



CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA

FACULDADE DE TECNOLOGIA DE LINS PROF. ANTONIO SEABRA

**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

CIBELE SILVA LHAMAS

RICARDO RODRIGUES LIMA

**AUTOMAÇÃO ROBÓTICA DE PROCESSOS PARA EXTRAÇÃO DE
RELATÓRIOS**

Escaneie a imagem para verificar a autenticidade do documento
Hash SHA256 do PDF original #61b28ae29fe906fab8ce2150048fb415973a2dac5f9e6688051ee5cf962a1914
<https://valida.ae/b2cd8da73ddef1aa73820780a592327289c846676bb0c025b>



**LINS/SP
2º SEMESTRE/2023**



CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA

**FACULDADE DE TECNOLOGIA DE LINS PROF. ANTONIO SEABRA
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

**CIBELE SILVA LHAMAS
RICARDO RODRIGUES LIMA**

AUTOMAÇÃO ROBÓTICA DE PROCESSOS PARA EXTRAÇÃO DE RELATÓRIOS

Trabalho de Conclusão de Curso (TCC) apresentado à
Faculdade de Tecnologia de Lins para a obtenção do
título de Tecnólogo (a) em Análise e Desenvolvimento
de Sistemas.

Orientador: Me. Felipe Maciel Rodrigues.

Escaneie a imagem para verificar a autenticidade do documento
Hash SHA256 do PDF original #61b28ae29fe906fab8ce2150048fb415973a2dac5f9e6688051ee5cf962a1914
<https://valida.ae/b2cd8da73ddef1aa73820780a592327289c846676bb0c025b>





Escaneie a imagem para verificar a autenticidade do documento
Hash SHA256 do PDF original #61b28ae29fe906fab8ce2150048fb415973a2dac5f9e6688051ee5cf962a1914
<https://valida.ae/b2cd8da73ddef1aa73820780a592327289c846676bb0c025b>

Lhamas, Cibele da Silva

L688a Automação robótica de processos para extração de relatórios / Cibele da Silva Lhamas, Ricardo Rodrigues Lima. — Lins, 2023.

43f.

Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) — Faculdade de Tecnologia de Lins Professor Antonio Seabra: Lins, 2023.

Orientador(a): Me. Felipe Rodrigues

1. RPA. 2. Python. 3. Automação. 4. Robótica. I. Lima, Ricardo Rodrigues. II. Rodrigues, Felipe. III. Faculdade de Tecnologia de Lins Professor Antonio Seabra. IV. Título.

CDD 004.21

Gerada automaticamente pelo módulo web de ficha catalográfica da FATEC Lins mediante dados fornecidos pelo(a) autor(a).





Escaneie a imagem para verificar a autenticidade do documento
Hash SHA256 do PDF original #61b28ae29fe906fab8ce2150048fb415973a2dac5f9e6688051ee5cf962a1914
<https://valida.ae/b2cd8da73ddef1aa73820780a592327289c846676bb0c025b>

CIBELE SILVA LHAMAS
RICARDO RODRIGUES LIMA

AUTOMAÇÃO ROBÓTICA DE PROCESSOS PARA EXTRAÇÃO DE RELATÓRIOS

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de Lins como parte dos requisitos para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas sob orientação do Me. Felipe Maciel Rodrigues.

Data de aprovação: ____/____/____

Orientador (Prof. Me. Felipe Maciel Rodrigues)

Examinador 1 (Profa. Me. Gisele Molina Becari)

Examinador 2 (Prof. Dr. Fernando Augusto Garcia Muzzi)





Escaneie a imagem para verificar a autenticidade do documento
Hash SHA256 do PDF original #61b28ae29fe906fab8ce2150048fb415973a2dac5f9e6688051ee5cf962a1914
<https://valida.ae/b2cd8da73ddef1aa73820780a592327289c846676bb0c025b>





Escaneie a imagem para verificar a autenticidade do documento
Hash SHA256 do PDF original #61b28ae29fe906fab8ce2150048fb415973a2dac5f9e6688051ee5cf962a1914
<https://valida.ae/b2cd8da73ddef1aa73820780a592327289c846676bb0c025b>



Nós dedicamos a todos que fizeram parte ou que acompanharam nosso trajeto acadêmico, passando pelos professores e família.

Ricardo Rodrigues Lima



AGRADECIMENTOS

Primeiramente, gostaria de expressar minha profunda gratidão a Deus pela graça de me conceder o dom da vida e por guiar-me na superação dos desafios encontrados ao longo deste curso.

À minha dedicada família, que incondicionalmente apoiou minha jornada acadêmica ao longo dos anos, destaco meu pai, Silvio, cuja determinação e excelência profissional são fontes de inspiração para mim.

Ao meu colega de trabalho, Ricardo, agradeço por seu constante encorajamento e por compartilhar comigo as dificuldades enfrentadas durante o desenvolvimento deste trabalho.

Por último, mas não menos importante, expresso minha gratidão ao nosso professor, orientador e amigo, Felipe Maciel. Sua orientação, aliada ao bom humor e paciência, desempenhou um papel fundamental na elaboração deste documento, nos incentivando a perseverar até o final.

Cibele da Silva Lhamas





AGRADECIMENTOS

Agradeço a Deus por guiar e presentear-me com uma família tão espetacular e incrível, aos professores que me apoiaram durante todo o curso.

A minha amiga Cibeles, agradeço por compartilhar os seus conhecimentos e colaborar para esta conclusão de curso.

Ao Me. Felipe Maciel, meu querido amigo e orientador, os meus mais sinceros agradecimentos, por acreditar e encorajar a realizar este documento; me inspiro no seu profissionalismo e caráter.

Ricardo Rodrigues Lima





RESUMO

A utilização da Automação Robótica de Processos (RPA) em uma empresa oferece diversas vantagens, contribuindo para melhorias significativas em eficiência, produtividade e qualidade nos processos organizacionais. Os robôs de software podem executar tarefas repetitivas e operacionais de maneira rápida, consistente e sem variações. Neste estudo são apresentados referenciais teóricos acerca das características de um RPA, tal como os aspectos essenciais que permitem sua evolução. O objetivo deste trabalho é mostrar que a implantação dessa tecnologia é vantajosa para organizações que buscam maior eficiência, e promover redução de atividades que são repetitivas e massivas para um humano fazer. Nesse contexto, para garantir a eficiência atrelada a um robô construído em RPA é apresentado um estudo de caso aplicado na empresa JBS Transportadora de Lins - SP e implementada na linguagem Python juntamente com suas bibliotecas. Os levantamentos e estudos realizados indicam que o benefício dessa implantação tornou uma melhor administração do horário dos funcionários, usando-os em tarefas mais estratégicas, ou seja, resultando em ganho de produtividade e qualidade para o negócio.

Palavras-Chave: RPA.Python.Automação.Robótica.





ABSTRACT

The use of Robotic Process Automation (RPA) in a company offers several advantages, contributing to significant improvements in efficiency, productivity and quality in organizational processes. Software robots can perform repetitive and operational tasks quickly, consistently and without variations. This study presents theoretical references on the characteristics of a RPA, as well as the essential aspects that allow it to evolve. The aim of this work is to show that implementing this technology is advantageous for organizations seeking greater efficiency, and to promote a reduction in activities that are repetitive and massive for a human to do. In this context, in order to guarantee the efficiency associated with a robot built using RPA, a case study is presented which was applied to the JBS Transportadora company in Lins - SP and implemented in the Python language along with its libraries. The surveys and studies carried out indicate that the benefit of this implementation has been better management of employee hours, using them for more strategic tasks, in other words, resulting in productivity and quality gains for the business.

Keywords: RPA.Python.Robotic Automation.





LISTA DE ILUSTRAÇÕES

Figura 3.1 - JBS.Updater.Corporate.....	23
Figura 3.2 - Tela de Autenticação do sistema Corporate.....	24
Figura 3.3 - Tela inicial do ERP Corporate.....	26
Figura 3.4 – Código da tela Razão.....	27
Figura 3.5 – Tela Razão Contábil Detalhado.....	27
Figura 3.6 – Relatório de Razão Contábil Detalhado.....	29
Figura 3.7 - Relatório Razão Contábil Detalhado para download.....	30
Figura 3.8 – Arquivo R. C. Detalhado.xlsx.....	31
Figura 3.9 – Código da tela Razão de Frete.....	32
Figura 3.10 – Tela Razão de Frete Consolidado.....	33
Figura 3.11 – Relatório de Razão de Frete Consolidado.....	34
Figura 3.12 – Arquivo R. C. Detalhado.xlsx.....	35
Figura 3.13 – Tela Pasta Relatório Zip.....	38
Figura 3.14 – Operação Transporte TRP.....	40
Figura 3.15 – Inclusão de novo documento.....	40





LISTA DE GRÁFICOS

Gráfico 2.1 – Evolução da RPA.....	15
Gráfico 3.1 – Diagrama.....	22





LISTA DE TABELAS

Tabela 2.1 - Bibliotecas Python.....	17
--------------------------------------	----





LISTA DE ABREVIATURAS E SIGLAS

ERP -	Planejamento de Recursos Empresariais
RPA -	Automatização Robotica de Processos
PIP -	Gerenciador de Pacotes do Python
PyPI -	Índice de Pacotes Python
JBS -	José Batista Sobrinho
Zip -	Arquivo Compactado





SUMÁRIO

1 INTRODUÇÃO	13
2 FERRAMENTAS PARA DESENVOLVIMENTO DO PROJETO	14
2.1 RPA (ROBOTIC PROCESS AUTOMATION)	14
2.3 LINGUAGEM PYTHON	15
2.3.1 Pip - (Gerenciador de Pacotes do Python)	16
2.2.2 Pyautogui	18
2.2.3 Selenium	19
2.2.4 Pandas	19
2.2.5 Logging.....	20
3 ESTUDO DE CASO	21
3.1 METODOLOGIA.....	21
3.2 ROBÔ.....	23
3.2.1 Login Corporate.....	23
3.2.2 Relatório Razão Contábil Detalhado	26
3.2.3 Relatório Razão de Frete Consolidado.....	31
3.2.4 Junção de Dados Razão de Frete e Base Segmento	35
3.2.5 Cria Arquivo Zip.....	37
3.2.6 Upload do arquivo Zip na Intranet	38
4 CONCLUSÕES	41
REFERÊNCIAS BIBLIOGRÁFICAS:.....	43
REPOSITÓRIO DAS BIBLIOTECAS:	43





1 INTRODUÇÃO

Nas últimas décadas, tem havido um grande progresso na área da tecnologia, que tem se estendido para muitas empresas de diferentes setores da indústria. A cada dia, a evolução tecnológica impulsiona o desempenho das empresas, trazendo benefícios notáveis. Por exemplo, a introdução de robôs na montagem de carros foi uma inovação significativa no setor automotivo, aumentando a eficiência das concessionárias e automatizando ainda mais a indústria. A partir desta premissa, a automação tem sido aplicada para a realização de tarefas repetitivas em ambientes administrativos para a gestão de processos empresariais.

