

ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

AULA 13 – INTERFACE GRÁFICA

WALTER TRAVASSOS SARINHO

@PROF.WALTERTRAVASSOS

WALTER.TRAVASSOS@SECTRAS.EDU.BR



Aula de Hoje

- Interface Gráfica com tkinter.
- Atividade de revisão.

GUI em Python - tkinter

GUI em Python

- GUI = Graphical User Interface.
- Python provê vários módulos (scripts) para o desenvolvimento de interfaces gráficas.
- Uma das mais importantes (e antigas) é a interface tkinter. Ele provê uma biblioteca padrão do Python para a programação de interfaces gráficas.
- Com ele podemos criar de forma rápida e fácil aplicações com GUI.

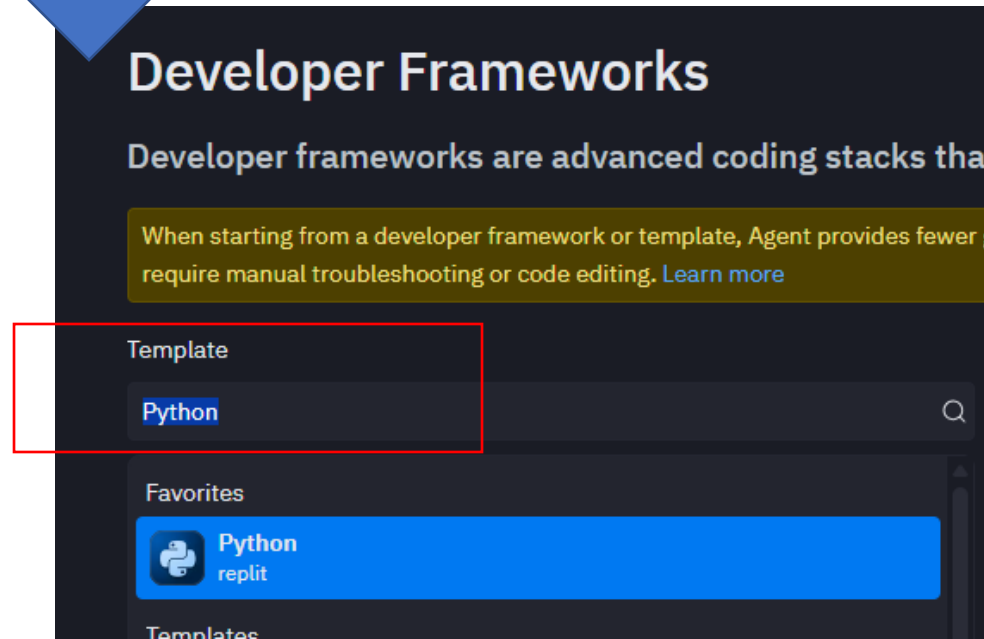
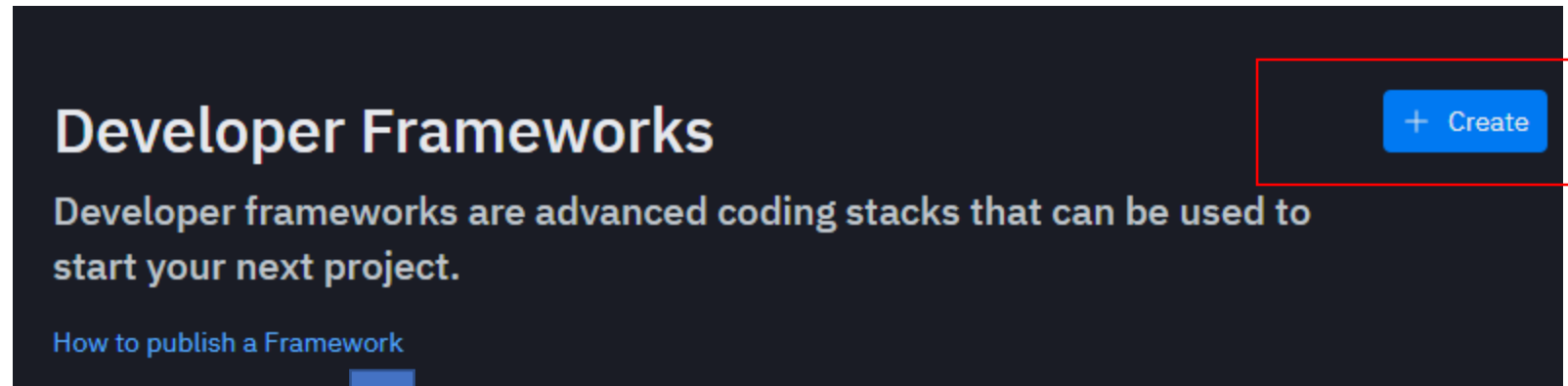
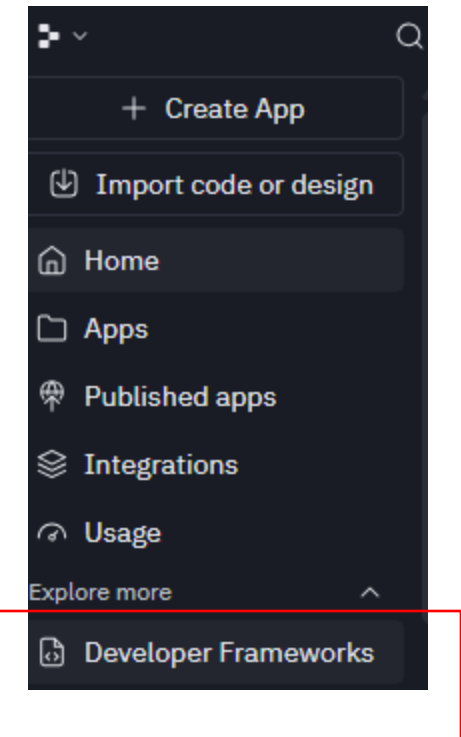
História

