VI EIGEDIN
Encontro Internacional de Gestão, Desenvolvimento e Inovação
16 a 19 de novembro de 2022
Edição Online
ISSN: 2594-8083
NOVAS DINÂMICAS DA SOCIEDADE:
desafios e soluções

USO DO PYTHON NA AUTOMAÇÃO EMPRESARIAL

Claudiomar da Silva Fonseca Junior,
Universidade do Mato Grosso do Sul (UFMS),
claudiomar.junior@UFMS.br

Emanuel Adrian de Almeida Frazão da Fonseca,
Universidade do Mato Grosso do Sul (UFMS),
emanuel_adrian@UFMS.br

Gabriel Henrique Baracat Sarmento,
Universidade do Mato Grosso do Sul (UFMS),
gabriel.sarmento@UFMS.br

RESUMO

Atualmente vive-se em uma realidade em que as pessoas necessitam realizar tarefas e atividades de rotinas de seu trabalho em um curto espaço de tempo, registrar e enviar emails automáticos de documentos empresariais dos funcionários é algo frequente em uma organização. Essa demanda trouxe uma ferramenta de automação eficaz, onde empresas de grande porte, poderão enviar todos os documentos de seus colaboradores via email, na data correta , de forma automatizada. Uma atividade que demoraria horas de trabalho, em apenas um clique de compilação de código, utilizando a linguagem de programação *Python*, com a ferramenta "Jupyter" será realizada em fração de segundos.

Palavras-chave: Automação: Linguagens, Python, Jupyter.

1 INTRODUÇÃO

Segundo os dados do *Calium* ensino e inovação(2020a), *Python* foi criado em 1990 por Guido Van Rossum no Centro de Matemática *Stichting* na Holanda como uma sucessora da linguagem ABC. Guido é lembrado como o principal autor do *Python*, mas outros programadores ajudaram com muitas contribuições. Essa linguagem veio para facilitar a codificação dos programadores e contribuir com automatização de processos, pois apresenta fácil entendimento de interpretação.

O projeto "Jupyter" é uma organização sem fins lucrativos criada para desenvolver software de código aberto, padrões abertos e serviços para computação interativa em dezenas de linguagens de programação. Originado do *Python* em 2014, o nome do projeto é uma referência às três principais linguagens de programação suportada por "Jupyter", Julia, Python

e R, é também uma homenagem aos cadernos de anotações de Galileu que registraram a descoberta das luas de Júpiter. Um Notebook "Jupyter" é um ambiente computacional web para a internet rica para criação de documentos para a plataforma "Jypyter". O termo “Notebook” pode, dependendo do contexto, fazer referência a entidades distintas como (aplicativo Web), Jupyter Python (servidor Web) ou ao formato do documento para a plataforma. Um documento Jupyter Notebook é estruturado no formato JSON[1], contendo uma lista ordenada de células de entrada e saída que podem conter código, texto, matemática e gráficos, geralmente com a extensão “ipynb”.

> Segundo Cezar(SENA,P.C.P.2019) Cita que o BIM pode ser compreendido como parte de um movimento mais abrangente de informatização, de integração das informações e de gerenciamento de conhecimento de sistemas produtivos, possíveis somente a partir do aumento do poder computacional, destaca-se a capacidade de automatizar processos, de forma a reduzir trabalho e evitar erros humanos e desperdícios de recursos naturais.(EASTMAN etc al.,2012,p.21)

À medida que a tecnologia evolui, as organizações vão se modernizando mais, na criação de ambientes de trabalho híbridos e na simplificação de suas atividades remotas. As instituições têm buscado ferramentas que agregam na qualidade e eficiência dos processos, onde possam poupar tempo de trabalho de seus colaboradores, e buscar uma maior performance. Uma empresa de grande porte necessita de recursos que irão automatizar tarefas repetitivas.

Python é uma das linguagens que mais tem crescido devido sua compatibilidade(roda na maioria dos sistemas operacionais) e tem capacidade de auxiliar outras linguagens. Programas como *Dropbox*, *Reddit* e *Instagram* são escritos em *Python*, com a ferramenta *Jupyter* é possível realizar várias tarefas em um curto espaço de tempo, e em apenas uma compilação (BARRO, 2022). É uma linguagem de programação indicada para automatização de processos, ela é interpretada, orientada a objetos, de alto nível e com semântica dinâmica. A simplicidade reduz a necessidade de manutenção de um programa e aumenta a facilidade de ser realizada, suporta módulos pacotes, que encoraja a programação modularização e reuso de código.

1.1 Revisão da Literatura

Python é uma linguagem de programação orientada a objetos de propósito geral usado bastante em *data science*, *machine learning*, desenvolvimento de *web*, desenvolvimento de aplicativos, automação de processos, *fintechs* e mais. Por ser uma linguagem de fácil entendimento e com sintaxe lógica, faz com que o uso dela, resulta numa prototipagem de software rápida promovendo manutenções e depurações práticas.