A automação robótica de processos, também conhecida pela sigla RPA (Robotic Process Automation), é uma tecnologia que tem como objetivo automatizar processos de negócios, através da configuração e gerenciamento de robôs de software que atuam como usuários de aplicativos sintéticos. Esses robôs são capazes de automatizar tarefas operacionais repetitivas e estruturadas nos sistemas da organização, proporcionando ganhos significativos de eficiência e produtividade (QUALITAT, 2020). Os bots, como também são conhecidos, têm demonstrado ser uma solução valiosa em diversas áreas, incluindo finanças, direito, tecnologia da informação, turismo, educação, entre outras (CASTRO, 2020).

A implementação da automação robótica de processos pode trazer benefícios significativos, como a redução de custos e tempo, diminuição de riscos e aumento da flexibilidade nas operações. Com a automação de tarefas antes executadas pelos funcionários, estes podem dedicar mais tempo a atividades estratégicas que requerem análises mais elaboradas e planejamento. A eliminação de possíveis equívocos gerados por fadiga e falta de atenção humana é outra vantagem dos robôs programados. Além disso, a flexibilidade da tecnologia permite que o robô seja implementado em diferentes escalas, com a possibilidade de replicação em várias máquinas e execução de tarefas com diferentes dados de forma contínua, sem interrupções, acelerando assim os processos desejados.

Nesse contexto, este trabalho de graduação tem como objetivo a criação de um robô de software construído com a linguagem de programação Python para ser implantado na empresa JBS Transportadora, realizando o serviço de extração de dados para 4 segmentos por meio da geração dos relatórios “Razão Contábil Detalhado” e “Razão de Frete Consolidado” no sistema ERP Corporate. De maneira estratégica, o robô extrai cada relatório de acordo com os parâmetros já usados no





dia a dia de cada segmento, várias vezes ao dia, possibilitando que a operação tenha sempre dados atualizados para gestão financeira, patrimonial e econômica da empresa.

2 FERRAMENTAS PARA DESENVOLVIMENTO DO PROJETO

Neste capítulo, serão abordados os tópicos fundamentais para a compreensão deste projeto, ou seja, estabelecendo as significativas considerações sobre o processo de geração de relatórios no ERP Corporate, as tecnologias empregadas no trabalho e uma explicação mais detalhada sobre o que representa o RPA.

2.1 RPA (ROBOTIC PROCESS AUTOMATION)

A RPA, automatização robótica de processos é um sistema cuja sua funcionalidade é operar da mesma forma que um humano, o sistema é visto como uma forma de otimizar processos repetitivos, reduzindo o peso de tarefas simples para os trabalhadores (*Van Der Aalst et al., 2018*).

Para melhor definição, a automatização de processos robóticos é baseada em regras de software que não estão ativos, chamados *BOT* que podem imitar as ações feitas por um ser humano em computadores de forma mais ágil, segura e com maior massa de serviços (*Madakam et al., 2019*).

Com a evolução da tecnologia, para (*Madakam et al., 2019*) as empresas que implementarem a RPA desempenhará maior produtividade aos funcionários, e muito provável de que os robôs venham a ter maior eficácia, com até 45% dos postos de trabalho nas indústrias até o ano de 2025.

Gráfico de Impacto das tecnologias nos empregos do mundo até 2025 (em milhões).





Gráfico 2.1 - Evolução da RPA



Fonte: Gartner, 2017. Adaptado pelos autores.

Segundo *Graglia et al. (2018)*, a Inteligência criará mais empregos do que eliminará, um ganho de 2,3 milhões de empregos até o ano 2020, enquanto eliminará 1,8 milhões de empregos. Com o crescente aumento de tecnologias nas empresas, muitas inovações caracterizavam um período de transição, de perdas de empregos a adaptação e recuperação.

Com a tecnologia em uso, melhoramos a produtividade de muitas organizações, eliminando milhões de posições de médio e baixo nível, assim, criando milhões de médio e alto nível (*Graglia et al., 2018*).

2.3 LINGUAGEM PYTHON

Python é uma linguagem rápida, poderosa, simples e de código aberto, que suporta várias tecnologias. Sua estrutura de dados é de fácil legibilidade, tornando-a de fácil compreensão e muito próxima a linguagem humana, ou seja, de alto nível (*SRINATH, 2017*).

Assim como outras linguagens de programação, a linguagem Python conta com bibliotecas que auxiliam os desenvolvedores a ganharem agilidade na implementação de projeto. A utilização de bibliotecas no Python pode ser realizada por meio do sistema de gerenciamento de pacotes padrão Preferred Installer Program,





popularmente conhecido como pip. Deste modo, a instalação de uma biblioteca/módulo/pacote Python pode ser realizada por meio do comando “pip install nome-do-pacote”.

2.3.1 Pip - (Gerenciador de Pacotes do Python)

O comando "pip install" é uma ferramenta fundamental no ecossistema Python que permite instalar pacotes do Python e de terceiros, esses pacotes são conhecidos como bibliotecas no ambiente Python. O "pip" é o gerenciador de pacotes padrão para Python e facilita o processo de obtenção e instalação de pacotes de código aberto disponíveis no repositório do Python Package Index (PyPI) ou Índice de pacote Python (GIBBS,2021).

GIBBS (2021) afirma, que uma vez utilizado o “pip install nome-do-pacote” essas bibliotecas são armazenadas no computador e com isso, é possível importar a biblioteca no código e utilizar as suas funções/métodos disponíveis. Vale ressaltar que é possível importar mais de uma biblioteca no código.

A linguagem Python é diversificada e possui uma grande comunidade de desenvolvedores que criam bibliotecas e ferramentas para ajudar a tornar o desenvolvimento de software mais rápido e eficiente.





Tabela 2.1 - Bibliotecas Python

Nome da biblioteca	Descrição
pyautogui	Permite automatizar tarefas repetitivas.
pymsgbox	Exibe caixas de diálogo, facilitando a interação com o usuário.
datetime	É um modulo que permite alterar datas e horas.
logging	Informa sobre as depurações do código (avisos e erros).
os	Faz a interação com o sistema operacional nativo.
subprocess	Permite iniciar e controlar processos externos pelo programa.
pandas	Realiza análise e manipulação de dados tabulares, como planilhas e bancos de dados.
time	Oferece funcionalidades relacionadas ao tempo e à medição de tempo.
calendar	Fornece uma variedade de funções para facilitar a manipulação de datas e calendários.
shutil	Permite copiar, mover, renomear e excluir arquivos e diretórios.
Chrome	Realiza testes em aplicativos da web, faz raspagem de dados da web pelo navegador Chrome.
webdriver	Controlar um navegador de forma programática





A Tabela exibe de maneira sucinta as bibliotecas utilizadas neste projeto. As Subseções: Pyautogui, Selenium, Pandas e Logging abordam com mais detalhes as bibliotecas consideradas cruciais para o projeto.

Dentre as bibliotecas supracitadas, pode-se destacar, Pyautogui, Selenium, Pandas e Logging o qual são abordadas com mais detalhes nas subseções.

2.2.2 Pyautogui

Pyautogui é uma biblioteca Python que permite automatizar ações de mouse e teclado. Deste modo, a utilização da biblioteca possibilita que sejam criados scripts Python para executar tarefas repetitivas, tais como: clicar em botões, digitar textos, mover o mouse, pressionar teclas e até mesmo tirar screenshots (Sweigart, v. 25 2020).

Cibi (2022) ressalta atenção na resolução da tela utilizada, para que o mapeamento das ações do mouse guiada pelas coordenadas de x e y dos pixels; um exemplo é que um *BOT* com a resolução de 1080p, só irá funcionar com maior precisão nas telas semelhantes a esta resolução.

A biblioteca é útil para automatizar tarefas que geralmente são executadas manualmente, como preenchimento de formulários ou realização de testes de software, tornando os processos dos utilizadores mais fácil e ágil (Cibi, et al 2022).

Para começar a usar o Pyautogui, é preciso instalá-lo usando uma ferramenta de gerenciamento de pacotes como o pip,

```
pip install pyautogui
```

Em seguida, importar a biblioteca em seus scripts Python

```
import pyautogui
```





2.2.3 Selenium

O Selenium é uma biblioteca de código aberto muito utilizada em linguagens Java, JavaScript, C#, Python e Ruby para realizar testes em navegadores. GARCIA (2022), também diz que a biblioteca é amplamente usada para automações de teste e interações com os navegadores web. Com o Selenium, é possível controlar um navegador da web automaticamente, simular a interação humana em sites e executar testes automatizados em aplicativos da web. O Selenium é particularmente útil para realizar testes de regressão, automatizar tarefas repetitivas na web e extrair informações de páginas da web (GARCIA, Boni, 2022).

Para NYAMATHULLA (2021), o Selenium pode se comunicar com diversos navegadores da web, incluindo Google Chrome, Mozilla Firefox, Microsoft Edge, entre outros. Isso é possível através de um driver específico para cada navegador que serve como uma conexão entre o código Python e o navegador real.

Para começar a usar o Selenium, é preciso instalá-lo usando uma ferramenta de gerenciamento de pacotes como o pip,

```
pip install Selenium
```

Em seguida, importar a biblioteca em seus scripts python,

```
From selenium import webdriver
```

2.2.4 Pandas

O Pandas é uma das bibliotecas mais adoradas em Python quando se trata de análise e manipulação de dados (CHEN,2018). É uma ferramenta fácil de usar e eficiente para lidar com dados tabulares, como tabelas, planilhas e bancos de dados. O Pandas é amplamente utilizado em campos como ciência de dados, análise de dados, engenharia de dados e áreas relacionadas.

CHEN (2018), aborda que o Pandas é altamente otimizado para lidar com grandes conjuntos de dados, garantindo um desempenho excepcional. Muitas vezes, ele é combinado com outras bibliotecas populares do Python, como NumPy (para





operações numéricas), Matplotlib (para visualização) e Scikit-Learn (para aprendizado de máquina).

Para começar a usar o Pandas, é preciso instalá-lo usando uma ferramenta de gerenciamento de pacotes como o pip,

```
pip install pandas
```

Após a importação da Pandas, é possível criar DataFrames, ler dados de arquivos, executar operações de análise de dados, tornando o Pandas, uma ferramenta essencial para qualquer pessoa envolvida em análise de dados usando Python (CHEN,2018).

