Algoritmia e Estruturas de Dados

A biblioteca Tkinter / CustomTkinter

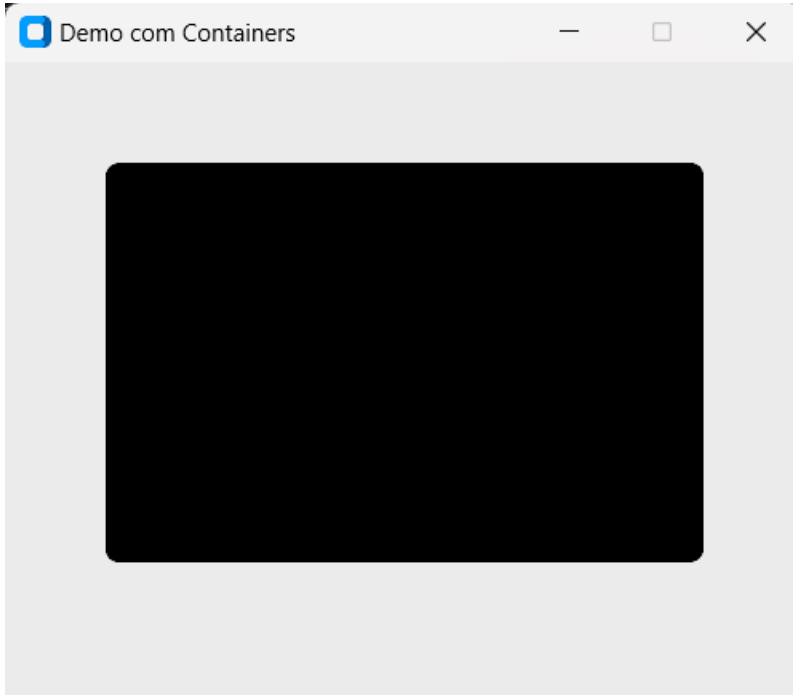
Parte II - Containers

Licenciatura em
Tecnologias e Sistemas de Informação para a Web
#ESMAD #P.PORTO

- Biblioteca CustomTkinter | Containers
  - Frame
  - ScrollableFrame
  - Tabview
  - CTkImage

❖ Containers

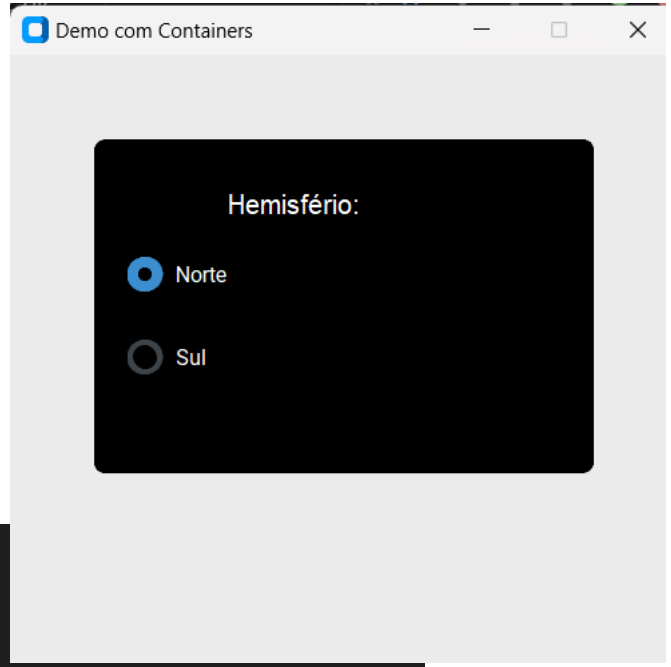
- **Frame**
  - width
  - height
  - border_width
  - Border_color
  - fg_color
  - bg_color




```
1  # FRAME
2  frameRadioButtons = customtkinter.CTkFrame(app, width=300, height=200, fg_color="black")
3  frameRadioButtons.place(x=50, y=50)
4
```