- O tkinter é um **wrapper Python** para o toolkit Tk, que foi originalmente desenvolvido para a linguagem de programação TCL (Tool Command Language). O Tkinter começou a ser incluído como uma biblioteca padrão na distribuição principal do Python a partir da versão 1.4, lançada em 1996.
- Por causa de sua longa história e integração direta com o Python, o Tkinter é **frequentemente a escolha inicial para muitos desenvolvedores Python** que desejam criar interfaces gráficas simples e funcionais.
- Embora Tkinter possa ser considerado mais simples em comparação com algumas outras opções de GUI mais recentes, como **PyQt** ou **Kivy**, ele ainda é amplamente utilizado em uma variedade de aplicativos e projetos devido à sua confiabilidade e facilidade de aprendizado.

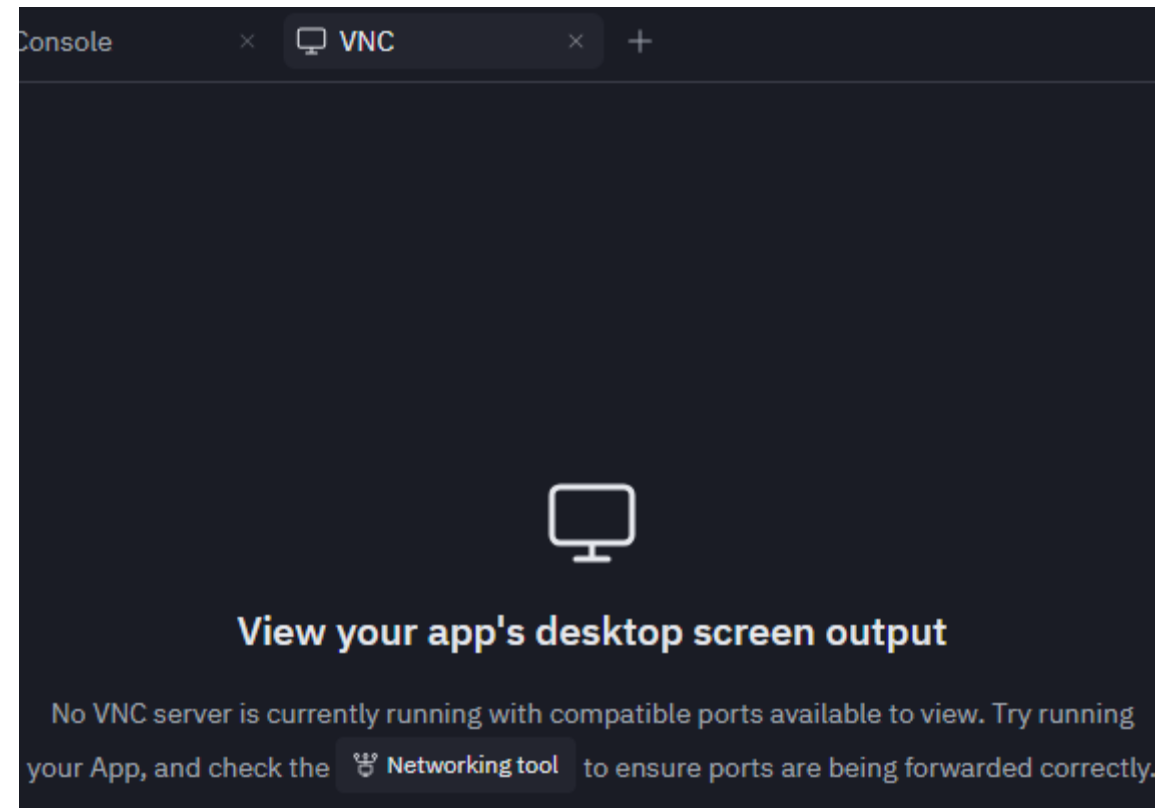
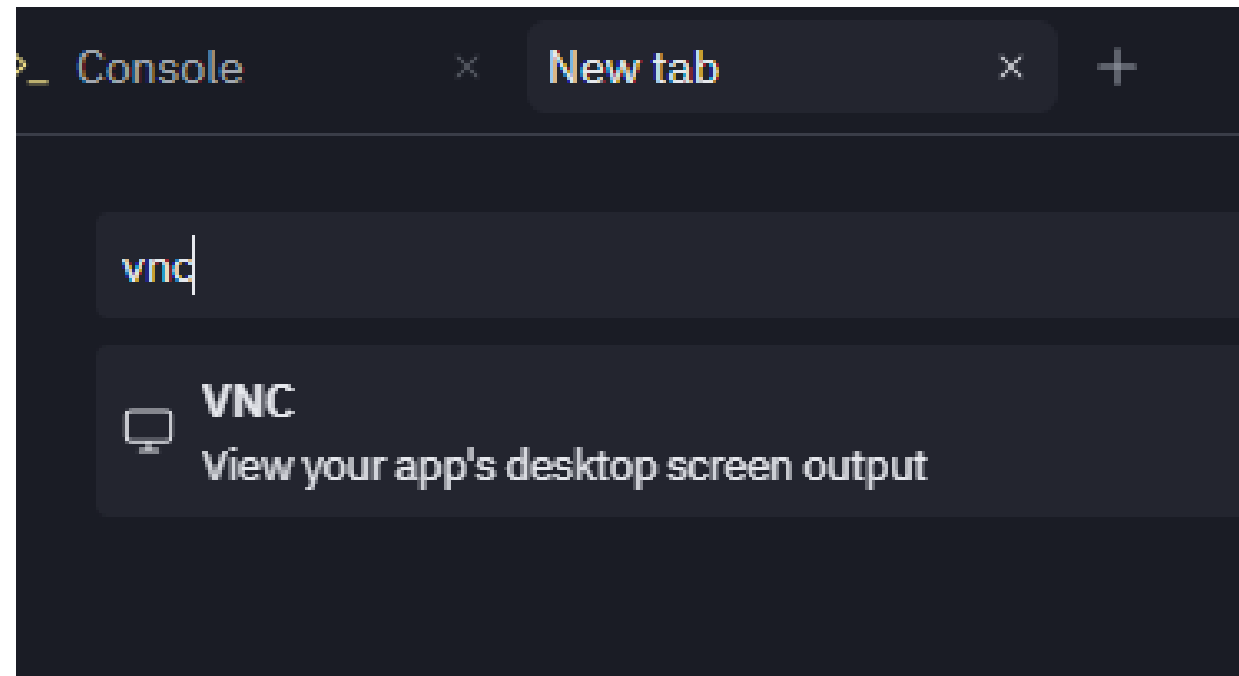
Passo a passo para utilizar o tkinter em módulos Python

