Criando Interfaces

Weslei Prudencio

Fundamentos

* Views

* View Groups

  * Agrupa outras Views

* Programaticamente

* XML

Programaticamente

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

    TextView myTextView = new TextView(this);
    setContentView(myTextView);
    myTextView.setText("Hello, Android");
}
```

XML

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

</RelativeLayout>
```

XML

- Similar ao HTML

- Recomendado pelo Google

- Altamente customizável

- Desempenho

Dimensões

- px (screen pixels)

- in (physical inches)

- pt (physical points)

- mm (physical millimeters)

- dp (density-independent pixels) ✓

- sp (scale-independent pixels) ✓

Dimensões

- match_parent

- wrap_content

- ~~fill_parent~~

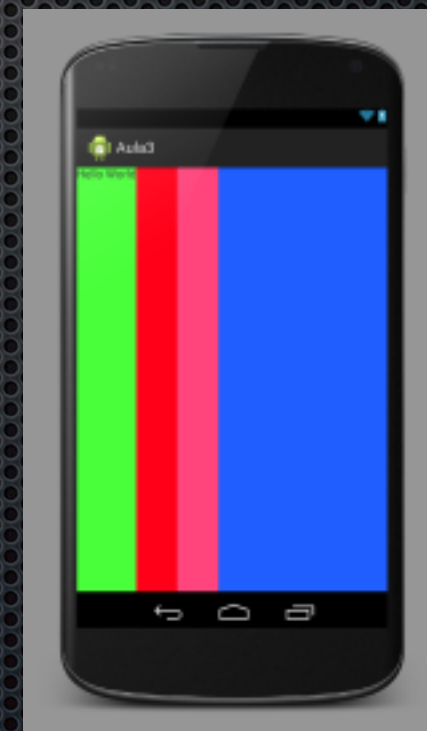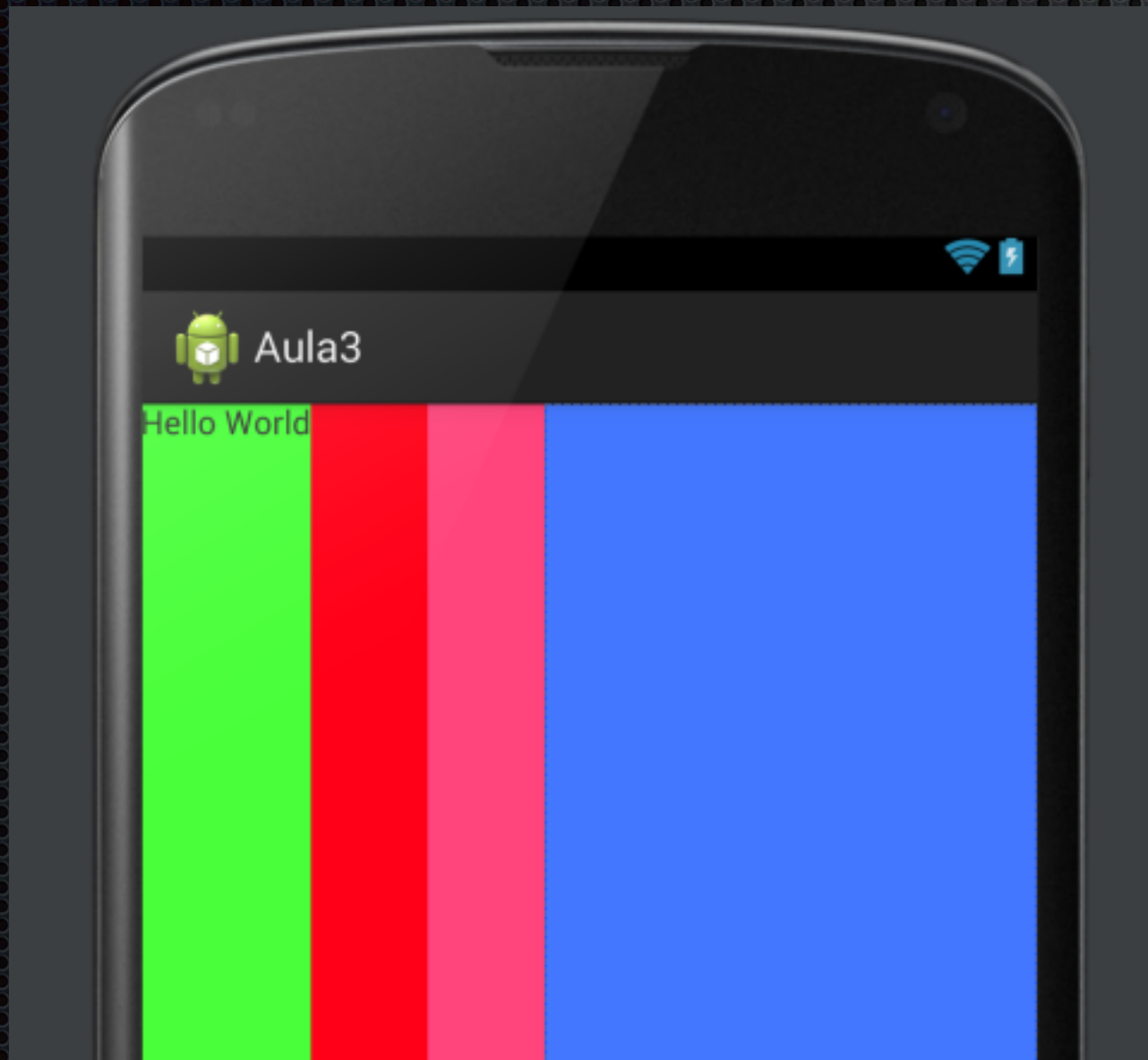# Layouts

