



# Resources e Layouts

Professor Emerson Alencar  
[emerson@imd.ufrn.br](mailto:emerson@imd.ufrn.br)

# RESOURCES

- ▶ Exemplos de recursos (resources)
  - Textos
  - Imagens
  - Sons
  - Layouts de tela
- ▶ O ideal é não referenciar os resources diretamente no código
  - Repetição de código
  - Difícil de manter
- ▶ O Android possui mecanismos para trabalhar com resources de forma fácil e centralizada

# LOCALIZAÇÃO DOS RESOURCES

- ▶ Os resources ficam localizados dentro do diretório /res
- ▶ Este diretório contém subdiretórios de acordo com tipo de recurso que representam
  - /res/values-xxx
  - /res/layout
  - /res/drawable-xxx
  - /res/mipmap
  - etc.
- ▶ Dentro dos subdiretórios ficam os arquivos de resources
- ▶ Documentação:  
<https://developer.android.com/guide/topics/resources/providing-resources.html>

# STRING RESOURCES

- ▶ Representam textos (strings) como recursos
- ▶ Estão localizados no diretório /res/values
- ▶ A representação dos resources é feita em XML
  - O nome do XML não é importante para o Android

```
1 <resources>
2   <string name="app_name">Meu Projetoto</string>
3   <string name="app_botao">Pressione Aqui</string>
4 </resources>
```

Nome

Valor

# COLOR RESOURCES

- ▶ Semelhantes às String resources
- ▶ Também localizados em `/res/values`

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <color name="colorPrimary">#3F51B5</color>
4   <color name="colorPrimaryDark">#303F9F</color>
5   <color name="colorAccent">#FF4081</color>
6 </resources>
```

Nome


Valor

# A CLASSE R

- ▶ A classe R é um recurso muito interessante do Android para referenciar resources no código
- ▶ Ela é gerada automaticamente

# A CLASSE R

```
1  <resources>
2      <string name="app_name">Meu Projetoto</string>
3      <string name="app_botao">Pressione Aqui</string>
4  </resources>
```



```
public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        Button btnText = new Button( context: this);
        String text = getString(R.string.app_botao);
        btnText.setText(text);
        setContentView(btnText);
    }
}
```

# IMPLEMENTAÇÃO DA CLASSE R

```
/* AUTO-GENERATED FILE.  DO NOT MODIFY.
 *
 * This class was automatically generated by the
 * aapt tool from the resource data it found.  It
 * should not be modified by hand.
 */

package softblue.android;

