

KivyMD

Live de Python #156

Roteiro



1. Material Design

Afinal, o que é um design system?

2. Widgets

Hoje, amanhã e sempre

3. Temas

Como deixar as coisas mais bonitinhas

4. Comportamento

Adicionando animações, elevações, mudanças dinâmicas e etc...







apoia.se/livedepython



PIX



Ajude o projeto



Ademar Peixoto, Alex Lima, Alexandre Harano, Alexandre Santos, Alexandre Souza, Alexandre Tsuno, Alysson Oliveira, Amaziles Carvalho, Anderson Araujo, Andre Rodrigues, André Rocha, Antonio Neto, Arnaldo Turque, Bianca Rosa, Bruno Oliveira, Caio Nascimento, Carlos Cardoso, Carlos Chiarelli, César Almeida, César Moreira, Davi Ramos, David Kwast, Diego Guimarães, Dilenon Delfino, Douglas Bastos, Edgard Sampaio, Elias Soares, Eugenio Mazzini, Everton Alves, Fabio Barros, Fabio Castro, Fabrício Coelho, Flavkaze Flavkaze, Franklin Silva, Fábio Serrão, Gabriel Simonetto, Gabriel Soares, Gabriela Santiago, Geandreson Costa, Guilherme Felitti, Guilherme Marson, Guilherme Ostrock, Haroldo Júnior, Henrique Machado, Hélio Neto, Isaac Ferreira, Israel Fabiano, Italo Silva, Jeison Sanches, Johnny Tardin, Jonatas Leon, Jorge Plautz, José Prado, Jovan Costa, João Lugão, João Schiavon, Juan Gutierrez, Jônatas Silva, Júlia Kastrup, Kaneson Alves, Leonardo Cruz, Leonardo Galani, Leonardo Mello, Lorena Ribeiro, Lucas Barros, Lucas Ferreira, Lucas Mello, Lucas Mendes, Lucas Teixeira, Lucas Valino, Luiz Lima, Maiguel Leonel, Maiguel Leonel, Marcela Campos, Marcelo Rodrigues, Maria Clara, Natan Cervinski, Nicolas Teodosio, Nilo Pereira, Otavio Carneiro, Patric Lacouth, Patricia Minamizawa, Patrick Gomes, Paulo Tadei, Pedro Andrade, Pedro Pereira, Peterson Santos, Reinaldo Silva, Rodrigo Campos, Rodrigo Ferreira, Rodrigo Vaccari, Ronaldo Silva, Rubens Gianfaldoni, Sandro Mio, Silvio Xm, Thiago Araujo, Thiago Borges, Thiago Bueno, Tyrone Damasceno, Valdir Junior, Victor Geraldo, Vinícius Bastos, Vinícius Ferreira, Wesley Mendes, Willian Lopes, Willian Rosa, Wilson Duarte



Obrigado você



Afinal, o que é material design?



Material design



MD é um design system. Pode ser comparado a uma documentação de padrões de design.

No nosso mundo, são basicamente descrições de como os componentes devem se comportar, qual aparência ele deve ter e como um componente deve se comportar

O kivyMD



KivyMD é uma coleção de widgets baseados em Material Design para o Kivy

pip install kivy kivymd

O ecossistema Buildozer KivyMD Kivy Pyinstaller **Biblioteca Widgets Instaladores**

O ecossistema **Aplicação** Buildozer KivyMD Kivy Pyinstaller **Biblioteca Widgets Instaladores**

O ecossistema Android / iOS **Aplicação** Buildozer KivyMD Kivy Windows / OSX Pyinstaller **Biblioteca Widgets Instaladores**

Buildando para mobile



Como vamos buildar para android hoje, devemos instalar as bibliotecas necessárias para isso

pip install buildozer cython

Estrutura inicial



Todo aplicativo em kivy tem uma estrutura mínima, onde definimos o nome do nosso aplicativo e o que será retornado como widget principal

```
-\square \times
    from kivymd.app import MDApp
2
3
    class MyApp(MDApp):
        def build(self):
             return None # Widget pai
6
8
   MyApp().run()
```

Hoje, amanhã e sempre

Wid gets



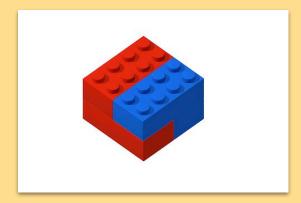
Os widgets são os componentes que compõem o design system.

Um botão é um exemplo de widget.

BUTTON



Um aplicativo kivy é na verdade um único widget que contém diversos widgets que por consequência possuem mais alguns widgets dentro deles.