1. Importar o módulo tkinter.
2. Criar uma janela principal (a janela inicial do módulo).
3. Adicionar um ou mais itens (widgets) na janela.
4. Adicionar um evento de loop principal para obter ações (eventos) do usuário.
5. Implementar funções que serão executadas de acordo com as ações do usuário.

Abram o replit



Abram o VNC no replit



Exemplo 1

Criando a Janela Principal

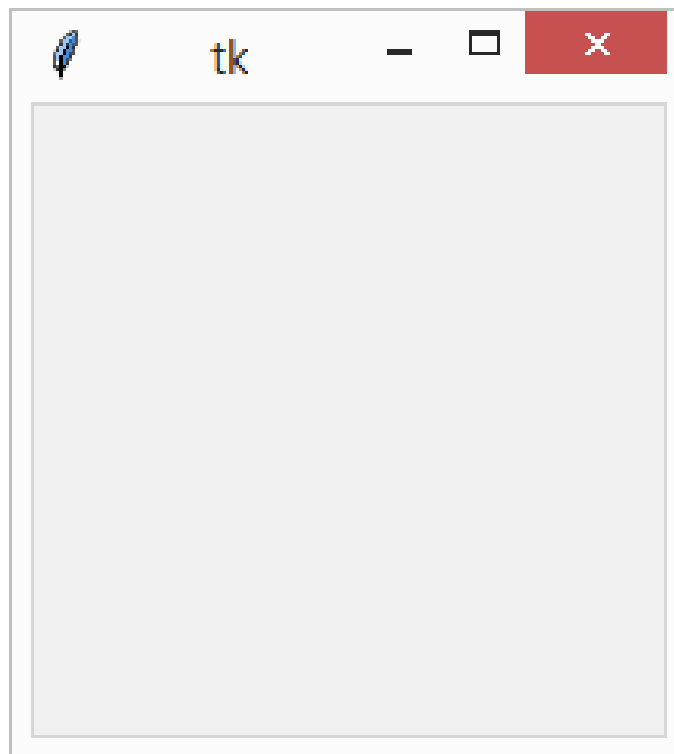
Exemplo 1

Criando a Janela Principal

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

Exemplo 1

Criando a Janela Principal



Exemplo 2

Mudando o Título da Janela Principal

Exemplo 2

Mudando o Título da Janela Principal

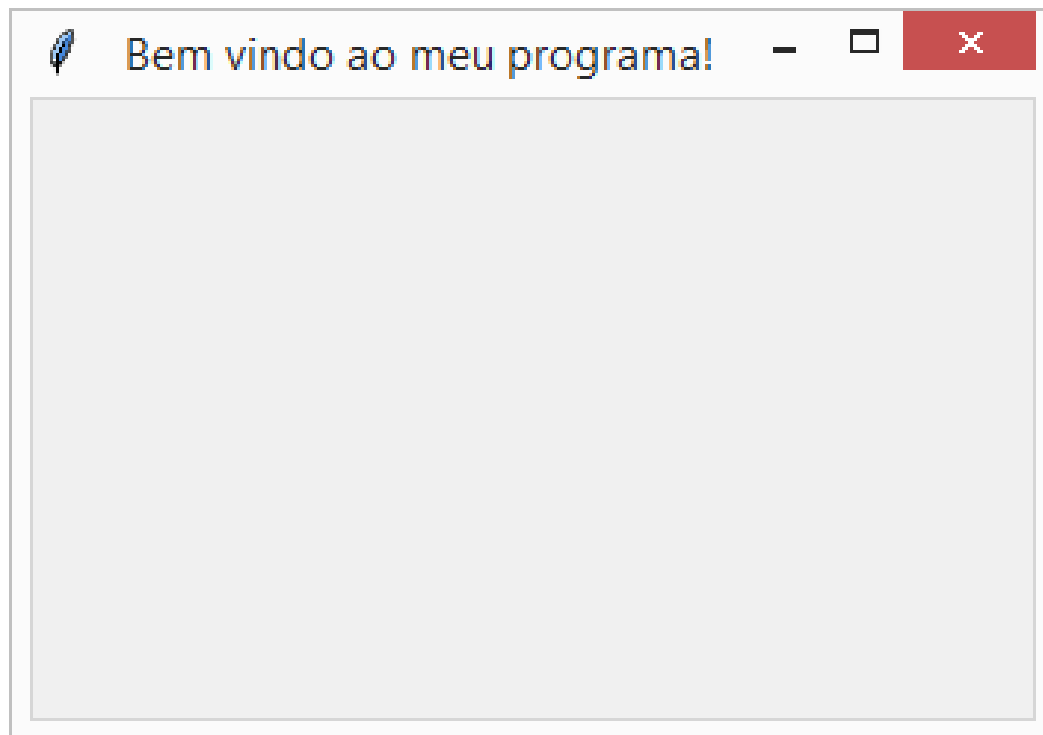
```
import tkinter

janela = tkinter.Tk()
janela.title('Bem vindo ao meu programa!')

janela.mainloop()
```