[1] "é um arquivo que contém uma série de dados estruturados em formato texto e é utilizado para transferir informações entre sistemas.". (SOUZA, 2020)

Segundo o site Harve(Harve,Python para que serve: top 5 utilidades .2019) Cita que a *StackOverflow* considerou *Python* a linguagem de renome que mais cresce, e a linguagem que os programadores mais desejam aprender. Como por exemplo em 2014 nos Estados Unidos, 80% das 10 melhores universidades de ciência da computação (e 69% dos top 39) ensinam o python em cursos introdutórios.

Jupyter Notebook é um ambiente computacional interativo, no qual você pode combinar execução de código, texto comum, matemática, gráficos e mídias, o código é dividido em vários blocos para que o usuário possa selecionar deles será rodado com isso podendo descobrir mais facilmente qual parte do código está dando erro, essa divisão também permite que se use blocos diferentes para cada tipo de funcionalidade como por exemplo ao mesmo tempo que em um bloco está sendo escrito um código é possível colocar em outro um texto explicando o que esse esse código irá fazer facilitando assim a revisita a esse código.

A biblioteca "*pyautogui"* funciona como uma extensão para o *Python* com isso adicionando funções novas, essa biblioteca tem como principal foco a automação de processos repetitivos e pode ser usado pelo *Python2* e *Python3* e fornece meios para se controlar o *mouse* e o teclado, enquanto esse modulo estiver ativo é recomendado a usuário ou programador que estiver testando não usar o teclado ou mexer no *mouse* pois isso pode acabar interferindo no bom funcionamento dos métodos inseridos no código.

2 DESENVOLVIMENTO

2.1 Iniciando a automação

Como foi apresentado anteriormente a linguagem de programação *Python* é uma boa alternativa quando é preciso automatizar algumas ações, nessa linguagem de programação existe uma biblioteca que são conjuntos de métodos e funções que serão usados nessa otimização, a biblioteca em questão é a "*pyautogui"* que adiciona recursos para a automação de processos, para se executar esse tipo de implementação devemos primeiro preparar o ambiente de implementação, caso já tenha instalado o *python* na máquina usando "*pip install pyautogui"* no *prompt* de comando da sua interface de desenvolvimento integrado(*IDE*), confirmar a instalação e esperar ela ser instalada, após a instalação da biblioteca devemos abrir a *IDE* importar a biblioteca com o comando "*import pyautogui"* a seguir algumas das funções dessa biblioteca:

Tabela 1: Comandos que usam o mouse

Função:	Retorno:
screenWidth, screenHeight = pyautogui.size()	Retorna para as variáveis os valores da altura e largura da tela da máquina.

ISSN: 2594-8083

currentMouseX, currentMouseY = pyautogui.position()	Retorna para as variáveis a coordenadas do mouse com relação a resolução da tela.
pyautogui.click()	Executa a ação de clicar do mouse na posição atual pode receber coordenadas como argumentos e uma string para o botão do mouse que será usado(left,right ou middle)
pyautogui.moveTo()	Essa função recebe três argumentos dois inteiros como coordenada e uma string para o botão do mouse que será usado(left,right ou middle)
pyautogui.move()	Move para a posição descrita na coordenada funciona de forma parecida ao moveTo()
pyautogui.doubleClick()	Executa a ação de duplo clicar do mouse na posição atual essa função pode receber três argumentos inteiros os dois primeiros são as coordenadas e o último
pyautogui.onScreen()	Recebe uma tupla/lista de inteiros como argumento e retorna se está dentro dos limites da tela como verdadeiro ou falso
pyautogui.scroll() ou pyautogui.hscroll(){Para linux e OS X}	Recebe um inteiro para rolar a quantidade de "cliques" é possível mover o mouse para uma posição x,y passando sua coordenada

Fontes: Dos autores com base no documento de Sweigart, Al. *PyAutoGUI* 0.9.53

Tabela 2: Comandos que fazem o uso do teclado

Função	Retorno:
pyautogui.write(' ', interval=)	Digita os caracteres na string que for passada.Interval o usuário adiciona um atraso para pressionar cada tecla de caractere.
pyautogui.press(' ')	É realmente apenas um wrapper para as funções keyDown()e keyUp().
pyautogui.keyDown(' ')	Simula pressionar uma tecla para baixo e depois soltá-la.

pyautogui.keyUp(' ')	Simula pressionar uma tecla para baixo e depois soltá-la.

