



**UNIVERSIDADE ESTADUAL DA PARAÍBA
SECRETARIA DE ESTADO DA CIÊNCIA, TECNOLOGIA, INOVAÇÃO E ENSINO
SUPERIOR - SECTIES
PÓLO JOÃO PESSOA
CURSO SUPERIOR DE TECNOLOGIA EM CIÊNCIA DE DADOS**

ARTUR MEDEIROS DO LAGO SOARES

APRIMORANDO PROCESSOS COM AUTOMAÇÃO E WEB SCRAPING

**JOÃO PESSOA - PB
2025**

ARTUR MEDEIROS DO LAGO SOARES

APRIMORANDO PROCESSOS COM AUTOMAÇÃO E WEB SCRAPING

Trabalho de Conclusão de Curso, apresentado ao Centro de Ciências e Tecnologia da Universidade Estadual da Paraíba como requisito parcial à obtenção do título de Tecnólogo em Ciência de Dados.

Orientadora: Prof.^a Dr. Ana Patrícia Bastos Peixoto.

É expressamente proibida a comercialização deste documento, tanto em versão impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que, na reprodução, figure a identificação do autor, título, instituição e ano do trabalho.

S676a Soares, Artur Medeiros do Lago.
Aprimorando processos com automação e web scraping
[manuscrito] / Artur Medeiros do Lago Soares. - 2025.
31 f. : il. color.

Digitado.

Trabalho de Conclusão de Curso (Graduação em
Tecnologia em ciência de dados) - Universidade Estadual da
Paraíba, Centro de Ciências e Tecnologia, 2025.

"Orientação : Prof. Dra. Ana Patricia Bastos Peixoto de
Oliveira, Departamento de Estatística - CCT".

1. Linguagem de programação. 2. Produtividade
empresarial. 3. Multithreading. I. Título

21. ed. CDD 004.36

ARTUR MEDEIROS DO LAGO SOARES

APRIMORANDO PROCESSOS COM AUTOMAÇÃO E WEB SCRAPING

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Tecnologia em Ciência de Dados da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Tecnólogo em Tecnologia em Ciência de Dados

Aprovada em: 09/06/2025.

BANCA EXAMINADORA

Documento assinado eletronicamente por:

- **Ricardo Alves de Olinda** (***.035.004-**), em **24/06/2025 22:06:11** com chave **956c1400516011f0a0662618257239a1**.
- **Ana Patricia Bastos Peixoto de Oliveira** (***.335.455-**), em **24/06/2025 22:01:36** com chave **f159a9ea515f11f0924c1a7cc27eb1f9**.
- **Silvio Fernando Alves Xavier Junior** (***.025.684-**), em **25/06/2025 09:58:01** com chave **06a0c46e51c411f08a7a06adb0a3afce**.

Documento emitido pelo SUAP. Para comprovar sua autenticidade, faça a leitura do QrCode ao lado ou acesse https://suap.uepb.edu.br/comum/autenticar_documento/ e informe os dados a seguir.

Tipo de Documento: Folha de Aprovação do Projeto Final

Data da Emissão: 25/06/2025

Código de Autenticação: 227602



À minha mãe, por todo o apoio, companheirismo,
dedicação e esforço para comigo, mesmo diante de
momentos difíceis de sua vida, DEDICO.

AGRADECIMENTOS

Agradeço à Secretaria de Estado da Ciência, Tecnologia, Inovação e Ensino Superior (SECTIES) e à Fundação de Apoio à Pesquisa do Estado da Paraíba (FAPESQ) pelo apoio fundamental para a realização deste trabalho.

RESUMO

A automação consiste em delegar tarefas que normalmente seriam executadas por humanos a sistemas computacionais. Essa técnica se popularizou nos últimos anos, sendo amplamente adotadas em empresas que buscam aumentar sua produtividade. O principal objetivo desse trabalho é relatar a experiência do estagiário durante período de trabalho na empresa SMR CLIPPING. As principais atividades realizadas foram a automação de tarefas repetitivas, utilização de *web scraping* para extração de dados *online* e melhoria das importações de notícias. Para realizar essas atividades, foram usadas linguagens de programação/*script* como Python, C#, *batch script* e PowerShell, com a utilização de bibliotecas como Selenium, BeautifulSoup e PyAutoGUI. Também foi utilizado a técnica de *multithreading*, possibilitando a redução de 90% do tempo de execução de um dos processos. Foi observado o ganho de 149% de produtividade nas importações de notícias. A utilização de automações trouxe enormes ganhos de produtividade dentro da SMR CLIPPING, demonstrando o quão importante é o uso no mundo corporativo. A experiência obtida pelo estagiário ajudou a perceber seu interesse pela área da engenharia de dados.

Palavras-Chave: linguagem de programação; produtividade; *multithreading*.

ABSTRACT

Automation consists of delegating tasks that would normally be performed by humans to computer systems. This technique has become popular in recent years and has been widely adopted by companies seeking to increase their productivity. The main objective of this paper is to report the intern's experience during his work period at SMR CLIPPING. The main activities performed were the automation of repetitive tasks, the use of web scraping to extract online data, and the improvement of news imports. To perform these activities, programming/scripting languages such as Python, C#, batch script, and PowerShell were used, with the use of libraries such as Selenium, BeautifulSoup, and PyAutoGUI. The multithreading technique was also used, enabling a 90% reduction in the execution time of one of the processes. A 149% increase in productivity was observed in news imports. The use of automation brought enormous productivity gains within SMR CLIPPING, demonstrating how important its use is in the corporate world. The experience gained by the intern helped him realize his interest in the area of data engineering.

Keywords: programming language; productivity; multithreading.

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
CAPTCHA	<i>Completely Automated Public Turing test to tell Computers and Humans Apart</i>
CLI	<i>Command Line Interface</i>
CSS	<i>Cascading Style Sheets</i>
DOM	<i>Document Object Model</i>
GUI	<i>Graphical User Interface</i>
HTML	<i>HyperText Markup Language</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
ID	<i>Identifier</i>
IP	<i>Internet Protocol</i>
LEGO	Sistema de Importação criado em C#
PI	Plataforma de Importação
PPC	Projeto Pedagógico de Curso
RPA	<i>Robotic Process Automation</i>
SMI	Sistema <i>Web</i> da SMR CLIPPING
TI	Tecnologia da Informação
URL	<i>Uniform Resource Locator</i>
XPATH	<i>XML Path Language</i>

SUMÁRIO

1	INTRODUÇÃO	9
2	REVISÃO DE LITERATURA.....	10
2.1	AUTOMAÇÃO	10
2.2	WEB SCRAPING	11
2.3	APLICAÇÕES EM CIÊNCIA DE DADOS	13
3	METODOLOGIA.....	14
3.1	CONTRIBUIÇÃO DA ÁREA PARA A MISSÃO DA EMPRESA	14
3.2	IDENTIFICAÇÃO DA RELAÇÃO COM OUTRAS ÁREAS DA EMPRESA	14
3.3	DESCRIÇÃO DO ESTÁGIO E DAS PRINCIPAIS ATIVIDADES REALIZADAS ...	16
3.3.1	O estágio	16
3.3.2	Importação	17
3.3.3	Criação de scripts para automação de processos	21
4	RESULTADOS E DISCUSSÃO	22
4.1	PROGRAMA COM FECHAMENTO ALEATÓRIO	22
4.2	AUTOMAÇÃO COM PYTHON E OTIMIZAÇÃO COM C#	23
4.3	AUMENTO EXPRESSIVO DE PRODUTIVIDADE	26
5	CONSIDERAÇÕES FINAIS.....	29
	REFERÊNCIAS	30

1 INTRODUÇÃO

O estágio supervisionado é um fator indispensável para formar um cientista de dados, pois permite aplicar de forma prática os conhecimentos que foram sendo absorvidos durante todo o período da graduação. Os estágios funcionam como uma forma de impulsionar os estudantes, oferecendo a oportunidade de poderem "[...]combinar os conhecimentos teóricos com situações de trabalho do mundo real e colocá-los em prática" (Sahrir *et al.*, 2016 *apud* Bawica, 2021, p. 95, tradução própria). Levando em consideração que "o aspecto mais importante na realização de um estágio [...] é produzir estudantes que estejam mais preparados para o mundo do trabalho e para serem profissionais" (Lutfia; Rahadi, 2020, p. 199, tradução própria). Essa imersão prática é tão importante na vida do estudante que, segundo os mesmos autores:

O aprimoramento da competência do estudante por meio do programa de estágio pode alcançar os padrões da profissão e servir como uma provisão de experiência para trabalhar no próximo lugar (Lutfia; Rahadi, 2020, p. 203, tradução própria).

Nesse caso, o trabalho exercido no estágio foi direcionado especificadamente ao uso de automação e *web scraping*, que são duas técnicas muito conhecidas e utilizadas no mundo corporativo para otimizar tarefas repetitivas e extrair dados de forma mais eficiente e em grande escala.