Exemplo 2

Mudando o Título da Janela Principal



Exemplo 3

Mudando o Tamanho da Janela Principal

Exemplo 3

Mudando o Tamanho da Janela Principal

```
import tkinter

janela = tkinter.Tk()
janela.title('Bem vindo ao meu programa!')
janela.geometry('400x300')

janela.mainloop()
```

Exemplo 3

Mudando o Tamanho da Janela Principal



Exemplo 4

Adicionando Widgets na Janela Principal

Rótulo, campo de texto e botão.

Exemplo 4

Adicionando Widgets na Janela Principal: Rótulo

```
import tkinter

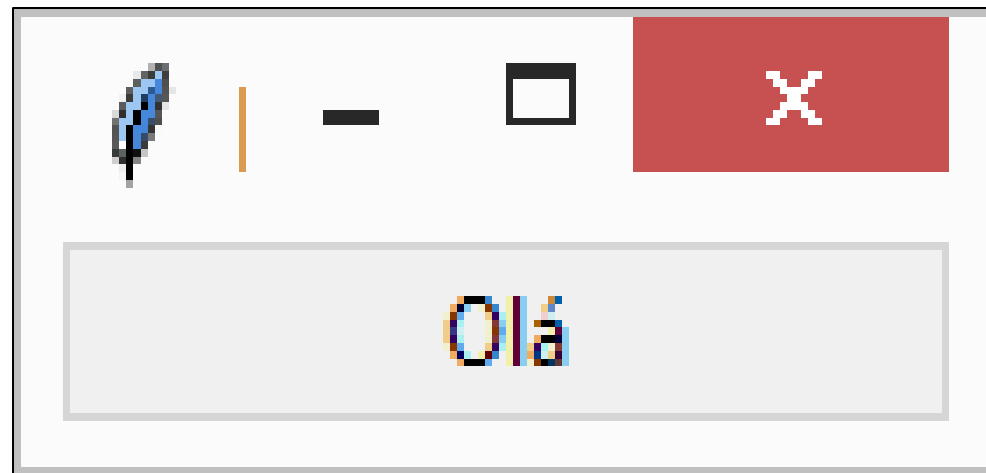
janela = tkinter.Tk()
janela.title('Bem vindo ao meu programa!')

rotulo = tkinter.Label(janela, text='Olá')
rotulo.pack()

janela.mainloop()
```

Exemplo 4

Adicionando Widgets na Janela Principal



Exemplo 4

Adicionando Widgets na Janela Principal: Campo de Texto

```
import tkinter

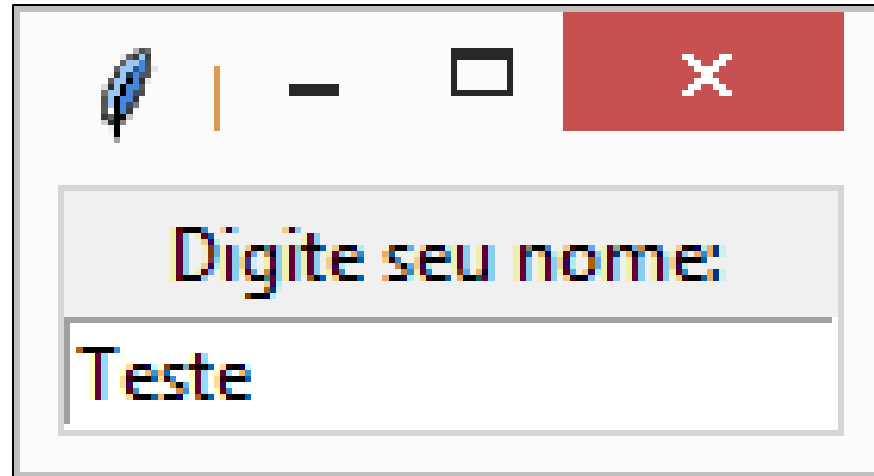
janela = tkinter.Tk()
janela.title('Bem vindo ao meu programa!')
rotulo = tkinter.Label(janela, text='Digite seu nome:')
rotulo.pack()

campo = tkinter.Entry(janela)
campo.pack()

janela.mainloop()
```

