

**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO NORTE
Campus Natal - Central

Aperfeiçoando a Interface com o Usuário

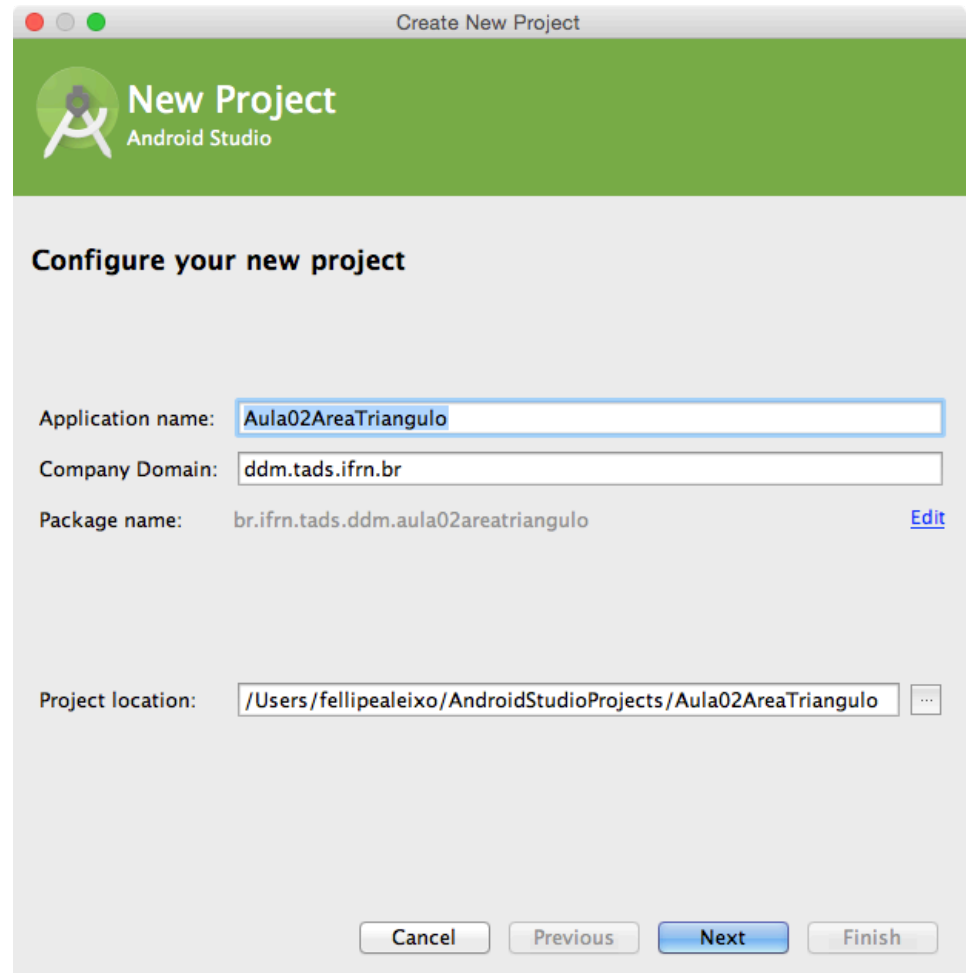
Prof. Fellipe Aleixo (fellipe.Aleixo@ifrn.edu.br)

Conteúdo

- Desenvolvimento de uma aplicação para calcular a área de um triângulo
- Utilização de Componentes de Interface
 - `TableLayout`, `TableRow`, `TextView`, `EditText`, `Button`
- Manipulação de Componentes no Código
- Tratamento de Eventos
 - `Listeners`

Criação do Projeto

- No início da IDE, selecione a opção “Start a new Android Studio project”
- Nomeie o projeto e configure o SDK



Interface activity_main.xml

- A interface gerada é apresentada abaixo:

```
<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=".MainActivity">

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

</RelativeLayout>
```



TableLayout

- Substitua o `RelativeLayout` por um `TableLayout` que organiza os componentes em forma de tabela
- Sete a propriedade `stretchColumns` do `TableLayout` para “*”, expandindo as colunas de cada linha em toda a largura da tela
- Remova o componente `TextView`

