

CRIAÇÃO DE APLICAÇÃO GUI

Biblioteca Tkinter

Conceitos

Neste módulo, o nosso foco será na criação de uma aplicação GUI. Para isso, a biblioteca que vamos usar como referência é a Tkinter.

Comentário

Essa biblioteca GUI é considerada a padrão para desenvolvimento de interface gráfica no python. Ela fornece uma interface orientada a objetos que facilita a implementação de programas interativos.

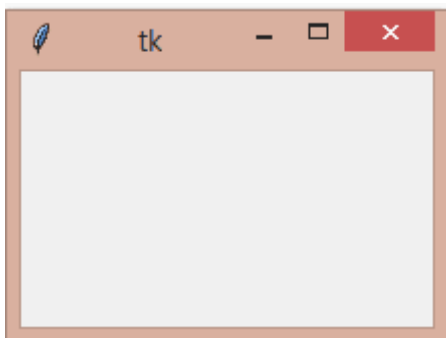
Para usar o Tkinter – supondo que você já instalou o Python e o pacote Tkinter –, é necessário executar as seguintes etapas:

1. Importar a biblioteca Tkinter.
2. Criar a janela principal do programa GUI.
3. Adicionar um ou mais dos widgets (componentes).
4. Entrar no loop de evento principal para tratar os eventos disparados pelo usuário.

O código fica assim:

```
import tkinter  
janela = tkinter.Tk()  
janela.mainloop()
```

Esse programa produz a seguinte saída:



Exemplo de aplicação Tkinter.

Widgets Tkinter

O Tkinter possui diversos componentes (widgets), tais como botões, rótulos e caixas de texto usados para criar aplicações interativas com o usuário.

Os principais widgets do Tkinter, de acordo com Meier (2015), são:

Botão (Button)	É usado para exibir os botões na aplicação. São usados, por exemplo, para confirmar uma ação de salvar os dados.
Telas (Canvas)	É usado para desenhar formas, como linhas, ovais, polígonos e retângulos.
Botão de verificação (Checkbutton)	É usado para exibir várias opções como caixas de seleção. O usuário pode selecionar várias opções ao mesmo tempo.
Entrada de texto (Entry)	É usado para exibir uma caixa de texto de linha única para que o usuário digite valores de entrada.
Quadros (Frame)	É usado como um widget de contêiner, isso significa que outros componentes são adicionados a ele com o objetivo de organizar outros widgets.
Rótulo (Label)	É usado para fornecer uma legenda de linha única para outros widgets. Também pode conter imagens.
Caixa de listagem (Listbox)	É usado para fornecer uma lista de opções para um usuário.
Menubutton	É usado para exibir opções no menu.

Menu	É usado para fornecer várias possibilidades de comandos a um usuário. Esses comandos estão contidos no Menubutton.
Mensagem (Message)	É usado para exibir uma mensagem de texto e um botão para o usuário confirmar uma ação/td>
Botão de rádio (Radiobutton)	É usado para exibir várias opções, como botões de rádio. O usuário pode selecionar apenas uma opção por vez.
Escala (Scale)	É usado para fornecer um widget de controle deslizante.
Barra de rolagem (Scrollbar)	É usado para adicionar capacidade de rolagem a vários widgets.
Texto (Text)	É usado para exibir texto em várias linhas.
Toplevel	É usado para fornecer um contêiner de janela separado.
Spinbox	É uma variante do widget Entry padrão. Ele é usado para selecionar um número fixo de valores.
PanedWindow	É um widget de contêiner que pode conter qualquer número de painéis, organizados horizontalmente ou verticalmente.
LabelFrame	É um widget de contêiner simples. Seu objetivo principal é atuar como um espaçador, ou contêiner para layouts de janela.
tkMessageBox	Este módulo é usado para exibir caixas de mensagens.

Cada um desses widgets possuem propriedades que permitem personalizar tamanhos, cores e fontes que serão exibidos para o usuário.

Na próxima seção, apresentaremos alguns exemplos que vão ajudar a entender como desenvolver uma aplicação GUI.

