Para que seu script Python de automação da interface gráfica (UI automation) seja executado automaticamente ao iniciar o computador ou sair da hibernação, a ferramenta mais robusta e recomendada no Windows é o **Agendador de Tarefas (Task Scheduler)**.

O Agendador de Tarefas permite que você defina gatilhos complexos, incluindo "Ao fazer logon" (para quando você inicia o PC e faz login) e "Na retomada (do sono/hibernação)" (para quando o PC sai do modo de suspensão ou hibernação).

Você não pode simplesmente colocar o arquivo .py em uma pasta de inicialização para que ele funcione para "retomar da hibernação", e scripts Python não são executáveis diretamente pelo sistema operacional da mesma forma que um .exe. Você precisará especificar o interpretador Python para executá-lo.

Vamos ao passo a passo usando o Agendador de Tarefas:

Método Recomendado: Agendador de Tarefas (Task Scheduler)

Este método oferece o maior controle e é o mais confiável para scripts de automação de UI.

**Pré-requisitos:**

1. **Caminho Completo para o seu Script:** Digamos que seu script se chame automacao\_chrome.py e esteja em C:\Scripts\Automacao\automacao\_chrome.py.
2. **Caminho Completo para o python.exe:** Você precisa saber onde o interpretador Python está instalado. Geralmente é algo como C:\Users\SEU\_USUARIO\AppData\Local\Programs\Python\Python3x\python.exe (substitua SEU\_USUARIO e Python3x pela sua versão). Para descobrir, você pode digitar where python no Prompt de Comando.
3. **Certifique-se de que todas as dependências (keyboard, pyautogui, pygetwindow, Pillow, opencv-python) estão instaladas no ambiente Python que você vai usar.**

**Passos:**

1. **Abra o Agendador de Tarefas:**
   * Pressione Win + R, digite taskschd.msc e pressione Enter. OU
   * Pesquise por "Agendador de Tarefas" no menu Iniciar do Windows.
2. **Crie uma Nova Tarefa:**
   * No painel da direita, clique em Criar Tarefa... (Não "Criar Tarefa Básica...", pois queremos mais controle sobre os gatilhos).
3. **Geral:**
   * **Nome:** Dê um nome significativo (ex: Abrir Chrome com Perfil).
   * **Descrição:** Opcional, mas útil (ex: Abre o Chrome no perfil 'Pessoa 1' e navega para o Terra.com.br).
   * **Opções de segurança:**
     + Marque Executar somente quando o usuário estiver logado (Essencial para automação de UI, pois a tarefa precisa interagir com a interface gráfica do seu usuário logado).
     + Marque Executar com privilégios mais altos (Recomendado para scripts de automação, caso precise de acesso a certas áreas ou para evitar problemas de permissão).
     + Configure o usuário que irá executar a tarefa (geralmente o seu próprio usuário).
4. **Disparadores (Triggers):**
   * Clique na aba Disparadores e em Novo....
   * **Para inicialização do sistema após login:**
     + **Iniciar a tarefa:** Selecione Ao fazer logon.
     + **Configurações:** Qualquer usuário ou Usuário específico (escolha o seu usuário).
     + Marque Atrasar tarefa por: e defina um tempo (ex: 1 minuto). Isso é crucial para dar tempo ao sistema para carregar completamente antes que o script tente interagir com a UI.
   * **Para saída da hibernação/suspensão:**
     + Clique em Novo... novamente para adicionar outro disparador.
     + **Iniciar a tarefa:** Selecione Na retomada (do sono/hibernação).
     + Marque Atrasar tarefa por: (ex: 30 segundos a 1 minuto). Isso é importante para dar tempo ao sistema para reativar todos os componentes da UI.
   * Clique OK para cada disparador.
5. **Ações (Actions):**
   * Clique na aba Ações e em Novo....
   * **Ação:** Iniciar um programa.
   * **Programa/script:** Insira o **caminho completo para o seu python.exe**.
     + Ex: C:\Users\SEINT-RS\AppData\Local\Programs\Python\Python39\python.exe (ou onde quer que seu Python esteja).
   * **Adicionar argumentos (opcional):** Insira o **caminho completo para o seu script Python**.
     + Ex: C:\Scripts\Automacao\automacao\_chrome.py
   * **Iniciar em (opcional):** Este é o diretório de trabalho. Coloque o **diretório onde seu script está localizado**.
     + Ex: C:\Scripts\Automacao\ (Isso é importante se seu script usa arquivos relativos, como a imagem pessoal1.png se ela estivesse no mesmo diretório do script, ou para logs).
   * Clique OK.
6. **Condições (Conditions):**
   * Revise a aba Condições. Para automação de UI, geralmente você não quer que a tarefa execute se estiver em bateria (se for um laptop), ou se não houver rede (se o script depender da internet).
   * **Energia:** Desmarque Iniciar a tarefa somente se o computador estiver na energia AC (se você quiser que ele rode na bateria).
   * **Rede:** Marque Iniciar somente se a seguinte conexão de rede estiver disponível e escolha Qualquer conexão se o script depender da internet.
7. **Configurações (Settings):**
   * Revise a aba Configurações.
   * **Permitir que a tarefa seja executada sob demanda**: Marque.
   * **Parar a tarefa se ela for executada por mais de**: Defina um tempo limite razoável (ex: 30 minutos) caso seu script trave.
   * **Se a tarefa falhar, reiniciar a cada**: Opcional, mas útil para robustez (ex: 1 minuto, tentar até 3 vezes).
   * **Se a tarefa já estiver em execução, aplicar a seguinte regra:**: Não iniciar uma nova instância (geralmente é o mais seguro para UI automation).
8. **Finalize:**
   * Clique OK para salvar a tarefa. O Agendador de Tarefas pode pedir a sua senha de usuário para confirmar as configurações de segurança.