public final class R {
    public static final class attr {
    }
    public static final class string {
        public static final int button_text=0x7f050003;
        public static final int hello_world=0x7f050002;
    }
}
```

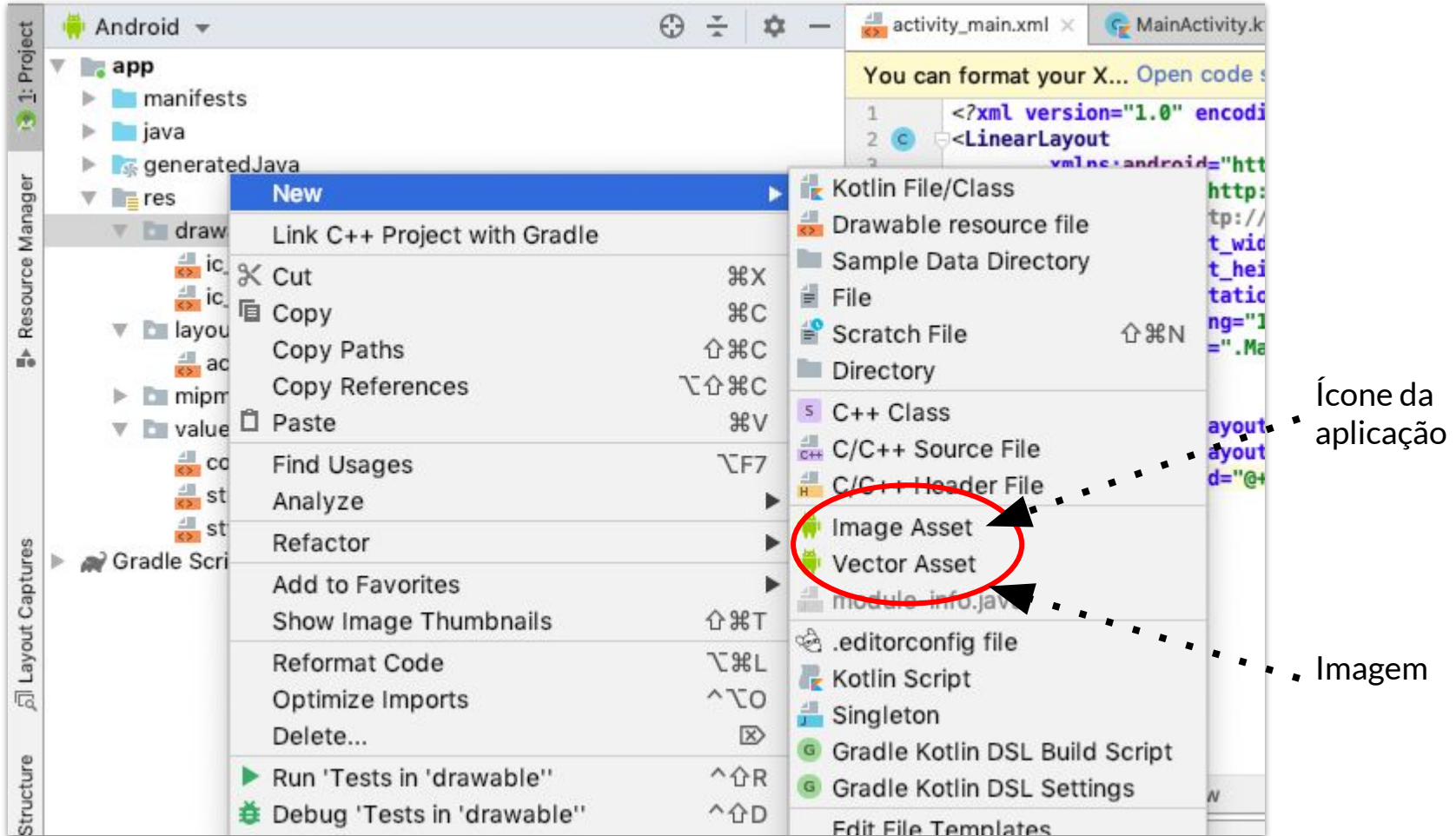


# RESOURCES DRAWABLE

- ▶ Resources que representam imagens
- ▶ Localizados no diretório /res/drawable
  - Extensões .png, .jpg, .gif e XML são suportadas



# RESOURCES IMAGE



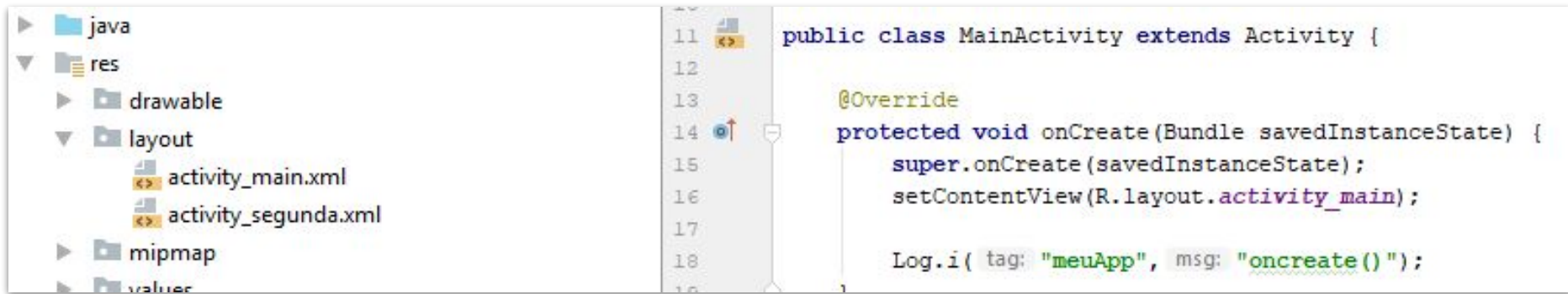
# LAYOUT RESOURCES

- ▶ Permitem representar o layout das telas da aplicação em formato XML
- ▶ A composição da interface fica separada do código-fonte da aplicação
- ▶ Utilizar arquivos de layout é a forma recomendada para criar a interface gráfica da aplicação
- ▶ Os layouts ficam localizados em **/res/layout**

# ARQUIVO DE LAYOUT

