PYTHON ACADEMY

COMO CRIAR INTERFACES GRÁFICAS (GUI) COM TKINTER E PYTHON

Aprenda a usar o Tkinter do Python para criar interfaces gráficas (GUI)

PYTHONACADEMY.COM.BR

Salve salve Pythonista! 👋

Hoje vamos abordar um tema bastante interessante: a criação de interfaces gráficas com **Python** e **Tkinter**.

Não só isso, mas vamos criar um executável que gera QR Codes!

O *Tkinter* é uma ferramenta comumente usada com interface gráfica para programas em *Python*.

Esta ferramenta é muito útil para a criação de interfaces das mais simples às mais complexas.

E de bônus, você ainda vai aprender a empacotar sua interface em um executável para Windows!

Sem enrolação, vamos ao conteúdo!

Instalação das dependências

O primeiro passo (sei que vocês já estão antenados, mas não custa lembrar) é instalar as dependências do projeto.

O Tkinter faz parte da biblioteca padrão do Python, portanto não precisamos instalá-lo via `pip`.

As outras dependências são: `qrcode` (para fazer a geração do QR Code) e a biblioteca de manipulação de imagens `pillow`.

Antes de fazer a instalação, crie e ative um ambiente virtual para não bagunçar as bibliotecas em seu computador.

> *Se ainda não sabe como criar um ambiente virtual, clique aqui e aprenda a criar ambientes virtuais utilizando o Virtualenv!*

Com o ambiente virtual criado e ativado, instale as dependências com o seguinte comando `pip`:

```
pip install qrcode pillow
```

Desenvolvimento do Código

Agora, abra sua IDE ou editor de código e crie um arquivo chamado `main.py`.

Vamos começar o código importando todos os módulos necessários, que já foram instalados via `pip`:

```
from tkinter import *
import qrcode
from PIL import Image, ImageDraw
from tkinter import messagebox
```

Agora vamos escrever a função que vai salvar o Código QR em uma imagem (caso contrário o Código QR será gerado, mas sem ser salvo no HD):

```
def gera_qr_code():
  url = website_entry.get()

  if len(url) == 0:
    messagebox.showinfo(
      title="Erro!",
      message="Favor insira uma URL válida")
  else:
    opcao_escolhida = messagebox.askokcancel(
      title=url,
      message=f"O endereço URL é: \n "
              f"Endereço: {url} \n "
              f"Pronto para salvar?")

    if opcao_escolhida:
      qr = qrcode.QRCode(version=1, box_size=10, border=5)
      qr.add_data(url)
      qr.make(fit=True)
      img = qr.make_image(fill_color='black', back_color='white')
      img.save('qrExport.png')
```

A lógica dessa função é a seguinte: - Primeiramente o código checa se o *input* do usuário (`website_entry`) tem ao menos um *caracter* para continuar. - Se o *input* contiver dados, o programa vai mandar uma mensagem ao usuário confirmando o *input* e pede confirmação se é para salvar ou não.

Confirmado que é para salvar, o programa vai continuar gerando um Código QR com tamanhos específicos (você pode fazer outros tamanhos ou até mesmo diversas opções), e em seguida irá salvar a imagem do QR code no seu HD (na mesma pasta do script `main.py`).

O nome do arquivo será `qrExport.png` .

Agora que temos a lógica pronta para a entrada de *input* e a geração de Códigos QR, só falta criarmos uma interface de usuário e está tudo pronto!

```
window = Tk()
window.title("Gerador de Código QR")
window.config(padx=10, pady=100)

# Labels
website_label = Label(text="URL:")
website_label.grid(row=2, column=0)

# Entries
website_entry = Entry(width=35)
website_entry.grid(row=2, column=1, columnspan=2)
website_entry.focus()
add_button = Button(text="Gerar QR Code", width=36, com-
        mand=gera_qr_code)
add_button.grid(row=4, column=1, columnspan=2)

window.mainloop()
```

Aqui nós criamos um programa com o título “Gerador de Código QR” e colocamos espaçamentos.

Em seguida, adicionamos o título e a janela de *input* (Classe `Entry`).

Note que a lógica aqui é a formatação em grade, onde temos fileiras e colunas.

Então o título do *input* e o campo do *input* vão estar na mesma fileira (`row=2`).