TableLayout

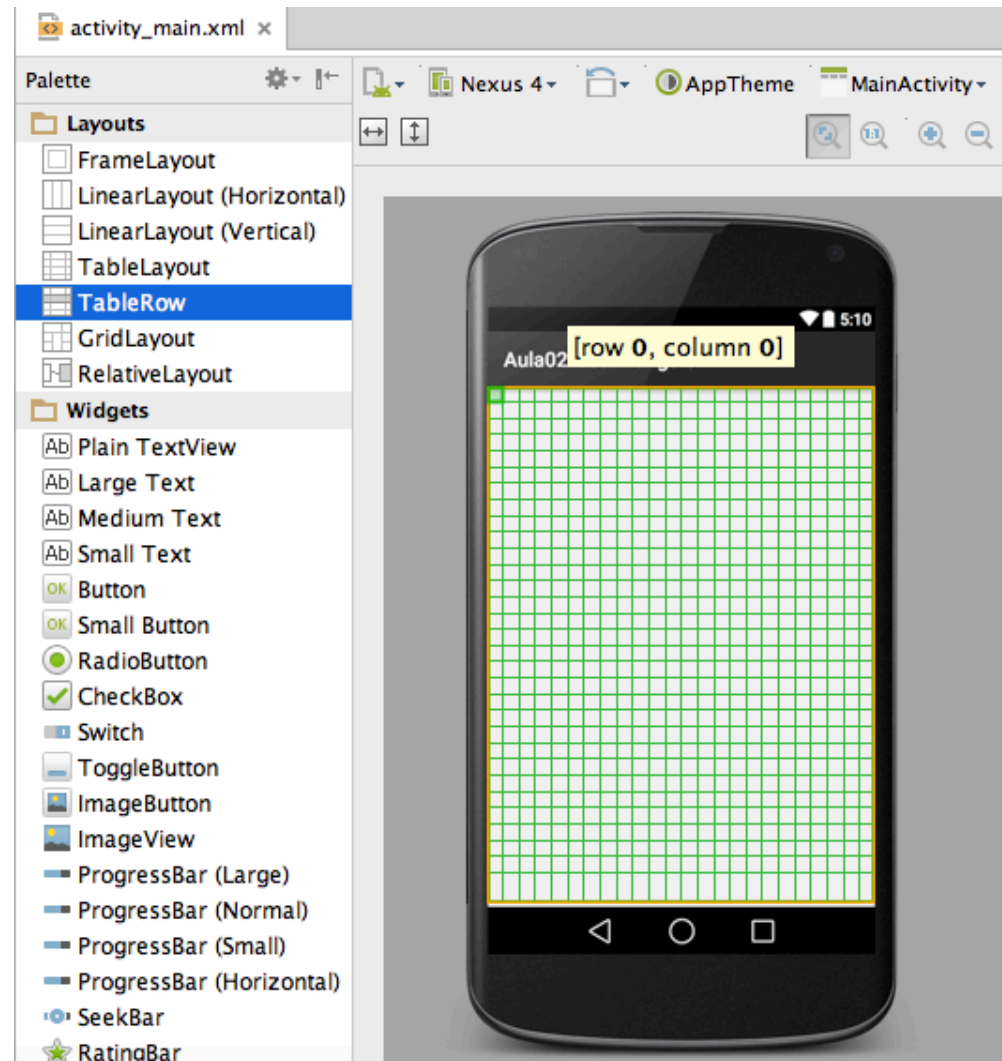
`<TableLayout`

```
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"  
android:stretchColumns="*" >
```

`</TableLayout>`

TableRows

- A partir da “palette” de componentes, adicione quatro **TableRows** um abaixo do outro



TableRows

```
<TableLayout
```

```
  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=".MainActivity"  
  android:stretchColumns="*"  
  android:id="@+id/tableLayout">
```

```
  <TableRow
```

