

IMPACTO DA AUTOMAÇÃO NA EFICIÊNCIA DE PROCESSOS ADMINISTRATIVOS: Um estudo experimental utilizando Python

Mirthys Marinho do Carmo Melo

Mestra em Desenvolvimento de Processos Ambientais
mirthysmcmelo@gmail.com

Matheus de Araújo Ventura

Graduando em Sistemas de Informação – UniRios
araujomatheusventura@gmail.com

RESUMO

As empresas estão sendo cada vez mais pressionadas a melhorar seu desempenho e manter-se atualizadas para atender às demandas do mercado, e devido a isso, a automação de processos administrativos se mostrou um recurso poderoso para alcançar esses objetivos de forma mais rápida e eficiente. Esta pesquisa tem como objetivo o desenvolvimento de um experimento para avaliar o impacto da automação nas tarefas administrativas utilizando a linguagem Python, comparando assim o método manual de realizar tarefas versus o automatizado. A metodologia adotada é de caráter experimental e exploratório, e inclui abordagens qualitativas e quantitativas, descrevendo todas as informações referentes ao experimento e aos resultados. Ao realizar o experimento foram coletados dados relevantes, como o tempo, a precisão e a consistência na execução de tarefas, onde o tempo necessário para executá-las com o método automatizado se destacou por ser consideravelmente menor em comparação ao método manual, e a precisão da automação possibilitou uma melhor eficiência das atividades realizadas. Em conclusão, os resultados deste estudo destacam a importância e a contribuição da automação em Python para as empresas, logo isso sugere que ao automatizarmos tarefas com Python, podemos reduzir significativamente o tempo de trabalho nas organizações, potencializando os serviços e otimizando os recursos disponíveis, facilitando assim a gestão organizacional.

Palavras-chave: Automação de Processos. Eficiência Administrativa. Python.

IMPACT OF AUTOMATION ON THE EFFICIENCY OF ADMINISTRATIVE PROCESSES: An experimental study using Python

ABSTRACT

Companies are increasingly under pressure to improve their performance and stay up to date to meet market demands, and because of this, the automation of administrative processes has proven to be a powerful resource for achieving these goals more quickly and efficiently. This research aims to develop an experiment to evaluate the impact of automation on administrative tasks using the Python language, thus comparing the manual method of performing tasks versus the automated method. The methodology adopted is experimental and exploratory in nature, and includes qualitative and quantitative approaches, describing all information relating to the experiment and results. When carrying out the experiment, relevant data were collected, such as time, precision and consistency in

executing tasks, where the time required to perform them with the automated method stood out for being considerably shorter compared to the manual method, and the accuracy automation enabled better efficiency in the activities carried out. In conclusion, the results of this study highlight the importance and contribution of automation in Python for companies, so this suggests that by automating tasks with Python, we can significantly reduce working time in organizations, enhancing services and optimizing available resources, thus facilitating organizational management.

Keywords: Process Automation. Administrative Efficiency. Python.

IMPACTO DE LA AUTOMATIZACIÓN EN LA EFICIENCIA DE LOS PROCESOS ADMINISTRATIVOS: Un estudio experimental utilizando Python

RESUMEN

Las empresas están cada vez más presionadas para mejorar su desempeño y mantenerse actualizadas para satisfacer las demandas del mercado, y debido a esto, la automatización de los procesos administrativos ha demostrado ser un poderoso recurso para alcanzar estos objetivos de manera más rápida y eficiente. Esta investigación tiene como objetivo desarrollar un experimento para evaluar el impacto de la automatización en tareas administrativas utilizando el lenguaje Python, comparando así el método manual de realización de tareas versus el método automatizado. La metodología adoptada es de naturaleza experimental y exploratoria, e incluye enfoques cualitativos y cuantitativos, describiendo toda la información relacionada con el experimento y los resultados. Al realizar el experimento se recolectaron datos relevantes como tiempo, precisión y consistencia en la ejecución de las tareas, donde el tiempo requerido para realizarlas con el método automatizado destacó por ser considerablemente menor en comparación con el método manual, y la automatización de la precisión permitió una mayor eficiencia en las actividades realizadas. En conclusión, los resultados de este estudio resaltan la importancia y aporte de la automatización en Python para las empresas, por lo que sugiere que al automatizar tareas con Python, podemos reducir significativamente el tiempo de trabajo en las organizaciones, mejorando los servicios y optimizando los recursos disponibles, facilitando así la gestión organizacional.

Palabras clave: Automatización de Procesos. Eficiencia Administrativa. Pitón.

1 INTRODUÇÃO

Com a revolução tecnológica, as empresas vêm passando por uma significativa transformação em suas operações diárias. No âmbito administrativo, a automação de processos tem se destacado de maneira exponencial como uma ferramenta importante para impulsionar a eficiência e otimizar recursos, o que antes demandava horas de trabalho, agora leva-se segundos

para ser realizado. Além disso, a automação reduz significativamente os erros humanos, garantindo maior precisão e consistência nos processos.

Com sua flexibilidade e escalabilidade, a automatização pode ser facilmente adaptada às necessidades em constante mudança das empresas, resultando em maior satisfação do cliente, com tempos de resposta mais rápidos e uma experiência mais personalizada. As organizações estão sempre buscando aumentar sua produtividade, e a automação surge como uma ótima solução neste cenário, garantindo que as tarefas repetitivas sejam executadas de maneira eficaz. Diante deste contexto, este artigo investigou: “Como a automação utilizando Python pode impactar na eficiência dos processos administrativos?”. Esta questão foi primordial para nortear todo o desenvolvimento desta pesquisa.

O presente estudo teve como principal objetivo avaliar o impacto causado pela automação nos processos administrativos utilizando a linguagem Python, adotando uma abordagem experimental, no qual através deste experimento buscou-se especificamente: Identificar a tarefa administrativa à automação; Desenvolver scripts (comandos) para automatizar a tarefa selecionada; Comparar o tempo necessário para realizar a tarefa manualmente versus automática e Analisar a precisão e consistência dos resultados entre métodos manuais e automatizados. Todos esses passos foram de extrema importância para concluir a pesquisa e garantir uma análise com foco na mensuração quantitativa e qualitativa, analisando a eficiência da automação feita em Python.

Este trabalho justifica-se pela necessidade de compreender de forma aprofundada a relação entre automação, eficiência administrativa e a aplicação do Python como ferramenta tecnológica. Ao apresentar um experimento prático que utiliza a linguagem Python para automatizar tarefas, pretende-se fornecer uma base sólida para organizações e quaisquer indivíduos tomarem decisões informadas sobre a implementação de soluções automatizadas em seus processos administrativos. Os resultados deste projeto visam demonstrar as vantagens e possibilidades que o Python oferece, contribuindo assim para a otimização e modernização das práticas administrativas.

Esta pesquisa se encontra estruturada em: introdução, metodologia, revisão de literatura, resultados, considerações finais e, por fim, referências. Os tópicos citados são de suma relevância para a pesquisa, sendo o alicerce para que a mesma fosse construída. A seção

posterior refere-se a metodologia, na qual será exibido tudo que foi utilizado para desenvolver o trabalho.

2 METODOLOGIA

A metodologia científica é definida como um conjunto de técnicas que tem como objetivo conduzir investigações e testar hipóteses para possíveis problemas científicos. O método científico auxilia os investigadores a seguir uma linha lógica e organizada, conduzindo desde a análise do problema até a solução do mesmo. Logo o seu uso é indispensável para o desenvolvimento eficiente de projetos acadêmicos.

Quanto aos objetivos propostos, esta pesquisa se qualifica como uma pesquisa exploratória, pois segundo Severino (2017), uma pesquisa exploratória busca colher informações sobre o objeto de estudo, delimitando assim o campo a ser estudado. Esta pesquisa em questão visou a coleta de dados confiáveis de diversas fontes para ser conduzida de maneira eficiente, logo, foi necessário compreender o assunto trabalhado para elaboração de hipóteses a fim de solucionar a problemática do tema.

