

Simplifica Python



APOSTILA

AUTOMAÇÃO COMPLETA EM PYTHON



Alexandre Sauer

@prof.alesauer

• Especialista em Python
• Mestre em Sistemas da Informação

O QUE VAMOS APRENDER?

Na aula Automação com Python, vamos desenvolver um projeto de automação. Imagine que você tenha usuários, produtos ou qualquer outro tipo de informação que precise cadastrar em um sistema.

Agora, imagine que você tenha milhares de informações para inserir. Fazer tudo isso manualmente seria bastante trabalhoso, não é mesmo?

Você perderia dias de trabalho realizando esse processo. Porém, nesta aula, eu vou te mostrar como automatizar esse tipo de tarefa usando Python.

Você vai aprender a controlar o mouse e o teclado para que executem tarefas por você. Sim, é exatamente isso que você leu: Python fará isso por você.

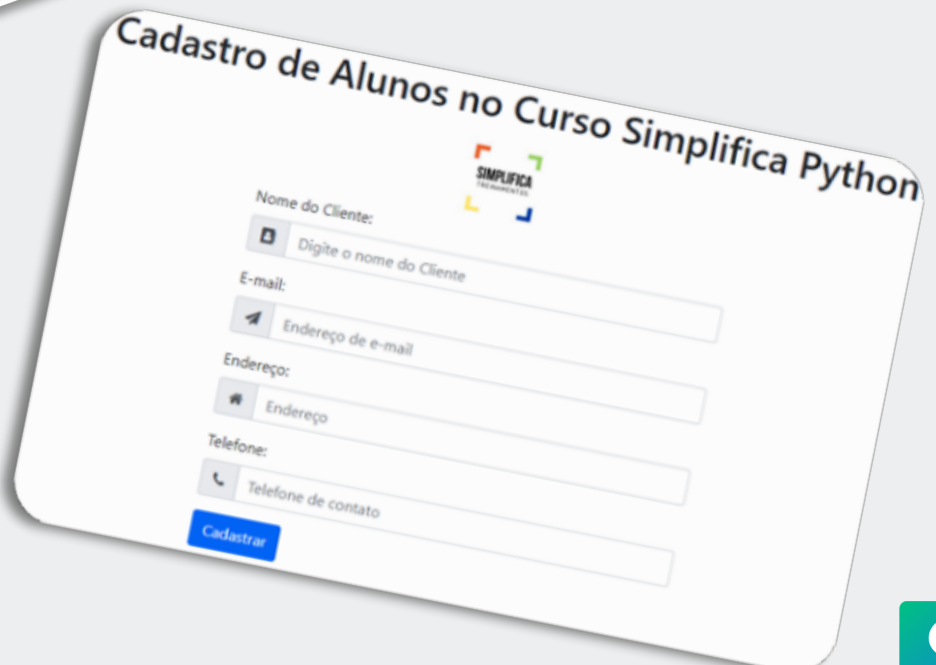


O QUE VAMOS APRENDER?

O objetivo do projeto é que você consiga acessar um site (criado especialmente para essa aula), fazer login e cadastrar todos os produtos de forma automática. Vamos utilizar uma base de dados com esses produtos para que você possa simular o cadastro de sua lista de produtos, preços ou qualquer outra informação que precise registrar.



The image shows a login form titled "Sistema dos Alunos da Simplifica". At the top is the logo for "SIMPLIFICA TREINAMENTOS", which consists of four colored squares (red, green, yellow, blue) arranged in a larger square shape. Below the logo, the title "Sistema dos Alunos da Simplifica" is written in a large, bold font. Underneath the title, there are two input fields: "Login:" and "Senha:". To the right of the "Login:" field is a button labeled "Digite seu login". To the right of the "Senha:" field is a button labeled "Digite sua senha". Below these two buttons is a blue button labeled "Entrar" and a grey button labeled "Limpar Dados".



The image shows a registration form titled "Cadastro de Alunos no Curso Simplifica Python". At the top is the logo for "SIMPLIFICA TREINAMENTOS", which consists of four colored squares (red, green, yellow, blue) arranged in a larger square shape. Below the logo, the title "Cadastro de Alunos no Curso Simplifica Python" is written in a large, bold font. Underneath the title, there are four input fields: "Nome do Cliente:", "E-mail:", "Endereço:", and "Telefone:". To the right of the "Nome do Cliente:" field is a button labeled "Digite o nome do Cliente". To the right of the "E-mail:" field is a button labeled "Endereço de e-mail". To the right of the "Endereço:" field is a button labeled "Endereço". To the right of the "Telefone:" field is a button labeled "Telefone de contato". Below these four buttons is a blue button labeled "Cadastrar".

