



**INSTITUTO FEDERAL
SANTA CATARINA**

CÂMPUS FLORIANÓPOLIS

DEPARTAMENTO ACADÊMICO DE METAL MECÂNICA

CURSO SUPERIOR DE ENGENHARIA MECATRÔNICA

WILLIAM MARQUES PEREIRA

**AUTOMAÇÃO DE PROCESSOS PARA GESTÃO DE
TRABALHO *HOME-OFFICE* COM USO DE
SOLUÇÕES EM RPA**

Florianópolis - SC

2022

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA - CAMPUS FLORIANÓPOLIS**

DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA

CURSO SUPERIOR DE ENGENHARIA MECATRÔNICA

WILLIAM MARQUES PEREIRA

**AUTOMAÇÃO DE PROCESSOS PARA GESTÃO DE TRABALHO
HOME-OFFICE COM USO DE SOLUÇÕES EM RPA**

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Engenheiro em Mecatrônica.

Professor Orientador: Valdir Noll, Dr.

FLORIANÓPOLIS, DEZEMBRO DE 2022.

Ficha de identificação da obra elaborada pelo autor.

Pereira, William Marques
AUTOMAÇÃO DE PROCESSOS PARA GESTÃO DE TRABALHO HOME-OFFICE
COM USO DE SOLUÇÕES EM RPA / William Marques Pereira;
orientação de Valdir Noll. - Florianópolis, SC,
2022.
85 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal
de Santa Catarina, Câmpus Florianópolis. Bacharelado
em Engenharia Mecatrônica. Departamento
Acadêmico de Metal Mecânica.
Inclui Referências.

1. Automação Robótica de Processos (RPA). 2. Eficiência
de processos. 3. Redução do tempo de execução.
4. Low Code. 5. UiPath. I. Noll, Valdir. II. Instituto
Federal de Santa Catarina. III. AUTOMAÇÃO DE PROCESSOS
PARA GESTÃO DE TRABALHO HOME-OFFICE COM USO DE SOLUÇÕES
EM RPA.

**AUTOMAÇÃO DE PROCESSOS PARA GESTÃO DE TRABALHO HOME-OFFICE
COM USO DE SOLUÇÕES EM RPA**

WILLIAM MARQUES PEREIRA

Este trabalho foi julgado adequado para obtenção do título de Engenheiro em Mecatrônica e aprovado na sua forma final pela banca examinadora do Curso Engenharia Mecatrônica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 09 de fevereiro de 2023.

Banca Examinadora

Valdir Noll, Prof. Dr. Eng.

Maurício Edgar Stivanello, Prof. Dr. Eng.

Delcino Picinin Júnior, Prof. Dr. Eng.

Instituto Federal de Santa Catarina



MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS

DECLARAÇÃO DE FINALIZAÇÃO DE TRABALHO DE CURSO

Declaro que o estudante WILLIAM MARQUES PEREIRA, matrícula nº 1720036829, do Curso de Engenharia Mecatrônica, defendeu o trabalho intitulado ***AUTOMAÇÃO DE PROCESSOS PARA GESTÃO DE TRABALHO HOME-OFFICE COM USO DE SOLUÇÕES EM RPA***, o qual está apto a fazer parte do banco de dados da Biblioteca Hercílio Luz do Instituto Federal de Santa Catarina, Campus Florianópolis.

Florianópolis, 09 de fevereiro de 2023.

Prof. Orientador do TCC: Valdir Noll

AGRADECIMENTOS

Chegou a hora de agradecer e, por isso, começo pelos meus pais que estiveram sempre ao meu lado em todos os momentos para garantir que minha meta seria alcançada, cuidando de mim, me apoiando, me ensinando a distinguir o certo do errado, me dando muito amor e sendo o meu porto seguro. Saibam que eu tenho um amor imensurável por vocês.

Pelo ambiente maravilhoso que encontrei todos os dias nesta instituição tão respeitada eu agradeço, porque só com um clima tão inspirador é que se atingem os grandes objetivos.

A todos os professores que acompanharam meu percurso ao longo dos últimos anos eu deixo uma palavra sincera de gratidão, porque sem essa paciência e sabedoria eu jamais seria esta pessoa tão realizada.

Ao meu orientador Valdir Noll, o meu mais sincero obrigado, você foi incrível, você fez essa etapa se tornar leve, sempre me orientando da melhor maneira, cobrando da maneira certa, no momento certo e sempre com muita paciência.

À minha família e a todos os meus amigos de verdade eu quero que saibam que reconheço tudo que fizeram por mim, a força que incutiram no meu pensamento para não desistir e o conforto de saber que nunca estarei só e serei sempre capaz de tudo por maiores que sejam as dificuldades.

Não posso deixar de falar do melhor grupo que eu poderia ter, o “Precisa-se de Psicólogo”, sem vocês a faculdade teria sido muito mais difícil, sentirei saudades de cada conversa, das quartas do desabafo, das risadas, dos momentos querendo matar um ao outro, das zoeiras, sempre lembrarei do Gabriel Netto (Mini Craque) e seu aplicativo de efeito sonoro do “vai dar namoro”, Leticia Mello (Lele) e suas garrafas de água, Hassan (Hassune) e sua filosofia de vida de que qualquer conversa se resolve com “Sim” e como esquecer do IMR. Sou eternamente grato a vocês e quero levar vocês para o resto da minha vida.

À meu grande amigo Guilherme Laureano, que sempre foi um cara incrível, mas neste último ano nos aproximamos um pouco mais, ele teve um papel importantíssimo nesse meu último ano de faculdade, sempre me apoiando, ajudando, um cara com uma humildade absurda e um coração enorme. Obrigado por me ajudar a fechar esse ciclo de uma maneira tão espetacular.

A quem não mencionei, mas fez parte do meu percurso eu deixo um profundo agradecimento porque com toda certeza tiveram um papel determinante nesta etapa da minha vida.

Chego neste momento para fazer os 3 últimos agradecimentos, para as pessoas mais importantes da minha vida.

Como poderia deixar de falar dele, Bruno Marques, mais do que meu primo, mais do que um dos meus melhores amigos, um irmão pra mim, um cara que me apoia em tudo, sempre tá ali comigo, cuida de mim, dá risada comigo, briga se necessário, você sempre me ensinou muito e você sabe o quão fundamental você é na minha vida!

Quero agradecer a minha parceira Amanda Telles, que em pleno caos mundial, em meio a uma pandemia, se tornou uma das pessoas mais especiais da minha vida, que nesses últimos 3 anos me fez evoluir como nunca, me apoiou quando mais precisei, me ajudou a tomar decisões que mudaram minha vida, cuidou de mim, esteve ao meu lado em cada segundo me dando amor, carinho, respeito e muito apoio. Obrigado por tudo meu amor!

Por último, eu quero agradecer a Jonathan Marques Pereira, meu irmão, que é a pessoa mais importante da minha vida! Você é simplesmente o cara mais incrível do mundo, o meu melhor amigo, a pessoa que eu mais amo, meu parceiro, você sempre cuidou de mim desde o dia que eu nasci, sempre esteve ao meu lado em todos os momentos e essa é mais uma conquista nossa!

RESUMO

Atualmente tem crescido o número de funcionários que trabalham em *home-office*. No entanto, este fato traz consigo algumas dificuldades de controle e monitoração de entrada e saída de funcionários, alocação de horas trabalhadas por projeto, ausência programada ou não programada, etc, aumentando-se os custos de controle, especialmente se for feito de modo manual.

Este trabalho tem como objetivo automatizar o processo de análise de apontamento de horas de trabalho, junto com a facilitação da inclusão da ausência de algum colaborador por determinado tempo, reduzindo tempo e custos por meio do uso da tecnologia conhecida como *Robotic Process Automation* (RPA). Deseja-se também a interconexão digital entre a automatização do processo e um software com interface gráfica para visualização dos resultados.

A demanda será automatizada de duas formas: através do software *UiPath Studio*® que trabalha com programação *Low-Code*, e outra com a linguagem de programação *Python*, que é de código aberto. Para a execução será necessário mapear o estado atual do processo, desenhar o novo processo a ser realizado pelo robô e implementá-lo. Conjuntamente com a automatização, será desenvolvido um aplicativo utilizando o software *PowerApps*® que permite que a empresa faça o controle do apontamento de ausências de seus funcionários, incluindo também dados que sejam necessários (férias, feriados e usuários).

Após o desenvolvimento serão realizados testes para avaliar o funcionamento das soluções, e por fim, realizar a comparação dos 3 processos (humano, programação *Low-Code* e codificação em *Python*) analisando-se diversos aspectos de cada solução, de forma comparativa.

Palavras-chave: Automação Robótica de Processos (RPA). Eficiência de processos. Redução do tempo de execução. *Low Code*. *UiPath*.

ABSTRACT

Currently, the number of employees working from home has grown exponentially. However, this fact brings with it some difficulties in controlling and monitoring the entry and exit of employees, allocation of hours worked by project, scheduled or unscheduled absence, etc., increasing control costs, especially if done manually. .

This work aims to automate the process of analyzing work hours, along with facilitating the inclusion of the absence of some employees for a certain time, reducing time and costs through the use of technology known as Robotic Process Automation (RPA) . The digital interconnection between the automation of the process and a software with a graphical interface to visualize the results is also desired.

The demand will be automated in two ways: through the UiPath Studio® software that works with Low-Code programming, and another with the Python programming language, which is open source. For the execution, it will be necessary to map the current state of the process, design the new process to be carried out by the robot and implement it. Along with automation, an application will be developed using the PowerApps® software that allows the company to control the absence of its employees, also including necessary data (holidays, holidays and users).

After the development, tests will be carried out to evaluate the functioning of the solutions, and finally, the comparison of the 3 processes (human, Low-Code programming and Python coding) will be carried out, analyzing various aspects of each solution, in a comparative way.

Keywords: Robotic Process Automation (RPA). Process efficiency. Reduction of execution time. Low Code. UiPath.

LISTA DE FIGURAS

Figura 1 - Gráfico de projeção especulativa da evolução capital de RPA.	19
Figura 2 - Ciclo do processo da pesquisa-ação.	21
Figura 3 - Componentes do RPA	22
Figura 4 - Recursos do UiPath Studio	23
Figura 5 - Modelo genérico de fluxograma criado via UiPath Studio	24
Figura 6 - Modelo de sequência criado durante o desenvolvimento do projeto deste TCC.	25
Figura 7 - Modelo de máquina de estado criado durante o desenvolvimento do projeto deste TCC.	26
Figura 8 - Painel de atividades tipicamente encontrado no UiPath Studio	27
Figura 9 - Campo do fluxo de trabalho onde são inseridas as atividades.	28
Figura 10 - Layout da atividade Assign	29
Figura 11 - Layout da atividade If	29
Figura 12 - Layout da atividade Invoke Workflow File	30
Figura 13 - Layout do painel que permite escolher quais argumentos devem ser importados do fluxo de trabalho invocado, usando os controles direcionais.	30
Figura 14 - Layout da atividade Log Message	31
Figura 15 - Layout da atividade Open Browser	31
Figura 16 - Layout da atividade Navigate to	32
Figura 17 - Layout da atividade Attach Browser	32
Figura 18 - Layout da atividade Click	33
Figura 19 - Layout da atividade Build Data Table	33
Figura 20 - Layout da atividade Read CSV	34
Figura 21 - Layout da atividade For Each Row in Data Table	34
Figura 22- Exemplo de seletor feito com a atividade Click para clicar no botão “Pesquisa Google” no site google.com	35
Figura 23 - Layout Orchestrator	37
Figura 24 - Fluxograma geral do processo em nível macro	43

Figura 25 - Fluxograma do processo em nível mais detalhado mostrando os possíveis caminhos / exceções do processo atual.	44
Figura 26 - Estrutura do arquivo timelog.csv baixado do portal do Azure DevOps.	45
Figura 27 - Estrutura dentro do bloco Initialization.	46
Figura 28- Estrutura inicial de dentro da atividade “If”	47
Figura 30 - Preparação do output, checagem de saúde do ambiente e inicializa a gravação do processo.	49
Figura 31 - InitAllApplication, etapa responsável por acessar o Azure, fazer o download da planilha de apontamento e planilha de ausências.	50
Figura 32 - Passos para acessar o Azure DevOps realizado dentro do Invoke Acessa_Azure workflow	51
Figura 33 - Sequence onde é realizada a etapa responsável por deletar todos os arquivos de “timelog.csv” existente dentro da pasta de downloads	52
Figura 34 - Sequence onde realizada a etapa de baixar o arquivo “timelog.csv” mais atual	53
Figura 35 - Sequence onde realizada a etapa responsável por deletar todos os arquivos de “Alocacao*.csv” existente dentro da pasta de downloads	54
Figura 36 - Sequence onde realizada a etapa de baixar e ler o arquivo “Alocacao*.csv”	55
Figura 37 - Etapas que ocorrem dentro do “Get Transaction Data”	56
Figura 38 - Etapas que ocorrem internamente no bloco “Process Transaction”	58
Figura 39 - Etapas que ocorrem internamente no “Invoke Process workflow”	59
Figura 40 - Processo que ocorre dentro da atividade “Do While”	61
Figura 41- Processo que ocorre dentro da atividade “If” dentro do “Do While”	63
Figura 42 - Bloco “End Process”	64
Figura 43 - Exemplo de e-mail enviado para os gestores da área ao final da execução do robô	65
Figura 44 - Bloco 1 da solução em Python desenvolvida no Jupyter Notebook	66
Figura 45 - Bloco 2 da solução em Python desenvolvida no Jupyter Notebook	66
Figura 46 - Bloco 3 da solução em Python desenvolvida no Jupyter Notebook	67
Figura 47 - Bloco 4 da solução em Python desenvolvida no Jupyter Notebook	68
Figura 48 - Bloco 5 da solução em Python desenvolvida no Jupyter Notebook	68

Figura 49 - Tela inicial na versão mobile do aplicativo corporativo desenvolvido via PowerApps 69

Figura 50 - Tela de ausências na versão mobile do aplicativo corporativo desenvolvido via PowerApps 70

Figura 51 - Tela de feriados na versão mobile do aplicativo corporativo desenvolvido via PowerApps 71

Figura 52 - Tela de usuários na versão mobile do aplicativo corporativo desenvolvido via PowerApps 72

LISTA DE TABELAS

Tabela 1 - Licenças UiPath

37

LISTA DE ILUSTRAÇÕES

Gráfico 1 - Testes manuais de execução do processo	73
Gráfico 2 - Testes utilizando a solução completa desenvolvida no UiPath	74
Gráfico 3 - Testes utilizando a parte comparativa desenvolvida no UiPath	75
Gráfico 4 - Testes realizados com a solução em Python	76
Gráfico 5 - Comparativo de Tempo X Meses das soluções desenvolvidas em relação às etapas equivalentes ao processo manual.	77
Gráfico 6 - Comparativo de Tempo X Meses da solução completa do processo desenvolvido com o UiPath em relação ao processo manual.	78

LISTA DE ABREVIATURAS OU SIGLAS

GUI - Graphical User Interface / Interface Gráfica do Usuário

RPA - Robotic Process Automation / Automação de Processos Robóticos

OCR - Optical Character Recognition / Reconhecimento ótico de caracteres

IA - Inteligência Artificial

SUMÁRIO

1. INTRODUÇÃO	15
1.1 Justificativa e Definição do Problema	15
1.2. Objetivo Geral	16
1.3. Objetivo específico	16
2. FUNDAMENTAÇÃO TEÓRICA	17
2.1. Sobre RPA	17
2.1.1 Competências e metodologia	19
2.2. Software RPA UiPath	21
2.2.1 UiPath Studio	23
2.2.1.1 Activities/Atividades	26
2.2.1.2 Selectors/Seletores	34
2.2.2 UiPath Robot	35
2.2.2.1 Tipos de Robôs	36
2.2.3 UiPath Orchestrator	36
2.2.5 Licenciamento	37
2.3. Linguagem de Programação Python	38
2.4. Microsoft Power Platform	39
2.4.1 Power Apps	39
2.4.2 Sharepoint	40
2.4.3 Licenças Power Platform	40
2.5. Canal direto usando Azure	41
2.6. Sobre o desafio	41
3. DESENVOLVIMENTO PRÁTICO	43
3.1. Planejamento	43
3.2. Desenho do processo	43

3.3. Desenvolvimento do processo	45
3.3.1 Processo desenvolvido utilizando UiPath	45
3.3.2 Processo desenvolvido utilizando Python	65
3.3.3 Interface visual do PowerApps	68
4. RESULTADOS OBTIDOS	73
4.1. Processo executado manualmente	73
4.2. Testes utilizando UiPath	73
4.3. Testes utilizando Python	75
4.4. Comparação das soluções	76
5. CONCLUSÃO	80
REFERÊNCIAS	82

1. INTRODUÇÃO

1.1 Justificativa e Definição do Problema

Nos últimos anos, o RPA vem ganhando força e popularidade como uma oportunidade de crescimento do produto e para agilizar processos, trazendo mais benefícios para o negócio. Segundo Costello e Rimol (2020) a previsão do mercado de RPA crescerá a taxas de dois dígitos até 2024, e apesar das pressões econômicas do COVID-19 a receita global de software de automação de processos robóticos (RPA) teve a projeção de US\$ 1,89 bilhão em 2021, um aumento de 19,5% em relação a 2020, de acordo com a última previsão do Gartner, Inc.

Como resultado, é necessário um melhor entendimento desta tecnologia, que reflete as vantagens e desvantagens do RPA, e como sua implementação visa gerar benefícios financeiros para as organizações. Este trabalho justifica e encapsula seu valor no potencial de atuar como ferramenta consultiva para organizações e programadores que buscam desenvolver soluções semelhantes para uso pessoal ou profissional.

A importância do RPA para as empresas dos mais variados ramos, é oferecer uma nova opção para estas, não significando que o trabalho manual não seria tão bem executando quanto, mas ter um “robô de software” que realize um processo definido pela empresa, de maneira automática, seguindo uma sequência de um processo que foi mapeado.

A escolha do software *Power Apps* para o desenvolvimento da Interface Humano-máquina, que é basicamente um aplicativo onde será realizado a inclusão de ausências e monitoramento destas, foi devido ao fato de que, com ele, os usuários de negócios, ou seja, pessoas que têm pouco conhecimento em programação possam desenvolver uma aplicação multiplataforma, utilizando o conceito de programação *Low-Code* tendo enfoque principal em soluções corporativas. Sendo assim, tal solução pode ser utilizada para sustentar algum modelo de negócio, no caso deste projeto uma solução para gestão e controle dos funcionários que trabalham na modalidade *home-office*.

1.2. Objetivo Geral

Este trabalho tem como objetivo principal desenvolver duas soluções de RPA, utilizando *UiPath Studio®* e *Python*, para reduzir o tempo de execução de uma demanda dentro

da área de gestão de pessoas, a fim de automatizar o processo de captura de informações do funcionário da empresa sobre seu trabalho em *home-office*, e atualizar estes dados em uma planilha.

Tais soluções serão realizadas usando a metodologia de pesquisa-ação, tendo em mente os principais conceitos relacionados aos métodos de programação para desenvolvimento do processo e com objetivo de demonstrar a eficiência de se automatizar processos manuais.

Além das duas soluções citadas acima, será desenvolvido um aplicativo para controle à distância e inclusão de ausências utilizando o *Power Apps*.

1.3. Objetivo específico

Elenca-se a partir do objetivo principal as etapas a serem desenvolvidos durante o desenvolvimento deste projeto:

- Estruturar o cenário atual do processo a ser automatizado;
- Documentar a automatização do processo;
- Automatizar o processo utilizando *software UiPath Studio®*;
- Automatizar o processo utilizando linguagem de programação *Python*;
- Criação do aplicativo via *Power Apps*;
- Testes de eficiência e redução de tempo;
- Comparação dos métodos.

2. FUNDAMENTAÇÃO TEÓRICA

Neste tópico as ferramentas utilizadas no desenvolvimento da solução são brevemente discutidas. São abordados diferentes aspectos dos *softwares* utilizados, bem como a metodologia utilizada para maximizar o uso dos recursos disponíveis e necessários.

2.1. Sobre RPA

A automação de processos robóticos (RPA) é uma ferramenta genérica de produtividade que permite ao usuário configurar um ou mais *scripts* (que alguns fornecedores chamam de “*bots*”) que interagem com elementos da GUI (*Graphical User Interface*) para concluir tarefas repetitivas baseadas em regras. O resultado é que os *bots* podem ser usados para imitar ou emular tarefas selecionadas (etapas de transação) dentro de um negócio geral ou processo de Tecnologia de Informação (TI). Inclui manipulação de dados, transmissão de dados de e/ou para diferentes aplicativos, acionamento de respostas ou execução de transações entre aplicativos. O RPA usa uma combinação de interação de interface do usuário e tecnologias de descritor. Os *scripts* (o programa em si) podem se sobrepor a um ou mais aplicativos de software.

Em vez de depender de códigos de programa feitos por um programador experiente, como é o caso do *python*, muitos dos *softwares* de RPA permite que os indivíduos estabeleçam automações e gerenciem fluxos de trabalho usando recursos como "arrasta e solta", que são totalmente visuais e não exigem conhecimento aprofundado de programação.

Além disso, ao contrário de outras ferramentas de captura de tela, alguns softwares RPA usam reconhecimento de caracteres (OCR) para se adaptar automaticamente às alterações de página sem o envolvimento contínuo de um trabalhador humano. Isso é possível porque o RPA também foi construído usando inteligência artificial. Na realidade, podemos dizer que RPA é uma combinação de inteligência artificial (IA) e automação. Os robôs RPA são programados para manipular dados e aplicativos da mesma forma que uma pessoa faria, por exemplo, usando um software. Essa tecnologia é extremamente eficaz, pois os robôs de automação robótica de processos (RPA) reconhecem padrões, realizam a comunicação do sistema, adaptam-se, tomam decisões e realizam uma ampla variedade de tarefas repetitivas, o robô não comete erros aleatórios (ainda que possam cometer erros sistemáticos por erros de programação) e minimiza os custos operacionais.

De acordo com TRIPATHI (2018, p. 11, tradução livre)

Hoje, o RPA amadureceu além de fazer tarefas repetitivas e mundanas e é visto como uma tecnologia transformacional que pode trazer um enorme valor para a organização que o adota. A capacidade de criar trilhas de auditoria completas é significativa para melhorar a qualidade do trabalho sendo feito e eliminar o erro humano. Depois de treinados, esses robôs realizarão tarefas com a mesma precisão repetidamente. Esses robôs podem interagir com aplicativos independentemente das tecnologias nas quais os aplicativos são construídos. Eles podem trabalhar com aplicativos populares de ERP, como SAP, Oracle ou Microsoft Dynamics e BPMs como sistemas Pega e Appian.

Mais e mais empresas estão adotando o digital todos os dias. Com a digitalização, a maior vantagem é a velocidade de execução. Mas o desafio da digitalização é que ela requer várias ferramentas e, portanto, uma força de trabalho multi qualificada para lidar com elas. Mas a mão de obra com múltiplas habilidades é escassa. Para resolver esses problemas, todo o departamento de TI está sempre em busca de soluções confiáveis, rápidas, inteligentes e poderosas. Esse requisito é atendido por meio do caminho da interface do usuário.

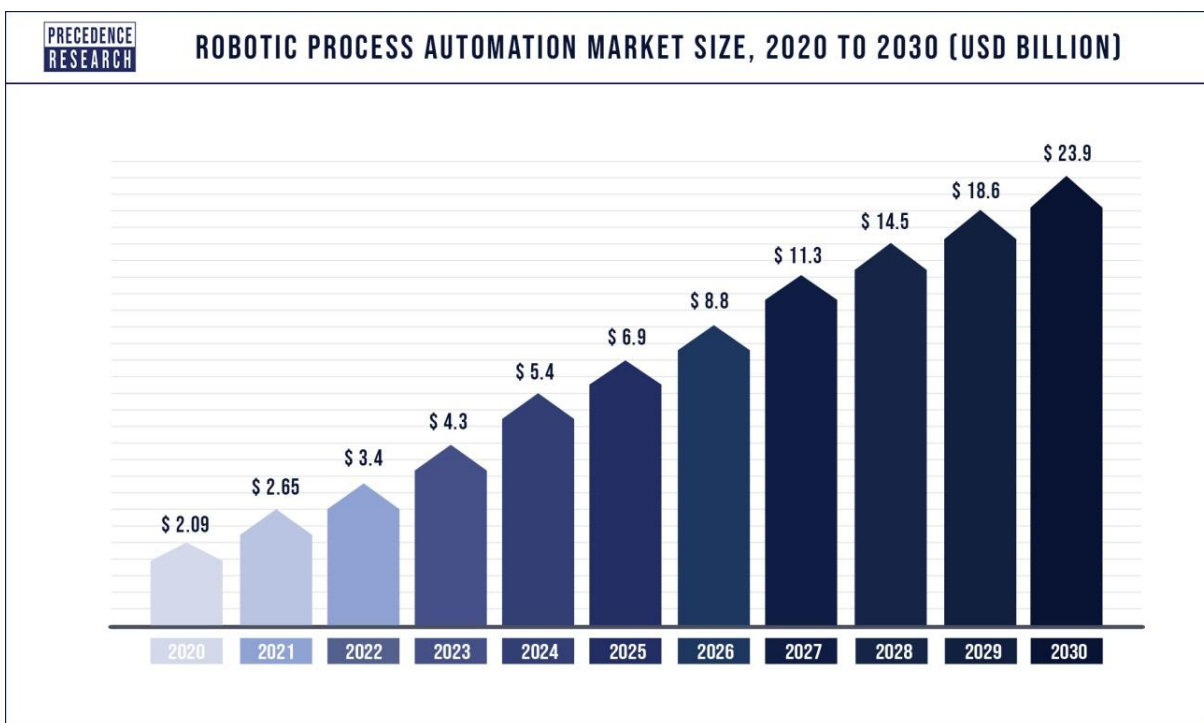
Nesse sentido é importante ressaltar a seguinte colocação:

A receita global do mercado de software de Automação Robótica de Processos (RPA) deve atingir cerca de US\$ 2,9 bilhões em 2022, o que representa um aumento de 19,5% em relação a 2021, segundo pesquisa do Gartner, Inc., líder mundial em pesquisa e aconselhamento para empresas. (ANÁPOLIS, 2022).