2.2.5 Logging

A biblioteca logging do Python desempenha um papel essencial na implementação bem-sucedida de Automação de Processos Robóticos (RPA), proporcionando uma estrutura robusta para o registro de mensagens durante a execução de scripts e automações (GIRIDHAR, 2019). Essa biblioteca é particularmente valiosa no contexto de RPA, onde a capacidade de rastrear e depurar eventos é crucial para garantir o desempenho adequado dos *bots*.

No âmbito específico de RPA, a biblioteca logging permite configurar diferentes níveis de severidade para as mensagens registradas, proporcionando flexibilidade no controle da quantidade e importância das informações a serem capturadas durante a execução dos *bots*. A capacidade de definir manipuladores (handlers) também permite direcionar essas mensagens para diferentes destinos, como consoles, arquivos de log ou serviços remotos, oferecendo uma abordagem versátil para o gerenciamento de logs em ambientes de automação.

Para começar a usar o Logging, é preciso instalá-lo usando uma ferramenta de gerenciamento de pacotes como o pip,

```
pip install logging
```





A flexibilidade e a eficácia da biblioteca logging são fundamentais para a manutenção e o monitoramento de scripts RPA, tornando-a uma escolha vantajosa na implementação de automações que exigem um controle preciso sobre eventos e erros.

3 ESTUDO DE CASO

A JBS Transportadora está implementando o uso de um robô para a extração de relatórios contábeis e de frete, com o objetivo de otimizar e acelerar o processo de análise contábil dos fretes realizados pelos segmentos da organização.

3.1 METODOLOGIA

Este projeto foi desenvolvido em três fases: mapeamento de contexto, avaliação de mapeamento, desenvolvimento e avaliação. Com isso, este trabalho possui como ênfase o processo de desenvolvimento do *bot* e a análise dos resultados obtidos a partir da implementação e estudo dos resultados para convertê-los em informações que contribuam para o aprimoramento constante do robô.

A fase inicial de mapeamento do contexto envolveu uma compreensão aprofundada do cenário atual dos controllers nos segmentos Boiadeiro, Frigorífica, Container Seara e Seara Agro da JBS Transportadora. Durante essa etapa, foram conduzidos levantamentos junto às respectivas áreas, visando abordar os seguintes pontos essenciais: a) o processo manual de geração de relatórios nas telas de Razão Contábil Detalhado e Razão de Frete Consolidado, adaptados às necessidades específicas de cada segmento; b) a duração média para execução do processo; c) a frequência diária de extração de dados; d) os horários específicos em que ocorrem essas extrações.

Com base nos filtros obtidos nas telas, foi constatado que a geração do relatório de Razão Contábil Detalhado demanda aproximadamente 5 minutos, enquanto o relatório de Razão de Frete Consolidado requer cerca de 8 minutos. Os colaboradores costumam realizar essas extrações três vezes ao dia, nos horários de 7h, 10h e 15h.

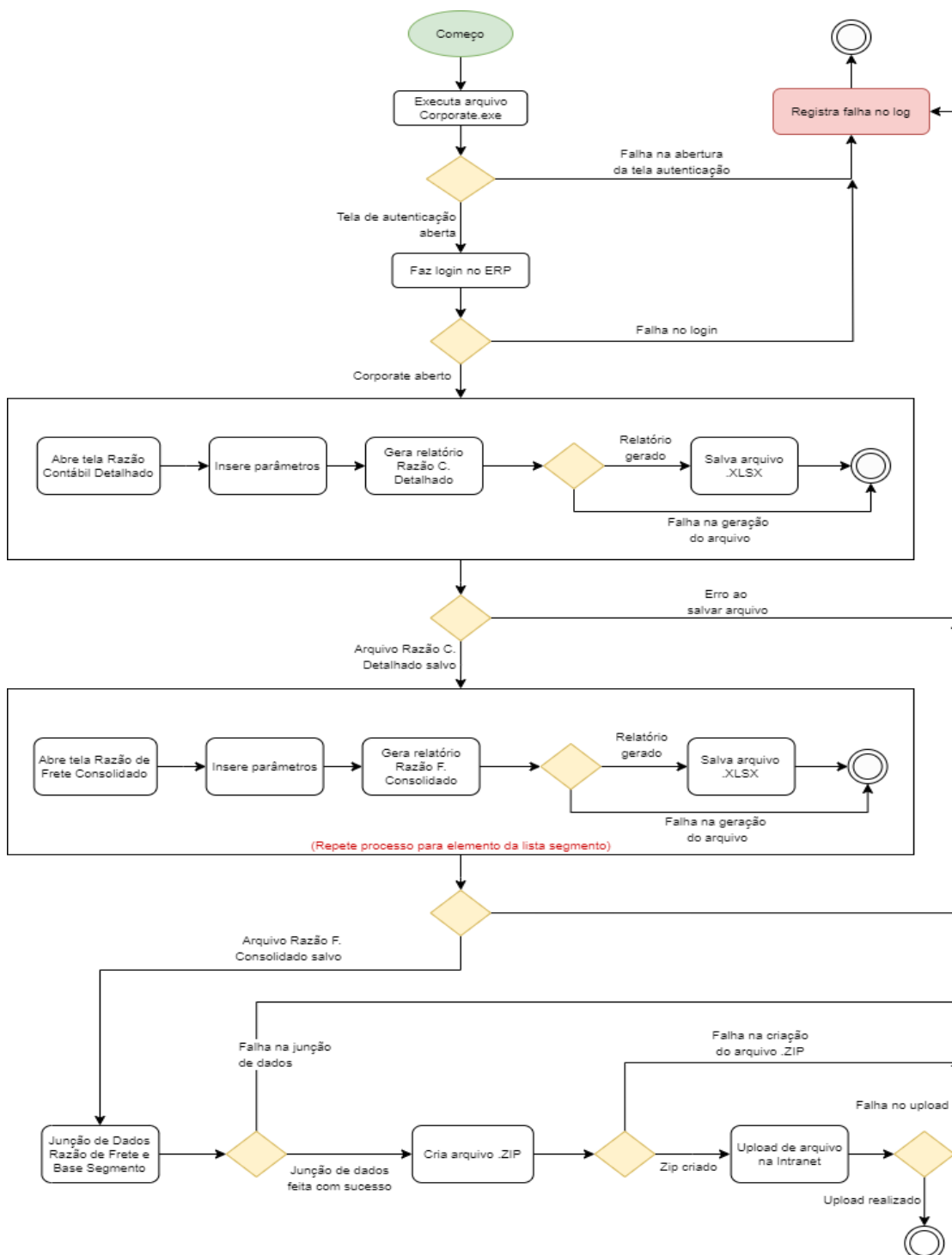
Prosseguindo no fluxo do projeto, foi conduzido um estudo detalhado do mapeamento anteriormente realizado, essencial para iniciar o desenvolvimento do plano de execução que o RPA seguiria para gerar cada relatório no ERP.





Cabe destacar que, para a eficiência da tecnologia RPA, é fundamental estabelecer um alinhamento sólido no plano de execução, a fim de evitar perdas de tempo e otimizar o desempenho do robô. Nesse contexto, a metodologia e a estratégia delineadas estão claramente definidas no diagrama apresentado abaixo:

Grafico 3.1 - Diagrama



Fonte: Elaborado pelos autores, 2023





3.2 ROBÔ

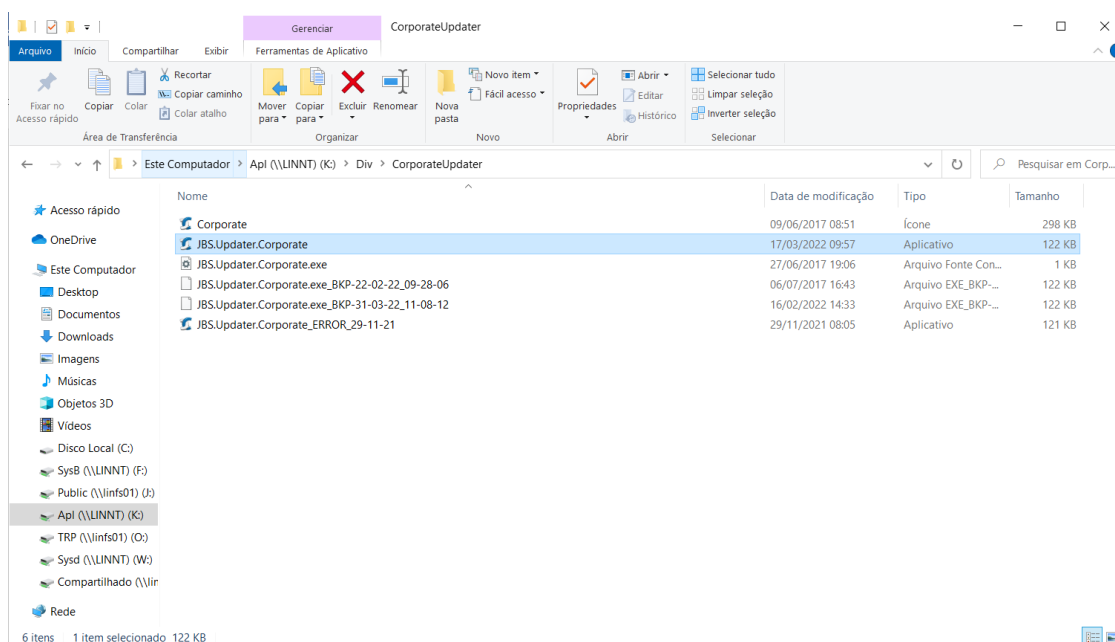
Nesta subseção, será apresentada de modo pontual, os importantes passos determinados para a composição do robô.

3.2.1 Login Corporate

Como mencionado no capítulo anterior, um processo de RPA deve operar de maneira semelhante à forma como um humano executaria a atividade em questão. Portanto, ao rodar o script do robô, a primeira ação programada consiste em abrir o arquivo "JBS.Updater.Corporate.exe" (Figura 1), o executável correspondente ao ERP Corporate contido no disco K:\, como especificado na linha de código a seguir:

os.startfile ('K:\Div\CorporateUpdater\JBS.Updater.Corporate.exe')

Figura 3.1 – JBS.Updater.Corporate



Fonte: Elaborada pelos autores, 2023.

Após a execução, espera-se que a tela de login do sistema (Figura 2) seja aberta, possibilitando assim a inserção das credenciais de acesso.





Figura 3.2 - Tela de Autenticação do sistema Corporate

Autenticação

Corporate

Usuário

Senha

Base de dados

Matriz

[Esqueci minha senha](#)

Acessar Sair

Corporate 8.710.00 - 16/11/2023 08:00:00

Fonte: Elaborada pelos autores, 2023.