Exemplos de aplicações GUI

Desenvolvendo exemplos de aplicações GUI

Recomendação

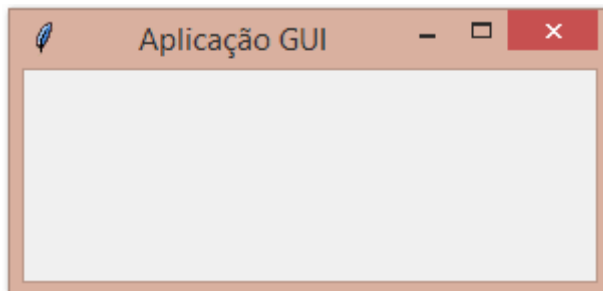
Antes de iniciar esta seção, é fortemente recomendado que você tenha feito a instalação da versão mais atual do python e tente executar os exemplos.

Vamos apresentar exemplos dos componentes:

1. Window
2. Label
3. Button
4. Entry
5. Radiobutton
6. Checkbox
7. Text
8. Message
9. Sliders
10. Dialog
11. Combobox

Widget Window (tk.Tk())

Para iniciar, a primeira aplicação vai exibir uma janela redimensionável, conforme a imagem seguinte.



Exemplo de uma janela redimensionável.

O código dessa aplicação é:

```
import tkinter as tk
janela = tk.Tk()
janela.title(" Aplicação GUI")
janela.mainloop()
```

Agora, cada uma das linhas será analisada.

- Linha 1 - É feita a importação da biblioteca Tkinter.
- Linha 2 - É criada uma instância da classe Tk no objeto “janela”.
- Linha 3 - O método “title” é usado para definir um título que aparece no topo da janela, no caso, “Aplicação GUI”.
- Linha 4 - A aplicação inicia o loop de evento principal da janela.

A janela da aplicação anterior é redimensionável, ou seja, o usuário pode alterar as dimensões da janela se ele clicar com o cursor do mouse na janela e arrastá-la.

Para fixar o tamanho da janela, é necessário determinar essa propriedade conforme o código seguinte:

```
import tkinter as tk
janela = tk.Tk()
janela.title(" Aplicação GUI NÃO Dimensionável")
janela.resizable(False, False)
janela.mainloop()
```

Atenção!

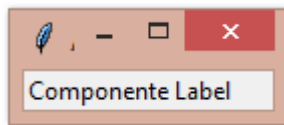
A principal diferença desse exemplo em relação ao anterior está na linha 4, onde a propriedade de redimensionar a janela é colocada como “Falso”.

Widget Label

O próximo exemplo apresenta como usar o widget “Label”. O código para gerar uma aplicação com o “Label” é dado por:

```
import tkinter as tk
from tkinter import ttk
janela = tk.Tk()
janela.title(" Aplicação GUI com o Widget Label")
ttk.Label(janela, text="Componente Label" ).grid(column=0, row=0)
janela.mainloop()
```

Na próxima imagem, é exibido o resultado da execução:



Exemplo do componente Label.

Na linha 5, é feito o posicionamento do componente “label” na “janela” com o gerenciador de layout “grid”.

Widget Button

O próximo exemplo apresenta como usar o widget “Button”.

O código para gerar uma aplicação com o componente “Button” é dado por:

```
import tkinter as tk
contador = 0
def contador_label(lblRotulo):
    def funcao_contar():
        global contador
```

```
contador = contador + 1
lblRotulo.config(text=str(contador))
lblRotulo.after(1000, funcao_contar)
funcao_contar()

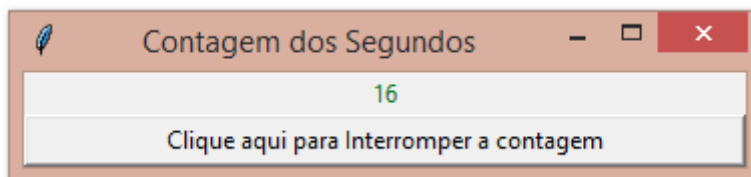
janela = tk.Tk()
janela.title("Contagem dos Segundos")
lblRotulo = tk.Label(janela, fg="green")
lblRotulo.pack()
contador_label(lblRotulo)

btnAcao = tk.Button(janela, text='Clique aqui para Interromper a
contagem', width=50, command=janela.destroy)

btnAcao.pack()
janela.mainloop()
```

Este programa vai gerar uma janela com um contador de segundos – que utiliza um componente “label” – e um componente botão com a mensagem “Clique aqui para interromper a contagem”.

