

A08 Interface Gráfica

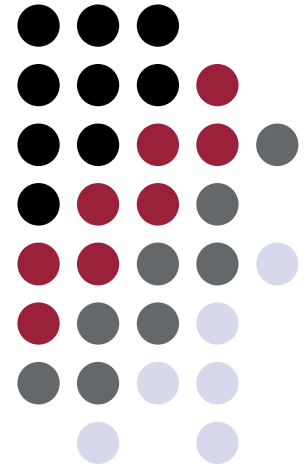
CSI436 – Computação Móvel

Prof. Dr. George H. G. Fonseca
Universidade Federal de Ouro Preto



UFOP

Universidade Federal
de Ouro Preto





- A base para representação de componentes de interface com o usuário
- Há dois tipos de componentes
 - Widgets - usados para criar componentes interativos (botões, campos de texto etc)
 - Gerenciadores de layout – definem a disposição dos widgets na tela
- A lista de métodos, propriedades e componentes é muito extensa.. cobriremos apenas os mais utilizados



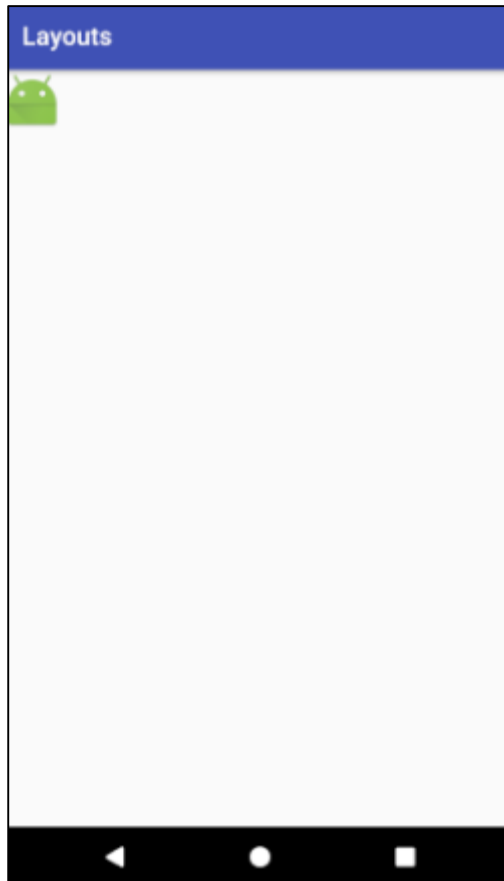


- **android:layout_height**
 - Especifica a altura do componente/view
- **android:layout_width**
 - Especifica a largura do componente/view
- Valores
 - **"match_parent"** O componente ocupa o tamanho definido por seu pai (layout)
 - **"wrap_content"** O componente ocupa apenas o tamanho necessário na tela
 - número Define numericamente o tamanho do componente, ex.:
"50px" "10dp" "5mm" "0.5in"