Método Alternativo (Mais Simples, Menos Robusto para Retomada): Pasta Inicializar (Startup Folder)

Este método é mais fácil, mas só funciona quando você **faz login** e não é ideal para "retomar da hibernação/suspensão" diretamente.

1. **Crie um arquivo .bat (Batch Script):**
   * Abra o Bloco de Notas.
   * Digite o seguinte (substitua pelos seus caminhos reais):
   * @echo off
   * C:\Users\SEINT-RS\AppData\Local\Programs\Python\Python39\python.exe C:\Scripts\Automacao\automacao\_chrome.py

exit

content\_copydownload

Use code [with caution](https://support.google.com/legal/answer/13505487" \t "_blank).Batch

(Adicionar @echo off e exit é boa prática para scripts batch).

* + Salve este arquivo como, por exemplo, rodar\_automacao.bat em um local fácil de acessar (ex: sua pasta de Documentos).

1. **Abra a Pasta "Inicializar" (Startup Folder):**
   * Pressione Win + R, digite shell:startup e pressione Enter. Isso abrirá a pasta "Inicializar" para o seu usuário.
2. **Mova o arquivo .bat:**
   * Mova ou crie um atalho do arquivo rodar\_automacao.bat para dentro desta pasta "Inicializar".

Agora, toda vez que você fizer login no Windows, este script .bat será executado, que por sua vez, executará seu script Python.

**Considerações Importantes para Automação de UI em Inicialização/Retomada:**

* **Tempo de Atraso (time.sleep()):** Os time.sleep() que você já tem no seu script são CRUCIAIS. Na inicialização do sistema, muitos processos e serviços estão carregando em segundo plano, e as janelas podem não estar prontas para interagir imediatamente. Você pode precisar aumentar esses tempos de espera.
* **Ambiente Gráfico:** O script DEVE ser executado na sua sessão de usuário (não como um serviço de sistema ou em uma sessão remota invisível) para que ele possa interagir com a UI. O Agendador de Tarefas com a opção "Executar somente quando o usuário estiver logado" garante isso.
* **Resolução da Tela:** Se a resolução da sua tela mudar entre a captura da imagem (pessoal1.png) e a execução do script (ex: conectar/desconectar monitor externo), o pyautogui.locateCenterOnScreen pode falhar.
* **Pop-ups Inesperados:** Se houver pop-ups (atualizações, notificações) que apareçam antes do Chrome, eles podem interferir na digitação e nos cliques.
* **Logging de Erros:** Como o script rodará em segundo plano, você não verá a saída do console. É uma excelente prática adicionar um try...except em torno de todo o seu script e escrever quaisquer erros e mensagens de depuração para um **arquivo de log** (.txt) no disco, para que você possa verificar o que aconteceu se ele falhar.

Escolha o método do Agendador de Tarefas para a solução mais robusta, especialmente para lidar com a retomada da hibernação.

**shell:startup não abriu no windows 11.**

É muito estranho que shell:startup não tenha funcionado no Windows 11, pois este comando é universal no Windows (do 7 ao 11) para abrir a pasta de inicialização do usuário atual.