Na próxima imagem, é exibido o resultado da execução do programa.



Exemplo do componente botão.

As linhas mais importantes deste código são:

- Linha 14 - Chamada para a função “contador_label”, função que faz a contagem dos segundos e a atualização dos dados do componente “label”.
- Linha 15 - Criação de uma instância do componente “botão” com uma mensagem, largura do componente e o estabelecimento de um comportamento, no caso, fechar a janela, quando o usuário pressionar o botão.

Widget Entry

Agora, vamos analisar o componente “entry”. Ele é uma das principais formas de o usuário entrar com dados no sistema.

A seguir, é apresentado um exemplo de como usar esse componente:

```
import tkinter as tk

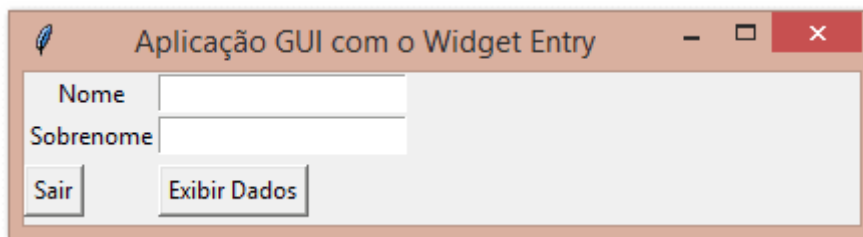
def mostrar_nomes():
    print("Nome: %s\nSobrenome: %s" % (e1.get(), e2.get()))

janela = tk.Tk()

janela.title("Aplicação GUI com o Widget Entry")

tk.Label(janela, text="Nome").grid(row=0)
```

O código produzirá uma janela com duas entradas de dados (widgets entry), dois botões e dois componentes rótulos, conforme podemos ver na imagem a seguir.



Exemplo do componente entry.

Os códigos que estamos apresentando estão evoluindo em termos de aplicações práticas.

No caso do exemplo da imagem anterior, com poucas modificações, pode ser aplicado para muitas situações reais.

Agora, analisaremos os principais aspectos do código.

- Linha 2 - É implementada a função “mostrar_nomes”, que vai exibir na linha de comando os nomes que estão escritos nas instâncias “e1” e “e2” do componente “entry”.
- Linha 8 e 9 - São feitas as instâncias “e1” e “e2” do componente entry.
- Linha 10 e 11 - “e1” e “e2” são posicionados na janela.
- Linha 12 e 13 - São instanciados objetos do componente “botão”. Em especial, na linha 13, a função “mostrar_nomes” é associada ao comportamento do botão.

Widget Radiobutton

O próximo exemplo apresenta como usar o widget “Radiobutton”. O código para gerar uma aplicação com esse componente é dado por:

```
import tkinter as tk

janela = tk.Tk()

v = tk.IntVar()

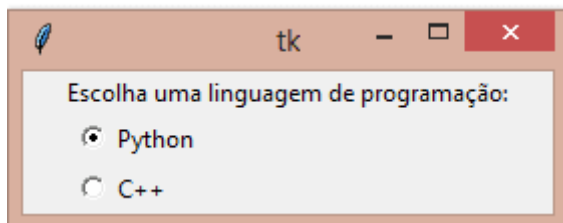
tk.Label(janela, text=""Escolha uma linguagem de
programação:"", justify = tk.LEFT, padx = 20).pack()

tk.Radiobutton(janela, text="python", padx =
25, variable=v, value=1).pack(anchor=tk.W)

tk.Radiobutton(janela, text="C++", padx =
25, variable=v, value=2).pack(anchor=tk.W)

janela.mainloop()
```

O código produzirá uma janela com dois “botões de rádio” (widgets radiobutton), conforme a imagem a seguir.