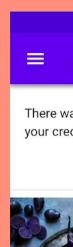


Vamos imaginar essa tela, por exemplo.

Ela é composta por diversos widgets:

- AppBar (A)
 - Menu button (1)
 - o Text (2)
 - Search button (3)
 - Menu Button (4)

• ...





There was a problem processing a transaction on your credit card.

FIX IT LEARN MORE





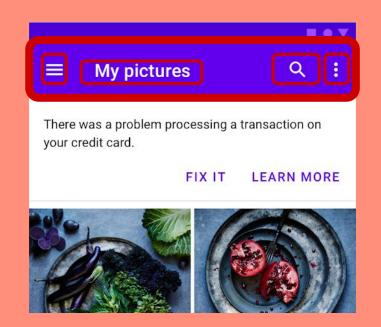


Vamos imaginar essa tela, por exemplo.

Ela é composta por diversos widgets:

- AppBar (A)
 - Menu button (1)
 - o Text (2)
 - Search button (3)
 - Menu Button (4)

• ...



Tipos de widgets



Existem diversas sub divisões sobre os tipos de widgets existentes no KivyMD, porém, vamos nos atentar ao fundamental:

- Widgets de layout
 - Basicamente são widgets feitos para posicionar widgets
- Widgets de interface
 - Widgets que tem alguma interação com a pessoa

Widgets de Layout











Linguagem para estruturar widgets

lang

Um KV



KVs podem ser escritos em arquivos com a extensão 'arquivo.kv' ou escrito em formato de string em um arquivo tradicional Python

Clica em mim

Nosso código para gerar aquele APP

```
from kivymd.app import MDApp
    from kivy.lang import Builder
    KV = '''
    FloatLayout:
        MDRaisedButton:
            text: 'Clica em mim!'
 9
10
     class MyApp(MDApp):
        def build(self):
12
13
             return Builder.load_string(KV)
14
15
    MyApp().run()
```

Clica em mim!

Tamanho dos widgets

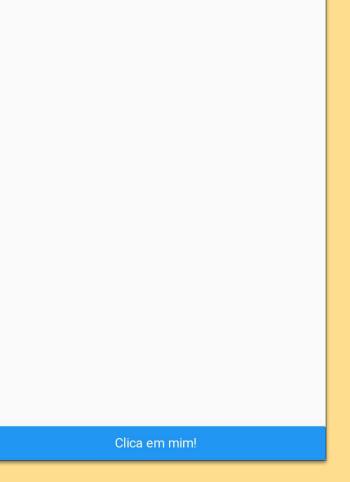


Os widgets tem seu tamanho default, mas isso pode ser um pouco inconveniente as vezes. Imagine que você queira que o botão ocupasse toda a horizontal, ou quase toda.

Podemos manipular o atributo **'size_hint_x**' do objeto, para termos o tamanho esperado

Tamanho dos widgets

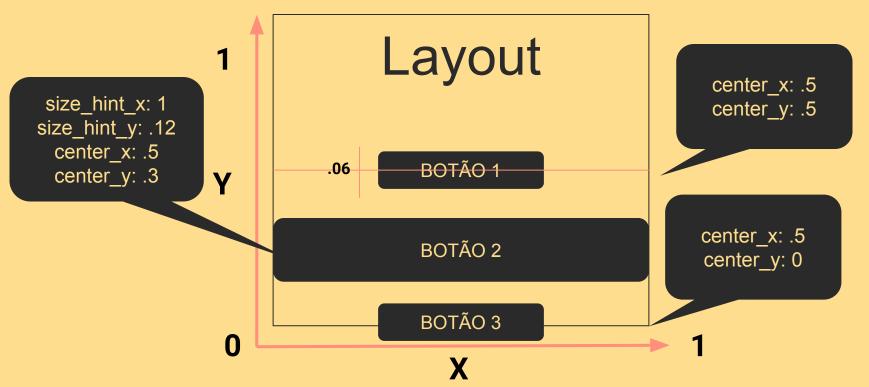
```
1 FloatLayout:
2 MDRaisedButton:
3 text: 'Clica em mim!'
4 size_hint_x: 1
```