O estágio foi realizado na empresa SMR CLIPPING, com sua sede em Brasília, fundada em 17 de fevereiro de 1995, começando suas atividades de forma manual, com funcionários recortando notícias de jornais da época, colando em folhas e entregando aos seus clientes. Com o passar dos anos e com a tecnologia avançando, a empresa que antes realizava um trabalho físico e manual, passou a realizar a maior parte das atividades de forma digital e automatizada. De início, a SMR CLIPPING expandiu seu serviço adicionando recorte de gravações de rádio. Logo após incluiu recortes de notícias de televisão, *websites* e produção de relatórios para clientes de médio e grande porte.

O objetivo principal do presente trabalho é trazer o relato das atividades que foram desenvolvidas durante o período de estágio no setor de Tecnologia da Informação (TI), com foco nas soluções construídas utilizando a automação e técnicas de raspagem de dados na *web*, cujas práticas contribuem diretamente para trazer a redução de falhas, economia de tempo e a melhoria da produtividade da empresa, reforçando a importância da utilização dessas técnicas na área de dados.

2 REVISÃO DE LITERATURA

Nesta seção será abordado os conceitos sobre a importância da utilização, ferramentas existentes e principais linguagens utilizadas para a construção de automações. Sobre *web scraping*, será discutido a definição, como funciona a utilização e as etapas geralmente usadas. Também será discutido sobre aplicações dessas técnicas na ciência de dados, abordando, por exemplo, o uso de *web scraping* para alimentar bancos de dados.

2.1 AUTOMAÇÃO

É possível visualizar a automação como o processo de atribuir tarefas que seriam feitas por humanos a sistemas de computação. O objetivo principal é o aumento da eficiência e redução de falhas que podem ocorrer caso seja uma pessoa na execução da tarefa. A técnica de automação se popularizou há alguns anos, sendo muito procurada e utilizada por empresas que buscam mais produtividade, principalmente quando se trata de tarefas repetitivas.

Segundo Andrade, Dallilo e Florian (2022), o *Robotic Process Automation* (RPA) é indicado para lidar com grandes volumes de dados e tarefas repetitivas, sendo utilizado com sucesso em vários setores do mundo empresarial. Com o RPA, podemos automatizar tarefas em *softwares*¹ como se fosse um ser humano realizando suas ações. Por exemplo, o preenchimento de planilhas, visitas a *websites* e extrações de dados de produtos em uma loja *online*. Processos que se fossem feitos manualmente por uma pessoa, dependendo da quantidade de micro tarefas e dados, demora horas ou até mesmo dias.

Algumas linguagens de programação/*script*² como o Python e *batch script* são bem conhecidas por sua simplicidade para criar automações de forma rápida. O Python é uma linguagem muito versátil, possui uma comunidade bastante ativa e uma grande quantidade de bibliotecas disponíveis para seus usuários, possibilitando a criação de *scripts* eficientes e com custo muito baixo, facilitando muito a implementação.

Ferramentas como UiPath (com versão grátis) e Power Automate possuem um certo destaque no mundo corporativo. Para quem não possui tanto conhecimento na área da programação e busca programas com interface gráfica e integrações com vários sistemas, essas três ferramentas podem ser muito úteis.

¹ *Softwares*: Programas utilizados nos dispositivos que permitem ao usuário executar uma série de tarefas.

² *Script*: Sequência de passos que o computador vai interpretar.

Fazer a utilização de soluções de automação impactam diretamente na redução de custos, redução de erros e aumenta o tempo para exercer atividades que necessitam de raciocínio humano. Empresas que não buscam se beneficiar desse tipo de tecnologia, estão ficando para trás, pois o mercado a cada dia está mais competitivo e eficiente quando se trata de produção no geral. De acordo com Freitas Júnior (2021), a automação permite que profissionais deixem de executar tarefas manuais e se concentrem em ações que geram maior valor para a organização.

2.2 WEB SCRAPING

O *web scraping* é uma técnica muito utilizada para fazer a extração de dados de *websites* de forma automatizada, as informações não estruturadas que geralmente estão em *HyperText Markup Language* (HTML³) - “[...]padrão oculto atrás de quase tudo o que vemos e fazemos ao navegar na Web” (Nascimento, 2017, p. 3) - são convertidas em estruturas organizadas e reutilizáveis. Essa técnica é muito utilizada quando os dados que queremos obter não estão disponibilizados via *Application Programming Interface* (API⁴) ou arquivos para fazer *download*, sendo considerada uma técnica estratégica de coleta de dados em grande escala. “Teoricamente, web scraping é a prática de coletar dados por qualquer meio que não seja um programa interagindo com uma API (ou, obviamente, por um ser humano usando um navegador web)” Mitchell (2019, p. 10).

O processo de *scraping* normalmente possui três etapas principais, a requisição da página *web*, a extração das informações e a transformação em dados estruturados. De acordo com Noortje Marres e Esther Weltevrede (2013), podemos entender o processo de *scraping* da seguinte forma:

Scraping, para dizer de forma bastante formal, é uma técnica proeminente para a coleta automatizada de dados online. Trata-se de uma das práticas mais distintas associadas às formas atuais de pesquisa social digital, marcadas pela ascensão da Internet e pela nova ubiquidade dos dados digitais na vida social. Scrapers, para dizer de maneira mais informal, são trechos de código de software que tornam possível baixar automaticamente dados da Web e capturar parte das grandes quantidades de dados sobre a vida social disponíveis em plataformas online como Google, Twitter e Wikipédia (Marres; Weltevrede, 2013, p. 313, tradução própria).

³ HTML: Linguagem de marcação usada para estruturar e dar significado ao conteúdo *web*.

⁴ API: Conjunto de regras ou protocolos que permitem que aplicativos de *software* se comuniquem entre si para trocar dados, recursos e funcionalidades.

Para conseguir localizar os elementos da página com maior precisão, como o título de uma matéria jornalística, uma das técnicas mais utilizadas é o XML *Path Language* (XPath). Segundo Mitchell (2019, p. 89), “O XPath é geralmente usado para obter um conteúdo textual, incluindo textos em tags filhas (por exemplo, uma tag <a> em um bloco de texto)”. Essa técnica possibilita a navegação pela estrutura do *Document Object Model* (DOM⁵), fazendo com que seja possível encontrar cada estrutura HTML existente. Também há outras formas de localizar os dados que queremos em uma página, como o uso de seletores *Cascading Style Sheets* (CSS⁶) e *parsing* de HTML. Mitchell (2019, p. 89) também afirma que, “Se o seletor CSS for usado para isso, todo o texto das tags filhas será ignorado”.

Quando a extração de dados é feita em *sites* estáticos, é possível utilizar bibliotecas mais simples, como o BeautifulSoup do Python. Essa biblioteca apenas faz uma requisição ao *website* que queremos fazer a raspagem de dados e captura automaticamente toda a estrutura HTML. “[...][O] BeautifulSoup tenta dar sentido ao que não faz sentido: ajuda a formatar e a organizar a web confusa, fazendo correções em um código HTML mal formatado e apresentando objetos Python que podem ser facilmente percorridos[...].” Mitchell (2019, p. 21). Existem também os *sites* dinâmicos, onde é necessário a renderização JavaScript⁷, fazendo com que seja necessária uma interação na página, como rolar para baixo para poder carregar os elementos da página, em casos como esse, é recomendado utilizar bibliotecas como Selenium, Scrapy ou Playwright, pois possuem a capacidade de renderizar páginas construídas em JavaScript.

Porém, no *web scraping* existem muitos desafios, o principal é a mudança de estrutura de *layout* dos *sites*, fazendo com que o profissional responsável pela extração dos dados tenha que fazer manutenções constantes nos códigos de captura dos elementos, majoritariamente sendo manutenções nos códigos XPath. Alguns *sites* podem utilizar estratégias de restrição como *Completely Automated Public Turing test to tell Computers and Humans Apart* (CAPTCHAs⁸), limites de requisição e bloqueio por *headers*⁹ HTTP. Por isso, para evitar bloqueios e até banimentos, é importante respeitar as regras presentes no arquivo *robots.txt* de cada portal *web*. Koster *et al.* (2022) definem o *robots.txt* como um arquivo disponibilizado pelos desenvolvedores que constroem os *sites* na *web*. Esse arquivo contém regras que devem ser seguidas pelos *scrapers*, orientando quais partes do *site* podem ou não acessar.

⁵ DOM: Interface de programação para os documentos HTML e XML.

⁶ CSS: Linguagem de regras de estilo usada para aplicar estilo ao conteúdo HTML.

⁷ JavaScript: Linguagem de programação que permite criar conteúdo dinâmico e interativo na *web*.

⁸ CAPTCHAs: Métodos de autenticação que validam os usuários como humanos.

⁹ *Headers*: Informações ou estruturas que fornecem metadados ou instruções adicionais.

2.3 APLICAÇÕES EM CIÊNCIA DE DADOS

Enquanto o *web scraping* irá fazer a extração de dados diretamente da *web*, sendo muito utilizado quando não é possível extrair os dados através de APIs, de acordo com Sarker (2021, p. 2, tradução própria), “a ciência de dados é o campo de aplicação de métodos avançados de análise e conceitos científicos para extrair informações úteis de negócios a partir dos dados.”, ela irá realizar a extração de conhecimento e *insights* através de enormes volumes de dados, utilizando diretamente a estatística, *machine learning*¹⁰ e técnicas de engenharia de *software*.

