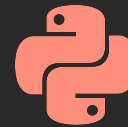




PySimpleGUI™

Live de Python #202



1. Sobre o PySimpleGUI

Uma apresentação da biblioteca

2. Widgets, Window e Layouts

O core da lib

3. Temas

Uma vasta e diversa lista

4. Sistema de eventos

Como executar as ações da interface

5. Playground

Bora fazer alguma tela, pq não?

6. Pontos de atenção

A hora de falar sério



picpay.me/dunossauro



apoia.se/livedepython



pix.dunossauro@gmail.com



Ajude o projeto <3



Acássio Anjos, Ademair Peixoto, Alexandre Harano, Alexandre Souza, Alexandre Takahashi, Alexandre Villares, Alex Lima, Alynne Ferreira, Alysso Oliveira, Ana Carneiro, Ana Padovan, Andre Azevedo, André Rocha, Aquiles Coutinho, Arnaldo Turque, Ayrton Santana, Bloquearsites Farewall, Bruno Barcellos, Bruno Freitas, Bruno Guizi, Bruno Oliveira, Bruno Ramos, Caio Nascimento, César Almeida, Christiano Moraes, Clara Battesini, Daniel Freitas, Daniel Haas, Danilo Segura, Dartz Dartz, David Kwast, Delton Porfiro, Dhyeives Rodovalho, Diego Farias, Diego Guimarães, Dilenon Delfino, Douglas Bastos, Douglas Braga, Douglas Martins, Douglas Zickuhr, Eli Júnior, Emerson Rafael, Érico Andrei, Eugenio Mazzini, Euripedes Borges, Evandro Avellar, Everton Silva, Fabio Barros, Fábio Barros, Fabio Castro, Fábio Thomaz, Felipe Rodrigues, Fernanda Prado, Fernando Rozas, Fernando Sousa, Flávio Meira, Flavkaze Flavkaze, Franklin Silva, Gabriel Barbosa, Gabriel Simonetto, Geandreson Costa, Guilherme Felitti, Guilherme Gall, Guilherme Ostrock, Gustavo Dettenborn, Heitor Fernandes, Henrique Junqueira, Igor Taconi, Ismael Ventura, Israel Gomes, Italo Silva, Jair Andrade, Jairo Lenfers, Janael Pinheiro, Jean Marcio, João Lugão, Johnny Tardin, Jonatas Leon, Jônatas Silva, Jorge Plautz, Jose Mazolini, Juan Gutierrez, Juliana Machado, Julio Silva, Kaio Peixoto, Kaneson Alves, Leandro Botassio, Leandro Miranda, Leonardo Cruz, Leonardo Mello, Leonardo Nazareth, Lucas Adorno, Lucas Mello, Lucas Mendes, Lucas Oliveira, Lucas Polo, Lucas Praciano, Lucas Teixeira, Lucas Valino, Luciano Silva, Luciano Teixeira, Luiz Junior, Luiz Lima, Luiz Paula, Maiquel Leonel, Marcelino Pinheiro, Marcelo Matte, Márcio Martignoni, Marco Mello, Marcos Gomes, Marco Yamada, Maria Clara, Marina Passos, Mario Deus, Mateus Lisboa, Matheus Silva, Matheus Vian, Mírian Batista, Murilo Andrade, Murilo Cunha, Murilo Viana, Natan Cervinski, Nicolas Teodosio, Osvaldo Neto, Otávio Barradas, Patricia Minamizawa, Patrick Felipe, Paulo Braga, Paulo Tadei, Pedro Duarte, Pedro Henrique, Pedro Pereira, Peterson Santos, Priscila Santos, Rafael Lopes, Rafael Rodrigues, Rafael Romão, Ramayana Menezes, Reinaldo Silva, Renan Moura, Renato Veirich, Riverfount Riverfount, Robson Maciel, Rodrigo Brandao, Rodrigo Ferreira, Rodrigo Freire, Rodrigo Junior, Rodrigo Vaccari, Rodrigo Vieira, Rogério Sousa, Ronaldo Silva, Rui Jr, Samanta Cicilia, Sara Selis, Thiago Araujo, Thiago Borges, Thiago Bueno, Thiago Curvelo, Thiago Moraes, Thiago S, Tiago Minuzzi, Tony Dias, Victor Wildner, Vinícius Bastos, Vinicius Figueiredo, Vítor Gomes, Vitor Luz, Vlademir Souza, Vladimir Lemos, Wellington Abreu, Wesley Mendes, William Alves, Willian Lopes, Wilson Neto, Yury Barros



Obrigado você



Um pouco sobre o
pysimplegui

A
Lib

0 projeto



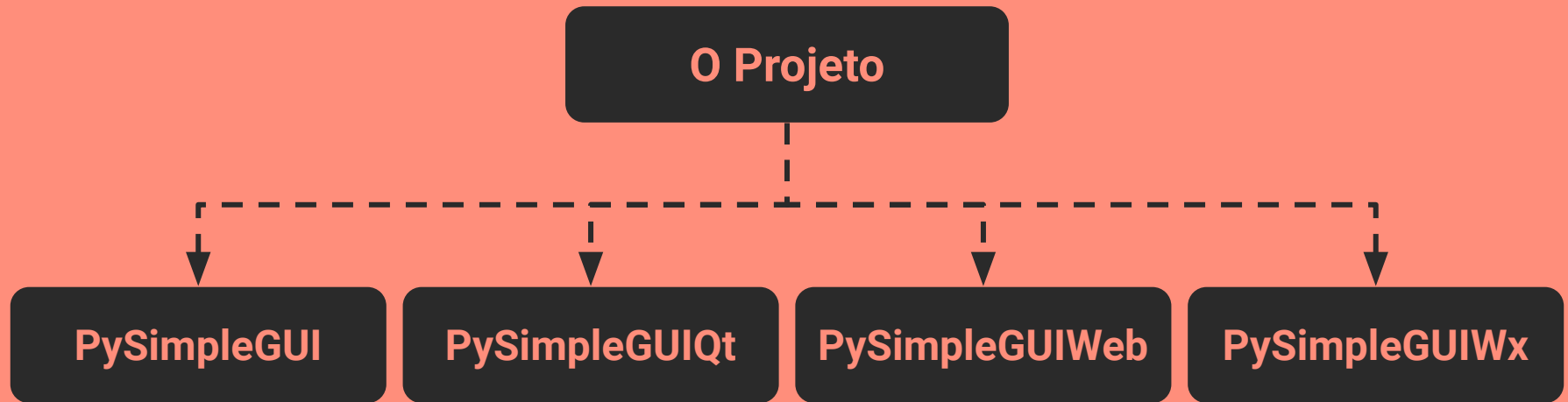
- PySimpleGUI
 - Python Simple Graphical User Interface
 - Interface Gráfica de Usuário Simples
- O projeto teve início 2018 (~4 anos)
- versão atual 4.60.0
- Mantido por @MikeTheWatchGuy
- GNU Lesser: Pode ser incorporada sem atribuição