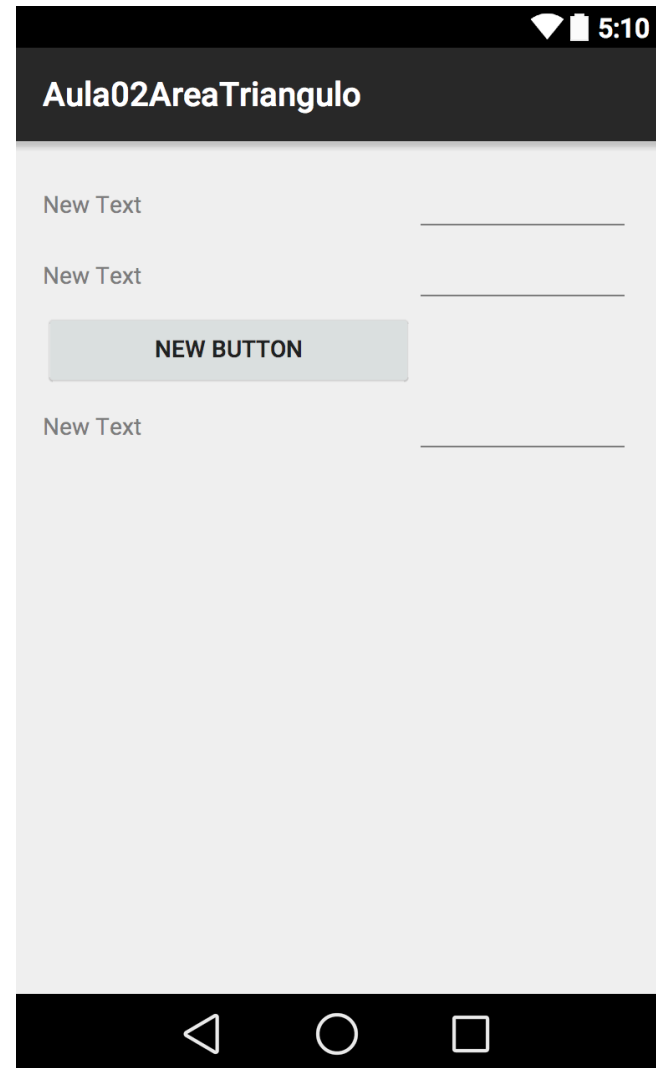
```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/tableRow1">
```

```
</TableRow>
```

```
...
```


TextView, EditText, Button

- Nas linhas 0, 1 e 3, insira
 - um Plain TextView e
 - um Plain Text (EditText)
- Na linha 2, insira um componente Button



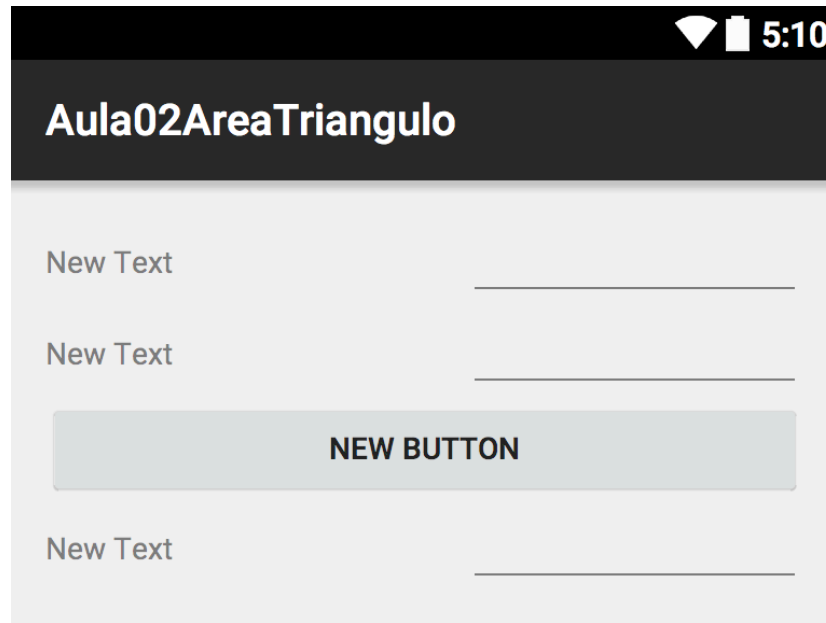
TextView, EditText, Button

- O XML da 1ª linha (**TableRow**):

```
<TableRow
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/tableRow2">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Text"
        android:id="@+id/textView2" />
    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/editText2" />
</TableRow>
```

Expandindo um Componente

- Um componente pode alongar-se por mais de uma coluna da linha, utilizando a propriedade `layout_span (= 2)`
- Altere esta propriedade no botão