Portanto, a automação e o *web scraping* são muito importantes para realizar a coleta, pré-processamento e atualização constante dos dados, possibilitando, por exemplo, criação de *pipelines*¹¹. Faria *et al.* (2023) mostraram que alimentar bases de dados via *web scraping* pode elevar o desempenho de modelos de classificação de imagens se os dados forem limpos e padronizados de forma correta.

Como foi visto anteriormente na seção 2.1, a automação será responsável por executar tarefas repetitivas. Na ciência de dados a automação é muito utilizada para criar agendamento de coletas, transformações e carregamento de dados em bancos, *data warehouses*¹² e *data lakes*¹³. Esse tipo de sistematização reduz significativamente os custos operacionais, diminui consideravelmente a quantidade de erros e traz um ganho expressivo na redução do tempo de análise. Como demonstrado por Andrade, Dallilo e Florian (2022), o uso de RPA no *site* do Instituto Brasileiro de Geografia e Estatística (IBGE) permitiu a extração recorrente de dados populacionais, tornando um processo manual que seria extremamente repetitivo e demorado, em uma tarefa muito eficiente, rápida e confiável.

Com a utilização dessas técnicas é possível criar sistemas que podem monitorar preços de produtos em tempo real e alimentar modelos preditivos, tarefas que não seriam possíveis de executar com tanta eficiência caso fosse uma pessoa realizando. Por isso, a utilização de técnicas como automação e *scraping* são tão importantes na área da ciência de dados para garantir escalabilidade, velocidade e confiança no desenvolvimento de projetos.

¹⁰ *Machine learning*: Ramificação da inteligência artificial (IA) focada em permitir que computadores e máquinas imitem a maneira como os seres humanos aprendem, realizem tarefas de forma autônoma e melhorem seu desempenho e precisão por meio da experiência e exposição a mais dados.

¹¹ *Pipelines*: Etapas ou processos que os dados percorrem desde a sua entrada até a saída final.

¹² *Data warehouses*: Sistema que agrega dados de diversas fontes em um único e consistente armazenamento de dados.

¹³ *Data lakes*: Repositório de armazenamento de dados que suporta grandes quantidades de dados, podendo ser estruturados e não estruturados.

3 METODOLOGIA

Neste tópico será discutido a importância da área de TI na empresa SMR CLIPPING, evidenciando seus setores e a interligação que eles possuem. Outro assunto discutido será as informações sobre o estágio, por exemplo a data de início, quantos dias na semana e quantas horas diárias eram exercidas. Também será abordado acerca das principais atividades realizadas pelo estagiário, como a importação de notícias (ver seção 3.3.2) e a criação de *scripts* para automação de processos dentro da organização (ver seção 3.3.3).

3.1 CONTRIBUIÇÃO DA ÁREA PARA A MISSÃO DA EMPRESA

Atualmente no mundo corporativo o setor de TI é considerado obrigatório em empresas de pequeno, médio e grande porte. Esse fato acontece, porque qualquer sistema digital de uma organização necessita de um profissional da área para garantir o gerenciamento e funcionamento ininterrupto do sistema. A falta de uma boa gestão de TI pode ocasionar em falhas de segurança, perda de dados de grande relevância e interrupção dos serviços prestados, impactando diretamente a capacidade da empresa de alcançar suas metas e se manter competitiva no mercado.

A SMR CLIPPING, empresa que presta serviços de "clipagem" (recorte das notícias), é um exemplo de corporação que depende do setor de TI. A Empresa funciona de forma integral no ambiente digital, utilizando sistemas desenvolvidos para capturar notícias de diversas fontes, sendo via rádio (áudio), televisão (vídeo) ou matérias publicadas em *websites* (texto e imagem).

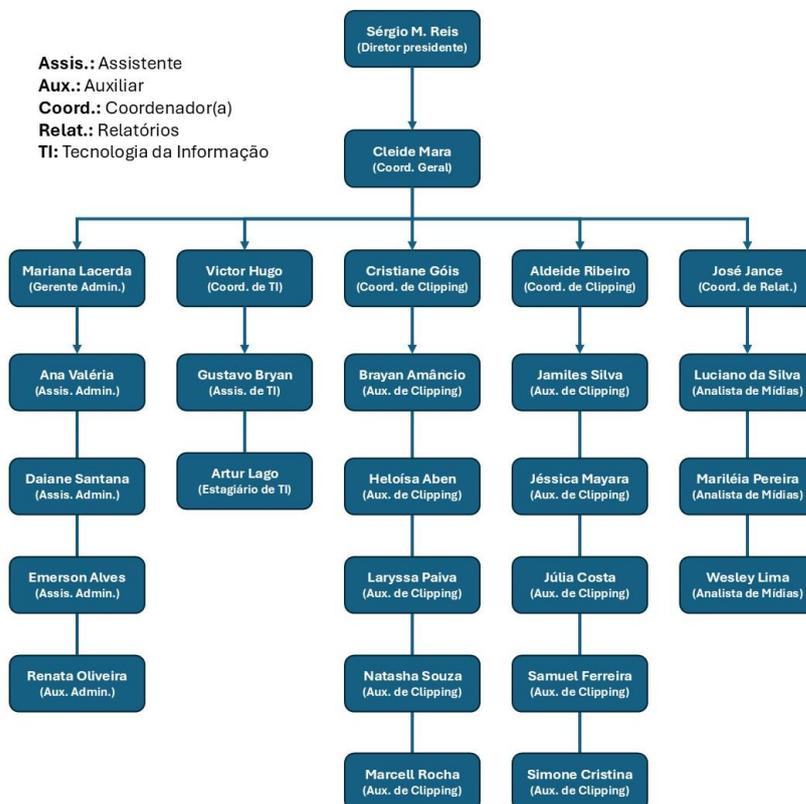
Esses dados extremamente necessários para a empresa, são tratados e armazenados em bancos de dados localizados em Brasília, no Distrito Federal. O gerenciamento desses bancos de dados é de responsabilidade do setor de TI, que além de assegurar a disponibilidade e boa performance, também realizam a implementação das políticas de segurança necessárias para a proteção desses sistemas da organização, destacando a importância desse setor para o funcionamento correto da organização.

3.2 IDENTIFICAÇÃO DA RELAÇÃO COM OUTRAS ÁREAS DA EMPRESA

A área de TI possui ligação direta com todas as outras áreas da empresa: Administração, Finanças, Recursos Humanos, Vendas, Relatórios e Clipping (área responsável pelos recortes).

A organização desses setores com seus colaboradores e cargos pode ser visualizada no organograma da corporação (Figura 1).

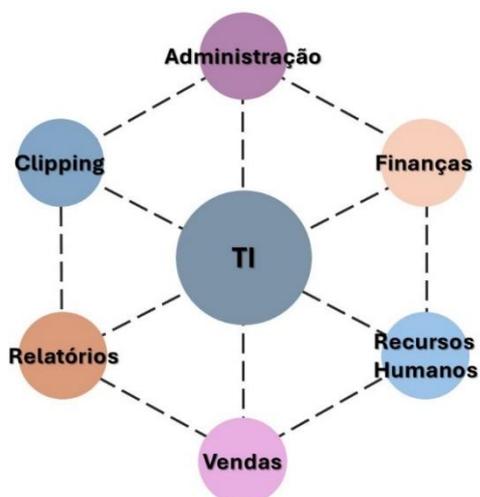
Figura 1 – Organograma da empresa SMR CLIPPING



Fonte: produção do próprio autor, 2024.

Esse fluxo interligado é fundamental para garantir boa comunicação e maior produtividade da empresa (Figura 2).

Figura 2 – Relação entre a área de Tecnologia da Informação (TI) e as demais áreas da empresa



Fonte: produção própria do autor, 2024.

Por exemplo a área administrativa colabora com a TI em tarefas como documentação, contratos que precisam de serviços específicos da TI e configuração para disparos de *e-mails* automáticos que são configurados no sistema da empresa. O mesmo acontece com o setor de Vendas, que depende da TI para verificar se há possibilidades de atender as demandas exigidas pelos clientes antes de concluírem um fechamento de negócios.

O setor de Clipping é o principal serviço da empresa, por isso a comunicação entre ele e a TI é constante. Enquanto a TI gerencia o sistema da empresa e desenvolve códigos de linguagem de programação para capturar as notícias corretamente em *websites*, rádios e canais de televisão, os colaboradores do setor de Clipping se comunicam de forma direta com os clientes quando existe a necessidade de tirar dúvidas de preferências ou mudanças específicas que possam ser feitas referente à captação das notícias.

Além do trabalho de comunicação com os clientes, o setor de Clipping também verifica as notícias capturadas, realizando a correção de pequenos erros de escrita que podem surgir por conta de alguma interpretação incorreta durante a transcrição. Por exemplo, quando o áudio de uma rádio está sendo transcrito, frases ou palavras faladas rapidamente podem ser entendidas de forma errada. Por conta desses motivos mencionados anteriormente que a comunicação entre essas duas áreas deve ser sempre clara e objetiva, evitando que surjam problemas futuros.

