

Arquivo: automacao_email.py

```
import pyautogui
import pyperclip
import time
from pathlib import Path
from PyQt6.QtWidgets import QMessageBox

# Define o caminho para a pasta onde as imagens da automação estão salvas
IMAGE_DIR = Path(__file__).parent.parent.parent / "assets" / "automation_images"

def find_and_click(image_name, description, confidence=0.9, timeout=10):
    """
    Procura por uma imagem na tela e clica nela.
    Retorna True se for bem-sucedido, False caso contrário.
    """
    print(f"Procurando por: {description} ('{image_name}')...")
    start_time = time.time()
    while time.time() - start_time < timeout:
        try:
            # Tenta encontrar a imagem na tela
            location = pyautogui.locateOnScreen(str(IMAGE_DIR) / image_name,
                                                confidence=confidence)
            if location:
                pyautogui.click(pyautogui.center(location))
                print(f"Sucesso: Clicou em '{description}'")
                time.sleep(1) # Pequena pausa para a UI reagir
                return True
        except pyautogui.ImageNotFoundException:
            time.sleep(0.5) # Espera um pouco antes de tentar novamente

    print(f"ERRO: Não foi possível encontrar '{description}' na tela após {timeout} segundos.")
    QMessageBox.critical(None, "Erro na Automação", f"Não foi possível encontrar o elemento '{description}' na tela.")
    return False

def paste_text(text):
    """Copia um texto para a área de transferência e o cola."""
    pyperclip.copy(text)
    pyautogui.hotkey('ctrl', 'v')
    print(f"Colou o texto: '{text[:30]}...'") # Mostra só os primeiros 30 caracteres
    time.sleep(0.5)

# --- FUNÇÃO PRINCIPAL DA AUTOMAÇÃO ---
def executar_automacao_email(destinatario, assunto, corpo_mensagem):
    """
    Executa uma sequência de passos de automação para enviar um e-mail.
    """
    print("\n--- INICIANDO AUTOMAÇÃO DE E-MAIL ---")
    # =====#
    #       COMO ADICIONAR NOVOS PASSOS (SEU GUIA)
```

```

# =====
# Para adicionar um novo passo, basta adicionar uma nova verificação 'if not ...'
# usando as funções 'find_and_click' ou 'paste_text'.
#
# Exemplo: Se você precisar clicar em um campo de "Assunto" antes de colar,
# você adicionaria a seguinte linha:
#
# if not find_and_click('campo_assunto.png', 'Campo de Assunto'):
#     return # Para a automação se o passo falhar
#
# E então o passo para colar o assunto:
# paste_text(assunto)
#
# =====

# --- PASSO 1: Clicar no botão para criar uma nova mensagem ---
# (Você precisa tirar um print do botão "Nova Mensagem" e salvar como 'nova_mensagem.png')
if not find_and_click('nova_mensagem.png', "Botão 'Nova Mensagem'"):
    return # Para a automação se o passo falhar

# --- PASSO 2: Colar o destinatário ---
# (Não precisa de imagem, apenas cola no campo que estiver focado)
paste_text(destinatario)
pyautogui.press('tab') # Pressiona Tab para pular para o próximo campo (geralmente Assunto)
time.sleep(0.5)

# --- PASSO 3: Colar o assunto ---
paste_text(assunto)
pyautogui.press('tab') # Pressiona Tab para pular para o corpo do e-mail
time.sleep(0.5)

# --- PASSO 4: Colar o corpo da mensagem ---
paste_text(corpo_mensagem)

# --- PASSO 5 (Opcional): Clicar em Enviar ---
# (Descomente as linhas abaixo quando estiver pronto para enviar de verdade)
# print("Procurando pelo botão Enviar...")
# if not find_and_click('botao_enviar.png', "Botão 'Enviar'"):
#     return

print("---- AUTOMAÇÃO FINALIZADA (Modo de Teste) ----")
QMessageBox.information(None, "Automação Concluída",
                        "O processo de automação foi finalizado.\n"
                        "O e-mail foi preenchido, mas não foi enviado (modo de teste)..")

```