

Maxmyller Ferreira de Freitas Carvalho

Automatização por Robô de Software para um Sistema Contábil

Natal – RN

Dezembro de 2020

Maxmyller Ferreira de Freitas Carvalho

Automatização por Robô de Software para um Sistema Contábil

Trabalho de Conclusão de Curso de Engenharia de Computação da Universidade Federal do Rio Grande do Norte, apresentado como requisito parcial para a obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Diogo Pinheiro Fernandes Pedrosa

Universidade Federal do Rio Grande do Norte – UFRN
Departamento de Engenharia de Computação e Automação – DCA
Curso de Engenharia de Computação

Natal – RN
Dezembro de 2020

Universidade Federal do Rio Grande do Norte - UFRN
Sistema de Bibliotecas - SISBI
Catalogação de Publicação na Fonte. UFRN - Biblioteca Central Zila Mamede

Carvalho, Maxmyller Ferreira de Freitas.

Automatização por robô de software para um sistema contábil /
Maxmyller Ferreira de Freitas Carvalho. - 2020.
48 f.: il.

Monografia (Graduação) - Universidade Federal do Rio Grande
do Norte, Centro de Tecnologia, Curso de Engenharia de
Computação, Natal, RN, 2020.

Orientador: Diogo Pinheiro Fernandes Pedrosa.

1. Robótica - Monografia. 2. Robotic Process Automation - RPA
- Monografia. 3. Python - linguagem - Monografia. 4. Automação -
processos - Monografia. I. Pedrosa, Diogo Pinheiro Fernandes.
II. Título.

RN/UF/BCZM

CDU 621.865.8

AGRADECIMENTOS

A Deus, pelo dom da vida e por tudo que tem me proporcionado. Sem Ele, nada disso estaria acontecendo.

À minha família, por todo suporte e carinho que me deram durante toda minha vida, em especial aos meus pais que sempre me incentivam e são minha fonte de inspiração.

Ao meu amor Jhenire, por estar sempre ao meu lado me apoiando e passando confiança e positividade nos momentos que eu mais preciso.

Ao meu orientador Prof. Dr. Diogo Pinheiro pelo empenho e disposição em me auxiliar durante o tempo do trabalho.

Aos meus amigos de curso e de estágio, que sempre estiveram ao meu lado, pela amizade incondicional e pelo apoio demonstrado ao longo de todo o período acadêmico, em especial Jordan Soares, Maria Luiza Cartaxo, Glen Martin, Leise Caroline, Elder Santos, Bruno Rossi e Matheus Dantas.

À empresa BWA Global, por ter acreditado em mim e ter me escolhido para o desenvolvimento desse projeto importante.

E à Universidade Federal do Rio Grande do Norte, essencial no meu processo de formação profissional e por tudo o que aprendi ao longo de todo o curso. Meu muito obrigado.

*“Você pode encarar um erro como uma besteira a ser esquecida,
ou como um resultado que aponta uma nova direção.”*

Steve Jobs

RESUMO

Durante a pandemia da Covid-19, todos os setores da sociedade foram atingidos negativamente. Diversas empresas passaram a ter problemas decorrentes da crise econômica que se agravou no país. Com o setor de serviços também não foi diferente. A crise gerada fez com que a manutenção de empregos fosse prejudicada. Diante disso, foi apresentada uma ideia de substituir funcionários por uma ferramenta de software automatizada na execução de tarefas rotineiras. Ela foi aplicada em um escritório de contabilidade no município de Natal, RN e implementada na linguagem Python juntamente com suas bibliotecas. O projeto desenvolvido foi a automatização em RPA (Robotic Process Automation) de um dos serviços contábeis chamado “SEFIP - Conectividade Social” da Caixa Econômica Federal. RPA nada mais é do que uma aplicação tecnológica que busca mecanizar, por via robô de software, as atividades que são demasiadamente repetitivas e massivas para um humano fazer. 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-chaves: RPA. Python. Automação. Robótica.

ABSTRACT

During the Covid-19 pandemic, all sectors of society were negatively affected. Several companies started to have problems resulting from the economic crisis that worsened in the country. It was no different with the service sector. The crisis generated has meant that the maintenance of jobs has been hampered. Therefore, an idea was presented to replace employees with an automated software tool in the performance of routine tasks. It was applied in an accounting office in Natal, RN, and implemented in the Python language together with its libraries. The project developed was the automation in RPA (Robotic Process Automation) of one of the accounting services called “SEFIP - Conectividade Social” of Caixa Econômica Federal. Robotic Process Automation is a technological application that seeks to mechanize, by software robot, activities that are very repetitive and massive for a human to do. The benefit of this implementation has made it easier to manage employees' schedules, using them in more strategic tasks, in other words, making gains of productivity and quality for the business.

Keywords: RPA. Python. Automation. Robotic.

LISTA DE ILUSTRAÇÕES

Figura 1 – Benefícios da tecnologia RPA no mercado de trabalho	13
Figura 2 – Mercado global de RPA 2017-2023	14
Figura 3 – Fluxograma do projeto	18
Figura 4 – <i>Layout</i> da planilha construída	20
Figura 5 – Aplicação SEFIP	23
Figura 6 – Arquivo “Sefip.re” inserido na pasta de uma empresa específica	25
Figura 7 – Janela “Impressão de Relatório”	27
Figura 8 – Relatório de inconsistência	27
Figura 9 – “Relatórios de Movimento” para <i>download</i>	28
Figura 10 – Pastas auxiliares para receber temporariamente os arquivos gerados	29
Figura 11 – Pasta de uma empresa com os arquivos gerados após a execução do robô SEFIP	30
Figura 12 – Janela do certificado digital para o acesso ao portal Conectividade Social	31
Figura 13 – Página principal do site Conectividade Social	33
Figura 14 – Pasta de uma empresa com os arquivos gerados após a execução do robô Conectividade	34
Figura 15 – Aplicação SEFIP - Importação do arquivo de extensão .XML	35
Figura 16 – Aplicação SEFIP - Janela “Erro” indicando empresas pró-labore (sem arquivo GRF)	36
Figura 17 – Aplicação SEFIP - Janela “Informação” indicando a geração do arquivo GRF	36
Figura 18 – Gráfico SEFIP - Unidade Natal	38
Figura 19 – Gráfico CONECTIVIDADE - Unidade Natal	39
Figura 20 – Gráfico SEFIP - Unidade Recife	40
Figura 21 – Gráfico CONECTIVIDADE - Unidade Recife	41
Figura 22 – Gráfico SEFIP - Unidade Rio de Janeiro (Barra)	42
Figura 23 – Gráfico CONECTIVIDADE - Unidade Rio de Janeiro (Barra)	43
Figura 24 – Gráfico SEFIP - Unidade Rio de Janeiro (Penha)	44
Figura 25 – Gráfico CONECTIVIDADE - Unidade Rio de Janeiro (Penha)	45

LISTA DE TABELAS

Tabela 1 – Flags das colunas “Status SEFIP” e “Status Conectividade” com seus significados	21
Tabela 2 – Flags da coluna “Observações” e suas ações	22
Tabela 3 – Status SEFIP - Unidade Natal	38
Tabela 4 – Status CONECTIVIDADE - Unidade Natal	38
Tabela 5 – Status SEFIP - Unidade Recife	40
Tabela 6 – Status CONECTIVIDADE - Unidade Recife	40
Tabela 7 – Status SEFIP - Unidade Rio de Janeiro (Barra)	42
Tabela 8 – Status CONECTIVIDADE - Unidade Rio de Janeiro (Barra)	42
Tabela 9 – Status SEFIP - Unidade Rio de Janeiro (Penha)	44
Tabela 10 – Status CONECTIVIDADE - Unidade Rio de Janeiro (Penha)	44

LISTA DE ABREVIATURAS E SIGLAS

RPA	<i>Robotic Process Automation</i>
SEFIP	<i>Sistema Empresa de Recolhimento do FGTS e Informações à Previdência Social</i>
JSON	<i>JavaScript Object Notation</i>
XML	<i>Extensible Markup Language</i>
CNPJ	<i>Cadastro Nacional da Pessoa Jurídica</i>
FAP	<i>Fator Acidentário de Prevenção</i>
GRF	<i>Guia de Recolhimento do FGTS</i>
PDF	<i>Portable Document Format</i>

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Motivação	12
1.2	Objetivos	12
1.3	Estrutura do Trabalho	12
2	FERRAMENTAS PARA DESENVOLVIMENTO DO PROJETO	13
2.1	<i>RPA - Robotic Process Automation</i>	13
2.2	<i>Serviço SEFIP - Conectividade Social</i>	14
2.3	<i>Linguagem Python</i>	15
2.3.1	PyautoGUI	16
2.3.2	Selenium	16
2.3.3	PyGetWindow	16
2.3.4	Google Sheets	17
3	METODOLOGIA	18
3.1	Importação da fonte de dados	19
3.1.1	Organização da planilha	20
3.2	Robô SEFIP	22
3.2.1	Existência de pendências	23
3.2.2	Existência do arquivo 'Sefip.re'	25
3.2.3	Erros de usuário	26
3.2.4	Geração dos arquivos	28
3.2.5	Realizar backup	30
3.3	Robô Conectividade Social	30
3.3.1	Existência de pendências	30
3.3.2	Certificado digital	31
3.3.3	Geração do arquivo da Conectividade	34
3.3.4	Importação do arquivo .XML no SEFIP	34
4	RESULTADOS OBTIDOS	37
4.1	Unidade Natal	38
4.2	Unidade Recife	40
4.3	Unidade Rio de Janeiro - Barra da Tijuca	42
4.4	Unidade Rio de Janeiro - Penha	44
5	CONCLUSÃO	46

REFERÊNCIAS 48

1 INTRODUÇÃO

As últimas décadas vêm sendo marcadas por um grande avanço na tecnologia, inserida em muitas empresas dos mais variados segmentos da indústria. A cada dia que passa, o avanço tecnológico alavanca o desempenho das companhias. Por exemplo, a criação de robôs na montagem de carros foi uma revolução no setor automobilístico que elevou a produtividade das concessionárias, tornando a vida industrial mais automatizada. Partindo dessa premissa, a automatização vem sendo aplicada na execução de tarefas monótonas dentro de escritórios para gestão de processos administrativos. Deveras essa é a corrente atual no Brasil e também em outros países do mundo, ganhando atenção no período da pandemia da Covid-19.

O RPA, do termo em inglês *Robotic Process Automation*, é uma tecnologia voltada a automatizar processos de negócio, que lida com a configuração e a gestão de “robôs” de software que agem como usuários de aplicações sintéticas, automatizando tarefas operacionais repetitivas e estruturadas nos sistemas da organização (QUALITAT, 2020). Os robôs, também conhecido por *bots*, têm apresentado soluções nos segmentos financeiros, jurídicos, TI, viagens, educação, entre outros (CASTRO, 2020).

A otimização de custos e tempo, a redução de riscos e flexibilidade são benefícios valiosos na sua implantação. Os funcionários que se dedicavam antes às tarefas monótonas, agora têm mais disponibilidade para outras atividades mais estratégicas que necessitam de planos e análises rebuscadas. Os robôs programados eliminam os possíveis equívocos que são gerados pelos humanos devido ao cansaço e ao déficit de atenção. E por ser flexível, o robô pode ser aplicado em diferentes escalas. Ou seja, os robôs de software podem ser replicados em mais máquinas para fazer a mesma tarefa com diferentes dados durante 24 horas por dia e 7 dias por semana, sem interrupções e acelerando o processo desejado.

Segundo a empresa de consultoria norte-americana Gartner Group (GARTNER, 2020), a pandemia e a consequente recessão econômica aumentaram o interesse de muitas empresas em RPA. Ela prevê que em até 2022, 90% das médias e grandes empresas terão adotado esse tipo de automatização de alguma forma. E ainda complementou fazendo a seguinte citação:

“The decreased dependency on a human workforce for routine, digital processes will be more attractive to end users not only for cost reduction benefits, but also for insuring their business against future impacts like this pandemic.” (GARTNER, 2020)

1.1 Motivação

O motivo a ter levado a criação desse projeto surgiu em meio a pandemia da Covid-19, quando funcionários de um escritório de contabilidade estavam sendo demitidos e, diante desse problema, viu-se a necessidade de desenvolver robôs de software para substituí-los em tarefas monótonas. A implementação de técnicas, a tomada de estratégias, a manipulação de strings e o uso de bibliotecas foram necessários para que o seu desenvolvimento fosse feito com sucesso. Inclusive, a motivação também surgiu como desafio ao colocar em prática, nesse projeto, os conhecimentos adquiridos em meio acadêmico.