Quanto à natureza, este trabalho está incluído na categoria de pesquisa descritiva, pois de acordo com Lakatos e Marconi (2022), a pesquisa descritiva tem como objetivo detalhar os aspectos de uma população, ou detectar relações entre variáveis. Nesta pesquisa, foi feito o uso de variáveis como o tipo da tarefa a ser automatizada com Python e variáveis de desempenho, como tempo, precisão e consistência, na qual foram analisadas e descritas as características dessas variáveis antes e depois da implementação da automação, permitindo uma comparação entre o método manual e automatizado na execução das tarefas administrativas.

Quanto aos procedimentos utilizados, esta pesquisa se caracteriza como experimental (Ver Resultados), pois de acordo com Severino (2017), este tipo de pesquisa coloca um determinado objeto em certas condições técnicas de observação e manipulação, onde este tipo de manipulação geralmente é realizada em laboratório. No contexto desta pesquisa, o experimento desenvolvido foi realizado em um ambiente virtual, no caso em um computador, em que o mesmo possuía as ferramentas necessárias para desenvolver a automação da tarefa em Python.

Para este trabalho, foi feita a coleta das informações através dos seguintes bancos de dados:

Google Acadêmico, Scielo e Minha Biblioteca Virtual, sendo o último fornecido pela UNIRIOS. Desses bancos de dados, foram selecionados autores importantes e relevantes envolvendo o assunto do trabalho. Dentre os autores estão: (OLIVEIRA, 2019; FILHO, 2014 e CAMARGO, 2015), que introduzem sobre a Automação e Processos Administrativos, destacando tudo que abrange, incluindo conceitos e evolução. (MARIANO, 2020; MADAKAM, et al. 2019, 2022; LAMB, 2015) discorrem sobre o RPA, fundamentando e enfatizando sua relação com a robótica, e a inovação de seu uso. (LAMBERT, 2022 e BANIN, 2018) falam sobre a linguagem de programação Python, definindo-a e expondo suas principais qualidades. (BARRETO, 2018; ROCHA, et al. 2012; SCHNAIDER, et al. 2004) explicam sobre as métricas de eficiência, descrevendo suas características e sua importância para o desempenho operacional.

Quanto à abordagem da pesquisa, esta se define como uma pesquisa quali-quantitativa, pois a mesma visa expor tanto dados quantitativos quanto qualitativos para detalhar a análise que foi feita durante o estudo. A pesquisa qualitativa segundo Lozada e Nunes (2018, p. 133), é uma investigação voltada para as características qualitativas do fenômeno estudado, ou seja, ela dá atenção aos aspectos da realidade que não são quantificados. Já a pesquisa quantitativa segundo Appolinário (2011, apud LOZADA & NUNES, 2018), possuem variáveis predeterminadas, medidas e que são expressas numericamente. Na seção a seguir será mostrado todo o embasamento teórico que foi utilizado para desenvolver o experimento da automação.

3 REVISÃO DA LITERATURA

3.1 Automação de Processos Administrativos

Sabe-se que a administração é uma prática ancestral, remontando aos primórdios da civilização, em que a venda e compra de mercadorias, assim como a organização de ideias para solucionar um problema, são situações onde a administração está sempre presente. No entanto, à medida que se passou o tempo, os processos envolvendo a administração começaram a evoluir de forma exponencial, e grandes empresas passaram a empregar a administração como uma ferramenta fundamental para conduzir seus negócios, abrangendo desde a liderança dos colaboradores até os processos organizacionais. De acordo com Oliveira (2019), um processo é um conjunto organizado de atividades sequenciais interligadas, visando atender e superar as necessidades e expectativas tanto dos clientes internos quanto externos de uma empresa, logo, os processos organizacionais se revelam fundamentais para o adequado funcionamento das organizações.

Gradualmente, a administração deixou de ser uma prática exclusivamente humana e começou a ser adotada por máquinas e sistemas, que visam tornar as corporações mais eficientes, automatizando a gerência feita pelas mesmas em seus processos administrativos. Até o surgimento da Revolução Industrial no século XVIII, a produção de processos era feita de forma artesanal. Os artesãos possuíam ferramentas e usavam das suas habilidades para executar processos de transformação, em que a energia para o trabalho vinha exclusivamente da ação humana. Entretanto, com os avanços tecnológicos que ocorreram, muitos processos passaram a ser mecanizados e substituíram a mão de obra, principalmente processos nocivos ao ser humano, e que ao serem automatizados regularmente, resultaram em um impacto significativo para as indústrias de produção (FILHO, 2014).

Segundo Camargo (2014), a automação refere-se ao procedimento de instalar controles automáticos em uma máquina, equipamento ou em um processo. Ou seja, a automação é uma maneira de simplificar tarefas sem a necessidade de intervenção humana, sendo utilizada para automatizar atividades repetitivas e aumentar a precisão na execução de tarefas complexas, revolucionando não apenas a eficiência, como também a escala e a qualidade dos processos empresariais. Com o avanço tecnológico, a integração de sistemas de inteligência artificial (IA) e análise de dados tornou-se uma realidade, capacitando as empresas a tomar decisões mais informadas e estratégicas.

3.2 RPA

O RPA (Robotic Process Automation), que significa Automação Robótica de Processos, é definido segundo (ALISHA; GRAEME, 2019; DRISCOLL, 2018; LACITY; WILLCOCKS; CRAIG, 2015a apud MARIANO, 2020), como um método avançado e inteligente de executar tarefas por meio de softwares (programas). Este tipo de tecnologia geralmente está relacionado com a visão de robôs físicos executando tarefas que são normalmente realizadas por humanos. (LACITY, et al., 2015, apud MADAKAM, et al., 2019).

De acordo com Lamb (2015), um robô é uma máquina eletromecânica que realiza tarefas tanto de forma autônoma quanto por orientação de terceiros. Essas máquinas são programadas com precisão para executar trabalhos específicos ou gerais, sendo comumente empregadas na manufatura automotiva, indústria de alimentos e bebidas, indústria farmacêutica, entre outras

áreas comuns de sua atuação. Entretanto, quando falamos do RPA, o mesmo está relacionado com a robótica, porém, possui uma abordagem diferente. Pois segundo Kirchner (2017, apud MARIANO, 2020), da mesma forma que os robôs convencionais automatizam a rotina de produção em fábricas, o RPA possui foco em automatizar as tarefas relacionadas a dados e informações nas empresas. Em outras palavras, essa ferramenta não apenas simula o comportamento humano ao realizar tarefas repetitivas e de rotina, mas também viabiliza um aumento significativo na produtividade dos serviços administrativos empresariais.

Ao considerarmos o contexto mais amplo da transformação digital, o RPA desempenha um papel crucial na hiperautomação, que busca a automação integrada de processos de negócios por meio de diversas tecnologias, incluindo IA, aprendizado de máquina e automação de processos digitais. Conforme Hofmann et al.(2020, apud MADAKAM, et al. 2022) os benefícios que serão conquistados com a implementação do RPA nos processos de negócios são a eficiência e eficácia operacional, além da diminuição dos erros, permitindo que a inteligência humana seja direcionada para necessidades mais estratégicas, como a concepção de novas soluções de produtos.

Segundo Sara Burnett, vice-presidente de pesquisa da Everest Group, a Automação Robótica de Processos representa a próxima inovação que transformará a terceirização como a conhecemos. Há uma clara tendência de empresas buscando se posicionar como líderes na prestação de serviços de automação, antecipando uma possível corrida armamentista pela inovação em ferramentas automatizadas, que inevitavelmente trará novas ofertas e modelos de entrega (apud MADAKAM et al., 2019). O RPA, está cada vez mais chamando a atenção das corporações com sua progressão contínua e adaptabilidade, e em breve todas as organizações estarão incorporando essa tecnologia em suas operações.