3.3 DESCRIÇÃO DO ESTÁGIO E DAS PRINCIPAIS ATIVIDADES REALIZADAS

O principal foco desta seção é evidenciar as atividades realizadas pelo estagiário durante sua jornada de trabalho. Serão abordados conceitos sobre a atividade de importação, onde terá fluxogramas com informações referentes ao passo a passo utilizado e forma de planejamento para execução das atividades. Abordaremos também a atividade de criação das automações, trazendo informações sobre linguagens de programação utilizadas e fluxograma para representar o passo a passo utilizado para executar a construção de uma automação.

3.3.1 O estágio

O início das atividades se deu no dia 24 de novembro de 2023, com encerramento no dia 10 de dezembro de 2024. A jornada de estágio, que durou mais de 1 ano, era de 6 horas diárias durante 5 dias da semana, com o total de 30 horas semanais. O estágio foi vinculado à disciplina de Projeto Integrador e Práticas Profissionais II, fazendo parte do Projeto Pedagógico de Curso (PPC). O objetivo desta disciplina é proporcionar ao estudante uma imersão no

mercado de trabalho, fazendo com que o discente tenha um contato direto com atividades práticas e proporcionando uma melhor experiência.

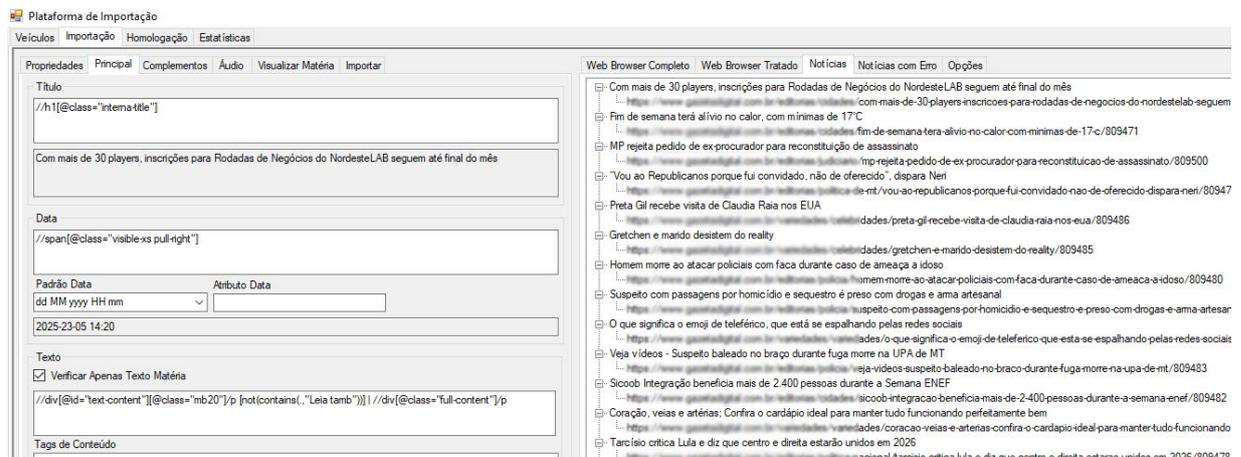
3.3.2 Importação

Como responsável pela importação¹⁴ de notícias em *websites*, meu trabalho começava quando auxiliares de Clipping enviavam novas demandas, pedidos de correção de algum veículo¹⁵ ao grupo de importação do Whatsapp ou quando era solicitado por algum profissional do setor de TI para realizar automações de trabalhos repetitivos quando era percebida alguma oportunidade.

Ao receber as demandas dos auxiliares de Clipping, faço a inserção das informações em uma planilha *online* do Google para obter um melhor controle das importações solicitadas e realizadas. A planilha contém as colunas ID, veículo, *Link*, Pedido (se é para Criar ou Corrigir), Data do pedido, Situação (vazio se ainda não estiver sido feito, Criado ou Corrigido) e Ok no grupo (para confirmar o *feedback* no grupo em relação ao resultado da demanda solicitada).

Com as informações da demanda inseridas na planilha, verifico se é um pedido de correção ou criação de uma importação. Caso seja para correção, abro a Plataforma de Importação (PI), como demonstrado na Figura 3, pesquiso pelo nome do veículo e verifico a situação analisando a quantidade de notícias enviadas ao sistema *web* da SMR CLIPPING (SMI) e verifico se há erros.

Figura 3 – Plataforma de Importação (Utilizando apenas códigos XPATH)



Fonte: produção própria do autor, 2024.

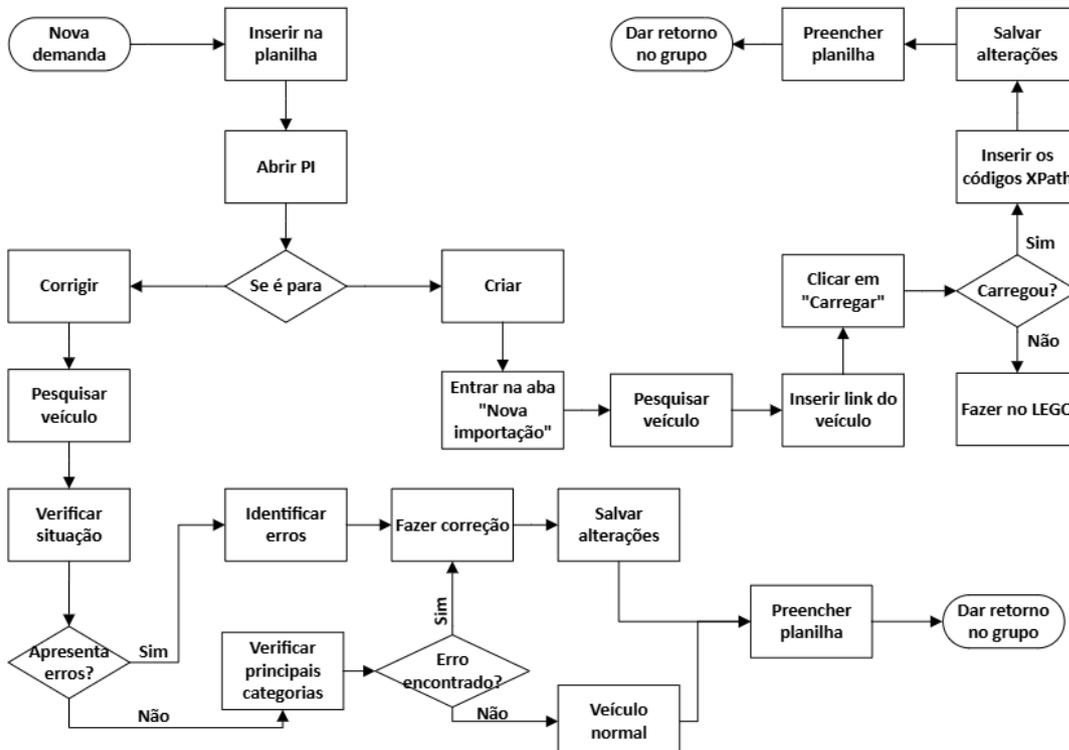
¹⁴ Importação: Processo de captura, recortes e envio de notícias ao SMI.

¹⁵ Veículo: Termo usado para identificar *website*, rádio ou canal a ser importado.

Se na Plataforma de Importação houver erros de captura ou envio de notícias ao SMI, faço a identificação do erro, que geralmente está na captura da *Uniform Resource Locator* (URL¹⁶) de cada notícia, do título ou do texto da matéria que foi publicada. Após fazer a identificação do erro que está ocorrendo, realizo a correção do código XPATH e logo após faço o salvamento.

Se a demanda solicitada for para criar, entro na aba “Nova Importação” e adiciono o *link* do *site* de notícias específico, geralmente a página de notícias mais recentes. Se a PI conseguir carregar o *website*, faço o preenchimento de todos os campos necessários e realizo o salvamento. Caso o portal de notícias seja um *site* dinâmico, não será possível realizar a importação através da PI, pois esse sistema foi construído utilizando alguma biblioteca C# que se assemelha à biblioteca BeautifulSoup do Python, onde não é possível realizar a captura dos elementos por conta que a biblioteca apenas faz uma requisição ao *site* e captura todo o HTML, inviabilizando a captura de elementos de *websites* dinâmicos por serem carregados de acordo com a interação do usuário. A metodologia utilizada para realização de importações na PI está representada na Figura 4.

Figura 4 – Fluxograma de importação utilizando a PI



Fonte: produção própria do autor, 2024.

¹⁶ URL: Endereço único que identifica um recurso na *web*.