Posicionamento dos widgets **WIDGET** DE LAYOUT

Posicionamento / Tamanho

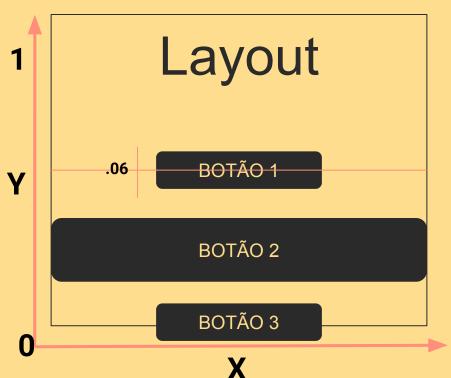




Posicionamento / Tamanho

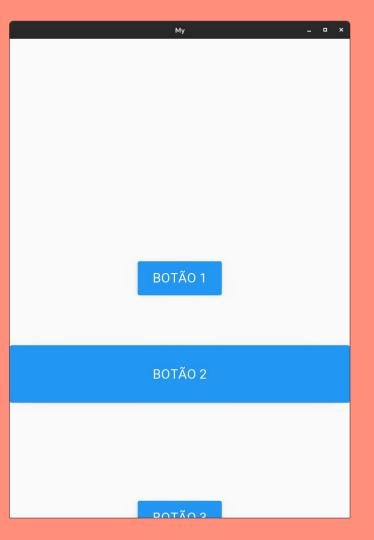


```
FloatLayout:
        MDRaisedButton:
            text: 'B0TÃ0 1'
            pos_hint: {'center_y': .5, 'center_x': .5}
        MDRaisedButton:
            text: 'BOTÃO 2'
            size_hint_y: .12
            size_hint_x: 1
            pos_hint: {'center_y': .3}
        MDRaisedButton:
10
11
            text: 'BOTÃO 3'
12
            pos_hint: {'center_y': 0, 'center_x': .5}
```



Tamanho / Posicionamento

```
FloatLayout:
        MDRaisedButton:
 3
            text: 'B0TÃ0 1'
            pos_hint: {'center_y': .5, 'center_x': .5}
 4
        MDRaisedButton:
            text: 'BOTÃO 2'
 6
            size_hint_y: .12
            size_hint_x: 1
 8
            pos_hint: {'center_y': .3}
10
        MDRaisedButton:
11
            text: 'B0TÃ0 3'
            pos_hint: {'center_y': 0, 'center_x': .5}
12
```



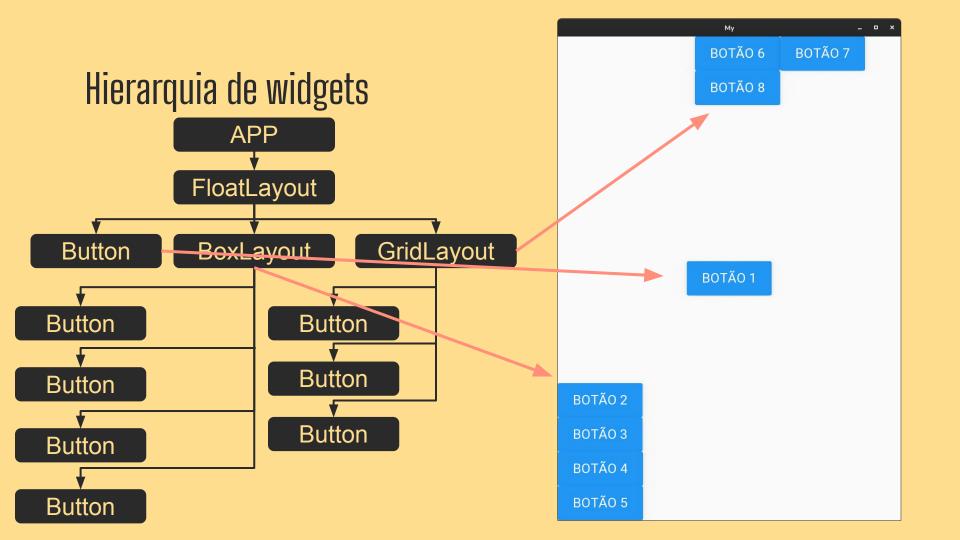
- 1. É possível posicionar layouts dentro de layouts
- 2. A posição é dada em relação ao wiget "superior"
 - a. center_x e center_y são o centro do widget filho e relação ao seu superior
- 3. O tamanho também
 - a. O tamanho é dado por size_hint (tupla com x e y), size_hint_y e size_hint_x





```
Hierarquia de widgets
               APP
            FloatLayout
                            GridLayout
 Button
             BoxLayout
Button
                     Button
                     Button
Button
                     Button
Button
Button
```

```
1 FloatLayout:
      MDRaisedButton:
           text: 'B0TÃ0 1'
           pos_hint: {'center_y': .5, 'center_x': .5}
      BoxLayout:
           orientation: 'vertical'
           MDRaisedButton:
               text: 'BOTÃO 2'
           MDRaisedButton:
               text: 'BOTÃO 3'
10
11
           MDRaisedButton:
12
               text: 'BOTÃO 4'
13
           MDRaisedButton:
               text: 'BOTÃO 5'
14
      GridLayout:
15
16
           cols: 2
17
           pos_hint: {'center_x': .9}
18
           MDRaisedButton:
               text: 'BOTÃO 6'
19
20
           MDRaisedButton:
21
               text: 'BOTÃO 7'
           MDRaisedButton:
               text: 'BOTÃO 8'
23
```