Como ressaltado na figura 1 e na citação abaixo:

Recente relatório publicado pela instituição de pesquisa *ReportsnReports* aponta que o mercado de RPA e Hiper automação deve alcançar US \$26 bilhões até 2027. Intitulada *RPA and Hyperautomation Market Global Business Function (IT, Operations & Supply Chain, and HR), Key Players and Forecasts to 2027*, a pesquisa revela que os mercados citados devem ter um crescimento anual composto (CAGR) de 23.1% entre 2022 e 2027, passando de US\$ 9.2 bilhões a US\$ 26 bilhões. (ANÁPOLIS, 2022).

Figura 1 - Gráfico de projeção especulativa da evolução capital de RPA.



Fonte: Globenewswire (2022).

2.1.1 Competências e metodologia

Segundo Thiollent (2009), em Metodologia da Pesquisa-ação, pode-se definir como:

”um tipo de pesquisa social com base empírica que é concebida e realizada em estreita associação com uma ação ou com a resolução de um problema coletivo e no qual os pesquisadores e os participantes representativos da situação ou do problema estão envolvidos de modo cooperativo ou participativo”. (THIOLLENT, 2009, p.16).

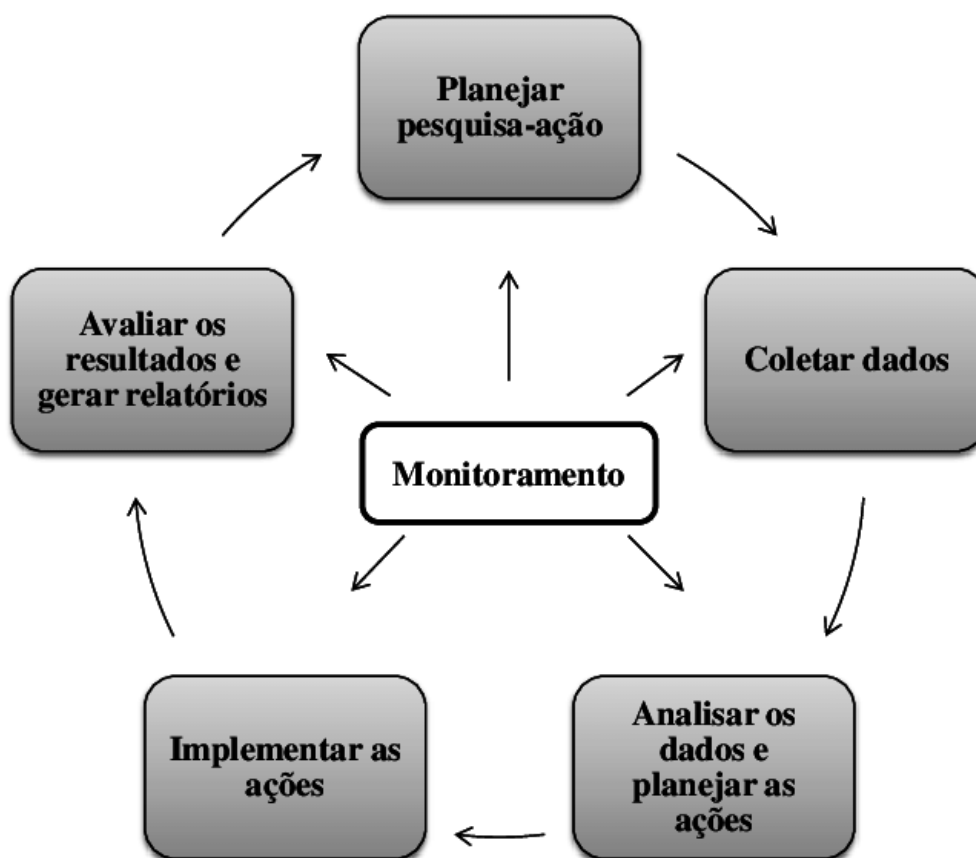
O ciclo do processo da pesquisa-ação ocorre em cinco etapas, conforme demonstrado na figura 2, porém todas as etapas devem ser rigorosamente monitoradas e sistematizadas. As etapas são as seguintes:

- Pesquisa-ação, também pode ser denominada como identificação do problema, onde o pesquisador deve identificar o problema ou definir a situação que será investigada. A identificação do problema cria a questão de pesquisa, seu propósito e escopo. No caso do projeto o problema principal está vinculado ao acompanhamento do horário de trabalho dos funcionários que trabalham em home-office;
- Coleta de dados, estes dados podem ser adquiridos das mais diversas formas e fontes, após a coleta desses dados passa-se pela parte de filtragem destes para

as que melhor se relacionarem com o caso. No caso atual, será utilizado o *timelog* para realizar a consulta dos dias, horários e tarefas em que o funcionário está vinculado;

- Análise dos dados e planejamento das ações, a interpretação e análise dos dados dependerá do foco de investigação. Outros dados podem ser medidos (frequência, número de ocorrências, resultados de testes). Com os dados do *timelog* em mãos, será analisado formas de interpretar esses dados, como extrair e manipular tais informações de maneira automatizada;
- Implementar ações, deve-se utilizar os resultados obtidos na etapa anterior para apresentar um plano de ação que possibilite a mudança e a viabilidade de estudar a implementação dessa mudança. Para o caso estudado será feita a implementação das soluções de RPA para uma manipulação dos dados de maneira mais rápida e eficiente, com o auxílio de um aplicativo para melhor visualização;
- Avaliar resultados e gerar relatórios, avaliação dos resultados de uma ação ou intervenção usada para resolver o problema original. A avaliação é necessária para determinar se a condição melhorou, nesse caso será feito a interpretação, tempo de execução da automação, custos e manipulação dos dados.

Figura 2 - Ciclo do processo da pesquisa-ação.



Fonte:

https://www.researchgate.net/figure/Figura-1-Estrutura-para-conducao-da-pesquisa-acao_fig1_312039546, Visto em 2022.

Portanto, o planejamento e desenvolvimento deste trabalho será feito a partir das etapas citadas no parágrafo anterior.

2.2. Software RPA UiPath

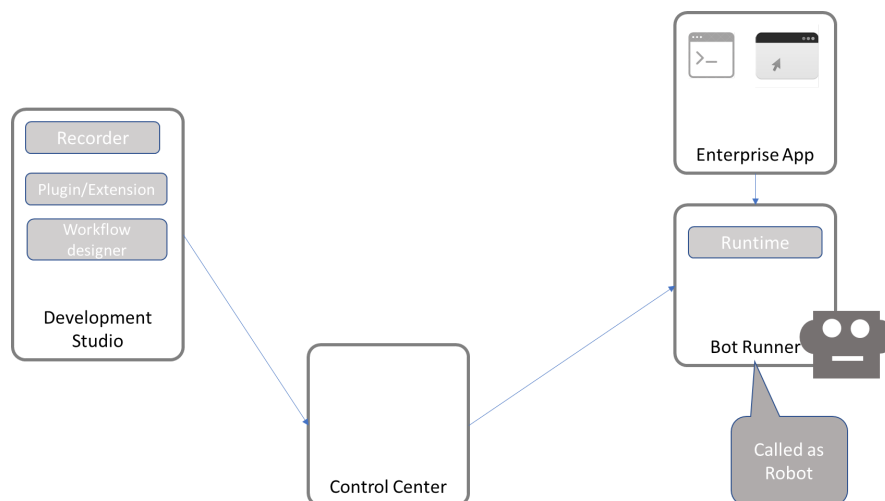
O *UiPath* criado por Daniel Dines é um fornecedor de tecnologia RPA que projeta e fornece software que ajuda a automatizar negócios de ponta a ponta em escala. A plataforma de RPA consiste em três partes:

- *UiPath Studio* para projetar os processos;
- *UiPath Robot* para automatizar tarefas projetadas no *UiPath Studio*;
- *UiPath Orchestrator* para executar e gerenciar os processos.

Como mostra a figura 3, “qualquer plataforma de RPA fornece alguns componentes básicos, que juntos constroem a plataforma” (TRIPATHI, 2018, p. 11, tradução livre), composto por:

- Recorder - é um dos componentes críticos do RPA. Ele adiciona a capacidade de automatizar aplicativos da Web, desktop e mainframe de maneira natural semelhante a uma macro, sem a necessidade de qualquer programação, codificação ou script;
- Development Studio - ajuda a projetar ou desenvolver fluxos de trabalho inteligentes de automação de processos e também permite que você instale pacotes de atividades, assistentes, gravadores e plug-ins personalizados.
- Plugin/Extensions - são o conjunto de programas que podem ser instalados junto com a ferramenta RPA, esses plug-ins lidam com diferentes tipos de tarefas, reduzindo os esforços de desenvolvimento;
- Bot runner - são usados para executar os bots de *softwares* desenvolvidos;
- Control center - uma plataforma baseada na web usada para controlar os bots de software criados, ele permite que os usuários agendem, gerenciem, controlem e dimensionem a atividade de uma grande quantidade de força de trabalho digital.

Figura 3 - Componentes do RPA

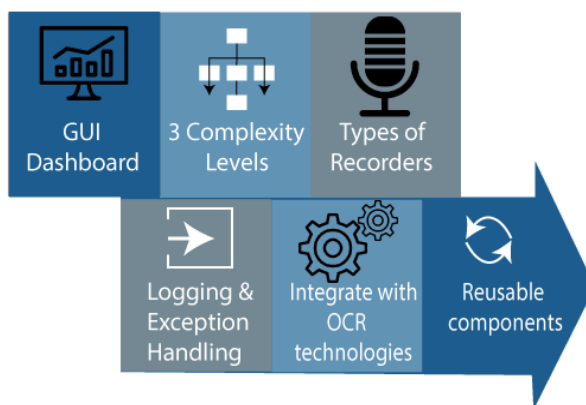


Fonte: TRIPATHI (2018).

2.2.1 UiPath Studio

UiPath Studio é um dos produtos oferecidos pela *UiPath*. É um *dashboard*, onde você pode criar seu fluxo de trabalho de automação e com a ajuda de atividades executar a automação projetada como visto na figura 4. “Sua simplicidade permite que até mesmo usuários de negócios não técnicos projetem/registrem etapas de um processo. Este *software* permite ao usuário configurar Robôs, ou seja, desenvolver etapas para realizar tarefas visualmente.” (TRIPATHI, 2018, p. 24, tradução livre).

Figura 4 - Recursos do *UiPath Studio*



Fonte: Example (2022).

GUI Dashboard - Oferece um painel *GUI* que consiste em atividades predefinidas para criar fluxos de trabalho de automação.

Complexity Levels - Permite criar projetos baseados em níveis de complexidade como Sequência, Fluxograma e Máquina de Estado.

Types of Recorders - Oferece vários tipos de gravadores para gravar ações em várias plataformas.

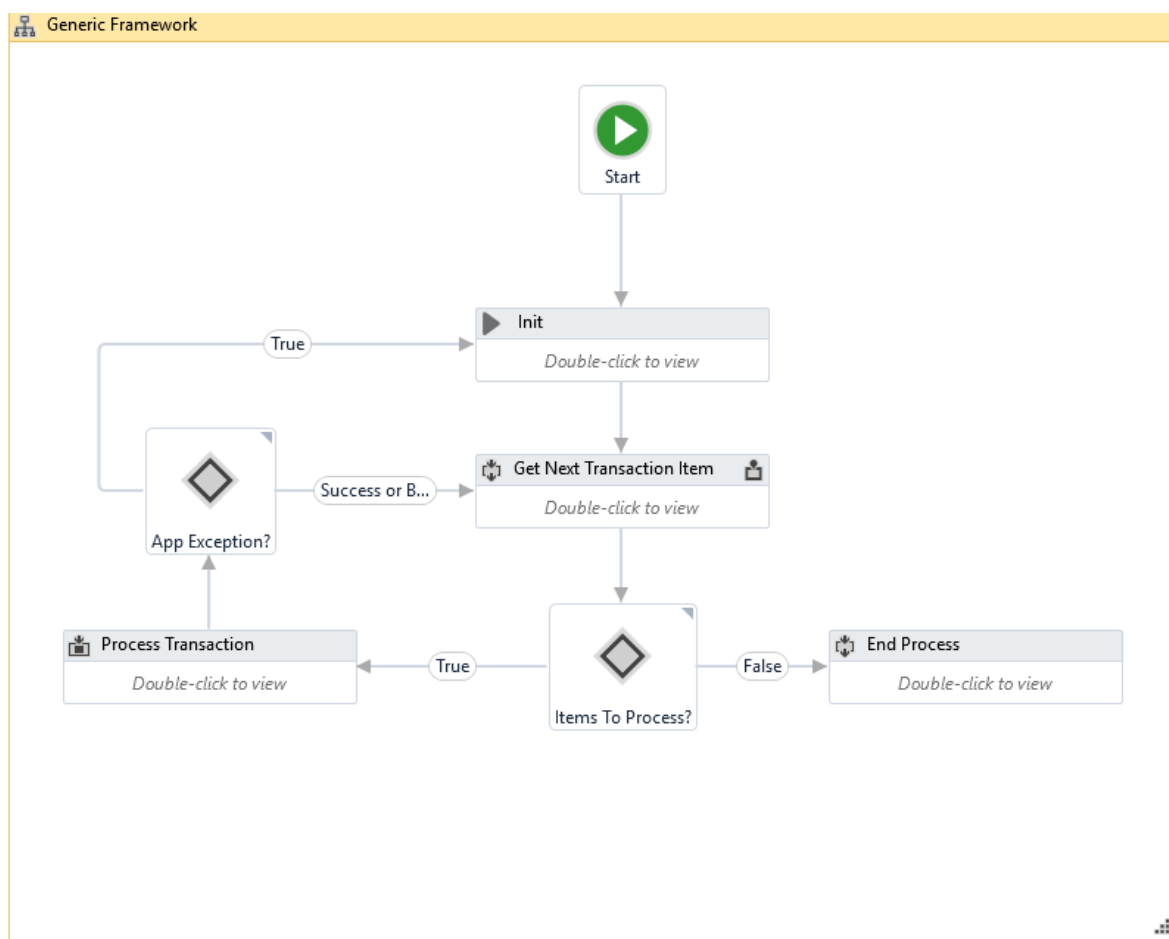
Logging & Exception Handling - Permite que você execute depuração e tratamento de exceções.

Integrate with OCR technologies - O *UiPath Studio* pode se integrar com várias tecnologias de OCR para realizar a captura de tela.

Reusable Components - Com o *UiPath Studio*, pode ser criado componentes reutilizáveis para publicá-los juntos como bibliotecas.

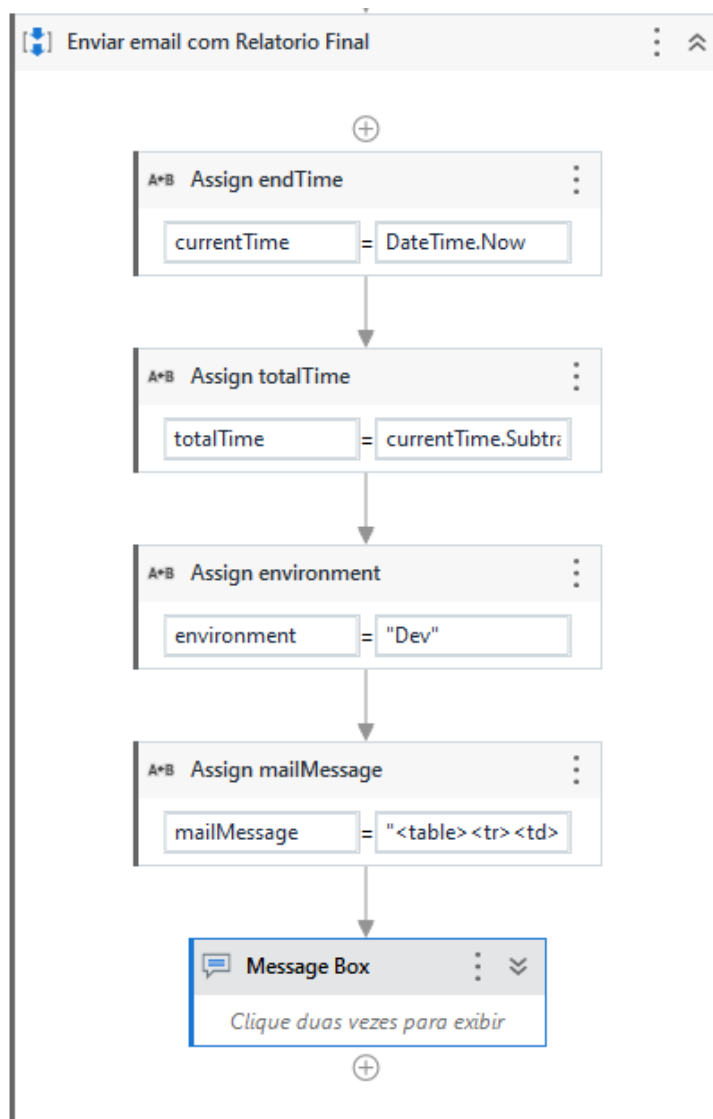
Visando demonstrar como é simplificado o processo de criação de uso do *UiPath*, como mostra as Figuras 5 a 7, há três maneiras de se confeccionar o desenvolvimento do robô no *UiPath Studio*: através de fluxogramas, de sequências ou de máquinas de estado. O usuário pode escolher “escrever” o programa da forma que mais lhe agrada.

Figura 5 - Modelo genérico de fluxograma criado via *UiPath Studio*



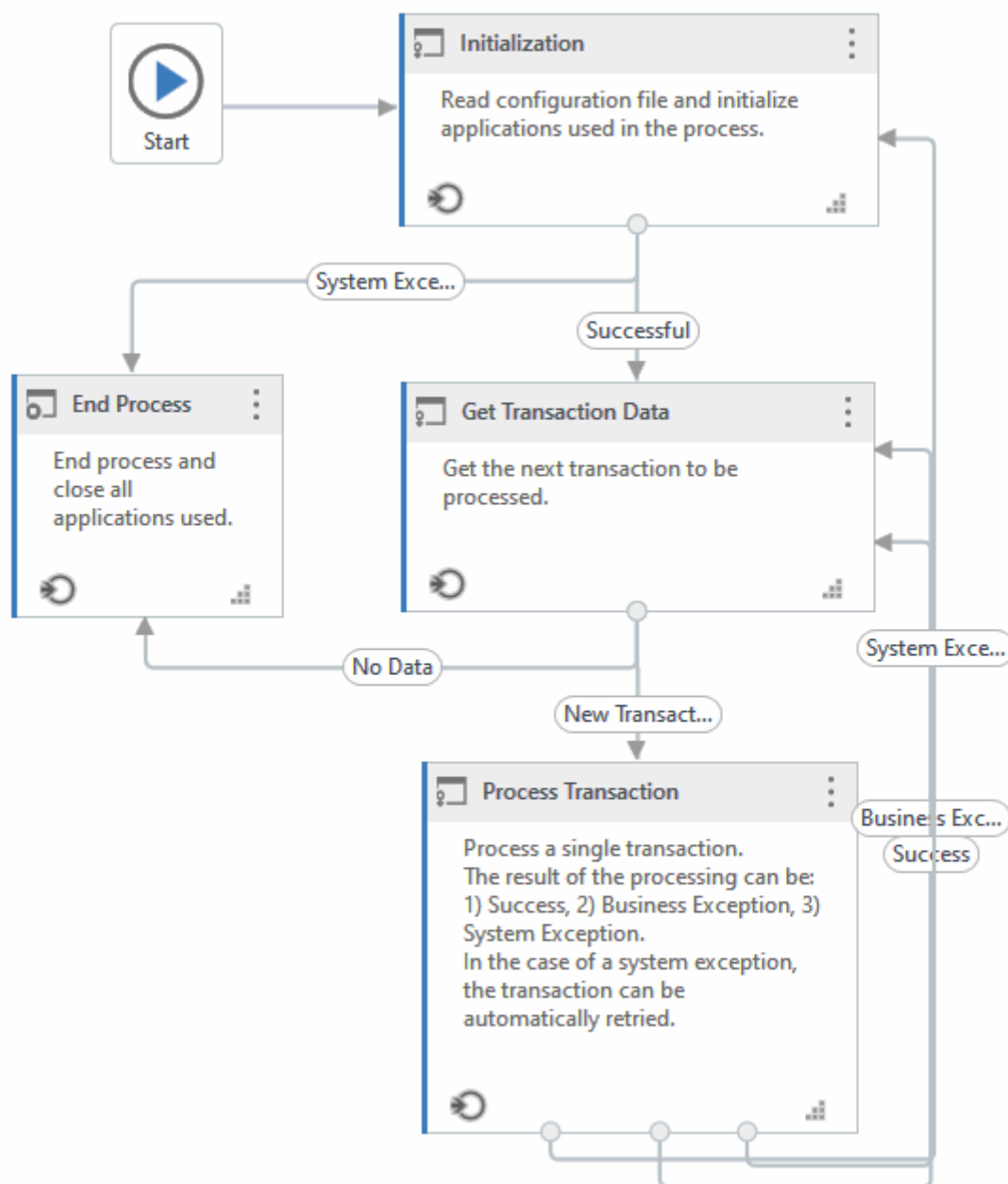
Fonte: *UiPath* (2022).

Figura 6 - Modelo de sequência criado durante o desenvolvimento do projeto deste TCC.



Fonte: Do autor (2022).

Figura 7 - Modelo de máquina de estado criado durante o desenvolvimento do projeto deste TCC.



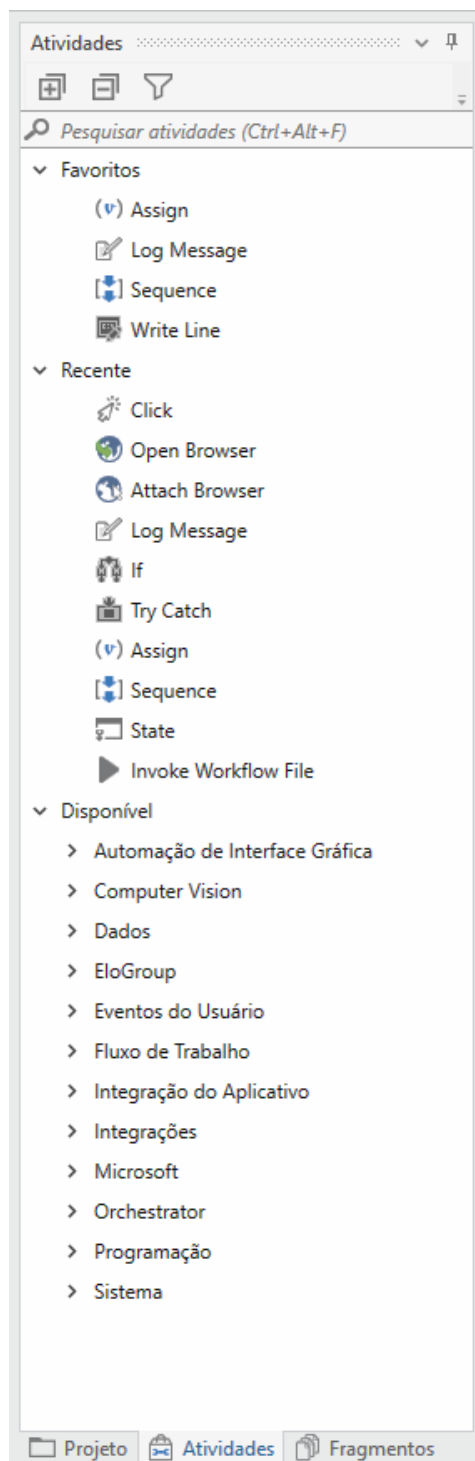
Fonte: Do autor (2022).

2.2.1.1 Activities/Atividades

De acordo com (TRIPATHI, 2018, p. 11, tradução livre) “No *UiPath Studio*, uma atividade representa a unidade de uma ação. Cada atividade executa alguma ação. Quando as atividades se combinam, torna-se um processo.”.

