

Sequência de construção exercício “Sabor express”

1. Crie um arquivo **app.py** e digite: Configuração inicial e importações:

```
# Importação de bibliotecas necessárias
import os

# Lista de dicionários representando os restaurantes
restaurantes = [{'nome': 'Praça', 'categoria': 'Japonesa', 'ativo': False},
                 {'nome': 'Pizza Suprema', 'categoria': 'Pizza', 'ativo': True},
                 {'nome': 'Cantina', 'categoria': 'Italiano', 'ativo': False}]
```

Importando o módulo `os` para operações do sistema e inicializando uma lista de restaurantes para termos alguns dados para trabalhar.

2. Funções de exibição e utilitárias:

Agora, vamos implementar as funções que serão usadas para exibir informações e realizar tarefas utilitárias:

```
def exibir_nome_do_programa():
    print("""
    Sabor express
    """)

def exibir_opcoes():
    print('1. Cadastrar restaurante')
    print('2. Listar restaurante')
    print('3. Alternar estado do restaurante')
    print('4. Sair\n')

def finalizar_app():
    exibir_subtitulo('Finalizando o app\n')

def voltar_ao_menu_principal():
    input('\nDigite uma tecla para voltar ao menu principal')
    main()
```

```
def opcao_invalida():
    print('Opção inválida!\n')
    voltar_ao_menu_principal()

def exibir_subtitulo(texto):
    os.system('cls') # Limpa a tela (funciona apenas no Windows)
    linha = '*' * (len(texto))
    print(linha)
    print(texto)
    print(linha)
    print()
```

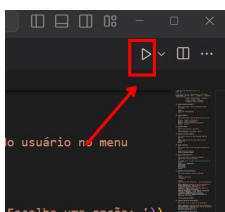
```
def escolher_opcao():
    """
    Função para processar a escolha do usuário no menu principal
    """
    try:
        opcao_escolhida = int(input('Escolha uma opção: '))

        if opcao_escolhida == 1:
            print('Opção 1 escolhida!\n')
            voltar_ao_menu_principal()
            # cadastrar_novo_restaurante()
        elif opcao_escolhida == 2:
            print('Opção 2 escolhida!\n')
            voltar_ao_menu_principal()
            # listar_restaurantes()
        elif opcao_escolhida == 3:
            print('Opção 3 escolhida!\n')
            voltar_ao_menu_principal()
            # alternar_estado_do_restaurante()
        elif opcao_escolhida == 4:
            finalizar_app()
        else:
            opcao_invalida()
    except:
        opcao_invalida()
```

```
def main():
    """
    Função principal que inicia o programa
    """
    os.system('cls') # Limpa a tela (funciona apenas no Windows)
    exibir_nome_do_programa()
    exibir_opcoes()
    escolher_opcao()

if __name__ == '__main__':
    main()
```

Testar rodando no terminal: **python app.py** ou clique:



3. Funções principais do programa:

Agora, vamos implementar as funções que realizam as operações principais do programa:

```

def cadastrar_novo_restaurante():
    """
    Função para cadastrar um novo restaurante

    Inputs:
    - Nome do restaurante
    - Categoria

    Outputs:
    - Adiciona um novo restaurante à lista de restaurantes
    """
    exibir_subtitulo('Cadastro de novos restaurantes\n')
    nome_do_restaurante = input('Digite o nome do restaurante que deseja cadastrar: ')
    categoria = input(f'Digite o nome da categoria do restaurante {nome_do_restaurante}: ')
    dados_do_restaurante = {'nome': nome_do_restaurante, 'categoria': categoria, 'ativo': False}
    restaurantes.append(dados_do_restaurante)
    print(f'O restaurante {nome_do_restaurante} foi cadastrado com sucesso!')

    voltar_ao_menu_principal()

```

```

def listar_restaurantes():
    """
    Função para listar todos os restaurantes cadastrados
    """
    exibir_subtitulo('Listando os restaurantes\n')

    print(f'{'nome_restaurante'.ljust(21)} | {'categoria'.ljust(20)} | Status')

    for restaurante in restaurantes:
        nome_restaurante = restaurante['nome']
        categoria = restaurante['categoria']
        ativo = 'ativado' if restaurante['ativo'] else 'desativado'
        print(f'-{nome_restaurante.ljust(20)} | {categoria.ljust(20)} | {ativo}')

    voltar_ao_menu_principal()