1.2 Objetivos

O objetivo principal desse trabalho foi a criação de um robô de *software* para ser implantado na empresa BWA Global, realizando o serviço contábil chamado “SE-FIP–Conectividade Social” da Caixa Econômica Federal. De maneira estratégica, ele é dividido em dois: robô SEFIP e robô Conectividade, fazendo a permuta de informações entre sistemas como se um usuário comum estivesse atuando.

Para auxiliar na conclusão do objetivo fundamental, foi feito o uso da API do Google Sheets para a captura dos dados de entrada das empresas, inclusive mapeando os status de funcionamento do robô como um *dashboard*, para acompanhar com circunspeção a situação em tempo real do *software*.

1.3 Estrutura do Trabalho

Este trabalho apresenta uma introdução sobre o tema, mostrando os aspectos que estimulam o interesse por essa ideia, além da justificativa, e dos objetivos. Depois, o Capítulo 2 apresenta a fundamentação teórica, necessária para contextualizar o leitor sobre os programas e ferramentas usados para a construção do trabalho. Em seguida, o Capítulo 3 relata a metodologia adotada no projeto, mostrando o passo-a-passo e a estratégia tomada para sua estruturação. O Capítulo 4 trata das análises dos resultados obtidos, dos arquivos gerados. E, por fim, o Capítulo 5 aborda as conclusões dos resultados apresentados e quais as contribuições deste trabalho.

2 FERRAMENTAS PARA DESENVOLVIMENTO DO PROJETO

Nesse capítulo será abordado os principais assuntos necessários para compreensão desse projeto, ou seja, definindo as acepções importantes sobre o serviço contábil SEFIP-Conectividade, as tecnologias usadas no trabalho e a explicação mais detalhada do que é o RPA.

2.1 RPA - *Robotic Process Automation*

Nos últimos anos, a Automação Robótica de Processos ou *Robotic Process Automation* (RPA) vem se transformando em um dos temas mais comentados dentro das empresas. RPA é uma tecnologia criada para automatizar tarefas operacionais e de negócio repetidas e estruturadas nos sistemas da organização (QUALITAT, 2020).

Embora o significado da sigla RPA possa facilmente estimular a noção de robôs físicos executando tarefas geralmente realizadas por humanos, o conceito verídico é a automatização das referidas tarefas através de soluções de software.

Basicamente ela foi desenvolvida partindo do princípio de imitar os mesmos caminhos e procedimentos realizados pelo ser humano. E para chegar a essa premissa, é feito um mapeamento do processo pretendido, analisando as melhores estratégias para cumprir o objetivo.



Figura 1 – Benefícios da tecnologia RPA no mercado de trabalho

Fonte – Blog ConTi Consultoria

Conforme é ilustrado na Figura 1, diversos benefícios são conquistados na utilização dessa tecnologia, evoluindo qualquer empresa aumentando produtividade, melhorando as questões de *compliance* e automatizando o trabalho reduzindo o tempo e o custo das implementações típicas de software por TI.

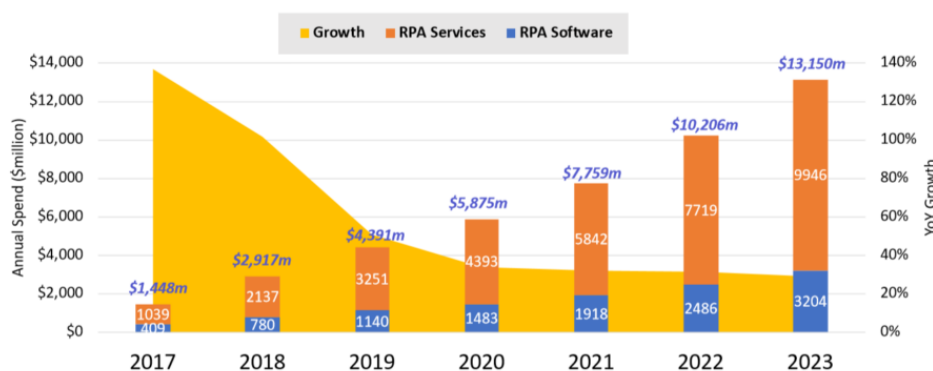


Figura 2 – Mercado global de RPA 2017-2023

Fonte – HFS Research 2019

O avanço da tecnologia está cada vez mais em foco no mercado, assim como mostra a Figura 2. O gráfico da HFS Research prevê que o mercado de software RPA crescerá mais de 30% ao ano nos próximos 3 anos, para mais de \$ 13 bilhões em 2023. Isso evidencia que o caminho para a transformação digital não tem volta. As empresas deverão promover uma mudança nos seus processos e na adoção de ferramentas de automação de processos para alavancar economicamente.

2.2 Serviço SEFIP - Conectividade Social

O Sistema Empresa de Recolhimento do FGTS e Informações à Previdência Social, conhecido como SEFIP, é um serviço desenvolvido pela Caixa Econômica Federal. A ferramenta é gratuita e realiza o processo de recolhimento regular do FGTS de maneira mais dinâmica e segura.

O sistema é destinado a todas as pessoas físicas, jurídicas e contribuintes equiparados à empresa, sujeitos ao recolhimento do FGTS. Ele é responsável por determinar os dados cadastrais e financeiros dos contribuintes e trabalhadores para repassar ao FGTS e à Previdência Social, conforme informa (CAIXA, 2013).

O SEFIP gera um arquivo contendo informações para ser transmitido à Caixa, por meio do portal Conectividade Social, que é acessível por qualquer computador conectado

à internet e com a utilização de Certificado Digital, emitido no padrão ICP-Brasil, para mais segurança no processo.

Após o envio, o Conectividade Social disponibilizará um arquivo que deve ser carregado no SEFIP para gerar o boleto de recolhimento do FGTS, chamado de Guia de Recolhimento do FGTS (GRF). Ele deve ser recolhido até o 7º dia do mês seguinte àquele em que a remuneração do trabalhador foi paga.

2.3 *Linguagem Python*

De acordo com (PYTHON.ORG, 2020), Python é uma linguagem de programação interpretada, orientada a objetos e de alto nível. Sua estrutura de dados, a possibilidade de ser usada como uma linguagem de script e para o desenvolvimento rápido de aplicativos, a tornam muito atraente. A sintaxe simples e fácil de aprender do Python enfatiza a legibilidade e, portanto, reduz o custo de manutenção do programa. O Python suporta módulos e pacotes, o que incentiva a modularidade do programa e a reutilização de código. Por ser código aberto, Python e sua extensa biblioteca estão disponíveis e podem ser distribuídos livremente.

A escolha dessa linguagem para o projeto foi pelo fato dela ter uma variedade de bibliotecas que auxilia no uso da tecnologia, como por exemplo: PyautoGUI para automatizar interações com teclado e mouse, Selenium para execução interativa em um navegador da Web, PyGetWindow para controlar as janelas dos softwares e o API do Google Sheets como fonte de dados na nuvem de fácil manutenção e monitoramento do estado do robô.

Para o desenvolvimento desse projeto foram utilizadas algumas bibliotecas para a automatização do serviço, as quais são exibidas abaixo. Dentre elas, destacam-se PyautoGUI, Selenium, PyGetWindow e API do Google Sheets, que explanaremos melhor ainda nessa seção.

```
import time
import os
import os.path
import pygetwindow
import pyautogui
import keyboard
import clipboard
import shutil
import gspread
import pyscreenshot          as ImageGrab
```

```
from threading import Thread
from datetime import date, datetime
from selenium import webdriver
from selenium.webdriver.common.by import By
from oauth2client.service_account import ServiceAccountCredentials
from pywinauto.application import Application
```

2.3.1 PyautoGUI

A biblioteca PyautoGUI, conforme relata na sua documentação (PYAUTOGUI, 2019), simula perfeitamente as ações de um usuário humano, permitindo que seja controlado o mouse e o teclado, automatizando interações com outros aplicativos.

É importante salientar que antes de usufruir este pacote, ter ciência da resolução de tela é fundamental. Isso se deve ao fato do mapeamento das ações do clique do mouse, que são guiadas pelas coordenadas x e y dos pixels. Mediante o exposto, se construir um *bot* na resolução 1080p, este só funcionará em outro computador com a mesma definição de tela. Se estiver fora da resolução, as ações que exigem mapeamento não serão executadas com êxito.

2.3.2 Selenium

O Selenium é uma biblioteca que apresenta conjunto de ferramentas para automação de navegadores da Web, que usa as melhores técnicas disponíveis para controlar remotamente as instâncias do navegador e emular a interação do usuário com o navegador (SELENIUM.DEV, 2020).

Ele suporta a automação dos principais navegadores do mercado através do uso do WebDriver. WebDriver é uma função que importamos do Selenium que faz o controle do comportamento dos navegadores da Web. Cada navegador tem o seu WebDriver específico, o qual se responsabiliza por realizar as ações de comunicação com o navegador desejado.

2.3.3 PyGetWindow

O PyGetWindow é um pacote que tem funções para obter objetos de um lugar na tela, de um título da janela, ou capturar todas as janelas. Através dele, a janela pode ser manipulada de várias maneiras, ainda tendo atributos para informar sua posição, tamanho e estado atual (PYGETWINDOW, 2020).

Uma análise a destacar é que esse pacote tem a função de ser uma espécie de um sensor, ou seja, ele fica no aguardo quando uma janela for ativa ou não através do nome do título. A partir dessa condição, os atuadores começam a agir os próximos passos

demarcados, como por exemplo, um movimento do mouse, a escrita num campo de texto. Diante dessa análise, o PyGetWindow traz consigo a lembrança do conceito de Agentes Inteligentes. Em outras palavras, ele capta o que ocorre naquele ambiente em relação as janelas do computador para definir os próximos passos do software.

2.3.4 Google Sheets

Segundo (CLIENT, 2019), o Google Sheets permite que você leia e edite qualquer célula de uma planilha. Existem muitas configurações que permitem criar tabelas e gráficos estatísticos, servir de base de dados na nuvem e utilizar como um *dashboard*, monitorando e administrando todo o funcionamento do robô.