- LinearLayout

- RelativeLayout
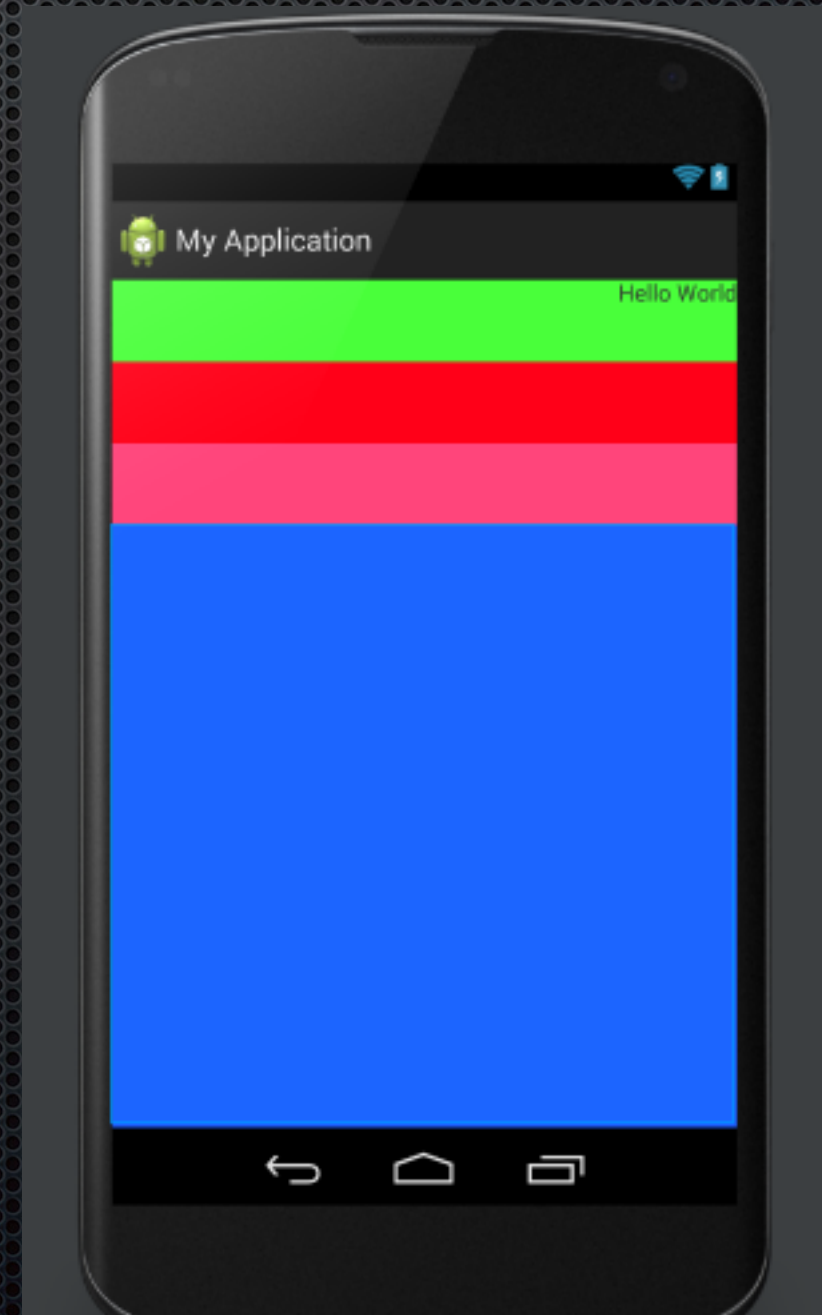
- FrameLayout

- ~~AbsoluteLayout~~

LinearLayout

- Componentes lado a lado (fila)

