

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS**

**FERNANDO CROZATTI**

**AUTOMAÇÃO DOS TESTES DE VALIDAÇÃO  
EM UMA PLATAFORMA DE COMÉRCIO ELETRÔNICO**

**CAMPINAS**

**2021**

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS  
CENTRO DE CIÊNCIAS EXATAS, AMBIENTAIS E DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM GESTÃO DE REDES DE  
TELECOMUNICAÇÕES  
FERNANDO CROZATTI**

**AUTOMAÇÃO DOS TESTES DE VALIDAÇÃO  
EM UMA PLATAFORMA DE COMÉRCIO ELETRÔNICO**

Dissertação apresentada ao Programa de Pós-Graduação *Stricto Sensu* em Gestão de Redes de Telecomunicações do Centro de Ciências Exatas, Ambientais e de Tecnologias, da Pontifícia Universidade Católica de Campinas, como exigência para obtenção do título de Mestre em Gestão de Redes de Telecomunicações.

Orientador: Prof. Dr. Eric Alberto de Melo Fagotto

Coorientador(a): Prof(a). Dr(a). Cecília de Freitas Morais

**CAMPINAS**

**2021**

Ficha catalográfica elaborada por Adriane Elane Borges de Carvalho CRB 8/9313  
Sistema de Bibliotecas e Informação - SBI - PUC-Campinas

001.63 C954a	<p>Crozatti, Fernando</p> <p>Automação dos testes de validação em uma plataforma de comércio eletrônico / Fernando Crozatti. - Campinas: PUC-Campinas, 2021.</p> <p>73 f.: il.</p> <p>Orientador: Eric Alberto de Melo Fagotto; Coorientador: Cecília de Freitas Morais.</p> <p>Dissertação (Mestrado em Gestão de Redes de Telecomunicações) - Gestão de Redes de Telecomunicações, Centro de Ciências Exatas, Ambientais e de Tecnologia, Pontifícia Universidade Católica de Campinas, Campinas, 2021. Inclui bibliografia.</p> <p>1. Automação. 2. Comércio eletrônico. 3. Estudo de caso. I. Fagotto, Eric Alberto de Melo. II. Morais, Cecília de Freitas III. Pontifícia Universidade Católica de Campinas. Centro de Ciências Exatas, Ambientais e de Tecnologia. Gestão de Redes de Telecomunicações. IV. Título.</p> <p>CDD - 22. ed. 001.63</p>
-----------------	--

**FERNANDO CROZATTI**

**AUTOMAÇÃO DOS TESTES DE VALIDAÇÃO EM UMA  
PLATAFORMA DE COMÉRCIO ELETRÔNICO**

Dissertação apresentada como exigência para obtenção do título de Mestre em Gestão de Redes de Telecomunicações ao Programa de Pós-Graduação em Gestão de Redes de Telecomunicações do Centro de Ciências Exatas, Ambientais e de Tecnologias.

Área de Concentração: Gestão de Redes e Serviços.

Orientador: Prof. Dr. Eric A. de M. Fagotto.

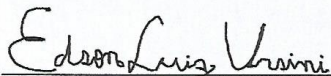
Coorientadora: Profa. Dra. Cecília de F. Moraes.

Dissertação defendida e aprovada em 17 de dezembro de 2021 pela Comissão Examinadora constituída dos seguintes professores:



---

Prof. Dr. Eric Alberto de Mello Fagotto  
Orientador da Dissertação e Presidente da Comissão Examinadora  
Pontifícia Universidade Católica de Campinas



---

Prof. Dr. Edson Luiz Ursini  
Universidade Estadual de Campinas - UNICAMP



---

Prof. Dr. Marcius Fabius Henriques de Carvalho  
Pontifícia Universidade Católica de Campinas

Dedico esse trabalho aos meus pais, Luiz Fernando Crozatti e Marjorie Ap. Bortolotti Crozatti que sempre me apoiaram e não mediram esforços em me proporcionar estudos na vida, sempre me apoiando em cada passo e conquista. Dedico também à minha esposa Letícia Ap. de Souza Crozatti que sempre me apoia e ajuda nos estudos e na vida.