Exemplo 4

Adicionando Widgets na Janela Principal



Exemplo 4

Adicionando Widgets na Janela Principal: Botão

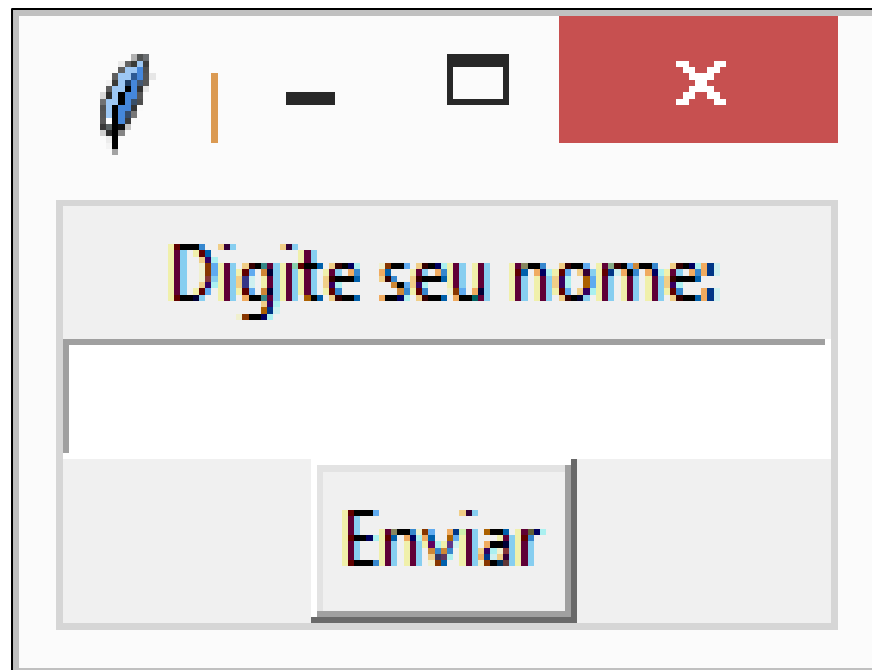
```
import tkinter
janela = tkinter.Tk()
janela.title('Bem vindo ao meu programa!')
rotulo = tkinter.Label(janela, text='Digite seu nome:')
rotulo.pack()
campo = tkinter.Entry(janela)
campo.pack()

botao = tkinter.Button(janela, text='Enviar')
botao.pack()

janela.mainloop()
```

Exemplo 4

Adicionando Widgets na Janela Principal



Exemplo 5

Adicionando um Evento

Exemplo 5

Adicionando um Evento: Botão

```
import tkinter

def clicou():
    print('O usuário clicou no botão!')

janela = tkinter.Tk()
(...)
botao = tkinter.Button(janela, text='Enviar', command=clicou)
botao.pack()

janela.mainloop()
```

Exemplo 5

Adicionando um Evento na GUI.

Pegue um nome digitado no campo de texto e apresente num rótulo.

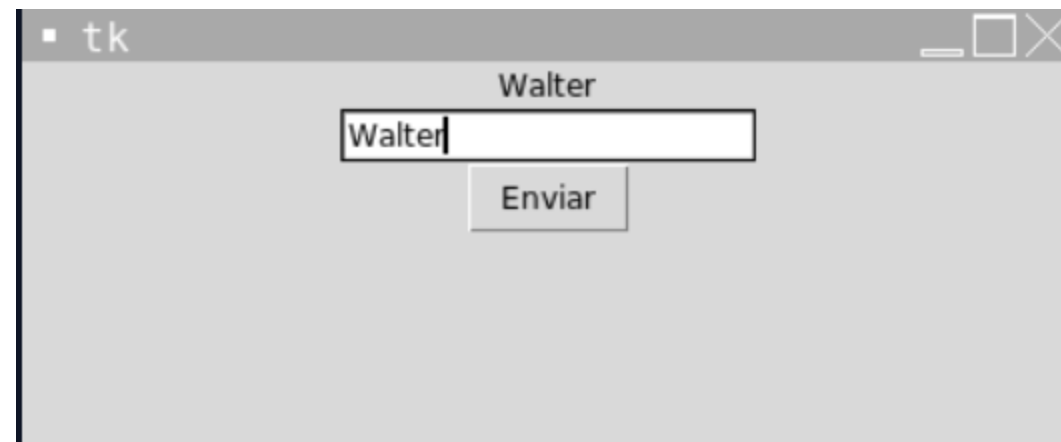
```
import tkinter

def clicou():
    resposta = campo.get()
    rotulo.config(text = resposta)

janela = tkinter.Tk()
(...)
botao = tkinter.Button(janela, text='Enviar', command=clicou)
botao.pack()

janela.mainloop()
```

Exemplo 5



Exemplo 6

Mostrando o nome digitado no console.

```
import tkinter
def clicou():
    print('Bem vindo, ' + campo.get())

janela = tkinter.Tk()

rotulo = tkinter.Label(janela, text='Nome:')
rotulo.pack()

campo = tkinter.Entry(janela)
campo.pack()

botao = tkinter.Button(janela, text='Enviar', command=clicou)
botao.pack()
janela.mainloop()
```

Exercícios

Exercício 1

- Faça um programa que leia o login e a senha de um usuário a partir de uma GUI.
- Após clicar no botão “**logar**”, um evento deverá checar se o login e a senha foram digitados corretamente, exibindo uma mensagem num rótulo abaixo do botão.
- A mensagem deve ser de sucesso ou fracasso.
- Considere que o login padrão do sistema é **café** e a senha é **forte**.
- Dica: crie uma função chamada `checaSenha()`.