ENTENDENDO A BASE DE DADOS

Os dados que utilizaremos são as informações contidas no arquivo "**alunos.csv**". Este arquivo contém diversos produtos que serão cadastrados no sistema automaticamente.

Quando você precisar realizar um procedimento similar, provavelmente terá uma lista de itens para cadastrar, certo? Então, aqui temos exatamente essa lista (base de dados) com as informações necessárias para o cadastro.

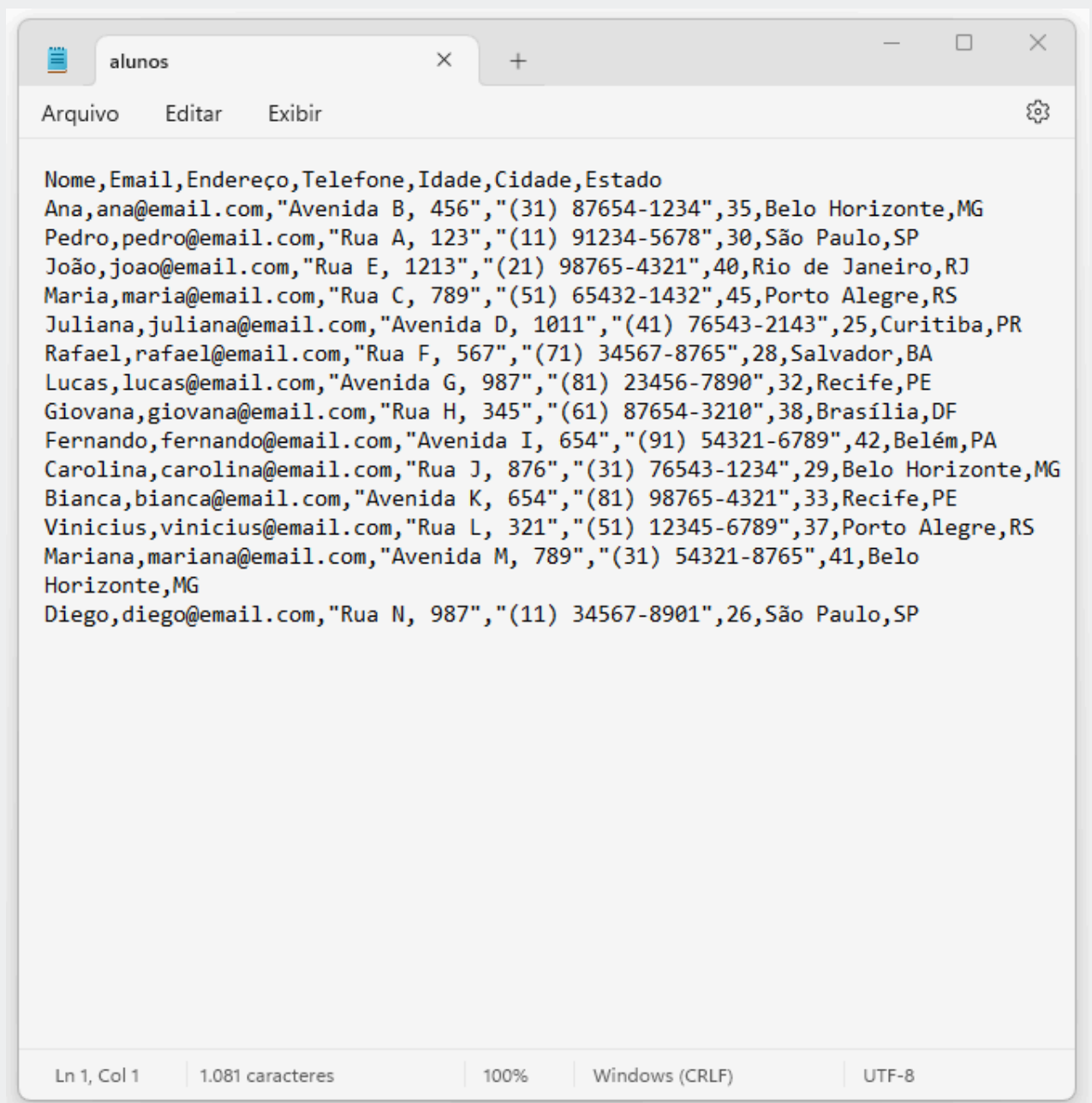
As informações incluem o nome do cliente, e-mail, endereço e telefone. No entanto, o primeiro ponto é que os dados estão no formato CSV, o que pode dificultar um pouco a visualização.