Parte 2



Bom, depois desse papo todo de layouts, você deve estar se perguntando sobre os componentes disponíveis e sim, temos uma vasta variedade no kivyMD.

São 55 grupos de widgets que vão de botões customizados até a players de vídeo prontos. Não tenho a intenção de cobrir todos aqui, só dar um pequeno passo aos mais populares

Grupo dos botões



Somente como exemplo, o grupo dos botões é composto de 12 componentes:

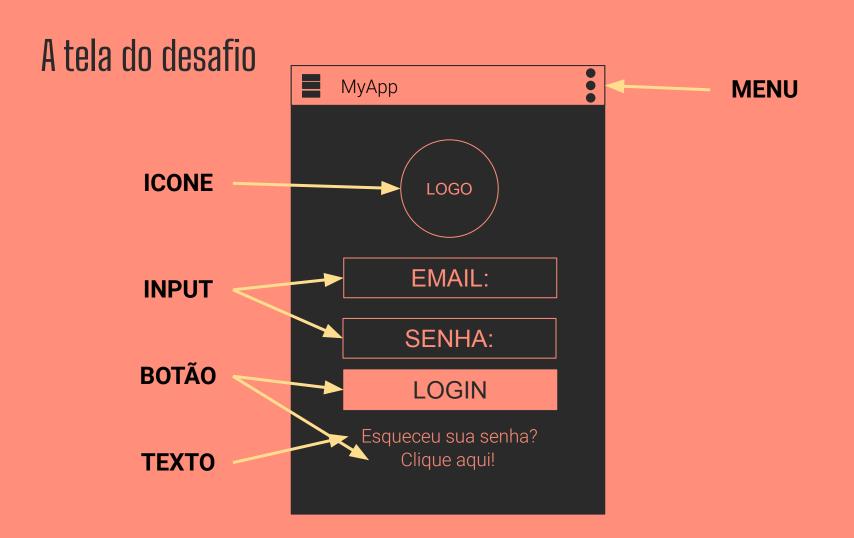
- IconButton
- FloatActionButton
- FlatButton
- RaisedButton
- RectangleFlatButton
- RectangleFlatIconButton
- RoundFlatButton

- RoundFlatIconButton
- FillRoudFlatButton
- FillRoudFlatIconButton
- TextButton
- FloatActionButtonSpeedDial

Minha ideia é fazer uma tela de login inicial, para que possamos conhecer os componentes principais do KivyMD







Deixando o bonito, mais bonito

lem

Temas (Matiz)



O material design nos fornece um esquema de cores que podemos usar nas nossas aplicações.



Trabalhando com a matiz



No kivyMD todos os recursos de temas podem ser modificados no atributo 'theme_cls' do MDApp

- primary_palette
 - Paleta principal
 - Defaulf Blue
- accent_palette
 - Paleta de contraste
 - Default Amber

```
1 class MyApp(MDApp):
2  def build(self):
3   self.theme_cls.primary_palette = 'Blue'
4   self.theme_cls.accent_palette = 'Amber'
5
6  return Builder.load_string(KV)
```

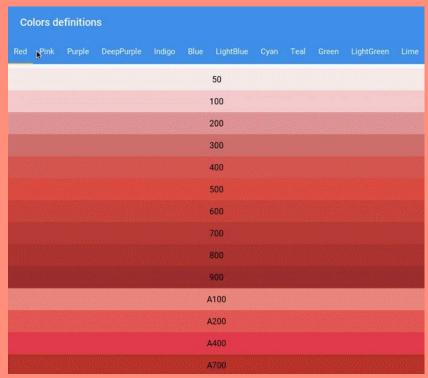
Saturação



O material design nos provê um sistema de saturação de cores que vai ser controlado pelos atributos:

- primary_hue
- accent_hue

Os valores vão de 50 a A700



Adicionando ações

Compor tamento