Exercício 1 – Resposta parte 1

```
import tkinter

janela = tkinter.Tk()
janela.title('Login')

rotulo_login = tkinter.Label(janela, text='Login:')
rotulo_login.pack()
login = tkinter.Entry(janela)
login.pack()

rotulo_senha = tkinter.Label(janela, text='Senha:')
rotulo_senha.pack()
senha = tkinter.Entry(janela)
senha.pack()

# ...
```

Exercício 1 – Resposta parte 2

```
# ...
def checaSenha():
    if login.get() == 'café' and senha.get() == 'forte':
        rotulo_resultado.config(text = 'Sucesso')
    else:
        rotulo_resultado.config(text = 'Falhou')

botao = tkinter.Button(janela, text='Logar', command=checaSenha)
botao.pack()

rotulo_resultado = tkinter.Label(janela, text='Resultado')
rotulo_resultado.pack()

janela.mainloop()
```

Tkinter

Formatando uma Interface Gráfica

Formatando uma Interface Gráfica

- O tkinter oferece janelas e widgets com uma aparência padrão, no entanto, esses componentes de GUI podem ser customizados.
- Cor de fundo, inserção de ícones e imagens e posicionamento de widgets são algumas configurações que podem ser customizadas no tkinter.

Exemplo 1

Mudar a Cor de Fundo de uma Janela

Exemplo 1

Mudar a Cor de Fundo de uma Janela

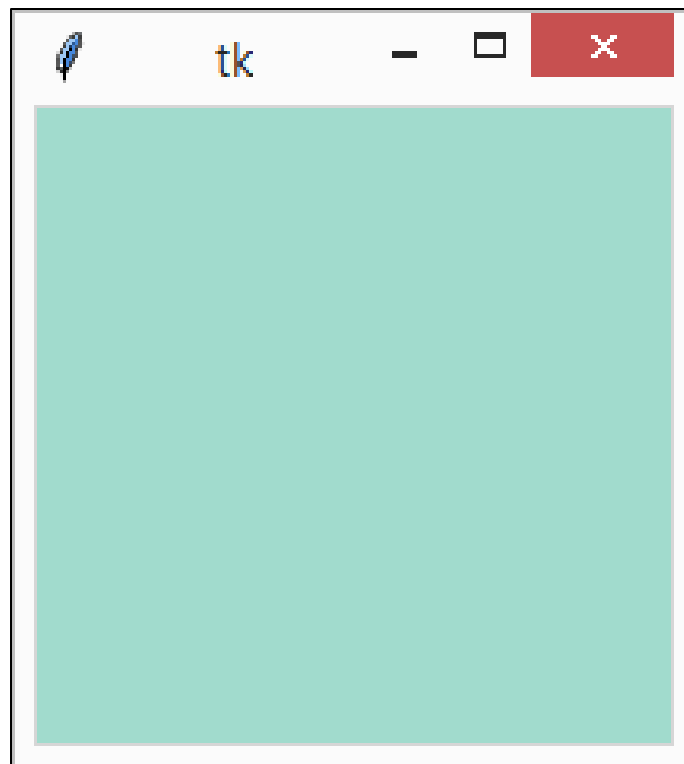
```
import tkinter

janela = tkinter.Tk()
janela.configure(background="#a1dbcd")
#janela.configure(background="cyan")

janela.mainloop()
```

Exemplo 1

Mudar a Cor de Fundo de uma Janela



Exemplo 2

Mudando a Cor de Fundo de um Widget

Exemplo 2

Mudando a Cor de Fundo de um Widget

```
import tkinter

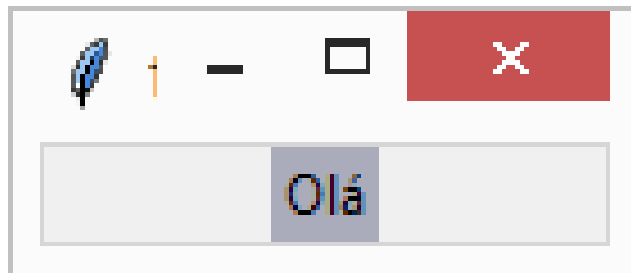
janela = tkinter.Tk()

rotulo = tkinter.Label(janela, text='Olá', bg="#a1dbc1")
rotulo.pack()

janela.mainloop()
```

Exemplo 2

Mudando a Cor de Fundo de um Widget



Exemplo 3

Mudando a Cor da Fonte de um Widget

Exemplo 3

Mudando a Cor da Fonte de um Widget

```
import tkinter

janela = tkinter.Tk()

rotulo = tkinter.Label(janela, text='Olá', bg="#000000", fg="#FFFFFF")
rotulo.pack()

janela.mainloop()
```

Exemplo 3

Mudando a Cor da Fonte de um Widget



Exemplo 4

Adicionando uma Imagem em uma Janela

Exemplo 4

Adicionando uma Imagem em uma Janela

```
import tkinter

janela = tkinter.Tk()

cafe = tkinter.PhotoImage(file='café.png')

rotulo = tkinter.Label(janela, image=cafe)
rotulo.pack()

janela.mainloop()
```