❖ Containers

- **Frame**
  - width
  - height
  - border_width
  - Border_color
  - fg_color, bg_color

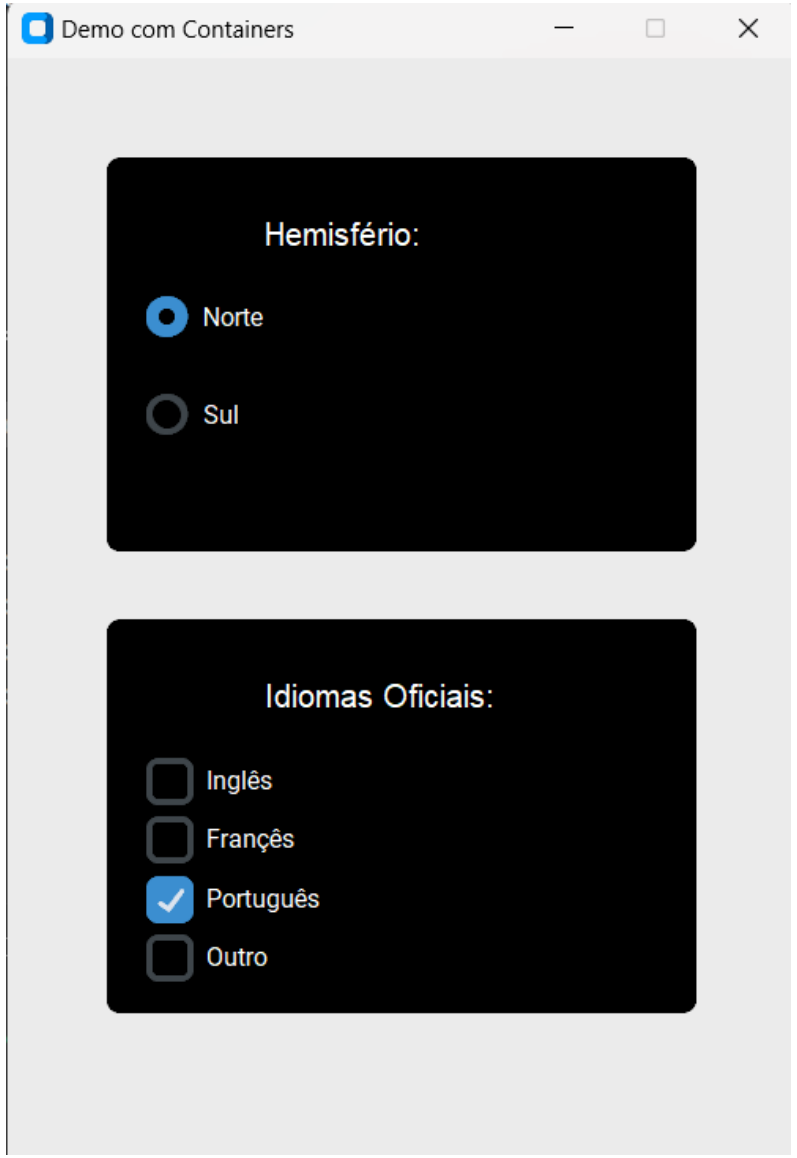



```python
# Label
labelHemisferio = customtkinter.CTkLabel(frameRadioButtons, text= "Hemisfério:", text_color="white",
                    bg_color="transparent", font=("Helvetica", 16))
labelHemisferio.place(x=80, y=25)

# RadioButtons
radioVariable = customtkinter.StringVar(value="Norte")
radiobutton1 = customtkinter.CTkRadioButton(frameRadioButtons, text="Norte", text_color="white",
                                            variable= radioVariable, value="Norte")
radiobutton1.place(x=20, y=70)
radiobutton2 = customtkinter.CTkRadioButton(frameRadioButtons, text="Sul",  text_color="white",
                                           variable= radioVariable, value="Sul")
radiobutton2.place(x=20, y=120)
```