## **AGRADECIMENTOS**

Agradeço aos meus orientadores Professor Dr. Eric Alberto de Mello Fagotto e Professora Dr(a). Cecília de Freitas Moraes pelo apoio prestado durante a construção deste trabalho, à PUC Campinas e aos docentes pela dedicação dispensada e aos meus amigos que me ajudaram e apoiaram durante todo o curso. Agradeço também a Rafael Gomes Alcântara que proporcionou a realização deste trabalho e apoiou em sua execução.

## RESUMO

Grandes empresas estão investindo cada vez mais em *software* de aplicações para aumentar suas vendas e expandir seus negócios. Atualmente, uma fração significativa do tempo de desenvolvimento de uma aplicação é destinada aos testes de validação, que tem o intuito de garantir a qualidade do produto final. Um dos principais motivos da demora na realização desses testes é o baixo nível de automação nos processos das empresas, causado principalmente pela falta de profissionais qualificados no mercado. A automação de processos robóticos (*Robotic Process Automation – RPA*) é uma tecnologia que visa facilitar o desenvolvimento e a implementação de robôs de software que podem realizar tarefas repetitivas e estruturadas em sistemas de maneira automática, bem como atuar na interface do usuário. Dentre os vários benefícios da aplicação de RPA, pode-se citar o aumento da produtividade, redução dos custos e erros na execução de tarefas. Neste contexto, esta dissertação teve como objetivo realizar um estudo de caso, utilizando ferramentas de automação de testes convencionais e de RPA para automatizar testes de software de uma empresa multinacional e posteriormente analisar qual a melhor ferramenta para a empresa. As ferramentas de automação de testes escolhidas foram a UiPath, que possui o pacote Test Suite desenvolvido especificamente para automação de casos de testes, e a Selenium. Primeiramente, foram aplicadas à área de testes de qualidade de software da empresa. Isso permitiu que o setor de garantia de qualidade (do inglês Quality Assurance - QA) tivesse mais tempo disponível para realizar outras tarefas, que exigem maior nível de criatividade dentro do ciclo de desenvolvimento de uma aplicação. Finalmente, analisaram-se os efeitos do emprego das ferramentas de RPA UiPath e a Selenium na automação de casos de testes em comparação com a execução manual. Em média, a redução do tempo de execução dos testes no UiPath ficou em 39%, porém, obtiveram-se reduções entre de 73% e 17% enquanto no Python a média da redução do tempo de execução foi de 26%, produzindo até mesmo tempos de execução maiores que os realizados por um analista de testes de maneira manual.

**Palavras-chave:** Automação de processos robóticos. UiPath. Testes de validação de *software*. Garantia de Qualidade.

## ABSTRACT

Big companies have been increasing investments in application software to improve sales and expand business. Currently, a significant fraction of the development time of an application is devoted to validation tests, which are intended to ensure the quality of the final product. One of the main reasons for the delay in carrying out these tests is the low level of automation in company processes, mainly caused by the lack of qualified professionals in the market. Robotic Process Automation (RPA) is a technology that aims to facilitate the development and implementation of software robots that can perform repetitive and structured tasks in systems automatically, as well as act on the user interface (UI). Among the various benefits of applying RPA, we can mention the productivity increase, cost and errors reduction. In this context, this dissertation aims to carry out a case study, using conventional and RPA tools to automate software tests in e-commerce platform of a multinational company and later analyze which tool is the best for the company. The test automation tools chosen were UiPath, which has the Test Suite package specifically developed for test case automation, and Selenium. First, they were applied to the company's software quality testing area. This provided the Quality Assurance (QA) sector with more time available to perform other tasks, which popped a greater level of creativity within the development cycle of an application. Finally, we analyzed the effects of using the RPA tools UiPath and Selenium on the automation of test cases as compared to manual execution. On average, the test execution time reduction with UiPath was 39%, but there were reductions between 73% and 17%. Regarding the Selenium, the average execution time reduction was 26% despite there were run times longer than those obtained with tests performed manually.