```
LinearLayout  TextView
1  <?xml version="1.0" encoding="utf-8"?>
2  C <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      android:orientation="vertical"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent">
7
8      <TextView
9          android:layout_width="match_parent"
10         android:layout_height="wrap_content"
11         android:text="Minha Aplicação!"
12         android:textSize="24dp"
13         android:textStyle="bold"
14     />
15
16     <Button
17         android:id="@+id/button"
18         android:layout_width="match_parent"
19         android:layout_height="wrap_content"
20         android:text="Pressione Aqui"
21         android:onClick="acao" />
22
23 </LinearLayout>
```

# APLICANDO UM LAYOUT



# APLICANDO UM LAYOUT

- ▶ A classe R é utilizada para referenciar resources no código
- ▶ Às vezes é necessário fazer a referência dentro de outro resource (definido em um XML)
  - Layout resources são exemplos típicos
- ▶ O Android possui uma convenção para este tipo de referência

```
15
16  <Button
17      android:id="@+id/button"
18      android:layout_width="match_parent"
19      android:layout_height="wrap_content"
20      android:text="@string/app_botao"
21      android:onClick="acao" />
22
```

# REFERENCIANDO RESOURCES

```
15
16  <Button
17      android:id="@+id/button"
18      android:layout_width="match_parent"
19      android:layout_height="wrap_content"
20      android:text="@string/app_botao"
21      android:onClick="acao" />
22
```



```
1  <resources>
2      <string name="app_name">Meu Projetoto</string>
3      <string name="app_botao">Pressione Aqui</string>
4  </resources>
5
```



# REFERENCIANDO RESOURCES

Resource	Dentro do Código	Dentro do XML
String	R.string.<nome_res>	@string/<nome_res>
Color	R.color.<nome_res>	@color/<nome_res>
Drawable	R.drawable.<nome_res>	@drawable/<nome_res>



# DEFININDO RESOURCES IDs

- ▶ Algumas vezes é necessário definir seus próprios IDs para resources para referência posterior dentro do código
  - Um exemplo típico é a recuperação de dados fornecidos pelo usuário na interface gráfica

# EXEMPLO DE DEFINIÇÃO DE ID

```
15
16 <Button
17     android:id="@+id/button"
18     android:layout_width="match_parent"
19     android:layout_height="wrap_content"
20     android:text="@string/app_botao"
21     android:onClick="acao" />
22
```

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

    Button botao = findViewById(R.id.button);
}
```

# SYSTEM RESOURCES

- ▶ Além dos resources que você define, o próprio Android tem alguns previamente definidos
- ▶ Os system resources são referenciados pela classe android.R