Se a situação mencionada acontecer, será necessário fazer criação da importação através do sistema de importação criado em C# (LEGO), apresentado na Figura 5, que utiliza uma biblioteca semelhante à Selenium do Python, tornando possível a captura de elementos de *websites* dinâmicos, por conta disso, os veículos considerados mais importantes majoritariamente estavam neste sistema. Enquanto a PI é um sistema com *Graphical User Interface* (GUI), o LEGO é um sistema criado na linguagem C# com Interface de *Command Line Interface* (CLI), dificultando um pouco mais o trabalho e consequentemente fazendo com que eu aprendesse essa linguagem de programação em um nível intermediário com ajuda do meu coordenador.

Figura 5 – Sistema LEGO (utilizando linguagem C#)

```

66     new WaitContentBlock("//*[@h1][1]")
67     .AddNextBlock(
68         new WaitBlock(3000))
69     .AddNextBlock(
70         new DefaultExtractorBlock(
71             "//html",
72             new List<string>() {
73                 //"editoria",
74                 "titulo",
75                 "subtitulo",
76                 //"autor",
77                 "data",
78                 "@padrao_data",
79             },
80             new List<string>() {
81                 //"p[contains(@class,\"hat mb-16 text-hat\")]",
82                 "(//h1)[1]",
83                 "//p[contains(@class,\"text-excerpt\")]",
84                 "//ul[@class=\"entry-meta clearfix\"]/li/a[1]",
85                 "//small[contains(@class,\"text-created-at\")]",
86                 "dd MM yyyy HH mm",
87             }
88         ))
89     .AddNextBlock(
90         new DefaultExtractorBlock(
91             "//html",
92             new TextArticleExtract(),
93             new List<string>() { "texto" },
94             new List<string>() { "//div[@class=\"section-text\"]/*[self::p or self::h2 or self::h3 or self::h4 or :

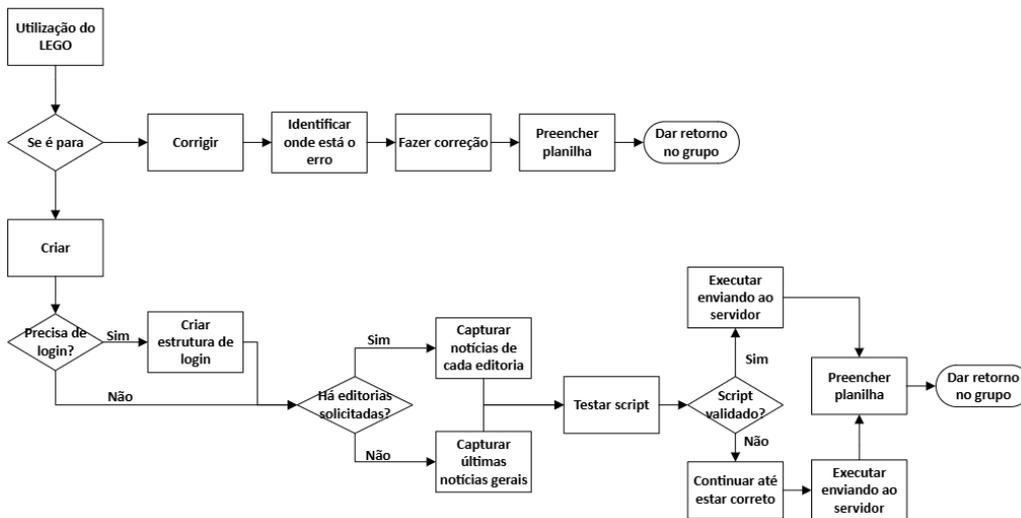
```

Fonte: produção própria do autor, 2024.

Pensando estrategicamente, dois modelos de código completo de importação foram preparados, um com *login* e outro sem, ajudando a diminuir o tempo de realizar cada importação. A primeira ação realizada no LEGO é verificar se será necessário logar no *site* para conseguir acessar o conteúdo das notícias. Se for necessário, começo a criação do código pela etapa de *login*. Quando não é necessário, vou diretamente para a criação dos códigos de captura dos *links* e conteúdo de cada notícia. Após realizar os testes e observar que tudo está funcionando corretamente, faço o envio do código para o servidor que executa esses *scripts* 24 horas por dia.

Quando é necessário apenas fazer a correção da importação no LEGO, faço um teste executando o *script* criado para ver se há algum erro na captura dos elementos, assim como na PI, o erro costuma estar no XPATH da URL, título ou texto da matéria. Após identificar o erro, faço a correção garantindo que os elementos estão sendo capturados corretamente e faço o envio do código ao servidor, atualizando automaticamente para o novo *script*. Para garantir uma melhor visualização, criei um fluxograma, que é possível observar na Figura 6, dessas etapas.

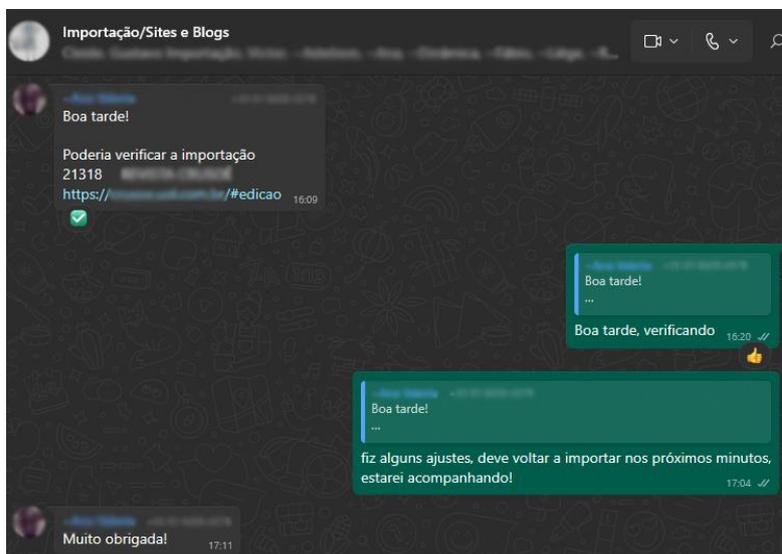
Figura 6 – Fluxograma de importação utilizando o sistema LEGO (continuação da Figura 5)



Fonte: produção própria do autor, 2024.

Após todos esses processos citados, é feita a atualização na planilha de importação e retorna-se as informações ao grupo do Whatsapp (Figura 7).

Figura 7 – Retorno ao grupo do Whatsapp



Fonte: produção própria do autor, 2024.

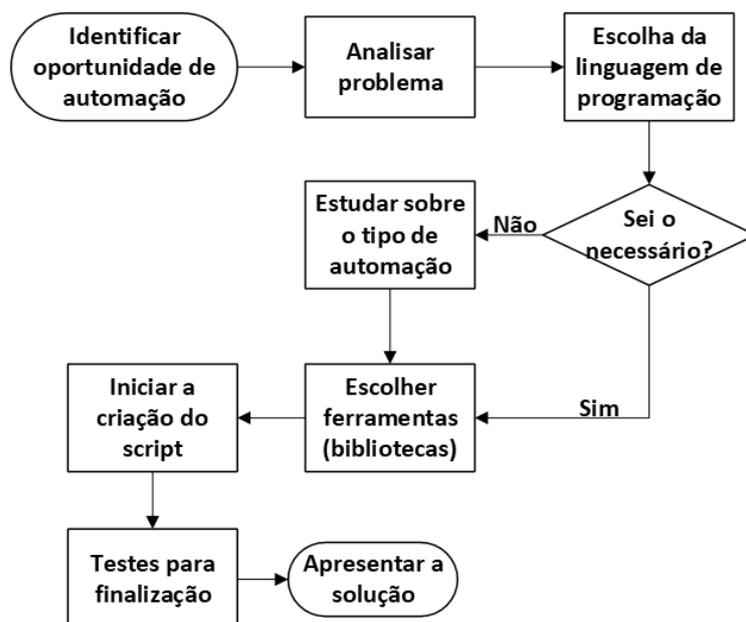
3.3.3 Criação de scripts para automação de processos

Durante meu período de trabalho, eu percebia que havia algumas tarefas que eram feitas manualmente, de forma repetitiva e que demoravam bastante. Por conta disso, comecei a analisar quais tarefas poderiam ser automatizadas para trazer um melhor desempenho e produtividade para as equipes, que ao invés de estarem dedicando parte de seu tempo em atividades repetitivas, estariam focadas em trabalhos que exigem mais atenção e que entregam maior valor.

Utilizando Python, C#, *batch script* e PowerShell criei algumas automações que foram muito elogiadas pelo meu gestor e coordenadores. Um exemplo de uma dessas automações foi a criação de um *script* usando *batch script* para verificar se um dos sistemas principais da empresa estava sendo executado. Se por algum motivo o programa estivesse fechado, o *script* que desenvolvi identifica e executa novamente o programa, garantindo o funcionamento sem interrupções, evitando que o sistema da empresa fique parado sem que ninguém perceba rapidamente.

Esse processo de analisar a possibilidade de criar uma automação para tarefas repetitivas está sendo representado graficamente no fluxograma (Figura 8) a seguir, mostrando de forma detalhada desde a parte da reflexão para a construção do *script* até a finalização e validação da automação.

Figura 8 – Fluxograma de passos para criação de *scripts* para automação de processos



Fonte: produção própria do autor, 2024.