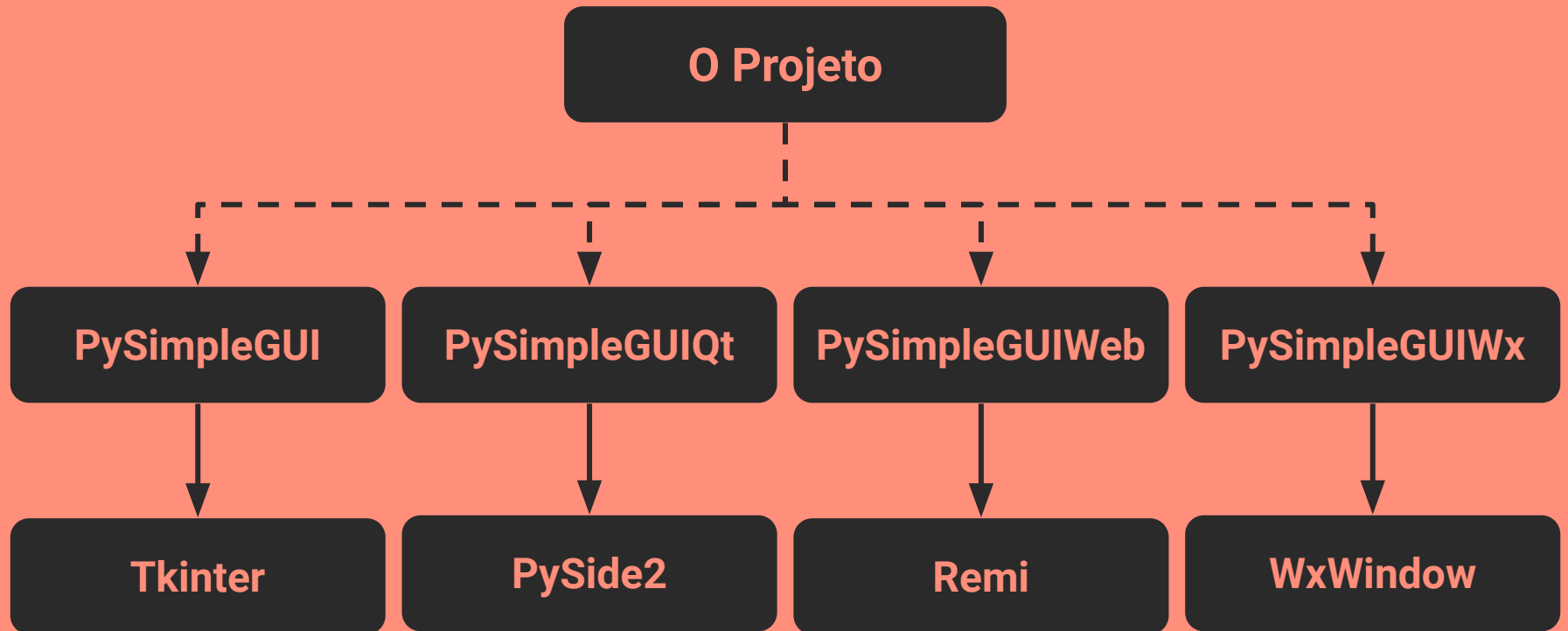
A biblioteca

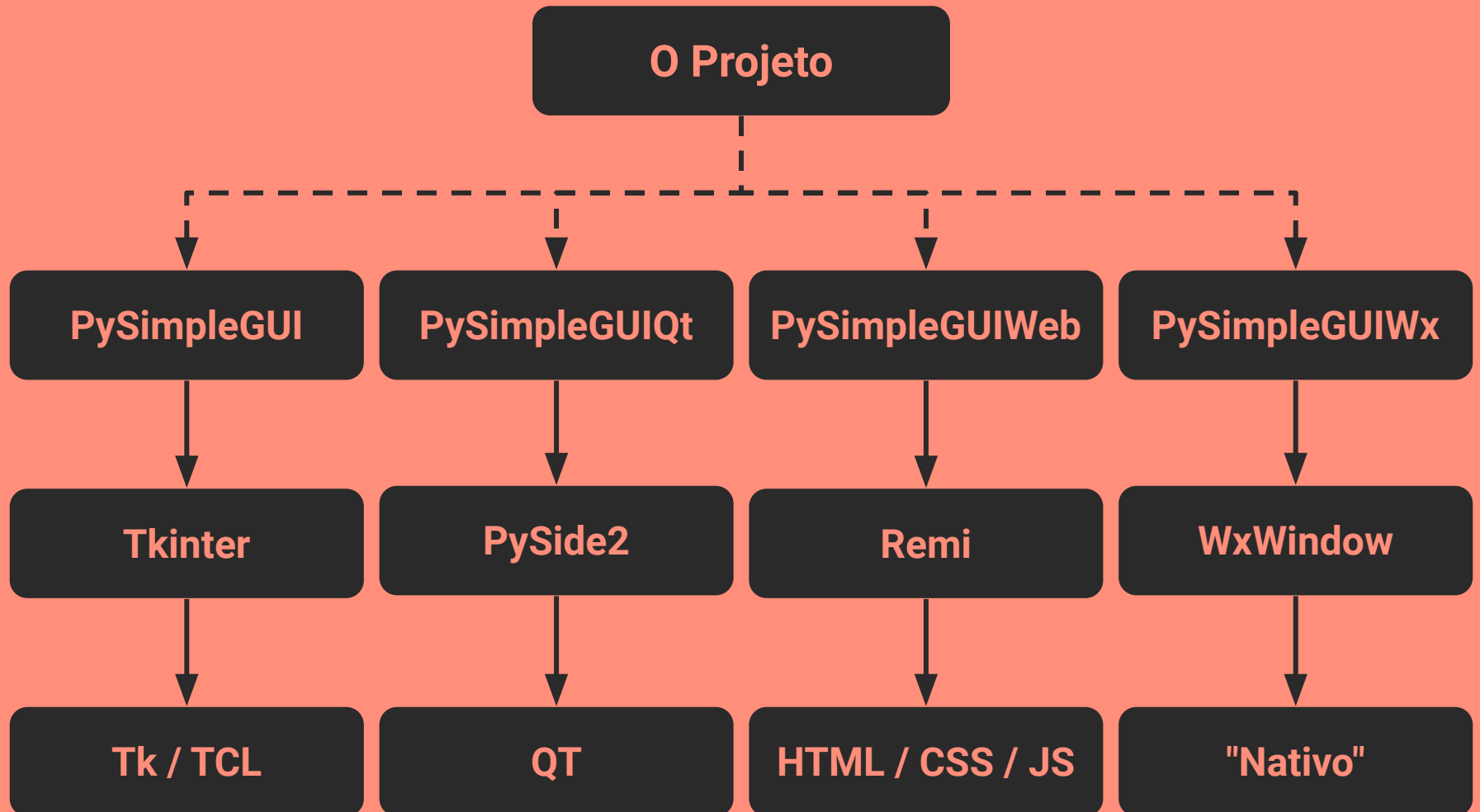


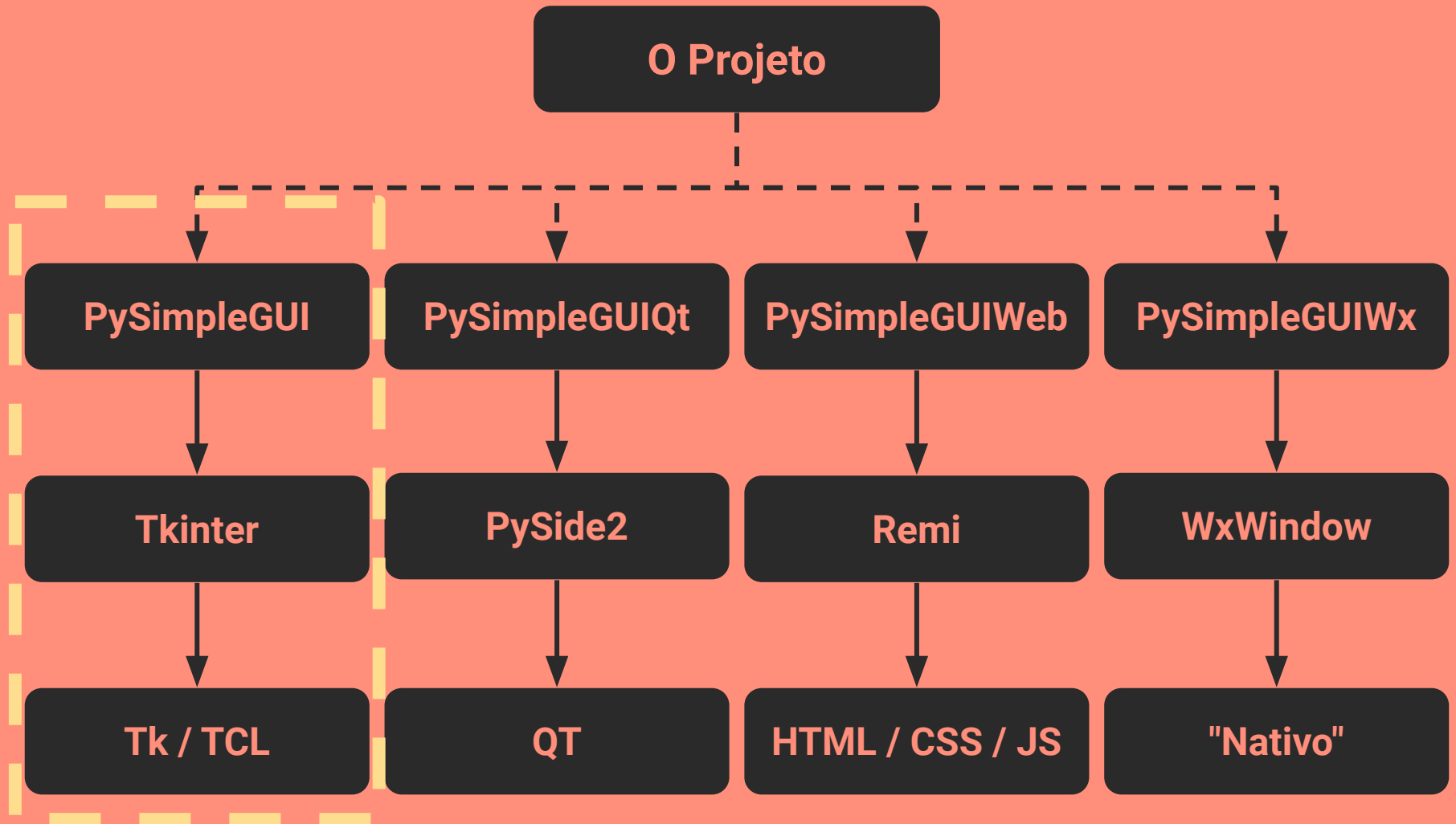
O PySimpleGUI **não é** um framework de janelas, como o TK, GTK, QT, WxWindow ou Kivy.

É **uma "família" de wrappers** para as bibliotecas originais. Parecido com o Toga, do projeto beeware (Live de Python #143)