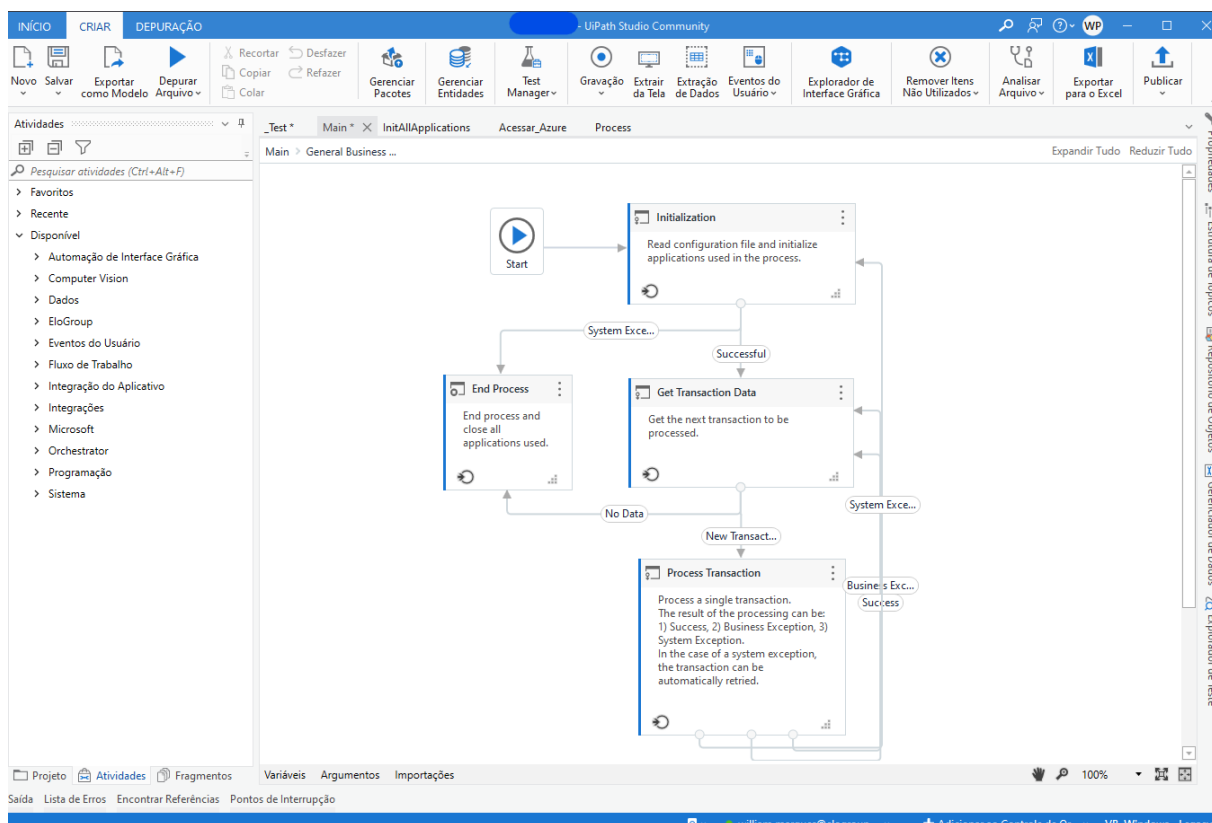
Estas ações podem ser facilmente usadas na criação de um projeto simplesmente arrastando do painel de atividades (Figura 8) e soltando a atividade necessária no campo do fluxo de trabalho (Figura 9).

Figura 8 - Painel de atividades tipicamente encontrado no *UiPath Studio*



Fonte: Do autor (2022).

Figura 9 - Campo do fluxo de trabalho onde são inseridas as atividades.



Fonte: Do autor (2022).

Descreve-se a seguir detalhes de ações utilizadas para desenvolver a solução desejada neste projeto:

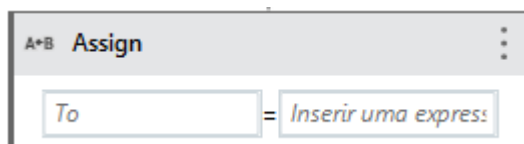
- *Sequence*: De acordo com o manual de utilização do *UiPath*, esta ação “Permite criar processos lineares compostos por muitas atividades filhas, que são executadas em ordem sequencial.” (*UiPath*, 2020, tradução livre).
- *Try-Catch*: Segundo o manual, esta ação “Captura um tipo de exceção especificado em uma sequência ou atividade e exibe uma notificação de erro ou a descarta e continua a execução.” (*UiPath*, 2020, tradução livre).
 - *Try* - contém a atividade que pode lançar uma exceção;
 - *Catches* - Indica o tipo de exceção e opcionalmente mantém uma atividade que informa o usuário sobre a exceção encontrada;
 - *Finally* - Esta seção é executada somente quando nenhuma exceção é lançada ou quando ocorre um erro e é detectado na seção *Catches*.
- *Assign*: Segundo *UiPath* (2022, tradução livre) esta ação:

Atribui qualquer valor para uma variável ou argumento. Ele pode ser usado para incrementar o valor de uma variável em um *loop*, somar o valor de duas ou mais variáveis e atribuir o resultado a uma variável diferente, atribuir valores a uma matriz e assim por diante. Por padrão, essa atividade também está incluída no grupo “Favoritos”.

Seu layout pode ser observado na figura 10.

- Os parâmetros de entrada (*inputs*) para execução dessa atividade são:
 - *To* - O nome da variável à qual será atribuído um valor;
 - *Value* - O valor que você deseja atribuir à variável.

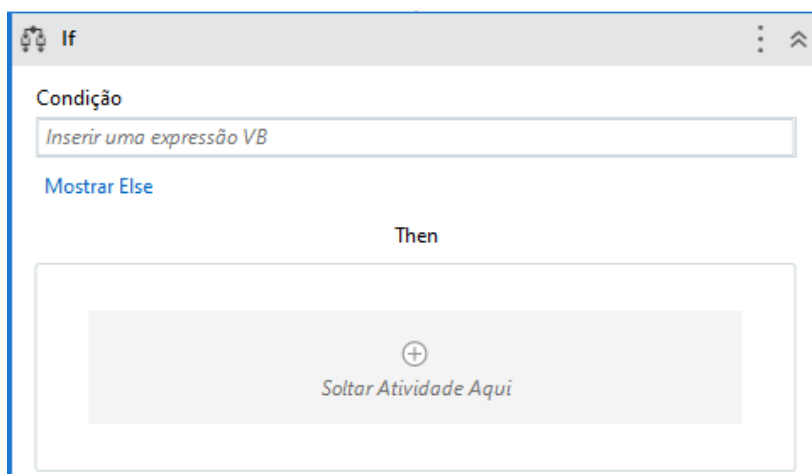
Figura 10 - *Layout* da atividade *Assign*



Fonte: Do autor (2022).

- *If*: Segundo o manual da *UiPath*, esta ação “permite que seu projeto tome um dos dois cursos de ação diferentes, dependendo se uma condição especificada for atendida. Esta atividade contém uma condição, uma ramificação “Then” e uma ramificação “Else” opcional.” (*UiPath*, 2021, tradução livre). Seu layout pode ser observado na figura 11.

Figura 11 - *Layout* da atividade *If*



Fonte: Do autor (2022).

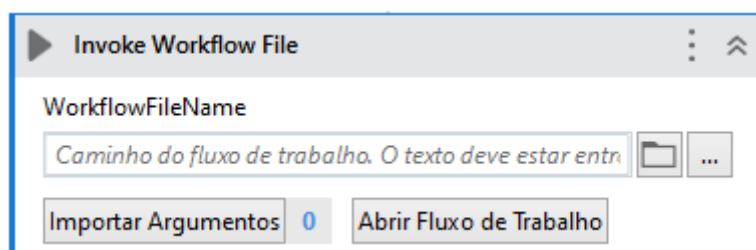
- *Invoke Workflow File*: De acordo com o *UiPath* (2022, tradução livre):

Invoca de forma síncrona um fluxo de trabalho especificado, opcionalmente passando a ele uma lista de argumentos de entrada. Os argumentos podem ser importados de um fluxo de trabalho especificado com essa atividade usando o botão Importar argumentos no corpo da atividade.

Seu layout pode ser observado nas figuras 12 e 13.

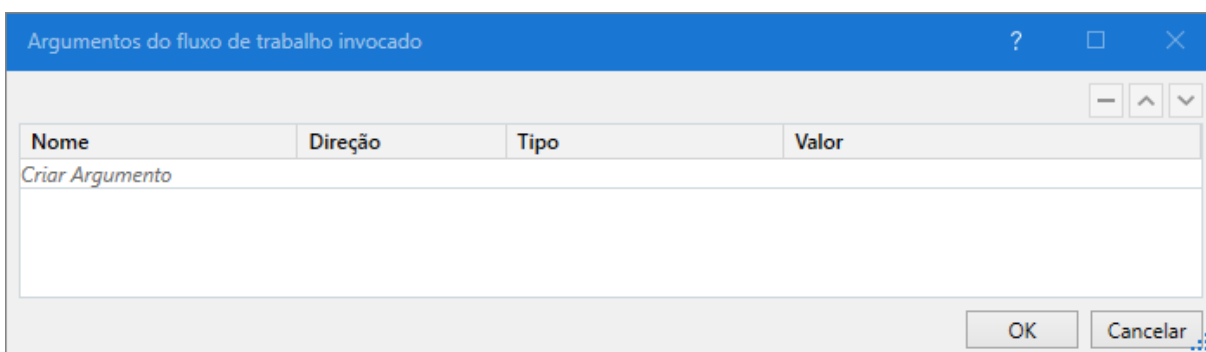
- Os parâmetros de entrada (*inputs*) para execução dessa atividade são:
 - Importar Argumentos - Os parâmetros que podem ser passados para a aplicação no momento da chamada.
 - WorkflowFileName - O caminho do arquivo no formato "xml" a ser invocado.

Figura 12 - *Layout* da atividade *Invoke Workflow File*



Fonte: Do autor (2022).

Figura 13 - *Layout* do painel que permite escolher quais argumentos devem ser importados do fluxo de trabalho invocado, usando os controles direcionais.

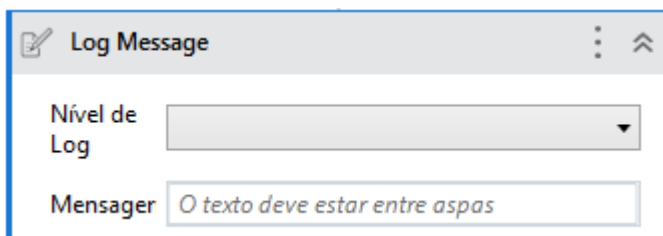


Fonte: Do autor (2022).

- *Log Message*: Esta ação “grava a mensagem de diagnóstico especificada no nível especificado. Essas mensagens também são enviadas ao orquestrador e exibidas na página *Logs*.” (UiPath, 2022, tradução livre). Seu layout pode ser observado na figura 14.
 - As entradas para execução dessa atividade são:

- *Nível de Log* - A gravidade da mensagem de login. As seguintes opções estão disponíveis: *Trace*, *Info*, *Warn*, *Error* e *Fatal*;
- *Messenger* - A mensagem que você deseja registrar.

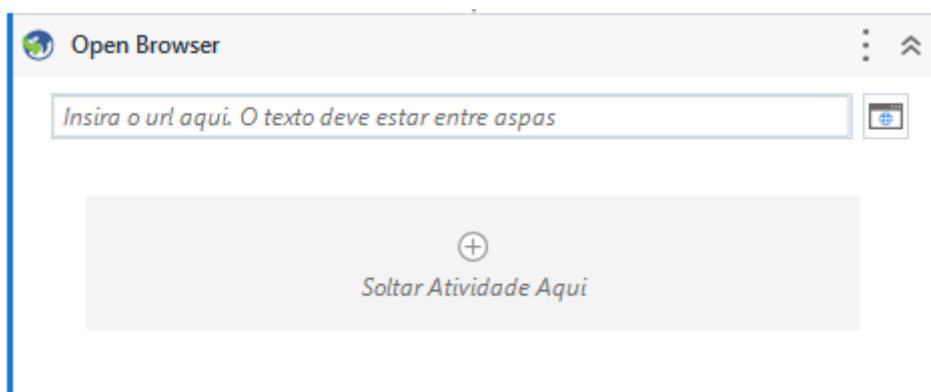
Figura 14 - *Layout da atividade Log Message*



Fonte: Do autor (2022).

- *Open Browser*: Esta ação é “um contêiner que permite abrir um navegador em um URL especificado e executar várias atividades nele.” (UiPath, 2022, tradução livre). Seu layout pode ser observado na figura 15.
 - O *input* para execução dessa atividade é:
 - *BrowserType* - Escolha o tipo de navegador que você deseja usar usando o botão suspenso. As seguintes opções estão disponíveis: IE, Firefox, Chrome, Edge, Custom.

Figura 15 - *Layout da atividade Open Browser*

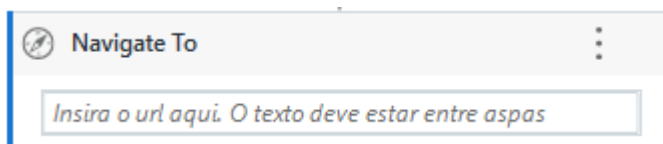


Fonte: Do autor (2022).

- *Navigate to*: Navega o navegador para um determinado *URL*. Seu *layout* pode ser observado na figura 16.
 - Os *inputs* para execução dessa atividade são:
 - *Url* - endereço da página web a ser exibida;

- *Browser* - navegador no qual você deseja navegar para a página especificada.

Figura 16 - *Layout* da atividade *Navigate to*

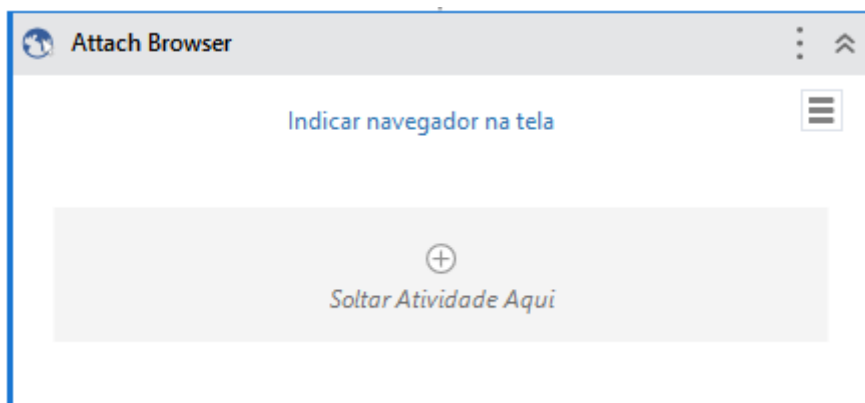


Fonte: Do autor (2022).

- *Attach Browser*: Esta ação é “um contêiner que permite anexar a um navegador já aberto e executar várias ações nele.” (UiPath, 2022, tradução livre). Seu layout pode ser observado na figura 17.

- Os *inputs* para execução dessa atividade são:
 - *Selector* - Propriedade de texto usada para localizar um elemento de interface do usuário específico quando a atividade é executada;
 - *BrowserType* - Escolha o tipo de navegador que você deseja usar. As seguintes opções estão disponíveis: *IE*, *Firefox*, *Chrome*, *Edge*, *Custom*.

Figura 17 - *Layout* da atividade *Attach Browser*



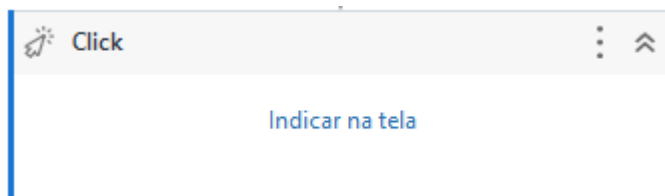
Fonte: Do autor (2022).

- *Click*: Clica em um elemento de interface do usuário especificado. Seu layout pode ser observado na figura 18.

- Os *inputs* para execução dessa atividade são:
 - *ClickType* - Especifica o tipo de clique do mouse (simples, duplo, para cima, para baixo);

- *Target.Selector* - Propriedade de texto usada para localizar um elemento de interface do usuário específico quando a atividade é executada.

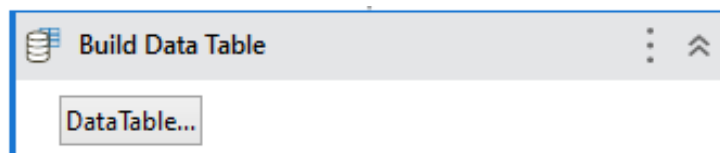
Figura 18 - *Layout* da atividade *Click*



Fonte: Do autor (2022).

- *Build Data Table*: Cria um *DataTable* de acordo com um esquema especificado. Seu layout pode ser observado na figura 19.
 - O *output* para execução dessa atividade é:
 - *DataTable* - Uma variável *DataTable* que armazena as informações adicionadas na janela *Build Data Table*.

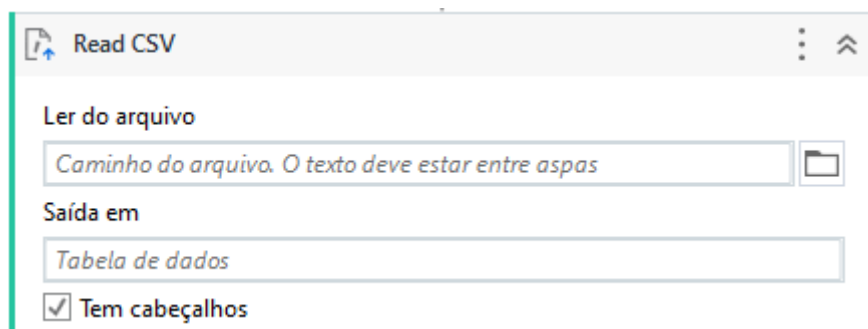
Figura 19 - *Layout* da atividade *Build Data Table*



Fonte: Do autor (2022).

- *Read CSV*: Lê todas as entradas de um arquivo CSV especificado. Seu layout pode ser observado na figura 20.
 - O *input* para execução dessa atividade é:
 - *FilePath* - Caminho completo do arquivo CSV.
 - O *output* para execução dessa atividade é:
 - *DataTable* - Representa o *DataTable* de saída que contém as informações obtidas do arquivo CSV.

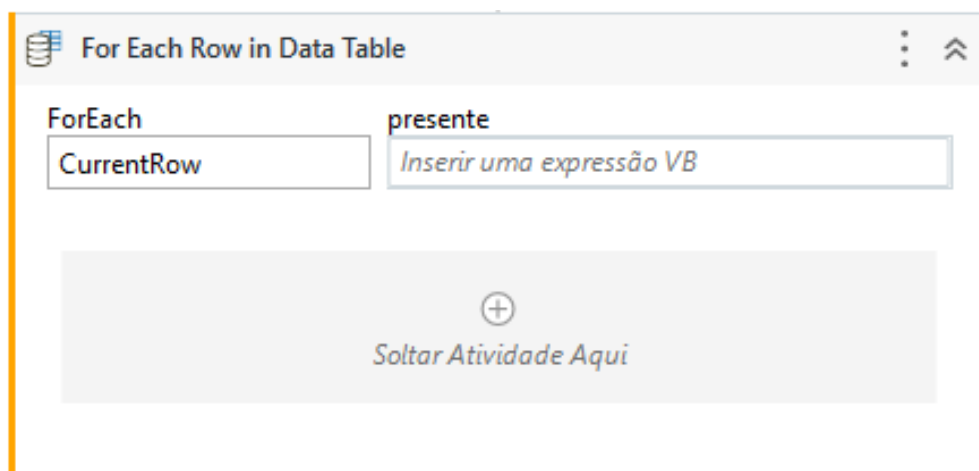
Figura 20 - Layout da atividade Read CSV



Fonte: Do autor (2022).

- *For Each Row in Data Table*: Executa uma ou mais ações uma vez para cada linha em uma variável *DataTable* especificada. Seu layout pode ser observado na figura 21.
 - O *input* para execução dessa atividade é:
 - *DataTable* - A variável *DataTable* para a qual uma ação deve ser executada uma vez para cada linha.

Figura 21 - Layout da atividade For Each Row in Data Table



Fonte: Do autor (2022).

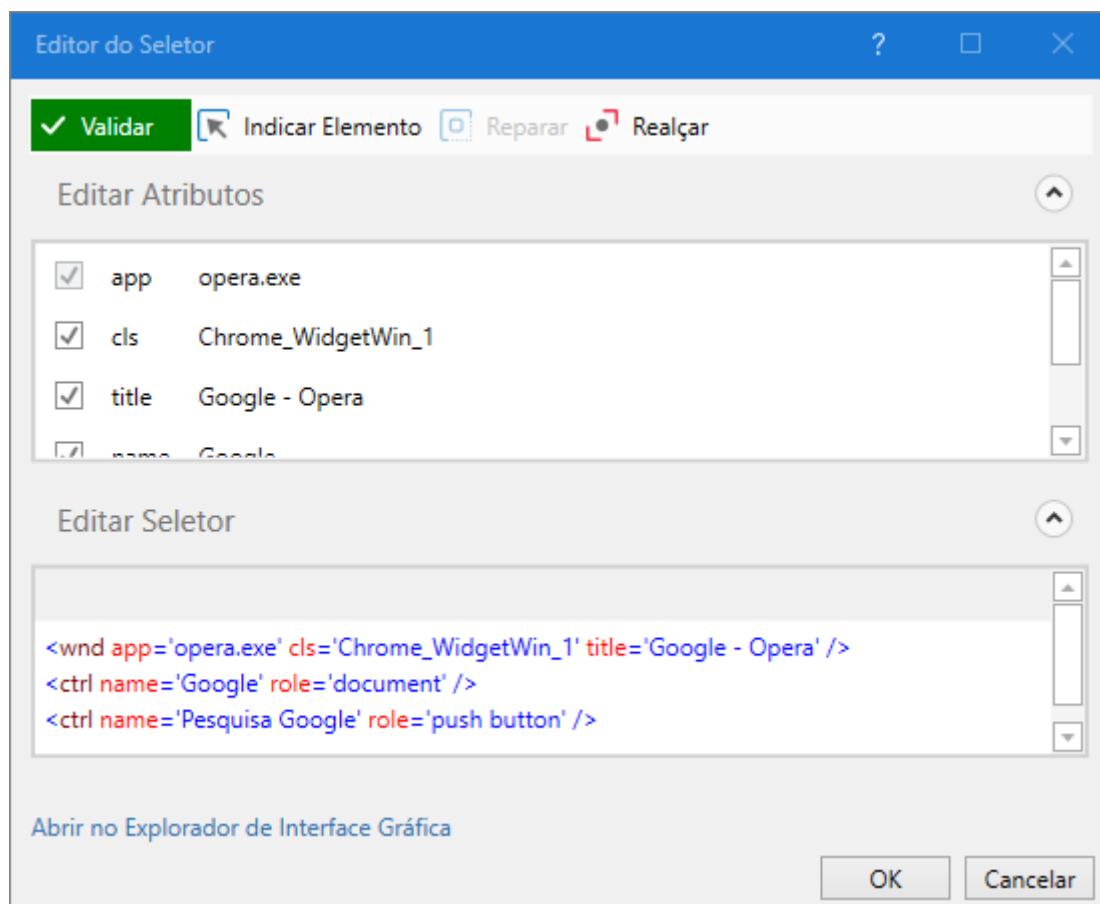
2.2.1.2 Selectors/Seletores

Um seletor é como um endereço textual para o *UiPath* reconhecer elementos da interface do usuário, como botões e *strings* em seu aplicativo.

Ao especificar exclusivamente o seletor e especificar o elemento de interface do usuário a ser manipulado, operações como clicar e entrada de texto podem ser estabelecidas e feitas para funcionar.

“Na maioria das vezes, os seletores são gerados automaticamente pelo *Studio* e não exigem mais informações do desenvolvedor, especialmente se os aplicativos que estiverem sendo automatizados tiver uma interface de usuário estática.” (*UiPath*, 2022, tradução livre). Seu layout pode ser observado na figura 22.

Figura 22 - Exemplo de seletor feito com a atividade Click para clicar no botão “Pesquisa Google” no site google.com



Fonte: Do autor (2022).

2.2.2 UiPath Robot

Como ressaltado na citação feita por TRIPATHI (2018, p. 25, tradução livre).

UiPath Robot é um serviço do *Windows* que pode abrir sessões de janela interativas/não interativas para executar processos ou um conjunto de etapas, desenvolvidas ou gravadas usando o *UiPath Studio*. Às vezes, também é chamado de agente de execução, pois executa projetos de automação, ou agente de tempo de

execução, pois executa instruções geradas por processos de desenvolvimento ou gravação no *UiPath Studio*.

“O *UiPath Robot* é o *host* de execução que executa processos projetados no *UiPath Studio*. No *Orchestrator*, o objeto robô representa a imagem do *UiPath Robot* e ajuda a definir seus recursos e acesso aos recursos do *Orchestrator*.” (*UiPath*, 2022, tradução livre).

O *UiPath Robot* funciona tanto em ambientes assistidos (trabalhando apenas no gatilho humano) quanto em ambientes autônomos (auto-gatilho e trabalho por conta própria).

