

## Criação Aplicação/Programa executável com Python

### 1. Orientações Gerais

Utilizar uma interface de front-end em Python e empacotar essa aplicação como um programa executável. Existem várias bibliotecas e ferramentas em Python que permitem criar interfaces gráficas de usuário (GUI), tais como:

- a) **Tkinter**: É a biblioteca de GUI padrão para Python e vem pré-instalada com o Python. É uma escolha comum para projetos simples devido à sua simplicidade.
- b) **PyQt ou PySide**: São conjuntos de bindings para o framework Qt que são mais avançados e oferecem uma ampla gama de widgets e ferramentas para criar interfaces profissionais.
- c) **Kivy**: É uma biblioteca de código aberto que é usada para desenvolver aplicativos multi-touch. Ela é capaz de rodar em diferentes plataformas, incluindo Windows, macOS, Linux, Android e iOS.
- d) **wxPython**: É um conjunto de bindings para o toolkit wxWidgets que fornece uma série de ferramentas para criar interfaces gráficas nativas em várias plataformas.

Após criar a interface de front-end usando uma dessas bibliotecas, é possível empacotar o aplicativo Python como um executável usando ferramentas como:

- a) **PyInstaller**: É uma ferramenta bem conhecida que pode converter programas Python em executáveis standalone para Windows, macOS e Linux. Ela empacota tudo o que é necessário para executar o aplicativo Python, incluindo a biblioteca de GUI, em um único arquivo executável.
- b) **cx\_Freeze**: É semelhante ao PyInstaller e pode criar executáveis a partir de scripts Python para várias plataformas.
- c) **py2exe**: É uma ferramenta para Windows que converte scripts Python em executáveis do Windows.
- d) **Py2app**: É uma ferramenta para macOS que converte scripts Python em aplicativos macOS standalone.

Estas ferramentas geralmente criam um diretório contendo o executável junto com todas as bibliotecas e arquivos de suporte necessários. Isso torna mais fácil distribuir o seu aplicativo para usuários finais, pois eles podem executá-lo sem a necessidade de instalar Python ou quaisquer dependências adicionais.

## 2. Criação Programa executável

### ➤ Converter arquivo notebook (de “.ipynb” para “.py”)

Primeiro, é necessário converter o notebook Jupyter em um script Python padrão e, em seguida, seguir o processo de conversão para um executável.

#### ○ Exportar arquivo do notebook como um script Python:

- i. No Jupyter Notebook: Vá para "File" > "Download as" > "Python (.py)".
- ii. No JupyterLab: Clique com o botão direito no notebook na barra lateral e selecione "Export Notebook As..." > "Executable Script".

### ➤ Criar o Executável:

- Após criação do arquivo “.py”, a próxima etapa é a criação de um executável usando o PyInstaller;
- Instalar as bibliotecas e pacotes do PyInstaller. Navegue até o diretório onde está armazenado o script Python (arquivo “.py”), abra um terminal ou prompt de comando e execute os comandos abaixo:
  - i. Instalação das bibliotecas: `pip install pyinstaller`
- Execução do PyInstaller para gerar o executável: `pyinstaller --onefile arquivo_script.py`
  - Substituir *arquivo\_script.py* pelo nome do script Python;
  - A opção `--onefile` cria um executável único em vez de um diretório com vários arquivos.
  - O PyInstaller gerará uma pasta chamada “*dist*” no diretório do onde está armazenado o script, e o executável estará dentro dela.