3.3 Linguagem Python

O Python é uma linguagem de programação amplamente adotada no mercado de desenvolvimento de sistemas, jogos e inteligências artificiais. Ele foi concebido pelo programador Guido Van Rossum no início dos anos 1990, com o propósito de resolver problemas em sistemas de computadores modernos (LAMBERT, 2022). Esta linguagem por ser de alto nível e de uso geral, se destaca na sua versatilidade e eficiência, pois possui portabilidade para diversos sistemas operacionais, aplicabilidade para diferentes áreas no

desenvolvimento de software e além disso, é uma das linguagens mais fáceis de aprender, pelo fato de sua sintaxe ser simples e intuitiva (BANIN, 2018). A linguagem Python ainda conta com uma gama de ferramentas (bibliotecas e frameworks) para facilitar o desenvolvimento de aplicações robustas e escaláveis, e possui código livre, isto é, qualquer um pode utilizá-la para desenvolver seus projetos sem qualquer restrição.

3.3.1 Python na Automação de Processos

Quando trazemos o Python para o ramo dos negócios, sua aplicabilidade se estende amplamente, sendo frequentemente utilizado para produzir soluções envolvendo a automação de tarefas rotineiras e análise de dados, incluindo seu uso em automações RPA. A automatização de tarefas é apenas o ponto de partida, Python oferece diversas possibilidades para otimizar processos, reduzir custos e aumentar a eficiência operacional das organizações, abrangendo desde a criação de scripts para simplificar operações repetitivas até o desenvolvimento de sistemas completos de gestão de recursos, contabilidade ou logística. À medida que as empresas buscam se manter competitivas em um mercado em constante evolução, o Python se apresenta não apenas como uma escolha viável, mas como um aliado estratégico na busca pela inovação e eficácia operacional.

3.4 Métricas de Eficiência

As métricas de eficiência são vitais no ambiente empresarial, pois são essenciais para monitorar e aprimorar o rendimento. Segundo Sommerville (2007, apud BARRETO, 2018), métricas referem-se a todas as medidas associadas a um sistema de software, processo ou documentação. De acordo com Rocha et al. (2012), essas medidas caracterizam, em termos quantitativos, propriedades de um objeto e fornecem dados numéricos para apoiar na tomada de decisões tanto técnicas quanto de negócios. Schnaider et al. (2004), salientam que os benefícios obtidos ao realizar medições são as ações e decisões tomadas com base nas análises dos dados, e não só apenas a coleta desses dados. Assim, o principal objetivo é determinar como essas informações serão empregadas para impulsionar a eficiência e o desempenho organizacional.

É necessário estabelecer processos robustos de análise e interpretação dessas métricas, transformando-as em informações utilizáveis. Nesse sentido, a implementação de sistemas de IA e análise de dados torna-se cada vez mais relevante, permitindo uma compreensão mais

profunda dos padrões e tendências subjacentes aos dados coletados. Além disso, é importante ressaltar que as métricas de eficiência devem ser alinhadas aos objetivos estratégicos da organização, de modo a garantir que as medidas de desempenho estejam diretamente vinculadas à missão e visão da empresa. Dessa forma, as métricas não se tornam apenas números isolados, mas sim ferramentas estratégicas para estimular o avanço da organização e atingir as metas predefinidas.

4 RESULTADOS

No contexto do presente artigo, para realizar o experimento da automação de tarefas, foi utilizado algumas das bibliotecas mais importantes que o Python dispõe, como o Pandas, o Plotly e o Smtplib, que atuam respectivamente, na manipulação e análise de base de dados, geração de gráficos personalizados e envio automático de emails. Essas ferramentas foram as peças chaves para que a automação fosse realizada de forma eficiente junto a outras bibliotecas relevantes que serviram de apoio. Nesta seção será exibido como foi feito o experimento e posteriormente os resultados que foram coletados.

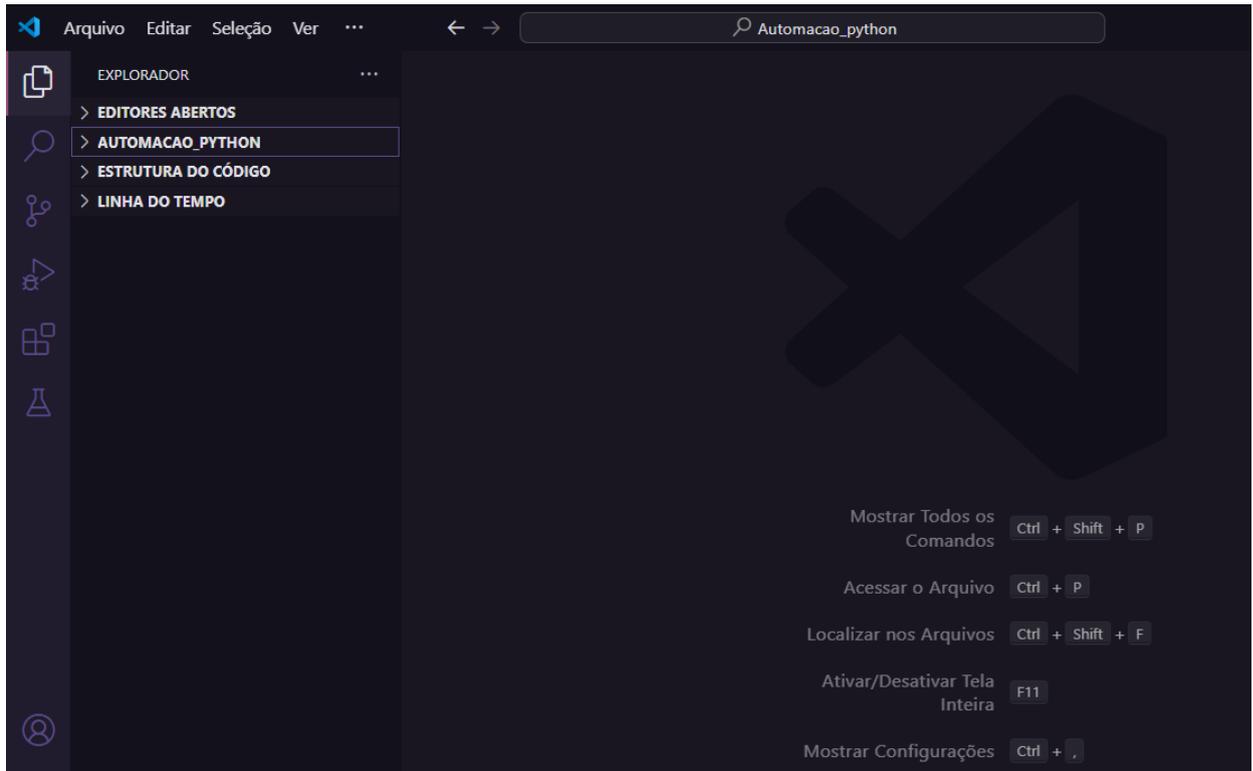
4.1 Tarefa administrativa que foi aplicada à automação

O seguinte experimento de automação ocorreu em uma tarefa administrativa de rotina, em que foi simulado a construção de um relatório de vendas mensal simples de uma loja de acordo com uma base de dados para ser enviado por e-mail. Esse tipo de tarefa é muito comum em organizações quando querem visualizar e entender como estão seu desempenho, para observar o que precisam melhorar e atribuir soluções para possíveis problemas.

4.2 Desenvolvimento dos scripts (comandos) Python para automatizar a tarefa

Para desenvolver o programa em Python foi utilizado um Ambiente de Desenvolvimento Integrado (IDE) chamado Visual Studio Code, onde o mesmo oferece uma interface para programar sites, aplicativos e outras soluções. Abaixo será exibido através de imagens os principais trechos de código que contribuíram para a elaboração da automação.