3 METODOLOGIA

Neste capítulo será relatado o desenvolvimento para a realização do projeto. A metodologia e a estratégia traçada estão definidas no fluxograma abaixo:

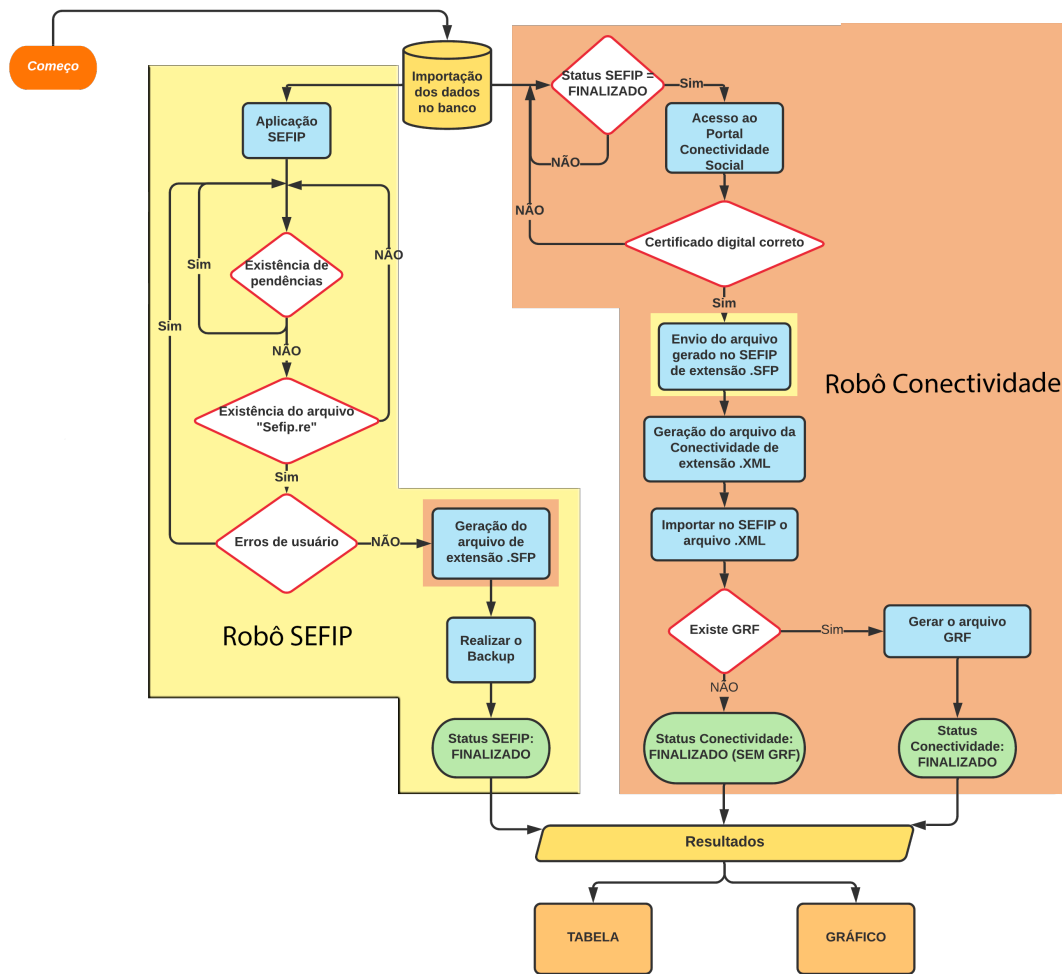


Figura 3 – Fluxograma do projeto

Fonte – Elaborado pelo autor

Vale ressaltar que a tecnologia RPA é muito importante ter um alinhamento no plano de construção, evitando que o robô perca tempo e desempenho no seu funcionamento. Nesse caso, podemos visualizar na Figura 3 que o bot do projeto SEFIP-Conectividade foi dividido em dois propositalmente, devido ao fato do portal Conectividade Social ser muito instável. E em uma situação adversa como esta, para o processo não ficar bloqueado à espera do seu retorno, o robô designado ao SEFIP fará toda a sua execução empresa por empresa, gerando o arquivo com extensão .SFP em destaque na figura acima, restando apenas o processo da Conectividade que poderá ser concretizada posteriormente.

Um outro ponto que merece destaque é que, por esse projeto ter sido percorrido dentro da empresa BWA Global, será necessário omitir determinados dados por serem sigilosos e privados.

Nas próximas seções serão explicadas em detalhes cada estado que o fluxograma apresenta nos robôs SEFIP e Conectividade.

3.1 Importação da fonte de dados

Diante do propósito desse projeto, a base de dados se tornou conveniente usar o serviço do Google Sheets, um programa de planilhas disponível na nuvem e compartilhada com os funcionários do setor responsável das quatro unidades da empresa (Natal, Recife, Rio de Janeiro - Penha e Rio de Janeiro - Barra). Devido a recente união dos escritórios para formar a BWA Global, o banco de dados ainda não se encontrava unificada. Portanto, os dados e informações das empresas que demandassem do serviço SEFIP-Conectividade eram inseridos na planilha da unidade do escritório referido para que os robôs a importassem para o ambiente de trabalho.

Para que o Google Sheets tivesse essa funcionalidade automatizada de *dashboard* monitorando, coletando e inserindo dados na planilha em comunicação com o servidor do Google, um script no formato JSON (*JavaScript Object Notation*) veio acompanhado com as devidas configurações e credenciais. Sendo assim, o Google Sheets tem o comportamento de *webscrapping*, ou seja, solicita dados de uma página Web e extrai todas as informações requeridas.

Através do código abaixo, é feito a requisição de acesso à planilha do Google:

```
scope = ["https://spreadsheets.google.com/feeds",  
        'https://www.googleapis.com/auth/spreadsheets',  
        "https://www.googleapis.com/auth/drive.file",  
        "https://www.googleapis.com/auth/drive"]  
creds = ServiceAccountCredentials.from_json_keyfile_name  
( 'credencial.json', scope )  
client = gspread.authorize(creds)  
sheet = client.open_by_key('chave_do_link').worksheet('BWA Natal')  
dados = sheet.get_all_values()
```

Nas regiões onde estão escritas as palavras “credencial” e “chave_do_link” são espaços destinados para colocar a credencial com extensão .JSON e a chave do link da página respectivamente. A captura dos dados da tabela foi cumprida usando a função

`get_all_values()`. Ela basicamente fez uma espécie de *download* de toda a planilha de uma determinada aba, que no caso da imagem foi da “BWA Natal”, e armazena as informações capturadas na variável *dados* no formato de uma lista de listas.

3.1.1 Organização da planilha

</

Figura 4 – *Layout* da planilha construída

Fonte – Elaborado pelo autor

O *layout* da planilha (Figura 4) foi estruturado na seguinte configuração: as cinco primeiras linhas destinadas ao cabeçalho com os dados privados da empresa, incluindo o mês e ano da competência em exercício; e as linhas posteriores são utilizadas para inserir os dados de cada empresa, informando seu código de cadastro no escritório, Razão social, CNPJ (Cadastro Nacional da Pessoa Jurídica), FAP (Fator Acidentário de Prevenção), Certificado digital e Observações. Estes campos citados são os funcionários que manuseiam com o preenchimento das informações.

Existem mais duas colunas na planilha, porém estas com permissão de escrita exclusiva aos robôs: “Status SEFIP” e “Status Conectividade”. Elas são indicadores para esclarecer aos funcionários qual a situação de cada empresa naquele instante de tempo.

Diante desse raciocínio, foram criadas as seguintes *flags* sinalizadoras, exibidas na Tabela 1:

FLAGS	SIGNIFICADOS
FINALIZADO	A execução do robô da empresa em análise foi concluída com sucesso.
FINALIZADO (SEM GRF)	A execução do robô da empresa em análise foi concluída sem gerar o arquivo GRF por ser uma firma pró-labore (<i>flag</i> exclusiva da coluna “Status Conectividade”).
ERRO DE USUÁRIO	Algum dado pendente ou incompatível na geração do arquivo “Sefip.re” (inconsistência causada por funcionários). Uma foto é capturada do relatório de erro e guardada na pasta “Erros de Usuário” da empresa em análise.
EM ANDAMENTO	A execução do robô da empresa em análise foi inicializada.
NÃO INICIADO	A execução do robô da empresa em análise não começou.
CAMPO FAP VAZIO	O campo FAP da planilha encontra-se vazio.
ARQUIVO NÃO ENCONTRADO	O arquivo “Sefip.re” não existe dentro da pasta da empresa em análise.

Tabela 1 – Flags das colunas “Status SEFIP” e “Status Conectividade” com seus significados

A coluna “Observações” tem um adendo especial em comparação das outras citadas, porque com essa informação o bot saberia qual empresa estaria pronta para ser executada, reexecutada ou se estaria classificada como “sem movimento”. As que apareciam com esta última identificação, eram empresas que não existiam recolhimento ao FGTS e informações à Previdência Social, e deviam transmitir pelo Conectividade Social um arquivo chamado “SEFIPCR.SFP” com indicativo de ausência de fato gerador. Este processo era feito manualmente pelos funcionários do escritório e o robô identificava as colunas “Status SEFIP” e “Status Conectividade” com o texto “SEM MOVIMENTO” e avançava para a próxima linha. Na Tabela 2 está melhor definido o que faz cada *flag*.

FLAGS	AÇÃO
EXECUTAR	Informa ao robô SEFIP que a empresa em análise está preparada e a execução é inicializada, atualizando as colunas “Status SEFIP” e “Status Conectividade” para EM ANDAMENTO e NÃO INICIADO, respectivamente.
REEXECUTAR	Informa ao robô SEFIP que a empresa em análise (independente do estado que esteja) terá seu processo reexecutado, apagando todos os arquivos gerados anteriormente. As colunas “Status SEFIP” e “Status Conectividade” são atualizadas para EM ANDAMENTO e NÃO INICIADO, respectivamente.
SEM MOVIMENTO	Informa ao robô SEFIP que a empresa em análise não existe recolhimento ao FGTS e informações à Previdência Social. Dessa maneira, avança para a próxima linha. As colunas “Status SEFIP” e “Status Conectividade” são ambas atualizadas para SEM MOVIMENTO.
“Campo vazio”	Informa ao robô SEFIP que a empresa em análise não está preparada para nenhuma situação. Dessa maneira, avança para a próxima linha.

Tabela 2 – Flags da coluna “Observações” e suas ações

3.2 Robô SEFIP

Nesta subseção, será apresentada, de maneira pontual, os importantes passos que foram idealizados para a composição do robô SEFIP.

3.2.1 Existência de pendências

Figura 5 – Aplicação SEFIP

Fonte – Elaborado pelo autor

Ao executar o script do robô SEFIP, a primeira ação do código é abrir a aplicação do SEFIP (Figura 5) pelo atalho “Executar” do Windows. A partir desse ponto, é praticado a verificação se os dados em cada linha foram preenchidos na tabela, para que, a empresa que esteja em análise, tenha seu funcionamento dentro dos conformes. Caso não esteja de acordo, será realizada análise nos campos.

Primeiramente, o robô observa se o campo “FAP” contém dado, como mostra abaixo.

```
# Se o campo FAP não estiver preenchido, passa para próxima empresa
if dadossefip[3] == '':
    print('Empresa sem FAP! Próxima...')
    time.sleep(2)
    sheet.update_acell(colSEFIP, 'CAMPO FAP VAZIO')
    time.sleep(2)
    sheet.update_acell(colConectividade, 'CAMPO FAP VAZIO')
    time.sleep(2)
    sheet.update_acell(colObservacoes, '')
    time.sleep(2)
    continue
```

Para um melhor entendimento, “FAP” é um fator multiplicador que varia de 0,5 a 2,0 que pode diminuir ou dobrar o valor que as pessoas jurídicas pagam para aposentadoria decorrente de acidentes no trabalho. Então se este multiplicador por ventura não existir, o robô identificará “CAMPO FAP VAZIO” e avança para próxima linha.

Na sequência, ele averigua a coluna “Observações”. Se estiver vazia ou classificada como “SEM MOVIMENTO”, passará para próxima linha, como são exibidos a seguir. Mas se a identificação for “EXECUTAR” ou “REEXECUTAR”, o robô inicia/reinicia o serviço na empresa em questão colocando “EM ANDAMENTO” na coluna “Status SEFIP” e “NÃO INICIADO” na “Status Conectividade”.

```
# Se o campo "Observações" estiver vazio, passará para próxima empresa
```

```
if dadossefip[7] == '':  
    print('Próxima empresa...')  
    time.sleep(2)  
    continue
```

```
# Se a empresa for sem movimento, passará para próxima
```

```
if dadossefip[7] == 'SEM MOVIMENTO':  
    print('Empresa sem movimento! Próxima...')  
    time.sleep(2)  
    sheet.update_acell(colSEFIP, 'SEM MOVIMENTO')  
    time.sleep(2)  
    sheet.update_acell(colConectividade, 'SEM MOVIMENTO')  
    time.sleep(2)  
    sheet.update_acell(colObservacoes, '')  
    time.sleep(2)  
    continue
```

Para realizar a ação de escrita na planilha, a função *update_acell()* foi utilizada, passando como parâmetros a posição da célula e o texto desejado. E o comando *time.sleep()* é necessário, porque a conexão com o Google Sheets requer um pouco de tempo para efetuar as atualizações na planilha.