2.2.2.1 Tipos de Robôs

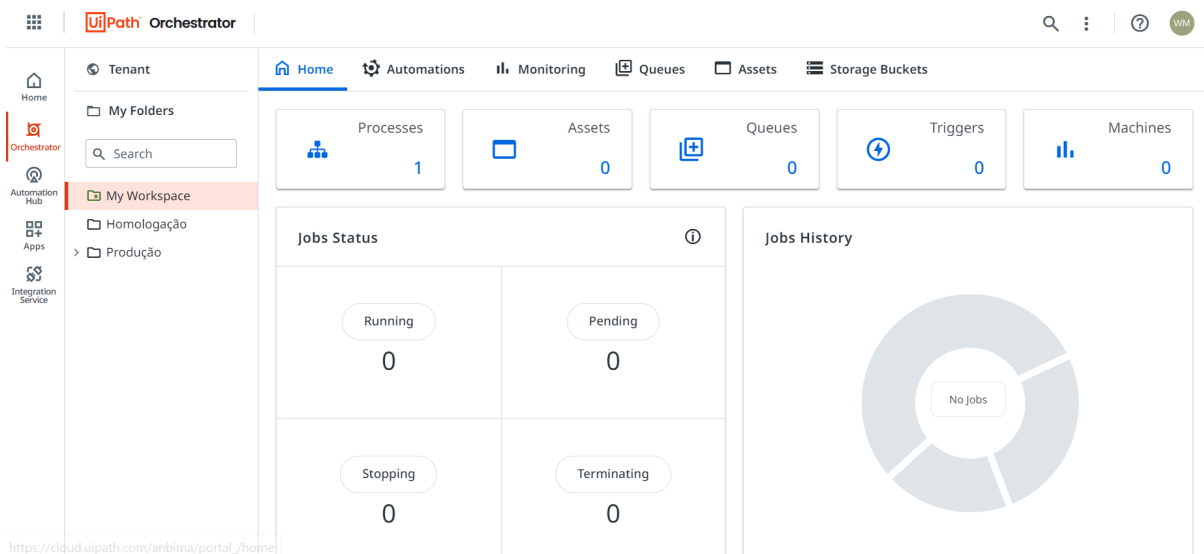
Os robôs desenvolvidos consistem em dois tipos:

- Assistidos: O robô assistido opera na mesma estação de trabalho que um humano para ajudar o usuário a realizar tarefas diárias. Geralmente é acionado por eventos do usuário. Não é possível iniciar um processo a partir do *Orchestrator* nesses tipos de robôs e eles não podem ser executados em uma tela bloqueada;
- Não assistidos: O robô não assistido pode ser executado sem supervisão em ambientes virtuais e pode automatizar qualquer número de processos. Além dos recursos do robô assistido, este é responsável pela execução remota, monitoramento, agendamento e suporte para filas de trabalho.

2.2.3 *UiPath Orchestrator*

O *UiPath Orchestrator* “ é uma plataforma baseada na *Web* que executa e gerencia robôs. Ele é capaz de implantar vários robôs e monitorar e inspecionar suas atividades.” (TRIPATHI, 2018, p. 26, tradução livre). Na figura 23 mostra-se a tela principal. É possível observar que o *Orchestrator* pode executar um robô, colocando-o em 4 estados de execução: rodando, em espera, parado e terminando.

Figura 23 - Layout Orchestrator



Fonte: Do autor (2022).

2.2.5 Licenciamento

O *UiPath* é composto por três tipos de licença, sendo elas, gratuita, Pro Starting e Enterprise.

Tabela 1 - Licenças *UiPath*

<i>Free</i>	<i>Pro Starting</i>	<i>Enterprise</i>
Acesso ao uso pessoal de recursos de desenvolvimento e assistidos gratuitamente.	A partir de US \$420/mês.	Este plano oferece soluções flexíveis para as empresas, sejam soluções na nuvem <i>UiPath</i> ou no local, com pacotes de suporte. Necessário contatar a equipe de vendas.
Execuções ilimitadas de automação individual.	Robôs para execução remota sob demanda.	Execute testes automatizados e fluxos de trabalho de RPA com robôs de teste;

Ferramentas para projetar automações individuais.	Ferramentas avançadas de design de automação com governança do usuário.	Ferramentas avançadas de design de automação com governança do usuário.
Integrações pré-construídas da categoria com produtos populares.	Produtos adicionais para envolver pessoas e robôs para uma colaboração de processo perfeita.	100 licenças <i>Automation Express</i> gratuitas.
X	<i>UiPath Standard Support.</i>	Produtos adicionais — descoberta de processos, gerenciamento de testes, IA, análise para agregar valor.
X	X	Implantação multirregional flexível
X	X	Gerenciamento avançado de autenticação.

Fonte: *UiPath* (2022).

2.3. Linguagem de Programação Python

De acordo com o site <http://www.python.org/>:

A linguagem Python foi criada em 1991, por Guido van Rossum, com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional, priorizando a legibilidade do código com uma sintaxe elegante, concisa e clara. É uma linguagem interpretada de alto nível, orientada a objetos, porém procedural e funcional, de tipagem dinâmica, tipagem forte e multiplataforma. (Python, 2009).

Segundo Melo, “*Python* é uma linguagem de programação de alto nível, ou seja, com sintaxe mais simplificada e próxima da linguagem humana, utilizada nas mais diversas aplicações, como *desktop*, *web*, servidores e ciência de dados.” (MELO, 2020).

A linguagem fornece recursos como tipagem dinâmica, ou seja, a variável assume o tipo do conteúdo inserido, programação multiparadigma (programação funcional e imperativa) e recursos poderosos em biblioteca padrão e por meio de módulos e frameworks desenvolvidos pela comunidade. Esta linguagem está disponível gratuitamente e pode ser executada em praticamente qualquer sistema operacional.

Para o desenvolvimento da solução em *Python*, foi utilizado o *Jupyter Notebook*, que é uma aplicação *web* de código aberto, o que significa que seu código pode ser usado e alterado para várias finalidades. Por ser um software de código aberto, o *Jupyter* também é gratuito, pois não é necessária licença para usá-lo. Isso permite que o aplicativo seja amplamente distribuído e aprimorado por meio dos esforços da comunidade.

De acordo com Code (2019),

O *Jupyter Notebook* é um interface gráfica que permite a edição de *notebooks* em um navegador web, tais como Google Chrome ou Firefox. O nome Jupyter é um acrônimo criado a partir das linguagens de programação que inicialmente foram aceitas pelo Projeto Jupyter: Julia, Python e R. Além dessas, hoje, o Projeto Jupyter suporta também C++, Ruby, Fortran e outras.

Também sabe-se que o Jupyter Notebook “é o aplicativo web original para criar e compartilhar documentos computacionais. Ele oferece uma experiência simples, simplificada e centrada em documentos.” (JUPYTER, 2022, tradução livre).

2.4. Microsoft *Power Platform*

O *software Microsoft Power Platform* “é um conjunto de aplicativos de *software* para *business intelligence*, desenvolvimento de aplicativos e conectividade de aplicativos. A *Microsoft* criou a linguagem de programação *Power Fx* para expressar a lógica em toda a *Power Platform*.” (MICROSOFT, 2018, tradução livre).

2.4.1 *Power Apps*

A plataforma do *Power Apps* “faz parte do *Microsoft Power Platform* maior que também inclui o *Power BI* e o *Power Automate*, utilizando a infraestrutura comum do *Microsoft Dataverse* e Conectores de Dados.” (MICROSOFT, 2022).

O *Power Apps* “é um conjunto de aplicativos, serviços e conectores, bem como uma plataforma de dados que oferece um ambiente de desenvolvimento rápido de aplicativos para criação dos apps personalizados para suas necessidades de negócios.” (MICROSOFT, 2022).

Com o uso do *Power Apps* é possível “criar soluções rapidamente com ferramentas com pouco código acessíveis a todos. Resolver problemas de negócios com aplicativos do *Power Apps* e automação de fluxo de trabalho, IA, acesso seguro a dados e recursos perfeitos de análise de dados e visualização.” (MICROSOFT, 2022).

2.4.2 Sharepoint

O *Microsoft SharePoint* “é uma plataforma de aplicativos da *Web* da *Microsoft* usada para a criação de portais e intranets corporativos, gerenciamento de conteúdo, gerenciamento de documentos e criação de portais colaborativos, bem como a publicação de aplicativos da *Web*.” (MICROSOFT, 2022).

A ferramenta *Microsoft SharePoint* “pode satisfazer as necessidades básicas de colaboração e comunicação das organizações, logo após a implementação de uma solução, muitas organizações podem optar por ir além da funcionalidade principal.” (GOLUBENKO, 2019).

2.4.3 Licenças Power Platform

Com informado dentro das hospedagens da *Microsoft* “As organizações podem obter licenças licenciando o *Microsoft Power Apps* ou *Microsoft Power Automate* especificamente ou incluindo-o na licença de outra oferta do *Microsoft Cloud Service*. Por exemplo, *both Microsoft 365* e *Dynamics 365* fornecem qualificações para *Power Apps* e *Power Automate*.” (MICROSOFT, 2022), ainda, que a *Microsoft* informe que, como na maioria de suas licenças, é possível misturar e combinar usuários conforme necessário, atribuindo qualificações adicionais.

O *Power Apps* e o *Power Automate* agora também contam com planos de avaliação. As avaliações gratuitas duram 30 dias para os planos do *Power Apps* e 90 dias para os planos do *Power Automate*. Além dos planos de avaliação, há também um plano para desenvolvedores do *Power Apps* gratuito, neste plano os ambientes são para uso individual, não há capacidade de compartilhar com outros usuários.

2.5. Canal direto usando Azure

O canal direto caracteriza-se por um arquivo com extensão .csv extraído da plataforma *web Azure DevOps*. Este arquivo é de uso interno da gerência da área executante para controle do apontamento de horas dos demais integrantes da equipe. É constituído por informações particularmente inerentes do dia de trabalho dos usuários, tais como minutos, data, bem como o usuário que o preencheu, o *type* que é o tipo de alocação e *comment* são informações complementares para uma melhor análise.

Cada linha do documento corresponde a um apontamento feito por um usuário e contém todas as informações mencionadas acima que são relevantes para o evento, e toda vez que um usuário faz um novo lançamento esse documento aumenta uma linha com todos os dados necessários.

Detalhes mais aprofundados dos dados dos usuários não podem ser monitorados para fins de confidencialidade. Uma melhor compreensão da relação entre este e o robô será explorada em uma discussão do processo utilizado.

2.6. Sobre o desafio

De modo a seguir com extração dos dados web e realizar a verificação do apontamento de horas de cada usuário que trabalham de forma remota e visando otimizar o processo de automação do mesmo, cada etapa desse tipo de fluxo foi descrita segundo as diretrizes de uma divisão em partes, assim dizendo, cada estágio foi dividido em estágios menores para que o problema pudesse ser abordado como problemas menores com melhor viabilidade e visibilidade da resolução, em vez de um problema único.

3. DESENVOLVIMENTO PRÁTICO

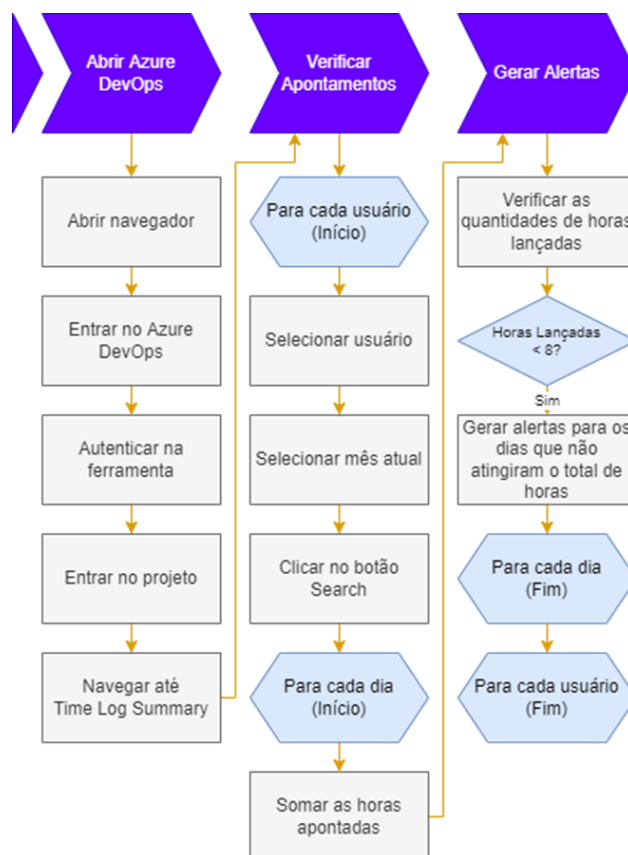
3.1. Planejamento

O objetivo desse processo consiste em executar a verificação do preenchimento do apontamento de horas de todos os usuários e gerar alertas quando houver apontamentos não realizados.

3.2. Desenho do processo

Nesta seção, é representado o processo AS-IS, que é a definição da situação atual do processo organizacional ou de negócios, na figura 24 é mostrado o fluxo geral do processo em nível macro e o fluxo do processo em nível mais detalhado mostrando os possíveis caminhos / exceções do processo atual.

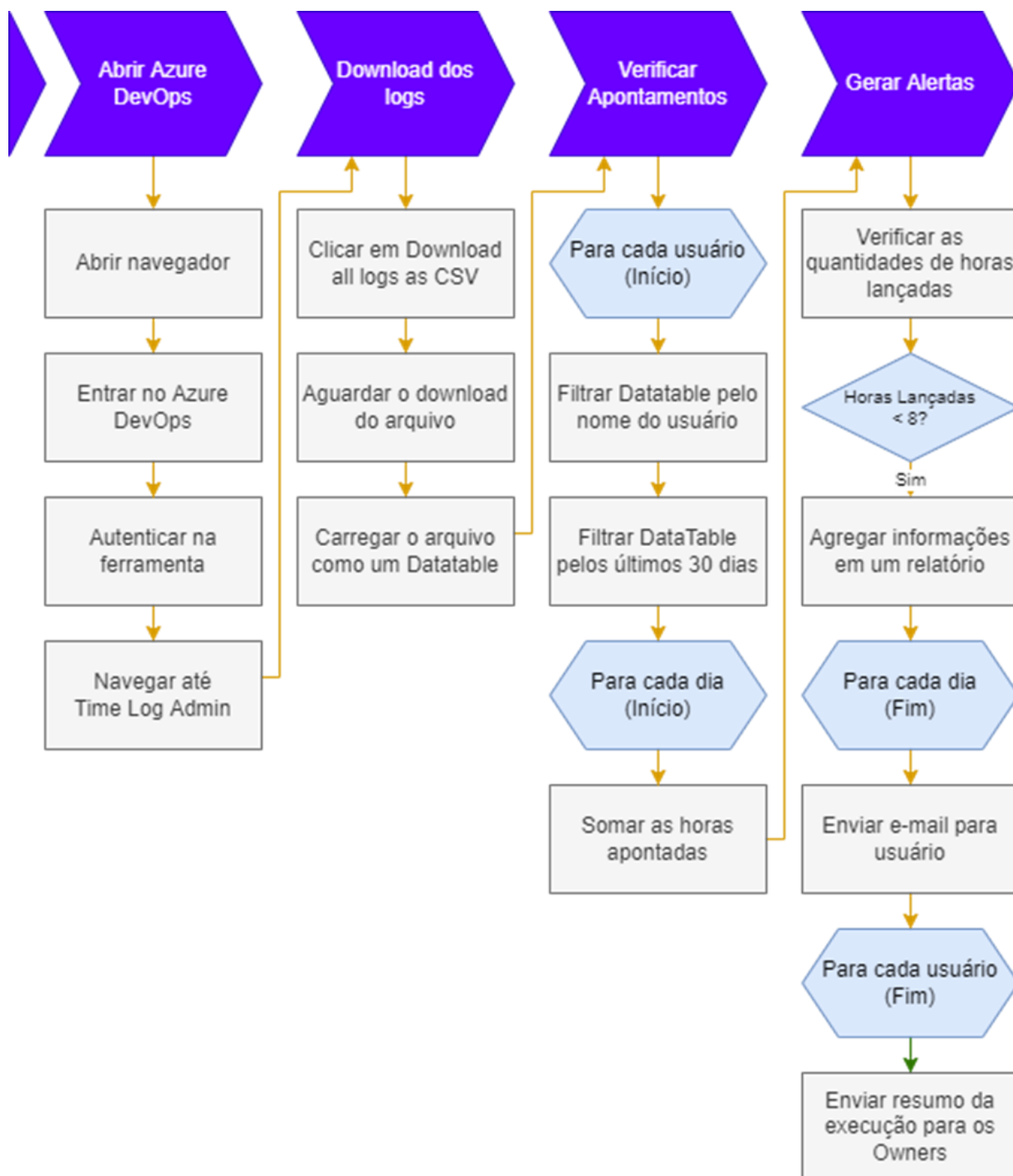
Figura 24 - Fluxograma geral do processo em nível macro



Fonte: Do autor (2022).

Nesta seção, na figura 25 é apresentado e detalhado o processo TO-BE, mostrando o passo a passo da automação que será feita.

Figura 25 - Fluxograma do processo em nível mais detalhado mostrando os possíveis caminhos / exceções do processo atual.



3.3. Desenvolvimento do processo

Essa etapa do projeto mostrará o desenvolvimento das duas soluções para automatizar o processo descrito anteriormente, tendo maior enfoque na solução desenvolvida a partir do *UiPath* em que foi desenvolvido o processo por completo. Já a solução utilizando *Python*, as etapas de “Abrir Azure DevOps”, “Download dos logs”, “Gerar alertas, a partir do bloco (primeiro bloco de preparação (Para cada dia (fim)))” não serão desenvolvidas.

O arquivo .csv baixado de dentro do portal do *Azure DevOps*, pode ser visto em sua estrutura na figura 26 abaixo.

Figura 26 - Estrutura do arquivo timelog.csv baixado do portal do *Azure DevOps*.

A1	:	✕	✓	<i>fx</i>	id,minutes,user,userId,workItemId,date,dateWeek,type,comment				
	A	B	C	D	E	F	G	H	I
1	id,minutes,user,userId,workItemId,date,dateWeek,type,comment								
2	0001ed2e-abab-4be7		6ce5c6b8f2d6,90,"Renan		",397929bc-ab2f-6b44-932c-2b1f7				
3	00024795-d024-4cc5-		-f5a05773209b,60,"Gabriel		",2705aad2-5dcf-6016-940f-82284				
4	0015e373-52e9-4027-		-4257617c739f,60,"Cesar		",ce344fdf-b083-63c4-9355-eac9f78				

Fonte: Do autor (2022).

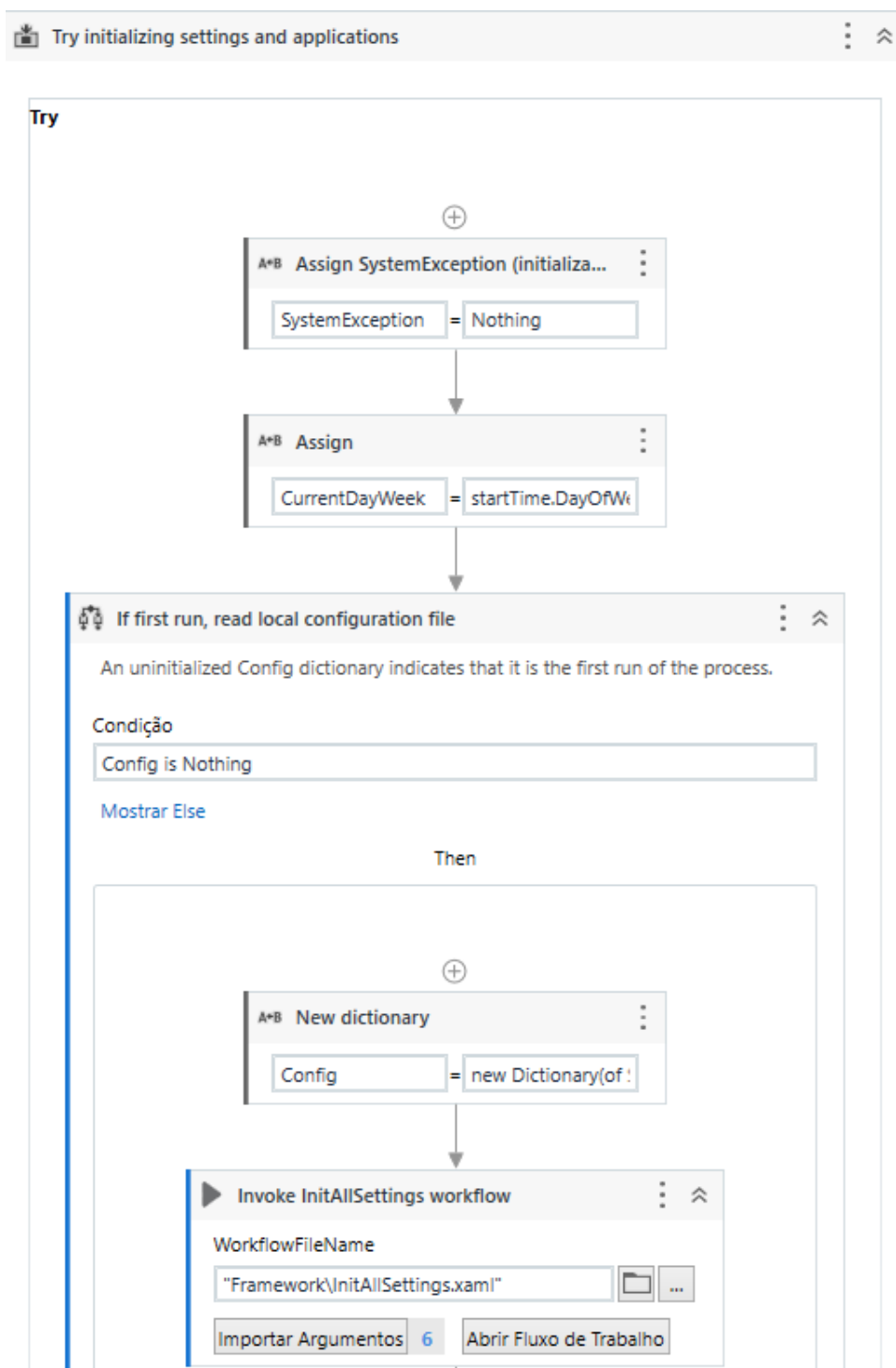
3.3.1 Processo desenvolvido utilizando *UiPath*

Como pode ser visto no capítulo 2.2.1, o processo foi desenvolvido em 4 blocos diferentes, sendo eles: Initialization, Get Transaction Data, Process Transaction e End Process.

O bloco Initialization, é o responsável por iniciar todas as variáveis que serão utilizadas para funcionamento do robô, além disso, também é nele que é realizado as etapas de “Abrir Azure DevOps” e “Download dos logs”.

Dentro do bloco Initialization (figura 27), podemos ver uma atividade “Try Catch”. Essa atividade foi inserida para verificar se ocorreu um erro na inicialização e seja lançada uma exceção, para que se possa entender qual foi o problema que ocorreu durante a execução. Também podemos ver algumas outras atividades que são responsáveis pela inicialização do robô.

Figura 27 - Estrutura dentro do bloco Initialization.

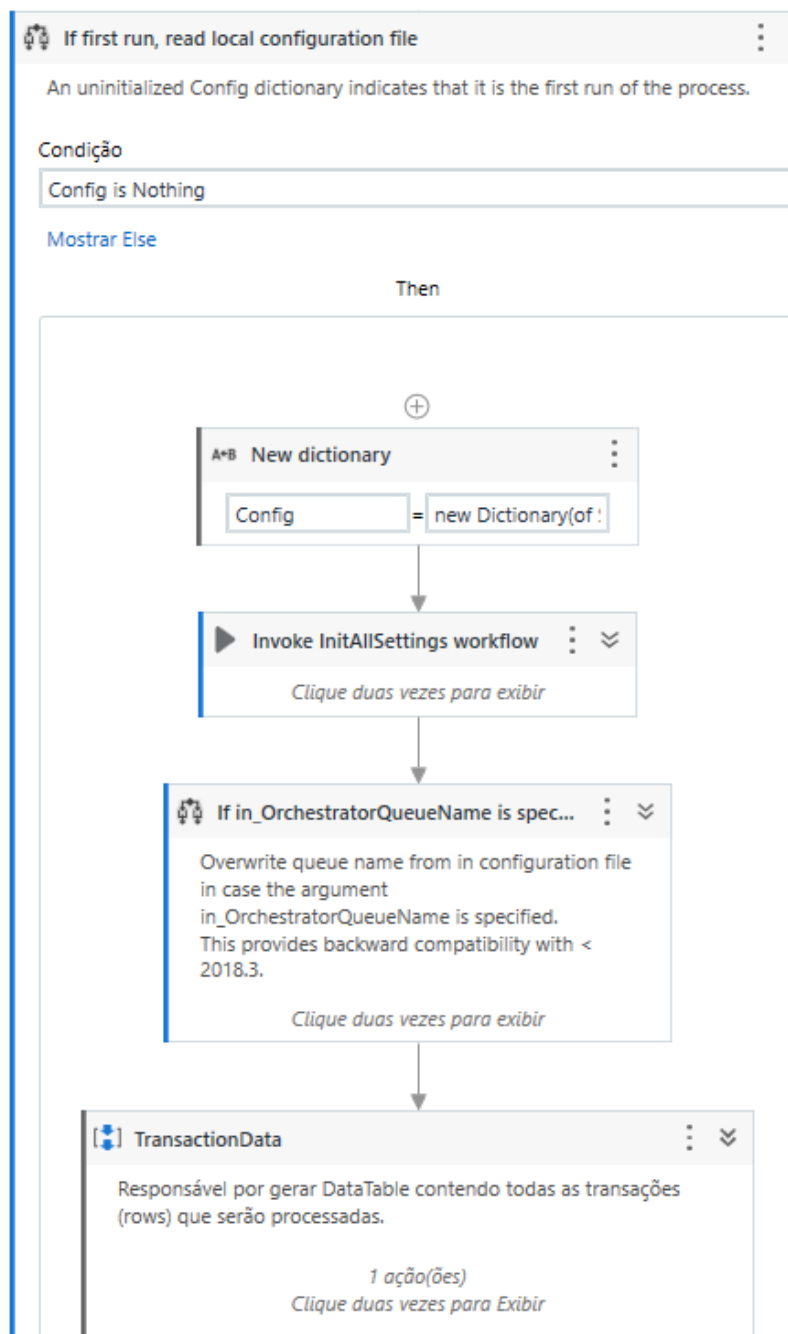


Fonte: Do autor (2022).