E decidimos alinhar o botão na mesma coluna que o *input* (`column=1`).

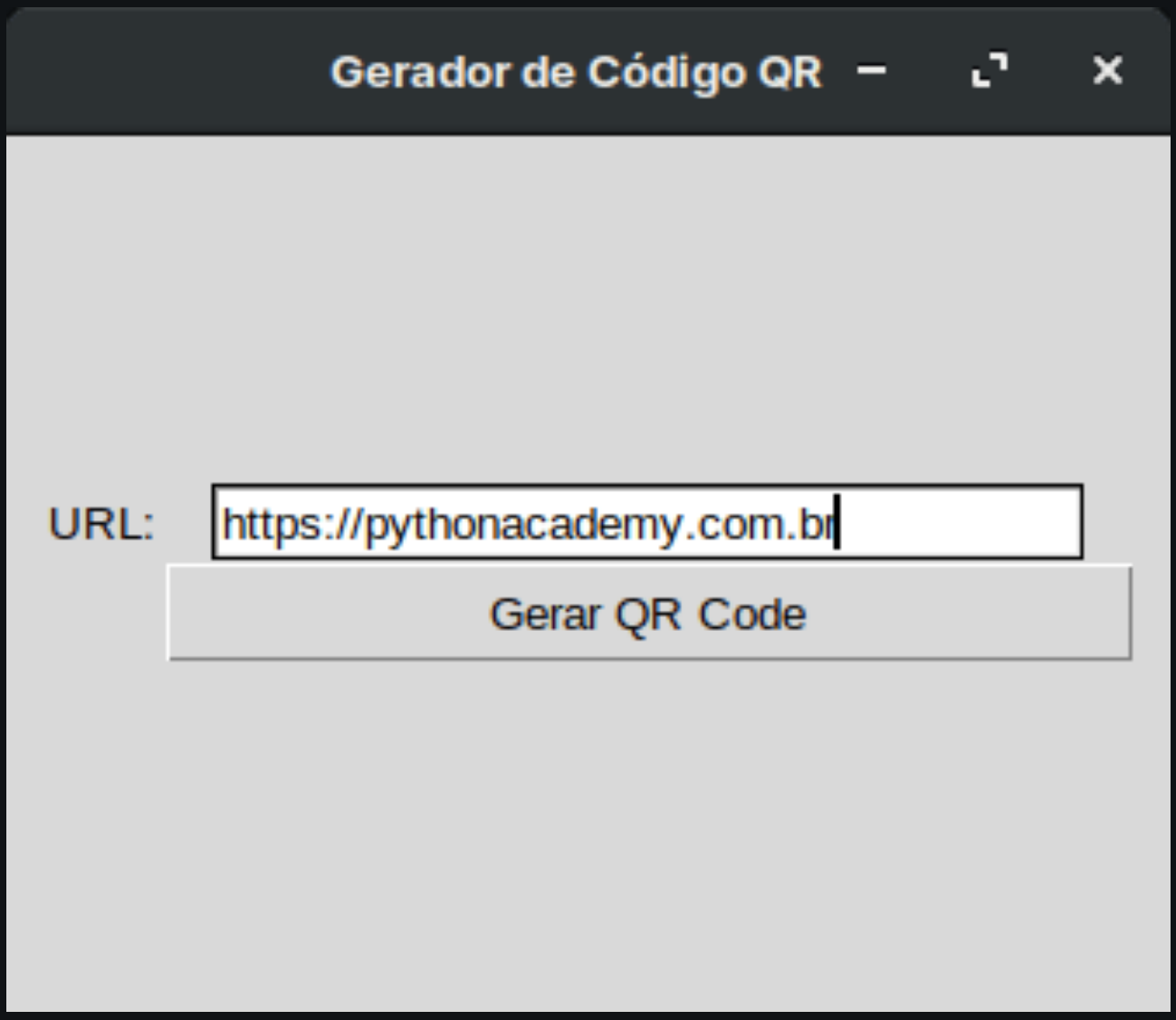
Portanto, o código completo é o seguinte:

```python
import qrcode
from tkinter import messagebox, Tk, Label, Entry, Button

def gera_qr_code():
    url = website_entry.get()

    if len(url) == 0:
        messagebox.showinfo(
            title="Erro!",
            message="Favor insira uma URL válida")
    else:
        opcao_escolhida = messagebox.askokcancel(
            title=url,
            message=f"O endereço URL é: \n "
                    f"Endereço: {url} \n "
                    f"Pronto para salvar?")

        if opcao_escolhida:
            qr = qrcode.QRCode(version=1, box_size=10, border=5)
            qr.add_data(url)
            qr.make(fit=True)
            img = qr.make_image(fill_color='black', back_color='white')
            img.save('qrExport.png')

if __name__ == '__main__':
    window = Tk()
    window.title("Gerador de Código QR")
    window.config(padx=10, pady=100)

    # Labels
    website_label = Label(text="URL:")
    website_label.grid(row=2, column=0)

    # Entries
    website_entry = Entry(width=35)
    website_entry.grid(row=2, column=1, columnspan=2)
    website_entry.focus()
    add_button = Button(text="Gerar QR Code", width=36, com-
        mand=gera_qr_code)
    add_button.grid(row=4, column=1, columnspan=2)
```