Neste momento, é crucial verificar se a janela de "Autenticação" foi efetivamente aberta ou se ocorreu algum problema durante a execução. Para essa verificação, o robô aguarda até 180 segundos (3 minutos) para que a janela com o título "Autenticação" seja aberta. Caso isso não ocorra dentro do prazo estipulado, o robô registra um erro no log e retorna True para indicar uma condição de falha, resultando no encerramento do processo devido a um erro crítico.

```
#verifica se tela de login abriu
timeout = time.time() + 180
while teste_abertura.title != 'Autenticação' and time.time() < timeout:
    teste_abertura = pg.getActiveWindow()
    pg.sleep(1)
if teste_abertura.title != 'Autenticação':
    taskkill = True
    motivo = 'Bot não conseguiu abrir tela login'
    logging.error(prefixo_log + motivo)
    return taskkill
```

Antecipando um cenário em que o processo ocorreu corretamente e a janela foi aberta, o *bot* prossegue digitando as credenciais de acesso em cada um dos campos, utilizando a biblioteca PyAutoGUI para simular as ações de mouse e teclado.





Após preencher os dados de usuário e senha, o robô mantém o campo "base de dados" com o valor "matriz", garantindo que, ao gerar relatórios, os dados saiam completos, o que não aconteceria caso a base especificada fosse outra. Para concretizar esta etapa, foi utilizado o comando da tecla ENTER para acionar o botão "acessar".

Novamente, é necessário verificar se a última ação foi concluída com êxito ou se houve alguma instabilidade. O cenário ideal é que a tela inicial do sistema ERP (Figura 3) seja aberta após a finalização do login para dar continuidade ao processo. Para garantir isso, o código aguarda até que a cor do pixel na posição (12, 5) corresponda à cor especificada (7, 86, 157) ou até que o limite de tempo de 60 segundos seja alcançado.

```
#verifica se Corporate está aberto
```

```
timeout = time.time() + 60
```

```
while not pg.pixelMatchesColor(12,5,(7,86,157)) and time.time() < timeout:
```

```
    pg.sleep(1)
```

```
if not pg.pixelMatchesColor(12,5,(7,86,157)):
```

```
    taskkill = True
```

```
    motivo = 'Bot não conseguiu abrir tela inicial erp pós login'
```

```
    logging.error(prefixo_log + motivo)
```

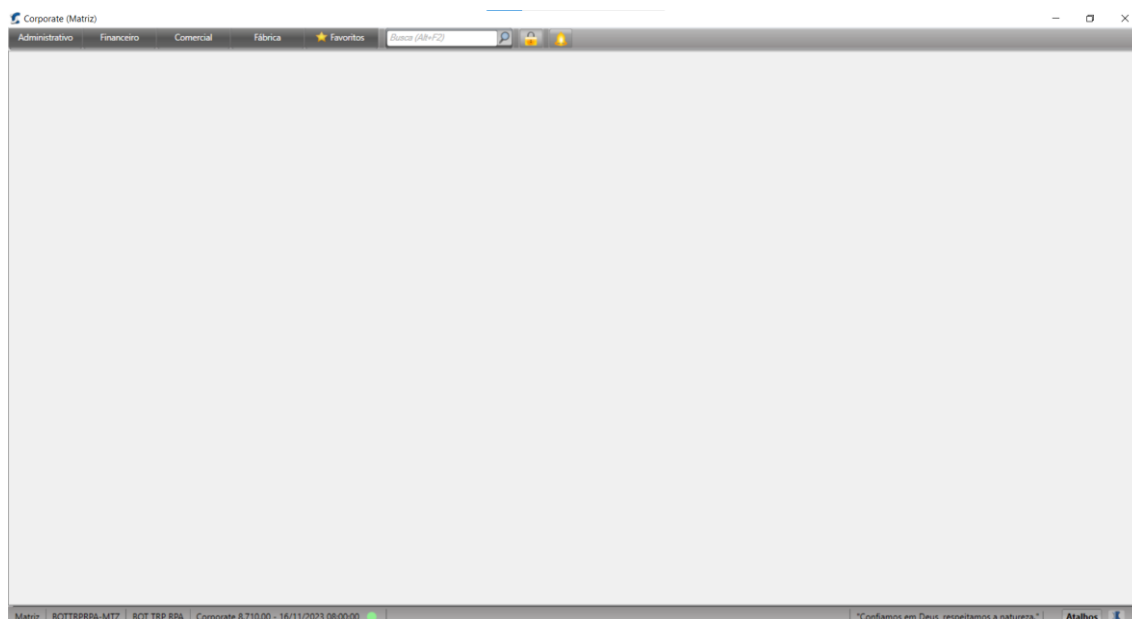
```
    return taskkill
```

Em termos mais simples, o código valida se a cor azul, correspondente ao símbolo do sistema Corporate no canto superior esquerdo da tela, é encontrada na coordenada especificada (12, 5).





Figura 3.3 - Tela inicial do ERP Corporate



Fonte: Elaborada pelos autores, 2023.

Se a condição não for atendida dentro do prazo de 1 minuto, é registra o erro no log e retorna True, indicando uma falha crítica de processo.

3.2.2 Relatório Razão Contábil Detalhado

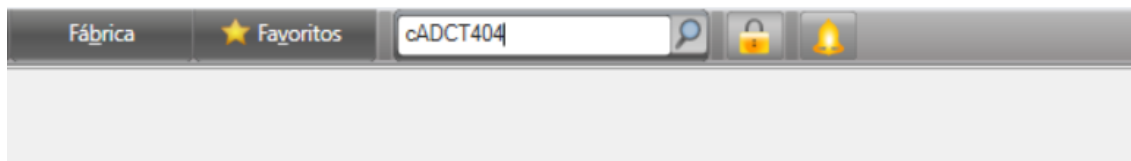
Nesta subseção, apresentaremos o processo de extração do primeiro relatório, o Razão Contábil Detalhado. Uma característica distintiva do Corporate é a atribuição de um código único a cada tela operacional do sistema. Ao inserir esse código na barra de pesquisa e pressionar a tecla "ENTER", a tela desejada é imediatamente exibida, proporcionando uma eficiente abordagem para a realização da tarefa. É dessa funcionalidade que o robô tira proveito. No caso do relatório Razão Contábil Detalhado, o código correspondente à esta tela é o cADCT404.

Mais uma vez, utilizando a biblioteca PyAutoGUI do Python, o robô simula a ação de digitar esse código na barra de pesquisa (Figura 4) e realiza o acionamento da tecla "ENTER". Esse processo automatizado agiliza a execução da tarefa de extração do relatório.





Figura 3.4 – Código da tela Razão Contábil



Fonte: Elaborada pelos autores, 2023.

Após a abertura da tela, o passo subsequente envolve a inserção dos parâmetros essenciais para a geração dos dados. O PyAutoGUI é empregado para automatizar a digitação, permitindo que o robô preencha os campos de data inicial, data final, resumido, estorno, grupo de empresa e código da conta contábil. É relevante observar que, após cada inserção de valor em um campo, a tecla TAB é acionada, facilitando a navegação para o próximo campo, sendo este processo repetido até o último parâmetro.

Figura 3.5 – Tela Razão Contábil Detalhado

Código	Número	Nome

Código	Número	Nome
3	3	RESULTADO LIQUIDO

Fonte: Elaborada pelos autores, 2023.

Para os campos "data inicial" e "data final", o robô preenche com as informações do primeiro e último dia do mês atual, respectivamente. O processo utiliza a função `datetime.now()` para obter o mês e ano atuais, seguido pelo módulo `calendar`





para determinar o último dia do mês corrente. Após armazenar cada informação em variáveis distintas e convertê-las para o tipo string, os valores são concatenados para ficarem no formato de data.

#obtenção de primeiro e último dia do mês

```
mes = datetime.now().month  
ano = datetime.now().year  
dia_max = calendar.monthrange(2023, mes)[1]  
ano = str(ano)  
mes = str(mes)  
dia_max = str(dia_max)  
primeiro_dia = '01' + mes + ano  
ultimo_dia = dia_max + mes + ano
```

Os campos "resumido" e "estorno" são ambos preenchidos com o texto "0 - não", enquanto nos campos "grupo de empresa" e "código" no grid Conta Contábil, é crucial inserir os valores 7 e 3, respectivamente, conforme representado na Figura 5. É crucial que nenhum código seja informado na grid Centro de Custo, desta forma é garantido que os dados contábeis de todos os segmentos venham em um único relatório, proporcionando uma significativa economia de tempo no processo. Além disso, assegura a precisão das informações, uma vez que cada controller pode acessar o mesmo arquivo excel e aplicar um filtro para encontrar os dados específicos de sua operação.

A próxima ação do robô é acionar a tecla F3, atalho do botão "executar", iniciando assim o processo de geração do relatório Razão Contábil Detalhado pelo ERP.

Nesta etapa, o script aguarda por 1000 segundos até que o relatório esteja pronto. O tempo prolongado é necessário, especialmente considerando que, conforme o mês se aproxima do fim mais dados contábeis são apurados e o relatório fica cada vez mais extenso, consequentemente o Corporate leva mais tempo para carregar esses dados.

Quando o ERP não consegue gerar o relatório, seja devido a um parâmetro digitado incorretamente ou a qualquer outra razão, o resultado é uma tela em branco com uma aba intitulada "sem dados". Em contrapartida, quando o relatório é gerado





corretamente, os dados são apresentados, e a aba é rotulada como "Resultado 1" (Figura 6).

Figura 3.6 – Relatório de Razão Contábil Detalhado

Fonte: Elaborada pelos autores, 2023.

Para avaliar o êxito ou insucesso da busca, o código analisa dois cenários: a) se o resultado é "sem dados" e b) se o símbolo da âncora azul está visível na tela (figura 6), indicando a carga bem-sucedida do relatório. Para realizar essa verificação, foi implementado o código abaixo que opera dentro do limite temporal de 1000 segundos. Durante esse período, a cor da âncora é continuamente examinada na posição específica da tela (x=347, y=148). Se a âncora não for localizada, o código procura na tela por uma imagem correspondente à aba "sem dados" em um arquivo .png. Caso seja identificado o resultado "sem dados" ou se nenhuma das condições for verdadeira, o robô pressiona a tecla F8 para fechar a tela do Razão C. Detalhado e registra um erro no log, indicando que o *bot* não conseguiu gerar o relatório. A variável `taskkill` é, então, retornada para encerrar o processo.