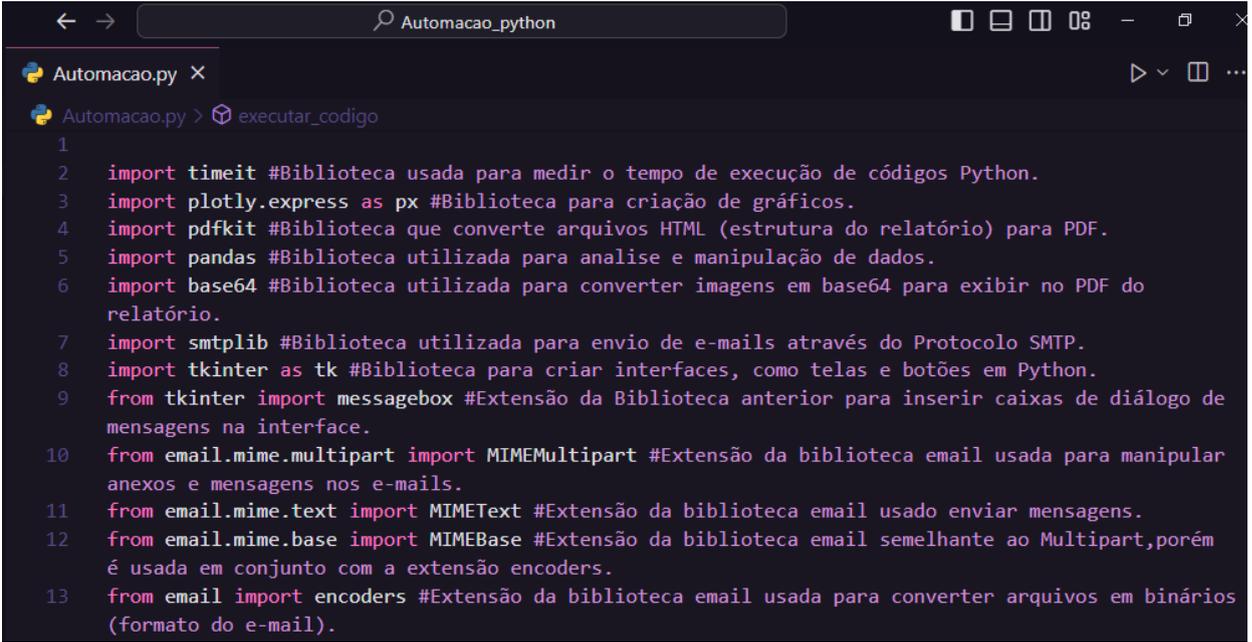
Figura 01 - IDE Visual Studio Code



Fonte: Autor (2024).

A Figura 01 mostra o ambiente em que foi desenvolvido todo o programa assim como também onde foram realizados todos os testes envolvendo a automação. A partir daqui será exibido todo o percurso para realização da automação em Python.

Figura 02 - Bibliotecas utilizadas no programa

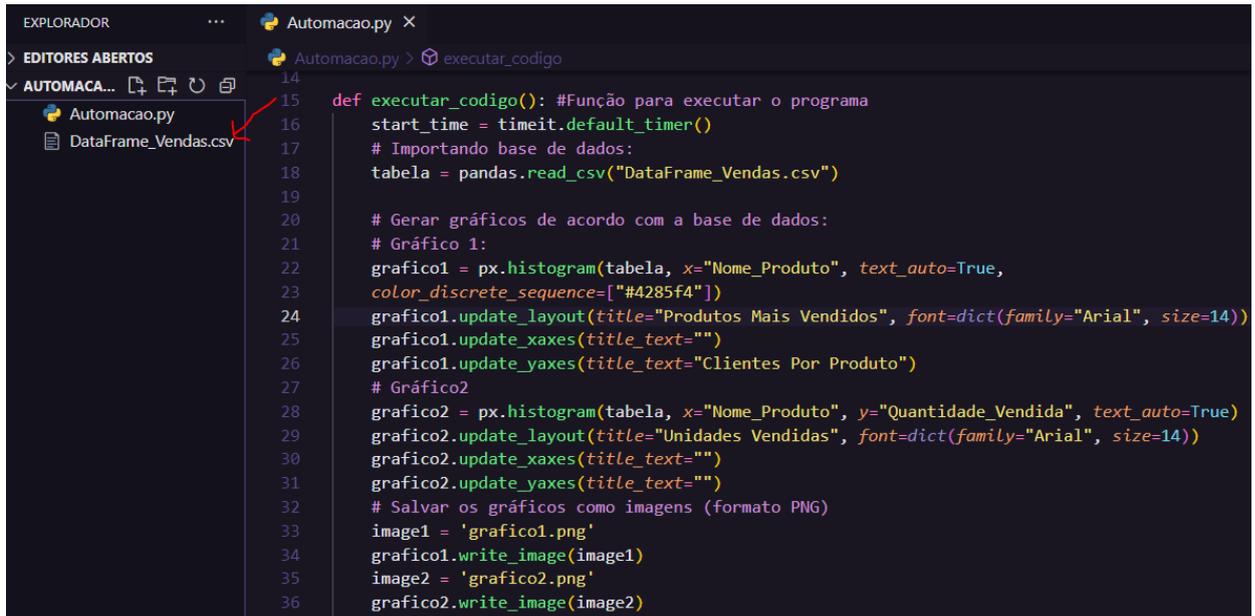


```
Automacao.py x
Automacao.py > executar_codigo
1
2 import timeit #Biblioteca usada para medir o tempo de execução de códigos Python.
3 import plotly.express as px #Biblioteca para criação de gráficos.
4 import pdfkit #Biblioteca que converte arquivos HTML (estrutura do relatório) para PDF.
5 import pandas #Biblioteca utilizada para análise e manipulação de dados.
6 import base64 #Biblioteca utilizada para converter imagens em base64 para exibir no PDF do
relatório.
7 import smtplib #Biblioteca utilizada para envio de e-mails através do Protocolo SMTP.
8 import tkinter as tk #Biblioteca para criar interfaces, como telas e botões em Python.
9 from tkinter import messagebox #Extensão da Biblioteca anterior para inserir caixas de diálogo de
mensagens na interface.
10 from email.mime.multipart import MIMEMultipart #Extensão da biblioteca email usada para manipular
anexos e mensagens nos e-mails.
11 from email.mime.text import MIMEText #Extensão da biblioteca email usado enviar mensagens.
12 from email.mime.base import MIMEBase #Extensão da biblioteca email semelhante ao Multipart, porém
é usada em conjunto com a extensão encoders.
13 from email import encoders #Extensão da biblioteca email usada para converter arquivos em binários
(formato do e-mail).
```

Fonte: Autor (2024).

A Figura 02 exhibe as bibliotecas (ferramentas) Python que foram utilizadas na criação do programa, cada biblioteca foi essencial para que a execução da automação obtivesse êxito. Na imagem também está comentado ao lado de cada biblioteca a sua função específica, para que facilite a legibilidade do código.