**Keywords:** Robotic Process Automation. UiPath. Software tests. Quality assurance.



## LISTA DE FIGURAS

Figura 1: Quadrante Mágico Gartner 2021 para Robotic Process Automation.....	27
Figura 2: Comparativo das três principais ferramentas de RPA .....	28
Figura 3: Diagrama ilustrativo dos modelos B2B e B2C .....	35
Figura 4: Fluxograma inicial .....	37
Figura 5: Fluxograma da Categoria Promoção-1 .....	38
Figura 6: Fluxograma da Categoria Promoção-2 .....	41
Figura 7: Fluxograma da Categoria Promoção-3 .....	45
Figura 8: Como funciona o UiPath .....	46
Figura 9: Vista de um caso de teste minimizado no Open Browser.....	47
Figura 10: Caso de teste no UiPath .....	47
Figura 11: Arquitetura do Selenium Webdriver .....	49
Figura 12: Primeira página do fluxo do caso de testes em Python .....	50
Figura 13: Segunda página para implementação do fluxo do caso de testes em Python .....	50
Figura 14: Terceira página para implementação do fluxo do caso de testes em Python.....	51
Figura 15: Quarta página para implementação do fluxo do caso de testes em Python.....	52
Figura 16: Tempo de desenvolvimento UiPath x Python .....	56
Figura 17: Comparativo da redução percentual no tempo entre Python e UiPath.....	58
Figura 18: Tempo de execução dos testes manuais x automatizados UiPath x automatizados Python .....	60

## LISTA DE TABELAS

Tabela 1: 10 Principais ferramentas RPA disponíveis no mercado .....	22
Tabela 2: 10 Principais ferramentas de automação de testes disponíveis no mercado .....	29
Tabela 3: Listagem dos casos de testes .....	54

# SUMÁRIO

<b>1</b>	<b>Introdução.....</b>	<b>12</b>
1.1	Motivação e Justificativa.....	13
1.2	Objetivos .....	13
1.3	Organização do trabalho .....	14
<b>2</b>	<b>Revisão Bibliográfica .....</b>	<b>15</b>
2.1	O que é Robotic Process Automation? .....	15
2.2	Funcionalidades das Ferramentas de RPA .....	16
2.3	Benefícios das ferramentas de RPA .....	17
2.4	Desafios das ferramentas de RPA.....	20
2.5	Principais ferramentas RPA disponíveis no mercado .....	22
2.5.1	Automation Anywhere (AA).....	22
2.5.2	Microsoft Power Automate .....	23
2.5.3	BluePrism.....	23
2.5.4	Workfusion .....	23
2.5.5	PegaSystems .....	24
2.5.6	EdgeVerve.....	24
2.5.7	Nice.....	25
2.5.8	Kryon.....	25
2.5.9	Kofax .....	26
2.5.10	UiPath.....	26
2.5.11	Quadrante mágico Gartner e escolha da ferramenta RPA .....	26
2.6	Automação de testes .....	29
2.6.1	Principais ferramentas de automação de testes disponíveis no mercado .....	29
2.6.2	Selenium .....	30
2.6.3	Robotium .....	31
2.6.4	Ranorex.....	31
2.6.5	Robot Framework .....	31
2.6.6	Cypress.....	31
2.6.7	Appium .....	31
2.6.8	UFT.....	32
2.6.9	Test Studio .....	32
2.6.10	Capbara.....	32

2.7	Trabalhos correlatos .....	32
<b>3</b>	<b><i>Metodologia</i></b> .....	<b>34</b>
3.1	Modelagem dos testes de automação .....	37
3.2	Materiais e ferramentas computacionais .....	46
3.2.1	UiPath .....	46
3.2.2	Selenium .....	48
<b>4</b>	<b><i>Resultados</i></b> .....	<b>53</b>
<b>5</b>	<b><i>Conclusões e Considerações Finais</i></b> .....	<b>61</b>
<b>6</b>	<b><i>Referências</i></b> .....	<b>63</b>
<b>7</b>	<b><i>Anexos</i></b> .....	<b>71</b>