Sub projetos



PSGTray

PSGBuilder

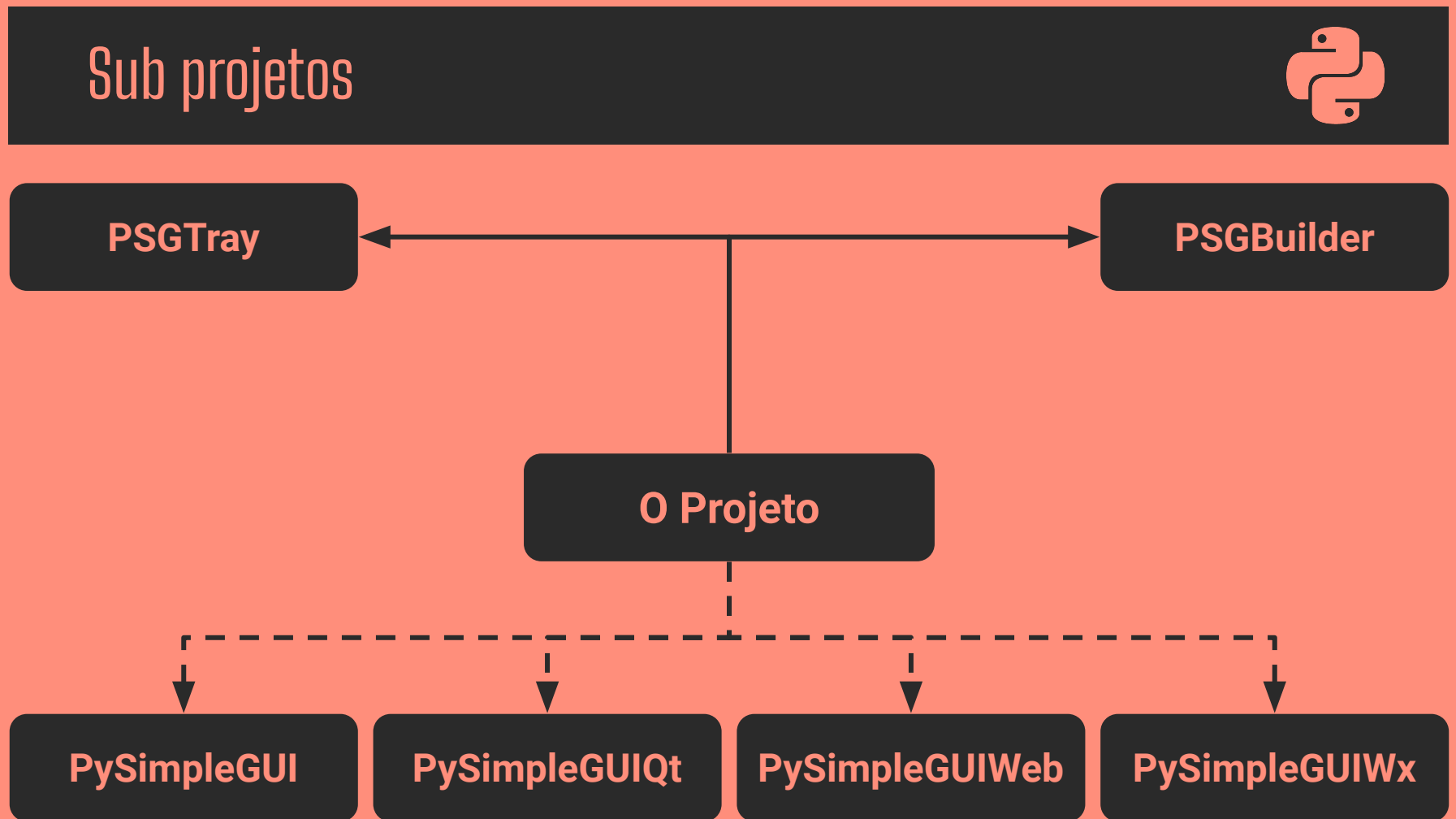
0 Projeto

PySimpleGUI

PySimpleGUIQt

PySimpleGUIWeb

PySimpleGUIWx



Sub projetos



PSGTray

Coloca a aplicação na bandeja de aplicativos

PSGBuilder

Faz o build para executável usando pyinstaller

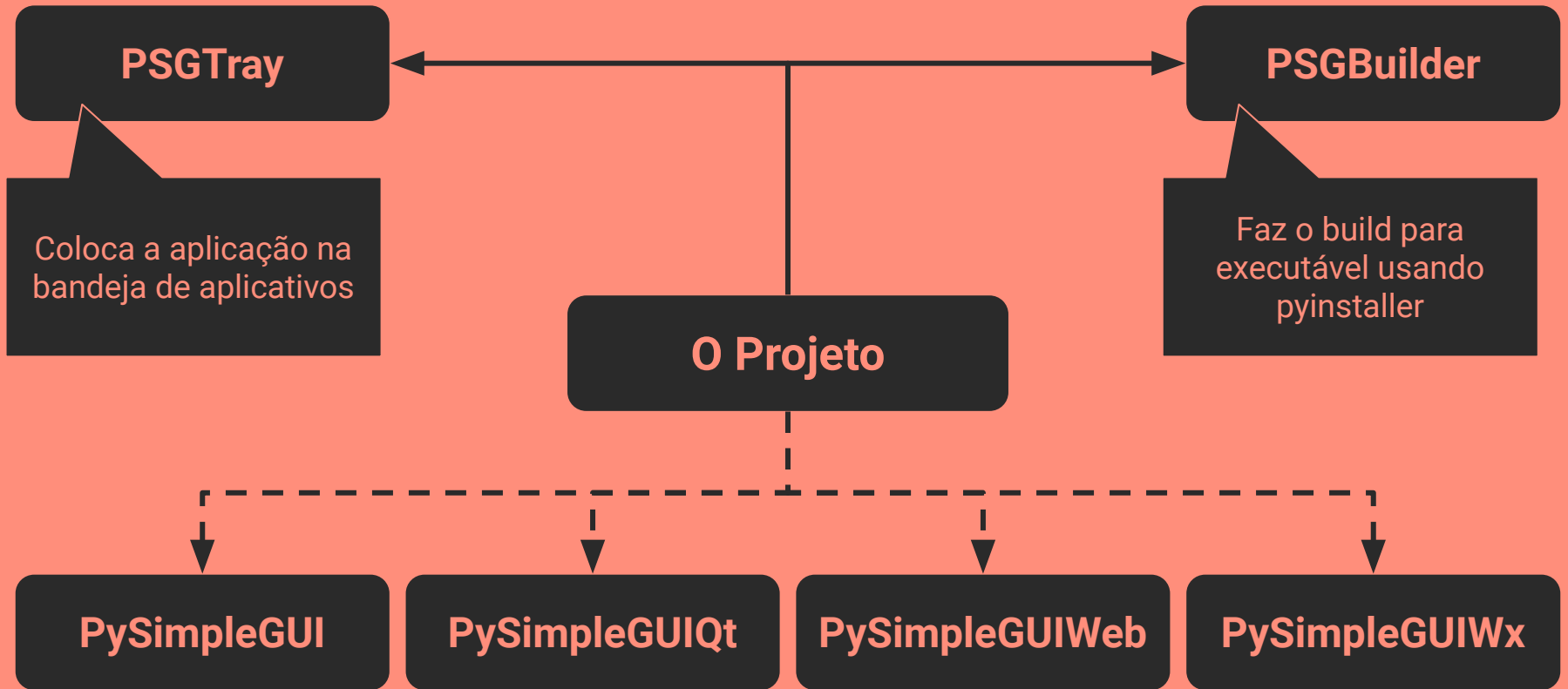
O Projeto

PySimpleGUI

PySimpleGUIQt

PySimpleGUIWeb

PySimpleGUIWx



```
pip install PySimpleGUI
```



Bora instalar



Window Widgets Layouts

0 core das
interfaces

O básico necessário



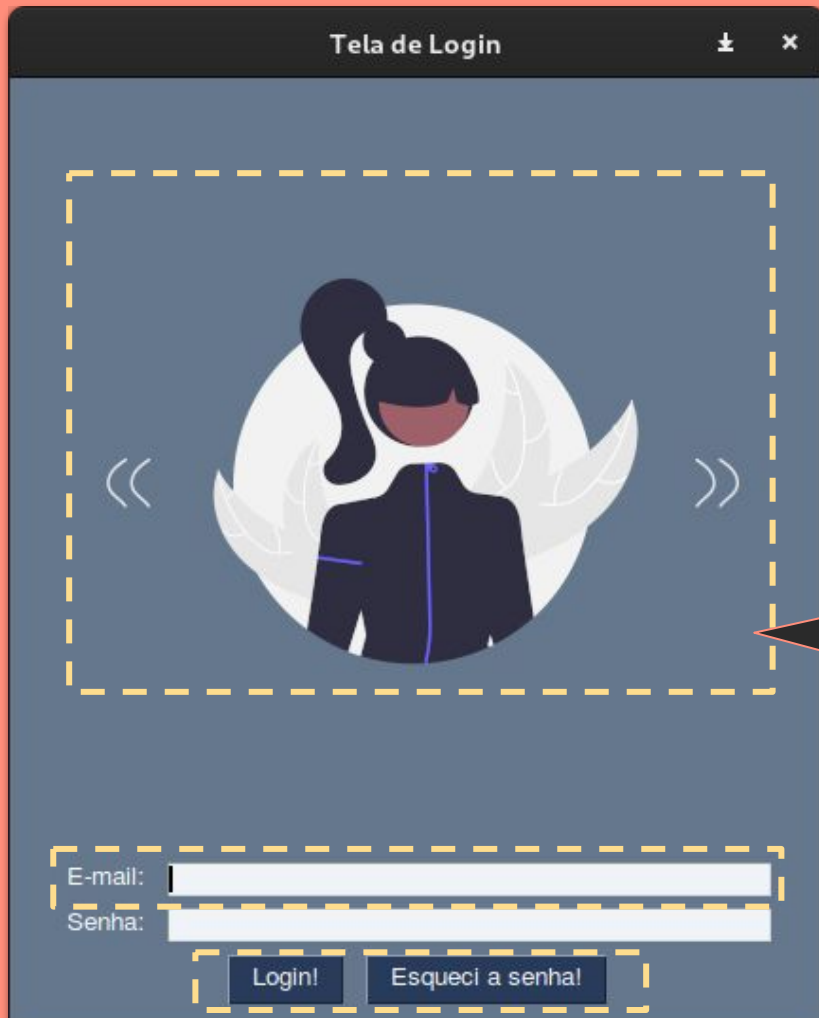
Título da Janela

Window
Janela

Ainda no básico



Tela de Login

A mockup of a login screen titled "Tela de Login". It features a central illustration of a person with dark hair in a ponytail, wearing a dark blue jacket, set against a white circular background with light blue wings. This illustration is enclosed in a dashed yellow rectangular border. Below the illustration, there are two input fields: the first is labeled "E-mail:" and the second is labeled "Senha:". At the bottom of the form, there are two buttons: "Login!" and "Esqueci a senha!". The entire form is set against a light blue background.

<< >>

E-mail:

Senha:

Login! Esqueci a senha!

Widgets

Criando uma Janela



```
1  from PySimpleGUI import Window
2
3  window = Window(
4      'Título da janela',
5      size=(600, 400),
6      layout=[[]]
7  )
8
9  window.read()