#aguarda gerar relatório. Verifica se resultado foi "sem dados" enquanto aguarda

```
timeout = time.time() + 1000
```





```
while not pg.pixelMatchesColor(347,148, (169,210,117)) and time.time() <
timeout:
```

```
pg.sleep(4)
```

```
semdados = pg.locateOnScreen(r'../imagens/semdados.png')
```

```
if semdados:
```

```
pg.press('f8')
```

```
motivo = 'Sem dados para consulta'
```

```
logging.info(prefixo_log + motivo)
```

```
return taskkill
```

```
if not pg.pixelMatchesColor(347,148, (169,210,117)):
```

```
taskkill = True
```

```
pg.press('f8')
```

```
motivo = 'bot não conseguiu gerar relatório'
```

```
logging.error(prefixo_log + motivo)
```

```
return taskkill
```

Não apresentando erros, o robô gera de imediato o arquivo de extensão .xlsx com o auxílio das ações do mouse e do teclado, e com nome ajustado para “R. C. Detalhado.xlsx”. O arquivo é salvo em uma pasta do projeto específica para armazenar os relatórios do projeto.

Figura 3.7 - “Relatório Razão Contábil Detalhado” para download

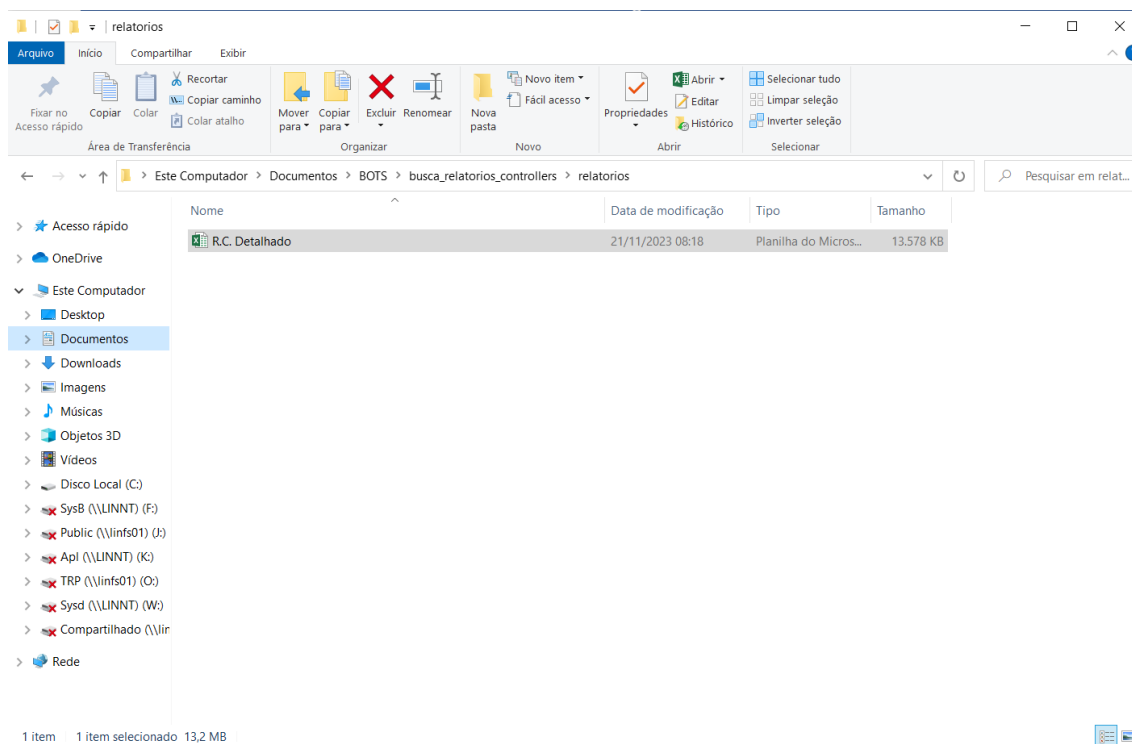
Centro de Custo		Conta Contábil		Lançamento		Tercero	
Código	Nome	Código	Nome	Código	Saldo	Código	Nome
1193	VEICULO SEQ-4677	21799	320.100.200000000...	151194	1402,59	1402,59	0,00
1193	VEICULO SEQ-4678	21799	320.100.200000000...	151194	491,56	491,56	0,00
1193	VEICULO SEQ-4679	21799	320.100.200000000...	151194	1883,70	1883,70	0,00
1193	VEICULO SEQ-4680	21799	320.100.200000000...	151194	1331,61	1331,61	0,00
1193	VEICULO SEQ-4681	21799	320.100.200000000...	151194	1306,00	1306,00	0,00
1193	VEICULO SEQ-4682	21799	320.100.200000000...	151194	2009,23	2009,23	0,00
1193	VEICULO SEQ-4683	21799	320.100.200000000...	151194	2511,53	2511,53	0,00
1193	VEICULO SEQ-4684	21799	320.100.200000000...	151194	299,32	299,32	0,00
1193	VEICULO SEQ-4685	21799	320.100.200000000...	151194	1333,88	1333,88	0,00
1193	VEICULO SEQ-4686	21799	320.100.200000000...	151194	1707,89	1707,89	0,00
1193	VEICULO SEQ-4687	21799	320.100.200000000...	151194	2335,78	2335,78	0,00
1193	VEICULO SEQ-4688	21799	320.100.200000000...	151194	1808,30	1808,30	0,00
1193	VEICULO SEQ-4689	21799	320.100.200000000...	151194	1813,48	1813,48	0,00
1193	VEICULO SEQ-4690	21799	320.100.200000000...	151194	1813,38	1813,38	0,00
1193	VEICULO SEQ-4691	21799	320.100.200000000...	151194	2084,87	2084,87	0,00
1193	VEICULO SEQ-4692	21799	320.100.200000000...	151194	2235,36	2235,36	0,00
1193	VEICULO SEQ-4693	21799	320.100.200000000...	151194	2009,28	2009,28	0,00
1193	VEICULO SEQ-4694	21799	320.100.200000000...	151194	1738,03	1738,03	0,00
1193	VEICULO SEQ-4695	21799	320.100.200000000...	151194	2310,66	2310,66	0,00
1193	VEICULO SEQ-4696	21799	320.100.200000000...	151194	1707,84	1707,84	0,00
1193	VEICULO SEQ-4697	21799	320.100.200000000...	151194	1808,30	1808,30	0,00
1193	VEICULO SEQ-4698	21799	320.100.200000000...	151194	2586,93	2586,93	0,00
1193	VEICULO SEQ-4699	21799	320.100.200000000...	151194	130,75	130,75	0,00

Fonte: Elaborada pelos autores, 2023.





Figura 3.8 – Arquivo R. C. Detalhado.xlsx contido na pasta Relatorios



Fonte: Elaborada pelos autores, 2023.

3.2.3 Relatório Razão de Frete Consolidado

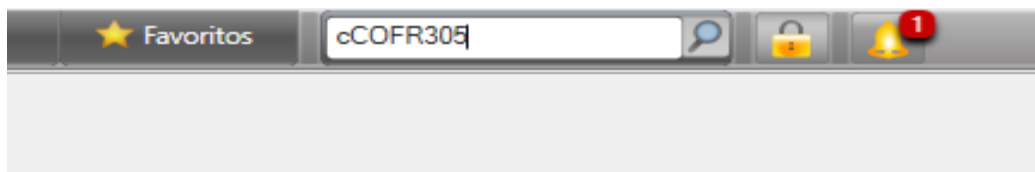
O próximo passo consiste na geração do arquivo do Razão de Frete Consolidado. Para iniciar esta etapa do projeto, pressiona-se a tecla F8 no teclado para fechar a tela atual e manter a janela limpa, como ilustrado anteriormente na Figura 3, antes de abrir a próxima tela. É crucial manter um ambiente organizado, evitando a abertura simultânea de várias telas para prevenir possíveis desorientações do robô durante a execução da atividade e para otimizar seu desempenho.

Após esse procedimento, repete-se a inserção do código da tela na barra de pesquisa. Para este relatório específico, o código a ser fornecido é o cCOFR305.





Figura 3.9 – Código da tela Razão de Frete



Fonte: Elaborada pelos autores, 2023.

Após inserir o código e simular a tecla ENTER para confirmar a pesquisa, a tela do Razão de Frete Consolidado é aberta. Apesar de ser uma tela distinta com parâmetros diferentes, muitas das ações executadas no último subcapítulo se repetem neste cenário.

Durante o mapeamento do contexto, observou-se que, ao contrário do Razão C. Detalhado, não é possível gerar um único arquivo do Razão de Frete Consolidado para os 4 segmentos clientes. Assim, a solução adotada foi utilizar a função criada em Python para executar todo o processo de extração desse relatório múltiplas vezes, ajustando apenas os parâmetros específicos para cada segmento. No total, a função é executada 8 vezes, dado que, conforme mencionado anteriormente, é necessário extrair 4 relatórios para o segmento Container Seara, um para cada código de terceiro.

Para ilustrar o processo de extração do relatório de Frete, será utilizado os filtros do segmento Seara Agro da JBS Transportadora apenas para exemplificar o processo.





Figura 3.10 – Tela Razão de Frete Consolidado gerado corretamente

Corporate (Matriz)

Administrativo | Financeiro | Comercial | Fábrica | Favoritos | Busca (Alt+F2)

Razão de Frete Consolidado

F3 Executar | F8 Finalizar | F11 Maximizar

Parâmetros

Razão de Frete Consolidado

Data Inicial: 01/11/2023 a 31/01/2023

Empresa: []

Grupo Empresa: 7 - DIVISÃO TRANSPORTES

Mapa: 106 Synchro - Documento Fiscal

Evento: []

Terceiro: []

Instrução: []

Sub-Contratada: 0-Não Tipo de Relatório: Matriz

Código	Nome
588 S - CTe - Emissor Integrado - TRP - Primário - Rodoviário	
715 S - TRP - CTe - Rodoviário - Complemento de ICMS	

Matriz | BOTTRPRPA-MTZ | BOT TRP RPA | Corporate 8.710.00 - 16/11/2023 08:00:00 | "Confiamos em Deus, respeitamos a natureza." | Atalhos

Fonte: Elaborada pelos autores, 2023.

A Figura 3.10 apresenta o instante em que são exibidos os parâmetros necessários para a geração do relatório em foco. Novamente com a biblioteca PyAutoGUI, o *bot* automatiza a inserção de dados nos campos "Data Inicial", "a" (data final), "Grupo de Empresa", "Mapa" e "Código" na gride "Tipo de Documento". A ilustração acima demonstra como o layout ficaria após a devida inserção dos filtros da Seara Agro.

Assim como feita na tela de Razão Contábil Detalhado, a tecla F3 é acionada para começar a gerar os dados e repete-se a validação do resultado, se o resultado será sem dados ou se o robô encontrará o ícone azul da ancora que aparece quando o relatório é gerado.