Exemplo do componente radiobutton.

- Linha 4 - É instanciado um componente “label” com a parametrização das propriedades “text”, “justify” e “padx” que correspondem, respectivamente, ao texto, a como ele será justificado e à largura em pixels do componente.
- Linha 5 e 6 - São instanciados os componentes “radiobuttons”. A propriedade “variable” recebe o valor “v”, indicando que o componente será colocado na vertical.

Widget Checkbox

O próximo exemplo apresenta como usar o widget “checkbox”. O código para gerar uma aplicação com esse componente é dado por:

```
import tkinter as tk
```

```

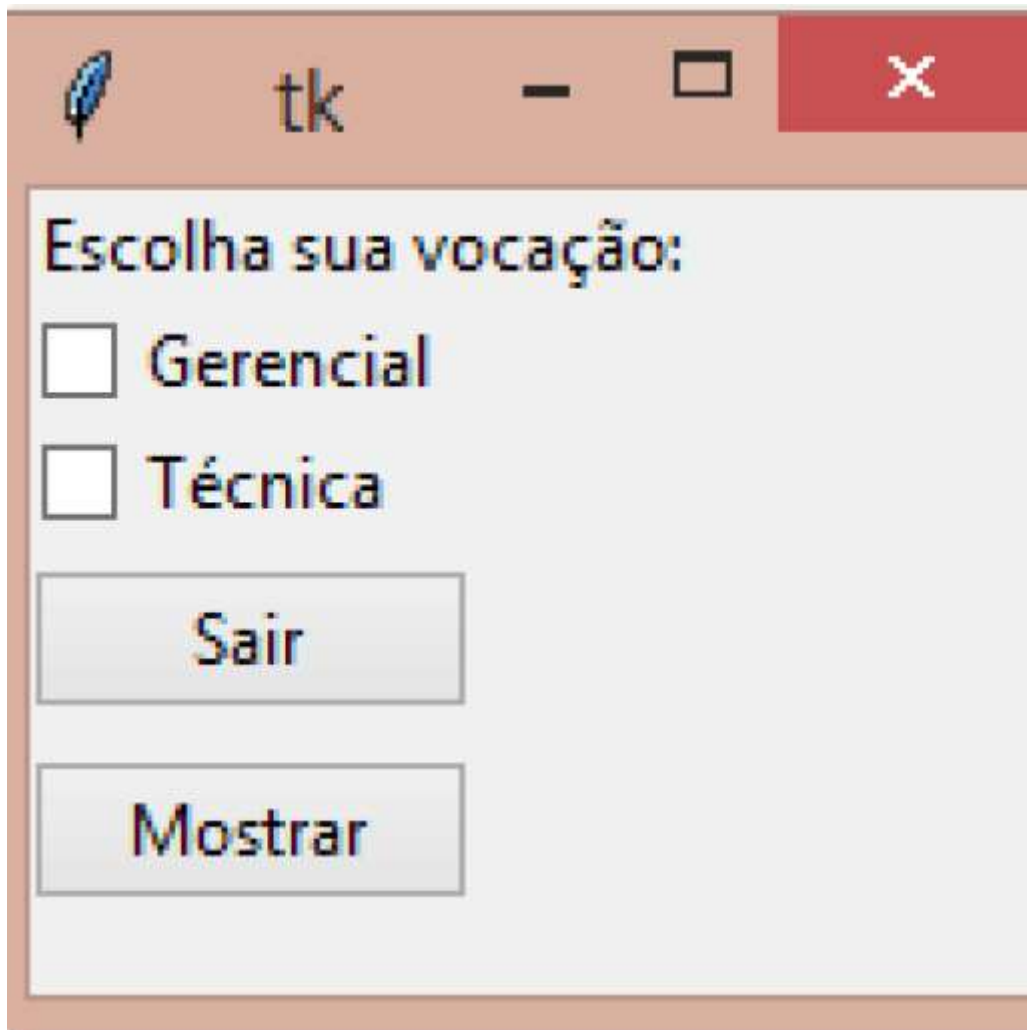
from tkinter import ttk

janela = tk.Tk()

def escolha_carreira():
    print("Gerencial: %d,\nTécnica : %d" % (var1.get(), var2.get()))
    ttk.Label(janela, text="Escolha sua vocação:").grid(row=0, sticky=tk.W)
    var1 = tk.IntVar()
    ttk.Checkbutton(janela, text="Gerencial", variable=var1).grid(row=1,
        sticky=tk.W)
    var2 = tk.IntVar()
    ttk.Checkbutton(janela, text="Técnica", variable=var2).grid(row=2,
        sticky=tk.W)
    ttk.Button(janela, text='Sair', command=janela.quit).grid(row=3,
        sticky=tk.W, pady=4)
    ttk.Button(janela, text='Mostrar',
        command=escolha_carreira).grid(row=4, sticky=tk.W, pady=4)
janela.mainloop()