10
11 window.close()
```

Criando uma Janela

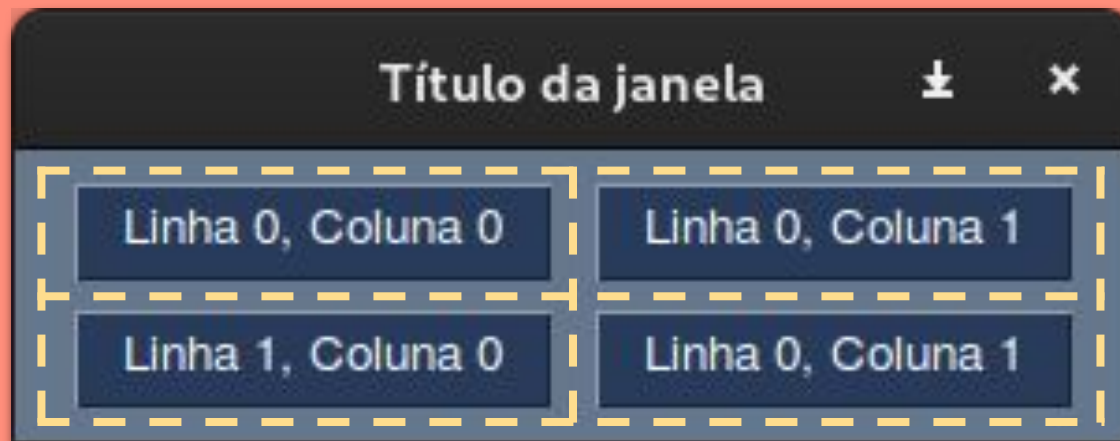


```
1  from PySimpleGUI import Window
2
3  window = Window(
4      'Título da janela',
5      size=(600, 400),
6      layout=[[]])
7  )
8
9  window.read()
10
11 window.close()
```

Layouts



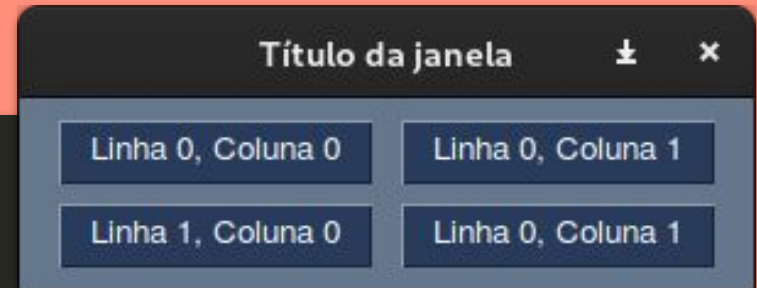
Para o PySimpleGUI, o layout é uma matriz de widgets.



Layouts



```
1  from PySimpleGUI import Button, Window
2
3  window = Window(
4      'Título da janela',
5      layout=[
6          [Button('Linha 0, Coluna 0'), Button('Linha 0, Coluna 1')],
7          [Button('Linha 1, Coluna 0'), Button('Linha 0, Coluna 1')]
8      ],
9  )
10
11 window.read()
12
13 window.close()
```



Widgets



A biblioteca de widgets disponíveis no PSG é composta por 34 widgets.

Sendo **27 componentes**, e **5 widgets de estrutura** e variações de componentes.

Componentes básicos:

- Texto
- Input de texto
- Botão
- Menu
- Imagem
- Slider
- Barra de progresso



Al

all_widgets.py

- Slider
- Barra de progresso

Um Layout Básico



Tela de Login

A diagram of a login form layout. The form is titled "Tela de Login" and features a central image of a person with wings, flanked by double arrows. Below the image are two input fields for "E-mail:" and "Senha:". At the bottom are two buttons: "Login!" and "Esqueci a senha!". The entire form is enclosed in a dashed yellow border.