```

Testar rodando no terminal: **python app.py**. (Obs: Não esqueça de alterar na função: **def escolher_opcao()**)

```
def escolher_opcao():
    """
    Função para processar a escolha do usuário no menu principal
    """
    try:
        opcao_escolhida = int(input('Escolha uma opção: '))

        if opcao_escolhida == 1:
            # print('Opção 1 escolhida!\n')
            # voltar_ao_menu_principal()
            cadastrar_novo_restaurante()
        elif opcao_escolhida == 2:
            # print('Opção 2 escolhida!\n')
            # voltar_ao_menu_principal()
            listar_restaurantes()
        elif opcao_escolhida == 3:
            print('Opção 3 escolhida!\n')
            voltar_ao_menu_principal()
            # alternar_estado_do_restaurante()
        elif opcao_escolhida == 4:
            finalizar_app()
        else:
            opcao_invalida()
    except:
        opcao_invalida()
```

```
def alternar_estado_do_restaurante():
    """
    Função para ativar ou desativar um restaurante
    """
    exibir_subtitulo('Alternando estado do restaurante\n')
    nome_restaurante = input('Digite o nome do restaurante que deseja
    alterar o estado: ')
    restaurante_encontrado = False

    for restaurante in restaurantes:
        if nome_restaurante == restaurante['nome']:
            restaurante_encontrado = True
            restaurante['ativo'] = not restaurante['ativo'] # Inverte o
            estado (Ex. False para True)
            mensagem = f'O restaurante {nome_restaurante} foi ativado com
            sucesso!' if restaurante['ativo'] else f'O restaurante
            {nome_restaurante} foi desativado com sucesso!'
            print(mensagem)

    if not restaurante_encontrado:
        print('O restaurante não foi encontrado!')

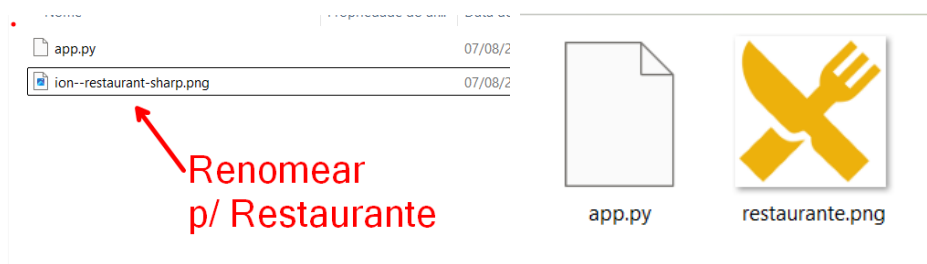
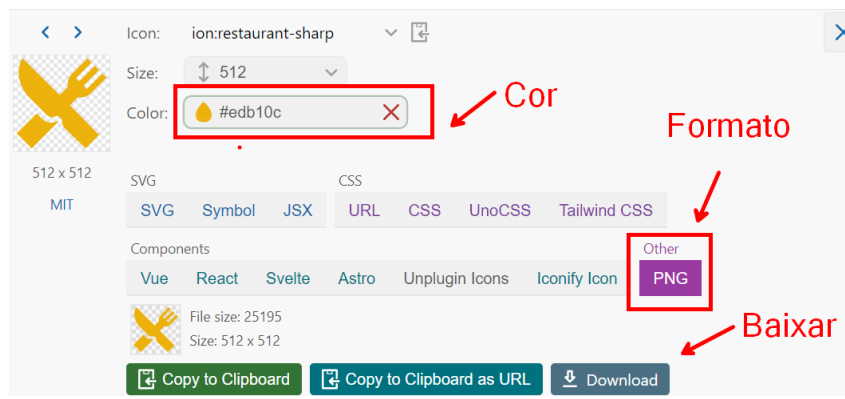
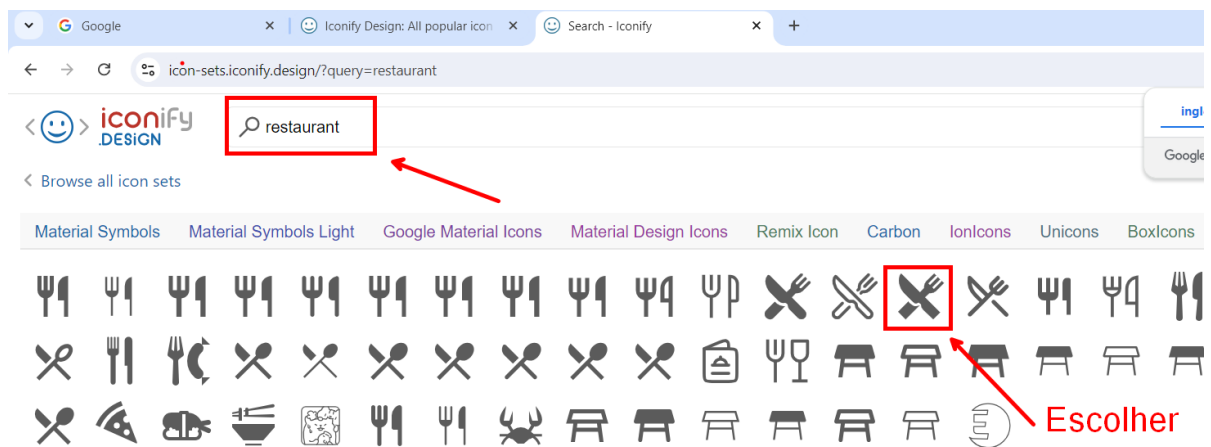
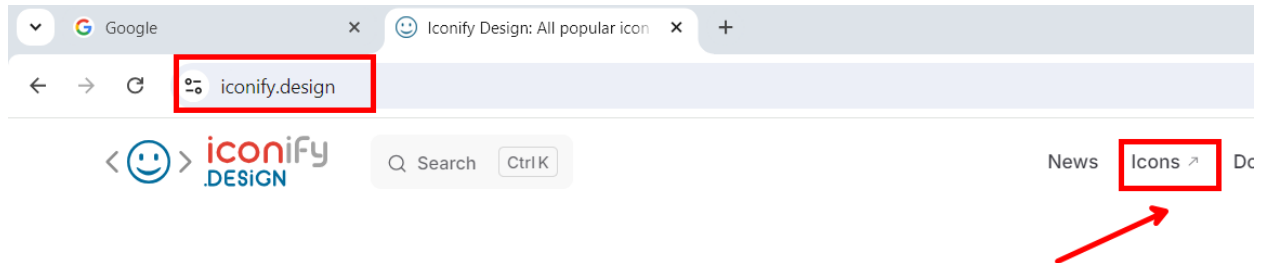
    voltar_ao_menu_principal()
```