Dentro do “If” (figura 28), ocorre a leitura dos documentos que terão suas informações extraídas e atribuídas a variáveis para serem utilizadas no robô. Além disso, como pode ser visto na figura 29 e 30, também é feita a criação das transações que se baseiam na planilha de

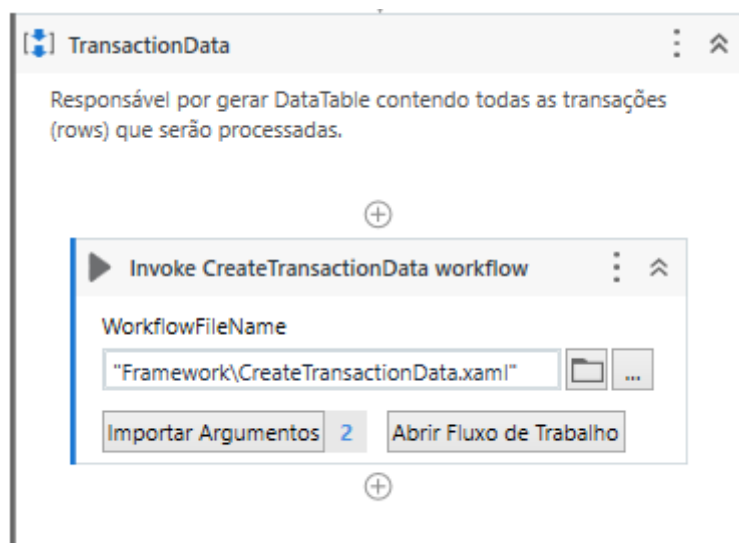
usuários a serem analisados, feita a preparação do arquivo de saída (*output*), checagem do ambiente e inicia a gravação do processo.

Figura 28- Estrutura inicial de dentro da atividade “If”



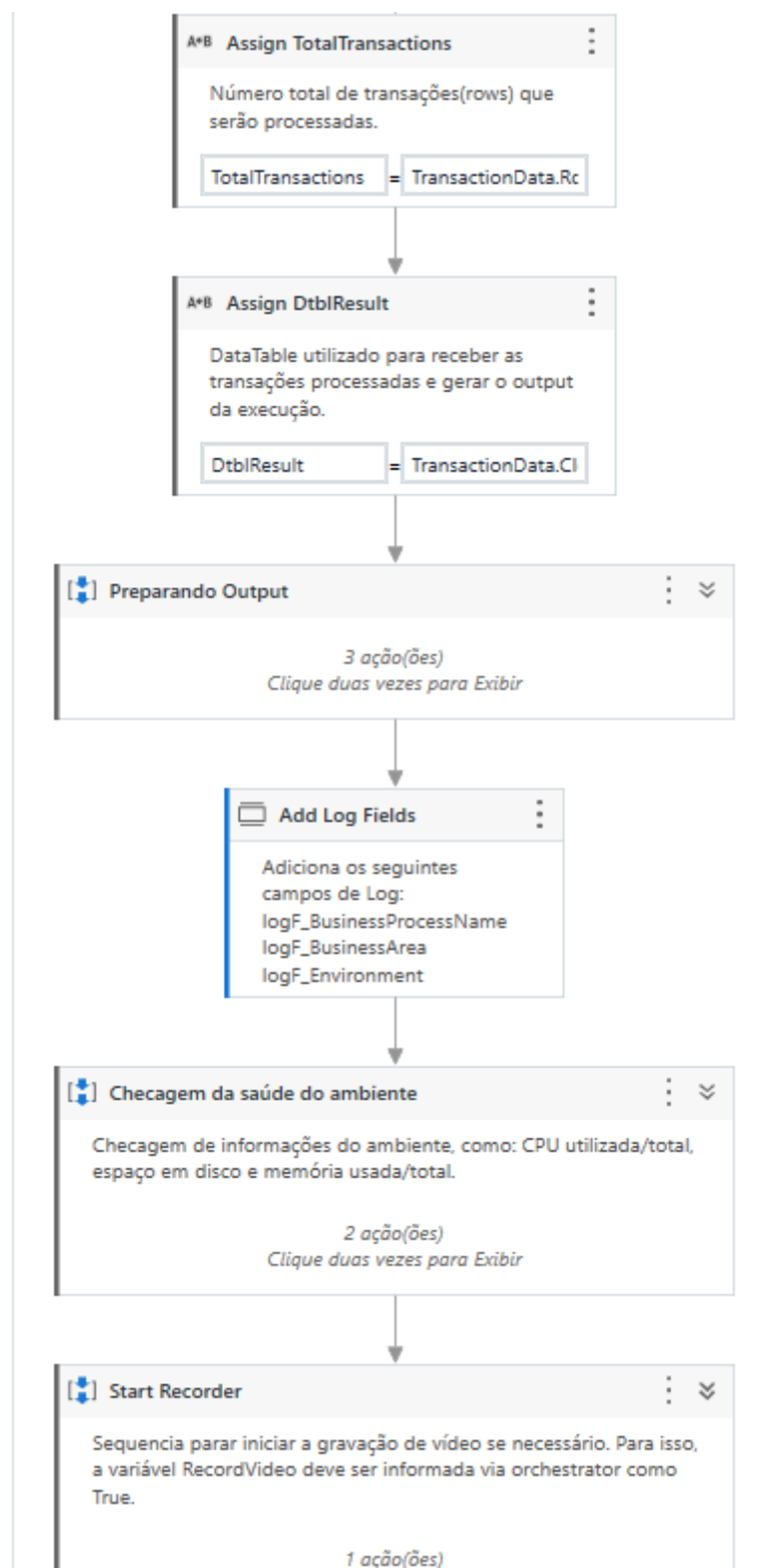
Fonte: Do autor (2022).

Figura 29 - Estrutura da criação da transição a partir da tabela de usuários



Fonte: Do autor (2022).

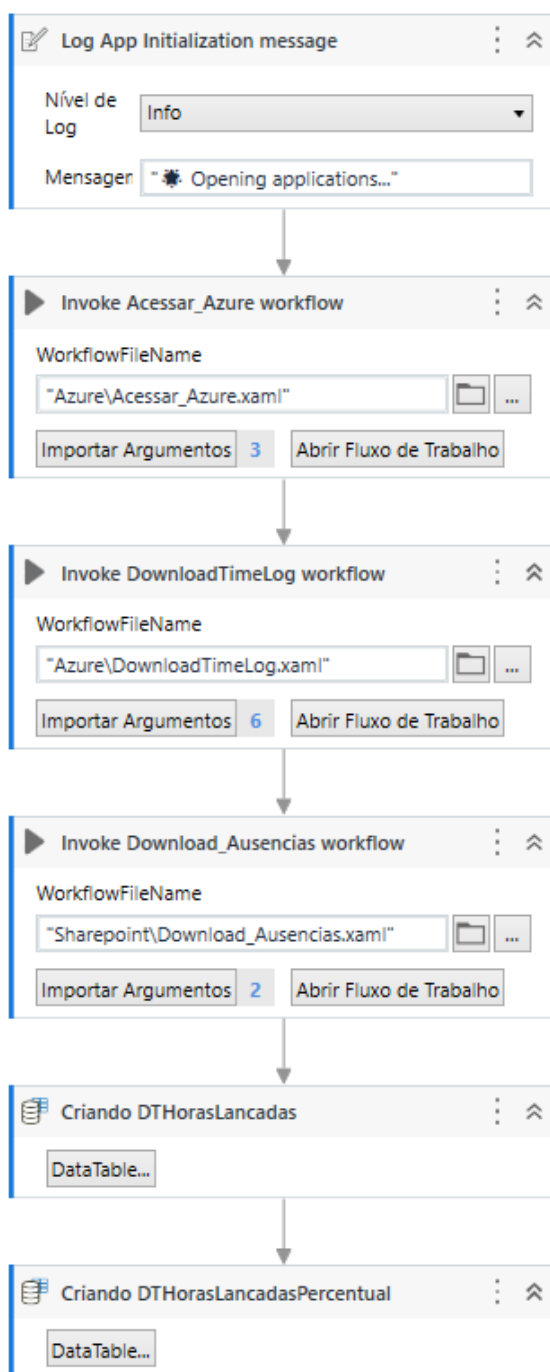
Figura 30 - Preparação do output, checagem de saúde do ambiente e inicializa a gravação do processo.



Fonte: Do autor (2022).

Passada por todas as etapas dentro do “If”, é feita a inicialização dos aplicativos que serão utilizados no robô, todos os passos dentro do InitAllApplications podem ser vistos na figura 31.

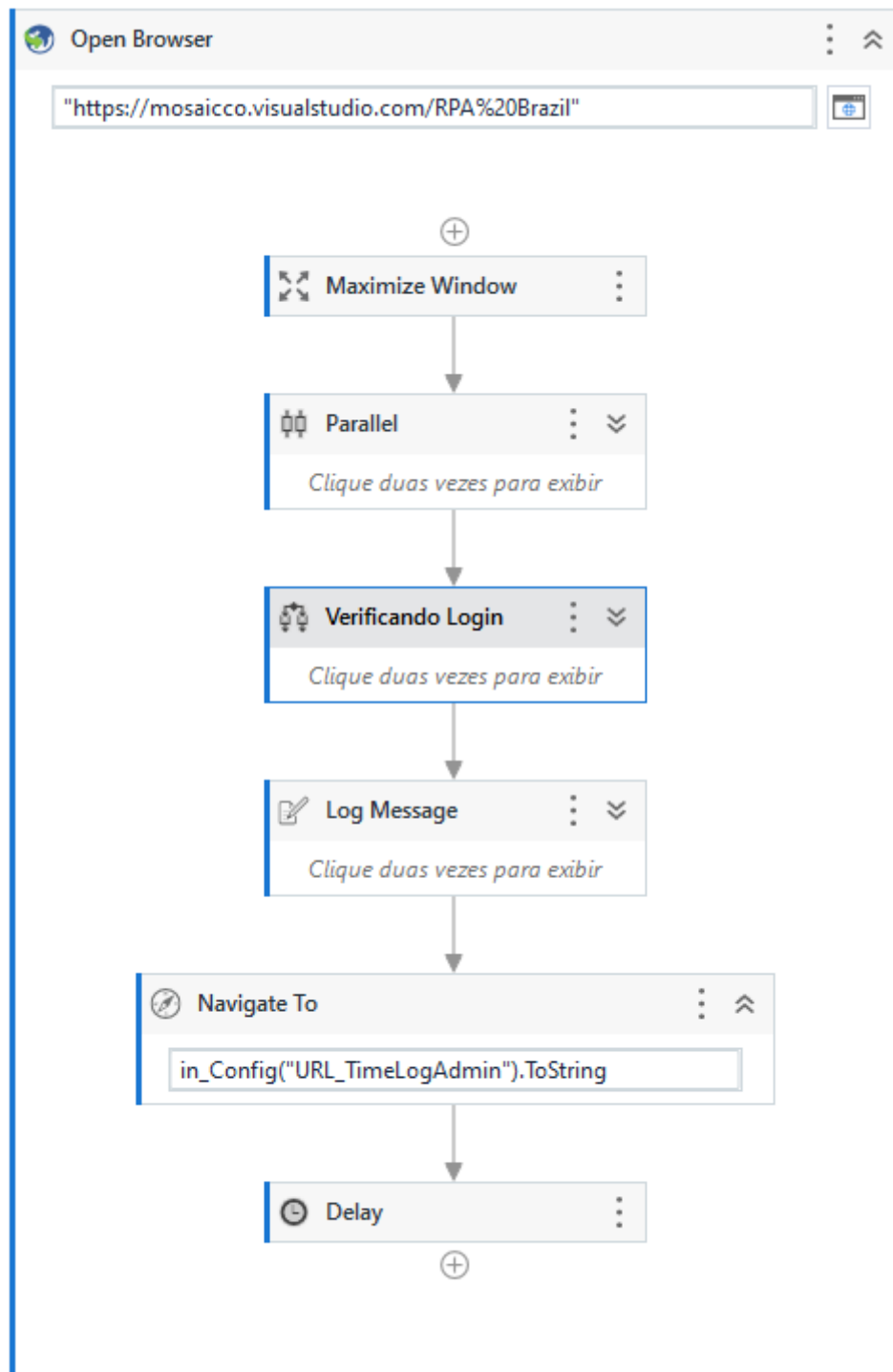
Figura 31 - InitAllApplication, etapa responsável por acessar o *Azure*, fazer o download da planilha de apontamento e planilha de ausências.



Fonte: Do autor (2022).

O “*Invoke Acessa_Azure workflow*” pode ser visto na figura 32, essa é a etapa responsável por abrir o navegador, acessar o servidor do *Azure*, verificar se o login já foi realizado e caso o mesmo ainda não tenha sido feito realizar o login.

Figura 32 - Passos para acessar o *Azure DevOps* realizado dentro do *Invoke Acessa_Azure workflow*

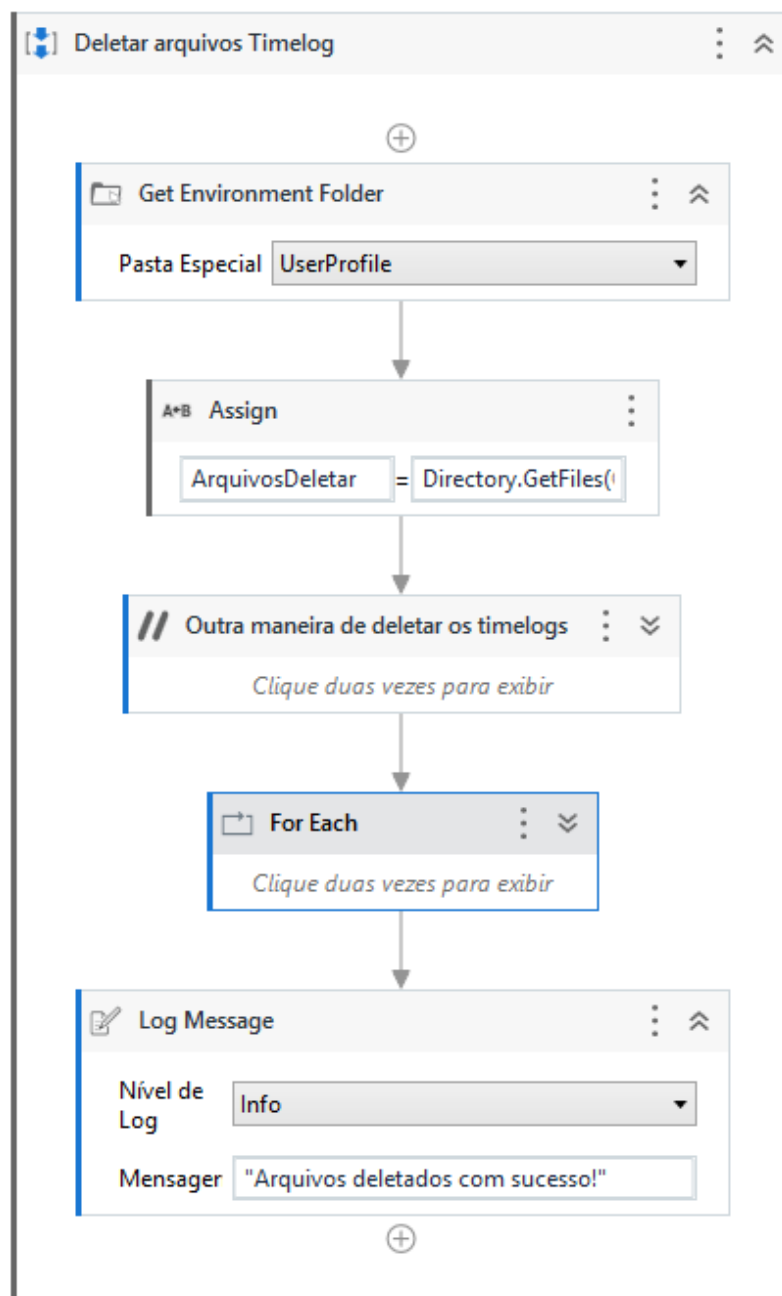


Fonte: Do autor (2022).

O “*Invoke DownloadTimelog workflow*”, é responsável por realizar duas etapas:

- Como pode ser visto na figura 33, a primeira parte é deletar todos arquivos “timelog*.csv” de dentro da pasta principal onde é realizado os downloads de arquivos da máquina em que está sendo executado o robô.

Figura 33 - Sequence onde é realizada a etapa responsável por deletar todos os arquivos de “timelog.csv” existente dentro da pasta de downloads

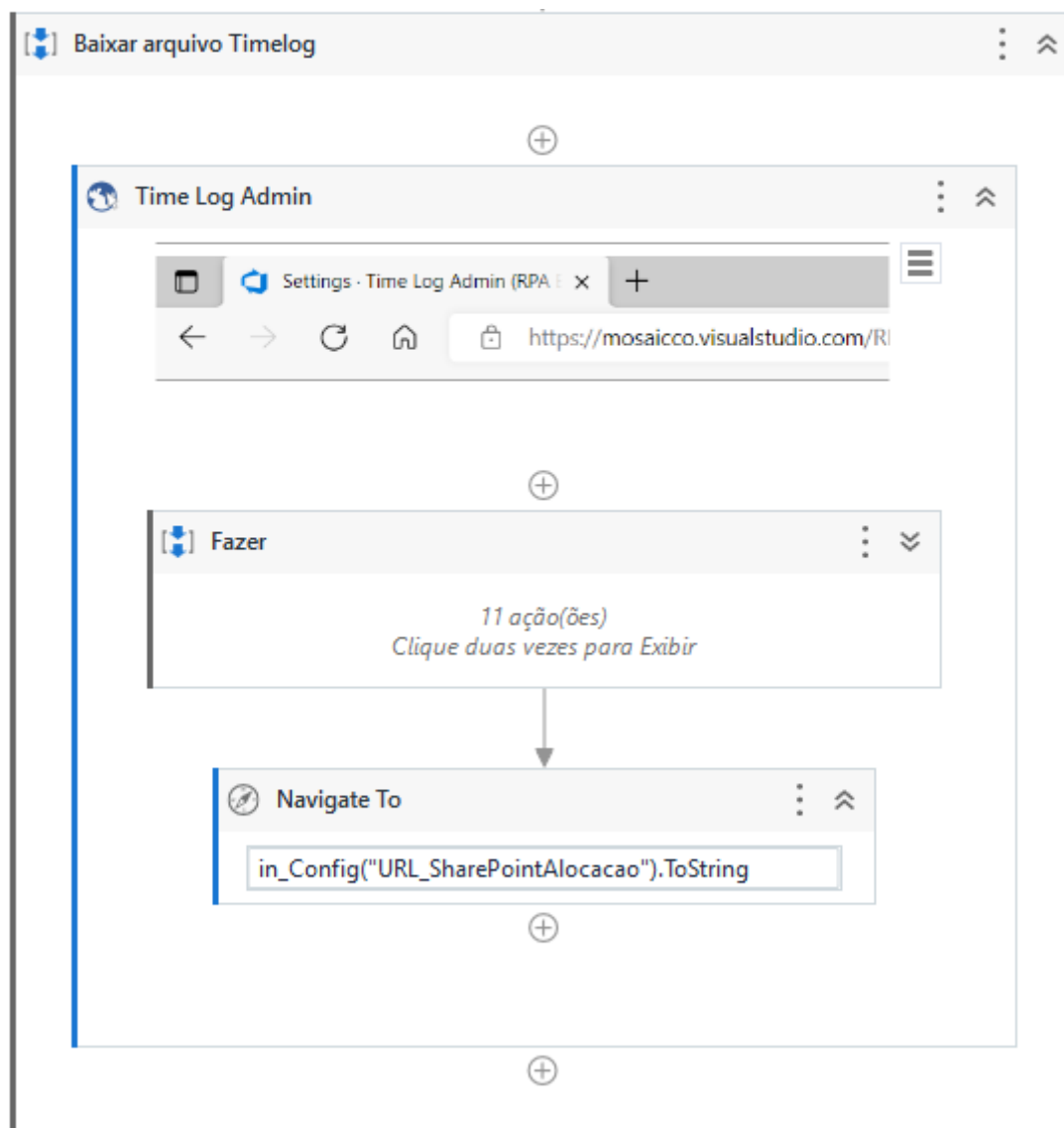


Fonte: Do autor (2022).

- Como pode ser visto na figura 34, a segunda parte é responsável por realizar o download do arquivo “timelog.csv”, que é o arquivo onde encontra-se o apontamento de horas

dos usuários, e posteriormente fazer o redirecionamento para o *Sharepoint*, onde será buscada as alocações de ausências dos usuários.

Figura 34 - Sequence onde realizada a etapa de baixar o arquivo “timelog.csv” mais atual

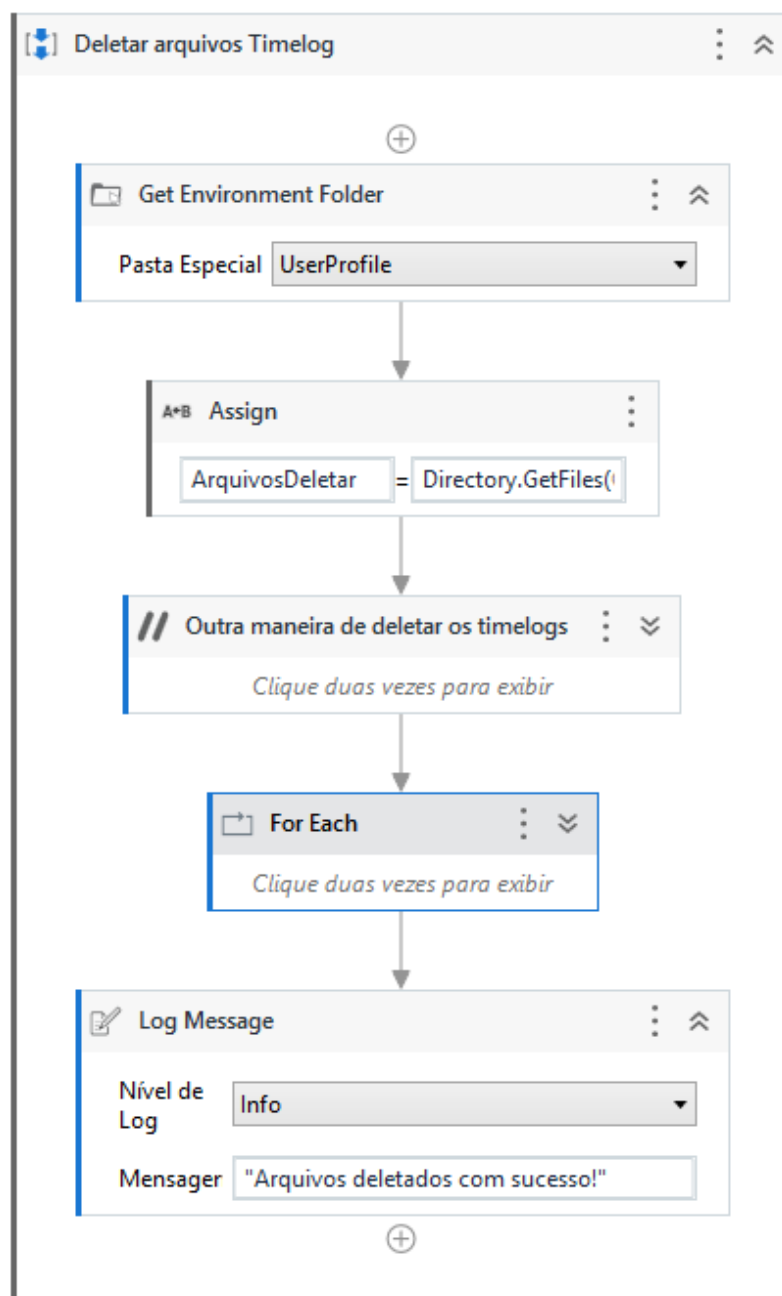


Fonte: Do autor (2022).

O “*Invoke Download_Ausencias workflow*”, é responsável por duas etapas, sendo elas:

- Como pode ser visto na figura 35, a primeira parte é deletar todos arquivos “Alocacao*.csv” de dentro da pasta principal onde é realizado os downloads de arquivos da máquina em que está sendo executado o robô.