Exemplo 4

Adicionando uma Imagem em uma Janela



Os principais widgets do Tkinter

- **Label:** Utilizado para exibir texto ou imagens na interface gráfica.
- **Button:** Um botão clicável que pode executar uma ação quando pressionado.
- **Entry:** Uma caixa de entrada de texto onde o usuário pode digitar informações.
- **Checkbutton:** Uma caixa de seleção que permite ao usuário selecionar ou desmarcar uma opção.
- **Radiobutton:** Um grupo de botões de opção dos quais apenas um pode ser selecionado por vez.
- **Listbox:** Uma caixa de listagem que exibe uma lista de itens, permitindo que o usuário selecione um ou mais itens.
- **Scrollbar:** Uma barra de rolagem que permite rolar conteúdo em um widget, como uma Listbox ou Text.
- **Canvas:** Uma área onde gráficos e outros elementos podem ser desenhados e manipulados.
- **Frame:** Um contêiner retangular que pode ser usado para organizar e agrupar outros widgets.
- **Menu:** Um menu de opções que pode conter itens de menu, como opções de arquivo, editar, etc.
- **Message:** Similar a um Label, mas pode exibir texto de várias linhas.
- **Scale:** Uma barra de controle deslizante que permite ao usuário selecionar um valor em um intervalo.

Exercícios

Exercício 1

Faça um programa que leia (a partir de uma GUI) a base e a altura de um retângulo. Após clicar no botão “Calcular Área”, um evento deverá calcular a área do retângulo, exibindo o resultado no console.

OBS.: Tente formatar a GUI!

Exercício 1 – Resposta parte 1

```
import tkinter

janela = tkinter.Tk()
janela.title('Área do retângulo')

rotulo_base = tkinter.Label(janela, text='Base:')
rotulo_base.pack()
base = tkinter.Entry(janela)
base.pack()

rotulo_altura = tkinter.Label(janela, text='Altura:')
rotulo_altura.pack()
altura = tkinter.Entry(janela)
altura.pack()

# ...
```

Exercício 1 – Resposta parte 2

```
# ...  
  
def calculaArea():  
    area = float(base.get())*float(altura.get())  
  
    tkinter.messagebox.showinfo('Área: ', area)  
  
botao = tkinter.Button(janela, text='Calcular Área', command=calculaArea)  
botao.pack()  
  
janela.mainloop()
```

Exercício 2

Faça um programa que leia (a partir de uma GUI) o nome de um aluno, a disciplina que o aluno se matriculou e as 3 notas obtidas durante o semestre.

Após clicar no botão “Checar”, um evento deverá calcular a média do aluno, notificando o aluno se o mesmo está aprovado ou reprovado (considere que a média da disciplina é 7).

Por exemplo: O aluno _____ está reprovado na disciplina
_____.

Exercício 2 – Resposta parte 1

```
import tkinter

janela = tkinter.Tk()
janela.title('Sistema Acadêmico')

rotulo_nome = tkinter.Label(janela, text='Nome do Aluno:')
rotulo_nome.pack()
nome = tkinter.Entry(janela)
nome.pack()

rotulo_disciplina = tkinter.Label(janela, text='Disciplina:')
rotulo_disciplina.pack()
disciplina = tkinter.Entry(janela)
disciplina.pack()
# ...
```

Exercício 2 – Resposta parte 2

```
# ...  
  
rotulo_nota1 = tkinter.Label(janela, text='Nota 1:')  
rotulo_nota1.pack()  
nota1 = tkinter.Entry(janela)  
nota1.pack()  
  
rotulo_nota2 = tkinter.Label(janela, text='Nota 2:')  
rotulo_nota2.pack()  
nota2 = tkinter.Entry(janela)  
nota2.pack()  
  
rotulo_nota3 = tkinter.Label(janela, text='Nota 3:')  
rotulo_nota3.pack()  
nota3 = tkinter.Entry(janela)  
nota3.pack()  
  
# ...
```

Exercício 2 – Resposta parte 3

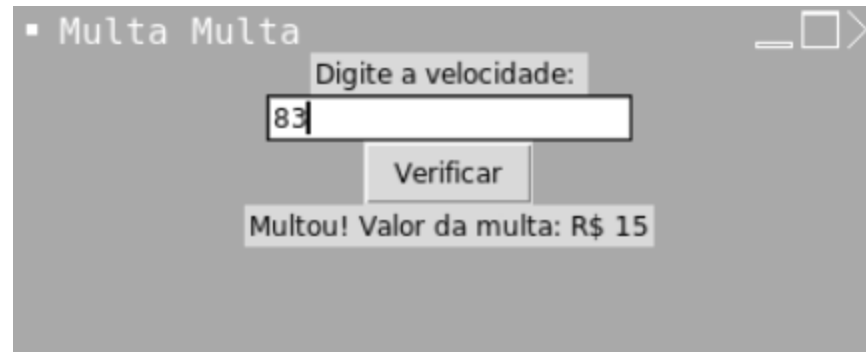
```
# ...
def calculaMedia():
    media = (float(nota1.get())+float(nota2.get())
            +float(nota3.get())) / 3
    if media >= 7:
        tkinter.messagebox.showinfo('Aprovado!', 'O aluno ' +
                                    nome.get() +
                                    ' está aprovado na disciplina ' +
                                    disciplina.get())
    else:
        tkinter.messagebox.showinfo('Reprovado!', 'O aluno ' +
                                    nome.get() +
                                    ' está reprovado na disciplina ' +
                                    disciplina.get())

botao = tkinter.Button(janela, text='Checar',
                      command=calculaMedia)
botao.pack()

janela.mainloop()
```

Exercício 3

- Faça um programa que leia (a partir de uma GUI) a velocidade do carro do usuário.
- Após clicar no botão “Checar Multa”, um evento deverá verificar se o usuário deverá ser multado ou não:
- Caso ultrapasse 80 km/h, o evento deverá exibir uma mensagem dizendo que o usuário foi multado.
- Neste caso, exiba o valor da multa, cobrando R\$ 5,00 por km acima de 80 km/h.