Input

Image

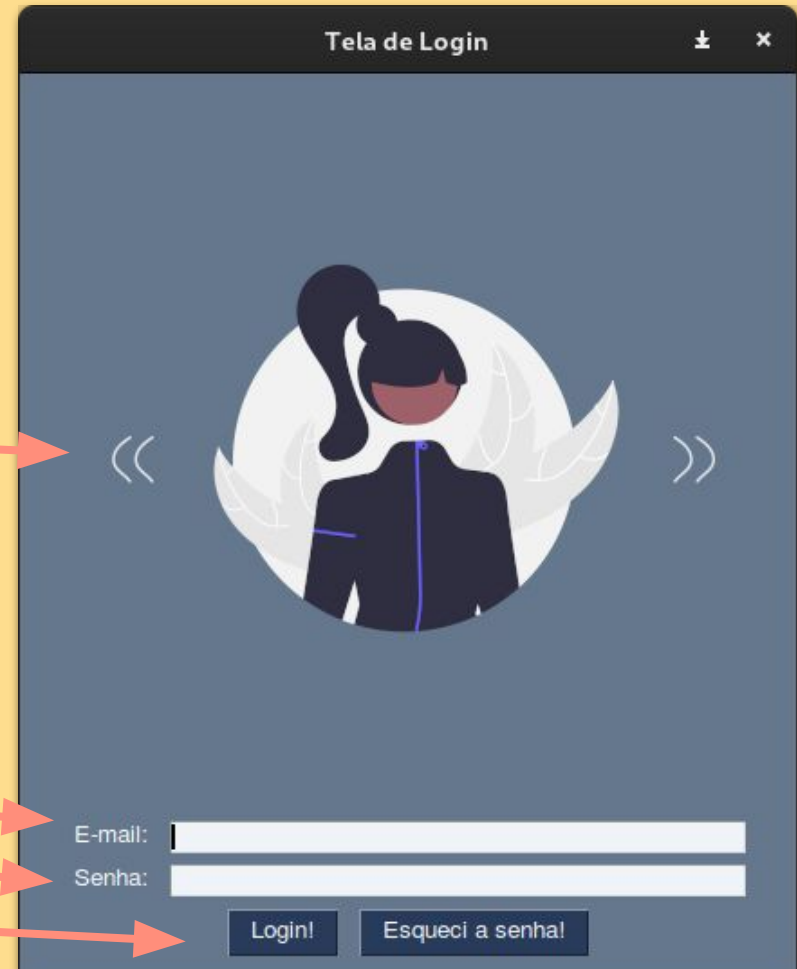
Text

Button

Aplicando o layout no código



```
1  from PySimpleGUI import (  
2      Button, Image, Input, Text  
3  )  
4  
5  
6  layout = [  
7      [Image()], # Linha 1  
8  
9  
10  
11  
12  
13     [text(), Input()], # Linha 2  
14     [text(), Input()], # Linha 3  
15     [Button(), Button()], # Linha 4  
16 ]
```

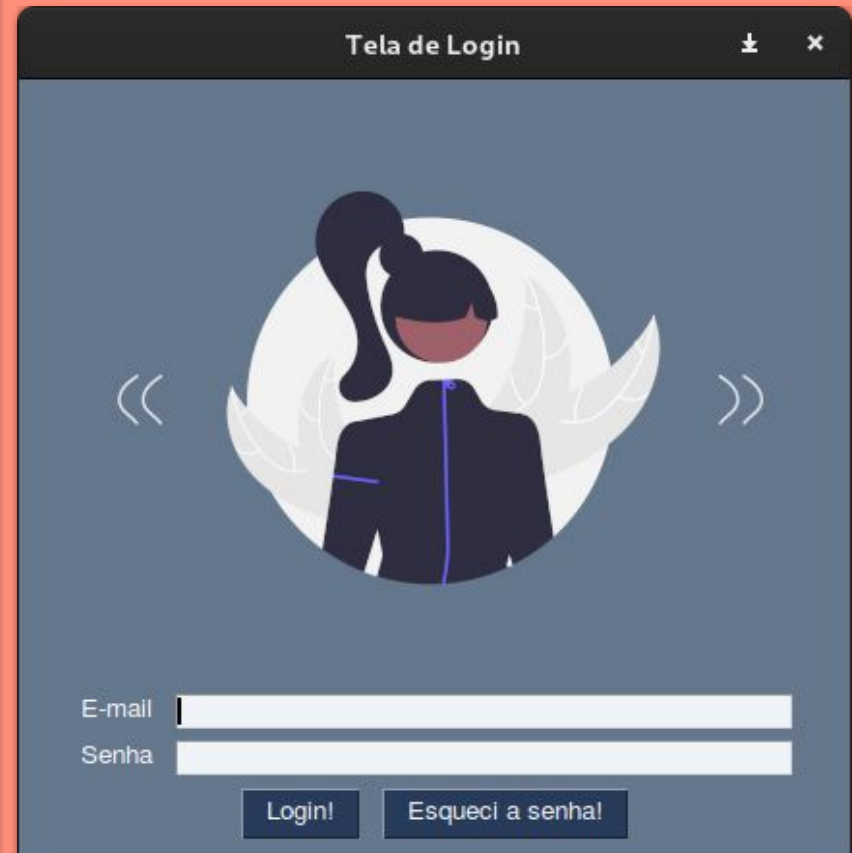


Com código de verdade



— □ ×

```
1  # exemplo_01.py
2  from PySimpleGUI import (
3      Button, Image, Input, Text, Window
4  )
5
6  layout = [
7      [Image(filename='avatar.png')],
8      [Text('E-mail'), Input()],
9      [Text('Senha'), Input(password_char='*')],
10     [Button('Login!'), Button('Esqueci a senha!')],
11 ]
12
13 window = Window(
14     'Tela de Login',
15     layout=layout,
16     element_justification='c'
17 )
18
19 window.read()
20
21 window.close()
```



```
pip install PySimpleGUIQt  
pip install PySimpleGUIWx  
pip install PySimpleGUIWeb *
```



Olhando as variações



Widgets de layout




- Column
- Seperator
 - Horizontal
 - Vertical
- Push
 - Horizontal
 - Vertical
- Sizer

Overview



Tela de Login



<< >>

E-mail

Senha

Login! Esqueci a senha!

The image shows a web browser window titled "Tela de Login". The window has a dark header bar with the title and standard window controls (minimize, maximize, close). The main content area is light blue and is split into two sections by a vertical line. On the left, there is a circular illustration of a person with dark hair in a ponytail, wearing a dark jacket, and having large white wings. The person is facing away from the viewer. To the left of the circle are two left-pointing chevrons (<<), and to the right are two right-pointing chevrons (>>). On the right side of the window, there are two input fields: "E-mail" and "Senha". Below the "E-mail" field is a "Login!" button, and below the "Senha" field is an "Esqueci a senha!" button.

Overview



Tela de Login



E-mail

Senha

Login!

Esqueci a senha!