Figura 03 - Importação da base de dados para geração de gráficos



```
EXPLORADOR
... Automacao.py X
EDITORES ABERTOS
AUTOMACA... Automacao.py > executar_codigo
Automacao.py
DataFrame_Vendas.csv
14
15 def executar_codigo(): #Função para executar o programa
16     start_time = timeit.default_timer()
17     # Importando base de dados:
18     tabela = pandas.read_csv("DataFrame_Vendas.csv")
19
20     # Gerar gráficos de acordo com a base de dados:
21     # Gráfico 1:
22     grafico1 = px.histogram(tabela, x="Nome_Produto", text_auto=True,
23                             color_discrete_sequence=["#4285f4"])
24     grafico1.update_layout(title="Produtos Mais Vendidos", font=dict(family="Arial", size=14))
25     grafico1.update_xaxes(title_text="")
26     grafico1.update_yaxes(title_text="Clientes Por Produto")
27     # Gráfico2
28     grafico2 = px.histogram(tabela, x="Nome_Produto", y="Quantidade_Vendida", text_auto=True)
29     grafico2.update_layout(title="Unidades Vendidas", font=dict(family="Arial", size=14))
30     grafico2.update_xaxes(title_text="")
31     grafico2.update_yaxes(title_text="")
32     # Salvar os gráficos como imagens (formato PNG)
33     image1 = 'grafico1.png'
34     grafico1.write_image(image1)
35     image2 = 'grafico2.png'
36     grafico2.write_image(image2)
```

Fonte: Autor (2024).

Na Figura 03 é feita a importação da base de dados (destacada pela seta vermelha) utilizando a biblioteca Pandas e a partir desta base de dados foram gerados dois gráficos utilizando a biblioteca Plotly (sendo representada por “px”), em que a mesma utiliza um método semelhante ao excel na geração de gráficos, o uso dos eixos x e y. Assim que os gráficos foram gerados foi realizada sua conversão para arquivos PNG para serem anexados posteriormente em um arquivo HTML para criação do relatório.

Figura 04 - Criação da estrutura do relatório com HTML

```
Automacao.py X
Automacao.py > executar_codigo
42
43     # Calcular valor total das vendas para inserir no relatório:
44     valor_unidades = tabela['Preco_Unitario'] * tabela['Quantidade_Vendida']
45     # Agrupando e somando os valores totais de cada produto
46     total_vendas = "R${:,.2f}".format(valor_unidades.sum())
47
48     # Criando arquivo HTML para o relatório:
49     html_content = f"""
50     <html>
51     <head>
52     <title>Relatório de Vendas</title>
53
54     <style>
55         body {{
56             font-family: Arial;
57             max-width: 700px;
58             margin: 90px auto;
59             padding: 20px
60         }}
61
62         h2, h3 {{
63             text-align: center;
64             color: #000;
65         }}
66         h3 {{
67             margin-top: 40px;
68             margin-bottom: 40px;
69         }}
70
71         p {{
72             font-size: 15pt;
73             text-align: justify;
74             line-height: 1.6em;
```

Fonte: Autor (2024).

Na Figura 04, podemos ver uma parte da estrutura do relatório feito na linguagem de marcação HTML, e além disso, é possível visualizar também algumas funções para calcular certas informações da base de dados que foi importada anteriormente, como o valor total das vendas, e inseri-las no relatório para análise.

Figura 05 - Geração de arquivo HTML e conversão para PDF

```
Automacao.py X
Automacao.py > executar_codigo
111     # Salvar o HTML em um arquivo
112     html_filename = 'relatorio_vendas.html'
113     file = open(html_filename, 'w', encoding="UTF-8")
114     file.write(html_content)
115     # Converter o HTML para PDF usando a biblioteca Pdfkit
116     options = {
117         'encoding': 'utf-8' # Codificação UTF-8 para o PDF ser exibido sem erros de formatação
118     }
119     pdfkit.from_file(html_filename, 'relatorio_vendas.pdf', options=options)
120
```

Fonte: Autor (2024).

A Figura 05 mostra a geração do arquivo HTML e sua conversão em PDF utilizando a biblioteca Pdfkit, que garante que os dados convertidos estejam devidamente formatados e não apresentem nenhuma inconsistência. Logo depois de ser convertido para PDF, o relatório será enviado por e-mail.

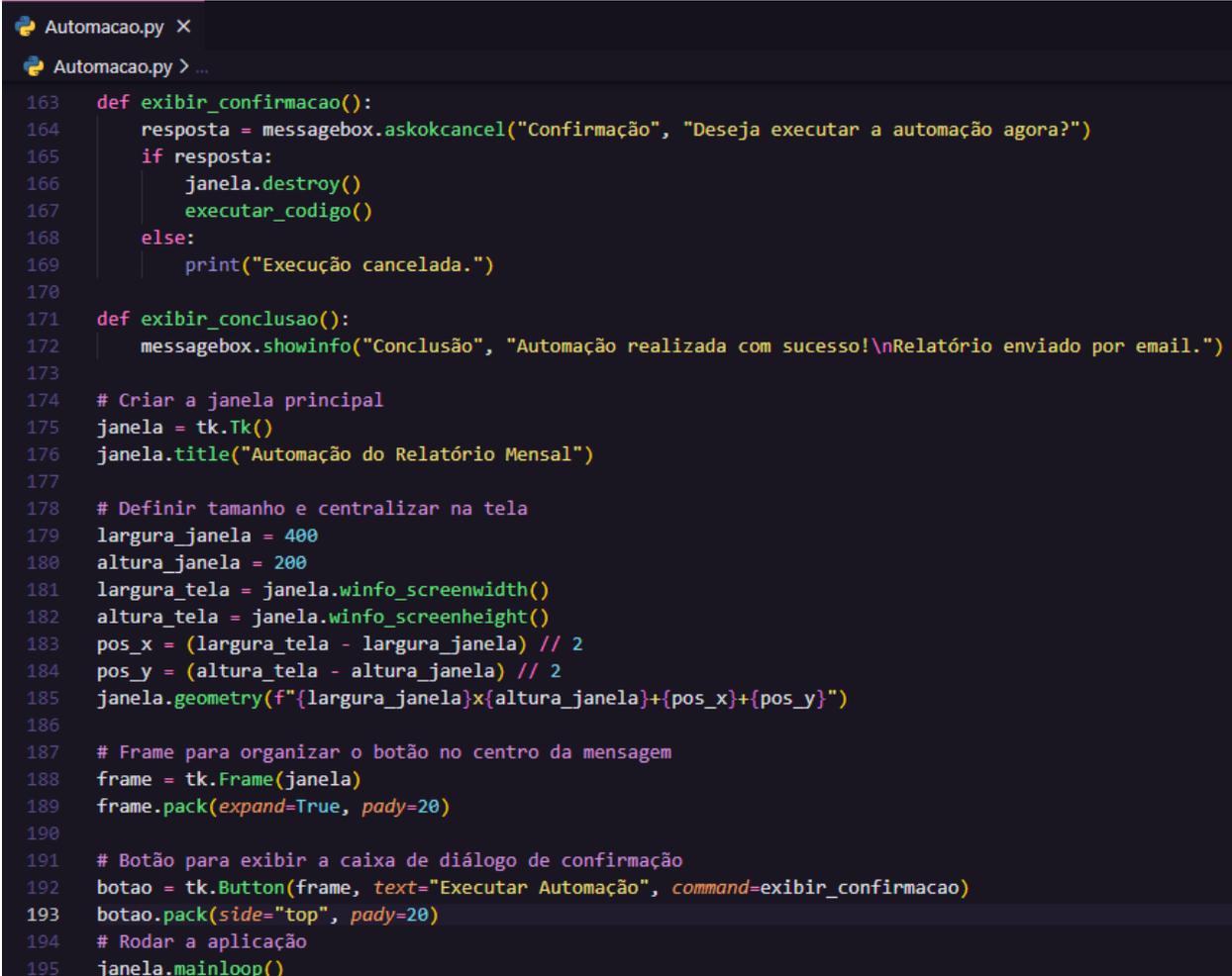
Figura 06 - Configuração para envio do relatório por e-mail

```
Automacao.py X
Automacao.py > executar_codigo
122     # Configurações do servidor SMTP para envio do e-mail
123     smtp_servidor = 'smtp.gmail.com'
124     smtp_porta = 587
125     smtp_email = 'example@gmail.com'
126     smtp_senha = 'senha'
127     # Iniciando o Servidor
128     server = smtplib.SMTP(smtp_servidor, smtp_porta)
129     server.starttls()
130     server.login(smtp_email, smtp_senha)
131     # Configurações dos detalhes do e-mail
132     remetente = 'example@gmail.com'
133     destinatario = 'example@gmail.com'
134     assunto = 'Relatório de Vendas Mensal - Envio automático'
135     corpo_email = 'Segue em anexo o relatório das vendas deste mês: 04/2024'
136     # Preparando mensagem para envio em formato Multipart
137     email_msg = MIME multipart()
138     email_msg['From'] = remetente
139     email_msg['To'] = destinatario
140     email_msg['Subject'] = assunto
141     email_msg.attach(MIMEText(corpo_email, 'html'))
142     # Abrir arquivo em modo em modo leitura e binário
143     anexo = open('relatorio_vendas.pdf', 'rb')
144     # Ler arquivo no modo binário, codificamos e atribuímos a variável (arquivos são enviados em formato binário por email)
145     anexar = MIMEBase('application', 'octet-stream')
146     anexar.set_payload(anexo.read())
147     encoders.encode_base64(anexar)
148     # Adicionando cabeçalho no anexo que será enviado por email
149     anexar.add_header('Content-Disposition', f'attachment; filename=relatorio_vendas.pdf')
150     # Fechar arquivo
151     anexo.close()
152     # Colocando anexo no corpo do email
153     email_msg.attach(anexar)
154     server.sendmail(email_msg['From'], [email_msg['To']], email_msg.as_string().encode('utf-8'))
```