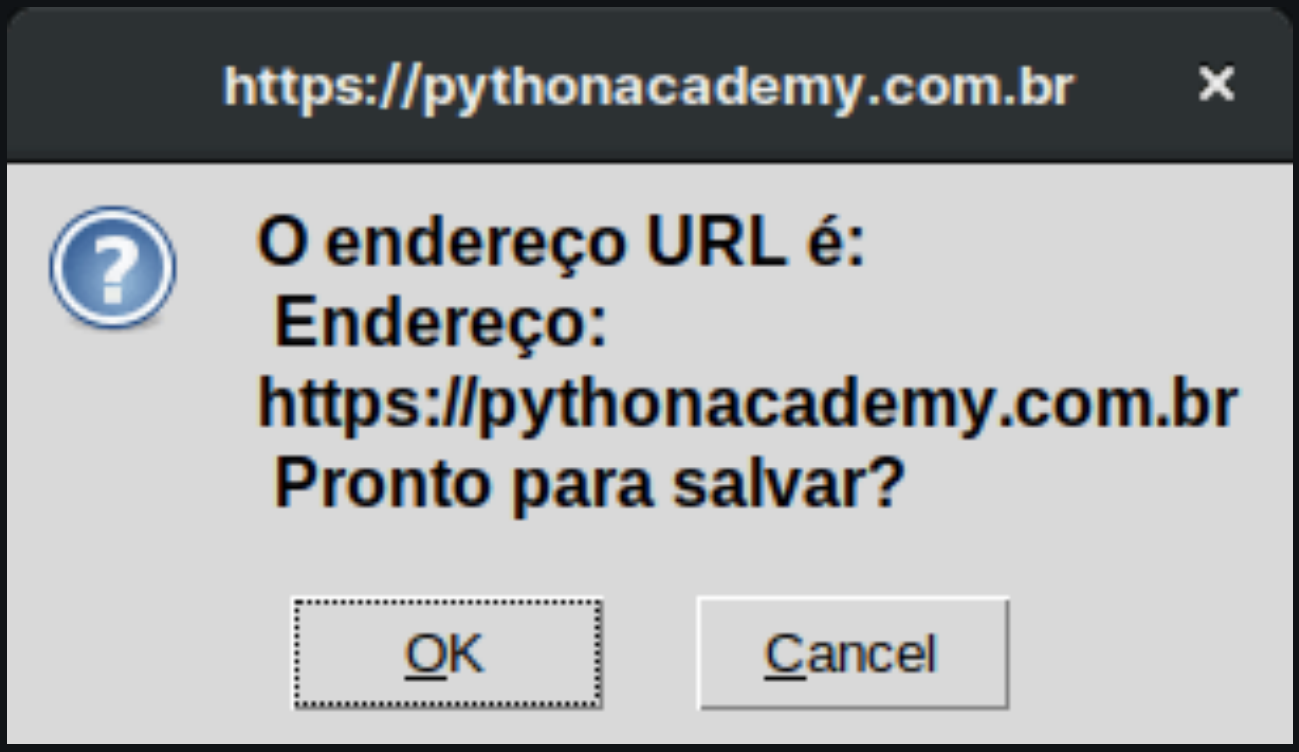
```
window.mainloop()
```

Resultado

Após colocar o código em execução, deverá aparecer a seguinte tela:

