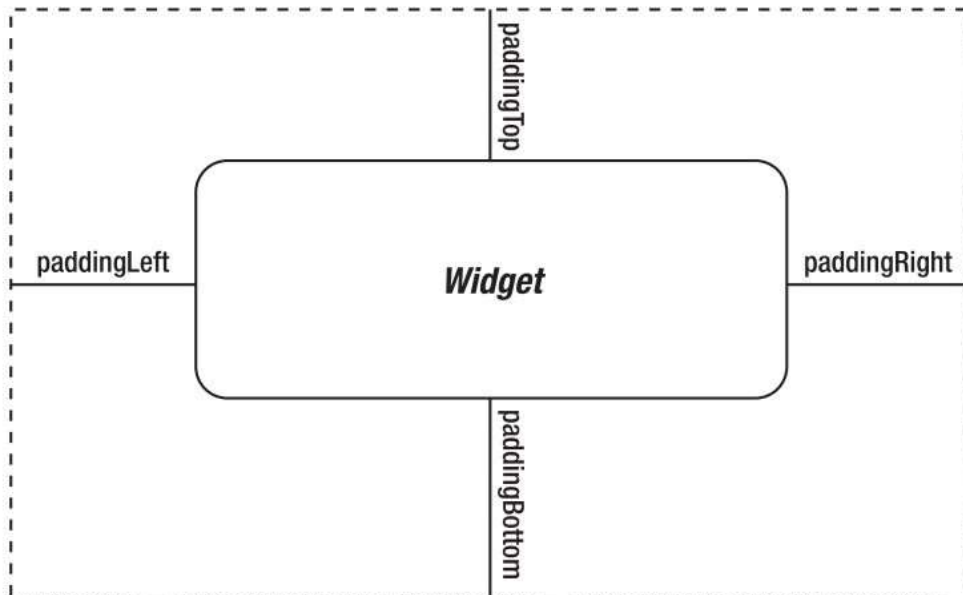
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp" >
    <EditText
        android:id="@+id/name"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/reminder" />
    <Spinner
        android:id="@+id/dates"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/times" />
    <Spinner
        android:id="@+id/times"
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentRight="true" />
    <Button
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/times"
        android:layout_alignParentRight="true"
        android:text="@string/done" />
</RelativeLayout>
```



android:layout_height : Altura do Elemento
android:layout_width : Largura do Elemento

match_parent: O elemento é maximizado para o tamanho do Pai (menos padding)

wrap_content: O elemento terá o espaço necessário só para ele caber.



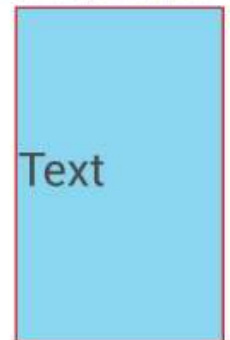
wrap_content

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    ...  
>
```



match_parent

```
android:layout_width="wrap_content"  
android:layout_height="match_parent"  
  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
  
android:layout_width="match_parent"  
android:layout_height="match_parent"
```



activity_main.xml

```
android:paddingBottom="@dimen/activity_vertical_margin"  
android:paddingLeft="@dimen/activity_horizontal_margin"  
android:paddingRight="@dimen/activity_horizontal_margin"  
android:paddingTop="@dimen/activity_vertical_margin"
```

Os valores estão na pasta res/values/Dimens.xml

```
<resources>  
  
    <!-- Default screen margins, per the Android Design guidelines. -->  
    <dimen name="activity_horizontal_margin">16dp</dimen>  
    <dimen name="activity_vertical_margin">16dp</dimen>  
  
</resources>
```

Fonte: www.blog.edersonlima.net/?p=261

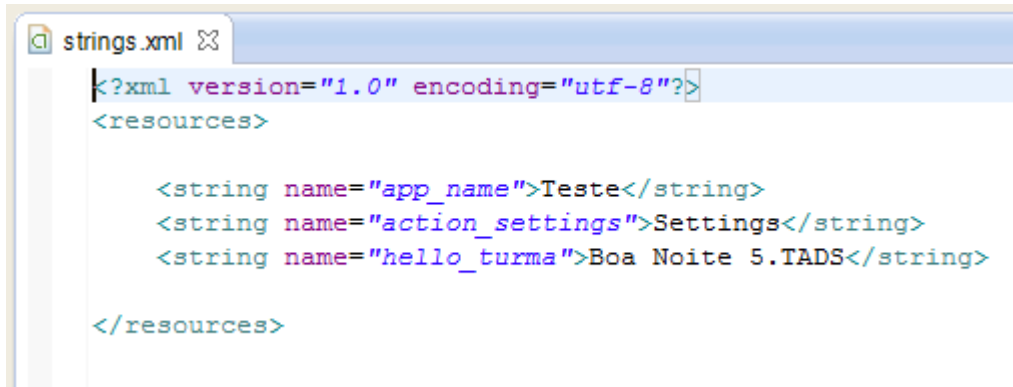
dp = Medida de Dimensão

Opção	Descrição
px (pixels)	Correspondente ao número de pixels da tela
in (polegadas)	Baseado no tamanho físico da tela
mm (milímetros)	Baseado no tamanho físico da tela
pt (pontos)	1/72 de uma polegada, baseado no tamanho físico da tela
dp ou dip	(Density-independent Pixels) Essa unidade é relativa à resolução da tela. Por exemplo, se a resolução da tela é de 160 dpi, significa que um dp representa 1 pixel em um total de 160.
sp	(Scale-independent Pixels) Idem ao dp, mas também considera o tamanho da fonte que o usuário está utilizando. É recomendado que use essa unidade quando especificar o tamanho de uma fonte, para que esta seja automaticamente ajustada conforme as preferências da tela do usuário.