Fontes: Dos autores com base no documento de Sweigart, Al. *PyAutoGUI* 0.9.53

Tabela 3: Comandos que apresentam caixa de texto

Função:	Retorno:
pyautogui.alert(text='', title='', button='OK')	Apresenta uma mensagem simples e um único botão de ok retornando um texto quando o botão for pressionado
pyautogui.confirm(text='', title='', buttons=['OK', 'Cancel'])	Mostra uma caixa de mensagem com botões de ok e cancelar e retorna o que foi clicado.
pyautogui.prompt(text='', title='' , default='')	Apresenta uma caixa de mensagem com texto que pode ser editada pelo usuário, um botão de ok e um de cancelar, retorna o texto caso ok for clicado caso contrário ele não retorna nada.
pyautogui.password(text='', title='', default='', mask='*'))	Mostra uma caixa de texto que pode ser inserido conteúdo, botões de ok e cancelar, o caractere usado no mask aparece no lugar dos dígitos, Retorna o texto inserido ou nada se for cancelado.

Fontes: Dos autores com base no documento de Sweigart, Al. *PyAutoGUI* 0.9.53

Tabela 4: Função para gerar imagens da tela

Função:	Retorno:
pyautogui.screenshot()	Retorna uma imagem como objeto é possível delimitar uma área adicionando com os pontos da área da imagem desejada region(0,0,300,400) como argumento.

Fontes: Dos autores com base no documento de Sweigart, Al. *PyAutoGUI* 0.9.53

Tabela 5: Função que localização de elementos por imagem

Função:	Retorno:
pyautogui.locateOnScreen("nome_da_imagem.png", confidence = 0)	Essa função procura pela imagem na tela e retorna as dimensões em que a imagem está enquadrada.

Fontes: Dos autores com base no documento de Sweigart, Al. *PyAutoGUI* 0.9.53

2.2 Aplicações de automação empresarial

2.2.1 Movendo um arquivo de na tela usando *pyautogui*

Exemplo 1

Como exemplo de uma automação rápida é possível usar o comando "*pyautogui.moveTo*()" em uma coordenada de algum arquivo essa coordenada pode ser obtida pelo "*pyautogui.position*()", podemos clicar nele com "*pyautogui.click*(x,y)" e mover com "*pyautogui.moveTo*(x,y)", com isso o arquivo escolhido é arrastado para uma posição desejada.

2.2.2 Envio de email automático Utilizando a biblioteca *pyautogui*

Apresentaremos um exemplo de automação com base em localizar um arquivo e enviar no *Gmail* de forma automática com o uso do *Pyoutogui*.

Exemplo 2

Antes de iniciar o código deve-se fazer a chamada de duas bibliotecas, a *pyautogui* que vai controlar mouse, teclado e tela do computador e a biblioteca time que vai permitir com que o usuário consiga fazer uma pausa antes de rodar o próximo código. Após isso é usado o comando "pyautogui.alert" um alerta ao usuário informando uma mensagem da sua escolha. Para finalizar é possível usar o "pyautogui.PAUSE" que é para que o programa faça uma pausa a cada código, pois algumas ações do computador não respondem com o código executado ao mesmo tempo.

Figura 1: Visualização dos comandos na *IDE*

```
import pyautogui
import time

pyautogui.alert("O código vai começar. Não utilize nada do computador até o código finalizar!")
pyautogui.PAUSE = 0.5

# Abrir o Google Drive no computador
pyautogui.press('winleft')
pyautogui.write('firefox')
pyautogui.press('enter')
#time.sleep(1)
pyautogui.write('https://drive.google.com')
pyautogui.press('enter')
```

Fonte: *Hashtag* treinamentos

ISSN: 2594-8083

Após esses comandos iniciais, foi usado uma função que permitiu com que o usuário pressionasse algumas teclas com o "pyautogui.press" e escrever utilizando o "pyautogui.write". Depois utiliza-se o time.sleep para fazer um “pause” no código além para o computador tenha um tempo maior para carregar a página antes de prosseguir para o próximo passo do código.

Figura 2: Demonstração da escrita do comando

```
# Entrar na área de trabalho
pyautogui.hotkey('winleft','d')
```

Fonte: *Hashtag* treinamentos

Utilizou-se o "pyautogui.hotkey" para poder utilizar atalhos, como "CTRL + C" , "CTRL + V" ou o que utilizamos que foi "Windows + D" para ir até a área de trabalho.

Figura 3: Em código a movimentando e arrastando o arquivo com o *mouse*

```
# Clicar no arquivo e arrastar
pyautogui.moveTo(567,38)
pyautogui.mouseDown()
pyautogui.moveTo(756,635)
```

Fonte: Hashtag treinamentos

Depois de ir até a área de trabalho, foi utilizado a função "pyautogui.moveTo" para mover o *mouse* até um local determinado. Utilizamos o "pyautogui.position" para verificar a posição do seu cursor no momento em que roda esse código. Após posicionar o *mouse*, usou-se o "pyautogui.mouseDown" para clicar com o botão do *mouse* e segurar.