Coluna

Coluna

Overview

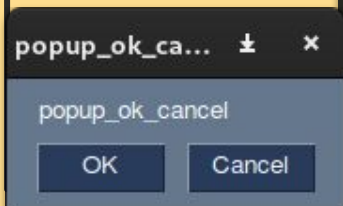
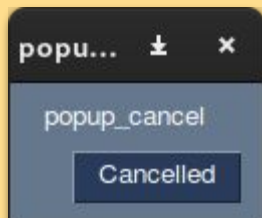
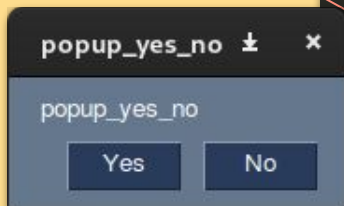


Separador

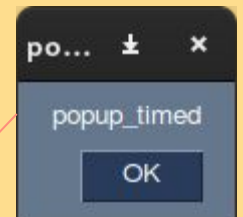
Push

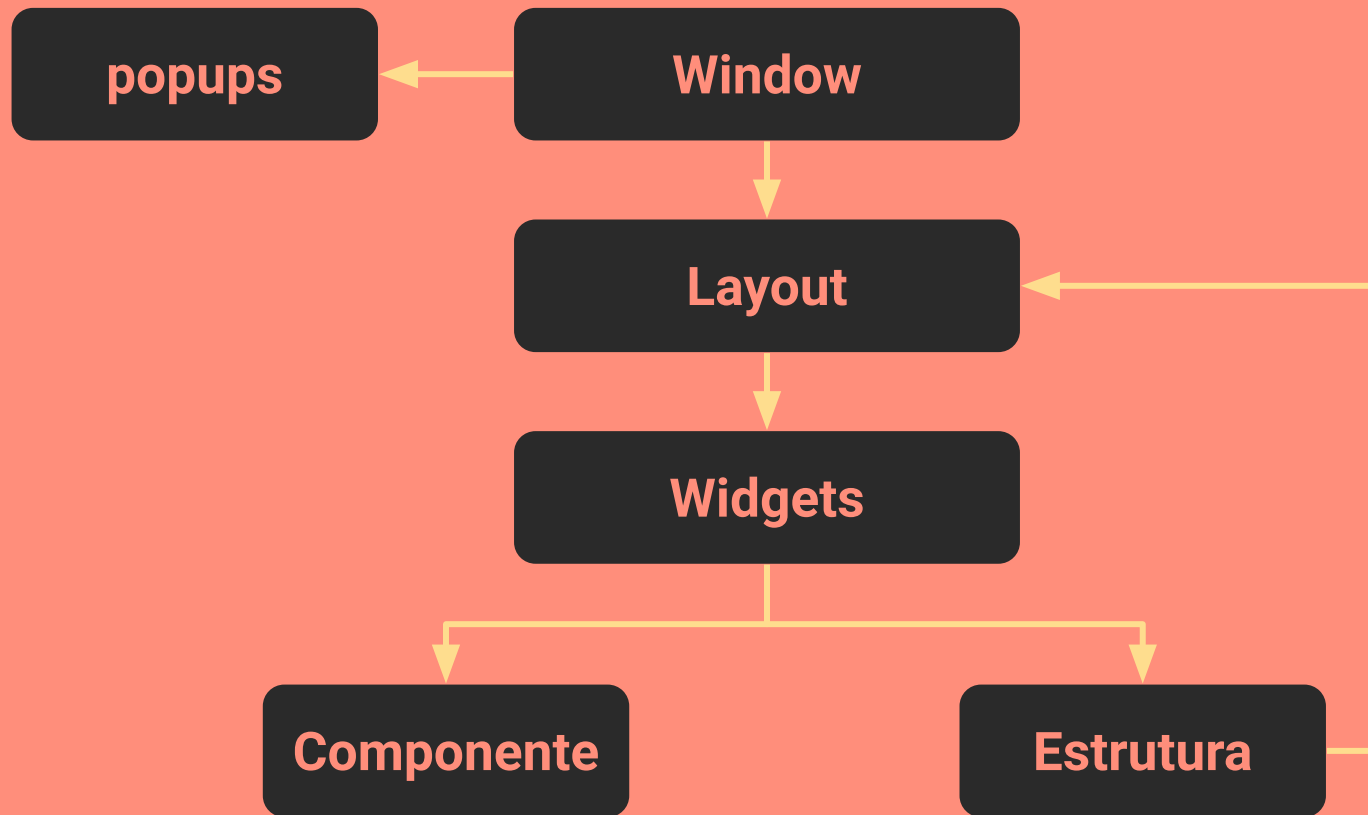
```
1 # exemplo_02.py
2 from PySimpleGUI import (
3     Button, Image, Input, Text, Window,
4     Column, VSeparator, Push
5 )
6
7 layout_a = [
8     [Image(filename='avatar.png')]
9 ]
10
11 layout_b = [
12     [Text('E-mail'), Push(), Input()],
13     [Text('Senha'), Push(), Input(password_char='*')],
14     [Push(), Button('Login!'), Push(), Button('Esqueci a senha!'), Push()],
15 ]
16
17 layout = [
18     [
19         Column(layout_a),
20         VSeparator(),
21         Column(layout_b),
22     ]
23 ]
```


PopUps



```
1  # exemplo_03.py
2  import PySimpleGUI as sg
3
4  sg.popup('popup')
5  sg.popup_ok('popup_ok')
6  sg.popup_yes_no('popup_yes_no')
7  sg.popup_cancel('popup_cancel')
8  sg.popup_ok_cancel('popup_ok_cancel')
9  sg.popup_error('popup_error')
10 sg.popup_timed('popup_timed')
11 sg.popup_auto_close('popup_auto_close')
12 sg.popup_annoying('annoying')
```