Figura 3.11 – Relatório de Razão de Frete Consolidado

Sigla Empresa	Data Emissão	Última aprov. NFE	Evento	Nome Evento	Veículo	Placa	Tipo Veículo	Nome Tipo Veículo	Instrução	Nome Instrução	Instrução Referência	Cidade Destino	UF Destino	Terceiro
CPG	02/10/2023		2110	Frete - Valor do Frete	417894	HQR9H28	1	PG - Truck até 11,50m (18 cab)				BRASILANDIA	MS	3774617 3M TRANSPORT
CPG	02/10/2023		2072	Frete - Desconto Lavagem	417894	HQR9H28	1	PG - Truck até 11,50m (18 cab)				BRASILANDIA	MS	3774617 3M TRANSPORT
ITA	02/10/2023		2072	Frete - Desconto Lavagem	341629	LPF7129	1	PG - Truck até 11,50m (18 cab)				BOM JESUS DA LAPA	BA	3650700 A M COMERCIO
ITA	02/10/2023		2110	Frete - Valor do Frete	341629	LPF7129	1	PG - Truck até 11,50m (18 cab)				BOM JESUS DA LAPA	BA	3650700 A M COMERCIO
ITA	02/10/2023		2110	Frete - Valor do Frete	342279	MWLD92	1	PG - Truck até 11,50m (18 cab)				BOM JESUS DA LAPA	BA	3650700 A M COMERCIO
ITA	02/10/2023		2072	Frete - Desconto Lavagem	342279	MWLD92	1	PG - Truck até 11,50m (18 cab)				BOM JESUS DA LAPA	BA	3650700 A M COMERCIO
ITA	02/10/2023		2072	Frete - Desconto Lavagem	316235	OKL8561	1	PG - Truck até 11,50m (18 cab)				BOM JESUS DA LAPA	BA	3650700 A M COMERCIO
ITA	02/10/2023		2110	Frete - Valor do Frete	316235	OKL8561	1	PG - Truck até 11,50m (18 cab)				BOM JESUS DA LAPA	BA	3650700 A M COMERCIO
ITA	02/10/2023		2072	Frete - Desconto Lavagem	341966	LR3A11	1	PG - Truck até 11,50m (18 cab)				ILUBU	BA	3650700 A M COMERCIO
ITGV	02/10/2023		1043	FG TRP - Gado Magro (Valor Serviço)	352711	FOASE26	237	PG - Carreta Gado 2 Pisos (54 cab)				CRIXAS	GO	62692 ADEBIAL RANIC
ITGV	02/10/2023		1044	FG TRP - Gado Magro (Outros)	352711	FOASE26	237	PG - Carreta Gado 2 Pisos (54 cab)				CRIXAS	GO	62692 ADEBIAL RANIC
ITGV	02/10/2023		1043	FG TRP - Gado Magro (Valor Serviço)	352539	BOC6A51	237	PG - Carreta Gado 2 Pisos (54 cab)				CRIXAS	GO	62692 ADEBIAL RANIC
ITGV	02/10/2023		1044	FG TRP - Gado Magro (Outros)	352539	BOC6A51	237	PG - Carreta Gado 2 Pisos (54 cab)				CRIXAS	GO	62692 ADEBIAL RANIC
ITGV	02/10/2023		1043	FG TRP - Gado Magro (Valor Serviço)	331565	CU11357	237	PG - Carreta Gado 2 Pisos (54 cab)				CRIXAS	GO	62692 ADEBIAL RANIC
ITGV	02/10/2023		1044	FG TRP - Gado Magro (Outros)	331565	CU11357	237	PG - Carreta Gado 2 Pisos (54 cab)				CRIXAS	GO	62692 ADEBIAL RANIC
ITGV	02/10/2023		1044	FG TRP - Gado Magro (Outros)	349298	CUH9H98	237	PG - Carreta Gado 2 Pisos (54 cab)				CRIXAS	GO	62692 ADEBIAL RANIC
ITGV	02/10/2023		1043	FG TRP - Gado Magro (Valor Serviço)	349298	CUH9H98	237	PG - Carreta Gado 2 Pisos (54 cab)				CRIXAS	GO	62692 ADEBIAL RANIC
ITGV	02/10/2023		1043	FG TRP - Gado Magro (Valor Serviço)	307913	EB06889	237	PG - Carreta Gado 2 Pisos (54 cab)				CRIXAS	GO	62692 ADEBIAL RANIC
ITGV	02/10/2023		1044	FG TRP - Gado Magro (Outros)	307913	EB06889	237	PG - Carreta Gado 2 Pisos (54 cab)				CRIXAS	GO	62692 ADEBIAL RANIC
AGB	02/10/2023		2110	Frete - Valor do Frete	4194	LWT2384	1	PG - Truck até 11,50m (18 cab)				NOVO SAO JOAO...	MT	3178325 ALIANCA TRAN
AGB	02/10/2023		2072	Frete - Desconto Lavagem	4194	LWT2384	1	PG - Truck até 11,50m (18 cab)				NOVO SAO JOAO...	MT	3178325 ALIANCA TRAN
AGR	02/10/2023		2110	Frete - Valor do Frete	60351	DT11218	1	PG - Truck até 11,50m (18 cab)				CANARANA	MT	3178325 ALIANCA TRAN

Fonte: Elaborada pelos autores, 2023.

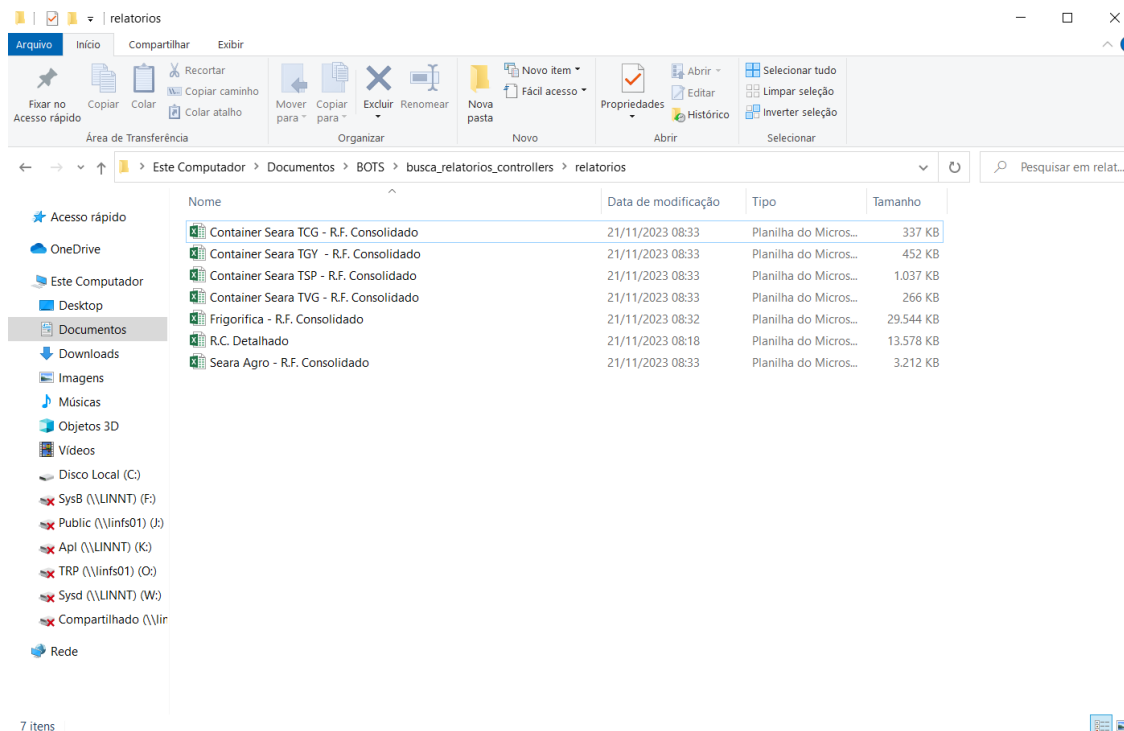
Se os dados forem carregados sem erros, o próximo passo consiste em salvar o arquivo seguindo a mesma especificação apresentada na Figura 7. A única distinção é que, por se tratar de outro relatório, o nome do arquivo salvo na pasta de relatórios será "Seara Agro - R. F. Consolidado.xlsx", com a ressalva de que cada relatório deve conter o nome do seu segmento para facilitar no momento que os controllers forem recolher seus arquivos no diretório.

Ao repetir o processo de extração do Razão F. Consolidado para os demais segmentos, os arquivos de Razão de Frete aparecem na pasta de relatórios da seguinte maneira:





Figura 3.12 – Arquivo R. C. Detalhado.xlsx contido na pasta Relatorios



Fonte: Elaborada pelos autores, 2023.

3.2.4 Junção de Dados Razão de Frete e Base Segmento

Na fase anterior, o robô concluiu a extração dos relatórios de Razão de Frete Consolidado e Razão Contábil Detalhado. Com os arquivos gerados e devidamente armazenados na pasta "Relatórios", a próxima etapa envolve uma manipulação simples dos dados do Razão de Frete para garantir uma entrega mais abrangente ao cliente.

No arquivo do Razão de Frete Consolidado, além dos dados referentes aos fretes realizados no mês, a única informação disponível sobre os veículos utilizados no transporte está registrada na coluna "Placa". Durante o processo de mapeamento, ficou esclarecido que os controllers realizam uma busca (PROCV) das placas no Razão de Frete Consolidado, utilizando uma planilha adicional denominada "base segmento". Essa planilha contém informações atualizadas e abrangentes sobre todos os veículos da frota, seja ativos ou inativos, em cada operação.

Ao realizar o PROCV, os controllers incorporam da base segmento informações suplementares, como Regional, Segmento, Filial, Ativo, Ccusto (centro de custo), Tipo de Veículo e Família, proporcionando uma visão mais completa e detalhada dos dados





relacionados aos veículos, o que possibilita uma análise mais robusta da operação em suas viagens.

Para dar início à etapa de junção dos dados, o robô, por meio da biblioteca Pandas, carrega a planilha Excel "Base_Segmentos_Filiais.xlsx" como um DataFrame do Pandas, atribuindo-o à variável `base_segmento`. Feito isso, inicia-se um loop sobre os elementos da lista `segmento`. Para cada elemento `i` na lista, o código executa as operações dentro do bloco do loop.