Exercício 3 – Resposta parte 1

```
1  import tkinter
2
3  def checaMulta():
4      velocidade = int(campo.get())
5
6      if velocidade < 80:
7          resultado.config(text="Sem multa")
8      else:
9          excedente = velocidade - 80
10         total = excedente * 5
11         resposta = "Multou! Valor da multa: R$ " + str(total)
12         resultado.config(text=resposta)
```

Exercício 3 – Resposta parte 2

```
14 janela = tkinter.Tk()
15 janela.title("Multa Multa")
16 janela.geometry('400x400')
17 janela.configure(background="darkgrey")
18
19 rotulo = tkinter.Label(janela, text="Digite a velocidade: ")
20 rotulo.pack()
21
22 campo = tkinter.Entry(janela)
23 campo.pack()
24
25 botao = tkinter.Button(janela, text="Verificar", command=checaMulta)
26 botao.pack()
27
28 resultado = tkinter.Label(janela, text="Status")
29 resultado.pack()
30
31 janela.mainloop()
```

Exercício 4

Faça um programa que leia (a partir de uma GUI) a distância que um passageiro deseja percorrer em km.

Após clicar no botão “Calcular Preço”, um evento deverá calcular o preço da passagem, cobrando R\$ 0,50 por km para viagens de até 200 km, e R\$ 0,45 para viagens mais longas.

Exercício 5

- Faça um programa que aprova o empréstimo bancário para a compra de uma casa. O programa deverá ler (a partir de uma GUI) o valor da casa a comprar, o salário do usuário e a quantidade de anos a pagar.
- Após clicar no botão “Calcula Prestação”, um evento deverá calcular o valor da prestação como sendo o valor da casa a comprar dividido pelo número de meses a pagar.
- OBS.: O valor da prestação mensal não pode ser superior a 30% do salário. Caso isso ocorra, o empréstimo não deve ser aprovado.

Exercício 6

- Faça um programa que calcule o preço a pagar pelo fornecimento de energia elétrica. O programa deverá ler (a partir de uma GUI) a quantidade de kWh consumida e o tipo de instalação: R para residências, I para indústrias e C para comércios.
- Após clicar no botão “Calcular Preço”, um evento deverá calcular o preço a pagar de acordo com a tabela apresentada no próximo slide.

Exercício 6 - continuação

Tipo	Faixa (kWh)	Preço
Residencial	Até 500	R\$ 0,40
	Acima de 500	R\$ 0,65
Comercial	Até 1000	R\$ 0,55
	Acima de 1000	R\$ 0,60
Industrial	Até 5000	R\$ 0,55
	Acima de 5000	R\$ 0,60

Exercício 7

- Faça um programa que controle uma pequena máquina registradora. O programa deverá solicitar (a partir de uma GUI) o código de um produto e a quantidade comprada. Utilize a tabela de códigos do próximo slide para obter o preço de cada produto.
- Após clicar no botão “Adicionar Valor”, um evento deverá adicionar o total a ser pago para o produto selecionado para o total geral e limpar os campos de código e quantidade para que o usuário possa adicionar um novo produto.
- Após clicar no botão “Exibir Total a Pagar”, um evento deverá exibir o total geral a ser pago pelo cliente.

Exercício 7 - continuação

Código	Preço
1	0,50
2	1,00
3	4,00
5	7,00
9	8,00

Outras bibliotecas gráficas

Outras GUI para Python

- PyQt/PySide: Esses módulos fornecem wrappers para a biblioteca Qt, que é uma estrutura de desenvolvimento de GUI multiplataforma amplamente utilizada. O PyQt é baseado na licença GPL, enquanto o PySide tem uma licença mais permissiva.
- wxPython: É uma ligação para o wxWidgets, uma biblioteca de GUI C++, oferecendo uma alternativa madura e robusta para o desenvolvimento de aplicativos de desktop em Python.
- Kivy: É uma biblioteca Python de código aberto para desenvolvimento de aplicativos móveis e outras interfaces multitouch. Ele é multiplataforma e suporta não apenas desktops, mas também dispositivos móveis (Android e iOS) e até mesmo Raspberry Pi.
- GTK+ (PyGTK): É uma biblioteca multiplataforma para criar interfaces gráficas. O PyGTK é um wrapper Python para GTK+, permitindo que os desenvolvedores criem aplicativos desktop que se integram ao ambiente de desktop GNOME.
- PyGUI: É uma biblioteca de interface gráfica de usuário simples e de plataforma cruzada para Python. É uma opção mais leve e simples, adequada para aplicativos que não exigem recursos avançados de GUI.
- Toga: Uma biblioteca de interface gráfica multiplataforma que tem como objetivo fornecer uma API unificada para desenvolvimento de aplicativos de desktop e móveis em Python.

Outras GUI para Python

- PySimpleGUI

Explore o assunto!

Referências

USING PYTHON. GUI Programming. [Internet]. [citado em 2015 Nov 10]. Disponível em: <https://usingpython.com> .

ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

AULA 13 – INTERFACE GRÁFICA

WALTER TRAVASSOS SARINHO

@WALTEROPROFESSOR

WALTER.TRAVASSOS@SECTRAS.EDU.BR