```

O código produzirá uma janela com um “label”, dois “checkboxes” e dois botões, conforme a Figura 16.



Exemplo do componente checkbox.

Agora, vamos analisar os principais aspectos do código.

- Linha 4 - É implementada a função “escolha_carreira”, que exibirá os valores dos objetos “var1” e “var2”, que estão relacionados aos “checkboxes”.
- Linhas 7 e 9 - São instanciados os objetos “var1” e “var2”, que serão associados aos “checkboxes”. Quando um “checkbox” for selecionado, o seu respectivo objetivo vai retornar valor 1; caso o componente não seja selecionado, o valor do objeto será 0.
- Linhas 8 e 10 - São instanciados objetos dos componentes “checkboxes”, que são associados, respectivamente, às opções Gerencial e Técnica.

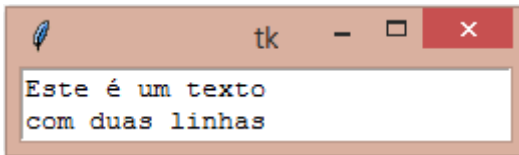
Widget Text

O próximo exemplo apresenta como usar o widget “Text”. O código para gerar uma aplicação com o componente “Text” é dado por:

```
import tkinter as tk
```

```
janela = tk.Tk()
T = tk.Text(janela, height=2, width=30)
T.pack()
T.insert(tk.END, "Este é um texto\ncom duas linhas\n")
tk.mainloop()
```

O código produzirá uma janela com um texto, conforme a imagem a seguir.



Exemplo do componente Text.

Agora, vamos analisar os principais aspectos do código.

- Linha 3 - É feita uma instância do componente “Text”.
- Linha 5 - É inserido um texto na instância do componente “text”, que será exibido na tela. Observe que o texto é separado em duas linhas com o uso do “\n”.

Widget Message

O próximo exemplo apresenta como usar o widget “Message”. O código para gerar uma aplicação com o componente “Message” é dado por:

```
import tkinter as tk

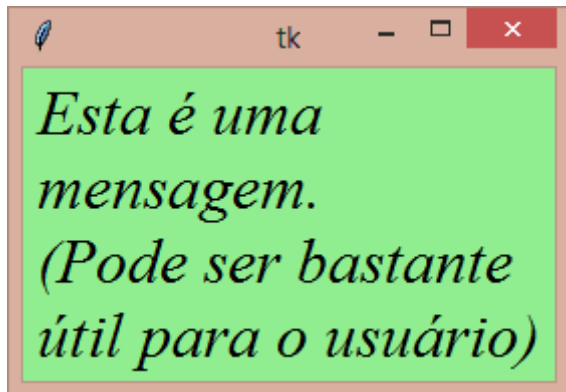
janela = tk.Tk()

mensagem_para_usuario = "Esta é uma mensagem.\n(Pode ser bastante útil para o usuário)"

msg = tk.Message(janela, text = mensagem_para_usuario)
msg.config(bg='lightgreen', font=('times', 24, 'italic'))
msg.pack()

janela.mainloop()
```

O código produzirá uma janela com uma mensagem, conforme pode ser visto na imagem seguinte.