Ainda utilizando a biblioteca Pandas, é feita a leitura do arquivo Excel do Razão de Frete. O nome do arquivo é construído usando o valor atual de `i`, que é um elemento da lista `segmento`, ou seja, para cada `segmento` encontrado no nome dos arquivos de Razão de Frete armazenados na pasta relatório.

```
#importa planilha base segmento DataFrame
base_segmento=pd.read_excel(r'\\mtzqvfs01\QlikSenseCorporativo\Apoio\TR
P/Base_Segmentos_Filiais.xlsx')

for i in segmento:
    razao_frete = pd.read_excel(caminho_relatorios + '\\' + i + 'f' - R.F.
Consolidado.xlsx', header=1)
```

Quanto ao arquivo da base segmento, é realizado um filtro das linhas em que a coluna 'Ativo' é igual a '1-Sim', e o resultado é armazenado em outro DataFrame chamado `base_segmento2`. Sendo assim, `base_segmento2` armazena apenas os dados de placas que estão ativas na JBS Transportadora.

```
#---Filtrar a base segmentos onde a coluna 'Ativo' é igual a '1-Sim'
base_segmento2 = base_segmento[(base_segmento['Ativo'] == '1-Sim')]
```

Em seguida, é realizada uma operação de junção (merge) entre os DataFrames `razao_frete` e `base_segmento2` com base na coluna 'Placa'. O resultado é armazenado em `tabela_unida`.

Por fim, o DataFrame resultante é salvo como um novo arquivo Excel porém mantendo o mesmo nome de pré-defino ao gerar os relatórios, dessa forma, ao salvar os DataFrames novamente na pasta de relatórios, o robô apenas irá sobrescrever os





antigos arquivos de Razão com os novos contendo as colunas a mais. O nome do arquivo é novamente construído com base no valor atual de i (segmento). O argumento `index=False` evita a inclusão de índices no arquivo Excel.

```
#unindo a tabela Razão de Frete (esquerda/left) com a Base Segmento
(direita/right) na chave 'Placa'
tabela_unida = razao_frete.merge(base_segmento2[['Placa', 'Regional',
'Segmento', 'Filial', 'Ativo', 'Ccusto', 'Tipo Veículo', 'Família']], left_on='Placa',
right_on='Placa', how='left')

#Criar um excel da tabela gerada
tabela_unida.to_excel(caminho_relatorios + '\\' + i + f' - R.F. Consolidado.xlsx',
index=False)
```

3.2.5 Cria Arquivo Zip

A penúltima etapa do robô consiste em gerar um arquivo .zip que inclui todos os documentos do diretório "relatórios". Essa medida visa facilitar a disponibilização dos arquivos para os segmentos.

O trecho de código abaixo utiliza a biblioteca `shutil` para criar um arquivo zip a partir da especificação do diretório Relatórios.

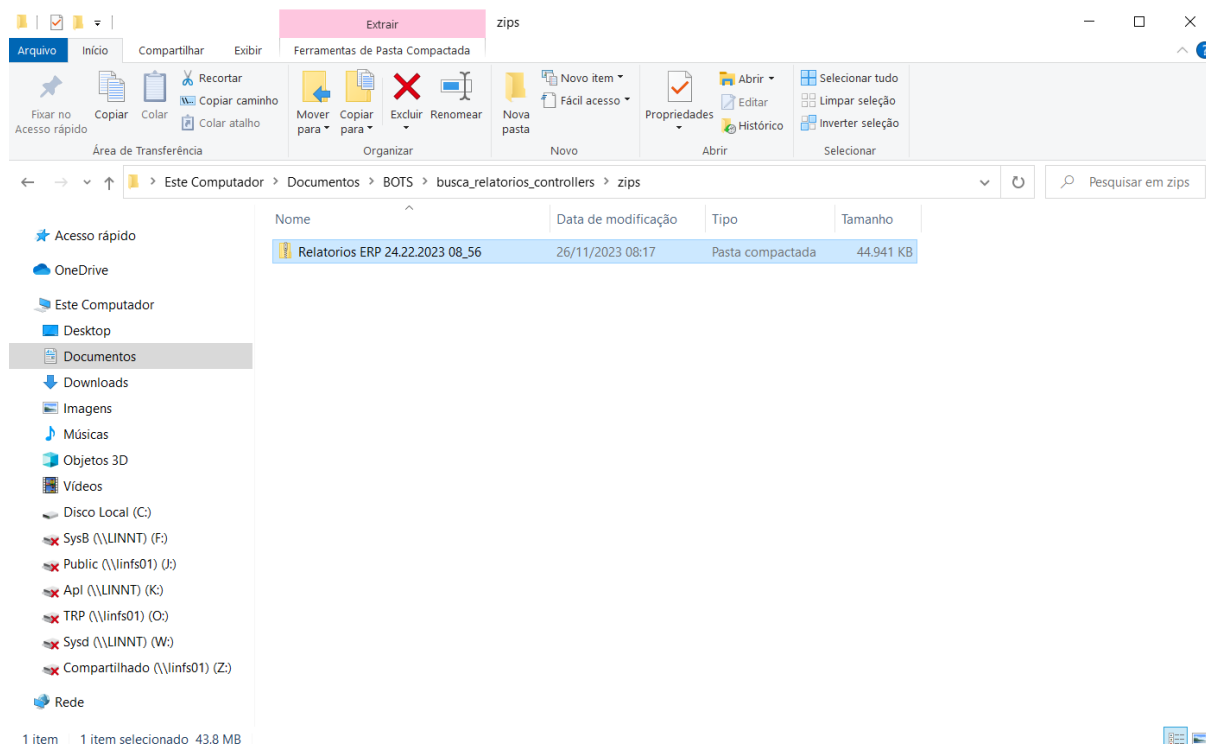
```
#criação de pasta zip com todo conteúdo
caminho_zip=.make_archive(os.path.join(caminho_zips, f'{nome_arquivo}',
'zip', caminho_relatorios)
```

No terceiro argumento é especificado o caminho do diretório que será salvo o zip criado (`caminho_relatorios`). Para todo arquivo compactado que o robô criar será salvo numa pasta específica para receber os zips.





Figura 3.13 – Tela Pasta Relatório Zip



Fonte: Elaborada pelos autores, 2023.

O nome da pasta incorpora a data e a hora de criação do arquivo, garantindo que, ao longo do dia e nas execuções subsequentes do bot, a pasta seja gerada com identificadores únicos. Essa abordagem evita a sobreposição com a pasta anterior e a possível substituição do último arquivo ZIP.

3.2.6 Upload do arquivo Zip na Intranet

Inicialmente, a proposta era disponibilizar os relatórios para os funcionários por meio de uma pasta de rede compartilhada. No entanto, devido à dispersão das equipes, especialmente aquelas que não operam na unidade de Lins, essa ideia foi inviabilizada. Como alternativa, optou-se por disponibilizar os arquivos no formato .ZIP na Intranet da empresa. A Intranet é uma plataforma web interna, acessível a todos os colaboradores da rede JBS, permitindo o acesso de qualquer local conectado à internet da organização.

Para automatizar a navegação no navegador Chrome, foi empregada a biblioteca Selenium. A função `driver.get` recebe a variável `caminho_intranet`, que





contém a URL da página de Compartilhamento de Projetos da Transportadora na Intranet.

Além disso, foi implementada a função `driver.implicitly_wait``, que estabelece um tempo máximo de espera para a localização de elementos na página antes de gerar uma exceção. Isso significa que, se o script estiver procurando por um elemento na página e não o encontrar imediatamente, ele aguardará até o tempo especificado, neste caso, 10 segundos, antes de desistir.

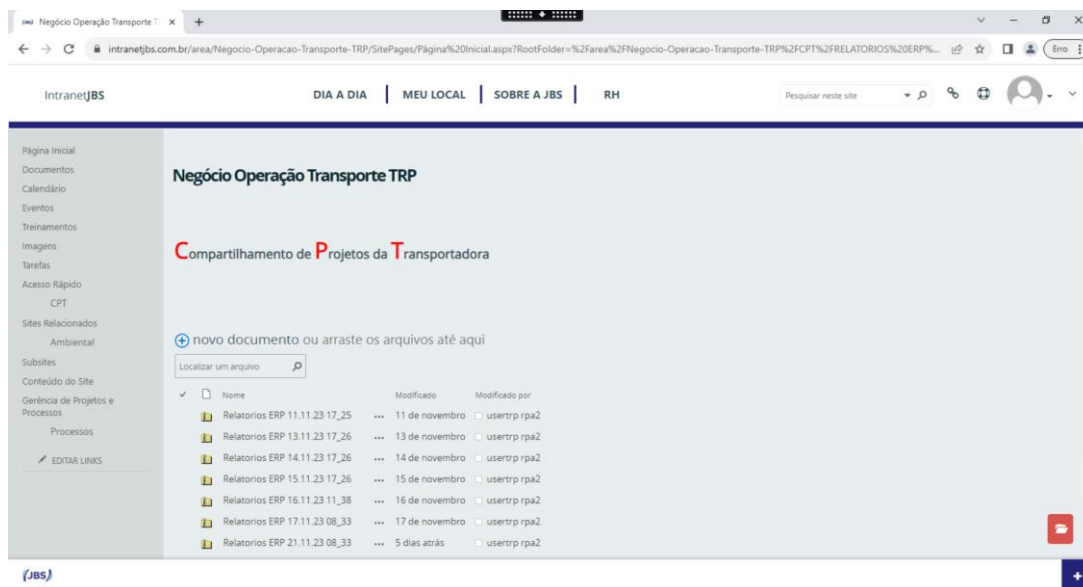
Segue o trecho de código responsável pelo upload da pasta na Intranet:

```
# Upload de pasta na Intranet
chrome_options = webdriver.ChromeOptions()
chrome_options.add_argument('--no-proxy-server')
driver=webdriver.Chrome('chromedriver.exe',
chrome_options=chrome_options)
driver.get(caminho_intranet)
pg.sleep(4)
driver.implicitly_wait(10)
``
```





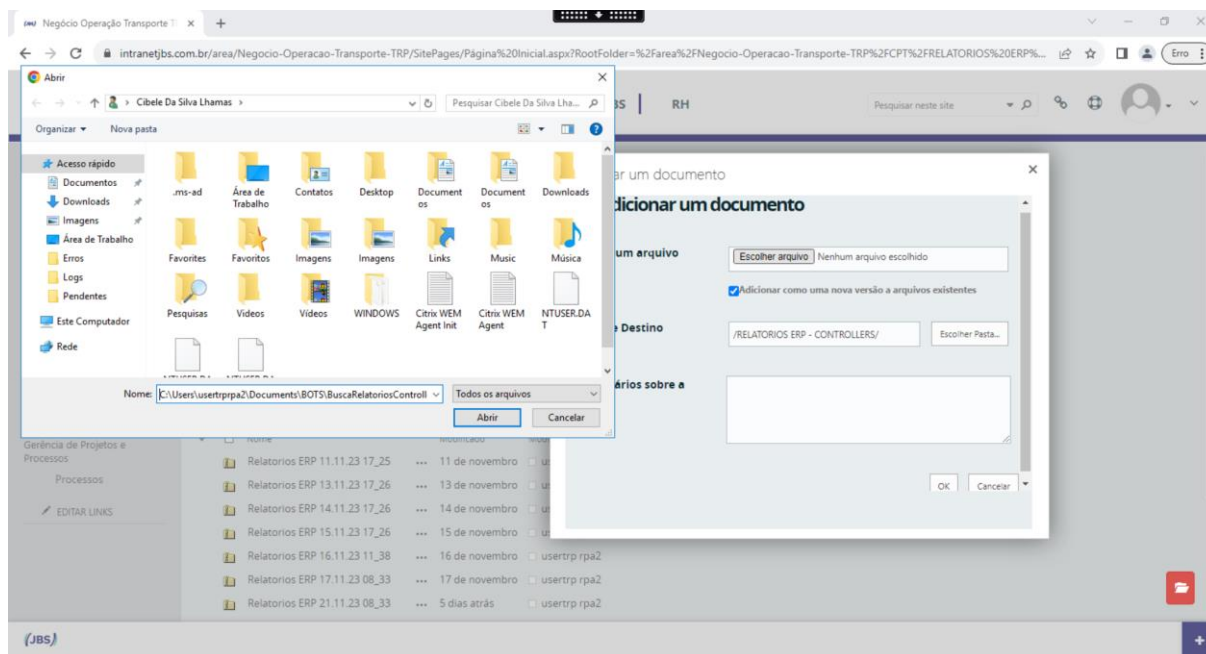
Figura 3.14 – Operação Transporte TRP



Fonte: Elaborada pelos autores, 2023.