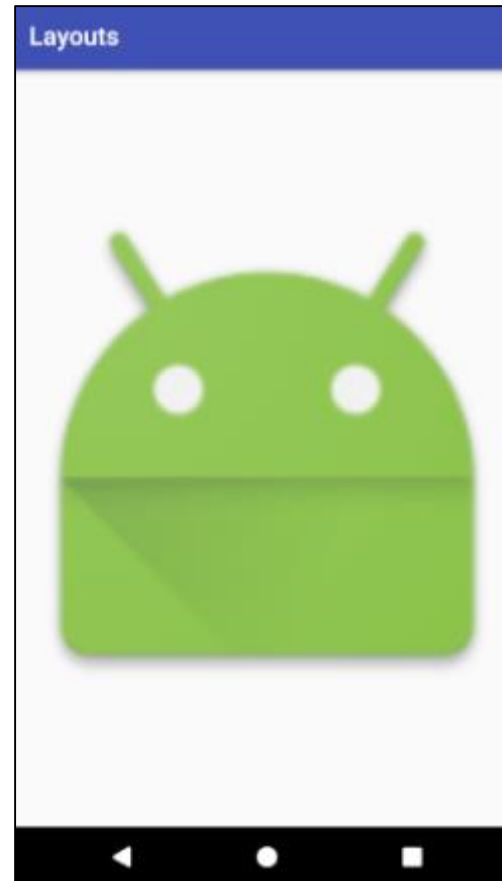
Gerenciadores de Layout



R.layouts.width_and_height



R.layouts.width_and_height2

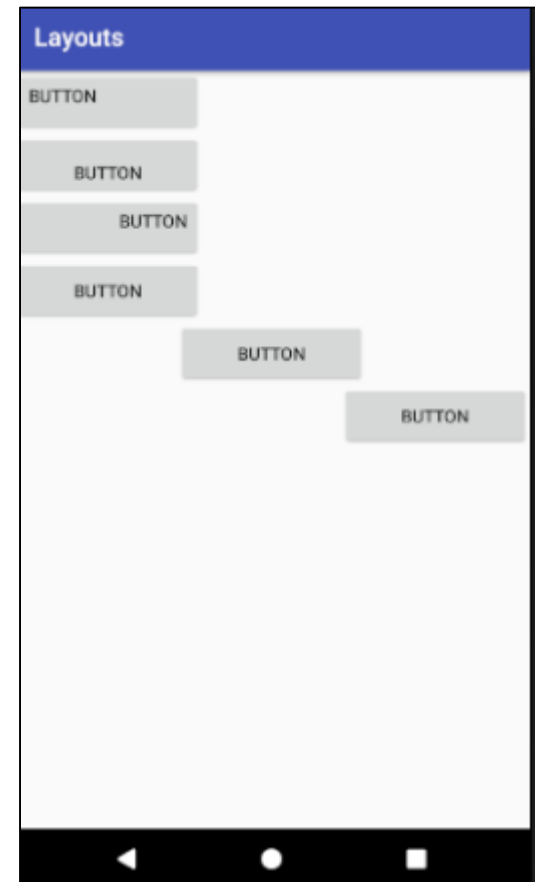




- **android:gravity** Define a gravidade do conteúdo da View
- **android:layout_gravity** Define o posicionamento da View no Layout

R.layouts.gravity

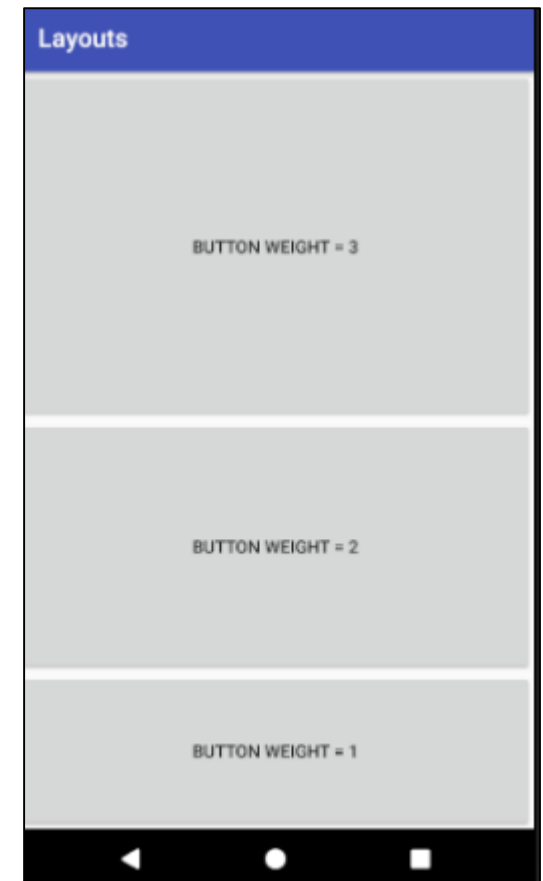
- Valores
 - top
 - center
 - bottom
 - left
 - right
 - center_horizontal
 - center_vertical
 - Podem ser combinados: “bottom|right”





- **android:layout_weight** Define o peso que o componente terá no layout
- O peso individual dividido pela somatória dos pesos determinará a relevância de cada componente