3.2.2 Existência do arquivo 'Sefip.re'

O “Sefip.re” é o arquivo que o sistema de folha de pagamento gera mensalmente para cada empresa que possua servidores de vínculo CLT, sem vínculo empregatício ou serviço temporário, e que colaborem mensalmente à Previdência Social (SEAD, 2018).

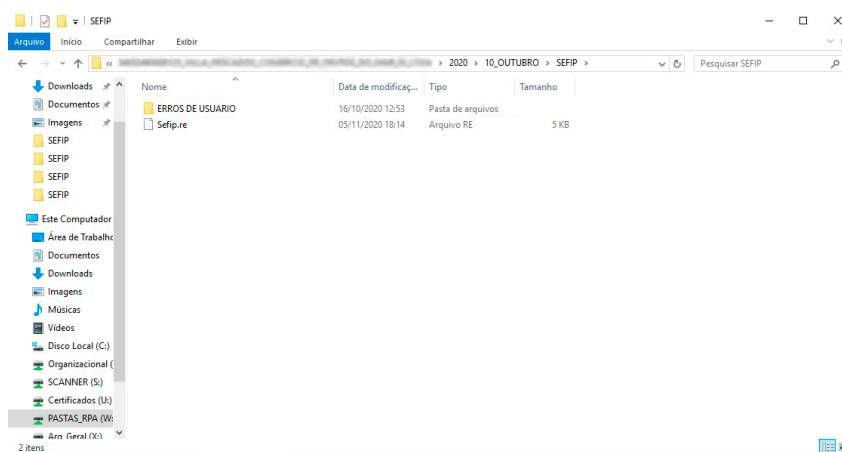


Figura 6 – Arquivo “Sefip.re” inserido na pasta de uma empresa específica

Fonte – Elaborado pelo autor

No projeto, esse arquivo já era gerado pelos funcionários do escritório de contabilidade e inserido na pasta de cada empresa envolvida, como é exibida na Figura 6. A tarefa do robô basicamente foi fazer uma varredura pelo caminho de pastas e identificar se havia esse arquivo específico no diretório de cada órgão. Caso fosse negado, o robô identificava nas colunas como “ARQUIVO NÃO ENCONTRADO”.

A árvore de pastas seguiu sempre um padrão, alterando apenas a pasta do nome da empresa, do ano e da competência. As duas últimas informações eram acessíveis de extrair porque já eram passadas diretamente pela planilha. Mas a pasta do nome da empresa envolvia uma estratégia.

As empresas que não tem filial, ou seja, apenas matriz, normalmente têm nomes únicos. Mas quando elas apresentam uma ou mais filiais, os nomes se repetem. Entretanto, o único identificador que os diferenciam é o CNPJ. Partindo dessa premissa, as pastas das empresas foram organizadas iniciando o nome pelo CNPJ sem caracteres especiais. Portanto, o CNPJ da linha em execução era extraído da planilha e efetuado a manipulação de string com o método *replace()*, deixando apenas números. Após isso, usando as funções *os.listdir()* e *startswith()*, o laço verificava a existência do diretório pelo início da string através do CNPJ analisado. Caso positivo, o robô guardava o nome da pasta na variável *nomePastaEmpresa* para ser utilizado no endereço do caminho. Segue o trecho do código abaixo que realiza o tratamento do dado CNPJ para a busca da pasta de cada empresa.

```
cnpj = dadossefip[2]
simbolos = "./-"

for i in range(0,len(simbolos)):
    cnpj = cnpj.replace(simbolos[i],"")

time.sleep(1)

caminhoAuxiliar = 'W:/'
for name in os.listdir(caminhoAuxiliar):
    if name.startswith(cnpj):
        nomePastaEmpresa = name
```

3.2.3 Erros de usuário

De repente um funcionário do escritório fixa na planilha um valor incompatível para a variável “FAP” de uma empresa específica ou alguma informação equivocada foi lançada por ele sobre o arquivo “Sefip.re” durante a sua geração na folha de pagamento. Isso provoca um erro na execução da firma, chamado “Erro de Inconsistência”.

Assim sendo, o primeiro passo foi analisar o título da janela que apresenta o indicativo de erro e se apresentava outras com mesmo nome ou não. Como o título “Impressão de Relatório” era único (Figura 7), o robô fazia o mapeamento dele usando o pacote *PyGetWindow* e, assim que o surgisse, um novo comando abriria o “Relatório de Inconsistência” (Figura 8) e tiraria uma foto da tela com a biblioteca *Pyscreenshot*.

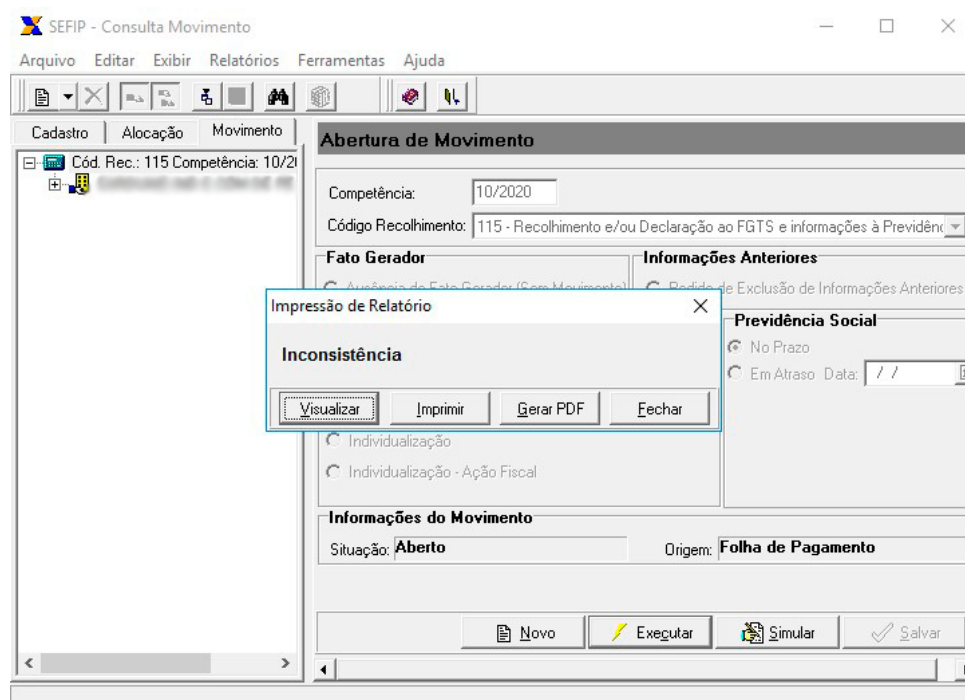


Figura 7 – Janela “Impressão de Relatório”

Fonte – Elaborado pelo autor

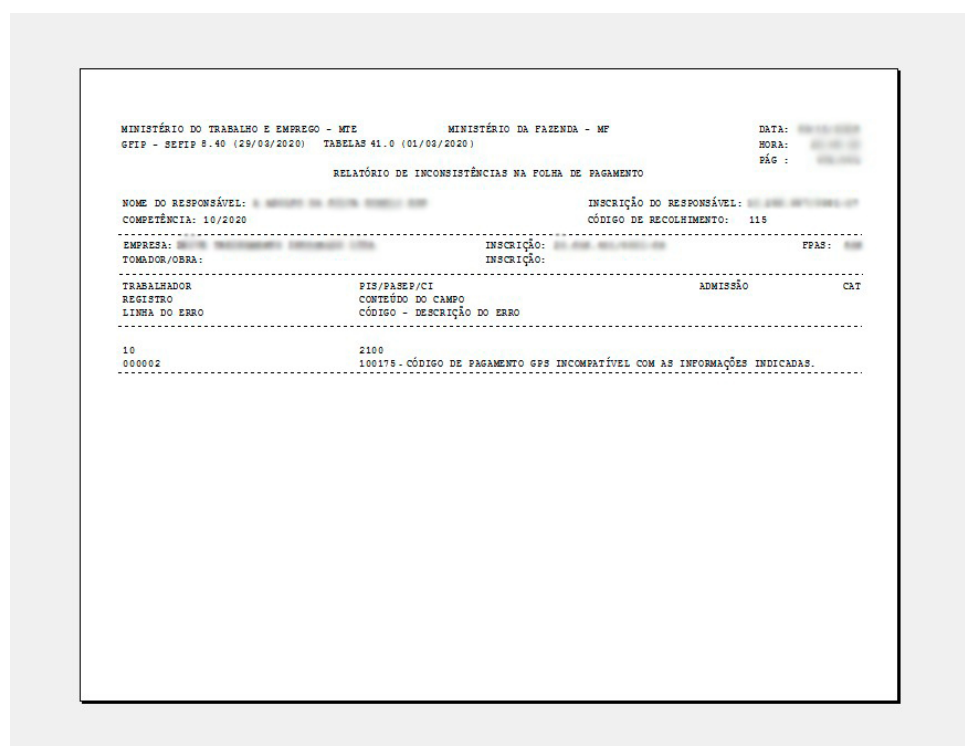


Figura 8 – Relatório de inconsistência

Fonte – Elaborado pelo autor

O *print* do “Relatório de Inconsistência” era enviada diretamente para a pasta

“Erros de Usuário” da empresa para algum funcionário fazer a correção do erro cometido. Após os ajustes devidos, o trabalhador do escritório colocaria a flag “REEXECUTAR” na coluna *Observações* para uma nova tentativa.

3.2.4 Geração dos arquivos

Nesta etapa, não apresentando os erros de usuário da subseção anterior, o robô gerava de imediato o arquivo de extensão .SFP com o auxílio da biblioteca *PyautoGUI* para as ações do mouse e do teclado, e também com nome ajustado para “*CNPJ+ano+mes.SFP*”. A seguir, ele verificava quais “Relatórios de Movimento” estiveram disponíveis para empresa em ação a fim de efetuar o *download* dos mesmos.

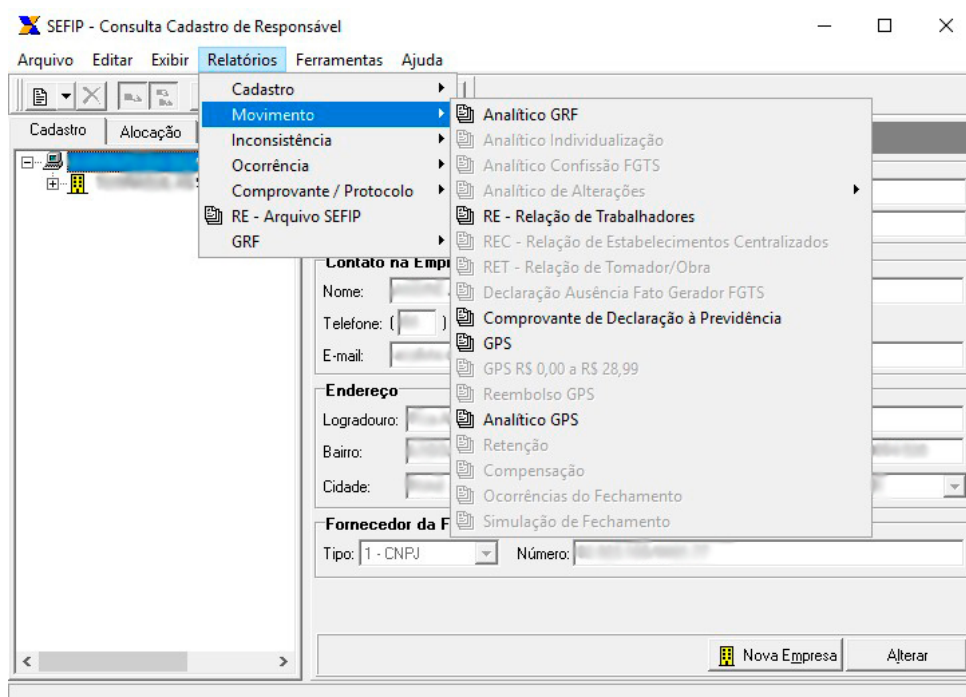


Figura 9 – “Relatórios de Movimento” para *download*

Fonte – Elaborado pelo autor

Na Figura 9 exibe o momento em que aparece a lista desses arquivos, porém as disponíveis são as que estão em negrito. Para concretizar esse passo, foi utilizado o comando *try/except* e o robô verificava cada opção realizando o comando da tecla ENTER. Se uma janela correspondente ao campo analisado abrisse na sequência, confirmaria a existência do arquivo. Porém, caso contrário, o robô ignoraria e faria a verificação da próxima opção. Ou seja, os tipos de relatórios de movimento gerados em tal empresa não necessariamente são iguais nas outras.

