

# Resumo da Aula: Automação de Cadastro de Alunos com PyAutoGUI

Nesta aula, desenvolvemos um script para automatizar o preenchimento de um cadastro de alunos em uma aplicação gráfica criada com Tkinter. O script lê nomes e e-mails de um arquivo de texto (`alunos.txt`) e usa o PyAutoGUI para preencher os campos da aplicação automaticamente, clicando no botão "Add" e gerando screenshots para auditoria.

## Arquivo de Entrada (`alunos.txt`)

Criamos um arquivo `alunos.txt` com dados de alunos, onde cada linha contém nome e e-mail separados por vírgula:

```
Rodrigo,rodrigo@exemplo.com
Fulano,fulano@email.com
Abc,abc@email.com
Sicrano,sicrano@email.com
José,jose@email.com
```

Este arquivo simula uma lista de presença que um professor poderia criar manualmente.

## Leitura do Arquivo

O script lê o arquivo `alunos.txt` e separa cada linha em nome e e-mail usando `line.split(',')`:

```
with open('alunos.txt', 'r', encoding='utf-8') as file:
    for line in file:
        aluno, email = line.strip().split(',')
```

O método `strip()` remove quebras de linha, e `split(',')` divide a linha em uma lista com nome e e-mail.

## Automação do Preenchimento

O script usa o PyAutoGUI para interagir com a aplicação gráfica. Antes de executar, usamos `mouseinfo` para obter as coordenadas dos campos e do botão:

```
import pyautogui
from time import sleep

pyautogui.PAUSE = 1
pyautogui.FAILSAFE = True

with open('alunos.txt', 'r', encoding='utf-8') as file:
    for line in file:
        aluno, email = line.strip().split(',')

        # Clica e preenche o campo Name (1417, 714)
        pyautogui.click(1417, 714, duration=1)
        sleep(1)
        pyautogui.write(aluno)
```

```
# Clica e preenche o campo Email (1423, 770)
pyautogui.click(1423, 770, duration=1)
sleep(1)
pyautogui.write(email)

# Clica no botão Add (1438, 797)
pyautogui.click(1438, 797, duration=0.5)

# Tira screenshot para auditoria
pyautogui.screenshot(f'aluno_{aluno}.png')
sleep(1)
```

O script: 1. Clica no campo "Name", digita o nome do aluno. 2. Clica no campo "Email", digita o e-mail. 3. Clica no botão "Add" para inserir os dados na tabela. 4. Tira um screenshot nomeado com o nome do aluno (ex.: `aluno_Rodrigo.png`).

## Pontos-Chave

- **Leitura de Arquivo:** O arquivo `alunos.txt` fornece os dados para automação.
- **Coordenadas:** Use `mouseinfo` para encontrar as posições exatas dos campos e botão.
- **Automação:** PyAutoGUI simula cliques e digitação para preencher a aplicação gráfica.
- **Auditoria:** Screenshots são gerados para cada aluno, permitindo verificar o preenchimento.
- **Estabilidade:** Mantenha a janela da aplicação na mesma posição para evitar erros nas coordenadas.

## Dicas para Alunos

- Use `pyautogui.FAILSAFE = True` para interromper o script movendo o mouse ao canto superior esquerdo.
- Teste as coordenadas com `mouseinfo` antes de rodar o script.
- Adicione pausas (`sleep`) para garantir que a automação siga o ritmo da aplicação.
- Verifique se o arquivo `alunos.txt` está no mesmo diretório do script.
- Para projetos reais, considere conectar a aplicação a um banco de dados para persistência dos dados.