Expandindo um Componente

```
<TableRow  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/tableRow3">  
  
    <Button  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="New Button"  
        android:id="@+id/button"  
        android:layout_span="2" />  
</TableRow>
```

strings.xml

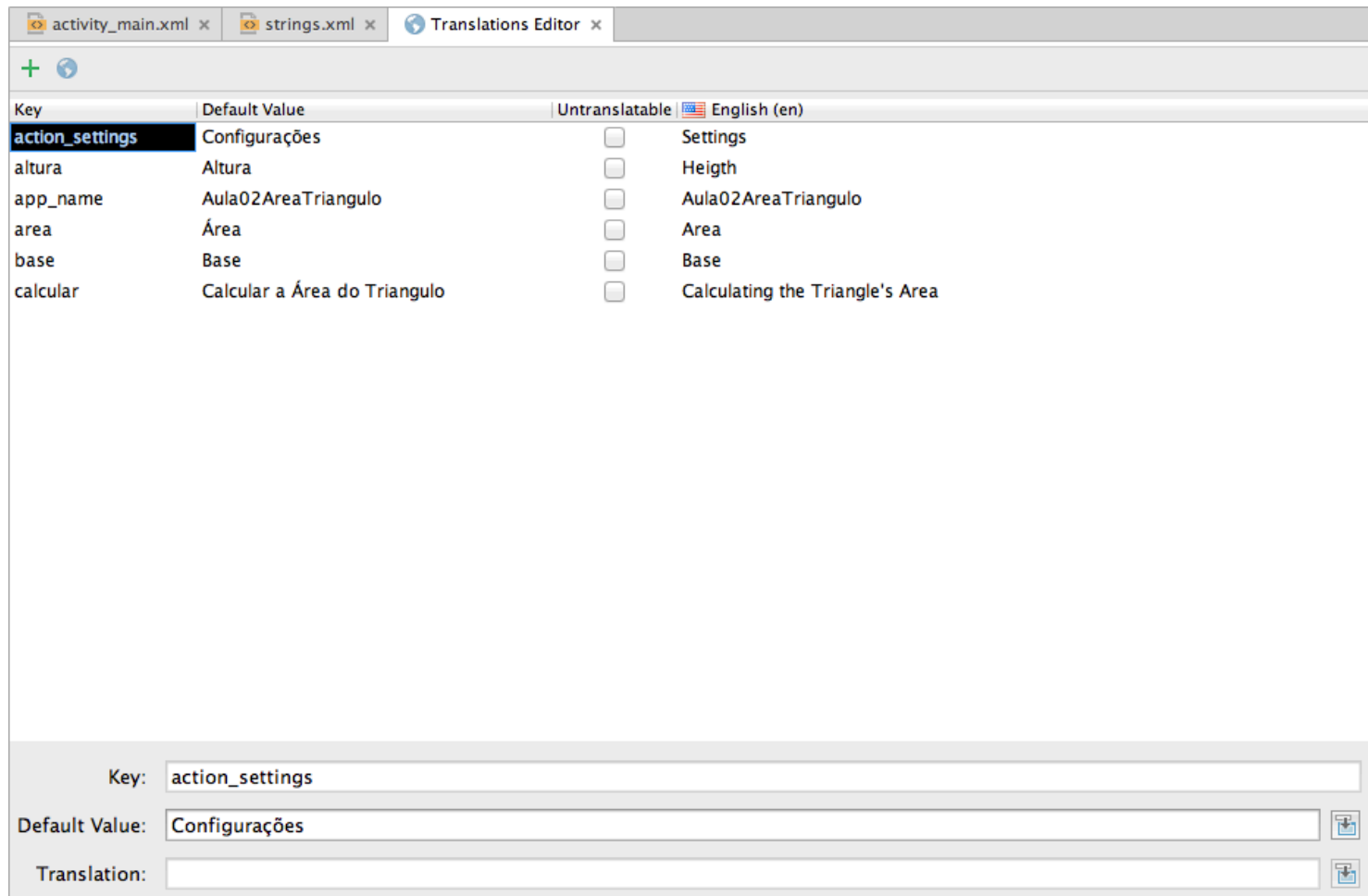
- No arquivo [strings.xml](#), defina as *Strings* a serem utilizadas na aplicação:

```
<resources>
  <string name="app_name">Aula02AreaTriangulo</string>
  <string name="action_settings">Settings</string>

  <string name="base">Base</string>
  <string name="altura">Altura</string>
  <string name="area">Área</string>
  <string name="calcular">Calcular</string>
</resources>
```