- Alinhamento horizontal ou vertical

- Gravity define o alinhamento

Exercício LinearLayout

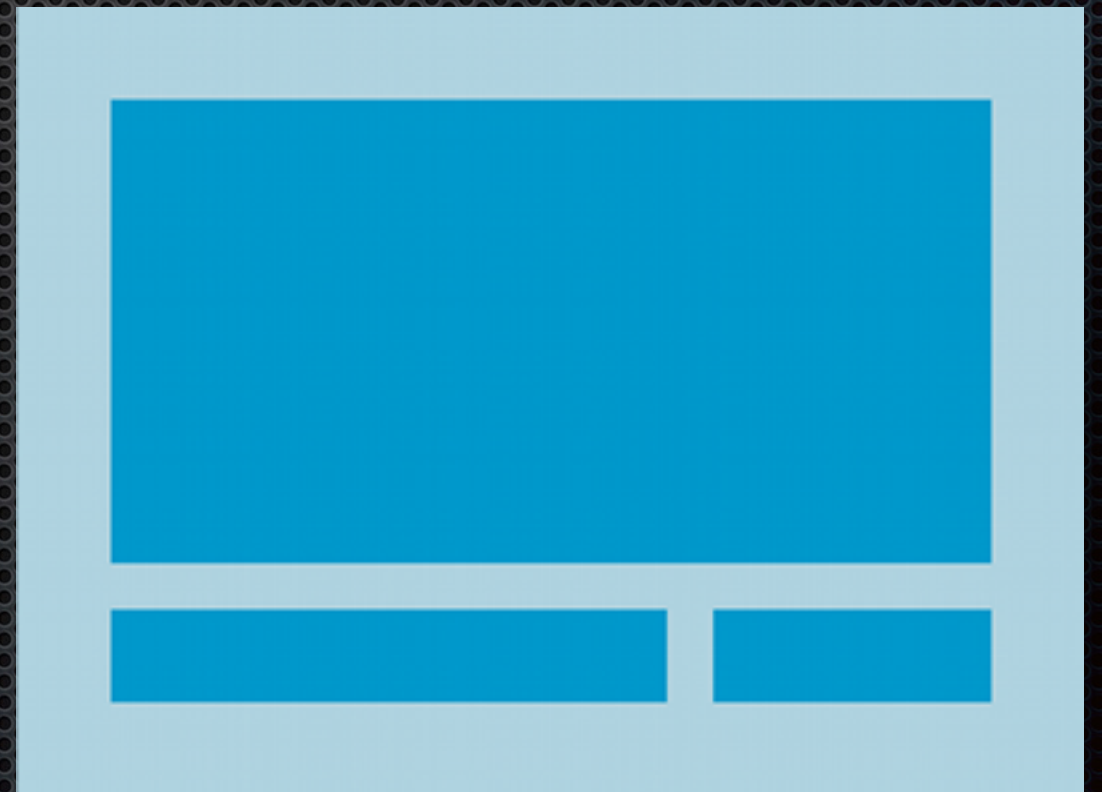# Exercício LinearLayout

- android:orientation?!