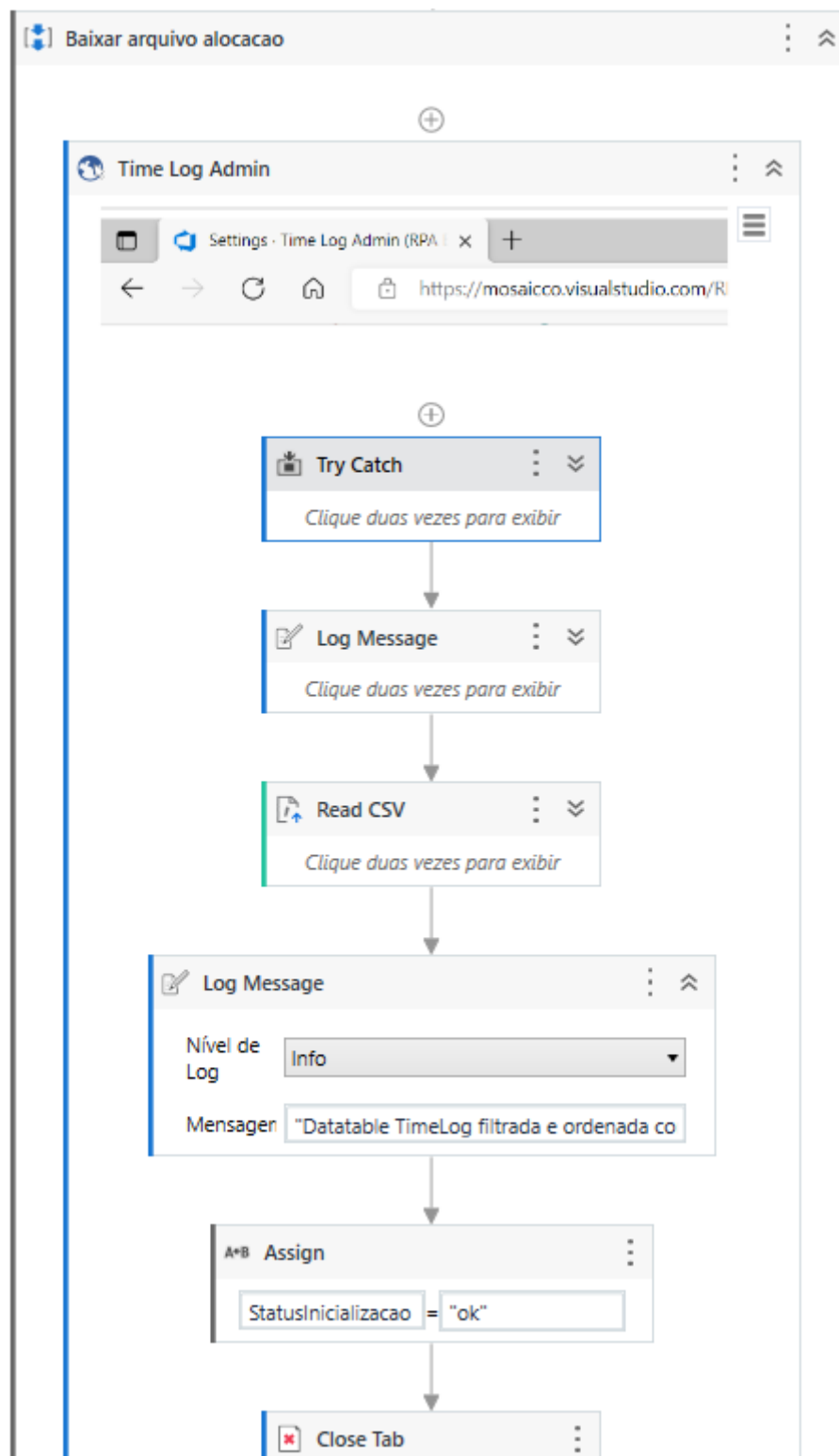
Figura 35 - Sequence onde realizada a etapa responsável por deletar todos os arquivos de “Alocacao*.csv” existente dentro da pasta de downloads



Fonte: Do autor (2022).

- Como pode ser visto na figura 36, a segunda parte é responsável por realizar o download do arquivo “Alocacao*.csv” e fazer a leitura do mesmo.

Figura 36 - Sequence onde realizada a etapa de baixar e ler o arquivo “Alocacao*.csv”



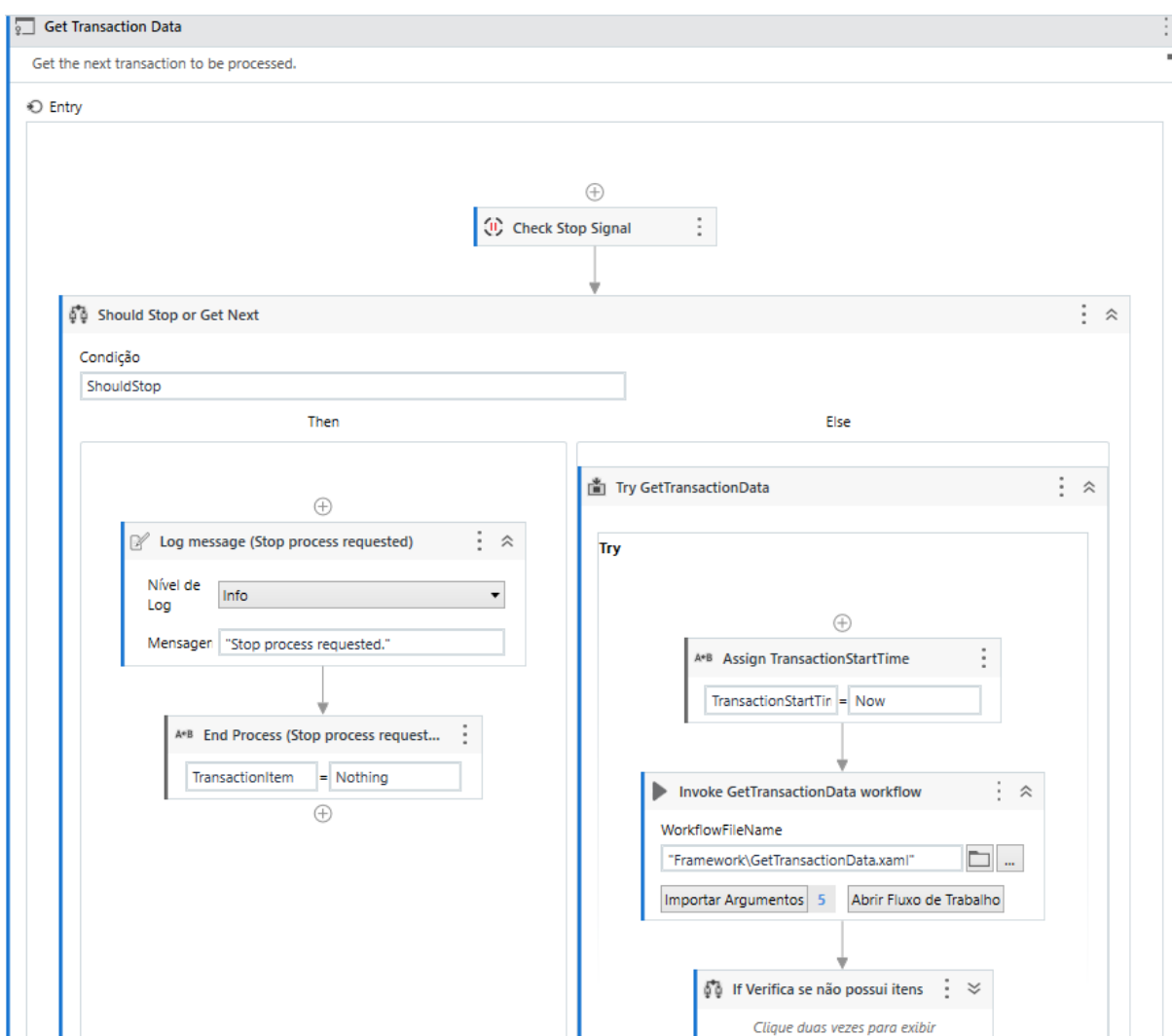
Fonte: Do autor (2022).

Após as etapas anteriores gera-se um *DataTable* contendo todas as transações que serão realizadas na etapa “Process”. Este momento é marcado pela leitura de uma planilha de

entrada chamada “DtblUsuarios”, e é nela que é encontrado todos os usuários que serão utilizados como transações para gerar os relatórios individuais e total.

Posteriormente ao bloco de inicialização, é passado pelo bloco “Get Transaction Data”; neste é selecionado o item que será feito o processamento. O bloco pode ser visto na figura 37.

Figura 37 - Etapas que ocorrem dentro do “Get Transaction Data”



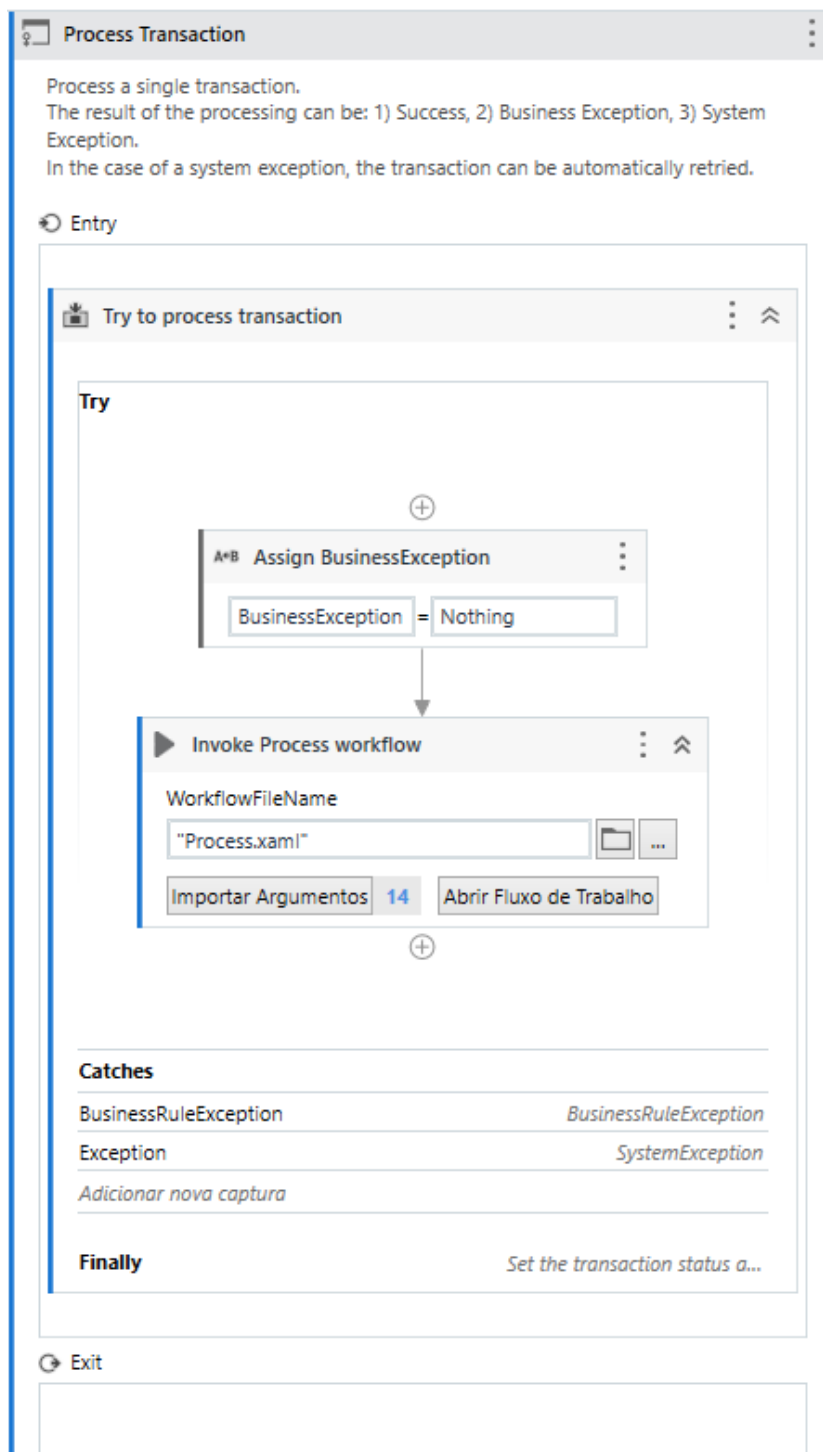
Fonte: Do autor (2022).

Neste bloco ele começa capturando se a função “Parar” foi acionada no *UiPath Orchestrator*, caso seja verdadeiro ele passará pelo lado do Then na atividade If, fazendo com que o robô pare de executar, do contrário passará pelo Else que irá executar a próxima transação.

Logo após o item a ser transacionado ser separado ele é passado para o bloco nomeado de “*Process Transaction*”, neste é onde ocorre toda a parte de tratativa do usuário individual.

Como vemos na figura 38, abrindo o bloco “*Process Transaction*” podemos ver que ele é composto por uma atividade *Try Catch*, que é utilizada para caso ocorra alguma inconsistência enquanto as informações dentro do bloco e devolva algum tipo de exceção. Olhando ainda dentro do bloco, nota-se que uma atividade de *invoke* é chamada, sendo ela a “*Invoke Process workflow*”, e dentro desta é onde o item passará pelas manipulações necessárias para averiguação do apontamento de horas correto até o presente dia do mês.

Figura 38 - Etapas que ocorrem internamente no bloco “*Process Transaction*”

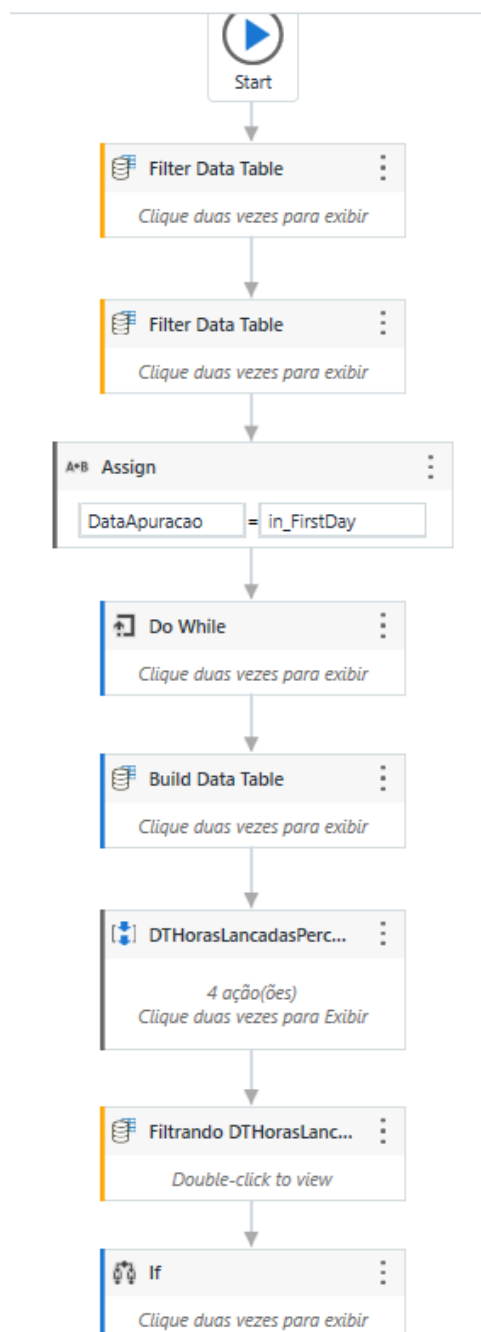


Fonte: Do autor (2022).

Cada etapa realizada dentro do *Invoke Process Workflow* é feita para que o processo seja executado da melhor maneira possível respeitando as etapas descritas nos dois últimos

blocos do fluxograma mostrado na Figura 25. Todas as etapas que ocorrem dentro do *Invoke Process Workflow* podem ser observadas na figura 39.

Figura 39 - Etapas que ocorrem internamente no “*Invoke Process workflow*”



Fonte: Do autor (2022).

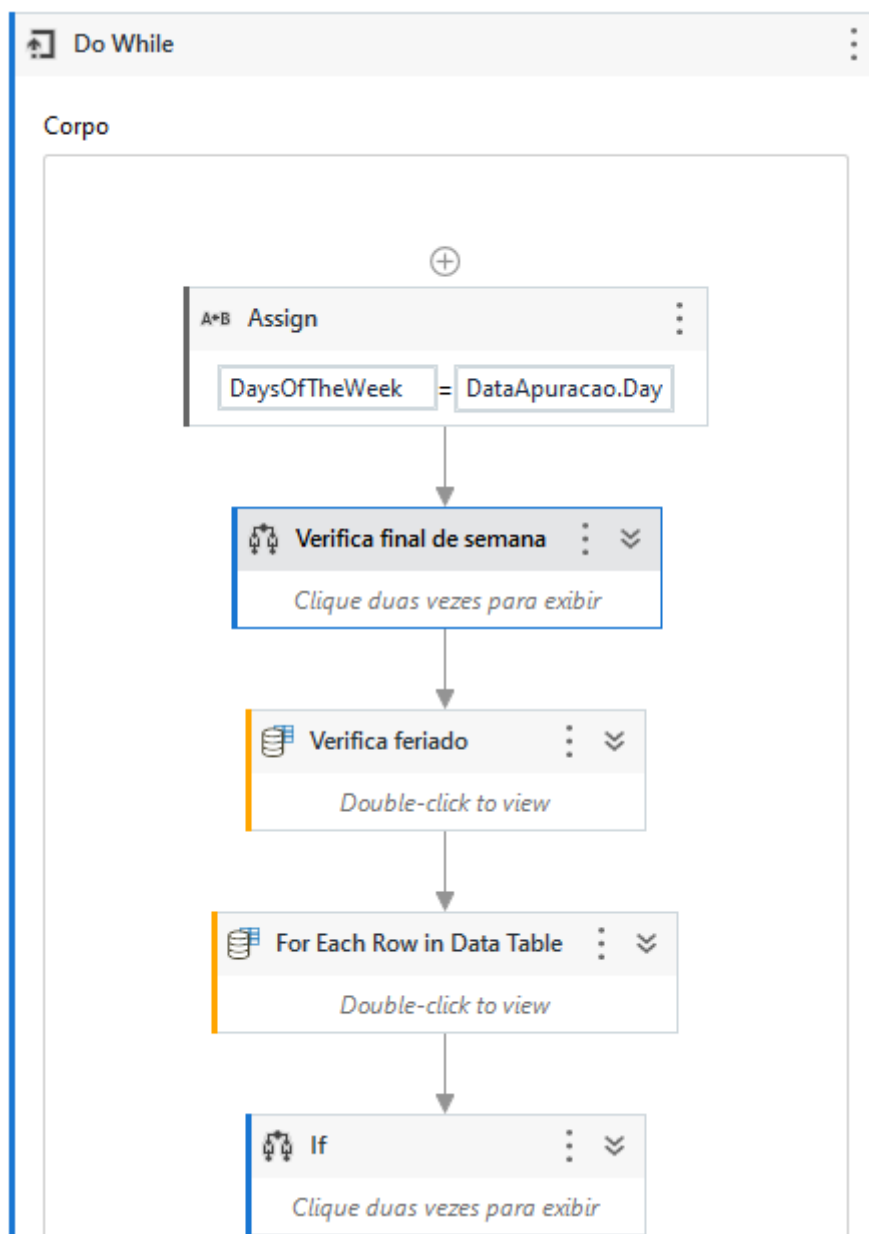
As duas atividades iniciais são responsáveis por filtrar as *DataTables* montadas a partir dos arquivos “timelog.csv” e “Alocacao.csv”, mantendo apenas linhas referente ao usuário que está sendo transacionado no momento. Os arquivos utilizados para viabilização desse

projeto têm aproximadamente 9.500 linhas e 9 colunas no primeiro arquivo, já o segundo é composto por aproximadamente 20 linhas e 8 colunas.

A terceira atividade é responsável por atribuir o primeiro dia do mês atual a variável “DataApuracao” que será manipulada na atividade seguinte “*Do While*”, a quarta atividade só irá rodar enquanto respeitar a condição “DataApuracao <= in_CurrentDay”, isto significa que ficará dentro desta atividade enquanto a data que estiver sendo verificada for menor ou igual a data atual que o robô está rodando.

De acordo com a figura 40, as atividades dentro do “*Do While*” ocorrem da seguinte maneira, primeiramente a atividade inicialmente verifica que dia da semana é o dia em questão que está sendo analisado, posteriormente ele verifica se este dia é um sábado ou domingo, logo após verifica se este é feriado e por fim se neste dia em questão o usuário está ausente, para então chegar no “*If*”.

Figura 40 - Processo que ocorre dentro da atividade “Do While”



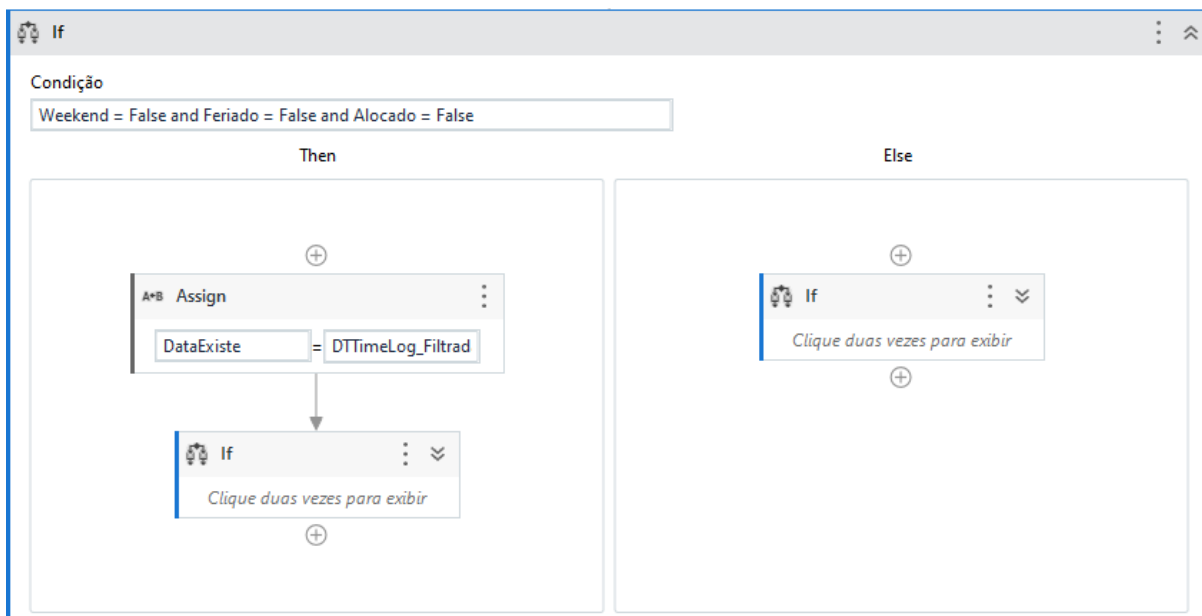
Fonte: Do autor (2022).

Conforme detalhado na figura 41, o “If” em questão é tratado pela seguinte condição:

- “Weekend = False and Feriado = False and Alocado = False”
 - Se a condição for verdadeira, passará para um próximo “If”, onde a condição é “DataExiste.Count > 0”, ou seja, o usuário fez algum apontamento neste dia;

- Se esta condição for verdadeira, ele somará o total de minutos apontados neste dia e irá inserir as informações em uma nova linha na *DataTable* DTHorasLancadas;
- Se esta condição for falsa, ele adiciona uma nova linha na *DataTable* DTHorasLancadas informando que no dia em questão não houve apontamento algum por parte do usuário;
- Se a condição for falsa, passará para um próximo “If”, onde este a condição é `Alocado = True`, ou seja, se o usuário neste dia tiver alguma alocação;
 - Se esta condição for verdadeira, passará para um próximo “If” onde este a condição é a alocação em horas é menor do que a quantidade que as horas diárias de trabalho do usuário;
 - Se verdadeira somará o total de minutos apontados neste dia e irá inserir as informações em uma nova linha na *DataTable* DTHorasLancadas;
 - Se esta condição for falsa, ele passará para próxima etapa, pois não é necessário cobrar o apontamento de horas do usuário neste dia.

Figura 41- Processo que ocorre dentro da atividade “If” dentro do “Do While”



Fonte: Do autor (2022).

Após o processo que ocorre dentro do “If”, é adicionado somado um dia na variável “DataApuracao” para continuar ou finalizar a etapa do “Do While”.

Passando para as atividades seguintes dentro do *Process*, é feita a criação de uma tabela de percentual e sequencialmente é feito o incremento de uma linha na mesma do percentual de horas alocadas no mês pelo usuário até a data atual, seguindo para a terceira atividade é feito um filtro na tabela DTHorasLancadas criada dentro “Do While”, para verificar se teve algum dia do mês que o usuário não fez o apontamento correto e tendo como output uma outra *DataTable* denominada DTHorasLancadasUsuario.