❖ Containers

- **Frame**
  - width
  - height
  - border_width, border_color
  - fg_color, bg_color

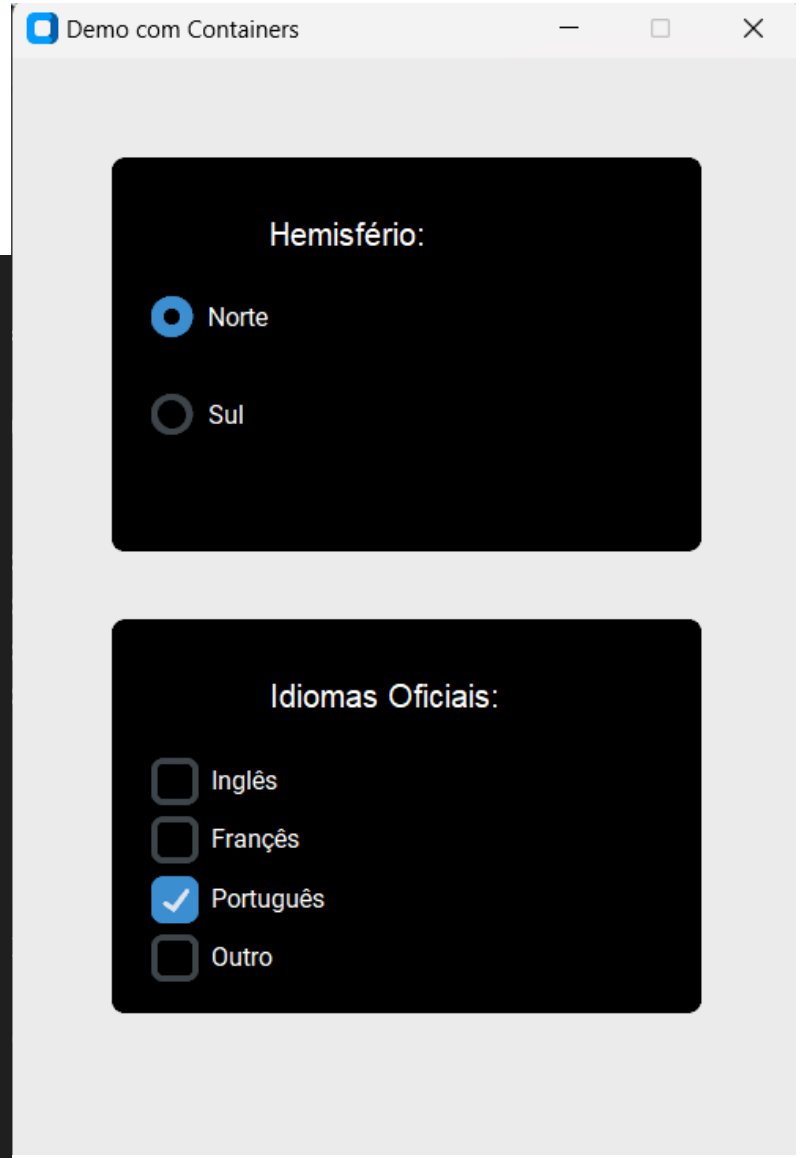
```python
# FRAME 2
frameIdiomas = customtkinter.CTkFrame(app, width=300, height=200, fg_color="black")
frameIdiomas.place(x=50, y=285)