4 RESULTADOS E DISCUSSÃO

Neste tópico serão apresentadas as principais oportunidades de desenvolvimento de automações de tarefas, os resultados obtidos tanto na criação de automação quanto nos processos de importação de notícias realizadas na empresa e uma discussão comparando os resultados obtidos com os resultados de outros pesquisadores da área.

4.1 PROGRAMA COM FECHAMENTO ALEATÓRIO

Um dos primeiros *scripts* de automação desenvolvidos solucionou um problema em um dos sistemas da empresa: o programa que gerenciava o envio de notícias da PI ao SMI encerrava aleatoriamente. De acordo com o gestor, isso resultava na perda de aproximadamente 500 notícias até que algum funcionário percebesse e relatasse o problema.

A forma mais direta de resolver esse problema foi utilizando o *batch script*, visto que o sistema operacional da empresa é o Windows. Conforme demonstrado na Figura 9, o *script* verifica a cada 3 segundos se o programa de envio das notícias da PI está ativo; caso não, o *script* o reinicia automaticamente. Essa ação de verificação e execução está dentro de um laço de repetição do tipo *loop*, fazendo com que seja executada infinitamente até que receba algum comando para interrupção.

Figura 9 – *Batch script* utilizado para manter o programa de importação aberto

```

1 @echo off
2 REM transformar em utf8
3 chcp 65001 >nul
4
5 color 04
6 echo
7 echo
8 echo
9 :start
10 REM Verifica se o programa de importação está em execução
11 tasklist /FI "IMAGENAME eq importacaoVeiculos-v2.exe" | find /i
    "importacaoVeiculos-v2.exe" > nul
12
13 if errorlevel 1 (
14     REM Se o programa de importação não estiver em execução, executa o
    inicializador
15     start "" "C:\Sistema_smr3\importacao-v2\importacaoVeiculos-v2.
    application"
16
17     REM Aguarda 3 segundos
18     timeout /t 3 /nobreak
19
20     REM Verifica se o arquivo "importacaoVeiculos-v2.exe" está em execução
21     tasklist /FI "IMAGENAME eq importacaoVeiculos-v2.exe" | find /i
    "importacaoVeiculos-v2.exe" > nul
22
23 if errorlevel 1 (
24     REM Se o arquivo não estiver em execução, abre novamente o programa
25     start "" "C:\Sistema_smr3\importacao-v2\importacaoVeiculos-v2.
    application"
26 )
27 )
28 goto start

```

Fonte: produção própria do autor, 2024.

Os encerramentos inesperados do programa de gerenciamento de envio das notícias acontecem em uma média de 3 vezes por dia, a cada “Aguardando 0 segundos[...]” (Figura 10) ocorreu um fechamento inesperado do programa, totalizando em uma média de 1500 notícias que eram perdidas diariamente. Com a solução apresentada, tornou-se possível zerar a perda de notícias por conta desses fechamentos aleatórios, resultando em um ganho de 100% em relação ao problema e utilizar essa mesma automação para garantir o funcionamento de outros programas como o TeamViewer.

Figura 10 – Script desenvolvido em *batch script* sendo executado

```

C:\Windows\system32\cmd.exe
APP IMPORT
Aguardando 0 segundos, pressione CTRL+C para sair ...
Programa que realiza a intermediação das importações!
Quantidade de Threads Ativas: 70
Quantidade de Threads em Espera: 3548
Quantidade de Threads Finalizadas: 67
Quantidade de Materias importadas no Último Ciclo: 6
Quantidade de Materias com Erro no Último Ciclo: 0
Quantidade de Materias importadas Hoje: 15899
Quantidade de Materias com Erro Hoje: 6663
  
```

Fonte: produção própria do autor, 2024.

4.2 AUTOMAÇÃO COM PYTHON E OTIMIZAÇÃO COM C#

Outro problema que estava sendo enfrentado era em relação ao bloqueio de *Internet Protocol* (IP¹⁷) do servidor que executa o sistema LEGO. Como o servidor estava sendo bloqueado pelos *websites*, vários veículos estavam sem importar notícias, causando grandes

¹⁷ IP: Conjunto de regras que governam a forma como os dados são enviados e recebidos pela *internet*.

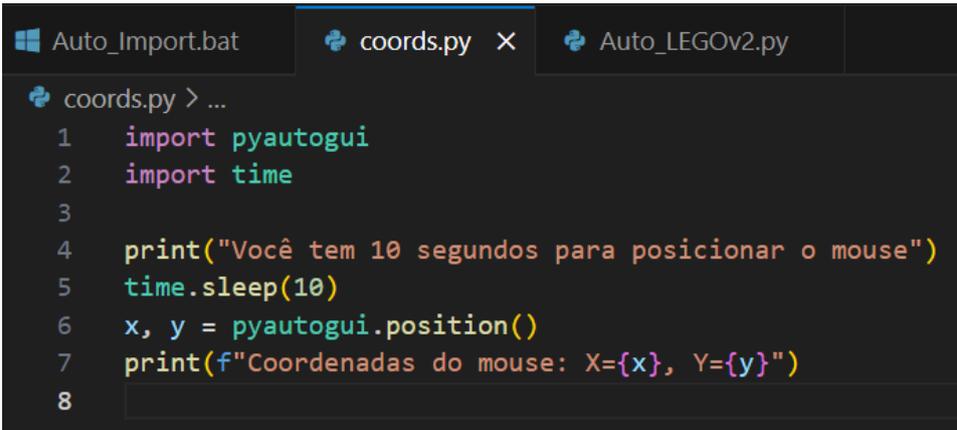
problemas, pois como foi citado anteriormente, os veículos que são importados no LEGO, majoritariamente são considerados os mais importantes.

Para contornar esse desafio, o sistema foi alocado em um computador localizado em Brasília, gerenciado de forma remota via TeamViewer pelos profissionais do setor de TI, exclusivo para os veículos que estavam com bloqueio. O problema do bloqueio de IP foi resolvido, mas logo surgiu outro. Por algum motivo desconhecido, alguns veículos da lista que era executada, estavam permanecendo em execução infinita, surgindo a necessidade de o processo ser interrompido e iniciado manualmente, impossibilitando também o uso de *loop* na execução do código.

Por conta da quantidade de veículos que eram executados, a duração da execução de um único ciclo de importação durava aproximadamente 2 horas, fazendo com que um funcionário a cada fim de ciclo do processo de importação, se conectasse com a máquina e iniciasse novamente o código.

Com todo esse cenário, percebi a oportunidade de criar mais uma automação para poder resolver esse impasse que havia surgido. Como eu já possuía um bom conhecimento na linguagem Python, decidi rapidamente que iria utilizar a biblioteca PyAutoGUI para desenvolver essa automação. Criei dois *scripts*, o primeiro, que está sendo representado pela Figura 11 faz a captura das coordenadas do *mouse* na tela, possibilitando identificar as posições onde a automação irá clicar.

Figura 11 – *Script* Python utilizado para obter as coordenadas do *mouse*



```
Auto_Import.bat  coords.py  Auto_LEGOv2.py
coords.py > ...
1  import pyautogui
2  import time
3
4  print("Você tem 10 segundos para posicionar o mouse")
5  time.sleep(10)
6  x, y = pyautogui.position()
7  print(f"Coordenadas do mouse: X={x}, Y={y}")
8
```

Fonte: produção própria do autor, 2024.

O segundo *script* realiza a automação utilizando as coordenadas para clicar na tela que foram obtidas pelo *script* que vimos acima, interagindo como se fosse uma pessoa, entre os

botões que a automação irá clicar, estão os botões de iniciar e parar o processo. Essa automação pode ser conferida na Figura 12 a seguir.

Figura 12 – Script Python utilizado para automatizar a execução do sistema LEGO

```

1 import pyautogui
2 import time
3 import os
4
5 def click_coords(x, y):
6     pyautogui.moveTo(x, y)
7     pyautogui.click()
8
9 # Icone vs barra de tarefas
10 coord1_x = 72
11 coord1_y = 1060
12 # Startar LEGO
13 coord2_x = 710
14 coord2_y = 96
15 # # Clicar no windows
16 coord3_x = 22
17 coord3_y = 1061
18 # # Clicar no cmd do lego
19 coord4_x = 121
20 coord4_y = 1059
21 # # parar lego
22 coord5_x = 1142
23 coord5_y = 196
24 # 600 = 10 min
25 tempo = 900
26
27 while True:
28     time.sleep(2)
29     click_coords(coord1_x, coord1_y)
30     time.sleep(2)
31     click_coords(coord2_x, coord2_y)
32     time.sleep(3)
33     click_coords(coord3_x, coord3_y)
34
35     os.system("cls" if os.name == "nt" else "clear")
36
37     print(f"AUTO_LEGO\n\nExecutando a cada {tempo/60} minutos!")
38     for i in range(tempo, 0, -1):

```

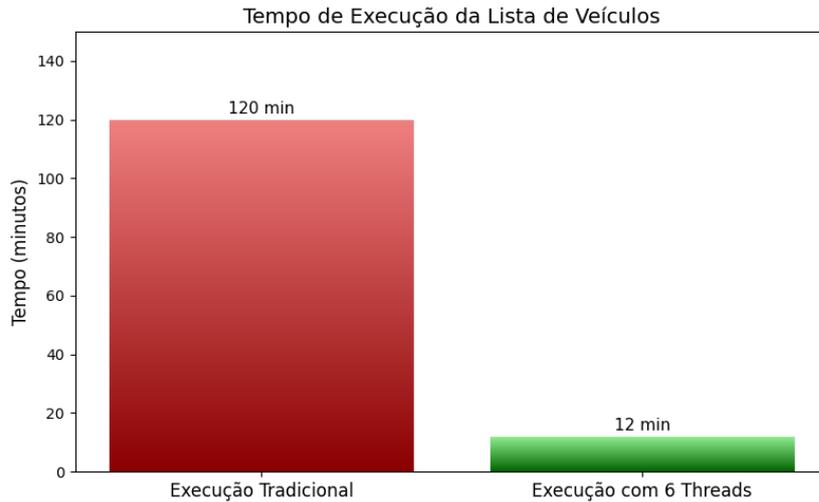
Fonte: produção própria do autor, 2024.