Figura 4: Trocando de tela por comandos na *IDE*

```
# Enquanto estiver arrastando mudar para a página do Google Drive
pyautogui.hotkey('alt','tab')
```

Fonte: Hashtag treinamentos

Foi utilizado para acionar o comando "ALT + TAB" para alterar as abas do computador e voltar para a janela onde temos o *Google Drive* aberto por isso o comando "pyautogui.hotkey(‘alt’,’tab’)" para poder utilizar esse comando das duas teclas com o atalho.

Figura 5: Soltando o arquivo que estava sendo arrastado pelo *mouse*

```
# Soltar o arquivo dentro do Google Drive
pyautogui.mouseUp()
```

Fonte: Hashtag treinamentos

Como o arquivo já está selecionado, arrastamos para a posição correta e já estamos na página do *Google Drive*, vamos apenas soltar o botão do mouse com o comando "pyautogui.mouseUp".

3 CONCLUSÕES

O "*pyautogui*" é uma plataforma de desenvolvimento de baixo custo que facilita a mudança de aplicativos e sua adaptação a novos requisitos, quer seja uma tarefa simples ou uma grande sequência de atividades complexas, os usuários podem alcançar seus objetivos sem escrever uma linha infinita de códigos. A plataforma fornece recursos que permitem aos usuários implementar novas estruturas ou modelos de sistema operacional rapidamente, tendo recursos que foram testados por diferentes implementações, portanto, as chances de bugs ou falhas de segurança são significativamente reduzidas e com a capacidade de criar mais aplicativos em menos tempo , contudo os custos diminuem automaticamente trazendo redução de mão de obra e trazendo alívio para as equipes de desenvolvimento já sobrecarregadas.

Baseando-se no desenvolvimento desses processos de automação para atividades empresariais, tendo como principal foco a melhoria da eficiência desses procedimentos, o *python* é uma boa opção devido a sua portabilidade e estrutura mais compreensível, sendo assim mais didático a algum profissional com menos experiência em programação, como também torna mais viável de ser apresentado a algum profissional veterano ou supervisor com pouco conhecimento na área.

Então, conclui-se que a automação de processos com *Python* ,com o uso da plataforma *Jupyter Notebook* e com a biblioteca *pyautogui* tem como benefícios a redução de ações repetitivas pelos humanos e a otimização do tempo gasto em envios de documentos empresariais.

REFERÊNCIAS

Barro, Bruna B. As 10 Linguagens de Programação Mais Usadas em 2022: Aprimore suas Habilidades em Desenvolvimento Web . Disponível em: https://www.hostinger.com.br/tutoriais/linguagens-de-programacao-mais-usadas#:~:text=1.-,Python,desenvolvimento%2C%20prototipa%C3%A7%C3%A3o%20e%20automa%C3%A7%C3%A3o%20web. Acesso em: 18/09/2022

Desmond, Kim. What is Python used for? Top 5 Python uses. Disponível em: https://codingnomads.co/blog/python/what-is-python-used-for-python-uses. Acesso em: 02/07/2022.

Harve. Python para que serve: top 5 utilidades Disponível em: https://harve.com.br/blog/programacao-python-blog/python-para-que-serve-top-5-utilidades/#:~:text=Python%20%C3%A9%20uma%20linguagem%20Open,de%20scripts%2C%20fintechs%20e%20mais.. Acesso em: 02/07/2022.

Lira. Automação de programa ou sistema com python. Hashtag treinamentos, 19 de abr. 2021. Disponível em: https://www.hashtagtreinamentos.com/automacao-programa-sistema-python. Acesso em: 15/06/2022.

Sena, Paulo C. P. AUTOMAÇÃO DE PROCESSOS DE PROJETO E PROGRAMAÇÃO EM BIM: DYNAMO, PYTHON E C#. USP. 2019. Disponível em: https://www.teses.usp.br/teses/disponiveis/102/102131/tde-12032020-144132/publico/DissCorrigidaPauloCezarPeixotoSena.pdf. Acesso em : 28/05/2022

Souza, ivan. Afinal, o que é JSON e para que ele serve? Descubra agora!, 22 de set. de 2020. Disponível em: https://rockcontent.com/br/blog/json/. Acesso em: 18/09/2022.

Sweigart, Al. PyAutoGUI 0.9.53 .The Python Package index, 07 de jul. de 2021.Disponível em: https://pypi.org/project/PyAutoGUI/. Acesso em: 15/06/2022.

Sweigart, Al. PyAutoGUI 0.9.53 .Write the docs, 07 de jul. de 2021Disponível em: https://pyautogui.readthedocs.io/en/latest/mouse.html. Acesso em: 15/06/2022.