Além disso, temos centenas de linhas de dados, o que significa que você teria que preencher quatro informações no sistema para cada linha da base de dados. Para evitar esse trabalho manual, vamos criar automações com {Python}.

Dessa forma, o trabalho será muito mais rápido, automático e sem chances de inserir informações incorretas.



ENTENDENDO A BASE DE DADOS



The screenshot shows a text editor window with the title 'alunos'. The menu bar includes 'Arquivo', 'Editar', and 'Exibir'. The text content is a CSV file with 15 rows of student data. The status bar at the bottom indicates 'Ln 1, Col 1', '1.081 caracteres', '100%', 'Windows (CRLF)', and 'UTF-8'.

```
Nome,Email,Endereço,Telefone,Idade,Cidade,Estado
Ana,ana@email.com,"Avenida B, 456","(31) 87654-1234",35,Belo Horizonte,MG
Pedro,pedro@email.com,"Rua A, 123","(11) 91234-5678",30,São Paulo,SP
João,joao@email.com,"Rua E, 1213","(21) 98765-4321",40,Rio de Janeiro,RJ
Maria,maria@email.com,"Rua C, 789","(51) 65432-1432",45,Porto Alegre,RS
Juliana,juliana@email.com,"Avenida D, 1011","(41) 76543-2143",25,Curitiba,PR
Rafael,rafael@email.com,"Rua F, 567","(71) 34567-8765",28,Salvador,BA
Lucas,lucas@email.com,"Avenida G, 987","(81) 23456-7890",32,Recife,PE
Giovana,giovana@email.com,"Rua H, 345","(61) 87654-3210",38,Brasília,DF
Fernando,fernando@email.com,"Avenida I, 654","(91) 54321-6789",42,Belém,PA
Carolina,carolina@email.com,"Rua J, 876","(31) 76543-1234",29,Belo Horizonte,MG
Bianca,bianca@email.com,"Avenida K, 654","(81) 98765-4321",33,Recife,PE
Vinicius,vinicius@email.com,"Rua L, 321","(51) 12345-6789",37,Porto Alegre,RS
Mariana,mariana@email.com,"Avenida M, 789","(31) 54321-8765",41,Belo Horizonte,MG
Diego,diego@email.com,"Rua N, 987","(11) 34567-8901",26,São Paulo,SP
```

ENTENDENDO A SOLUÇÃO FINAL

A solução deste projeto será a nossa automação, ou seja, ao final, todos os alunos da nossa base de dados estarão cadastrados no sistema de forma automática, como se você estivesse utilizando o computador para realizar o cadastro.

Ao executar o programa, as seguintes ações serão realizadas para completar o nosso cadastro:

- Abrir o navegador
- Acessar o site do sistema com login e senha
- Inserir todas as informações do aluno
- Enviar as informações para o sistema
- Repetir o processo até cadastrar todos os alunos

ENTENDENDO A SOLUÇÃO FINAL




Sistema dos Alunos da Simplifica

Login:

Senha:



Cadastro de Alunos no Curso Simplifica Python



Nome do Cliente:

E-mail:

Endereço:

Telefone:



Dados dos Alunos Cadastrados

Nome	E-mail	Endereço	Telefone
ALEXANDRE SAUER PAIS LEMES	alesauer@gmail.com	RUA Zenite, 670 apto 101	31995041815



Simplifica
TREINAMENTOS



PARTE 02

IMPORTAR E VISUALIZAR OS DADOS

PROF. SAUER

IMPORTANDO BASE DE DADOS

Vamos utilizar a biblioteca **pandas** para importar a base de dados e visualizá-la, para que você entenda como os dados estão organizados.

A base de dados é o arquivo "**alunos.csv**", então é importante ter o arquivo **no mesmo diretório** onde você criou seu script em Python.

Assim, você pode simplesmente usar o comando `pd.read_csv` e colocar o nome do arquivo (caso contrário, será necessário fornecer o caminho completo do arquivo).

Em seguida, basta usar o comando `print` para visualizar os dados.

