

# Aula 14 - Automação de tarefas

---

Como falado no início do curso, Python é uma das melhores linguagens para criar programas com tarefas automatizadas. Tarefas automatizadas são programas que executam uma série de ações no seu computador como se fosse o próprio usuário utilizando a máquina. Entretanto, é o próprio computador sozinho trabalhando, sem a ajuda de ninguém. Legal né?! É como se o computador fizesse o que você precisa fazer no seu lugar. Na aula de hoje iremos aprender como fazer.

## Pyautogui

---

A biblioteca **pyautogui** é um pacote externo a ser instalado no Python da sua máquina que serve exatamente para criar instruções de automação de tarefas. Dito isso, siga as instruções abaixo para criarmos o nosso primeiro programa para automação de tarefas:

1. Abra o terminal do seu Sistema Operacional.
2. No terminal, navegue até a pasta do seu projeto, e crie o ambiente virtual **venv**.
3. Ainda no terminal, digite o comando `pip install pyautogui`.
4. Aguarde um bocadinho (um bocadinho mesmo, demora....)

O objetivo do programa aqui é apenas fazer uma pequena demonstração de como fazer a automação, mas saiba que esse é um dos ramos mais promissores do Python, e que já possui uma gigantesca procura no mercado de trabalho. Vamos criar um programa que simplesmente abre o navegador de internet no seu PC. Crie um arquivo na pasta do seu projeto e digite o código-fonte abaixo, e depois execute:

```
In [ ]: # importa biblioteca e aplica um alias
import pyautogui as pag

pag.PAUSE = 0.5 # define o tempo de espera para cada comando do pyautogui, para

pag.press('win') # abre o menu iniciar
pag.write('edge') # digita na barra de pesquisa do menu iniciar o nome do programa
pag.press('enter') # aperta enter após a digitação do nome do programa
```

Se tudo der certo, o seu computador irá, sozinho, abrir o menu iniciar, digitar na barra de pesquisa **edge** (nome do navegador da Microsoft), e abri-lo.

## Meus parabéns!!! Você criou sua primeira automação!

A biblioteca **pyautogui** é bem completa e cheia de vários comandos. É importante dedicar um tempo para estudar a documentação dela. Vamos aproveitar a oportunidade para criarmos um programa para automatizar o commit e o push de um repositório para o **GitHub**.

**Pré-requisitos:** o projeto já tem que ter um primeiro *commit* no repositório local e um *commit* no repositório remoto. Fora isso, o programa pode ser executado para qualquer projeto. Outro pré-requisito é que o projeto em questão tenha sido o último a ser aberto no **VSCode**.

Crie um arquivo **.py** com o código-fonte abaixo:

```
In [ ]: # importa bibliotecas
import pyautogui
import time

# tempo que cada comando demora para executar
pyautogui.PAUSE = 1

# instruções
pyautogui.press('win')
pyautogui.write('vscode')
pyautogui.press('enter')

# espera 10 segundos para abrir o vscode e continuar com os comandos
time.sleep(10)

# continua as instruções
pyautogui.hotkey('ctrl', 'shift', '')
pyautogui.write('git add .')
pyautogui.press('enter')
pyautogui.write('git commit -m "Repositório atualizado por automação."')

# espera 5 segundos para dar tempo de fazer o commit
time.sleep(5)

# continua as instruções
pyautogui.press('enter')
pyautogui.write('git push')
```

Após terminarmos, salve e execute o seu código, e veja no seu GitHub se o seu projeto subiu corretamente no repositório.