R.layouts.weight

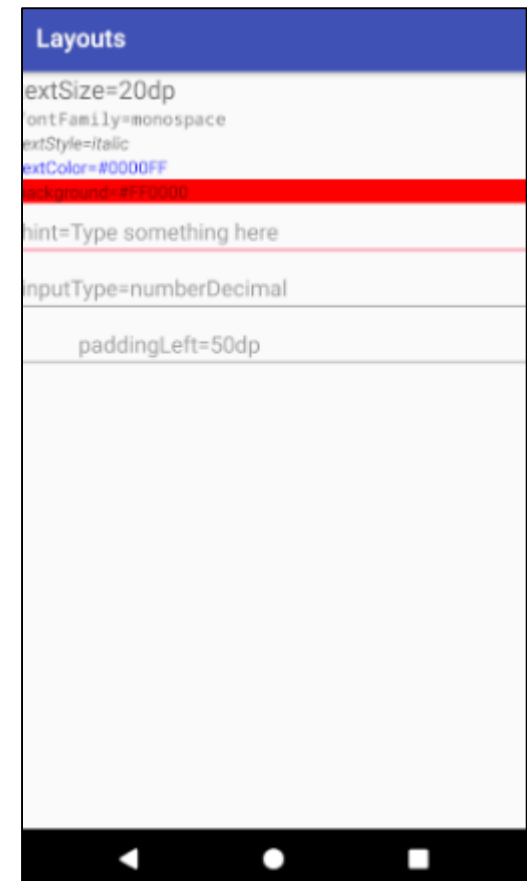


Properties



- Várias propriedades podem ser definidas para diversos tipos de Widgets
- O layout abaixo apresenta as mais comuns

R.layouts.properties





Classe	Descrição
TableLayout	Organiza os componentes em uma tabela, usando TableRow
FrameLayout	Utilizado por um componente único que normalmente preenche a tela inteira
LinearLayout	Organiza os componentes verticalmente ou horizontalmente
RelativeLayout	Posiciona os componentes uns em relação os outros, ex. componente a abaixo de componente b
ConstraintLayout	Introduzido na versão 2.3., é similar ao RelativeLayout, porém mais flexível

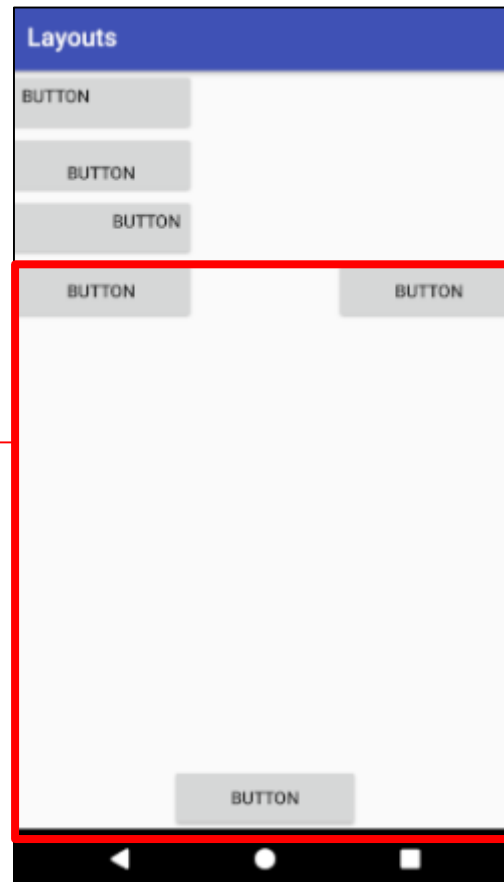
- Layouts podem ser livremente combinados!