Edição de Traduções

- Fácil de “internacionalizar” a aplicação

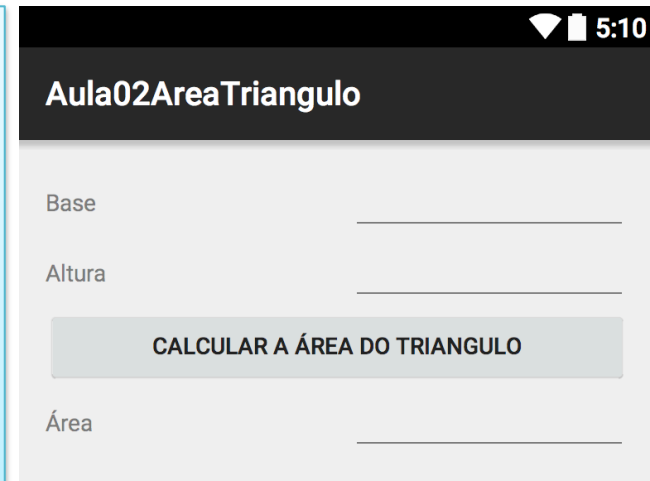


Utilizando as *Strings*

- No layout, substitua as propriedades **text** dos **TextViews** e do **Button**, referenciando as *Strings* no arquivo **strings.xml**

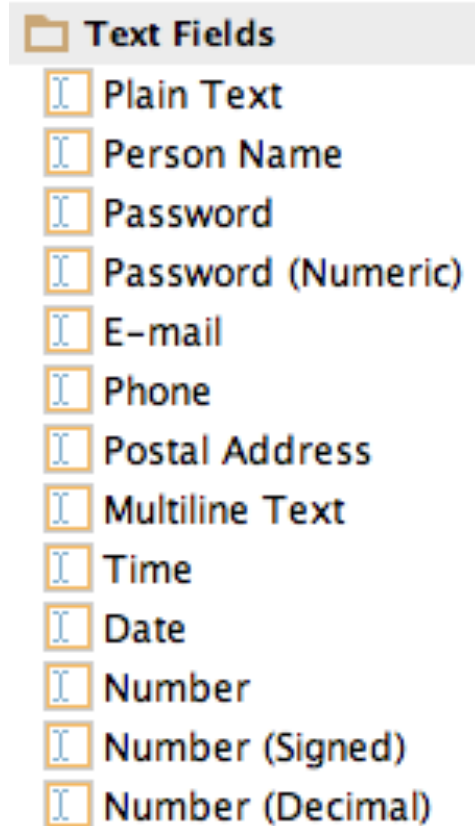
```
<TableRow
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/tableRow1">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/base"
        android:id="@+id/textView"
        android:layout_column="0" />
```



Propriedades do EditText

- Em todos os `EditTexts`, sete a propriedade `InputType` para `number`, uma vez que apenas números poderão ser entrados
- No `EditText` do resultado (Área), as propriedades `Focusable`, `LongClickable` e `Editable` devem ser falsas
- A interface está concluída!



Propriedades do EditText

```
<EditText  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:inputType="number"  
    android:id="@+id/editText3"  
    android:focusable="false"  
    android:longClickable="false"  
    android:editable="false" />
```

Componentes e Eventos

- Toda a programação do aplicativo é realizada na classe MainActivity.java
- Para referenciar os componentes, importar os pacotes que definem as classe dos componentes
 - `import android.widget.Button;`
 - `import android.widget.EditText;`
- Os eventos são manipulados com objetos **Listeners**
- Para programar o evento do botão, importar o pacote que define a classe **OnClickListener**
 - `import android.view.View.OnClickListener;`

Referenciando os Componentes

- Definir variáveis (atributos de MainActivity) para referenciar os componentes
 - `private Button button1;`
 - `private EditText editText1;`
 - `private EditText editText2;`
 - `private EditText editText3;`
- Recuperar as referências no método `onCreate`, utilizando o método `findViewById`
 - `button1 = (Button) findViewById(R.id.button1);`
 - `editText1 = (EditText) findViewById(R.id.editText1);`
 - `editText2 = (EditText) findViewById(R.id.editText1);`
 - `editText3 = (EditText) findViewById(R.id.editText1);`

Manipulação de Eventos

- Evento de clique em um botão
 - Instanciar um objeto `OnClickListener` e definir este objeto como listener (`setOnClickListener`)

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    button1 = (Button) findViewById(R.id.button1);  
    editText1 = (EditText) findViewById(R.id.editText1);  
    editText2 = (EditText) findViewById(R.id.editText2);  
    editText3 = (EditText) findViewById(R.id.editText3);  
    // Declara o "ouvinte" para o botão ...  
  