Fonte: Autor (2024).

Na Figura 06 acima, é possível visualizar como ocorre o envio do e-mail utilizando funções das bibliotecas Smtplib e Email, exibindo desde as configurações do servidor do protocolo SMTP (envio de e-mails) até o anexo e envio do arquivo por e-mail. Com o uso destas bibliotecas é possível criar scripts automáticos de e-mails em diversas ocasiões, tornando vantajoso para qualquer organização que as utilize.

Figura 07 - Criação de janela para executar a automação

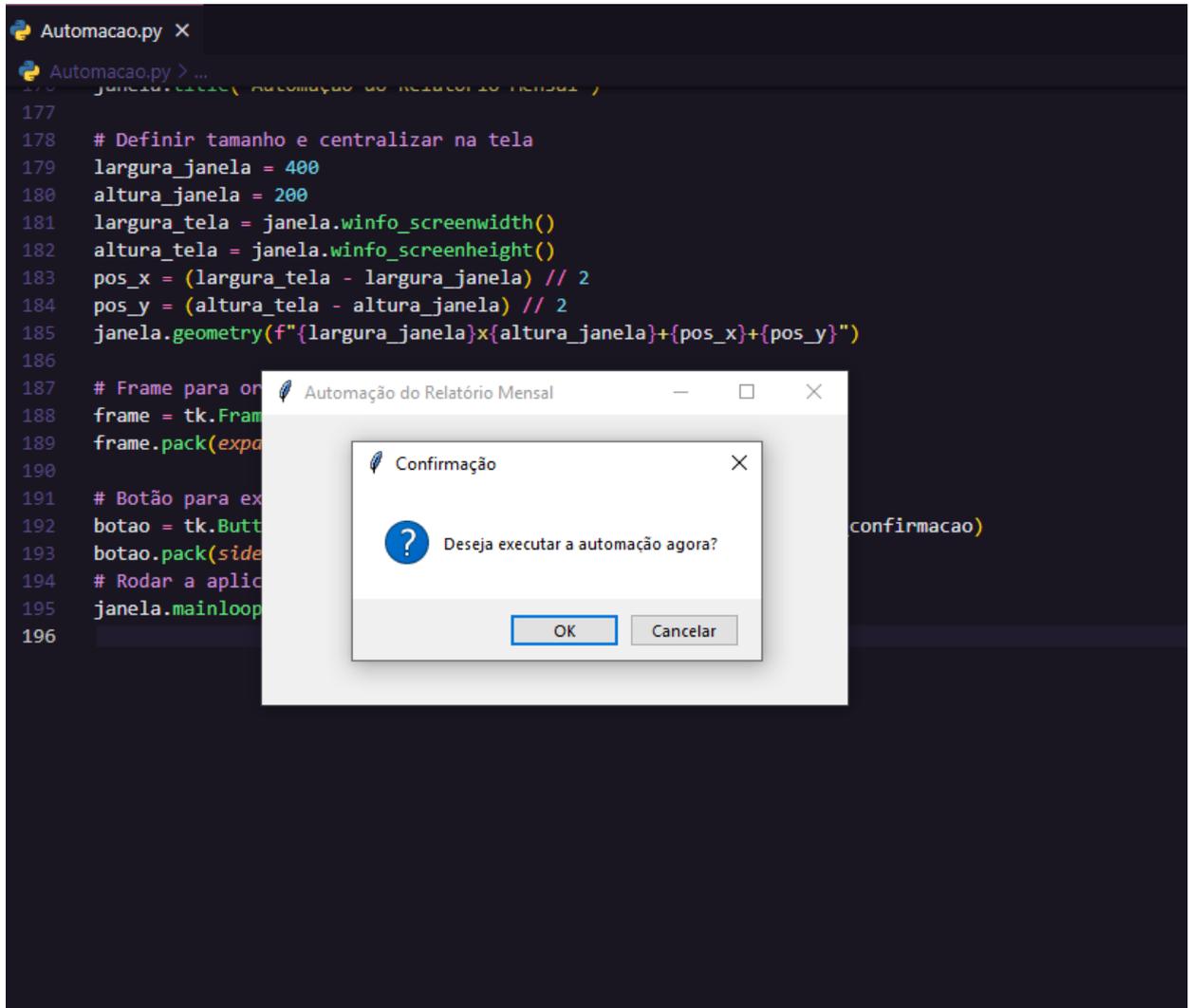


```
Automacao.py X
Automacao.py > ...
163 def exibir_confirmacao():
164     resposta = messagebox.askokcancel("Confirmação", "Deseja executar a automação agora?")
165     if resposta:
166         janela.destroy()
167         executar_codigo()
168     else:
169         print("Execução cancelada.")
170
171 def exibir_conclusao():
172     messagebox.showinfo("Conclusão", "Automação realizada com sucesso!\nRelatório enviado por email.")
173
174 # Criar a janela principal
175 janela = tk.Tk()
176 janela.title("Automação do Relatório Mensal")
177
178 # Definir tamanho e centralizar na tela
179 largura_janela = 400
180 altura_janela = 200
181 largura_tela = janela.winfo_screenwidth()
182 altura_tela = janela.winfo_screenheight()
183 pos_x = (largura_tela - largura_janela) // 2
184 pos_y = (altura_tela - altura_janela) // 2
185 janela.geometry(f"{largura_janela}x{altura_janela}+{pos_x}+{pos_y}")
186
187 # Frame para organizar o botão no centro da mensagem
188 frame = tk.Frame(janela)
189 frame.pack(expand=True, pady=20)
190
191 # Botão para exibir a caixa de diálogo de confirmação
192 botao = tk.Button(frame, text="Executar Automação", command=exibir_confirmacao)
193 botao.pack(side="top", pady=20)
194 # Rodar a aplicação
195 janela.mainloop()
```

Fonte: Autor (2024).

Na Figura 07, é exibido a criação da interface para executar a automação Python, na qual utiliza-se a biblioteca Tkinter (representado por “tk”), para construção de interfaces personalizadas. A interface desenvolvida serve para informar ao usuário que o programa está em execução e dar controle ao mesmo de executar ou não a automação.