RelativeLayout

- Regras de posicionamento

- Baixa hierarquia de Layout

- Portabilidade ✓

RelativeLayout

- android:layout_toRightOf="@+id/textView"
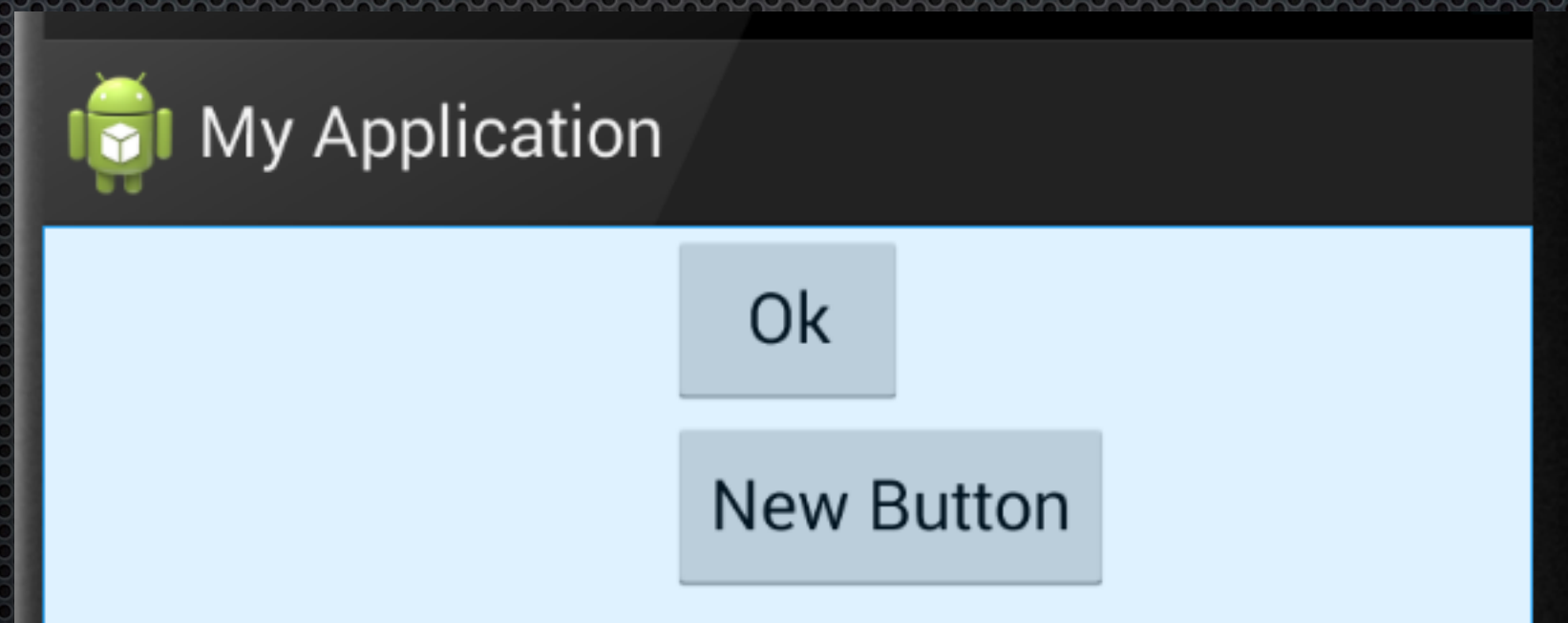
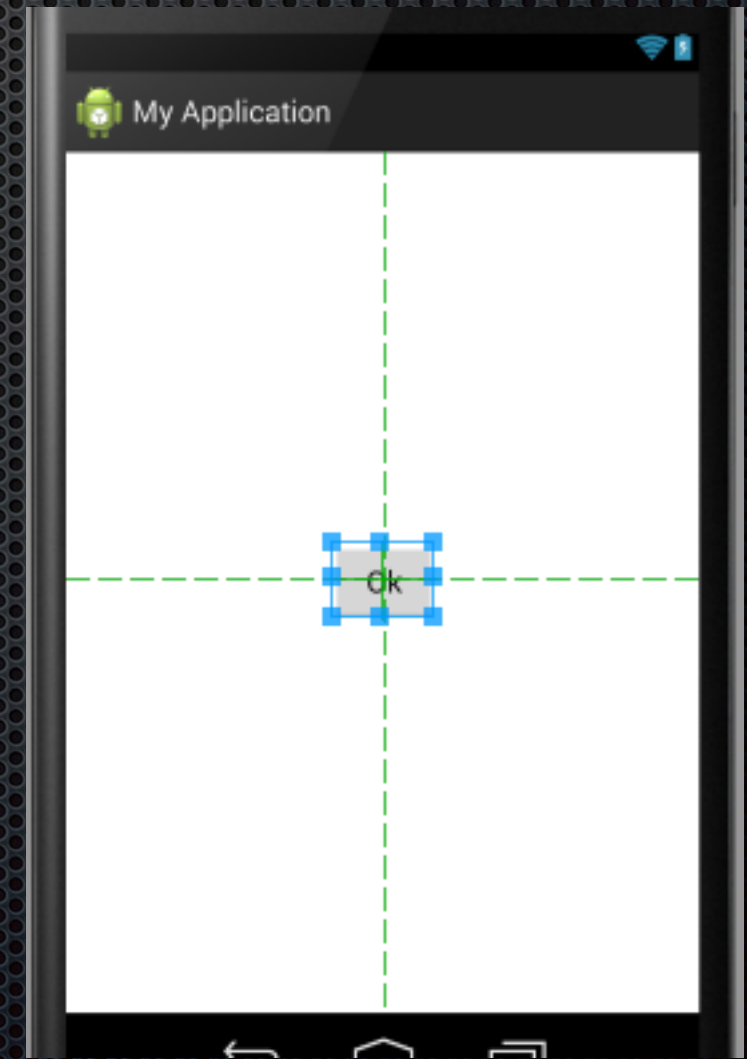# RelativeLayout