```



❖ Containers

❑ Frame

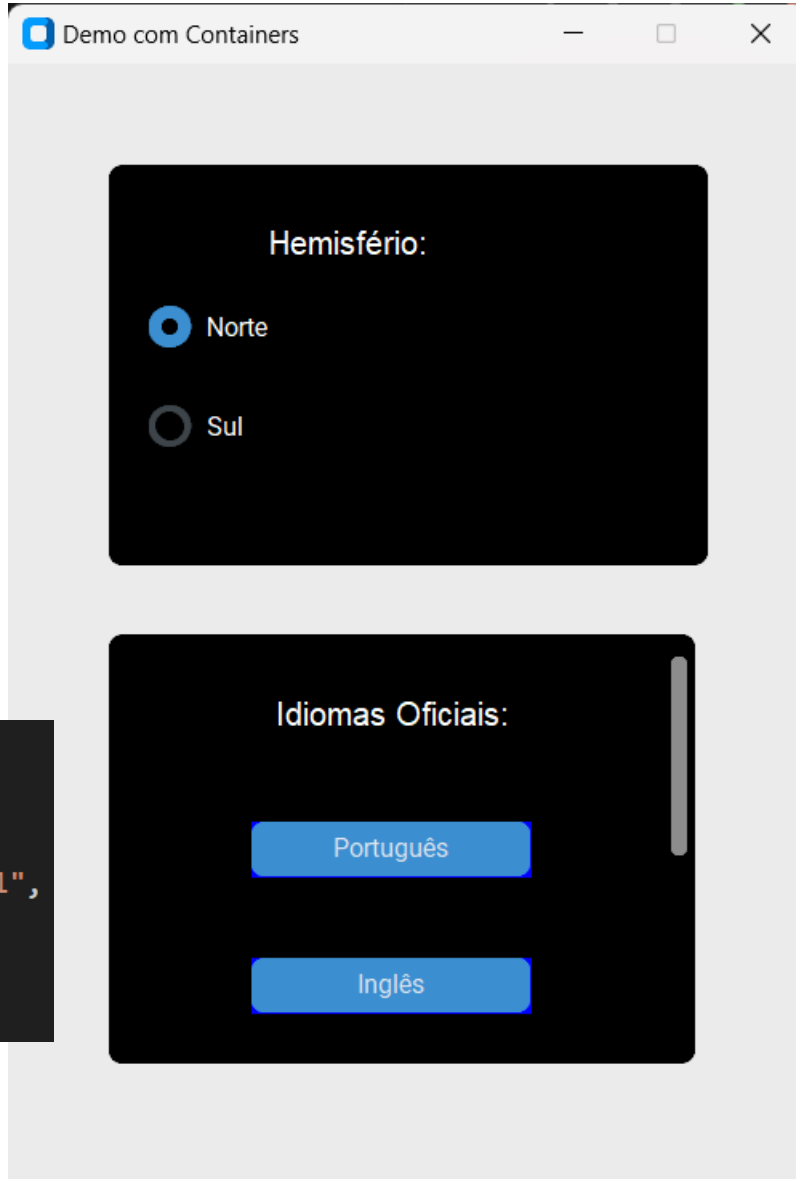
```
# Label
labelIdioma = customtkinter.CTkLabel(frameIdiomas, text= "Idiomas Oficiais:", text_color="white",
                    bg_color="transparent", font=("Helvetica", 16))
labelIdioma.place(x=80, y=25)

# CheckBox - Idioma Oficial
checkVar1 = customtkinter.StringVar(value="off")
checkVar2 = customtkinter.StringVar(value="off")
checkVar3 = customtkinter.StringVar(value="on")
checkVar4 = customtkinter.StringVar(value="off")

checkboxEN = customtkinter.CTkCheckBox(frameIdiomas, text="Inglês", text_color="white",
                                       variable=checkVar1, onvalue="on", offvalue="off")
checkboxFR = customtkinter.CTkCheckBox(frameIdiomas, text="Françês", text_color="white",
                                       variable=checkVar2, onvalue="on", offvalue="off")
checkboxPT = customtkinter.CTkCheckBox(frameIdiomas, text="Português", text_color="white",
                                       variable=checkVar3, onvalue="on", offvalue="off")
checkboxOT = customtkinter.CTkCheckBox(frameIdiomas, text="Outro", text_color="white",
                                       variable=checkVar4, onvalue="on", offvalue="off")

checkboxEN.place(x=20, y= 70)
checkboxFR.place(x=20, y= 100)
checkboxPT.place(x=20, y= 130)
checkboxOT.place(x=20, y= 160)
```