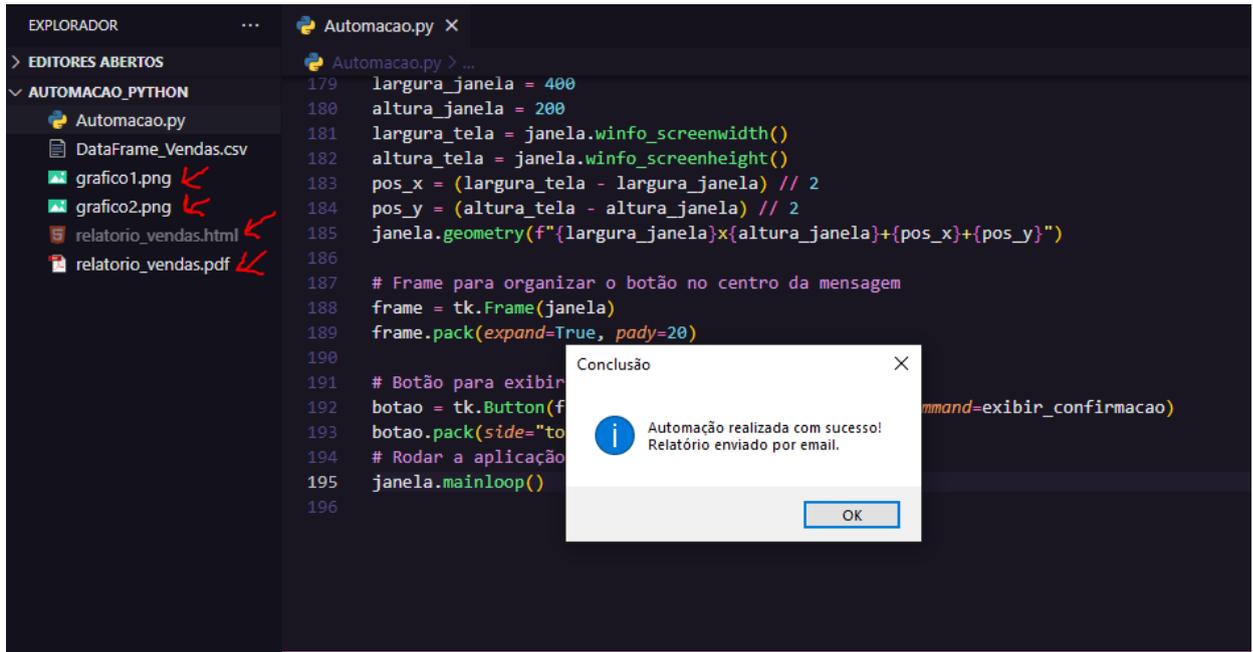
Figura 08 - Executando o Programa



Fonte: Autor (2024).

A Figura 08 nos mostra a interface para executar a automação, em que no contexto do experimento simulado, o funcionário do setor da loja pode optar por executar ou não a automação do relatório. Este programa foi executado em uma IDE, entretanto, é possível executá-lo de outra forma, utilizando o Agendador de Tarefas presente na maioria dos sistemas operacionais, inclusive no Windows, onde este experimento foi realizado.

Figura 09 - Automação executada



Fonte: Autor (2024).

Na Figura 09 é exibida a finalização da automação, na qual é possível visualizar do lado esquerdo da figura os gráficos gerados e o relatório em HTML e PDF. Na automação ele gera os arquivos automaticamente na mesma pasta do projeto que está sendo executado. Por fim, o programa coleta o arquivo PDF e o envia para o e-mail fornecido no código, no entanto, por questões de segurança foi utilizado e-mails fictícios para exibir nas imagens do programa presentes neste trabalho.

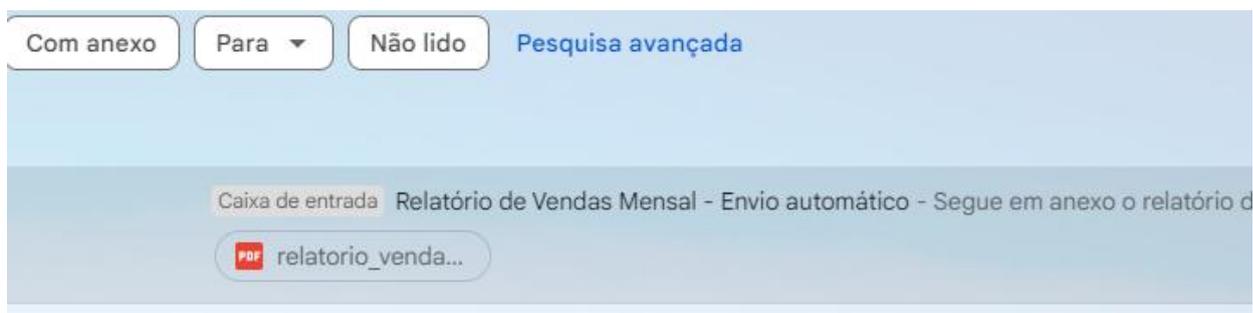
Figura 10 - Relatório Gerado



Fonte: Autor (2024).

A Figura 10 exibe um parte do relatório gerado em Python, no qual possui informações como período a qual é referente, o total das vendas do mês e um gráfico mostrando a quantidade de clientes que compraram determinados produtos.

Figura 11 - Envio por email



Fonte: Autor (2024).

A Figura 11 mostra o registro do relatório enviado por email para o destinatário, concluindo assim o procedimento da automação da tarefa. Para tarefa ser automatizada foi necessário uma série de passos que foram exibidos nas imagens anteriores, no qual o algoritmo desenvolvido possui diversas linhas de código, e apesar de se tratar de um procedimento simples, necessitou de muitas funções específicas para ser concretizado, logo, ainda que tenha demandado certo tempo para desenvolver o programa, o mesmo mantém sua eficácia, pois só precisou ser criado uma única vez para realizar uma função quantas vezes forem necessárias.

4.3 Comparação entre os método manual e automático: relevando o tempo, a precisão e consistência dos resultados

Para analisar de maneira eficaz os métodos na realização do experimento, foram feitos vários testes, realizando a mesma tarefa em ambos mais de três vezes, para se ter resultados mais consistentes, e com base na coleta desses dados foi construída uma tabela comparativa geral.

Tabela 01 - Comparação entre os métodos

	Método Manual	Método Automatizado (Python)
Tempo de Execução	O tempo de execução do método manual foi variável e calculado entre 16, 14 e 13 minutos, em que a cada teste o tempo de execução da tarefa diminuiu consideravelmente.	O tempo de execução do método automatizado possui uma diferença considerável em relação ao método anterior, pois o programa conseguiu realizar a tarefa em questão de segundos nos testes realizados, logo, foi calculado valores entre 51, 14 e 13 segundos, melhorando significativamente o tempo de resposta para concluir a tarefa. No entanto, a depender em qual local o programa for executado, o mesmo pode apresentar rapidez ou lentidão na execução, porém nada que impacte negativamente em relação ao método manual.
Precisão	A precisão não foi eficiente, ocorrendo alguns erros, como erros de digitação, falta de atenção e fadiga do operador.	Precisão ao executar a tarefa foi eficiente, não cometendo erros. Porém, por se tratar de um programa, pode apresentar certas falhas, como incompatibilidade com determinadas configurações do sistema operacional utilizado.
Consistência	A consistência foi uniforme, por ser uma tarefa relativamente simples, a maioria dos dados foram padronizados. No entanto, não é garantido que o operador da tarefa mantenha a consistência por longos períodos de tempo, podendo acarretar inconsistências futuras.	A consistência dos resultados dos testes foi uniforme, já que o código por ser automático, pode ser executado por tempo indeterminado, e raramente comete erros. Os resultados mantiveram o padrão, possuindo melhor desempenho em comparação com o método manual.