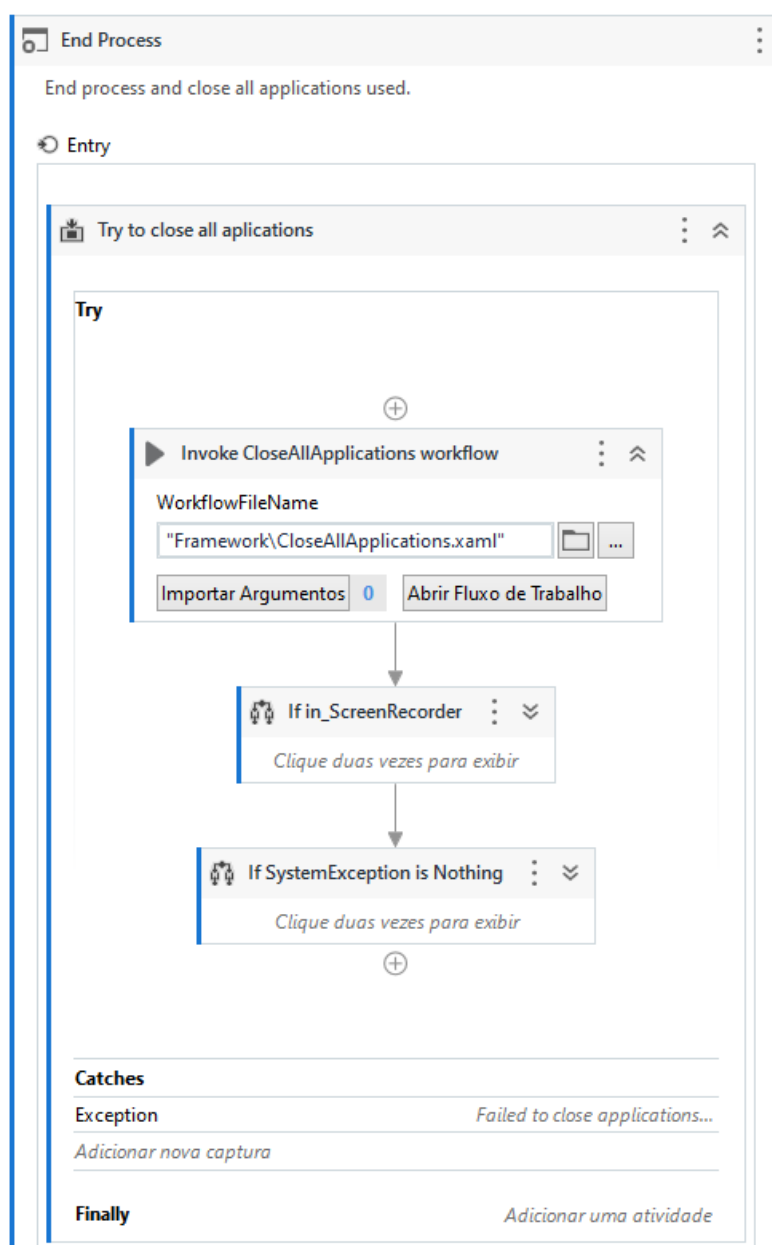
A última atividade *Process* é novamente um “If”, dentro dele ocorrerá o envio de e-mail para o usuário e as condições para ocorrer são as seguintes:

- in_CurrentDayWeek = "Wednesday", ou seja, o dia em que o robô está rodando é quarta-feira. Esta verificação é feita somente para quando for quarta-feira o e-mail ser enviado somente para os usuários, do contrário o gestor também receberá;
 - Se a condição for verdadeira, passará para um próximo “If”, onde este a condição é “DTHorasLancadasUsuario.RowCount > 1”;
 - Se a condição for verdadeira, irá enviar um e-mail para o usuário com os dias que o mesmo não apontou corretamente.

- Se a condição for falsa, passará para um próximo “*If*”, onde este a condição é “DTHorasLancadasUsuario.RowCount > 1”;
- Se a condição for verdadeira, irá enviar um e-mail para o usuário e para os gestores com os dias em que o usuário não apontou corretamente.

Após todas as etapas do *Process* serem realizadas o mesmo voltará para o bloco do “Get Transaction Data”, caso haja mais alguma transação a ser feita o mesmo fará do contrário irá para o bloco “*End Process*”. O bloco “*End Process*” pode ser visto na figura 42.

Figura 42 - Bloco “*End Process*”



Fonte: Do autor (2022).

Este bloco é responsável por finalizar o robô, nele é fechado todos os aplicativos que são utilizados neste processo, é feito o encerramento da gravação de tela e enviado o e-mail final para os gestores com os usuários, período contemplado (horas), Porcentagem de alocação (%), horas pendentes e o e-mail do usuário. Na figura 43 abaixo, é visto um exemplo de e-mail enviado para os gestores, detalhes mais aprofundados não podem ser monitorados para fins de confidencialidade.

Figura 43 - Exemplo de e-mail enviado para os gestores da área ao final da execução do robô

Prezados gestores,

A automação de conferência de apontamentos foi executada e foram encontradas as seguintes inconsistências nos apontamentos do time:

Nome	Período contemplado (horas)	Porcentagem de alocação (%)	Horas pendentes	E-mail
Anna [REDACTED]	123.5	81.25	28.5	anna[REDACTED].com.br
Ariel [REDACTED]	85.5	59.38	58.5	ariel[REDACTED].com.br
Cesar [REDACTED]	147.5	97.04	4.5	cesar[REDACTED].com.br
Gabriel [REDACTED]	108	100	0	gabriel[REDACTED].com.br
Renan [REDACTED]	113	100.89	-1	renan[REDACTED].com.br
Robert [REDACTED]	144	100	0	robert[REDACTED].com.br

Todos os usuários já foram notificados!

Fonte: Do autor (2022).

3.3.2 Processo desenvolvido utilizando *Python*

Conforme informado no subcapítulo 2.3, o desenvolvimento da solução em *Python* começou partindo da escolha do ambiente em que seria desenvolvido, que neste caso foi o *Jupyter Notebook*, com o decorrer do desenvolvimento do projeto foi tomada a decisão de que esta solução só iria automatizar os blocos “Verificar Apontamentos” e “Gerar Alertas” do fluxo.

Partindo do início da solução em *Python*, foi seguido a seguinte ordem de blocos dentro do código desenvolvido:

- Bloco 1 - Foram importadas as bibliotecas necessárias para o funcionamento correto do projeto e definido o vetor denominado `vetor_dia_da_semana`. O código do bloco 1 pode ser visto na figura 44.

Figura 44 - Bloco 1 da solução em *Python* desenvolvida no *Jupyter Notebook*

```
In [1]: import pandas as pd
        from datetime import datetime, date, timedelta
        from calendar import monthrange

        vetor_dia_da_semana = {0:"Segunda",
                                1:"Terça",
                                2:"Quarta",
                                3:"Quinta",
                                4:"Sexta",
                                5:"Sabado",
                                6:"Domingo"}
```

Fonte: Do autor (2022).

- Bloco 2 - Responsável pela leitura dos arquivos que serão utilizados dentro do projeto e ajustes necessários nestes. O código do bloco 2 pode ser visto na figura 45.

Figura 45 - Bloco 2 da solução em *Python* desenvolvida no *Jupyter Notebook*

```
In [2]: df_usuarios = pd.read_excel(r"C:\Users\WilliamMarquesPereir\Desktop\TCC\Projeto Python\InputRElo001.xlsx",
                                   sheet_name = "Usuarios")
        df_usuarios.rename(columns={"Quantidade de horas diarias": "hora_contrato", "Nome": "usuario"}, inplace = True)

        df_feriados = pd.read_excel(r"C:\Users\WilliamMarquesPereir\Desktop\TCC\Projeto Python\InputRElo001.xlsx",
                                   sheet_name = "Feriados")
        df_feriados.data = df_feriados.data.apply(lambda x: x.strftime('%Y-%m-%d'))
        df_feriados.feriado = 1

        df_ausencia = pd.read_csv(r"C:\Users\WilliamMarquesPereir\Desktop\TCC\Projeto Python\Alocacao.csv", sep = ",")
        df_ausencia = df_ausencia[df_ausencia.Nome.isin(df_usuarios.usuario.unique())]

        df_timelog = pd.read_csv(r"C:\Users\WilliamMarquesPereir\Desktop\TCC\Projeto Python\timelog.csv", sep = ",")
        df_timelog.rename(columns={"user": "usuario", "date": "data"}, inplace = True)
        df_timelog = df_timelog[df_timelog.usuario.isin(df_usuarios.usuario.unique())]
```

Fonte: Do autor (2022).

- Bloco 3 - Encarregado pela definição das funções do projeto. O código do bloco 3 pode ser visto na figura 46.

Figura 46 - Bloco 3 da solução em *Python* desenvolvida no *Jupyter Notebook*

```
In [3]: def dias_entre_datas(data_inicio, data_fim):
        vetor_datas = []
        while (data_inicio <= data_fim):
            vetor_datas.append(data_inicio)
            data_inicio = data_inicio + timedelta(days = 1)
        return vetor_datas

def compatibiliza_datas(vetor_datas, dicionario_datas, nome, horas_diarias):
    if nome in dicionario_datas:
        for data in vetor_datas:
            dicionario_datas[nome][data] = horas_diarias
    else:
        dicionario_datas[nome] = {}
        for data in vetor_datas:
            dicionario_datas[nome][data] = horas_diarias
    return dicionario_datas

def ajusta_ausencias(df_ausencia):
    df_ausencia["Data Inicial"] = df_ausencia["Data Inicial"].apply(lambda x: datetime.strptime(x, '%d/%m/%Y'))
    df_ausencia["Data Final"] = df_ausencia["Data Final"].apply(lambda x: datetime.strptime(x, '%d/%m/%Y'))
    df_ausencia["Datas"] = df_ausencia.apply(lambda x: dias_entre_datas(x["Data Inicial"], x["Data Final"]), axis = 1)
    dicionario_datas = {}
    df_ausencia.apply(lambda x: compatibiliza_datas(x["Datas"], dicionario_datas, x["Nome"], x["Quantidade de horas diárias"]),
                      axis = 1)
    return dicionario_datas

def verifica_ausencia(data, usuario, dicionario_ausencias):
    data = datetime.strptime(data, '%Y-%m-%d')
    if data in dicionario_ausencias[usuario]:
        return dicionario_ausencias[usuario][data]
    else:
        return 0

def verifica_maior(a,b):
    if a > b:
        return a
    else:
        return b

def verifica_menor(a,b):
    if a < b:
        return a
    else:
        return b

def verifica_conformidade(hora_contrato, minutos):
    return verifica_maior(hora_contrato - minutos, 0)
```

Fonte: Do autor (2022).

- Bloco 4 - Incubido de verificar o apontamento de cada usuário e inserir na *DataTable* o apontamento dos usuários até o dia atual do mês executado. O código do bloco 4 pode ser visto na figura 47.

Figura 47 - Bloco 4 da solução em *Python* desenvolvida no *Jupyter Notebook*

```
In [4]: data_base = datetime.today()
last_day = data_base.day
vetor_data = []
for i in range(1, last_day+1):
    vetor_data.append(data_base.replace(day = i).strftime('%Y-%m-%d'))
df_base = pd.DataFrame(vetor_data, columns = ["data"])

df = pd.DataFrame(columns = ["usuario", "data"])
for nome in df_usuarios["usuario"].unique():
    df_base["usuario"] = nome
    df = pd.concat([df, df_base])

df["dia_da_semana"] = df["data"].apply(lambda x: vetor_dia_da_semana[datetime.strptime(x, '%Y-%m-%d').weekday()])

df = df.merge(df_usuarios[["usuario", "hora_contrato"]], on = "usuario", how = "left")
df = df.merge(df_feriados, on = "data", how = "left")

dicionario_ausencias = ajusta_ausencias(df_ausencia.copy())
df["ausencia"] = df.apply(lambda x: verifica_ausencia(x["data"], x["usuario"], dicionario_ausencias), axis = 1)

df_timelog = df_timelog[["data", "usuario", "minutes"]].groupby(["data", "usuario"]).sum().reset_index()
df_timelog["minutes"] = df_timelog["minutes"]/60

df = df.merge(df_timelog, on = ["usuario", "data"], how = "left")
df.fillna(0, inplace = True)

df = df[~(df.dia_da_semana.isin(["Sabado", "Domingo"])) & (df.feriado == 0)]

df["ausencia"] = df.apply(lambda x: verifica_menor(x["ausencia"], x["hora_contrato"]), axis = 1)
df["hora_contrato"] = df["hora_contrato"]-df["ausencia"]
df["conformidade"] = df.apply(lambda x: verifica_conformidade(x["hora_contrato"], x["minutes"]), axis = 1)

df = df[["usuario", "hora_contrato", "minutes", "conformidade"]].groupby(["usuario"]).sum().reset_index()
df["Porcentagem de alocação (%)"] = round(df["minutes"]/df["hora_contrato"]*100,2)
df = df[["usuario", "minutes", "Porcentagem de alocação (%)", "conformidade"]]
df.rename(columns={"usuario": "Nome", "minutes": "Período contemplado (horas)", \
    "conformidade": "Horas pendentes"}, inplace = True)
```

Fonte: Do autor (2022).

- Bloco 5 - Gera o arquivo no formato “.xlsx” com os apontamentos dos usuários no mês atual até a data da execução, este arquivo será gerado na pasta de execução do projeto. O código do bloco 5 pode ser visto na figura 48.

Figura 48 - Bloco 5 da solução em *Python* desenvolvida no *Jupyter Notebook*

```
In [5]: df.to_excel(r"C:\Users\WilliamMarquesPereir\Desktop\TCC\Projeto Python\Alocacoes.xlsx", index = False)
```

Fonte: Do autor (2022).

3.3.3 Interface visual do *PowerApps*

A interface visual do *PowerApps* foi criada com a identidade visual do IFSC e este pode ser utilizado tanto em formato mobile, web ou através do *Teams* que é um *software* da *Microsoft* desenvolvido para a colaboração de equipes corporativas.

O aplicativo corporativo desenvolvido para este projeto tem uma tela inicial onde não é necessário realizar login, a partir de uma lista de usuários criado no *Sharepoint* é feito o link de conexão com o e-mail acessado no aplicativo da *PowerApps* e caso haja um usuário cadastro na lista com este e-mail o mesmo já irá identificar o e-mail e perfil do mesmo, esta tela pode ser vista em dois formatos, para aqueles que tem o perfil “Gestor” estes verão a logo do IFSC e três botões clicáveis (Ausências, Feriados, Usuários), já para os que tiverem o perfil “User” verão a logo do IFSC e apenas os dois primeiros botões clicáveis. A tela inicial versão mobile pode ser vista através da figura 49.

Figura 49 - Tela inicial na versão mobile do aplicativo corporativo desenvolvido via *PowerApps*



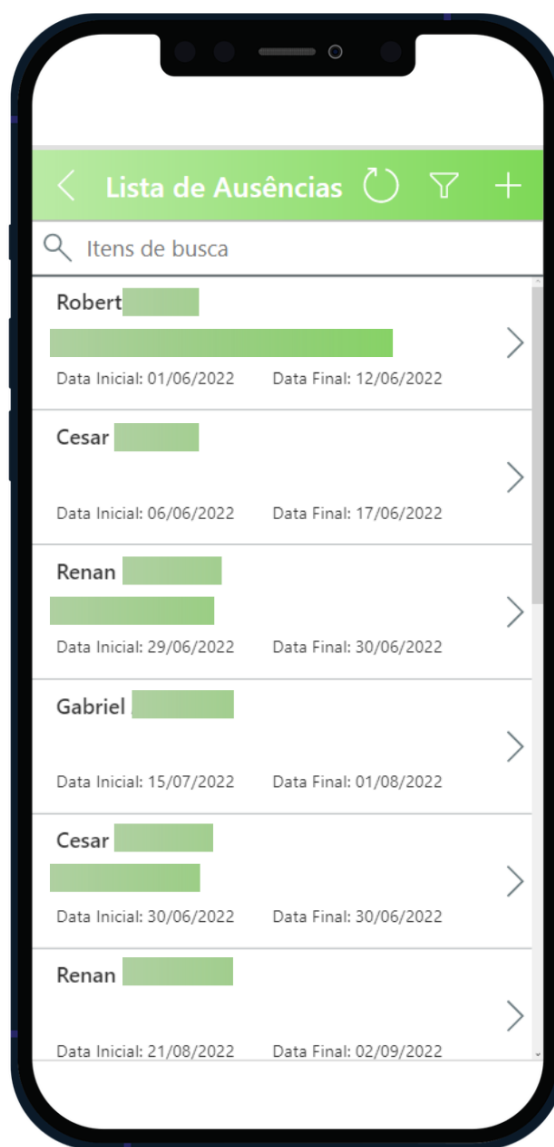
Fonte: Do autor (2022).

Os botões clicáveis direcionam para outras telas, sendo estas:

- Ausência - Direcionado para a tela de ausências, onde mostram todas as ausências dos usuários, nesta pode ser feito filtros específicos (data, usuário e tipo de ausência), clicando no ícone “>” é direcionado para a tela onde mostra todas as informações da ausência específica e também pode ser clicado no ícone de “+” que será redirecionado para

tela de cadastro de nova ausência. A tela de ausências em versão mobile pode ser vista na figura 50, algumas informações foram cobertas para fins de confidencialidade.

Figura 50 - Tela de ausências na versão mobile do aplicativo corporativo desenvolvido via *PowerApps*



Fonte: Do autor (2022).

- **Feriados** - Direcionado para a tela de feriados, onde mostram todos os feriados anuais cadastrados na lista de feriados dentro do *Sharepoint*, nesta podem ser feito filtros específicos (mês e ano) e também pode ser clicado no ícone de “+” que será redirecionado para tela de cadastro de novo feriado. A tela de feriados em versão mobile pode ser vista na figura 51, algumas informações foram cobertas para fins de confidencialidade.

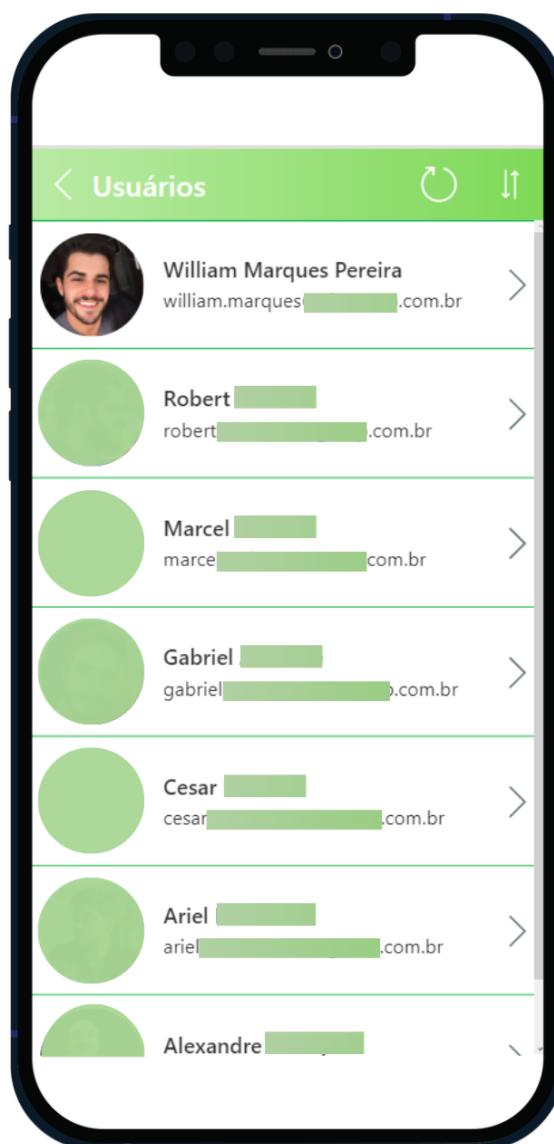
Figura 51 - Tela de feriados na versão mobile do aplicativo corporativo desenvolvido via *PowerApps*



Fonte: Do autor (2022).

- **Usuários** - Direcionado para a tela de usuários, onde mostram todos os usuários cadastrados na lista de usuários dentro do *Sharepoint*. A tela de usuários em versão mobile pode ser vista na figura 52, algumas informações foram cobertas para fins de confidencialidade.

Figura 52 - Tela de usuários na versão mobile do aplicativo corporativo desenvolvido via *PowerApps*



Fonte: Do autor (2022).

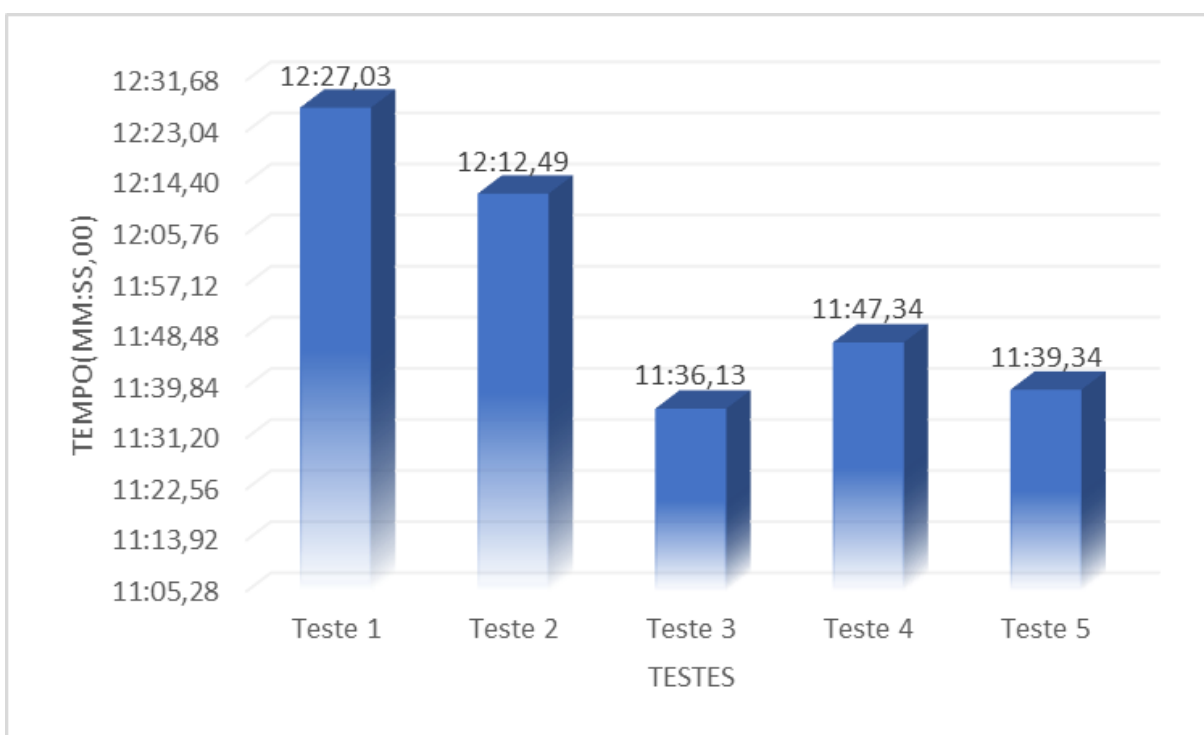
4. RESULTADOS OBTIDOS

Com as duas soluções desenvolvidas, os testes foram executados repetidas vezes para captura de dados e posteriormente análise dos mesmos. Para fins de comparação de velocidade de automação será comparado apenas as etapas equivalentes entre as duas soluções e os testes feitos manualmente.

4.1. Processo executado manualmente

Os testes manuais foram feitos por uma pessoa com conhecimentos básicos de excel e de navegação web, e foram realizados 5 testes. De acordo com o Gráfico 1, os testes manuais tiveram resultados similares, mantendo uma média de 11,94min, com um desvio padrão de 0,37min.

Gráfico 1 - Testes manuais de execução do processo



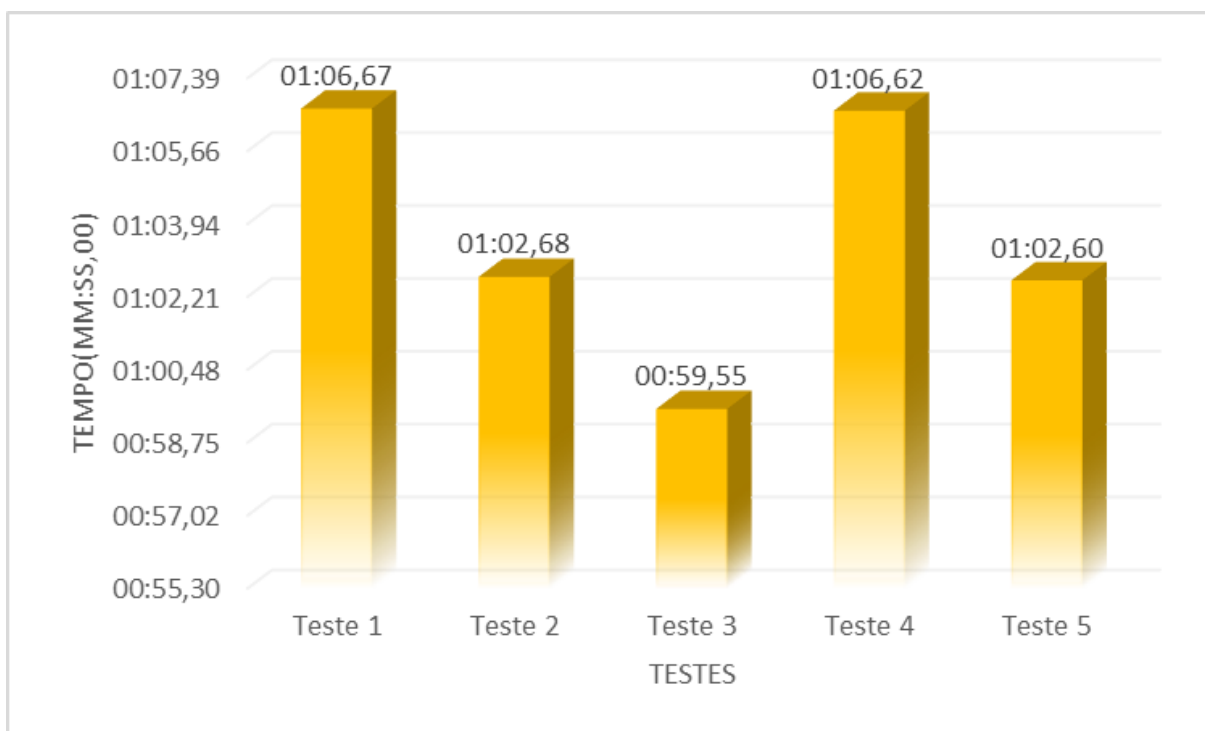
Fonte: Elaborado pelo autor (2022).