❖ Containers

- **ScrollableFrame**
  - width
  - height
  - border_width, border_color
  - fg_color, bg_color
  - orientation (“horizontal”, “vertical”)

```
# FRAME 2
frameIdiomas = customtkinter.CTkScrollableFrame(app, width=270, height=200, orientation="vertical",
                                                fg_color="black")
frameIdiomas.place(x=50, y=285)

```



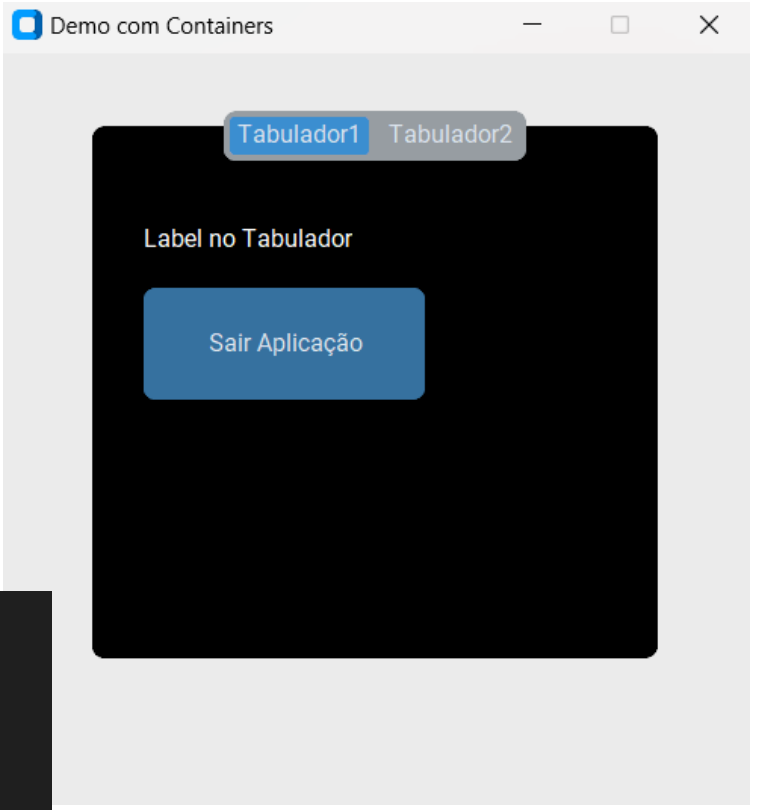
❖ Containers

- **ScrollableFrame**
  - width
  - height
  - border_width, border_color
  - fg_color, bg_color
  - orientation (“horizontal”, “vertical”)

```python
# Label
labelIdioma = customtkinter.CTkLabel(frameIdiomas, text= "Idiomas Oficiais:", text_color="white",
                    bg_color="transparent", font=("Helvetica", 16))
labelIdioma.pack(pady=20)

listIdiomas= ["Português", "Inglês", "Francês", "Mandarim", "Alemão"]
for i in range (len(listIdiomas)):
    customtkinter.CTkButton(frameIdiomas, bg_color="blue", text=listIdiomas[i],
           width= 200, height=40).pack(pady=10)

```

❖ Containers

- **Tabview**
  - width, height
  - border_color
  - fg_color, texto_color
  - state