- Linear layout com frame layout como filho

R.layouts.composed_layout

FrameLayout



Layout Composto



- Frame layout com filhos: linear layout e progress bar (centralizado)

R.layouts.composed_layout2

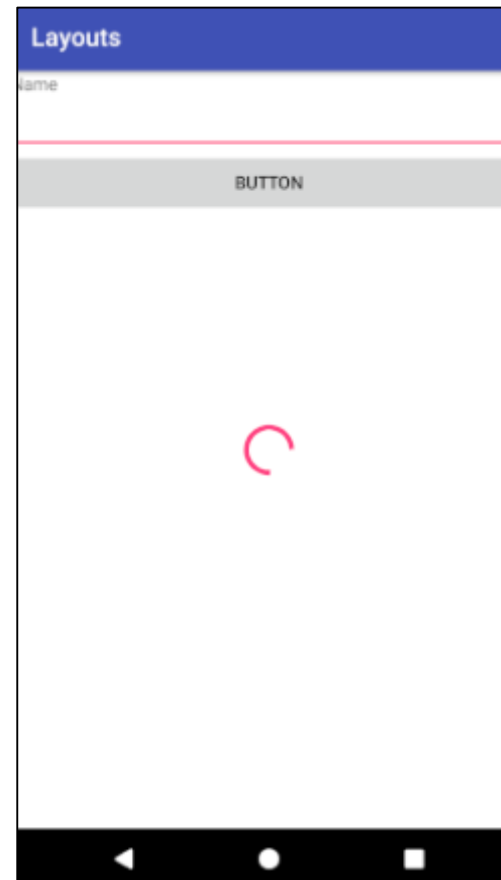


Table Layout



- Tem como filhos TableRow, cada qual definindo o conteúdo de uma linha da tabela
- **android:stretch_columns**
Define quais colunas devem ser esticadas caso haja espaço disponível
- Cada coluna é indexada de 0 a n

R.layouts.table_layout

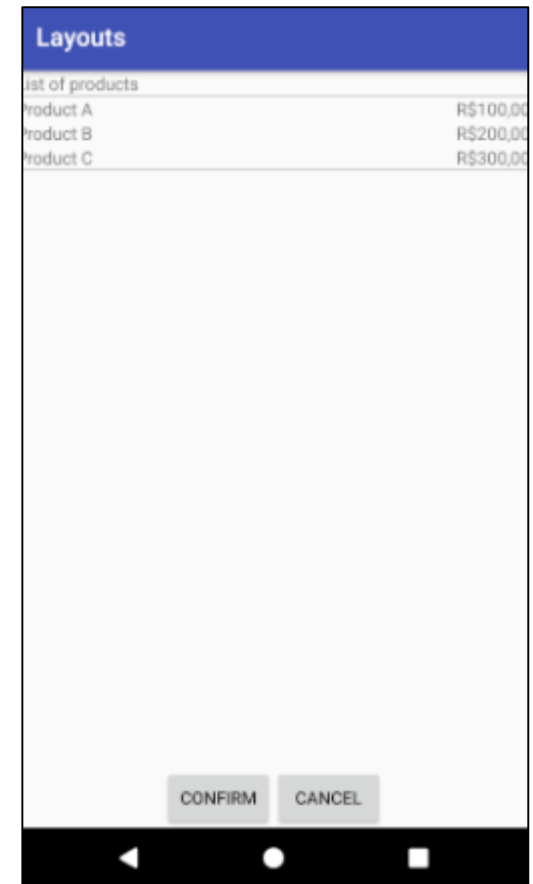
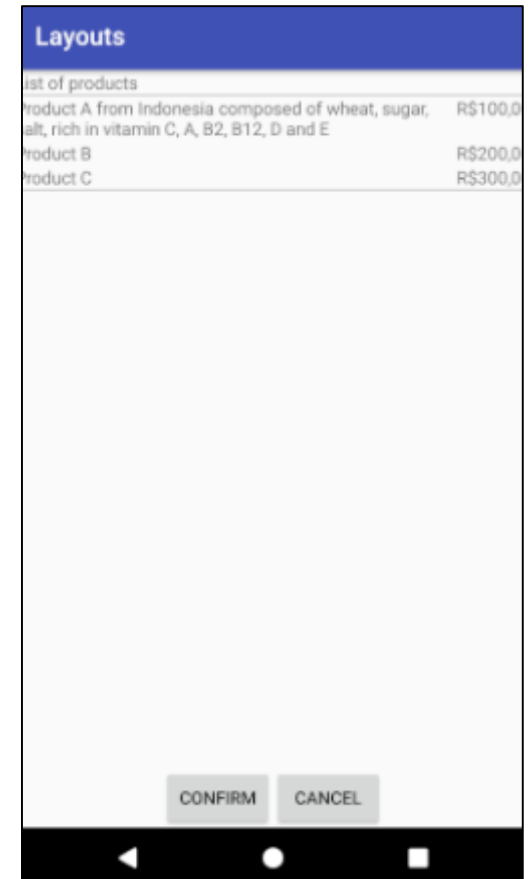