Devido ao fato do aplicativo ser antiquado para os tempos atuais, o espaço para informar o diretório de destino dos arquivos criados precisa ser acessado pasta por pasta. E

mais: se utilizar as funções do teclado, só consegue identificar o primeiro caractere digitado. Portanto se, por exemplo, tinham dez diretórios nomeados com CNPJ's iniciados pelo número 0, só seria acessado o primeiro destes.

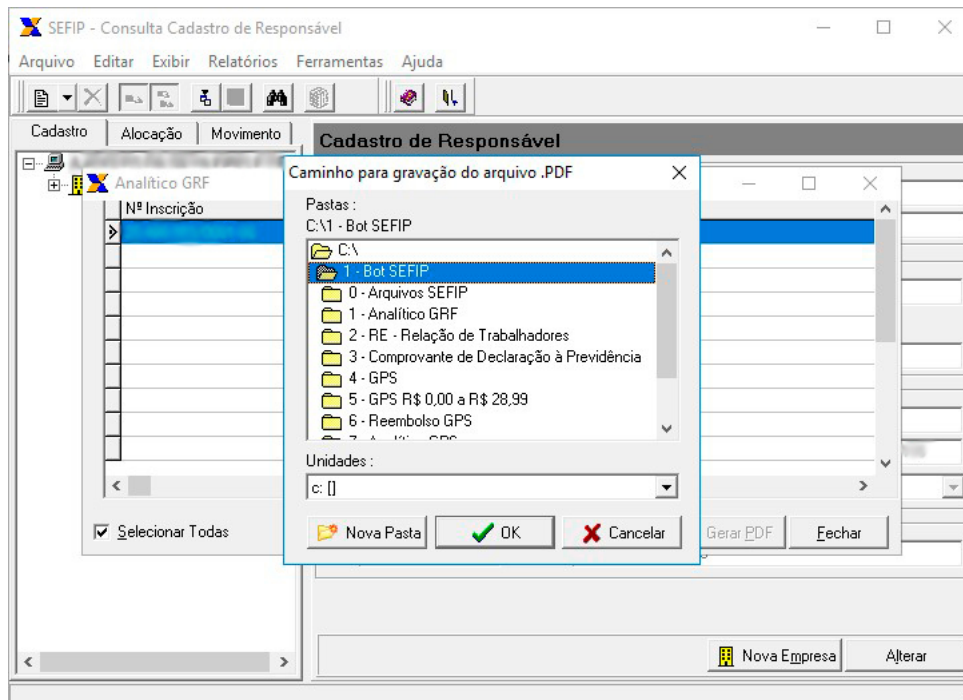


Figura 10 – Pastas auxiliares para receber temporariamente os arquivos gerados

Fonte – Elaborado pelo autor

Partindo desta premissa, foi preciso realizar uma adaptação estratégica como exibe a Figura 10, criando pastas auxiliares enumeradas de 0 a 9 para facilitar o uso do teclado. Cada arquivo gerado era enviado para sua respectiva pasta temporária e, por fim, dentro do código ele era renomeado e encaminhado para a pasta da empresa em ação. A estrutura do nome ficou sendo “*Tipo_de_Relatorio mes+ano.PDF*”, como é apresentado na Figura 11.

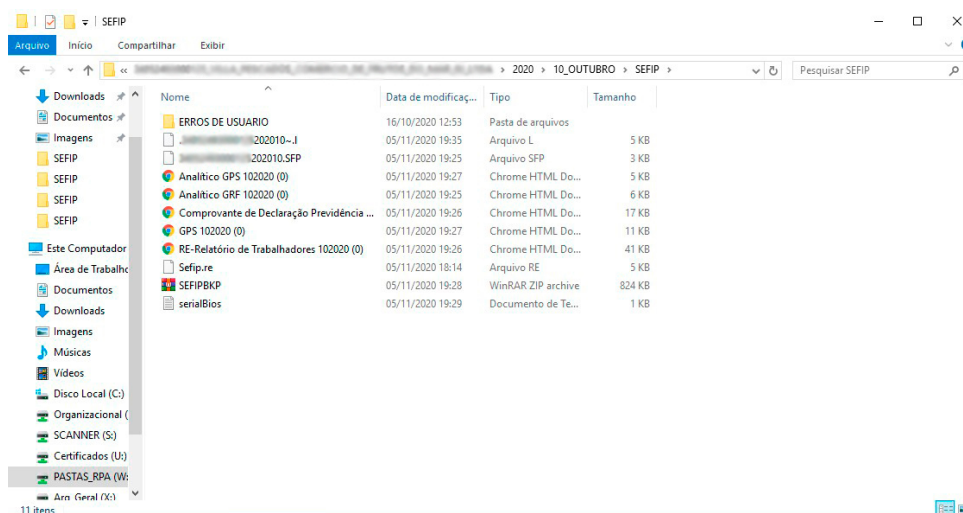


Figura 11 – Pasta de uma empresa com os arquivos gerados após a execução do robô SEFIP

Fonte – Elaborado pelo autor

3.2.5 Realizar backup

Para finalizar o robô SEFIP, é feito o procedimento de backup, guardando na pasta de cada empresa executada todo o procedimento concretizado até esse momento, em um arquivo zipado chamado “SEFIPBKP.zip”.

3.3 Robô Conectividade Social

Agora nesta subseção, será apresentada as importantes etapas que foram construídas para a formação do robô Conectividade.

3.3.1 Existência de pendências

Assim como no bot do SEFIP, o robô Conectividade só trabalha após uma verificação se as condições o permitem inicializar uma determinada firma. E a sua primeira abordagem é observar se a coluna “Status Conectividade” na linha em execução está vazia ou não. Se sim, o robô passa para a próxima linha. Caso contrário, na sequência ele identifica se na coluna “Status SEFIP” está “FINALIZADO”. Sendo afirmativo, o bot está apto para realizar a etapa da Conectividade Social. Ou seja, devido a dependência do arquivo gerado no SEFIP para dar sequência no portal da Conectividade, é necessário que a empresa em análise tenha o “Status SEFIP” como foi citado acima.

3.3.2 Certificado digital

Segundo (VIVIAN, 2018), o certificado digital é uma identidade eletrônica para pessoas ou empresas. Ele equivale à uma carteira de identidade do mundo virtual. É muito útil para agilizar assinatura de documentos, pois é a partir dele que é possível obter a assinatura digital. Ela imprime autenticidade em transações online e outras funcionalidades.

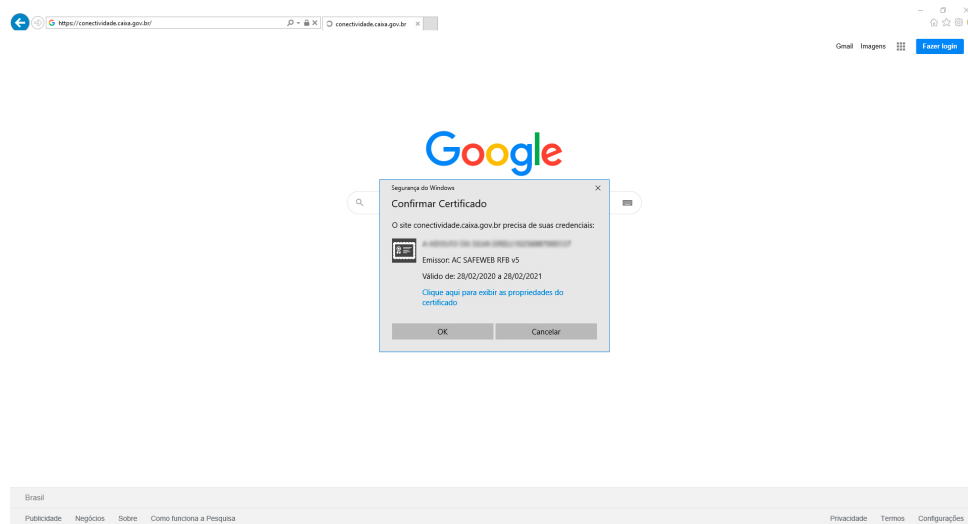


Figura 12 – Janela do certificado digital para o acesso ao portal Conectividade Social

Fonte – Elaborado pelo autor

Para entrar no portal da Conectividade Social, é necessário o uso do certificado digital (Figura 12). Mas para o robô obter êxito em passar por essa etapa, teve que ser feito uma *thread*.

A explicação começa pelo uso do pacote *Selenium*. A função *driver.get()* navega para uma página web dada pela URL. O WebDriver aguardará até que a página esteja totalmente carregada antes de retornar o controle ao script. Porém, com a utilização do certificado, a página segue o evento “onload” num *loop* infinito, aguardando a ação da identidade eletrônica. Então para isso foi necessário a implantação de uma *thread* para agir em paralelo nessa janela de autenticação.

O pacote que providencia as funcionalidades de *Thread* é chamado de *threading* e foi importado no começo do código.

No código a seguir tem a classe *Th*, que inicializou a classe *Thread* e implementou o método *run()*. O CNPJ foi omitido pelos caracteres “XXXXX” por privacidade de informação.

```
class Th(Thread):
    def __init__(self, num):
```

```
Thread.__init__(self)
self.num = num

def run(self):
    time.sleep(10)

    dado = dadossefip[4]
    if 'XXXXXXXXXXXXXXX' in dado:
        pyautogui.press(['tab','tab','enter'], interval=0.5)
        time.sleep(3)
```

Quando chamada a função *start()*, que é definida na classe *Thread*, e herdada e iniciada pela classe *Th*, o conteúdo do método *run()* executa em paralelo ao código principal. Abaixo segue o trecho de inicialização da *thread* juntamente com o driver do navegador.

```
a = Th(1)
a.start()
driver = webdriver.Ie("C:/IEDriverServer.exe")
driver.get('https://conectividade.caixa.gov.br/')
time.sleep(10)
```

Vale salientar também que na função *run()* foi utilizado um método da biblioteca *PyautoGUI*, o *pyautogui.press()*, para solucionar a janela do certificado usando as teclas do teclado. Mas antes de entrar nestes comandos, o bot precisaria saber qual certificado seria trabalhado em tal empresa. E para isso, na linha em análise, é verificado o texto da célula na coluna “Certificados”, através da variável *dadossefip[4]*. E por fim, a página inicial do Conectividade Social concluiria o carregamento, como mostra na Figura 13.



Figura 13 – Página principal do site Conectividade Social

Fonte – Elaborado pelo autor

Sem ter conhecimento do quanto tempo poderia alongar o carregamento da página, o próximo clique foi planejado por reconhecimento de imagem, usando a biblioteca *Pyautogui*, da seguinte maneira: foi efetuado a captura da área onde desejava que ocorresse o clique do *mouse*. E pela função *pyautogui.click()*, foi-lhe passado o caminho da foto retirada. Dentro de um *loop* “infinito”, essa foto foi analisada por toda a tela. Concretizando sua existência, o clique é efetuado e sai da repetição. Caso contrário, espera por mais dois segundos para uma nova análise, como apresenta no código abaixo.

```
while True:
    try:
        pyautogui.click(r'C:\\1 - Bot SEFIP\\caixapostal.png')
        pyautogui.moveTo(57, 173,duration=0.2)
        pyautogui.click()
        break
    except:
        time.sleep(2)
```

É importante destacar que a resolução de tela é de extrema importância nesse caso, porque está diretamente ligada ao clique do *mouse* e a qualidade da imagem. Primeiramente, o *mouse* é programado pelo posicionamento dos pixels nos eixos “x” e “y”, e o monitor de resolução mais alta tem uma extensão maior de pixels em comparação a telas menores, impossibilitando assim um mapeamento compatível. E por último, se a foto da área

capturada apresentar alguma imperfeição inesperada em comparação a imagem que estiver sendo exibida, a ação do clique também será ignorada.