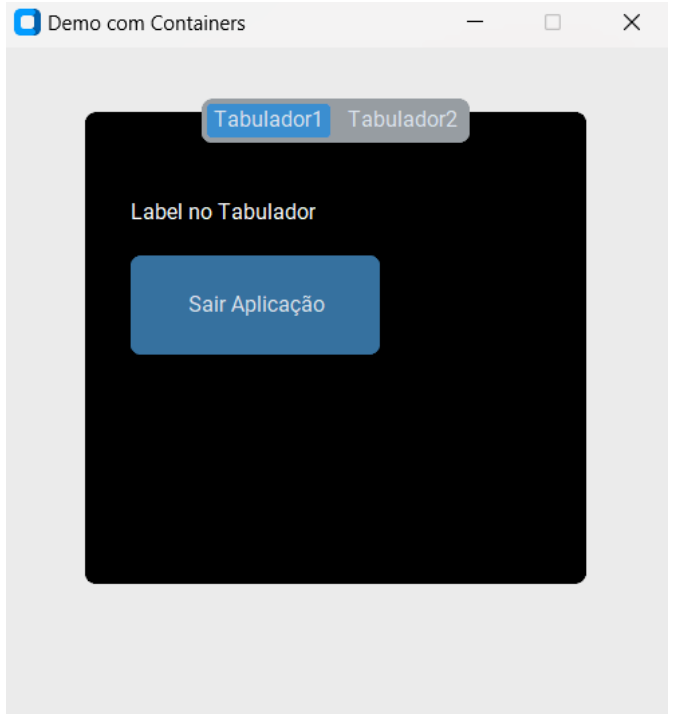

```python
# definir componente Tabview
tabview = customtkinter.CTkTabview(app, width=300, height=300, fg_color="black")
tabview.pack(padx=20, pady=20)

# Adicionar tabuladores ao tabview
tab1 = tabview.add("Tabulador1")  # adicionar tab
tab2 = tabview.add("Tabulador2")  # adicionar tab

# Definir o tabulador ativo / visível num dado momento
tabview.set("Tabulador1")

```

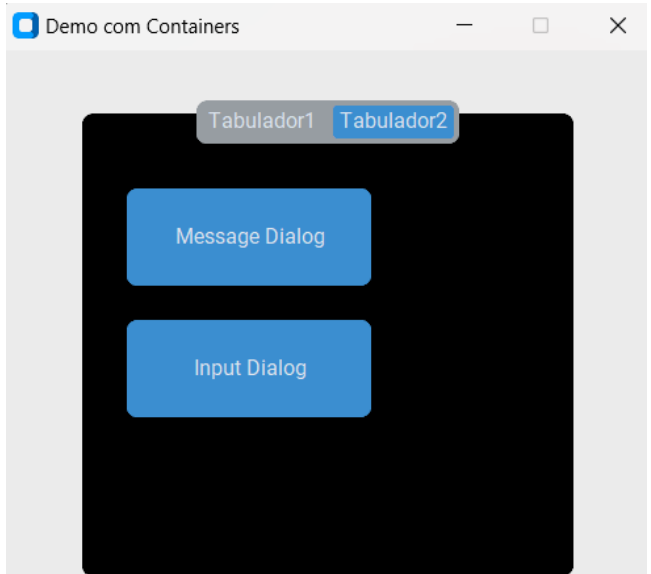
❖ Containers

❑ Tabview

- ❑ width, height
- ❑ border_color
- ❑ fg_color, texto_color
- ❑ state

```python
# Tabulador1
lblText = customtkinter.CTkLabel(tab1, text = "Label no Tabulador", text_color="white")
lblText.place(x=20, y=20)

btnTabulador = customtkinter.CTkButton(tab1, width=150, height=60,
                                text = "Sair Aplicação", command= app.destroy)
btnTabulador.place(x=20, y=60)

# Tabulador2
btnMessage = customtkinter.CTkButton(tab2, width=150, height=60,
                                text = "Message Dialog", command=messageDialog)
btnMessage.place(x=20, y=20)

btnDialog = customtkinter.CTkButton(tab2, width=150, height=60,
                                text = "Input Dialog", command=dialogEvent)
btnDialog.place(x=20, y=100)
```