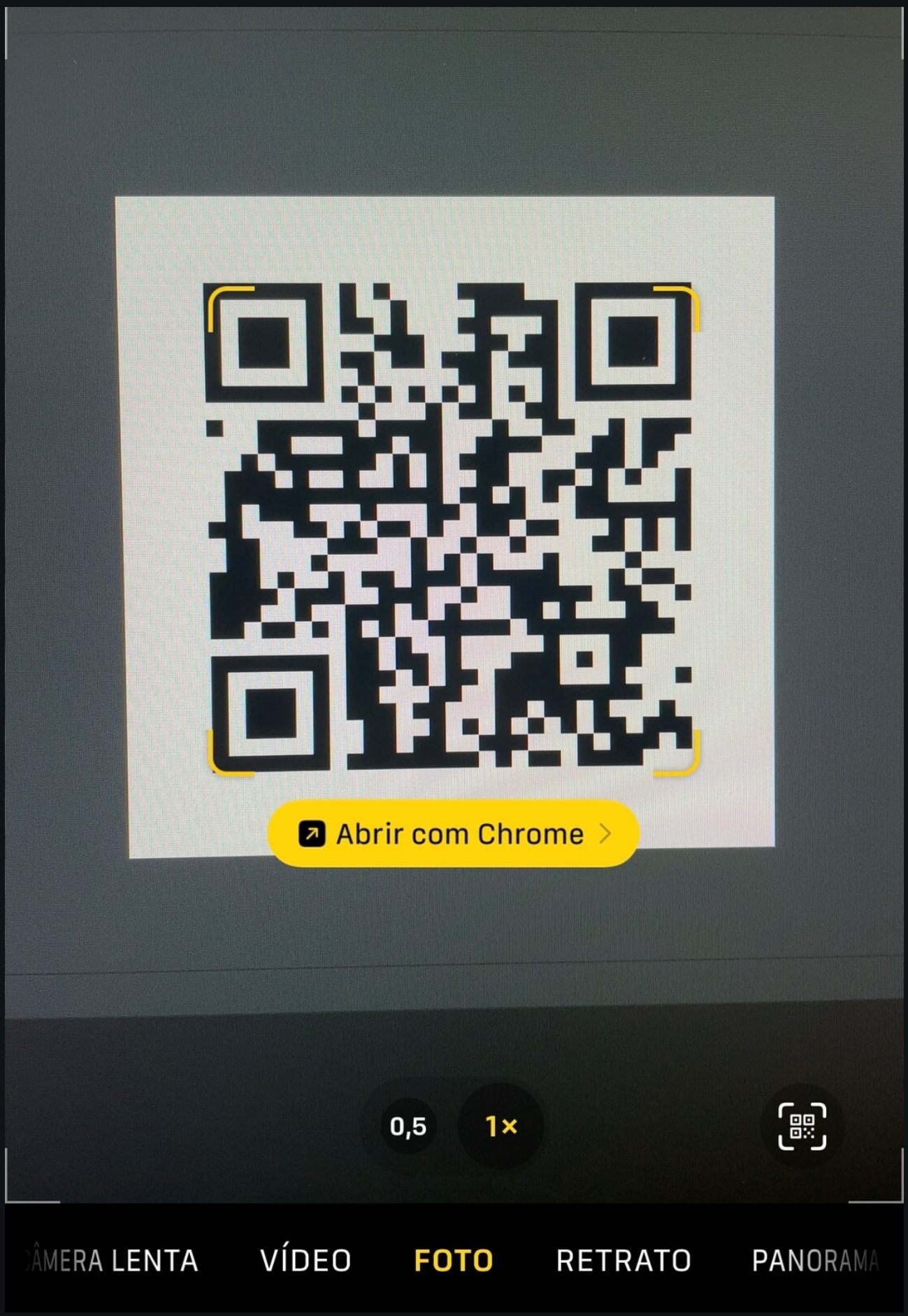


Adicionando uma URL e clicando em “Gerar QR Code”, a seguinte tela de confirmação deverá aparecer:



Ao clicar em “Ok”, o arquivo `qrExport.png` será criado na mesma pasta onde o script `main.py` foi executado e terá o seguinte conteúdo:

Agora basta apontar a câmera do seu celular pra ele e…


Abrir com Chrome
0,5
1×
CÂMERA LENTA
VÍDEO
FOTO
RETRATO
PANORAMA

Voilá! 😍 Você fez sua primeira interface gráfica com `tkinter`!

