Aqui está o script completo para copiar as pastas necessárias para o diretório dist:

|  |
| --- |
| import shutil  import os  def prepare\_dist():  # Define os diretórios de origem e destino  source\_dirs = ['\_pages', '\_css', '\_image', '\_scripts', '\_models\_print']  dest\_dir = 'dist'    # Verifica se o diretório dist existe  if not os.path.exists(dest\_dir):  os.makedirs(dest\_dir)  # Copia os diretórios para o dist  for dir\_name in source\_dirs:  src = os.path.join(dir\_name)  dst = os.path.join(dest\_dir, dir\_name)  if os.path.exists(src):  shutil.copytree(src, dst, dirs\_exist\_ok=True)  else:  print(f"Diretório de origem não encontrado: {src}")  if \_\_name\_\_ == "\_\_main\_\_":  prepare\_dist()  print("Preparação do diretório dist concluída!") |

**Passos para Usar o dist.py**

1. **Crie o Script dist.py**: Salve o código acima em um arquivo chamado dist.py no diretório raiz do seu projeto.
2. **Execute o Script**: Execute o script dist.py para copiar todas as pastas necessárias para o diretório dist.

|  |
| --- |
| python dist.py |

**Compile o Executável com PyInstaller**: Após preparar o diretório dist, você pode usar o PyInstaller para criar o executável. Certifique-se de incluir as pastas copiadas no diretório dist:

|  |
| --- |
| pyinstaller --onefile --add-data "\_pages;\_pages" --add-data "\_css;\_css" --add-data "\_image;\_image" --add-data "\_scripts;\_scripts" --add-data "\_models\_print;\_models\_print" main.py |

**Verifique a Estrutura do Diretório dist**: Após a execução do script dist.py, a estrutura do diretório dist deve estar semelhante a esta:

|  |
| --- |
| dist/  ├── main.exe  ├── \_pages/  ├── \_css/  ├── \_image/  ├── \_scripts/  └── \_models\_print/ |

**Distribua o Projeto**: Você pode distribuir a pasta dist ou criar um arquivo ZIP para facilitar a distribuição.

**Pacote ZIP**: Crie um pacote ZIP que inclui o executável e todas as pastas necessárias. Isso facilitará a distribuição para os usuários finais.

|  |
| --- |
| zip -r projeto\_sistema\_enfermagem.zip dist/ |

**Nota**

Certifique-se de que o PyInstaller está instalado e funcionando corretamente em seu ambiente. O comando --add-data no PyInstaller ajuda a incluir arquivos e pastas adicionais no executável, tornando-o mais portátil e completo.

Com isso, você terá um script dist.py que prepara seu projeto para distribuição e um executável que funciona em qualquer máquina com Windows 10 ou superior, sem necessidade de instalar o Python.