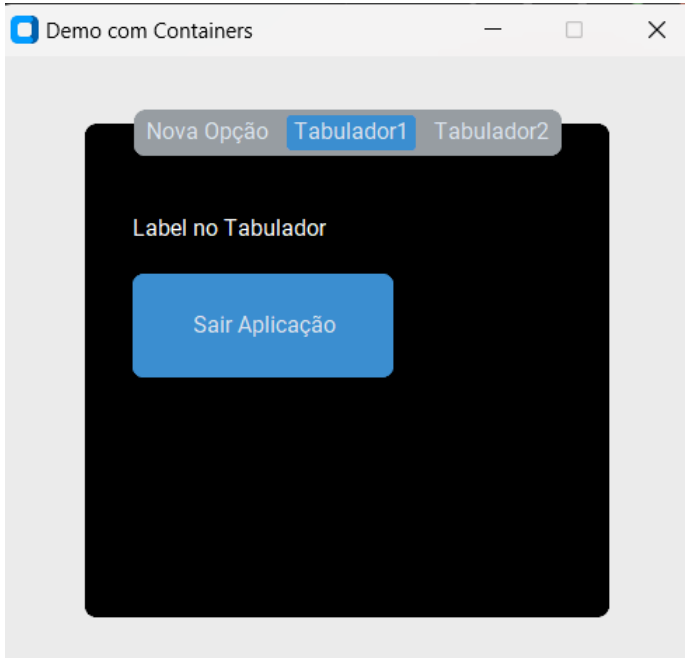


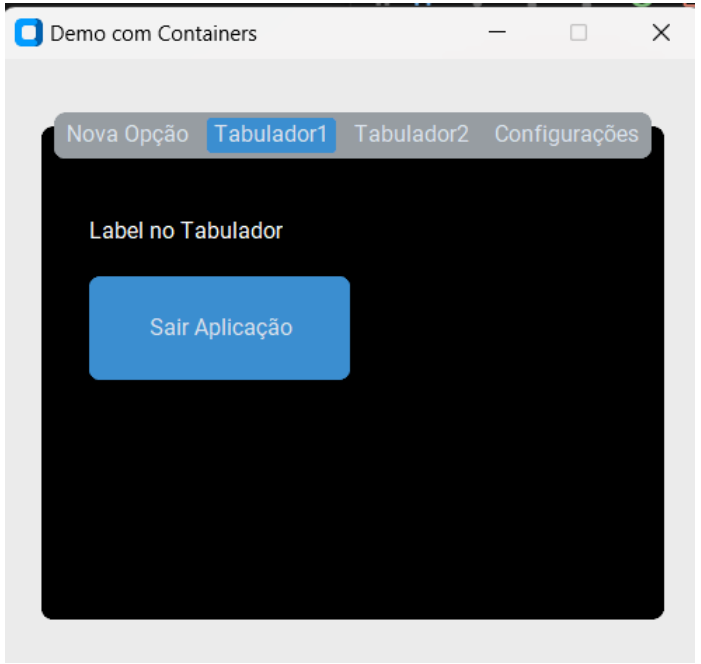


❖ Containers

- **Tabview**
  - Métodos
    - insert(index, name)
    - add(name)
    - rename(name1, name2)

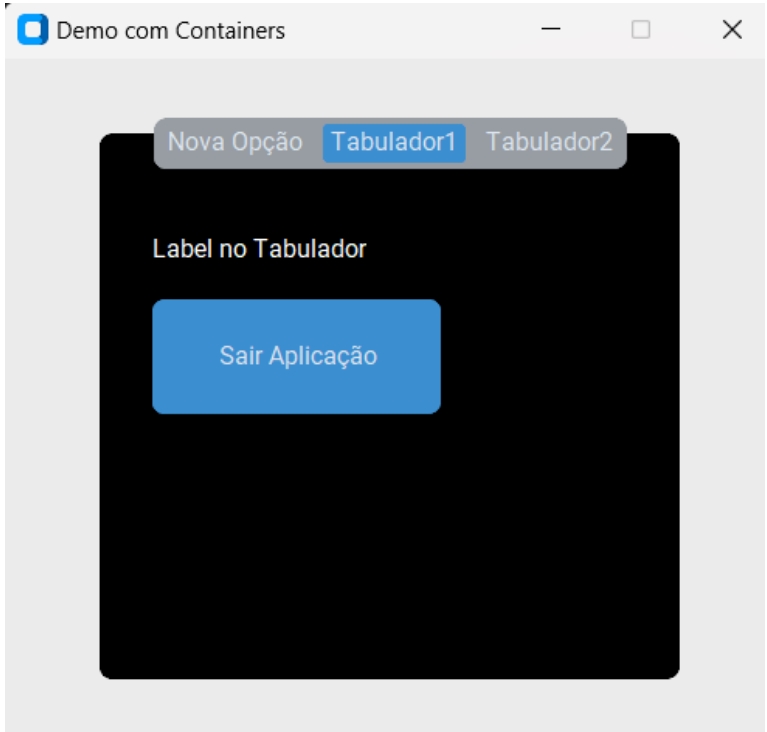
```python
#EXEMPLOS MÉTODOS

tabview.insert(0, "Nova Opção")
tabview.add("Configurações")
tabview.rename("Configurações", "Settings")
```