IMPORTANDO BASE DE DADOS

	Nome	Email	Endereço	...	Idade	Cidade	Estado
0	Ana	ana@email.com	Avenida B, 456	...	35	Belo Horizonte	MG
1	Pedro	pedro@email.com	Rua A, 123	...	30	São Paulo	SP
2	João	joao@email.com	Rua E, 1213	...	40	Rio de Janeiro	RJ
3	Maria	maria@email.com	Rua C, 789	...	45	Porto Alegre	RS
4	Juliana	juliana@email.com	Avenida D, 1011	...	25	Curitiba	PR
5	Rafael	rafael@email.com	Rua F, 567	...	28	Salvador	BA
6	Lucas	lucas@email.com	Avenida G, 987	...	32	Recife	PE
7	Giovana	giovana@email.com	Rua H, 345	...	38	Brasília	DF
8	Fernando	fernando@email.com	Avenida I, 654	...	42	Belém	PA
9	Carolina	carolina@email.com	Rua J, 876	...	29	Belo Horizonte	MG
10	Bianca	bianca@email.com	Avenida K, 654	...	33	Recife	PE
11	Vinicius	vinicius@email.com	Rua L, 321	...	37	Porto Alegre	RS
12	Mariana	mariana@email.com	Avenida M, 789	...	41	Belo Horizonte	MG
13	Diego	diego@email.com	Rua N, 987	...	26	São Paulo	SP

```
1 import pandas as pd
2 import pyautogui
3 import time
4
5 # Importar a base de alunos pra cadastrar
6 tabela = pd.read_csv('alunos.csv')
7 print(tabela)
```



Simplifica
TREINAMENTOS



PARTE 03

BIBLIOTECA DE AUTOMAÇÃO

PYAUTOGUI

PROF. SAUER

ENTENDENDO A BIBLIOTECA

Você deve ter percebido que importamos a biblioteca `pyautogui`. Essa biblioteca permitirá que você controle o mouse e o teclado para realizar automações no seu computador usando Python.

Para instalar a biblioteca, basta digitar `pip install pyautogui` no seu terminal.

Nesta aula, utilizaremos basicamente três comandos da biblioteca `Pyautogui`:

- **`pyautogui.press`** – Funciona como pressionar uma tecla do teclado.
- **`pyautogui.write`** – Funciona para digitar com o teclado.
- **`pyautogui.click`** – Funciona para clicar com o mouse

ENTENDENDO A BIBLIOTECA

Em relação ao **pyautogui.click** você vai notar que precisamos passar uma **posição x e y** para que ele saiba onde tem que fazer o clique do mouse.

IMPORTANTE: Essa posição que nós vamos passar é em relação ao SEU monitor, porque ele leva em consideração o tamanho do monitor.

Dentro dos arquivos da aula nós temos um arquivo chamado **pegar_posição.py**, que já tem um código para pegar a posição do seu mouse. Com isso você vai ter a posição exata em relação ao seu computador para executar sua automação.

Ao executar o código, ele demora 5 segundos para pegar a posição para que você tenha tempo suficiente de colocar o mouse onde precisa clicar.

É muito importante que você pegue essas posições, pois se utilizar o padrão do nosso exemplo, é bem provável que dê errado e acabe clicando onde não deve.

ENTENDENDO A BIBLIOTECA

```
1  import time
2  import pyautogui
3
4  time.sleep(5)
5  print(pyautogui.position())
6  pyautogui.scroll(200)
```

Este é o script "pegar_posição.py", que permite capturar a posição atual do seu mouse, ajudando você a saber exatamente onde clicar durante a automação.

Este passo é crucial para garantir que você clique no local correto. Sempre que utilizar o pyautogui para clicar, é necessário realizar este procedimento.

ENTENDENDO A BIBLIOTECA

É importante, pois a posição é relativa ao monitor, e o que estamos usando pode ser diferente do seu, resultando em cliques em locais errados.

Por isso, sempre obtenha essas informações antes de executar seu código para evitar cliques indesejados. Este trecho de código não faz parte da automação principal; ele é um script independente para você capturar a posição do mouse nos pontos em que será necessário clicar, garantindo a precisão durante a execução do seu código.