> 💡 *Estou desenvolvendo o **DevBook**, uma plataforma que usa IA para gerar ebooks técnicos profissionais. Não deixe de conferir clicando no botão abaixo!*

BÔNUS: Criando um executável para Windows

Agora que nosso programa está funcionando corretamente, vamos transformá-lo em um executável do Windows (`.exe`).

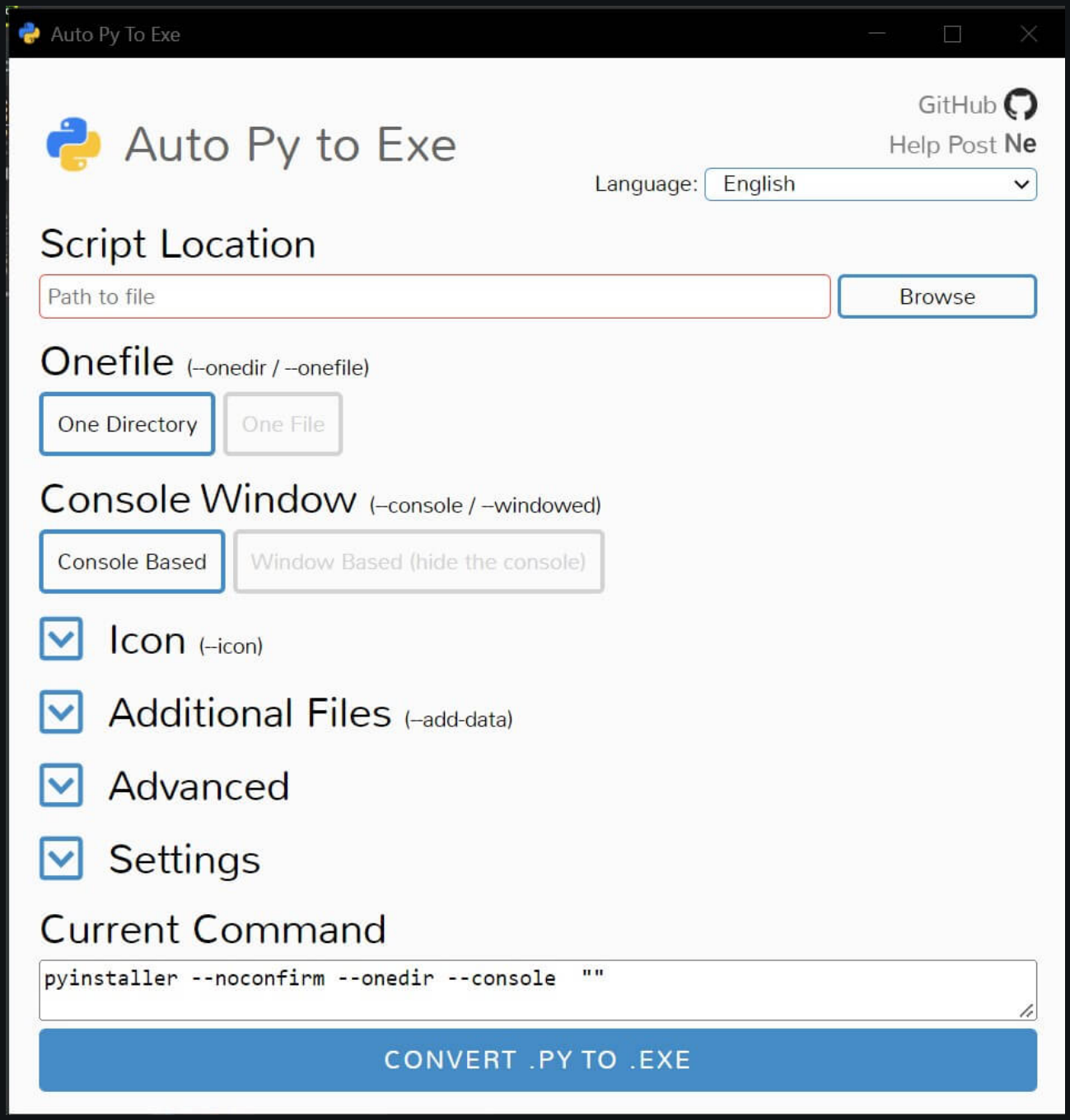
Para isto, nós vamos usar uma ferramenta chamada `auto-py-to-exe`.