❖ Containers

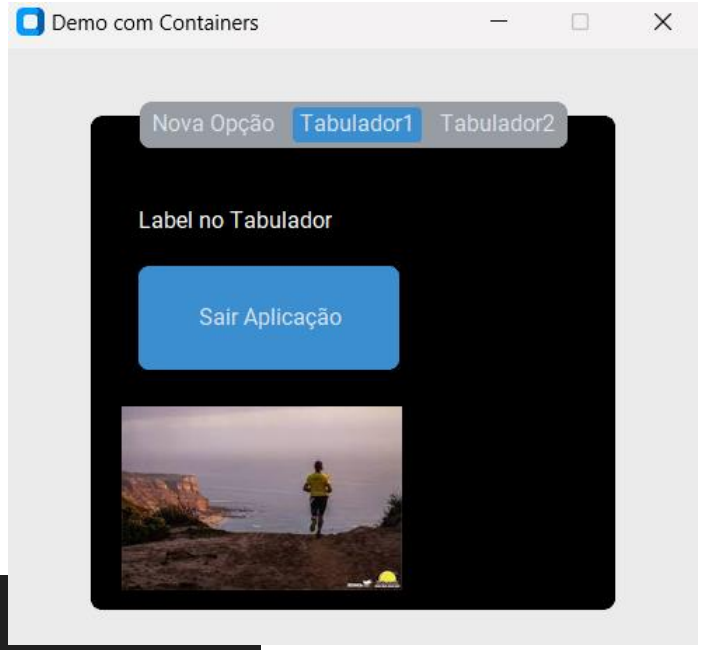
- ❑ **Tabview**
  - ❑ Métodos
    - ❑ insert(index, name)
    - ❑ add(name)
    - ❑ rename(name1, name2)




```
1  tabview.delete("Settings")
2
3  tabview.set("Tabulador1")   # define tabulador ativo
4  tabAtivo = tabview.get()   # Obtém o nome do tabvulador ativo
```

❖ Containers

- **CTkImage**
  - size (tuplo, width, height)
  - light_image (atribuir objeto imagem para o modo light)
  - dark_image (atribuir objeto imagem para o modo dark)




```
# CTKImage
tabview.set("Tabulador1")   # define tabulador ativo
# CTKImage define objeto do tipo Image
img1 = customtkinter.CTkImage(Image.open(".\\images\\Praia.png"), size=(160, 105))
# Ancorar image numa Label ou num Button
lblImage = customtkinter.CTkLabel(tab1, image = img1, text = "",  width = 160, height = 105)
lblImage.place(x=10, y=140)
```

❖ Containers

- ❑ **Exercício de consolidação**

- ❑ App: 400x700
- ❑ Tabview: 400x400
- ❑ 2 tabuladores: “Gestor de Presenças” e “Consultar”
- ❑ Imagem: size=(300, 280), contida numa Label
- ❑ Buttons: 300x30
- ❑ Messagebox para confirmar saída da aplicação
- ❑ TreeView: height=15
- ❑ Data e Hora com width=140, Temperatura com width = 140