Com os dois códigos simples em Python foi possível resolver o problema do trabalho manual, porém a duração de execução ainda era de 2 horas, fazendo com que cada nova notícia publicada nos *websites* desses veículos, fosse cadastrada com um enorme atraso no SMI. Analisando a situação, foi feita a proposta de utilizar *multithreading*. Comecei a estudar para aplicar esta técnica de paralelismo no código C# e, no mesmo dia, consegui trazer a solução.

O ciclo completo de execução da lista de veículos que antes demorava aproximadamente 2 horas, estava sendo executado em apenas 12 minutos utilizando 6 *threads*, cada *thread* executando 1 veículo simultaneamente. Fazendo os cálculos, pela quantidade de horas que demorava antes e pelo tempo atual de 12 minutos, essa solução trouxe uma redução de

aproximadamente 90% do tempo, trazendo também, reconhecimento ao meu trabalho diante da liderança da empresa. Para uma melhor visualização observa-se o gráfico da Figura 13.

Figura 13 – Redução do tempo de execução da lista de veículos



Fonte: produção própria do autor, 2025.

4.3 AUMENTO EXPRESSIVO DE PRODUTIVIDADE

Meu ingresso na empresa aconteceu no dia 24 de novembro de 2023, tive 5 dias de instrução, após esse período comecei a realizar as importações. Como sempre fui inserindo as demandas realizadas na planilha de controle, é possível analisar o ganho de produtividade. Antes a importação estava sendo dividida entre 3 funcionários da TI (Figura 14), com a minha entrada, os colegas de equipe puderam dedicar totalmente o tempo às suas atividades principais.

Figura 14 – Planilha mostrando as importações sendo feitas pela equipe de TI

ID	Veículo	Link	Pedido	Data do Pedido	Situação	Responsável	OK no grupo
28587	Criar	03/11/2023	Criado	Victor	ok
12383	corrigir		Corrigido	Victor	ok
7935	Corrigir	03/11/2023		Letiane	
22227	Corrigir	03/11/2023	Corrigido	Gustavo	ok
37756	Corrigir	03/11/2023	Corrigido	Victor	ok
9835	Corrigir	03/11/2023	Corrigido	Victor	ok
38522	Criar	04/11/2023	Criado	Victor	ok
42801	Corrigir	06/11/2023	Corrigido	Gustavo	ok
7950	Criar	06/11/2023	Corrigido	Victor	ok
19267	Criar	06/11/2023	Corrigido	Victor	ok
32275	Corrigir	06/11/2023	Corrigido	Victor	ok
2689	Corrigir	07/11/2023	Corrigido	Victor	ok
21203	Corrigir	08/11/2023	Corrigido	Victor	ok
476	Corrigir	08/11/2023	Corrigido	Gustavo	ok
39965	Criar	08/11/2023	criado	Gustavo	ok
40529	Criar	08/11/2023	criado	Gustavo	ok
22227	Corrigir	09/11/2023	Corrigido	Victor	ok
43830	Criar	10/11/2023	Criado	Victor	ok
8318	Criar	10/11/2023	criado	Victor	ok
21034	Corrigir	10/11/2023	corrigido	Victor	ok
14350	Corrigir	13/11/2023	corrigido	Victor	ok
25643	Corrigir	13/11/2023	corrigido	Victor	ok
7935	Corrigir	13/11/2023	corrigido	Victor	ok
21984	Corrigir	13/11/2023	corrigido	Victor	ok
38653	Corrigir	13/11/2023	corrigido	Victor	ok

Fonte: produção própria do autor, 2024.

Com meu trabalho totalmente dedicado às importações e criação de automação foi possível trazer um ganho expressivo de demandas realizadas mensalmente. Por exemplo, no mês de novembro de 2023 foram realizadas 73 demandas de importação pela equipe de TI. No mês seguinte, que dei início ao meu trabalho (Figura 15), o número de pedidos finalizados foi de 226.

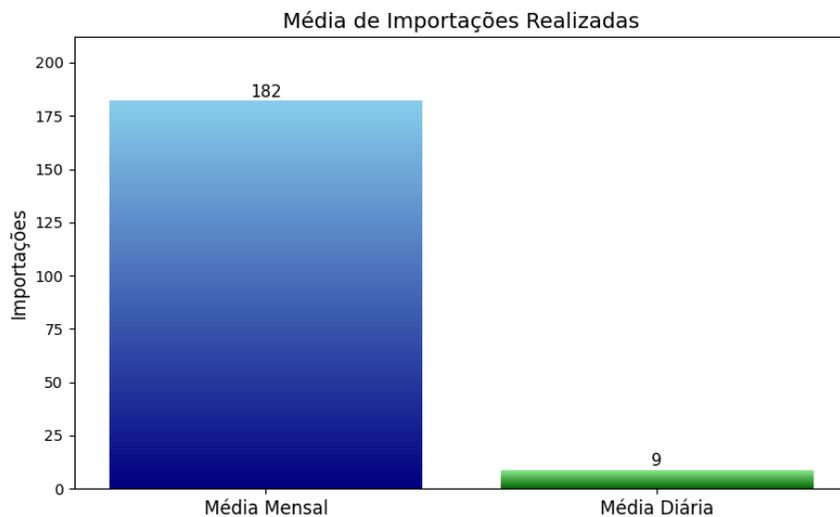
Figura 15 – Planilha mostrando as importações sendo feitas por Artur

ID	Veículo	Link	Pedido	Data do Pedido	Situação	Responsável	OK no grupo
39295	VEICULO DE TRANSPORTE - VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/39295	Criar	1/12/2023	criado	Artur	ok
1404	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/1404	Criar	1/12/2023	criado	Artur	ok
43216	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/43216	Criar	2/12/2023	criado	Artur	ok
44033	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/44033	Criar	2/12/2023	criado	Artur	ok
27209	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/27209	Criar	2/12/2023	criado	Artur	ok
33687	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/33687	Criar	2/12/2023	criado	Artur	ok
14626	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/14626	Criar	2/12/2023	criado	Artur	ok
44031	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/44031	Criar	2/12/2023	criado	Artur	ok
36029	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/36029	Corrigir	2/12/2023	criado	Artur	ok
34089	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/34089	Corrigir	2/12/2023	criado	Artur	ok
27356	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/27356	Corrigir	2/12/2023	criado	Artur	ok
30537	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/30537	Corrigir	2/12/2023	criado	Artur	ok
41694	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/41694	Criar	2/12/2023	criado	Artur	ok
16053	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/16053	Corrigir	2/12/2023	corrigido	Artur	ok
5385	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/5385	Corrigir	2/12/2023	corrigido	Artur	ok
42418	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/42418	Corrigir	2/12/2023	corrigido	Artur	ok
38936	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/38936	Criar	2/12/2023	criado	Artur	ok
35392	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/35392	Criar	4/12/2023	criado	Artur	ok
35393	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/35393	Criar	4/12/2023	criado	Artur	ok
25713	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/25713	Criar	4/12/2023	criado	Artur	ok
38933	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/38933	Criar	4/12/2023	criado	Artur	ok
9405	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/9405	Criar	4/12/2023	criado	Artur	ok
27021	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/27021	Criar	4/12/2023	criado	Artur	ok
22105	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/22105	Criar	4/12/2023	criado	Artur	ok
32264	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/32264	Criar	4/12/2023	criado	Artur	ok
29883	VEICULO DE TRANSPORTE - 08	https://www.transparencia.org.br/veiculos/29883	Criar	4/12/2023	criado	Artur	ok
Artur			Criar	4/12/2023	criado	Artur	ok

Fonte: produção própria do autor, 2024.

Ao contabilizar o número de demandas realizadas durante os 12 meses de estágio, obteve-se 2184 importações. Fazendo a divisão pelo número de meses, a média mensal foi 182. Dividindo por 20 (dias mensais que eu trabalho), encontramos uma média de 9 importações realizadas diariamente (Figura 16).