    button1.setOnClickListener(button1Listener);  
}
```

Objeto *Listener* do Botão

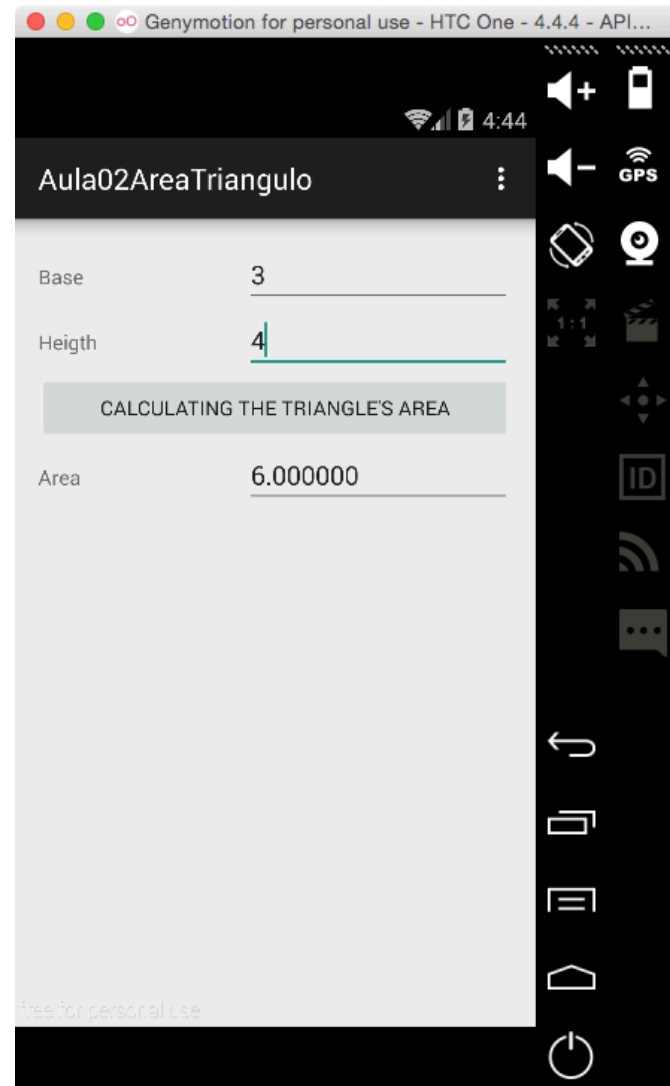
- O objeto `OnClickListener` invoca o método `onClick` quando o botão é clicado e as instruções no método são então executadas
- No exemplo:
 - o valor da base e altura são recuperados dos `editText1` e `editText2`, respectivamente
 - a área é calculada e apresentada no `editText3`

Objeto *Listener* do Botão

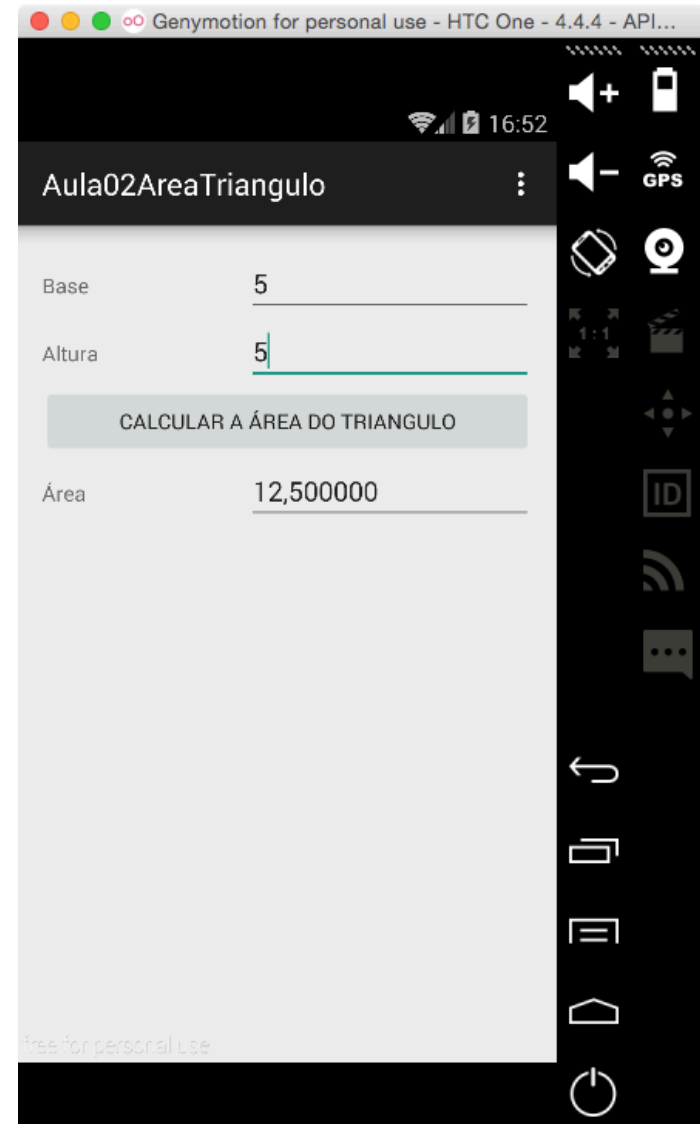
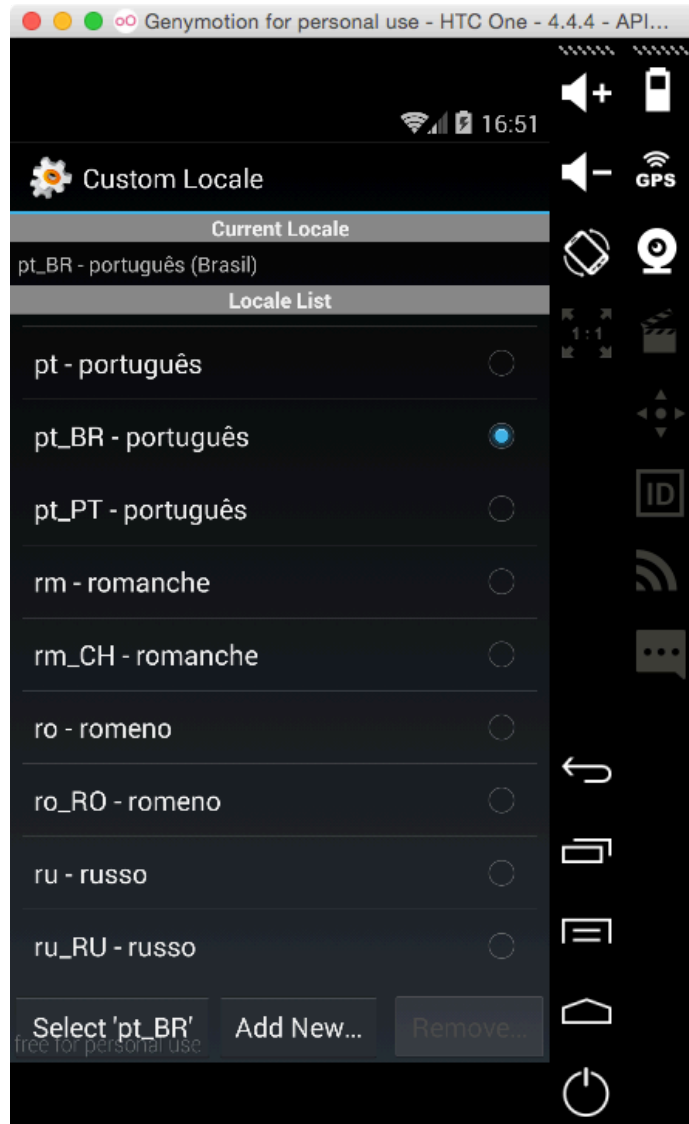
```
// Declara o "ouvinte" para o botão ...
OnClickListener button1Listener = new OnClickListener()
{
    @Override
    public void onClick(View v) {
        double base = Double.parseDouble(editText1.getText().toString());
        double altura = Double.parseDouble(editText2.getText().toString());
        double area = base * altura / 2;
        editText3.setText(String.format("%f", area));
    }
};
```

Execução da Aplicação

- No menu **Run**,
selecione a opção
Run 'app' (ou você
pode clicar no botão
correspondente)



Execução da Aplicação



Referências

- Android para Programadores – Uma abordagem baseada em aplicativos. Paul Deitel ... [et al.]. Bookman, 2013
- Google Android – Aprenda a criar aplicações para dispositivos móveis com o Android SDK. Ricardo R. Lecheta. Novatec, 2013