Vamos tentar algumas alternativas e verificar o que pode estar acontecendo:

Opção 1: Abrir o Explorador de Arquivos e Navegar Diretamente

Esta é a forma mais manual, mas infalível, para chegar à pasta:

1. Abra o **Explorador de Arquivos** (pode ser clicando no ícone da pasta na barra de tarefas ou pressionando Win + E).
2. Na barra de endereços (onde você vê algo como "Este PC > Documentos"), digite ou cole o seguinte caminho e pressione Enter:  
   %APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup
   * %APPDATA% é uma variável de ambiente que aponta para C:\Users\<SeuNomeDeUsuário>\AppData\Roaming. Usá-la garante que você vá para a pasta correta do seu usuário.

**Observação:** A pasta AppData é uma pasta oculta por padrão no Windows. Se você tentar navegar manualmente sem usar %APPDATA%, pode ser que você não a veja. Para ver pastas ocultas: no Explorador de Arquivos, vá em **"Exibir"** na barra de menus superior, depois em **"Mostrar"** e marque a opção **"Itens ocultos"**.

Opção 2: Tentar Novamente shell:startup com Cautela

Pode ter sido um erro de digitação ou algo temporário.

1. Pressione **Win + R** para abrir a caixa de diálogo "Executar".
2. Digite **exatamente** shell:startup (tudo em minúsculas, sem espaços extras) e pressione **Enter**.

Opção 3: Para a Pasta de Inicialização de Todos os Usuários (Menos Comum)

Se você quisesse que o script rodasse para *qualquer* usuário que faça login, existe outra pasta de inicialização:

1. Pressione **Win + R**.
2. Digite shell:common startup e pressione Enter.
   * Este comando aponta para C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp.
   * Para esta pasta, é necessário ter privilégios de administrador para mover ou copiar arquivos.

Depois de Abrir a Pasta "Inicializar"

Independentemente de como você abriu a pasta, os próximos passos são os mesmos:

1. **Crie um arquivo .bat (Batch Script):**
   * Abra o **Bloco de Notas**.
   * Cole o conteúdo abaixo, substituindo os caminhos pelos seus caminhos reais:
   * @echo off
   * REM Navega para o diretório do seu script Python (importante se ele usa arquivos relativos)
   * cd C:\Users\SEINT-RS\Downloads\
   * REM Executa o script Python usando o interpretador python.exe
   * REM Substitua C:\Path\To\Your\Python\python.exe pelo caminho real do seu python.exe
   * "C:\Users\SEINT-RS\AppData\Local\Programs\Python\Python39\python.exe" "C:\Users\SEINT-RS\Downloads\seu\_script\_automacao.py"
   * REM Opcional: Pausa a execução do batch para ver erros no console (remover em produção)
   * REM pause

exit

content\_copydownload

Use code [with caution](https://support.google.com/legal/answer/13505487" \t "_blank).Batch

* + - **Caminho do python.exe:** Para descobrir, abra o Prompt de Comando e digite where python e pressione Enter. Ele mostrará o caminho completo.
    - **Caminho do seu script:** Use o caminho completo do seu arquivo .py.
    - Use aspas duplas "" nos caminhos se houver espaços neles.
    - A linha cd C:\Users\SEINT-RS\Downloads\ é importante se o seu script pyautogui usa uma imagem como pessoal1.png que está no mesmo diretório que o script, sem o caminho completo. Se IMAGE\_PATH\_PESSOA1 no seu script Python já usa o caminho completo (r'C:\Users\SEINT-RS\Downloads\pessoal1.png'), então o cd não é estritamente necessário, mas pode ajudar a manter o contexto.
  + Salve o arquivo com um nome relevante (ex: iniciar\_automacao\_chrome.bat) dentro da pasta "Inicializar" que você abriu.

**Importante:**

* **Pausas (time.sleep)**: Os time.sleep() dentro do seu script Python são absolutamente essenciais quando ele é executado na inicialização. O Windows leva um tempo para carregar todos os serviços e a interface gráfica. Você pode precisar aumentar esses tempos para garantir que as janelas estejam prontas para a interação.
* **Visibilidade:** O script .bat abrirá uma janela de console preta e a manterá aberta enquanto o script Python estiver em execução. Se você quiser que o script rode em segundo plano **sem mostrar a janela do console**, você precisaria criar um script .vbs (Visual Basic Script) para lançar o Python em vez de um .bat. Se precisar, posso fornecer o código para isso.
* **Depuração:** Se o script não funcionar na inicialização, adicione um bom tratamento de erros e log de saída no seu script Python (escrevendo para um arquivo de texto) para entender o que deu errado, já que você não verá o console.