Uma grande
variedade

Temas

Observando a variedade de temas

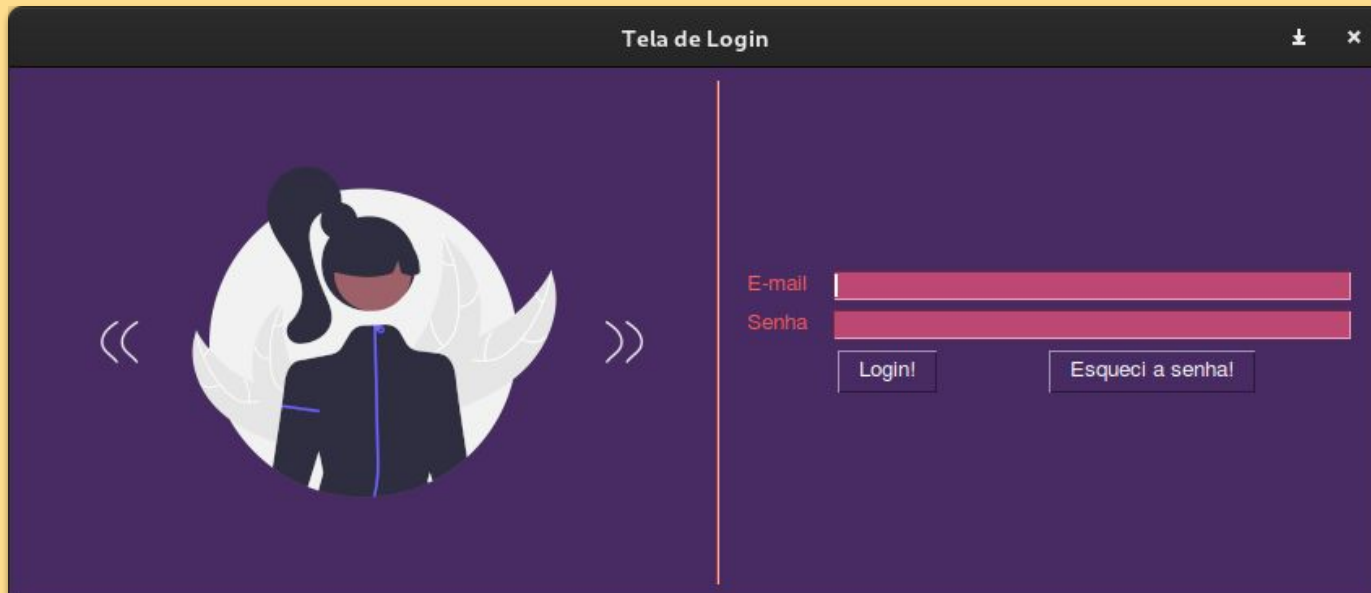


```
1  from PySimpleGUI import theme_previewer
2  theme_previewer()
```

Usando os temas



```
1 from PySimpleGUI import theme
2 theme('DarkPurple')
```



Event

OS

Fazendo as coisas
acontecerem

Window.read()



Os eventos são lidos pela janela usando o método **.read()**. E os estados de todos os widgets são capturados todas às vezes.

```
1  # ...  
2  print(window.read( ))  
3  # ...
```

Window.read()



Os eventos são lidos pela janela usando o método **.read()**. E os estados de todos os widgets são capturados todas às vezes.

```
1  # ...  
2  evento, valores = window.read()  
3  # ...
```


Loop de eventos



```
1  while True:
2      evento, valores = window.read()
3
4      match(evento):  # Live de Python #171
5          case None:
6              print('Fecha a janela')
7              break  # Sai do while
8          case _:  # Live de Python #181
9              print(f'{evento=}, {valores=}')
10
11  window.close()
```

Loop de eventos



```
1  while True:
2      evento, valores = window.read()
3
4      match(evento):  # Live de Python #171
5          case None:
6              print('Fecha a janela')
7              break  # Sai do while
8          case _:  # Live de Python #181
9              print(f'{evento=}, {valores=}')
10
11  window.close()
```

Identificadores de widgets



Todos os widgets tem um atributo **key** ele é passado por parâmetro na instância do widget. Caso não seja usado, o PySimpleGUI vai definir um valor numérico a ele. Por exemplo:

```
1  python exemplo_04.py
2  evento='Login!', valores={2: 'email@mail.com', 3: 'senha_senha'}
```

Identificadores de widgets



Todos os widgets tem um atributo **key** ele é passado por parâmetro na instância do widget.

```
1  # ...
2  layout_b = [
3      [Text('E-mail'), Push(), Input(key='-EMAIL-')],
4      [Text('Senha'), Push(), Input(password_char='*', key='-SENHA-')],
5      [Push(), Button('Login!'), Push(), Button('Esqueci a senha!'), Push()],
6  ]
7  # ...
```

Identificadores de widgets



```
1  # ...
2  layout_b = [
3      [Text('E-mail'), Push(), Input(key='-EMAIL-')],
4      [Text('Senha'), Push(), Input(password_char='*', key='-SENHA-')],
5      [Push(), Button('Login!'), Push(), Button('Esqueci a senha!'), Push()],
6  ]
7  # ...
```

```
1  python exemplo_04.py
2  evento='Login!', valores={'-EMAIL-': 'email@mail.com', '-SENHA-': 'senha_senha'}
```

Um evento completo – Key do evento

```
1  # ...
2  layout_a = [
3      [Image(filename='avatar.png', key='-IMAGEM-')]
4  ]
5
6  layout_b = [
7      [Text('E-mail'), Push(), Input(key='-EMAIL-')],
8      [Text('Senha'), Push(), Input(password_char='*', key='-SENHA-')],
9      [Push(), Button('Login!'), Push(), Button('Esqueci a senha!'), Push()],
10 ]
11 # ...
```

```
1  while True:
2      evento, valores = window.read()
3      match(evento):
4          case 'Login!':
5
6              popup_ok(
7                  f"{valores['-EMAIL-']}, {valores['-SENHA-']}"
8              )
9
10         window['-IMAGEM-'].update(filename='avatar_2.png')
```

Um evento completo - Valores dos widgets

```
1  # ...
2  layout_a = [
3      [Image(filename='avatar.png', key='-IMAGEM-')]
4  ]
5
6  layout_b = [
7      [Text('E-mail'), Push(), Input(key='-EMAIL-')],
8      [Text('Senha'), Push(), Input(password_char='*', key='-SENHA-')],
9      [Push(), Button('Login!'), Push(), Button('Esqueci a senha!'), Push()],
10 ]
11 # ...
```

```
1  while True:
2      evento, valores = window.read()
3      match(evento):
4          case 'Login!':
5
6              popup_ok(
7                  f"{valores['-EMAIL-']}, {valores['-SENHA-']}"
8              )
9
10         window['-IMAGEM-'].update(filename='avatar_2.png')
```

Um evento completo - Update do widget

```
1  # ...
2  layout_a = [
3      [Image(filename='avatar.png', key='-IMAGEM-')]
4  ]
5
6  layout_b = [
7      [Text('E-mail'), Push(), Input(key='-EMAIL-')],
8      [Text('Senha'), Push(), Input(password_char='*', key='-SENHA-')],
9      [Push(), Button('Login!'), Push(), Button('Esqueci a senha!'), Push()],
10 ]
11 # ...
```

```
1  while True:
2      evento, valores = window.read()
3      match(evento):
4          case 'Login!':
5
6              popup_ok(
7                  f"{valores['-EMAIL-']}, {valores['-SENHA-']}"
8              )
9
10             window['-IMAGEM-'].update(filename='avatar_2.png')
```


Bora brincar um
pouco

Play
ground

Aten
ção!

Hora de tirarmos
algumas conclusões

Problemas



- Documentação confusa
- 24 mil linhas de código em um único arquivo
 - Versão TK
- Os ports ainda não contam com todos os widgets implementados
- Não existem testes
- Não suporta programação assíncrona
- **Bandit** issues:
 - 5 vulnerabilidades críticas
 - Eval e Popen
 - 94 vulnerabilidades ao total
 - 68 try_except_pass



picpay.me/dunossauro



apoia.se/livedepython



pix.dunossauro@gmail.com



Ajude o projeto <3