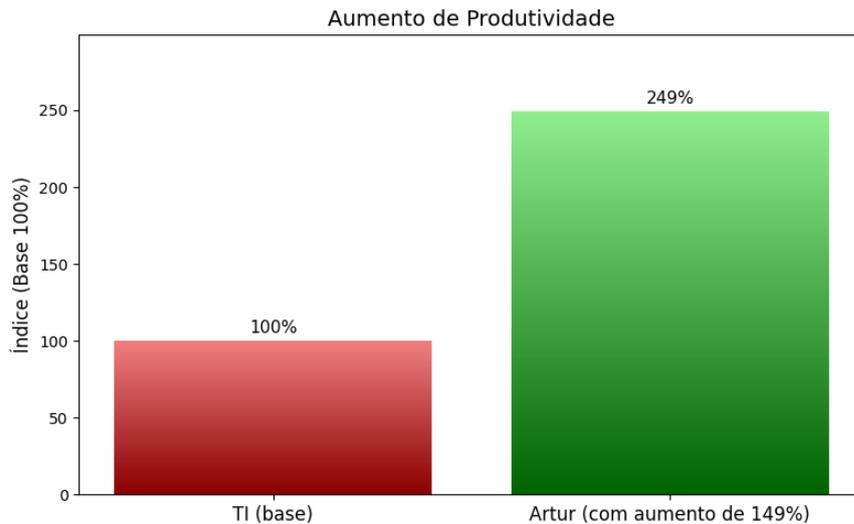
Figura 16 – Médias mensais e diárias de importações realizadas por Artur



Fonte: produção própria do autor, 2025.

Calculando a média mensal feita por mim em relação à quantidade feita pelos funcionários da TI, foi possível obter um aumento de aproximadamente 149% das demandas realizadas mensalmente (Figura 17). Durante meu estágio, recebi elogios e reconhecimento das coordenadoras de Clipping, por sempre fazer um bom trabalho, ser organizado, inovador e proativo.

Figura 17 – Aumento de produtividade



Fonte: produção própria do autor, 2025.

O presente trabalho pôde demonstrar ganhos de eficiência obtidos através do uso de automação e através do meu ingresso na empresa, como a redução de 90% no tempo de execução da lista de veículos alocados no computador em Brasília e 149% de aumento na produtividade de importações.

A eliminação da perda de aproximadamente 1500 notícias diárias através da automação por *batch script* (Seção 4.1) são ganhos consideráveis para a empresa, assim como Freitas Júnior (2021, p. 34) relatou que a automação do lançamento de notas fiscais, em um mês, processou "[...]1.036 [casos], sendo que 93,9% foram processados com sucesso e somente 6,1% foram retornados para a operação. O total de casos processados pelo robô apresentou uma economia de custo de R\$33.129,25". Essa automação também possibilitou "[...]liberar horas produtivas dos recursos da operação[...]" (Freitas Júnior, 2021, p. 37). Resultado semelhante ao trabalho que desenvolvi durante meu estágio na empresa SMR CLIPPING, que também resultou na otimização de recursos, possibilitando a realocação da equipe de TI para atividades mais estratégicas.

5 CONSIDERAÇÕES FINAIS

Através dessa oportunidade de estágio na SMR CLIPPING, pude obter experiência no mercado de trabalho, algo que para mim era muito importante, pois sei o valor que isso me traria, tanto pessoal como profissional. Além de aplicar os conhecimentos que adquiri ao longo da minha trajetória de estudos na graduação, vivenciei um ramo de mercado que ainda não conhecia, que é a clipagem de notícias, área com boa demanda e poucas empresas concorrentes, despertando em mim um lado mais empreendedor.

Desenvolvi ainda mais raciocínio lógico, pensamento crítico/analítico e criatividade por meio das análises, planejamento e desenvolvimento das automações. Foi possível aplicar meus conhecimentos em Python, aprender uma nova linguagem de programação como o C#, me aprofundar um pouco mais em *batch script* e PowerShell.

Além das *hard skills*¹⁸, aprimorei e adquiri novas *soft skills*¹⁹, como resiliência, comunicação clara e objetiva, organização e trabalho em equipe. Vi na prática como uma boa comunicação e harmonia entre equipes faz diferença no ambiente de trabalho. O estágio também me ajudou a entender a carreira que eu melhor me identifico, por conta disso, estou focado na área de engenharia de dados, onde posso continuar aplicando meus conhecimentos em automação e *web scraping*. Estou muito contente e sinto que estou muito bem direcionado.

¹⁸ *Hard skills*: Aptidões técnicas de um profissional.

¹⁹ *Soft skills*: Habilidades sociocomportamentais, ligadas diretamente às aptidões mentais de um candidato e à capacidade de lidar positivamente com fatores emocionais.

REFERÊNCIAS

- ANDRADE, J. P. F. de; DALLILO, F. D.; FLORIAN, F. Aumento de performance na extração de dados via RPA. **RECIMA21 - Revista Científica Multidisciplinar**, [S. l.], v. 3, n. 12, p. 1-13, 2022. DOI: <https://doi.org/10.47820/recima21.v3i12.2309>. Disponível em: <https://recima21.com.br/index.php/recima21/article/view/2309>. Acesso em: 22 mai. 2025.
- BAWICA, I. M. The university internship program and its effects on students' employability readiness. **International Journal of Academe and Industry Research**, Candelaria, v. 2, n. 3, p. 90-106, 2021. DOI: <https://doi.org/10.53378/348731>. Disponível em: https://iiari.org/journal_article/ijairv2-3-168. Acesso em: 22 mai. 2025.
- FARIA, M. V. M. *et al.* Aprimoramento de modelos de classificação com dados enriquecidos via webscraping: um estudo de caso da competição dog breed identification. In: ESCOLA REGIONAL DE INFORMÁTICA DO ESPÍRITO SANTO (ERI-ES), 2024, Vitória. **Anais da Escola Regional de Informática do Espírito Santo (ERI 2024)**. Porto Alegre: SBC, 2024. DOI: <https://doi.org/10.5753/eries.2024.244695>. Disponível em: <https://sol.sbc.org.br/index.php/eries/article/view/31613>. Acesso em: 23 mai. 2025.
- FREITAS JÚNIOR, V. G. de. **Automação de processos de negócio utilizando robotic process automation (RPA) em um centro de serviços compartilhados (CSC): um estudo de caso**. 2021. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) - Universidade Federal de Uberlândia, Uberlândia, 2021. Disponível em: <https://repositorio.ufu.br/handle/123456789/33758>. Acesso em: 22 mai. 2025.
- KOSTER, M. *et al.* **RFC 9309: Robots Exclusion Protocol**. United States: RFC Editor, 2022. DOI: <https://doi.org/10.17487/RFC9309>. Disponível em: <https://www.rfc-editor.org/info/rfc9309>. Acesso em: 22 mai. 2025.
- LUTFIA, D. D.; RAHADI, D. R. Analisis intership bagi peningkatan kompetensi mahasiswa. **Jurnal Ilmiah Manajemen Kesatuan**, [S. l.], v. 8, n. 3, p. 199-204, 2020. DOI: <https://doi.org/10.37641/jimkes.v8i3.340>. Disponível em: <https://jurnal.ibik.ac.id/index.php/jimkes/article/view/340>. Acesso em: 22 mai. 2025.
- MARRES, N.; WELTEVREDE, E. Scraping the social?: issues in live social research. **Journal of Cultural Economy**, London, v. 6, n. 3, p. 313-335, 2013. DOI: <https://doi.org/10.1080/17530350.2013.772070>. Disponível em: <https://www.tandfonline.com/doi/abs/10.1080/17530350.2013.772070>. Acesso em: 22 mai. 2025.
- MITCHELL, Ryan. **Web scraping com Python: coletando mais dados na web moderna**. Tradução de Lúcia A. Kinoshita. 2. ed. São Paulo: Novatec, 2019. ISBN 978-85-7522-734-3. Título original: *Web scraping with Python: collecting more data from the modern web*.
- NASCIMENTO, L. F. Combinando webscraping em R e ATLAS.ti na pesquisa em ciências sociais: as possibilidades e desafios da sociologia digital. In: CONGRESSO BRASILEIRO DE SOCIOLOGIA, 18., 2017, Brasília. **Anais do 18º Congresso Brasileiro de Sociologia**. Brasília: Sociedade Brasileira de Sociologia, 2017. 3-17. Disponível em: https://www.researchgate.net/publication/317343570_Combinando_webscraping_em_R_e_A

TLAS*ti*_na_pesquisa_em_ciencias_sociais_as_possibilidades_e_desafios_da_sociologia_digital. Acesso em: 23 mai. 2025.

SARKER, I. H. Data science and analytics: an overview from data-driven smart computing, decision-making and applications perspective. **SN Computer Science**, Singapore, v. 2, p. 1-22, 2021. DOI: <https://doi.org/10.1007/s42979-021-00765-8>. Disponível em: <https://link.springer.com/article/10.1007/s42979-021-00765-8>. Acesso em: 23 mai. 2025.