4.2. Testes utilizando *UiPath*

Os testes foram separados em duas partes. Primeiramente a solução completa desenvolvida no *UiPath* simulando todo o processo desde acessar páginas, baixar os arquivos

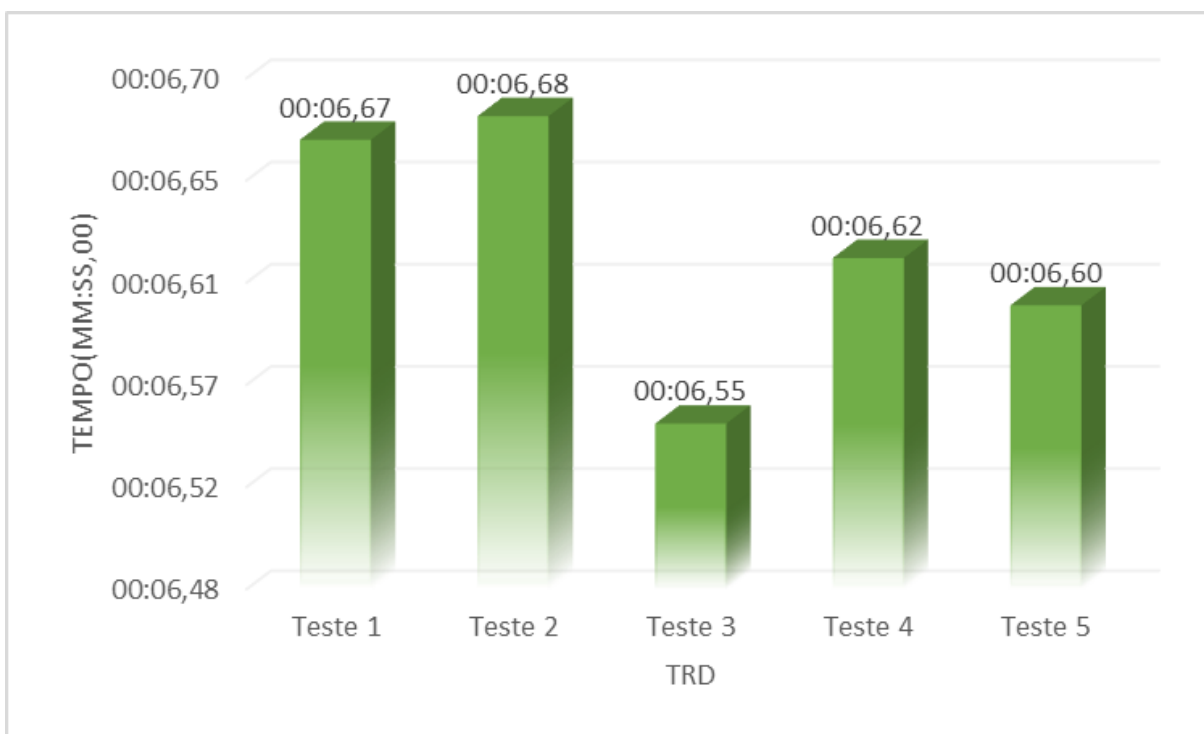
que serão utilizados para fazer a análise de apontamento de horas até o envio de e-mail final para os gestores da área. Os tempos desta ação podem ser vistos no Gráfico 2 e os testes obtiveram um tempo médio de execução de 1,06 min, com um desvio padrão de 0,05 min.

Gráfico 2 - Testes utilizando a solução completa desenvolvida no *UiPath*



Fonte: Elaborado pelo autor (2022).

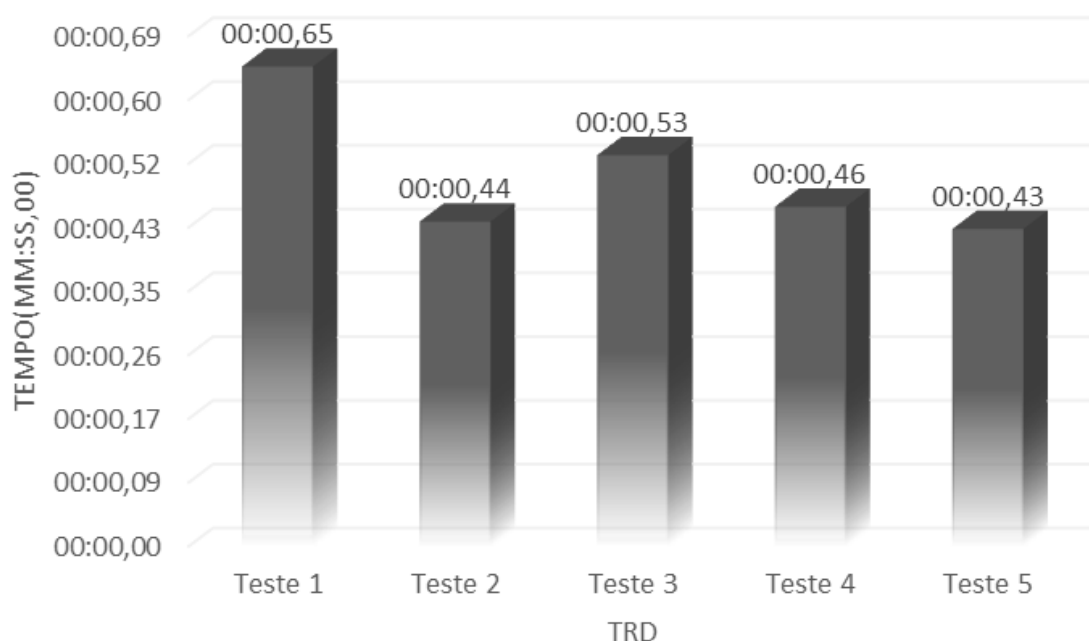
Os resultados da segunda solução são observados no Gráfico 3, onde é medido o tempo de execução referente às etapas equivalentes ao método manual (reduzindo o escopo de ação do UiPath. Chegou-se a um tempo médio de 6,52s com um desvio padrão de 0,05s.

Gráfico 3 - Testes utilizando a parte comparativa desenvolvida no *UiPath*

Fonte: Elaborado pelo autor (2022).

4.3. Testes utilizando *Python*

Os testes utilizando a solução desenvolvida em *Python* tiveram maior eficiência mantendo um tempo médio de 00,50s, com um desvio padrão de 0,09s, que podem ser observados no Gráfico 4.

Gráfico 4 - Testes realizados com a solução em *Python*

Fonte: Elaborado pelo autor (2022).

4.4. Comparação das soluções

Utilizando as três soluções equivalentes como comparação, nota-se que a automatização do processo realizada usando *Python* foi a que teve maior desempenho temporal pois não levou 1 segundo de execução. Comparando com a solução desenvolvida via *UiPath*, observa-se uma pequena diferença de tempo de execução tornando-as quase equivalentes.

A solução feita em *Python* por mais que execute totalmente em *backend* ou seja em segundo plano, é uma solução mais “difícil” de ser desenvolvida, pois ela envolve diversos fatores, como lógica de programação, os tipos de variáveis, os tipos de dados, entender o que é lista, o que é dicionário e por fim o conhecimento da linguagem de programação *Python* que por mais que seja consideravelmente fácil, é necessário conhecer os conceitos básicos de programação. Além dos fatores citados anteriormente, esta solução acaba tendo outros

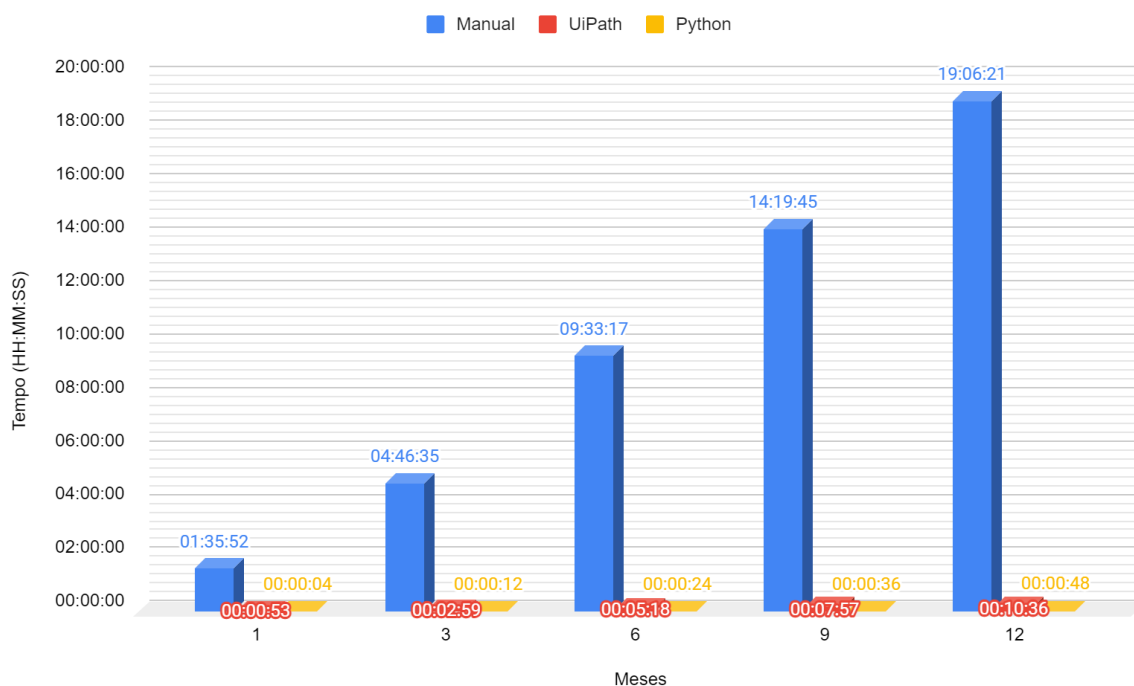
impedimentos que travam para o não desenvolvimento completo do processo, como por exemplo as etapas de acesso ao as páginas web onde serão feito a extração dos documentos para análise de apontamentos, por utilizarem acessos muitas vezes corporativos é necessário uma dupla autenticação e por muitas vezes as API's utilizadas em *Python* impedem esse tipo de tipo de acesso.

Já a solução elaborada via *UiPath*, por utilizar uma programação *Low-Code* além de ter uma metodologia de “arrasta e solta”, ela se torna mais fácil de ser lida e entendida por causa de sua interface visual muito amigável, mas assim como em *Python* também é necessário ter uma base de lógica de programação, mas diferente da solução em *Python*, ela permitiu realizar o processo por completo mais facilmente, pois além de acessar todas as páginas, realizou os tratamentos necessários no apontamento de horas dos usuários e fez o envio de e-mail final para os gestores e usuários.

Analisando-se o Gráfico 5, nota-se que o desempenho de automatização do processo via *UiPath* em relação ao processo manual é muito significativo. Em processos que precisam ser repetidos muitas vezes por semana, o tempo economizado é extremamente importante, podendo justificar a diminuição do quadro de funcionários de uma empresa. Percentualmente, o processo automatizado via *UiPath* equivale a 0,076% do tempo gasto em relação ao processo manual. Comparando os resultados do processo em *Python* em relação ao processo manual, o tempo gasto no processo é de 0,069%.

Para fins comparativos com o processo manual, se executar esse processo apenas 2 vezes por semana, durante um ano, o processo em *Uipath* economizaria o tempo de 18h55min45s, e o processo em *Python* economizaria 19h5min33s.

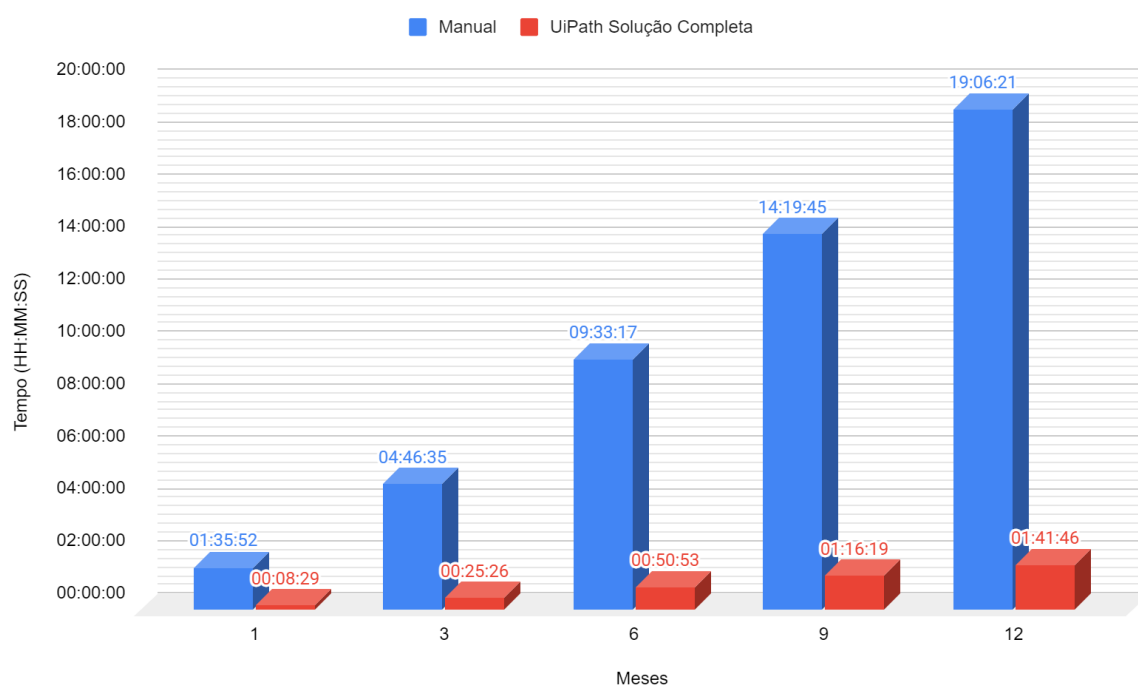
Gráfico 5 - Comparativo de Tempo X Meses das soluções desenvolvidas em relação às etapas equivalentes ao processo manual.



Fonte: Elaborado pelo autor (2022).

Levando em consideração do tempo que a solução completa desenvolvida com o *UiPath* gasta na execução processo é de 8,877% em relação ao processo feito manualmente, é uma redução média de tempo de 17,41h em 1(um) ano de trabalho executando o processo 2 vezes por semana.

Gráfico 6 - Comparativo de Tempo X Meses da solução completa do processo desenvolvido com o *UiPath* em relação ao processo manual.



Fonte: Elaborado pelo autor (2022).

5. CONCLUSÃO

É notório, pelos resultados obtidos, que os benefícios da automatização deste processo são muito significativos, pois o custo envolvido é menor quando se automatiza processos, ainda mais se são processos de alta frequência.

O fato de analisar-se um caso real, aplicado para empresas que possuem funcionários em *home office* em vários departamentos, economiza tempo de trabalho do funcionário responsável pelo processo, evitando falhas, tornando-a uma empresa ainda mais automatizada e bem conceituada tanto pelos funcionários quanto pelo público externo.

Sobre o processo manual, apesar de ser muito utilizado ainda, observa-se que o desvio padrão de tempo para sua realização não pode ser visto como constante ao longo do número de vezes com que ele é executado. A medida que o número de repetições for maior, a tendência de um processo manual é que o operador fique cansado, e com isso aumente o desvio padrão, e aumente a quantidade de erros cometidos pela operação manual. Por estes motivos, a tendência é retirar processos repetitivos, feitos manualmente, e trocar por um RPA.

Ao analisar-se essas soluções com *UiPath* e *Python* percebe-se que apresentam resultados similares, apontando para o uso do *UiPath* como solução, já que o mesmo é considerado pouco complexo de ser realizado, e com muitas funções embutidas na própria ferramenta, se comparados ao Python.

O *UiPath*, como pode ser visto durante o desenvolvimento do projeto, mostrou-se uma ferramenta simples de ser utilizada, intuitiva, de fácil aprendizado, com extremo potencial para desenvolvimento de soluções das mais simples às mais complexas, podendo fazer controle de login, extração de informações, manipulação de dados, leitura de imagens, entre muitas outras funções. A maior dificuldade encontrada no *UiPath* é conhecer o *framework*, ou seja, conhecer bem as funções que existem, e que são necessárias para uma melhor fluidez do processo construtivo do software.

Já a solução desenvolvida em *Python*, por mais que a execução de todas as etapas do processo sejam muito rápidas, mantendo um baixíssimo tempo de execução, apresentou algumas limitações durante o processo de desenvolvimento da mesma. A maior dificuldade é que precisa ter-se conhecimentos relativamente aprofundados da linguagem de programação e

das bibliotecas de acesso ao Excel e o uso do Pandas, o que se mostrou ser bem mais complicado de entender, se comparado com o *UiPath*.

Um outro ponto muito importante notado durante o desenvolvimento do projeto foi que o *UiPath* serve bem para processos que podem ser executados periodicamente e que não exijam tempos de execução muito pequenos. Do contrário, *Python* é mais aconselhável, apesar de ser mais difícil de programar.

Sugere-se para trabalhos futuros com soluções em RPA usar na área de controle de peças fabricadas, análise de qualidade, captura do tempo de funcionamento das máquinas utilizadas na mecatrônica e para geração de alerta de manutenção preventiva baseada em algoritmos de predição.

Por fim, as soluções desenvolvidas durante este projeto atenderam as expectativas como previsto no início do projeto, pois foram feitos 3 procedimentos diferentes para automação de dados de uma pessoa em home office, e se comparou estes dados, chegando a conclusão das vantagens e desvantagens de cada um deles. O foco foi também mostrar o processo de RPA em Low Code, objetivo que também se chegou corretamente. Conclui-se que o uso de RPA tem se mostrado como uma ferramenta eficiente, que tem qualidade de mercado, praticidade de manipulação e uma taxa de aprendizado alta se comparada a linguagens de programação tradicionais, tais como Python.

REFERÊNCIAS

ANÁPOLIS, Dm (ed.). **Gartner diz que mercado de software de RPA movimentará US\$ 2,9 bilhões em 2022**. Disponível em:

<https://www.dmanapolis.com.br/noticia/38049/gartner-diz-que-mercado-de-software-de-rpa-movimentara-us-2-9-bilhoes-em-2022>. Acesso em: 04 out. 2022.

ANÁPOLIS, Dm (ed.). **Mercado de RPA e Hiperautomação deve ir à US\$ 26 bi até 2027**. Disponível em:

<https://www.dmanapolis.com.br/noticia/35387/mercado-de-rpa-e-hiperautomacao-deve-ir-a-us-26-bi-ate-2027>. Acesso em: 04 out. 2022.

BALHESSA,

Mauro. **https://www.istoedinheiro.com.br/tedio-no-trabalho-tambem-afeta-a-saude-conheca-a-sindrome-de-boreout/**. 2021. Disponível em:

<https://www.istoedinheiro.com.br/tedio-no-trabalho-tambem-afeta-a-saude-conheca-a-sindrome-de-boreout/>. Acesso em: 16 jul. 2022.

CHANNEL, Info. **Mercado global de RPA deve faturar quase US\$ 3 bi, avalia o Gartner**. Disponível em:

<https://inforchannel.com.br/2022/08/02/mercado-global-de-rpa-deve-faturar-quase-us-3-bi-avalia-o-gartner/>. Acesso em: 04 out. 2022.

CODE, Let'S. **Introdução ao Jupyter Notebook**. 2019. Disponível em:

<https://letscode.com.br/blog/introducao-ao-jupyter-notebook>. Acesso em: 09 nov. 2022.

CONHECIMENTO, Revista Científica Núcleo do. **O que é Pesquisa-Ação? Quando utilizar a Pesquisa-Ação?** 2021. Disponível em:

<https://www.youtube.com/watch?v=HqmCZw5AkAE>. Acesso em: 11 jun. 2022.

COSTELLO, Katie; RIMOL, Meghan. **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: 16 jun. 2022.

EXAMPLE, Tutorial And. **RPA Tutorial**. Disponível em:

<https://www.tutorialandexample.com/rpa-tutorial>. Acesso em: 31 out. 2022.

FRANKENFIELD, Jake. **Robotic Process Automation (RPA)**. 2022. Disponível em:

<https://www.investopedia.com/terms/r/robotic-process-automation-rpa.asp>. Acesso em: 04 out. 2022.

GLOBENEWSWIRE. **Robotic Process Automation Market to Hit USD 23.9 Bn by 2030**. 2022. Disponível em:

<https://roboticulized.com/artificial-intelligence/2022/01/06/74858/robotic-process-automation-market-to-hit-usd-23-9-bn-by-2030/>. Acesso em: 01 nov. 2022.

GOLUBENKO, Sergey. **Important Considerations about SharePoint Customization**. 2019. Disponível em: <https://www.scnsoft.com/blog/sharepoint-customization>. Acesso em: 06 nov. 2022.

JUPYTER. **Jupyter Notebook: The Classic Notebook Interface**. Disponível em: <https://jupyter.org>. Acesso em: 09 nov. 2022.

KONRAD, Alex. **Daniel Dines cria empresa de tecnologia avaliada em R\$ 7 bilhões Leia mais em:** <https://forbes.com.br/principal/2020/03/daniel-dines-cria-empresa-de-tecnologia-avaliada-em-r-7-bilhoes/>. Disponível em: <https://forbes.com.br/principal/2020/03/daniel-dines-cria-empresa-de-tecnologia-avaliada-em-r-7-bilhoes/>. Acesso em: 31 out. 2022.

MARCONDES, Me. Renatto Cesar. **Psicólogo explica o que é a Síndrome de Boreout**. 2021. Disponível em: <https://revistavisaohospitalar.com.br/psicologo-explica-o-que-e-a-sindrome-de-boreout/>. Acesso em: 16 jul. 2022.

MELO, Diego. **O que é Python? [Guia para iniciantes]**. 2020. Disponível em: <https://tecnoblog.net/responde/o-que-e-python-guia-para-iniciantes/#:~:text=Python%20é%20uma%20linguagem%20de,vantagens%20neste%20guia%20para%20iniciantes>. Acesso em: 06 nov. 2022.

MICROSOFT. **What is Power Platform**. 2018. Disponível em: <https://powerplatform.microsoft.com/pt-br/what-is-power-platform/>. Acesso em: 06 nov. 2022.

MICROSOFT. **Administrando uma implantação do Power Apps Enterprise**. 2022. Disponível em: <https://learn.microsoft.com/pt-br/power-platform/admin/admin-powerapps-enterprise-deployment>. Acesso em: 06 nov. 2022.

MICROSOFT. **Por que usar o Power Apps como parte da Microsoft Power Platform**. 2022. Disponível em: <https://powerplatform.microsoft.com/pt-br/power-apps/>. Acesso em: 06 nov. 2022.

MICROSOFT. **O que é o Power Apps?** 2022. Disponível em: <https://learn.microsoft.com/pt-br/power-apps/powerapps-overview>. Acesso em: 06 nov. 2022.

Ministério da Saúde (org.). **Dor relacionada ao trabalho: Lesões por esforços repetitivos (LER) Distúrbios osteomusculares relacionados ao trabalho (Dort)**. Disponível em: https://bvsms.saude.gov.br/bvs/publicacoes/dor_relacionada_trabalho_ler_dort.pdf. Acesso em: 11 jun. 2022.

MICROSOFT. **Licensing overview for Microsoft Power Platform**. Disponível em: <https://learn.microsoft.com/en-us/power-platform/admin/pricing-billing-skus>. Acesso em: 06 nov. 2022.

NAIDITCH, Fernando. **PESQUISA-AÇÃO**. 2022. Disponível em: <https://gestrado.net.br/verbetes/pesquisaacao/>. Acesso em: 11 jun. 2022.

PEREIRA, Tábata Fernandes. **Estrutura para condução da pesquisa-ação**. Disponível em: https://www.researchgate.net/figure/Figura-1-Estrutura-para-conducao-da-pesquisa-acao_fig1_312039546. Acesso em: 11 jun. 2022.

PYTHON. Python Programming Language. Disponível em <http://www.python.org/>. Acesso em 06 nov. 2022

SAÚDE, Conexa. **Síndrome de boreout: o que é, sintomas, e como o RH pode combatê-la**. 2022. Disponível em: <https://www.conexasaude.com.br/blog/boreout/#:~:text=Portanto%20a%20síndrome%20de%20boreout,cansaço%20mental%2C%20falta%20de%20estímulo>. Acesso em: 11 jun. 2022.

TANAJURA, Laudelino Luiz Castro; BEZERRA, Ada Augusta Celestino. **ESQUISA-AÇÃO SOB A ÓTICA DE RENÉ BARBIER E MICHEL THIOLENT: APROXIMAÇÕES E ESPECIFICIDADES METODOLÓGICAS**. Disponível em: https://www.academia.edu/14466599/PESQUISA-AÇÃO_SOB_A_ÓTICA_DE_RENÉ_BARBIER_E_MICHEL_THIOLENT_APROXIMAÇÕES_E_ESPECIFICIDADES_METODOLOGICAS. Acesso em: 19 jun. 2022.

TRIPATHI, Alok Mani. **Learning Robotic Process Automation:: create software robots and automate business processes with the leading rpa tool *uipath..** Birmingham: Packt Publishing, 2018.

UIPATH (org.). **UiPath Studio**. Disponível em: <https://www.uipath.com/pt/product/studio-1>. Acesso em: 19 jun. 2022.

UIPATH. **Robots**. Disponível em: <https://docs.uipath.com/orchestrator/docs/about-robots>. Acesso em: 31 out. 2022.

UIPATH (org.). **Organização de projetos**. 2022. Disponível em: https://docs.uipath.com/studio/lang-pt_BR/docs/project-organization Acesso em: 01 nov. 2022.

UIPATH (org.). **Sobre Licenças**. Disponível em: https://docs.uipath.com/automation-suite/lang-pt_BR/docs/about-licensing. Acesso em: 02 nov. 2022.

VANESSA. **Automação com Robô RPA: como surgiu, o que é, e no que vai se transformar**. Disponível em: <http://www.ptdigital.com.br/automacao-com-robo-rpa-como-surgiu-o-que-e-e-no-que-vai-se-transformar/>. Acesso em: 04 out. 2022.