Table Layout



- **android:shrink_columns**

Define quais colunas devem ser continuadas na próxima linha para caber no espaço disponível



R.layouts.table_layout_shrink_columns

Relative Layout



- Define o posicionamento dos elementos baseado nas seguintes propriedades relativas:
 - `android:layout_toLeftOf`
 - `android:layout_toRightOf`
 - `android:layout_below`
 - `android:layout_above`

Posiciona elemento à esquerda/à direita /abaixo/acima de outro componente
 - `android:layout_alignParentLeft`
 - `android:layout_alignParentRight`
 - `android:layout_alignParentTop`
 - `android:layout_alignParentBottom`
- Alinha à esquerda/à direita /no topo/abaixo do comp. pai
- `android:layout_marginLeft`
 - `android:layout_marginRight`
 - `android:layout_marginTop`
 - `android:layout_marginBottom`
- Define espaço na margem esquerda /direita/superior/inferior do componente

Relative Layout



- Propriedades:
 - Day abaixo de Reminder
 - Day com margem de 20dp à esquerda
 - Day alinhado à esquerda no pai (layout)
 - Time abaixo de Reminder
 - Time alinhado à direita no pai (layout)
 - Time à direita de Day
-
- Confirm abaixo de Time
 - Confirm alinhado à direita no pai (layout)

R.layouts.relative_layout

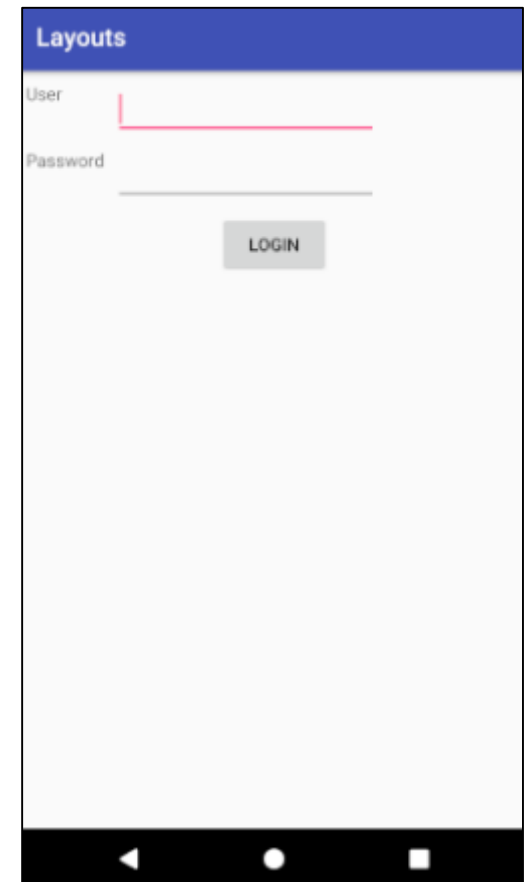


Constraint Layout



- Diversos tipos de restrições podem ser criadas
- Recomendável o uso do editor no modo Design
- Mais flexível que o RelativeLayout e mais fácil de usar com o editor de layouts

R.layouts.constraint_layout



Scroll View



- Define uma barra de rolagem que pode ser usada junto a qualquer layout



R.layouts.scroll_view



- Mudar setContentView() para ver os diferentes layouts!