Após aberta a tela de Compartilhamento de Projetos, é feita a inclusão do novo documento em uma pasta na Intranet conforme exibido na Figura 3.15.

Figura 3.15 – Inclusão de novo documento



Fonte: Elaborada pelos autores, 2023.





Para confirmar a efetivação do upload do arquivo ZIP na Intranet, o *bot* realiza uma validação adicional. Nesse estágio, usa-se a estrutura ``try/except``, na qual o robô procura por um elemento na página contendo um link de texto correspondente ao valor armazenado na variável ``nome_zip``, que armaneza o nome do último arquivo ZIP gerado.

Entretanto, caso essa busca não tenha êxito, o robô interrompe o processo e registra uma mensagem informativa por meio da biblioteca “logging” do Python. Essa mensagem indica que o *bot* não conseguiu concluir o upload do arquivo ZIP.

4 CONCLUSÕES

Após a conclusão dos scripts, estimou-se que o robô requer aproximadamente 10 minutos para gerar e disponibilizar os relatórios na Intranet até a metade do mês. Entretanto, mais próximo do encerramento do mês, esse período se estende para cerca de 20 minutos, considerando que os relatórios demandam mais tempo para carregar no Corporate devido ao expressivo volume de dados.

Conforme especificado na fase de mapeamento do contexto, os times anteriormente consumiam aproximadamente 5 minutos para a geração do relatório de Razão Contábil Detalhado e cerca de 8 minutos para o Razão de Frete Consolidado, incluindo a junção dos dados com a Base Segmento. Como resultado, a implementação do robô resultou em uma economia significativa de 15 minutos diários para o relatório Contábil e de 24 minutos para o relatório de Frete, considerando que ambos eram gerados manualmente cerca de 3 vezes ao dia para um acompanhamento mais eficaz de cada time.

Ao longo do desenvolvimento do projeto, ficou evidente o grande interesse por parte dos controllers em contar com um robô que não apenas gerasse e extraísse os relatórios, mas também realizasse a junção dos dados do arquivo Base Segmento com o relatório de Razão de Frete Consolidado. A utilização recorrente entre os usuários e a ferramenta proporcionou uma oportunidade contínua de melhorias, à medida que, com o tempo, foram identificados pontos no processo passíveis de otimização.

Após algumas semanas, os controllers identificaram novas funcionalidades que poderiam aprimorar ainda mais a eficiência do robô. Por exemplo, sugeriram que o bot começasse a gerar não apenas os relatórios do mês atual, mas também do mês





anterior até o 5º dia útil do mês. Essa sugestão surgiu devido à particularidade em alguns casos, nos quais não é possível faturar todos os fretes até o final do mês, resultando na necessidade de faturar ao longo dos primeiros dias do mês atual, que são contabilizados na competência do mês anterior.

Outra solicitação feita posteriormente foi a integração dos dados da base segmento com o relatório de Razão Contábil Detalhado, uma atividade que não tinha sido proposta na fase inicial de mapeamento. Essas adaptações e evoluções, impulsionadas pela experiência prática, refletem o caráter dinâmico do processo de automação e a capacidade de resposta do robô às necessidades emergentes dos usuários.

Por fim, espera-se implementar as novas solicitações até dezembro de 2023 e manter o suporte contínuo aos usuários para eventuais manutenções ou melhorias que possam ser requisitadas. A boa receptividade por parte dos usuários atuais resultou em um interesse adicional por parte de dois segmentos que inicialmente não estavam interessados em se envolver no projeto. Diante desse interesse, até janeiro de 2024, está planejado mapear e programar o robô para a geração de novos relatórios destinados a esses dois novos usuários do software. Essa expansão do escopo reflete não apenas a eficácia do robô, mas também a crescente demanda e reconhecimento da utilidade da automação dentro da organização.





REFERÊNCIAS BIBLIOGRÁFICAS:

- CASTRO, B. A. **O que é Robotic Process Automation (RPA) e como integrar com BPMS.** 2020. Disponível em: <https://blog.smlbrasil.com.br/robotic-process-automation-rpa/>
- CHEN, Daniel Y. **Análise de dados com Python e Pandas.** Novatec Editora, 2018.
- COSTA, Diogo; SÃO MAMEDE, Henrique; SILVA, Miguel Mira da. **Robotic Process Automation (RPA) adoption: a systematic literature review.** *Engineering Management in Production and Services*, v. 14, n. 2, p. 1-12, 2022.
- CIBI, C. et al. **Human Computer Interaction Using Arduino and Python.** In: **2022 1st International Conference on Computational Science and Technology (ICCST).** IEEE, 2022. p. 1-3.
- GIBBS, Fred. **Instalação de Módulos Python com pip. The Programming Historian em Português,** 2021.
- GARCIA, Boni. **Hands-On Selenium WebDriver with Java.** " O'Reilly Media, Inc.", 2022.
- GRAGLIA, Marcelo Augusto Vieira; LAZZARESCHI, Noêmia. **A indústria 4.0 e o futuro do trabalho: tensões e perspectivas.** *Revista Brasileira de Sociologia-RBS*, v. 6, n. 14, 2018.
- Giridhar, K. V. (2019). **Mastering Python Scripting for System Administrators: A practical guide to learn the fundamentals of Python scripting to deploy real-world projects.** Packt Publishing. ISBN-13: 978-1789133223.
- GRUPO QUALITAT. **Automação Robótica de Processos: Uma ferramenta necessária para a transformação digital.** São Paulo, 2020. 2 p.
- MADAKAM, Somayya; HOLMUKHE, Rajesh M.; JAISWAL, Durgesh Kumar. **The future digital work force: robotic process automation (RPA).** *JISTEM-Journal of Information Systems and Technology Management*, v. 16, 2019.
- NYAMATHULLA, S. et al. A Review on Selenium Web Driver with Python. **Annals of the Romanian Society for Cell Biology**, p. 16760-16768, 2021.
- SRINATH, K. R. **Python—the fastest growing programming language.** *International Research Journal of Engineering and Technology*, v. 4, n. 12, p. 354-357, 2017.
- SWEIGART, Al. **Pyautogui documentation. Read the Docs**, v. 25, 2020.
- VAN DER AALST, Wil MP; BICHLER, Martin; HEINZL, Armin. **Robotic process automation. Business & information systems engineering**, v. 60, p. 269-272, 2018.

REPOSITÓRIO DAS BIBLIOTECAS:

<https://pypi.org/project/PyAutoGUI/> - Pyautogui

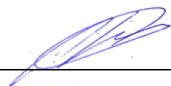
<https://selenium-python.readthedocs.io/installation.html> - Selenium

<https://zetcode.com/python/python-pandas/> - Pandas

<https://docs.python.org/pt-br/dev/howto/logging.html> - Logging



Página de assinaturas



Anderson Pazin
264.548.978-85
Signatário



Fernando Muzzi
145.859.038-01
Signatário











Felipe Rodrigues
345.484.558-09
Signatário



Gisele Becari
145.698.268-06
Signatário

HISTÓRICO

19 dez 2023 11:19:18		Anderson Pazin criou este documento. (E-mail: anderson.pazin@fatec.sp.gov.br, CPF: 264.548.978-85)
19 dez 2023 11:19:18		Anderson Pazin (E-mail: anderson.pazin@fatec.sp.gov.br, CPF: 264.548.978-85) visualizou este documento por meio do IP 177.95.133.194 localizado em Guaratingueta - Sao Paulo - Brazil
19 dez 2023 11:19:20		Anderson Pazin (E-mail: anderson.pazin@fatec.sp.gov.br, CPF: 264.548.978-85) assinou este documento por meio do IP 177.95.133.194 localizado em Guaratingueta - Sao Paulo - Brazil
19 dez 2023 17:55:59		Felipe Maciel Rodrigues (E-mail: felipe.rodrigues30@fatec.sp.gov.br, CPF: 345.484.558-09) visualizou este documento por meio do IP 201.182.122.11 localizado em Lins - Sao Paulo - Brazil
19 dez 2023 17:56:09		Felipe Maciel Rodrigues (E-mail: felipe.rodrigues30@fatec.sp.gov.br, CPF: 345.484.558-09) assinou este documento por meio do IP 201.182.122.11 localizado em Lins - Sao Paulo - Brazil
22 dez 2023 22:28:35		Gisele Molina Becari (E-mail: gisele.becari@fatec.sp.gov.br, CPF: 145.698.268-06) visualizou este documento por meio do IP 189.126.188.242 localizado em Lins - Sao Paulo - Brazil
22 dez 2023 22:32:18		Gisele Molina Becari (E-mail: gisele.becari@fatec.sp.gov.br, CPF: 145.698.268-06) assinou este documento por meio do IP 189.126.188.242 localizado em Lins - Sao Paulo - Brazil
19 dez 2023 11:46:08		Fernando Augusto Garcia Muzzi (E-mail: fernando.muzzi@fatec.sp.gov.br, CPF: 145.859.038-01) visualizou este documento por meio do IP 191.187.108.173 localizado em Marília - Sao Paulo - Brazil



19 dez 2023

11:47:20



Fernando Augusto Garcia Muzzi (E-mail: fernando.muzzi@fatec.sp.gov.br, CPF: 145.859.038-01) assinou este documento por meio do IP 191.187.108.173 localizado em Marília - Sao Paulo - Brazil

