

Tratando eventos de clique

Transcrição

O formulário já tem campos para preencher, estrelinhas para dar notas, um botão de salvar e a possibilidade de rolar a tela se necessário. O próximo passo é adicionar ao botão de "Salvar" uma reação ao ser pressionado. Para modificar o comportamento dele vamos na classe *java* na *activity*, isto é, voltamos na aba *FormularioActivity.java*.

Primeiro, é necessário criar o botão para depois construir o *feedback* (a reação). O *setContentView* da *activity* pega o 'layout' e gera as instâncias que estão na *.xml*. Para trazer as instâncias podemos usar o *findViewById* e após o *setContentView* damos um "Enter" e acrescentamos o *Id*:

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

    findViewById()

}
```

Vamos aproveitar o momento e voltar na *activity_formulario.xml* e já adicionar um *Id* em cada um dos campos, uma vez que precisaremos deles mais adiante. Portanto, vamos retornar na aba *activity_formulario.xml*.

Vamos começar pelo primeiro *EditText* cujo campo é "Nome". Acrescentamos nele *@+id/* e para organizá-lo melhor adicionamos o local onde ele está, que é *formulario_nome*:

```
android:id="@+id/formulario_nome"
```

Ficaremos com o *EditText* da seguinte maneira:

```
<EditText android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Nome"
    android:id="@+id/formulario_nome" />
```

Esse procedimento é repetido com todos os demais campos, inclusive, com a *RatingBar* e com o *Button*. Ficaremos com:

```
<EditText android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Endereco"
    android:id="@+id/formulario_endereco"/>

<EditText android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Telefone"
```

```

        android:id="@+id/formulario_telefone"/>

<EditText android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Telefone"
    android:id="@+id/formulario_site"/>

<RatingBar
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:max="10"
    android:numStars="5"
    android:id="@+id/formulario_nota"/>

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Salvar"
    android:id="@+id/formulario_salvar" />

```

Agora, todos os componentes estão identificados!

Vamos voltar para a aba `FormularioActivity` e acrescentar um comando para buscar o botão, o `findViewById(R.id.formulario_salvar)`. Depois de fazer o `findViewById` é preciso guardar a informação que buscamos, assim, acrescentamos na frente de tudo o `Button` `botaoSalvar`. Teremos:

```
Button botaoSalvar = findViewById(R.id.formulario_salvar);
```

E o código ficará da seguinte maneira:

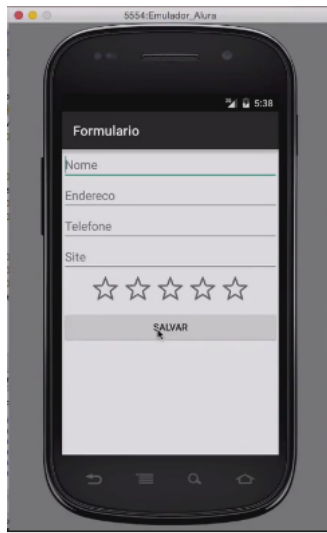
```

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

    Button botaoSalvar = (Button) findViewById(R.id.formulario_salvar);
}

```

Se o `'Button'` ficar assinalado em vermelho é porque falta importá-lo, nesse caso, basta dar um `"Alt+Enter"` em cima dele. Provavelmente ele continuará marcado com a cor vermelha, dessa forma, é preciso fazer o `cast`, assim, damos um `"Alt+Enter"` em cima do que ainda estiver em vermelho e selecionamos o `"Cast to android.widget.Button"`. Feito isso o botão passa a ter uma referência! Se rodarmos uma simulação vemos que o *Android* traz uma animação automática, como é possível perceber na imagem abaixo:



Vamos tratar o clique do botão, isto é, inserir alguma resposta quando o botão for clicado, como uma espécie de *feedback*. Para isso vamos utilizar um `Listener` que grosseiramente falando é como se fosse um "ouvinte". Com isso o *Android* avisa a quem estiver "escutando" que algo foi salvo. Para cadastrar adicionamos `botaoSalvar.setOnClickListener` e depois que adicionamos isso ele abre parenteses () e podemos usar o atalho "Comand+P" ou "Ctrl+P". Vamos utilizar aqui a ideia de classe anônima, portanto, adicionamos o `new OnClickListener`. Para fechar a instrução do `botaoSalvar` colocaremos o `;` no final. Ficará assim:

```
botaoSalvar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

    }
})
```

Falta completar o método `onClick` com a ação que ele deve desencadear ao ser pressionado. O objetivo é inserir uma mensagem que aparece e desaparece sempre que o botão for clicado. No *Android* isso é conhecido como *Toast*. Criaremos uma classe `Toast` que deve ser acompanhada do método `makeText`, este último pede alguns parâmetros, por exemplo, um contexto. O contexto serve para dizer ao *Android* quem somos, isto é, fornecer nossa identificação. No contexto a referência que queremos passar é `FormularioActivity` e junto disso adicionamos o `this`:

```
Toast.makeText(FormularioActivity.this);
```

Além disso, como queremos acrescentar uma mensagem escrevemos seu conteúdo dentro do parênteses, por exemplo: "Botao clicado!". É preciso definir por quanto tempo a mensagem será exibida e, por isso, usamos a constante `Toast.LENGTH_SHORT`. Ao terminar de digitar aparecerão duas sugestões automáticas: a `LENGTH_SHORT` e a `LENGTH_LONG`. Escolhemos `LENGTH_SHORT`. Teremos:

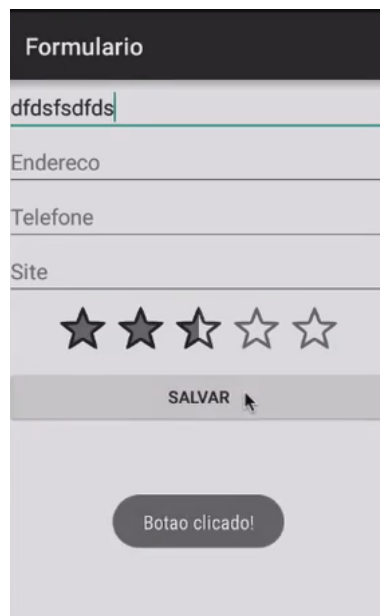
```
@Override
public void onClick(View v) {
    Toast.makeText(FormularioActivity.this, "Botao clicado!", Toast.LENGTH_SHORT);
```

Até o momento pedimos que a instância `Toast` fosse criada, agora, falta pedir que ela seja mostrada, então, acrescentamos "show" após o parênteses. Teremos:

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

    Button botaoSalvar = (Button) findViewById(R.id.formulario_salvar);
    botaoSalvar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick (View v) {
            Toast.makeText(FormularioActivity.this, "Botao clicado!", Toast.LENGTH_SHORT).show();
        }
    });
}
```

Vamos rodar o emulador para verificar como ficou!



Agora temos um feedback no botão!