# SYSTEM RESOURCES



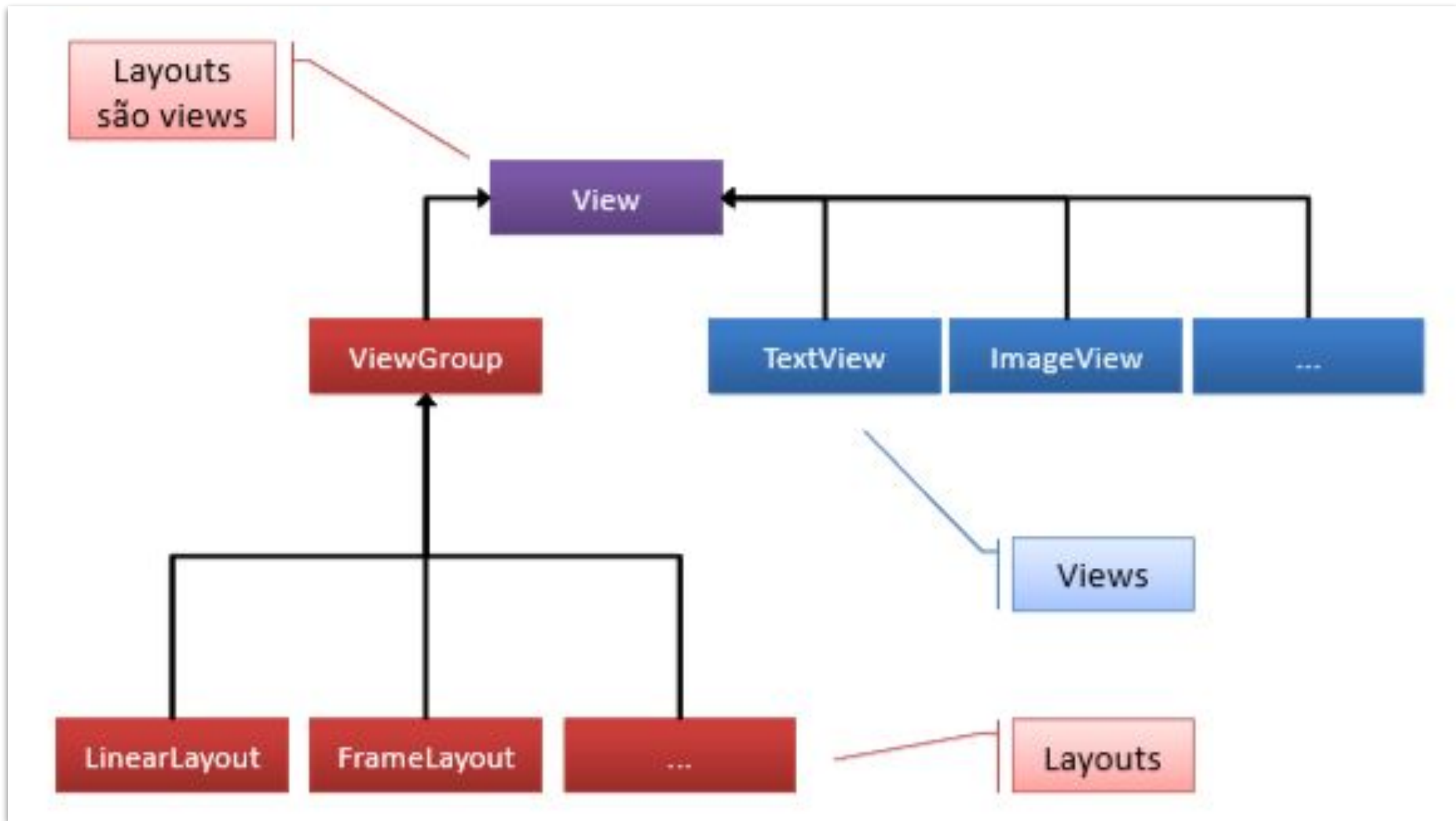
```
TextView textView = findViewById(R.id.textView);
textView.setText(android.R.string.yes);
```



# INTERFACES GRÁFICAS - ANDROID

- ▶ As telas das aplicações são compostas por views
  - Views herdam de `android.view.View`
- ▶ As views são organizadas nas telas através de layouts
  - Layouts herdam de `android.view.ViewGroup`

# INTERFACES GRÁFICAS - ANDROID



# LAYOUTS DO ANDROID

- ▶ O Android possui diversos layouts importantes utilizados para organizar interfaces gráficas
- ▶ Localizados no pacote `android.widget`
- ▶ Exemplos
  - `ConstraintLayout`
  - `LinearLayout`
  - `RelativeLayout`
  - `FrameLayout`
  - `TableLayout`
  - `GridLayout`

Obrigado!  
**Dúvidas?**

Professor Emerson Alencar  
[emerson@imd.ufrn.br](mailto:emerson@imd.ufrn.br)