- android:layout_centerHorizontal="true"

RelativeLayout

* android:layout_below="@+id/button"
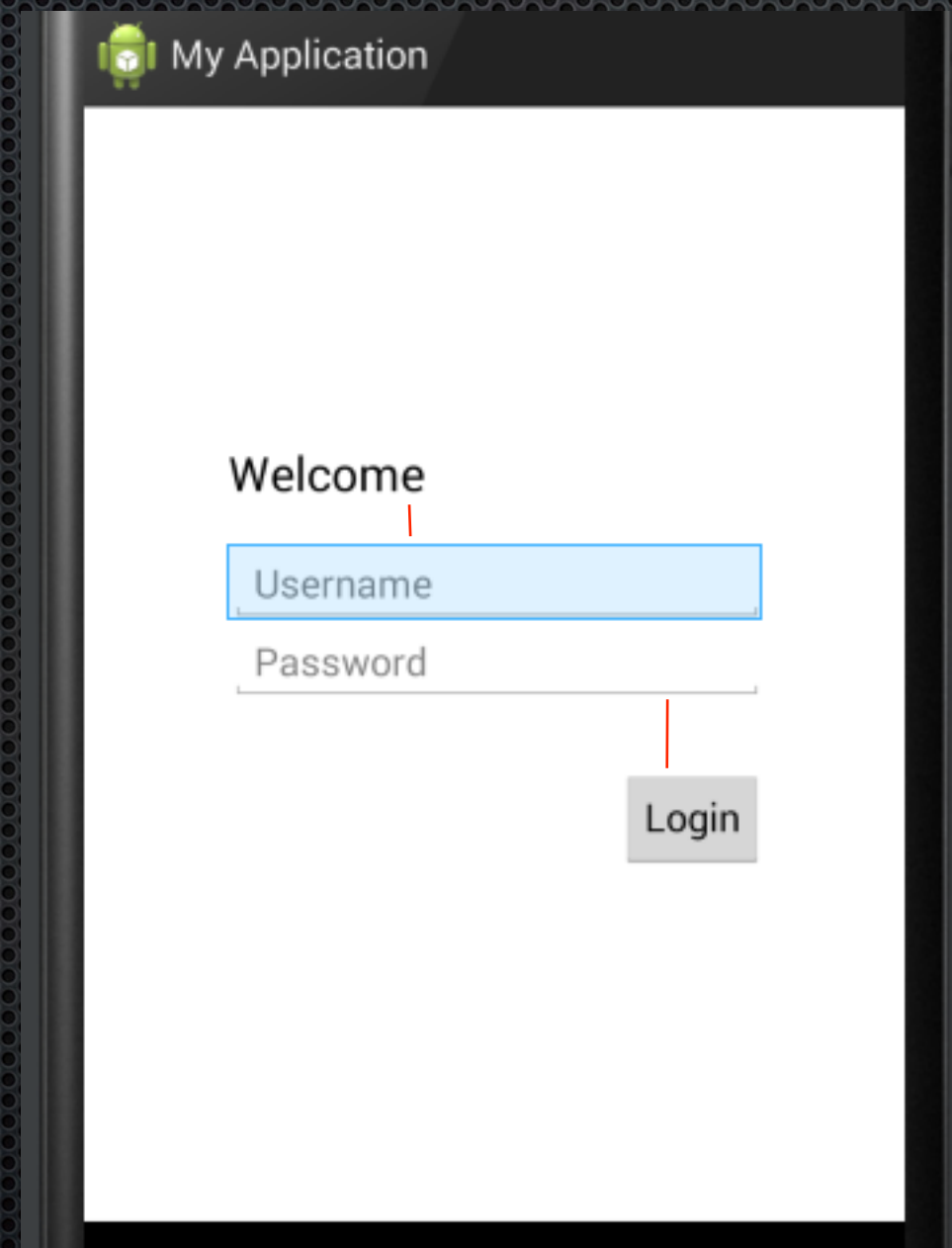android:layout_alignLeft="@+id/button"

RelativeLayout

- layout_centerVertical="true"
  layout_centerHorizontal="true"

Exercício RelativeLayout

FrameLayout

- Sobrepor Views

- Performático

Referências

* http://developer.android.com/reference/android/view/ViewGroup.LayoutParams.html

* http://developer.android.com/guide/topics/ui/layout/relative.html