Simplifica
TREINAMENTOS



PARTE 04

AUTOMAÇÃO COM O

PYAUTOGUI

PROF. SAUER

ABRINDO O NAVEGADOR

Agora que você já conhece o projeto e tem todos os arquivos e bibliotecas necessários, podemos começar nossa automação.

```
pyautogui.PAUSE = 0.5
# abrir o navegador (chrome)
pyautogui.press("win")
pyautogui.write("chrome")
pyautogui.press("enter")
# entrar no link
pyautogui.write("http://www.sauer.pro.br/python/automacao/index.html")
pyautogui.press("enter")
time.sleep(5)
```

Vamos iniciar utilizando o comando **pyautogui.PAUSE**, que define o tempo de espera entre os comandos do Pyautogui. Isso significa que haverá uma pausa entre cada comando no seu código, o que é essencial para garantir que tudo seja executado corretamente.

Em seguida, temos o comando **pyautogui.press**, que é usado para pressionar uma tecla do teclado.

ABRINDO O NAVEGADOR

Aqui usamos a tecla "win", que é a tecla do Windows no seu teclado, para abrir o menu iniciar.

Em seguida, utilizamos o comando **pyautogui.write**, que serve para digitar. Vamos escrever "**Chrome**", que é o navegador que vamos usar, mas você pode alterar para o navegador de sua preferência.

Lembre-se de que tudo aqui simula o que você faria manualmente. Portanto, você pode realizar o processo manualmente para verificar os passos necessários.

Como estamos reproduzindo suas ações, faça o procedimento passo a passo para entender cada etapa.

Nesta parte inicial, vamos abrir o navegador e digitar o link do site onde faremos o login para cadastrar os alunos. Após esse processo, utilizamos `time.sleep` para esperar o carregamento do site antes de prosseguir para a próxima etapa.

FAZENDO O LOGIN NO SISTEMA

Com o site ou sistema aberto, podemos iniciar o processo de login. Neste exemplo, iremos usar o usuário admin e senha admin para efetuar o login no sistema.

IMPORTANTE: Aqui vamos usar o `pyautogui.click`, então é essencial que você já tenha as coordenadas exatas de onde clicar para digitar o login e também para clicar no botão de entrar no sistema.

```
# Passo 2: Fazer login
# selecionar o campo de login
pyautogui.click(x=1213, y=441)
# escrever o usuário na posição capturada
pyautogui.write("admin")
pyautogui.press("tab") # passando pro próximo campo
#esrever a senha na posição capturada
pyautogui.write("SimplificaPython")
pyautogui.click(x=888, y=543) # clique no botao de login
```

FAZENDO O LOGIN NO SISTEMA

Outro comando que utilizamos aqui é o `pyautogui.press("tab")`, que simula a tecla Tab para passar para o próximo campo.

Isso é interessante, pois evita que você precise usar o clique do mouse várias vezes, eliminando a necessidade de capturar a posição de cada campo.

Isso será muito útil ao cadastrarmos os produtos, já que há vários campos, e você não vai querer pegar a posição de cada um deles, certo? Utilizando a tecla Tab para navegar entre os campos, o processo se torna mais eficiente e menos trabalhoso.



The image shows a login form for 'Simplifica Treinamentos'. At the top, there is a logo consisting of four colored L-shaped icons (orange, green, yellow, blue) arranged in a square, with the text 'SIMPLIFICA TREINAMENTOS' in the center. Below the logo, the title 'Sistema dos Alunos da Simplifica' is displayed in a large, bold, dark blue font. Underneath the title, there are two input fields. The first is labeled 'Login:' and contains the placeholder text 'Digite seu login'. The second is labeled 'Senha:' and contains the placeholder text 'Digite sua senha'. At the bottom of the form, there are two buttons: a blue button labeled 'Entrar' and a grey button labeled 'Limpar Dados'.

CADASTRANDO OS ALUNOS

```
# Passo 4: Cadastrar um produto
for linha in tabela.index:
    # clicar no campo de cliente/aluno
    pyautogui.click(x=1138, y=387)
    # pegar da tabela o valor do campo que a gente quer preencher
    nome = tabela.loc[linha, "Nome"]
    # preencher o campo
    pyautogui.write(str(nome))
    # passar para o proximo campo
    pyautogui.press("tab")
    # preencher o campo Email
    pyautogui.write(str(tabela.loc[linha, "Email"]))
    pyautogui.press("tab")
    pyautogui.write(str(tabela.loc[linha, "Endereco"]))
    pyautogui.press("tab")
    pyautogui.write(str(tabela.loc[linha, "Telefone"]))
    pyautogui.press("tab")
    pyautogui.press("enter") # cadastra o produto (botao enviar)
    # dar scroll de tudo pra cima
    pyautogui.scroll(5000)
# Passo 5: Repetir o processo de cadastro até o fim
```

Assim, poderemos iterar por todas as informações na tabela e inserir cada dado específico no sistema.