Primeiro, instale-a com o comando `pip install auto-py-to-exe`.

Após instalada, vamos rodar o seguinte comando no terminal:

```
python -m auto_py_to_exe
```

Se tudo funcionar direitinho você verá a interface a seguir:



Nesta interface, você pode explorar diferentes possibilidades e configurações de arquivos a serem gerados, mas no nosso caso, nós optamos por copiar o endereço (“`path`”) do nosso arquivo `main.py` no campo ***Script Location*** e selecionamos os botões: ***One Directory***, ***Windows Based*** e selecionamos todas as opções como visto na imagem acima.

Em seguida, simplesmente clicamos em ***Convert .PY to .EXE*** e pronto!

Conclusão

Nesse artigo você deu seus primeiros passos na construção de Interfaces Gráficas utilizando o `tkinter`.

Criamos nossa primeira aplicação `.exe` a partir de um programa ***Python***!!!

Agora você pode criar aplicações mais rebuscadas e completas 😉

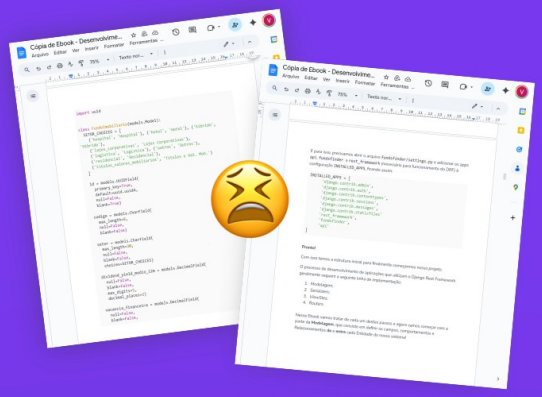
Um forte abraço do Yuri Falcão e do Programcoffee.net!

Não se esqueça de conferir!

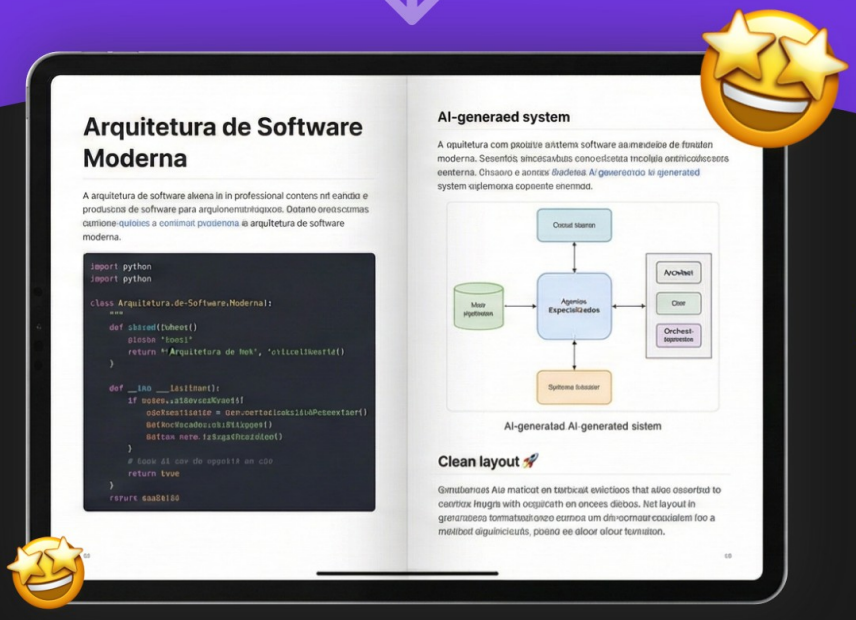
DevBook

Crie Ebooks técnicos em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código**!



Chega de formatar código no Google Docs



Syntax Highlight

Infográficos feitos por IA

Adicione Banners Promocionais

Deixe que nossa IA faça o trabalho pesado

Edite em Markdown em Tempo Real

TESTE AGORA

PRIMEIRO CAPÍTULO 100% GRÁTIS