3.3.3 Geração do arquivo da Conectividade

Na janela onde é gerado os arquivos .XML e .PDF da Conectividade, pode aparecer novamente a tela de certificado digital. Entretanto esta janela surge sem nenhuma interferência de comando do pacote *Selenium*. Partindo dessa premissa, a ação do robô foi suficiente comandada pelas bibliotecas *PyGetWindow* e *PyautoGUI*.

Ao descarregar os arquivos, estes são guardados na pasta da empresa em execução com os nomes alterados para “*CNPJ+ano+mes.XML*” e “*CNPJ+ano+mes.PDF*” (Figura 14). Isso auxilia os funcionários caso queiram efetuar auditoria sobre os dados algum dia.

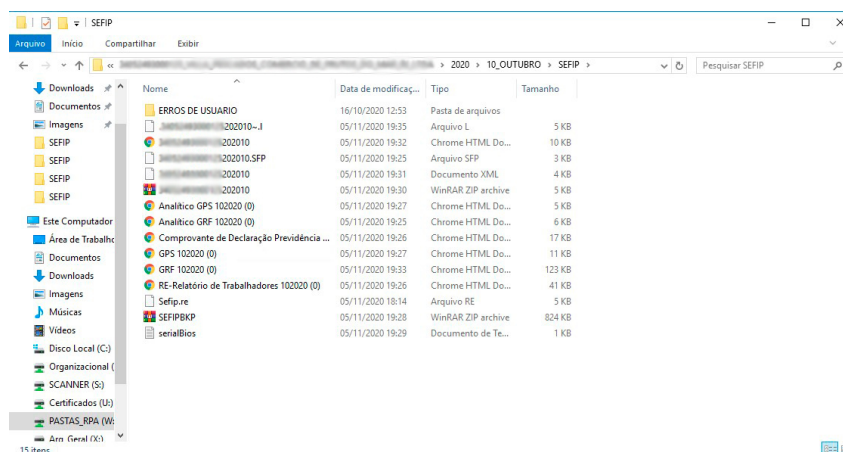


Figura 14 – Pasta de uma empresa com os arquivos gerados após a execução do robô Conectividade

Fonte – Elaborado pelo autor

3.3.4 Importação do arquivo .XML no SEFIP

O arquivo de extensão .XML, descarregado do portal da Conectividade Social, foi importado na aplicação SEFIP para gerar o arquivo de Guia de Recolhimento do FGTS, o GRF (Figura 15).

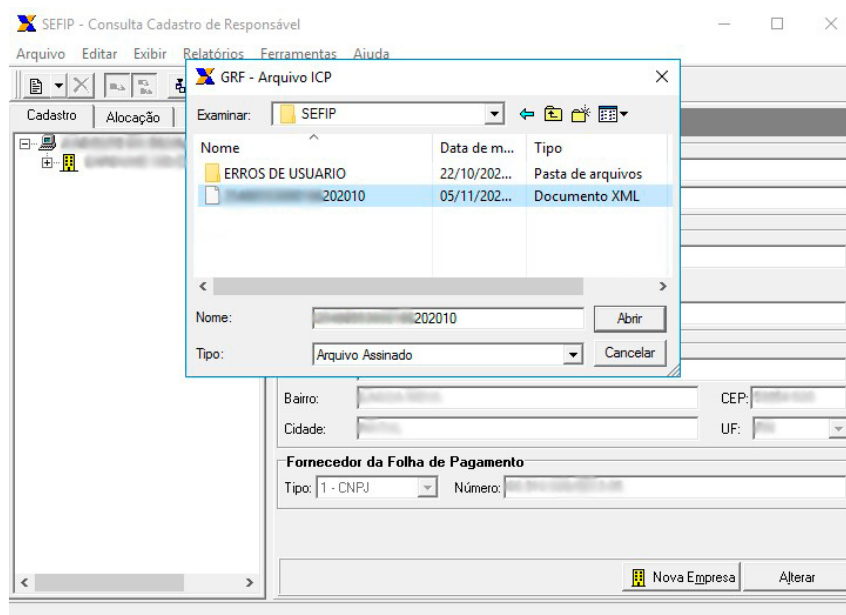


Figura 15 – Aplicação SEFIP - Importação do arquivo de extensão .XML

Fonte – Elaborado pelo autor

No próximo passo, se por ventura o pacote *PyGetWindow* tiver encontrado uma janela com título “Erro” (Figura 16), a empresa envolvida é classificada como “pró-labore”. Segundo (ANDRADE, 2020), pró-labore refere-se a firma que tenha a remuneração de sócios por atividades administrativas, sendo opcional e diferente da distribuição de lucros ou dividendos. Sobre ele, não existem regras obrigatórias em relação ao 13^a salário, Fundo de Garantia do Tempo de Serviço (FGTS), férias, entre outros. Por isso que apareceu essa janela, já que a configuração da empresa não permitiu que tivesse o GRF, ou melhor, o boleto do FGTS. Neste caso, na coluna “Status Conectividade” da planilha foi atualizado para “FINALIZADO (SEM GRF)”.

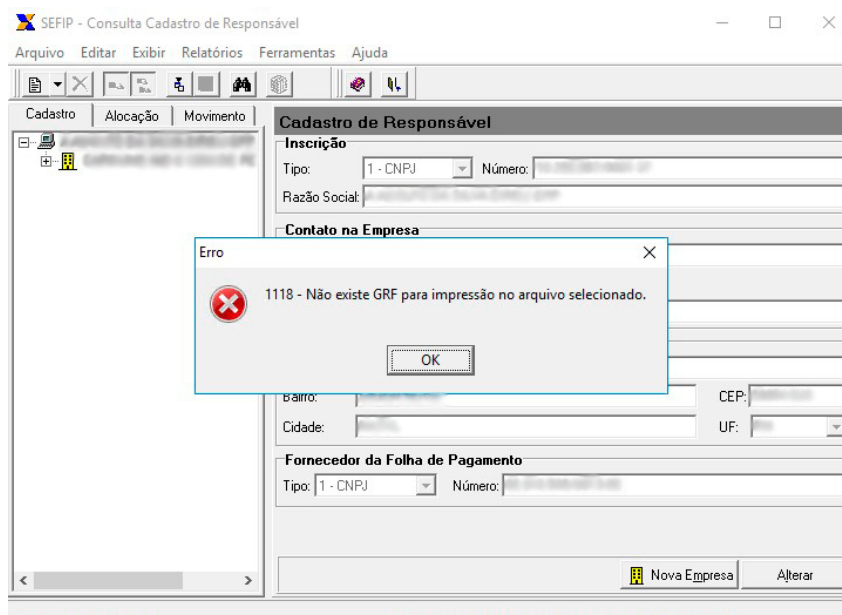


Figura 16 – Aplicação SEFIP - Janela “Erro” indicando empresas pró-labore (sem arquivo GRF)

Fonte – Elaborado pelo autor

Todavia, não surgindo esta janela exibida anteriormente, significaria que a empresa em análise não era pró-labore e o arquivo GRF por fim seria gerado (Figura 17) e armazenado na pasta da devida empresa. A planilha sofreria atualização na “Status Conectividade” para “FINALIZADO” e os passos de execução do robô Conectividade chegaria ao fim.

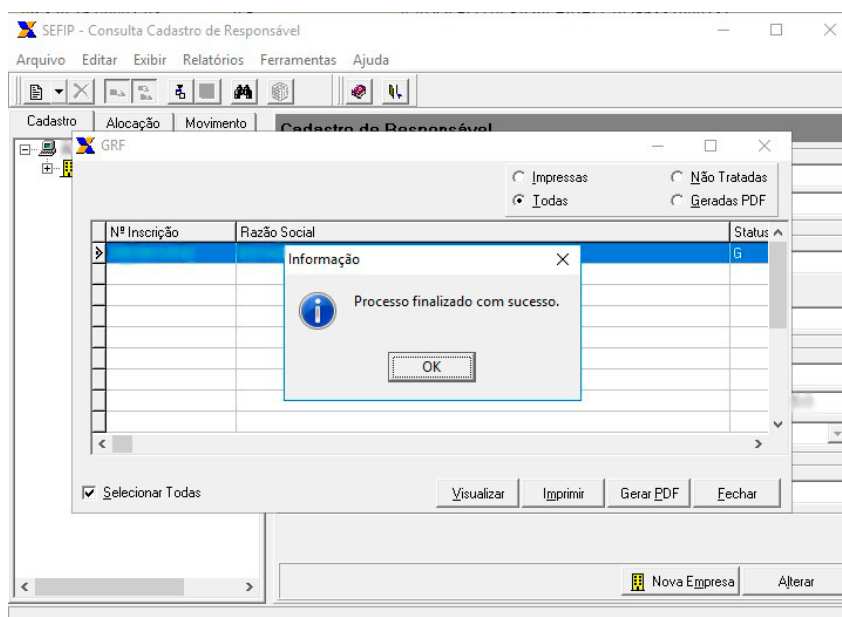


Figura 17 – Aplicação SEFIP - Janela “Informação” indicando a geração do arquivo GRF

Fonte – Elaborado pelo autor

4 RESULTADOS OBTIDOS

Este capítulo é destinado para exposição e discussão dos resultados obtidos durante o projeto.

A aplicação dos robôs foram sendo efetuados de maneira gradativa na BWA Global - Natal, iniciando a primeira execução completa na competência de Outubro em fase de testes. Após a validação por parte do Setor Trabalhista do escritório, os bots assumiram o mesmo serviço para a competência seguinte nas outras unidades: Recife, Rio de Janeiro - Barra da Tijuca e Rio de Janeiro - Penha.

Diante disso, na competência de Outubro, foi realizada uma análise estatística da apuração dos *status* exibidos por cada unidade da BWA Global, sendo apresentado gráficos com percentil correspondente à amostra de empresas executadas.

Vale recordar que as empresas que concluíram SEFIP e Conectividade com sucesso, não necessariamente apresentavam os mesmos tipos de arquivos gerados. Depende da situação de cada firma em análise.

E por conta desse fato, inclusive também por pequenos retardos no carregamento do portal da Conectividade, o tempo sofria variações e impossibilitava da execução ter uma duração bem definida. Portanto, após várias análises, 10 minutos foi o tempo médio por empresa que o serviço conduziu até ser concluído. Assim sendo, como curiosidade, foi levantado nas seções seguintes uma estimativa de tempo de trabalho dos robôs nas empresas que tiveram *status* de “FINALIZADO”.

Esse serviço realizado somente por funcionários é feito de maneira fracionária, onde cada dia é finalizado uma determinada quantidade de empresas, dentro de um período aproximado de 8 dias. Cada empresa precisa em torno de 12 minutos para ser concluída. Portanto, a seguir será comparado o desempenho e a produtividade ao visualizarmos os resultados conquistados.