Com base nos dados coletados e analisados é possível observar como o método automatizado possui vantagem à frente do método manual, não só por ser mais rápido, como também o algoritmo não se fadiga, muito menos se frustra com determinada rotina, pois ao trabalhar com automações, podemos dar mais destaque a tarefas mais urgentes e que necessitam de um operador humano para serem executadas. Esses são pontos importantes a se considerar quando falamos de automação de processos administrativos, que é indispensável para garantir o progresso das organizações.

5 CONSIDERAÇÕES FINAIS

Neste trabalho, foi investigado como a automação utilizando a linguagem Python pode impactar na eficiência dos processos administrativos aplicando um procedimento experimental. Ao comparar a forma manual de realizar tarefas versus a automática utilizando Python durante o experimento, foi analisado o tempo de execução, precisão e a consistência, obtendo como resultado informações vantajosas para as organizações que queiram aplicar a automação em seus processos de negócios.

Considerando os resultados alcançados, ficou evidente que a automação em Python melhora de forma significativa a realização de uma atividade de rotina, destacando-se pela velocidade e a eficiência com que a tarefa foi realizada. No entanto, a automação vista não se limita apenas a somente realizar atividades como essas, mas pode ser expandida para funções mais complexas e de maior escala em uma organização. O que foi visto no experimento na seção anterior, só foi um vislumbre do que o Python é capaz de fazer, logo, o seu uso pode ser adaptado para diversas situações.

As organizações seguem avançando, integrando novas tecnologias sempre que possível para alavancar seus negócios, e isto inclui a automação de processos, porém, muitas não investem seu tempo em recursos que podem garantir um melhor desempenho, por simplesmente não terem o conhecimento prévio de como funciona a aplicabilidade destes recursos. Com os conhecimentos adquiridos no presente estudo as organizações podem ter entendimento o suficiente para aplicarem o uso do Python em suas atividades, e garantir que seu empreendimento evolua a longo prazo. Atualmente existem corporações que já utilizam dessa linguagem de programação para se beneficiar de suas vantagens, mas nem todas possuem essa informação no mercado, e em detrimento disso acabam sendo prejudicadas. Adotando a automação com a linguagem Python, as empresas podem diminuir os custos, realizar mais tarefas em menor tempo e aumentar a eficiência dos seus processos, garantindo a priorização de atividades que devam ser realizadas com maior urgência, e desse modo aprimorar a gestão dos seus serviços organizacionais.

REFERÊNCIAS

- BANIN, Sérgio Luiz. **Python 3 - Conceitos e Aplicações - Uma abordagem didática**. São Paulo: Érica, 2018. E-book. ISBN 9788536530253. Disponível em: <<https://integrada.minhabiblioteca.com.br/#/books/9788536530253>>. Acesso em 07/04/2024, às 01h06min.
- BARRETO, Pedro Pires. **Uma Proposta de Métricas para Avaliar a Efetividade da Execução de Testes de Software**. Recife: Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Departamento de Computação, Universidade Federal Rural de Pernambuco, 2019. 79 f. Disponível em: <<https://repository.ufrpe.br/handle/123456789/1068>>. Acesso em 08/04/2024, às 12h56min.
- CAMARGO, Valter Luís Arlindo de. **Elementos de Automação**. São Paulo: Érica, 2014. E-book. ISBN 9788536518411. Disponível em: <<https://integrada.minhabiblioteca.com.br/#/books/9788536518411>>. Acesso em 10/04/2024, às 01h34min.
- FILHO, Guilherme Filippo. **Automação de Processos e de Sistemas**. São Paulo: Érica, 2014. E-book. ISBN 9788536518138. Disponível em: <<https://integrada.minhabiblioteca.com.br/#/books/9788536518138>>. Acesso em 17/03/2024, às 00h43min.
- LAKATOS, Eva Maria. **Metodologia do Trabalho Científico**. São Paulo: Atlas, 2021. E-book. Disponível em: <<https://integrada.minhabiblioteca.com.br/#/books/9788597026559>>. Acesso em 28/02/2024, às 19h30min.
- LAMB, Frank. **Automação Industrial na Prática (Tekne)**. São Paulo: Bookman, 2015. E-book. ISBN 9788580555141. Disponível em: <<https://integrada.minhabiblioteca.com.br/#/books/9788580555141>>. Acesso em 09/03/2024, às 23h22min.
- LAMBERT, Kenneth A.. **Fundamentos de Python: Primeiros Programas**. São Paulo: Cengage Learning Brasil, 2022. E-book. ISBN 9786555584301. Disponível em: <<https://integrada.minhabiblioteca.com.br/#/books/9786555584301>>. Acesso em 06/04/2024, às 23h23min.
- LOZADA, Gisele; NUNES, Karina da Silva. **Metodologia Científica**. São Paulo: SAGAH, 2019. E-book. Disponível em: <<https://integrada.minhabiblioteca.com.br/#/books/9788595029576>>. Acesso em 28/02/2024, às 19h40min.
- MADAKAM, Somayya; HOLMUKHE, Rajesh M.; JAISWAL, Durgesh Kumar. **The Future Digital Workforce: Robotic Process Automation (RPA)**. JISTEM USP Brazil: Journal of Information Systems and Technology Management, vol.16, 2019. Disponível em: <<https://www.scielo.br/j/jistm/a/m7cqFWJPsWSk8ZnWRN6fR5m/?lang=en#>>. Acesso em 11/03/2024, às 21h00min.
- MADAKAM, Somayya; HOLMUKHE, Rajesh M.; REVULAGADDA, Rajeev K. **Next Generation Intelligent Automation: Hyperautomation**. JISTEM USP Brazil: Journal of Information Systems and Technology Management, vol.19, 2022. Disponível em:

<<https://www.scielo.br/j/jistm/a/8BnnjHkvFGrmBFdtnXmhNtC/?lang=en>>. Acesso em 11/03/ 2024, às 21h30min.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Metodologia Científica**. São Paulo: Atlas, 2022. E-book.. Disponível em:
<<https://integrada.minhabiblioteca.com.br/#/books/9786559770670>>. Acesso em 28/02/2024, às 19h50min.

MARIANO, Alex Fernando. **Automação Robótica de Processos: Uma Análise Sobre a Governança de RPA para Grandes Empresas**. São Paulo, 2020. 94 f. Disponível em:
<<https://repositorio.fgv.br/items/a05fab9d-3b32-4af2-ae6e-0311b5e12d25>>. Acesso em 11/03/ 2024, às 19h37min.

OLIVEIRA, Djalma de Pinho Rebouças de. **Administração de Processos, 6ª edição**. São Paulo: Atlas, 2019. E-book. ISBN 9788597021301. Disponível em:
<<https://integrada.minhabiblioteca.com.br/#/books/9788597021301>>. Acesso em 19/03/2024, às 21h02min.

ROCHA, Ana Regina Cavalcanti da; SOUZA, Gleison dos Santos; BARCELLOS, Monalessa Perini. **Medição de Software e Controle Estatístico de Processos**. Brasília: Ministério da Ciência, Tecnologia e Inovação, 2012. Disponível em:
<https://nemo.inf.ufes.br/wp-content/uploads/Monalessa/LivroMedicao&CEP_RochaSantosBarcellos_2012.pdf>. Acesso em 08/04/2024, às 00h20min.

SCHNAIDER, Lílian; SANTOS, Gleison; MONTONI, Mariano; ROCHA, Ana Regina. **Uma Abordagem para Medição e Análise em Projetos de Desenvolvimento de Software**. Porto Alegre: Sociedade Brasileira de Computação, 2004. Disponível em:
<<https://sol.sbc.org.br/index.php/sbqs/article/view/16199>>. Acesso em 08/04/2024, às 00h46min.

SEVERINO, Antônio Joaquim. **Metodologia do Trabalho Científico**. São Paulo: Cortez Editora, 2017. E-book. ISBN 9788524925207. Disponível em:
<<https://integrada.minhabiblioteca.com.br/#/books/9788524925207>>. Acesso em 28/02/ 2024, às 19h55min.