Para o campo do código do produto, consultamos a coluna "código". Para o campo da marca do produto, buscamos na coluna "marca", e seguimos esse padrão para os demais campos.

A estratégia é aplicar esse método a todas as informações do produto, garantindo que cada detalhe seja corretamente cadastrado no sistema, até que o registro esteja completo.

CADASTRANDO OS ALUNOS

A estrutura de repetição for, da maneira que está escrita, pode ser interpretada basicamente assim:

"Para cada linha/número no índice da tabela, faça o seguinte".

O índice é simplesmente o número da linha na tabela. Portanto, o código percorrerá linha por linha, executando tudo que estiver dentro dessa estrutura de repetição.

Primeiro, usamos o comando `pyautogui.click` para clicar no primeiro campo onde preencheremos a primeira informação.

Em seguida, temos um comando um pouco diferente.

- `pyautogui.write` – Comando para digitar.
- `str` – Converte um valor para string (texto).
- `.loc` – Comando para buscar uma informação, neste caso, dentro da nossa tabela. Vamos buscar uma informação na linha x (dependendo de quantas vezes a repetição ocorreu) na coluna "codigo".

Portanto, o código busca a informação correta na linha atual e na coluna especificada, transformando-a em texto e digitando-a no campo apropriado do sistema.

CADASTRANDO OS ALUNOS

```
# Passo 4: Cadastrar um produto
for linha in tabela.index:

    # clicar no campo de cliente/aluno
    pyautogui.click(x=1138, y=387)
    # pegar da tabela o valor do campo que a gente quer preencher
    nome = tabela.loc[linha, "Nome"]
    # preencher o campo
    pyautogui.write(str(nome))
    # passar para o proximo campo
    pyautogui.press("tab")
    # preencher o campo Email
    pyautogui.write(str(tabela.loc[linha, "Email"]))
    pyautogui.press("tab")
    pyautogui.write(str(tabela.loc[linha, "Endereco"]))
    pyautogui.press("tab")
    pyautogui.write(str(tabela.loc[linha, "Telefone"]))
    pyautogui.press("tab")
    pyautogui.press("enter") # cadastra o produto (botao enviar)
    # dar scroll de tudo pra cima
    pyautogui.scroll(5000)
# Passo 5: Repetir o processo de cadastro até o fim
```

Agora vamos realizar o registro das informações no sistema.

Iniciaremos com a estrutura de repetição for, que percorrerá todas as linhas da base de dados, registrando as informações de cada produto.

Em cada iteração, passaremos por cada linha da tabela, preenchendo todas as informações no sistema e clicando em enviar.

CADASTRANDO OS ALUNOS

OBS: O comando `pyautogui.scroll` pode ser usado para rolar a página com a bolinha do mouse. No entanto, é possível que esse comando não seja necessário antes de clicar.

É recomendável fazer um teste manual para verificar o que acontece após o início do cadastro dos produtos.



Simplifica
TREINAMENTOS



PARTE 05

CONCLUSÃO

PROF. SAUER

CONCLUSÃO

A ideia deste projeto foi mostrar como você pode automatizar suas tarefas repetitivas utilizando Python. Essas são tarefas que você realiza no dia a dia, seja em casa ou no trabalho.

Já pensou no tempo que isso pode economizar? Imagine ter que cadastrar cerca de 300 produtos, cada um com 6 ou 7 informações diferentes. Isso tomaria um tempo enorme e poderia resultar em erros ao digitar ou colar uma informação no campo errado. Com a automação, você verifica tudo de antemão e o processo é repetido corretamente para todas as informações.

Com isso, você ganha muito tempo, pode focar em outras atividades e reduz drasticamente a chance de erros durante o cadastro.

Desafio você a reproduzir o que fizemos na aula e aplicar isso em um projeto seu ou no seu trabalho para automatizar suas tarefas. Isso vai aumentar sua produtividade, permitindo que a automação rode enquanto você realiza outras atividades ou até sai para o almoço.

CONCLUSÃO

Imagine isso: você sai para almoçar e deixa sua automação rodando. Quando voltar, tudo estará pronto, enquanto os outros ainda estão fazendo todo o processo manualmente!

Espero que tenha gostado da aula e percebido a utilidade e importância da automação de tarefas com Python. Além disso, utilizando a biblioteca pyautogui, tudo fica muito mais fácil, pois ela simplesmente replica os passos que você faria manualmente!