A07 Layouts

A screenshot of an Android application interface. At the top is a blue header bar with the word "Layouts" in white. Below the header is a list titled "List of products". The list contains 25 items, each consisting of a product name and a price. The products are: Product A (R\$100,0), Product B (R\$200,0), Product C (R\$300,0), Product B (R\$200,0), Product C (R\$300,0), Product B (R\$200,0), Product C (R\$300,0), Product B (R\$200,0), Product C (R\$300,0), Product B (R\$200,0), Product C (R\$300,0), Product B (R\$200,0), Product C (R\$300,0), Product B (R\$200,0), Product C (R\$300,0), Product B (R\$200,0), Product C (R\$300,0), Product B (R\$200,0), Product C (R\$300,0), Product B (R\$200,0), Product C (R\$300,0), Product B (R\$200,0), Product C (R\$300,0), Product B (R\$200,0), and Product C (R\$300,0). The list is displayed on a white background with a thin horizontal line separating the header from the list. At the bottom of the screen is a black navigation bar with three white icons: a back arrow, a circle, and a square.

TabHost e TabWidget



- Definem juntos uma tela que contém abas
 - TabHost: contêiner para visualização de itens em abas
 - TabWidget: componente visual das abas
- Qualquer view, como uma imagem pode ser usadas para as abas
- O FrameLayout após TabWidget contém o conteúdo de cada aba
- Ideal para telas complexas com muita informação (ex. Whatsapp)





- Widgets são elementos visuais que compõem a tela, já conhecemos o TextView, EditText e o Button...
 - CheckBox
 - RadioButton
 - Spinner
 - AutoCompleteTextView
 - ToggleButton
 - RatingBar
 - SeekBar
 - Switch
 - ImageButton
 - ListView
 - Chronometer



A07 Widgets

CheckBox

☐ CheckBox 1

☒ CheckBox 2

RadioButton

☐ Male ☒ Female

Spinner

ICEA

DECSI

AutoCompleteTextView

ToggleButton

DESLLIGADO

RatingBar

SeekBar

Switch

Sound

SUBMIT

A07 Widgets2

ImageButton

Dialog

SHOW DIALOG

ListView

Acre

Alagoas

Amapá

Amazonas

00:00

START

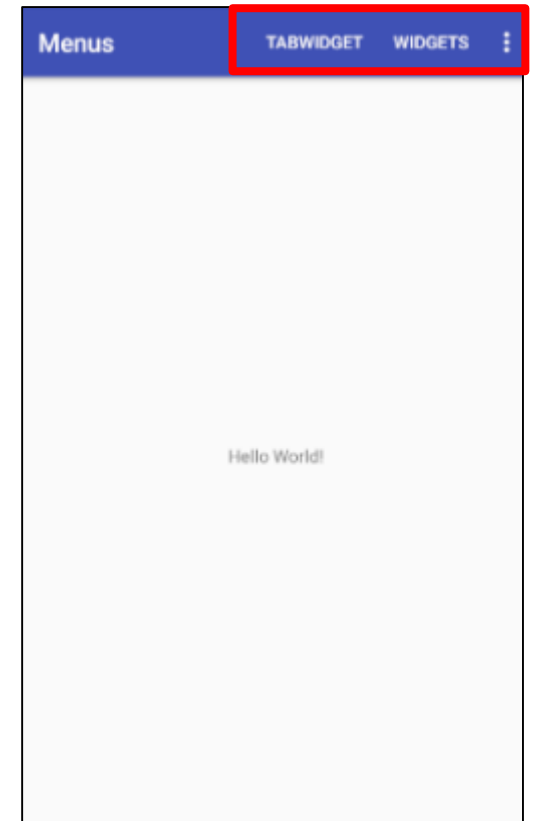
RESET



- Dialog é uma pequena janela que aparece ao usuário, comumente para tomar uma decisão ou prover informação adicional
- O Android provê por padrão o AlertDialog
- Porém dialogs ser personalizados via layout próprio



- Componente comum de interface com usuário com diversas aplicações





- Google and Open Handset Alliance n.d. **Android API Guide**. <http://developer.android.com/guide/index.html>. Acessado em Maio de 2017.
- Google and Open Handset Alliance n.d. **Android training guide**. <http://developer.android.com/training/index.html>. Acessado em Maio de 2017.
- Lecheta, R. R. **Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 3ª edição. São Paulo: Novatec Editora, 2013.