4.1 Unidade Natal

STATUS	QUANTIDADE	PERCENTIL
ARQUIVO NÃO ENCONTRADO	12	3,04%
SEM MOVIMENTO	62	15,70%
ERRO DE USUARIO	16	4,05%
EM ANDAMENTO	0	0,00%
FINALIZADO	305	77,22%
CAMPO FAP VAZIO	0	0,00%
TOTAL	395	100,00%

Tabela 3 – Status SEFIP - Unidade Natal

STATUS	QUANTIDADE	PERCENTIL
NÃO INICIADO	16	4,05%
FINALIZADO (SEM GRF)	19	4,81%
FINALIZADO	285	72,15%
ERRO DE USUARIO	1	0,25%
SEM MOVIMENTO	62	15,70%
CAMPO FAP VAZIO	0	0,00%
ARQUIVO NÃO ENCONTRADO	12	3,04%
TOTAL	395	100,00%

Tabela 4 – Status CONECTIVIDADE - Unidade Natal

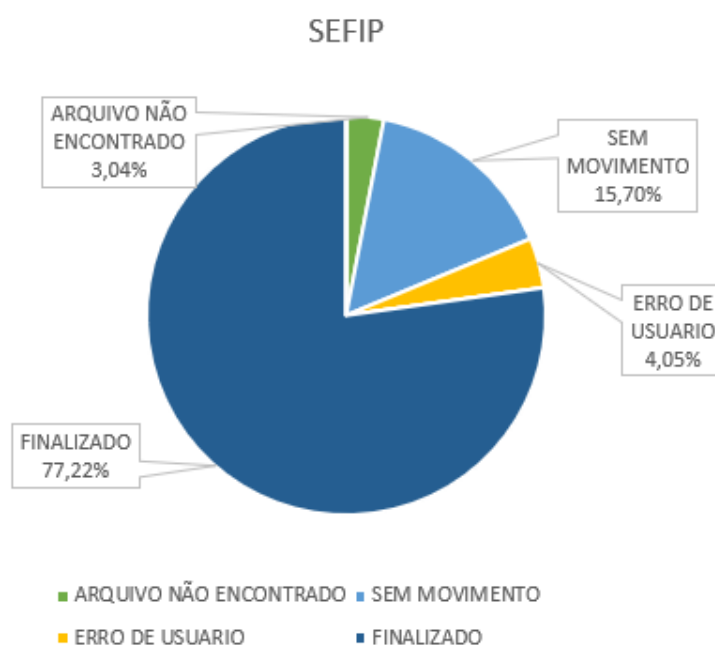


Figura 18 – Gráfico SEFIP - Unidade Natal

Fonte – Elaborado pelo autor

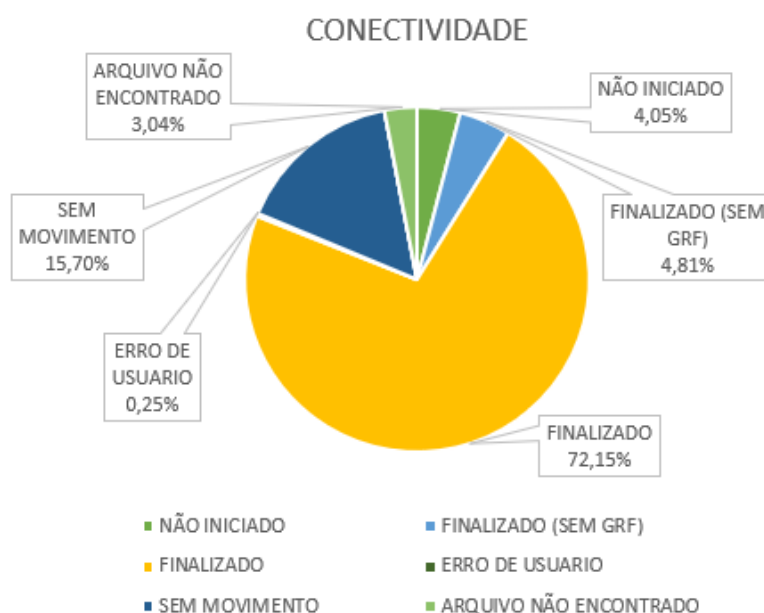


Figura 19 – Gráfico CONECTIVIDADE - Unidade Natal

Fonte – Elaborado pelo autor

Na BWA Global - Natal, de acordo com os dados das tabelas e gráficos acima, 395 empresas fizeram parte da amostra. A execução ocorreu com o utilização de duas máquinas virtuais em paralelo para acelerar o processo.

Analizando simultaneamente as tabelas dos dois *bots*, podemos observar que 305 empresas concluíram o robô SEFIP, mas apenas uma delas apresentou “Erro de Usuário” no *bot* Conectividade. Portanto, finalizaram por completo 304 empresas (com ou sem arquivo GRF), contabilizando 76,96% da amostra.

Considerando o tempo estimado de 10 minutos para cada empresa finalizar os dois processos, e que existiram duas máquinas dividindo o funcionamento ao mesmo tempo, conclui-se que foram necessários aproximadamente 1520 minutos ou 25,3 horas de execução.

4.2 Unidade Recife

STATUS	QUANTIDADE	PERCENTIL
ARQUIVO NÃO ENCONTRADO	1	1,85%
SEM MOVIMENTO	3	5,56%
ERRO DE USUARIO	0	0,00%
EM ANDAMENTO	0	0,00%
FINALIZADO	50	92,59%
CAMPO FAP VAZIO	0	0,00%
TOTAL	54	100,00%

Tabela 5 – Status SEFIP - Unidade Recife

STATUS	QUANTIDADE	PERCENTIL
NÃO INICIADO	0	0,00%
FINALIZADO (SEM GRF)	20	37,04%
FINALIZADO	30	55,56%
ERRO DE USUARIO	0	0,00%
SEM MOVIMENTO	3	5,56%
CAMPO FAP VAZIO	0	0,00%
ARQUIVO NÃO ENCONTRADO	1	1,85%
TOTAL	54	100,00%

Tabela 6 – Status CONECTIVIDADE - Unidade Recife

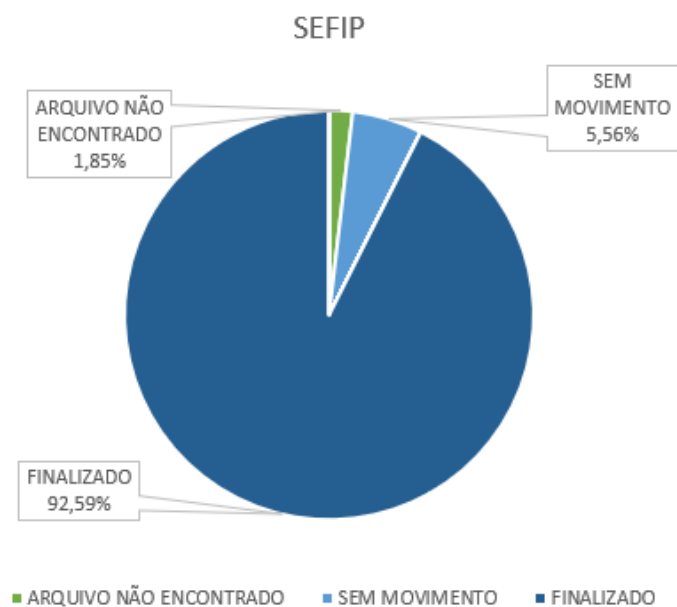


Figura 20 – Gráfico SEFIP - Unidade Recife

Fonte – Elaborado pelo autor

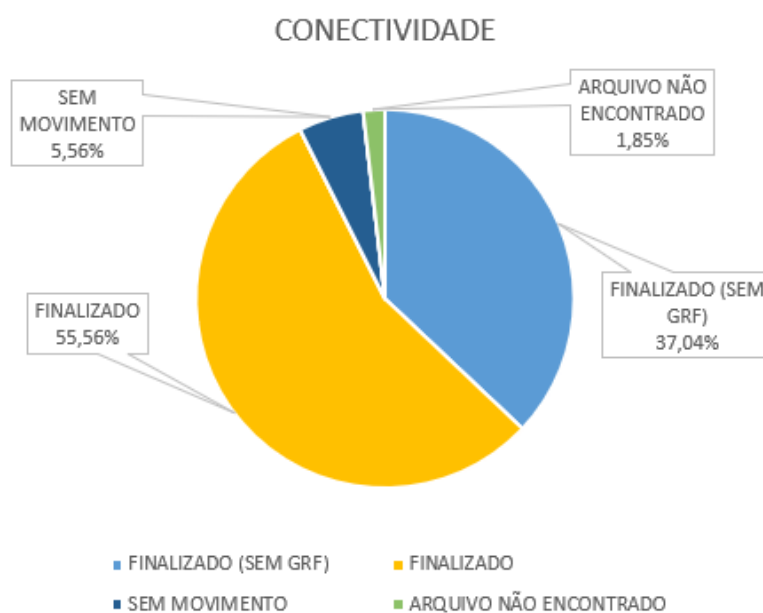


Figura 21 – Gráfico CONECTIVIDADE - Unidade Recife

Fonte – Elaborado pelo autor

Na unidade pernambucana da BWA Global, localizada na capital Recife, apenas 54 empresas foram destinadas para compor a amostra. Devido a essa quantidade, foi utilizada apenas uma máquina virtual para execução do robô.

Observando as estatísticas das tabelas e gráficos dessa unidade, podemos examinar que das 54 empresas, 50 finalizaram todo o processo (com ou sem arquivo GRF), totalizando 92,59% da amostra.

Diante disso, acatando os 10 minutos citados anteriormente, foram necessários cerca de 500 minutos ou 8,3 horas de execução dos robôs.

4.3 Unidade Rio de Janeiro - Barra da Tijuca

STATUS	QUANTIDADE	PERCENTIL
ARQUIVO NÃO ENCONTRADO	1	0,54%
SEM MOVIMENTO	94	50,81%
ERRO DE USUARIO	3	1,62%
EM ANDAMENTO	0	0,00%
FINALIZADO	87	47,03%
CAMPO FAP VAZIO	0	0,00%
TOTAL	185	100,00%

Tabela 7 – Status SEFIP - Unidade Rio de Janeiro (Barra)

STATUS	QUANTIDADE	PERCENTIL
NÃO INICIADO	3	1,62%
FINALIZADO (SEM GRF)	38	20,54%
FINALIZADO	49	26,49%
ERRO DE USUARIO	0	0,00%
SEM MOVIMENTO	94	50,81%
CAMPO FAP VAZIO	0	0,00%
ARQUIVO NÃO ENCONTRADO	1	0,54%
TOTAL	185	100,00%

Tabela 8 – Status CONECTIVIDADE - Unidade Rio de Janeiro (Barra)

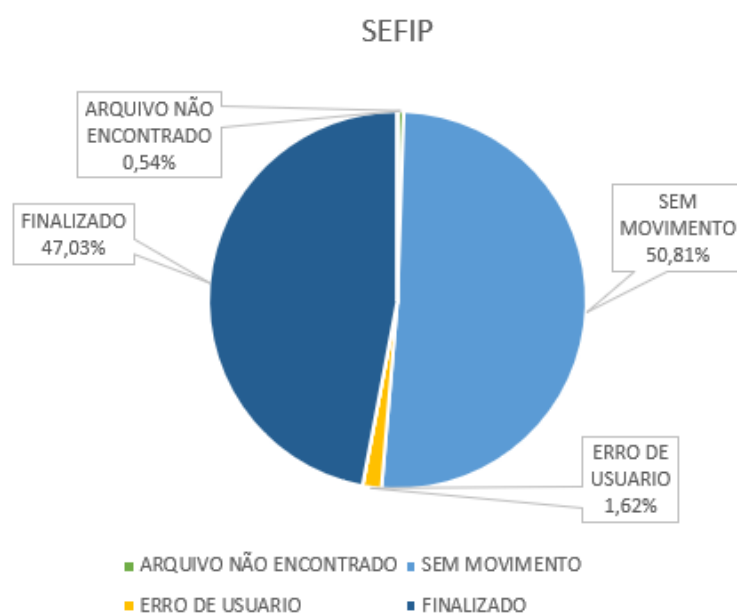


Figura 22 – Gráfico SEFIP - Unidade Rio de Janeiro (Barra)

Fonte – Elaborado pelo autor

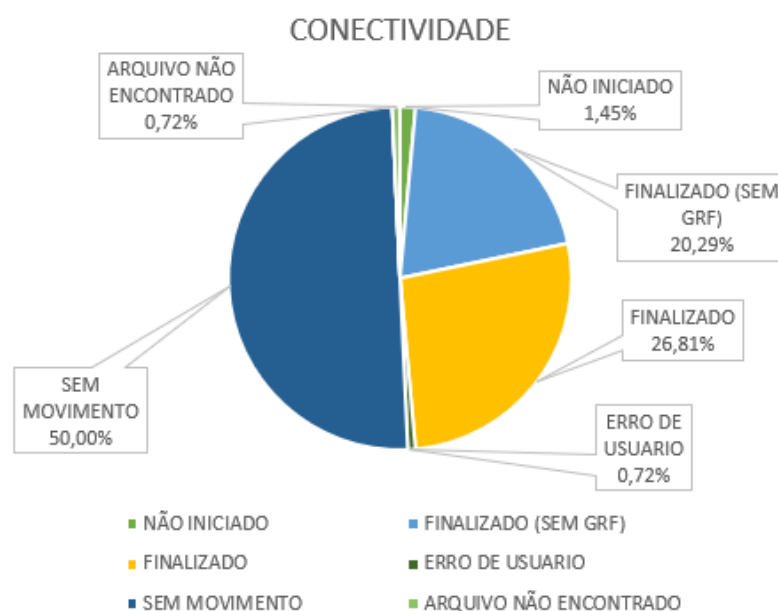


Figura 23 – Gráfico CONECTIVIDADE - Unidade Rio de Janeiro (Barra)

Fonte – Elaborado pelo autor

Na BWA Global - Barra da Tijuca, 185 empresas foram selecionadas para execução dos robôs. A execução ocorreu com o utilização de duas máquinas virtuais em paralelo para acelerar o processo.

