

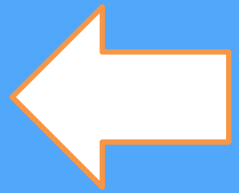


Componentes visuais

Parte III – A nova calculadora do mal

Mário Carvalho

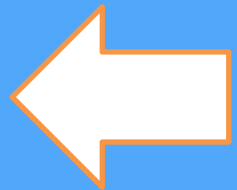
UFMS - CPPP



Conteúdo...



- 1 - Entendendo melhor a comunicação XML e Java**
- 2 - Formatação: Background, textColor, Input**
- 3 - Múltiplos layouts: LinearLayout dentro de outro**
- 4 - AlertDialog: Testando e entendendo**



1º passo:



Definir as variáveis no XML

```
<TextView
```

Tipo

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

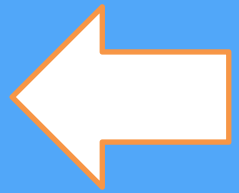
ID

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:textColor="#000000"
```

```
    tools:text="Meu primeiro APP" />
```



2º passo:



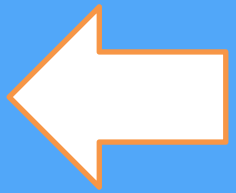
Ligar as variáveis do XML ao Java na Activity;

```
EditText txtNome;  
Button btnClique;
```

Criação - Java

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_principal);  
  
    txtNome = (EditText) findViewById(R.id.txtEntrada);  
    btnClique = (Button) findViewById(R.id.btnEntrar);  
}
```

Referência - XML



Background



- **Conceito:** Mudar a cor de fundo dos componentes.

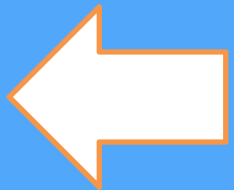
OBS: Serve para quase todos os componentes

Ex: `android:background="#00FFEE"`

As cores devem ser em hexadecimal:

Regras:

- Começar com #.
- Ter cerca de 6 caracteres, variando de: 0-9 e A-F



textColor



- **Conceito:** Mudar a cor dos textos exibidos.

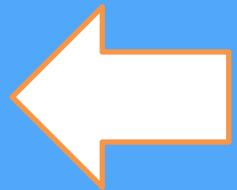
OBS: Visível em TextView, EditText, Button, etc.

Ex: `android:textColor="#00FFEE"`

As cores devem ser em hexadecimal:

Regras:

- Começar com #.
- Ter cerca de 6 caracteres, variando de: 0-9 e A-F



inputType



- **Conceito:** Definir o tipo de entrada das caixas de entrada.

OBS: EditText

Ex: `android:inputType="number"`

Poder ser definido para receber:

Números

Datas

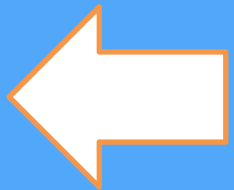
Textos

Emails

Número decimal

Número de telefones

Etc...



Layouts



- **Conceito:** As vezes será necessário usar as características de outro tipo de layout, que não é o layout pai.

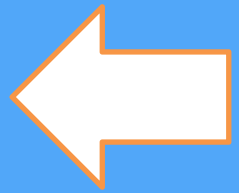
Solução: Trabalha com multiplos layouts.

Ex:

```
<LinearLayout
    android:layout_width="wrap_content"
    android:orientation="horizontal"
    android:layout_height="wrap_content">
    <EditText
        android:id="@+id/txtNumero22"
        android:hint="Ex: 5"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/btnSomar2"
        android:text="Somar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
```

AlertDialog



- **Conceito:** É uma caixa de texto que é exibida na tela
- Obs: Será necessário criar um o AlertDialog e iniciar.

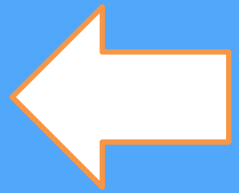
Para usar será necessário passar 3 parâmetros :

Contexto: `getApplicationContext()`

Título: String - vai ser o título

SMS: String - vai ser o corpo da caixa de dialogo

```
AlertDialog.Builder alerta = new AlertDialog.Builder(getApplicationContext());  
alerta.setTitle("Título");  
alerta.setMessage("Texto da caixa de diálogo");  
alerta.setNeutralButton("OK", null);  
alerta.create().show();
```

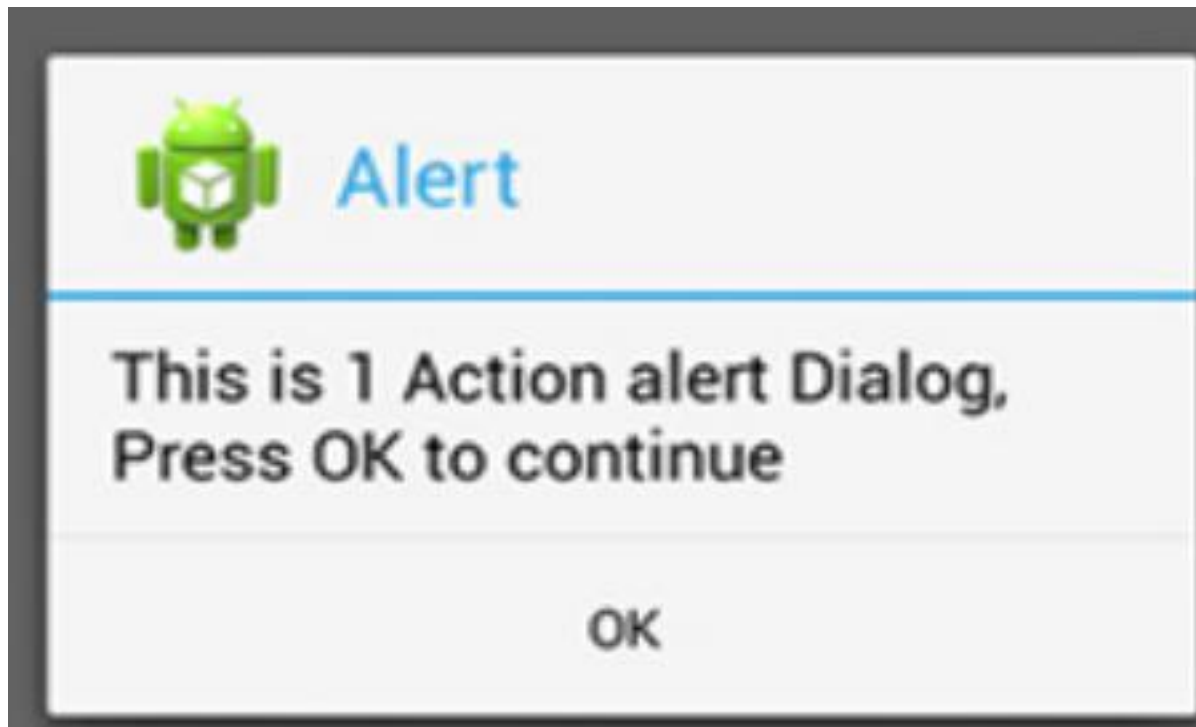


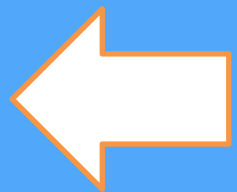
AlertDialog



- **Conceito:** O AlertDialog fornece um feedback fixo na tela.

Resultado:





E agora?



Macho, bóra lá.
É hora dos codes,
Birl....