4. Testar rodando no terminal: `python app.py`

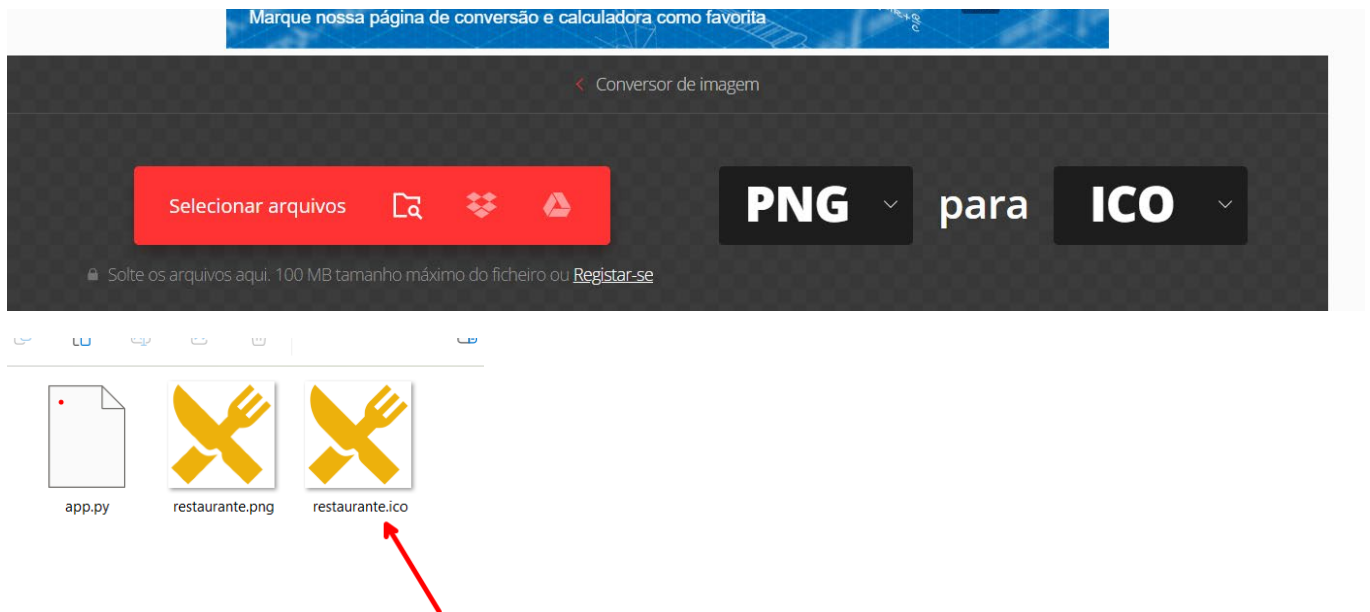
Obs: Passos abaixo **apenas para conhecimento**, o exe será gerado pelo Professor, pois exige senha adm.

Pegar ícones para projeto (*.png)

<https://iconify.design/>



Para ícone inicial converter um arquivo em .ico



Gerar executável de programa Python (Aguarde Professor)

Acessar a página: <https://pypi.org/project/auto-py-to-exe/>

Rodar (Instalar pip): `python -m ensurepip`

Obs: Rodar cmd como administrador

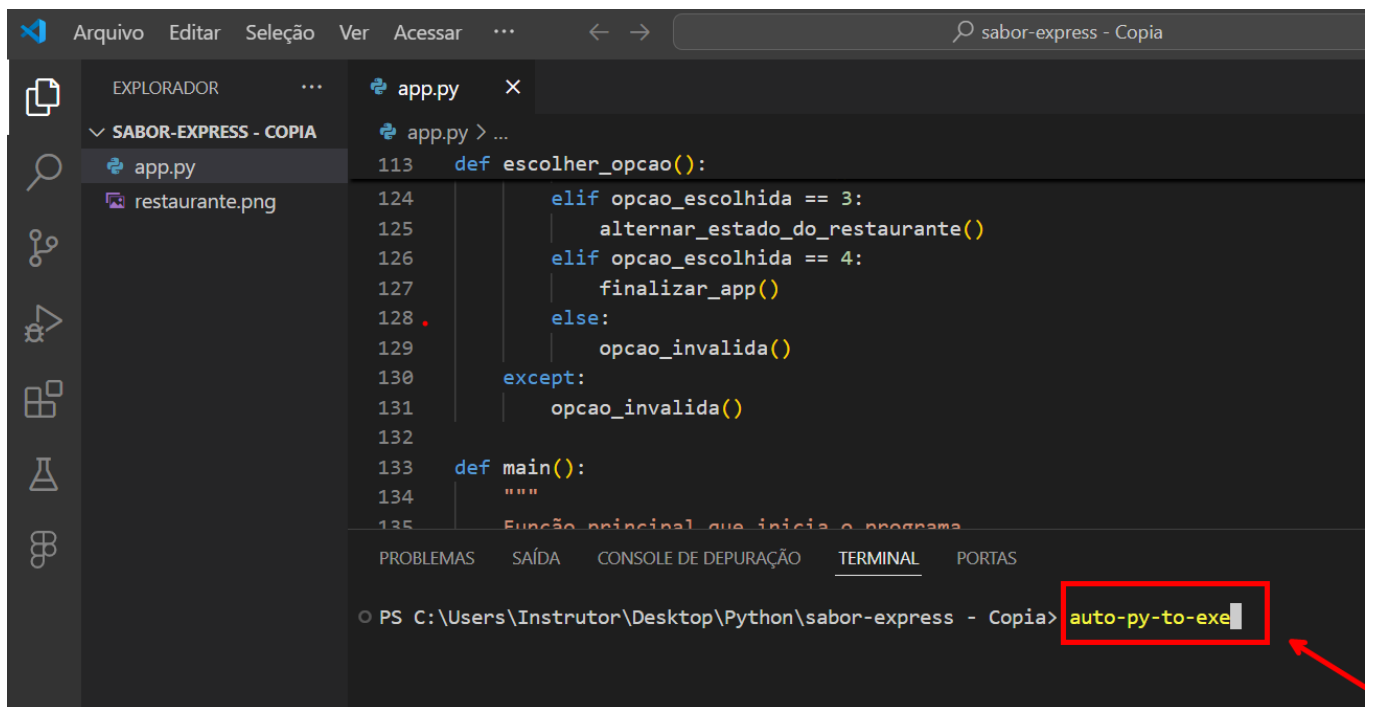
Rodar (Atualizar pip): `python -m pip install --upgrade pip`