Com os dados exibidos nas tabelas e gráficos, 87 empresas finalizaram o serviço completo (com ou sem arquivo GRF), assumindo 47,03% da amostra.

Seguindo a estimativa de 10 minutos para cada empresa e funcionando duas máquinas em paralelo, foram precisos de aproximadamente 435 minutos ou 7,25 horas de execução.

4.4 Unidade Rio de Janeiro - Penha

STATUS	QUANTIDADE	PERCENTIL
ARQUIVO NÃO ENCONTRADO	2	0,72%
SEM MOVIMENTO	138	50,00%
ERRO DE USUARIO	4	1,45%
EM ANDAMENTO	0	0,00%
FINALIZADO	132	47,83%
CAMPO FAP VAZIO	0	0,00%
TOTAL	276	100,00%

Tabela 9 – Status SEFIP - Unidade Rio de Janeiro (Penha)

STATUS	QUANTIDADE	PERCENTIL
NÃO INICIADO	4	1,45%
FINALIZADO (SEM GRF)	56	20,29%
FINALIZADO	74	26,81%
ERRO DE USUARIO	2	0,72%
SEM MOVIMENTO	138	50,00%
CAMPO FAP VAZIO	0	0,00%
ARQUIVO NÃO ENCONTRADO	2	0,72%
TOTAL	276	100,00%

Tabela 10 – Status CONECTIVIDADE - Unidade Rio de Janeiro (Penha)

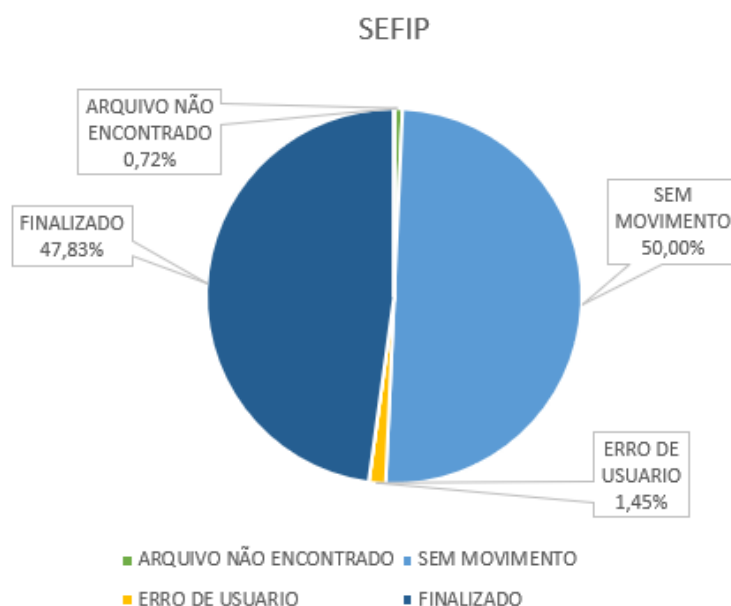


Figura 24 – Gráfico SEFIP - Unidade Rio de Janeiro (Penha)

Fonte – Elaborado pelo autor

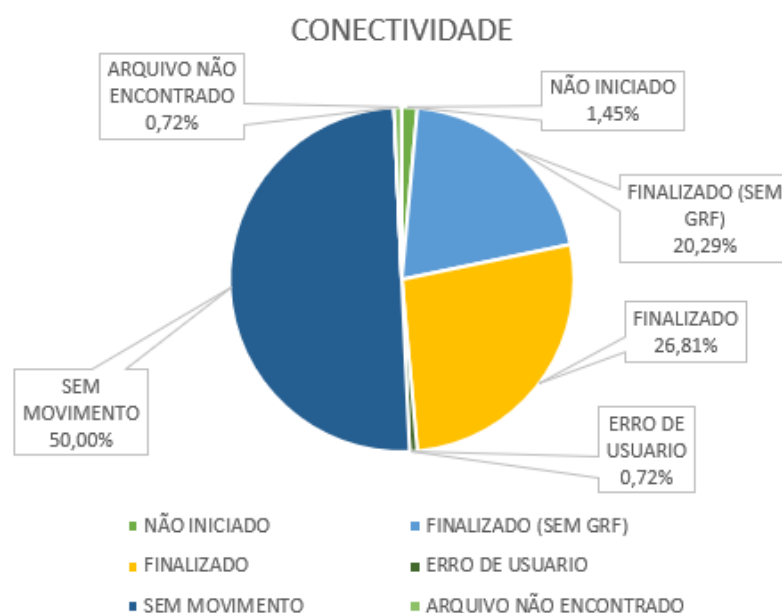


Figura 25 – Gráfico CONECTIVIDADE - Unidade Rio de Janeiro (Penha)

Fonte – Elaborado pelo autor

A outra unidade em território fluminense, a BWA Global - Penha, teve 276 empresas escolhidas para o funcionamento dos robôs. Devido a grande quantidade, também contou com o uso de duas máquinas virtuais em paralelo para reduzir o tempo de trabalho.

De acordo com as informações encontradas nas tabelas e gráficos acima, 132 empresas terminaram o robô SEFIP, porém tiveram duas que ocorreram “Erro de Usuário” no robô Conectividade. Dessa maneira, concluíram por completo 130 empresas (com ou sem arquivo GRF), assim sendo 47,1% da amostra.

Com a utilização de duas máquinas em paralelo, estima-se que aproximadamente 650 minutos ou 10,83 horas de execução foram necessários para terminar o serviço nas empresas desta unidade.

5 CONCLUSÃO

De acordo com o que foi exposto no desenvolvimento desse trabalho, a Automatização Robótica de Processos (RPA) basicamente é uma aplicação tecnológica que automatiza atividades humanas que são monótonas por meio de processamento de dados, sendo utilizados em diversos setores. Durante a pandemia de Covid-19, muitas empresas sofreram com a crise financeira, acarretando em diversas demissões de trabalhadores. Diante desse cenário, evitando algo pior, foi desenvolvido em um escritório contábil, com quatro unidades, um robô de *software* para a execução do serviço “SEFIP-Conectividade”.

Para idealizar o projeto, foi utilizada a linguagem Python e suas bibliotecas como PyautoGUI, PyGetWindow, Selenium e API do Google Sheets. Cada unidade da empresa tinha entre uma ou duas máquinas à disposição para paralelizar e otimizar o tempo da execução do robô.

Através dos números obtidos no capítulo anterior, ficou evidente o tamanho da notabilidade de um *bot* para o processamento de um trabalho repetitivo. Isso causou um impacto benéfico e elogiado dentro da empresa, onde alguns fatores foram destacados como consequência:

- A agilidade na execução das tarefas, podendo criar ainda mais máquinas virtuais e multiplicar o serviço em quantos robôs desejar, podendo reduzir ainda mais o tempo total de atuação;
- Teve um melhor aproveitamento da mão de obra qualificada, já que os funcionários não se preocupavam mais com tarefas repetitivas, focam apenas em atividades mais estratégicas/trabalhosas;
- A redução de falha humana, devido ao fato do robô ter alto índice de precisão.

O maior grau de dificuldade no desenvolvimento do projeto foi de analisar a estrutura da aplicação SEFIP e do portal Conectividade Social e ver qual a melhor estratégia a ser traçada para a construção do robô. O SEFIP é um programa antigo, com várias janelas, e tem a maioria de seus botões de difícil acesso para serem mapeados individualmente, tornando impossível o uso por completo do pacote *Pywinauto*. Diante dessa situação, a biblioteca que melhor se adequou, que pôde acessar todas as opções determinadas, foi a *PyautoGUI* com as funções do teclado, avançando tecla por tecla até chegar na preferência desejada. E em relação ao Conectividade Social é um portal que só permite acesso pelo navegador Internet Explorer e que necessita fazer algumas configurações de segurança no Java para que possa executar.

Conforme já citado, as bibliotecas utilizadas no Python também foram um desafio durante o estudo no início do projeto, para selecionar os pacotes com as melhores funções de acordo com a estratégia planejada. Ao mesmo tempo, foi prazeroso visualizar o passo a passo até atingir o resultado alcançado. Todo o conhecimento absorvido durante o curso de Engenharia de Computação foi imprescindível para a evolução de toda essa caminhada, com destaque aos princípios e conceitos encontrados em componentes curriculares como Inteligência Artificial, Sistemas de Tempo Real e Projeto e Desenvolvimento de Software.

O RPA não é adequado para processos que não são altamente orientados por regras. Isso pode levar a erros de processamento quando houver alterações frequentes nesses procedimentos-padrão. Essa melhoria é algo que pode ser desenvolvido posteriormente, como uma proposta de trabalho futuro, em tarefas que exijam ainda mais do robô, evoluindo em Inteligência Artificial / Aprendizado de Máquina (Machine Learning).

REFERÊNCIAS

- ANDRADE, M. *O que é pró-labore e por que ele é diferente de salário*. 2020. Disponível em: <<https://blog.contaazul.com/o-que-e-pro-labore/>>. Acesso em: 02.11.2020.
- CAIXA. *O que é SEFIP*. 2013. Disponível em: <<https://www.caixa.gov.br/empresa/fgts-empresas/SEFIP-GRF/Paginas/default.aspx>>. Acesso em: 21.10.2020.
- 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/>>. Acesso em: 21.10.2020.
- CLIENT, G. A. *Introduction to the Google Sheets API*. 2019. Disponível em: <<https://developers.google.com/sheets/api/guides/concepts>>. Acesso em: 21.10.2020.
- GARTNER. *Gartner Says Worldwide Robotic Process Automation Software Revenue to Reach Nearly \$2 Billion in 2021*. 2020. Disponível em: <<https://www.gartner.com/en/newsroom/press-releases/2020-09-21-gartner-says-worldwide-robotic-process-automation-software-revenue-to-reach-nearly-2-billion-in-2021>>. Acesso em: 20.10.2020.
- GRUPO QUALITAT. *Automação Robótica de Processos: Uma ferramenta necessária para a transformação digital*. São Paulo, 2020. 2 p.
- PYAUTOGUI. *Welcome to PyAutoGUI's documentation!* 2019. Disponível em: <<https://pyautogui.readthedocs.io/en/latest/index.html>>. Acesso em: 21.10.2020.
- PYGETWINDOW. *PyGetWindow - Project description*. 2020. Disponível em: <<https://pypi.org/project/PyGetWindow/>>. Acesso em: 21.10.2020.
- PYTHON.ORG. *What is Python? Executive Summary*. 2020. Disponível em: <<https://www.python.org/doc/essays/blurb/>>. Acesso em: 21.10.2020.
- SEAD. *Geração e Download do arquivo SEFIP.RE*. 2018. Disponível em: <https://consultas.prodam.com.br/sefip/arquivos/duvidas_mais_frequentes.asp>. Acesso em: 30.10.2020.
- SELENIUM.DEV. *The Selenium project and tools*. 2020. Disponível em: <https://www.selenium.dev/documentation/en/introduction/the_selenium_project_and_tools/>. Acesso em: 21.10.2020.
- VIVIAN, D. *O que é um certificado digital?* 2018. Disponível em: <<https://www.bry.com.br/blog/o-que-e-um-certificado-digital/>>. Acesso em: 30.10.2020.