Exemplo do componente Message.

Agora, vamos analisar os principais aspectos do código.

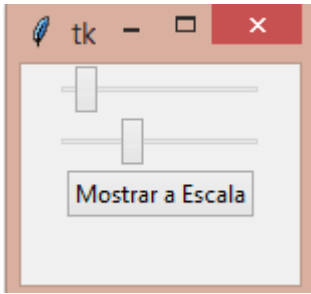
- Linha 4 - É instanciado um componente “Message” com uma mensagem para o usuário.
- Linha 5 - O componente é configurado, determinando a cor do “background” e detalhes sobre a fonte da mensagem.

Widget Slider

O próximo exemplo apresenta como usar o widget “Slider”. O código para gerar uma aplicação com esse componente é dado por:

```
import tkinter as tk
from tkinter import ttk
def mostrar_valores():
    print (w1.get(), w2.get())
janela = tk.Tk()
w1 = ttk.Scale(janela, from_=0, to=50)
w1.pack()
```

O código produzirá uma janela com duas linhas deslizantes, conforme a imagem a seguir.



Exemplo do componente Slider.

Agora, vamos analisar os principais aspectos do código.

- Linha 6 e 8 - São instanciados componentes “sliders”.

Além disso, são determinadas as propriedades “from”, “to” e “orient”, que são responsáveis, respectivamente, pelo espectro de escala componente e pela orientação do componente na tela.

Widget Dialog

O próximo exemplo apresenta como usar o widget “Dialog”. O código para gerar uma aplicação com o componente “Dialog” é dado por:

```
import tkinter as tk

from tkinter import messagebox as mb

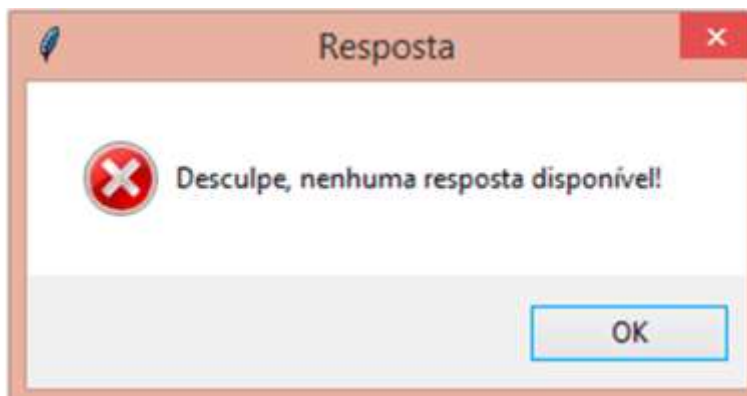
def resposta():
    mb.showerror("Resposta", "Desculpe, nenhuma resposta disponível!")

def verificacao():
    if mb.askyesno('Verificar', 'Realmente quer sair?'):
        mb.showwarning('Yes', 'Ainda não foi implementado')
    else:
```

O código produzirá uma janela com dois botões, conforme a imagem a seguir.

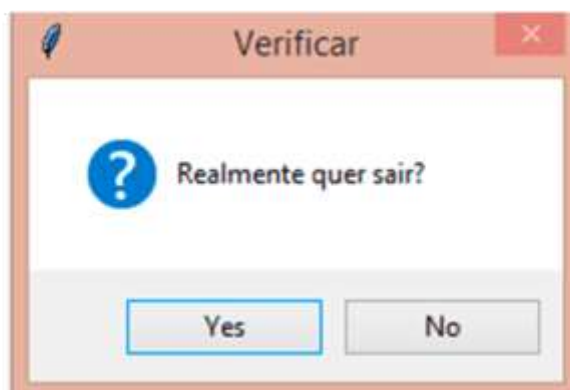


Exemplo com dois componentes botões.