Rodar (Atualizar pyinstaller): `python -m pip install pyinstaller --upgrade`

Rodar (Gerador exe): `pip install auto-py-to-exe`

Rodar (Interface): `auto-py-to-exe`

Com o programa instalado abrir o código do projeto e o terminal no VSCode e **Rodar (Interface):** `auto-py-to-exe`



Auto Py To Exe

GitHub Help Post

Language: English

Script Location

C:/Users/Instructor/Desktop/Python/sabor-express - Copia/app.py Browse

Onefile (--onedir / --onefile)

One Directory One File

Console Window (--console / --windowed)

Console Based Window Based (hide the console)

Icon (--icon)

C:/Users/Instructor/Desktop/Python/sabor-express - Copia/restaurante.ico Browse

Additional Files (--add-data)

Add Files	Add Folder	Add Blank
C:/Users/Instructor/Desktop/Python/sabor-express -	.	[-]
C:/Users/Instructor/Desktop/Python/sabor-express -	.	[-]

Be careful when using additional files with onefile mode; read this and update your code to work with PyInstaller.

If you want to put files in the root directory, put a period (.) in the destination.

☒ Advanced

☒ Settings

Current Command

```
pyinstaller --noconfirm --onefile --console --icon  
"C:/Users/Instructor/Desktop/Python/sabor-express - Copia/restaurante.ico" --add-data
```

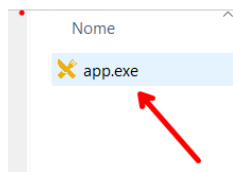
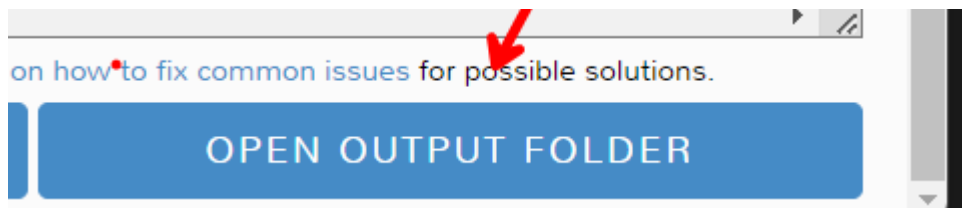
Se for terminal.
Se Windows outro.

Current Command

```
pyinstaller --noconfirm --onefile --console --icon  
"C:/Users/Instructor/Desktop/Python/sabor-express - Copia/restaurante.ico" --add-data
```

CONVERT .PY TO .EXE

Após



Rodar

```
C:\Users\Instrutor\Desktop\P: x + v

Sabor express

1. Cadastrar restaurante
2. Listar restaurante
3. Alternar estado do restaurante
4. Sair

Escolha uma opção: |
```