activity_main.xml

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/hello_turma" />
```

Os valores estão na pasta res/values/string.xml



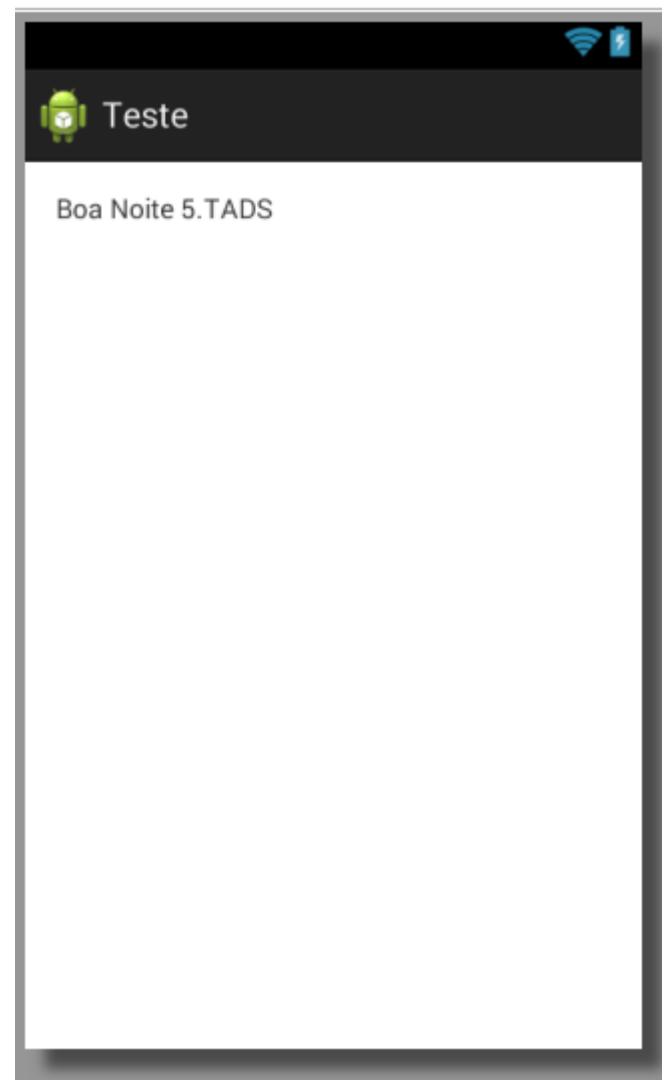
```
strings.xml
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">Teste</string>
    <string name="action_settings">Settings</string>
    <string name="hello_turma">Boa Noite 5.TADS</string>

</resources>
```



```
*activity_main.xml ⌕  
  
<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:paddingBottom="@dimen/activity_vertical_margin"  
    android:paddingLeft="@dimen/activity_horizontal_margin"  
    android:paddingRight="@dimen/activity_horizontal_margin"  
    android:paddingTop="@dimen/activity_vertical_margin"  
    tools:context=".MainActivity" >  
  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="@string/hello_turma" />  
  
</RelativeLayout>
```



Teste Manifest

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

    <uses-sdk
        android:minSdkVersion="15"
        android:targetSdkVersion="19" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.teste.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

- Define o nome do pacote com com.example.teste
- A versão do código é 1. Pode ser usado via código para verificar se a aplicação precisa ser atualizada
- Nome da versão 1.0. Normalmente utilizada para mostrar para o usuário.
- android:minSdkVersion: Define a versão mínima do Android para rodar a aplicação
- Android:icon : usa a imagem ic_launcher.png que está na pasta res/drawable
- intent-filter
 - A ação do intent-filter tem o nome de android.intent.action.MAIN para indicar que esta activity serve de início da aplicação
 - A categoria para o intent-filter é nomeada de android.intent.category.LAUNCHER para indicar que a aplicação pode ser disparada pelo ícone da aplicação

Um layout é atribuído a uma Activity pela chamada do método setContentView().

MainActivity.java

```
package com.example.teste;

import android.os.Bundle;

public class MainActivity extends Activity {

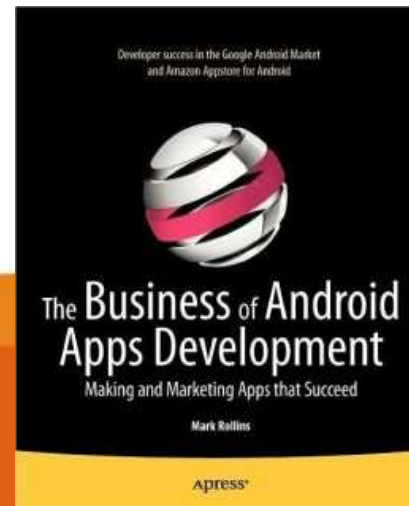
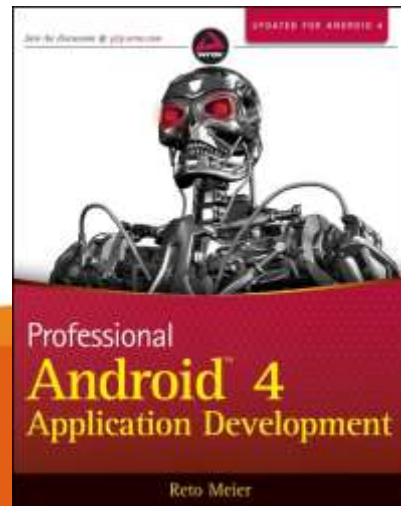
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

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

Folha 02-Exercícios



Anhanguera



Lauren Dorsey
Shane Conder