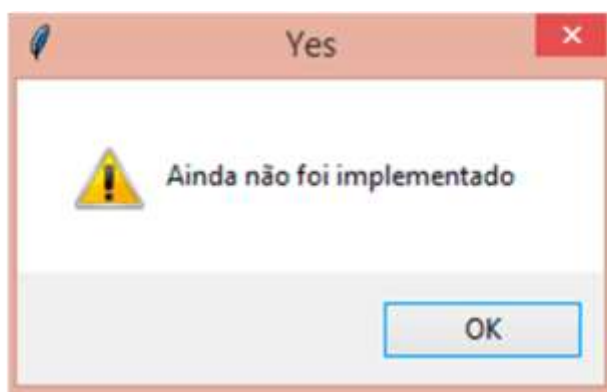
Resposta

Caso o usuário pressione o botão “Resposta”, aparecerá uma janela com a mensagem “Desculpe, nenhuma resposta disponível!”, conforme a figura.



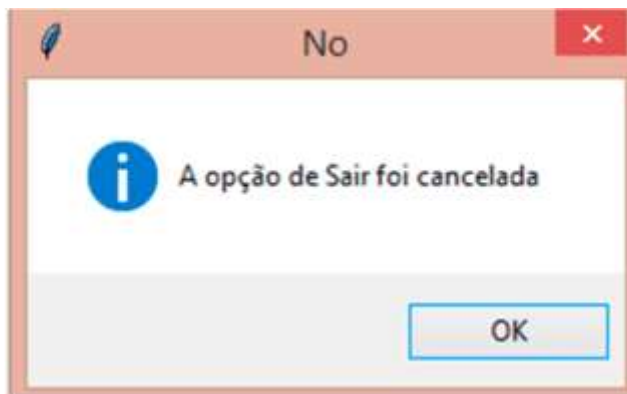
Sair

Caso o usuário tenha pressionado o botão “Sair”, aparecerá uma janela com a mensagem “Realmente quer sair?”, conforme mostrado na figura.



Resposta

Caso o usuário pressione o botão “Yes”, aparecerá uma janela com a mensagem “Ainda não foi implementado”, conforme mostra a imagem.



No

Por outro lado, se o usuário pressionar o botão “No” da janela da imagem, será exibida uma janela com a mensagem “A opção de Sair foi cancelada”, conforme mostrado na imagem.

Agora, vamos analisar os principais aspectos do código.

- Linha 4, 6, 7 e 9 - São instanciados componentes “messageDialog”.

Esse exemplo é bastante interessante, pois, apesar de pouca implementação, é possível que o usuário tenha bastante interação com o sistema.

Widget Combobox

O próximo exemplo apresenta como usar o widget “Combobox”. O código para gerar uma aplicação com esse componente é dado por:

```
import tkinter as tk

from tkinter import ttk

# Criação de uma janela tkinter
janela = tk.Tk()
janela.title('Combobox')
janela.geometry('500x250')

# Componente Label
ttk.Label(janela, text = "Combobox Widget",background = 'green',
foreground ="white",font = ("Times New Roman", 15)).grid(row = 0,
column = 1)

# Componente Label
```



```
ttk.Label(janela, text = "Selecione um mês :",font = ("Times New Roman", 10)).grid(column = 0,row = 5, padx = 10, pady = 25)
```

```
# Componente Combobox
```

```
n = tk.StringVar()
```

```
escolha = ttk.Combobox(janela, width = 27, textvariable = n)
```

```
# Adição de itens no Combobox
```

```
escolha['values'] = (' Janeiro',' Fevereiro',' Março',' Abril',' Maio',' Junho','  
Julho',' Agosto',' Setembro',' Outubro',' Novembro',' Dezembro')
```

```
escolha.grid(column = 1, row = 5)
```

```
escolha.current()
```

```
janela.mainloop()
```

O código produzirá uma janela com um “combobox”:



Exemplo do componente combobox.

Agora, vamos analisar os principais aspectos do código.

- Linha 17 - É instanciado um componente “combobox”.
- Linha 19 a 30 - Os meses do ano são atribuídos ao componente.