## LISTA DE ABREVIACOES E SIGLAS

AA	<i>Automation Anywhere</i>
API	<i>Application Programming Interface</i>
B2B	<i>Business to Business</i>
B2C	<i>Business to Consumer</i>
BDD	<i>Behavior Driver Development</i>
BPA	<i>Business Process Automation</i>
CI	<i>Continous Integration</i>
FTE	<i>Full-Time Employee</i>
GP	Gerente de Produto
GUI	<i>Graphical User Interface</i>
HTML	<i>HyperText Markup Language</i>
IA	Inteligncia Artificial
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
MAS	Metodologia gil Scrum
ML	<i>Machine Learning</i>
OCR	<i>Optical Character Recognition</i>
OOS	<i>Out of Stock</i>
QA	<i>Quality Assurance</i>
ROI	<i>Return Over Investment</i>
RPA	<i>Robotic Process Automation</i>
TI	Tecnologia da Informao
UFT	<i>Unified Functional Testing</i>
UI	<i>User Interface</i>
WFI	<i>Workforce Intelligence</i>

## 1 INTRODUÇÃO

Cada vez mais, a automação de testes é exigida nas empresas para assegurar a qualidade do *software* e ajudar o analista de testes na manutenção do código. Cerca de 25% do tempo de desenvolvimento de uma aplicação é dedicado aos testes de garantia, no entanto, menos de 20% dos testes são automatizados (CERNAT; STAICU; STEFANESCU, 2020). Existem diversas ferramentas específicas para automação de testes como, por exemplo, *Robot Framework* (ROBOT FRAMEWORK, 2021), Selenium (SELENIUM, 2021), Appium (APPIUM, 2021), Cypress (CYPRESS, 2021), entre outras. Contudo, as ferramentas de automação de processos robóticos (do inglês *Robotic Process Automation* - RPA) são mais acessíveis, por esse motivo é possível utilizá-las para testes de UI (interface do usuário, do inglês *user interface*), API (Interface de Programação de Aplicativos, do inglês *Application Programming Interface*), também na preparação de dados para testes, configuração de pré-condições, testes de regressão e verificações pós-condição, aumentando a eficiência dos testes. Alguns dos principais desafios da automação de testes de *software* são: o tempo de implementação, as constantes mudanças nos ambientes de testes, a necessidade de realização de testes de UI e o tempo despendido para aprendizado da ferramenta (YATSKIV *et al.*, 2019).

O interesse pelas ferramentas de RPA tem crescido nos últimos anos, pois são tecnologias que permitem automatizar tarefas por meio de componentes de *software*, que interagem com a interface do usuário de forma semelhante a um humano. Essa solução vem sendo aplicada em diversos contextos (CHACÓN-MONTERO; JIMÉNEZ-RAMÍREZ; ENRÍQUEZ, 2019), por se tratar de uma excelente alternativa para agilizar processos burocráticos e repetitivos que poderiam sofrer com erros humanos. Em (VAN DER AALST; BICHLER; HEINZL, 2018) é apresentada a seguinte definição:

“As ferramentas de RPA executam declarações [if, then, else] em dados estruturados, normalmente usando uma combinação de interações de interface de usuário, ou conectando-se a APIs para conduzir clientes-servidores, *mainframes* ou códigos HTML. Uma ferramenta RPA opera mapeando um processo em linguagem própria para que o robô de *software* siga, com tempo de execução alocado para executar o *script* por um painel de controle.”

Recentemente, passou-se a utilizar técnicas baseadas em RPA para automação de testes, o que motivou o surgimento de ferramentas específicas como é o caso do Test Suite do UiPath (UIPATH - TEST SUITE, 2021). Apesar do custo da licença, esse *software* de RPA tem potencial para aplicação nos mais diversos departamentos de uma empresa. Há uma vasta

quantidade de ferramentas de automação de testes que são usadas em aplicações WEB e *mobile*, todavia, as ferramentas de RPA atuais apresentam como vantagem sua flexibilidade de utilização em diversos tipos de testes (CERNAT; STAIKU; STEFANESCU, 2020).

## 1.1 MOTIVAÇÃO E JUSTIFICATIVA

Neste trabalho serão automatizados diversos casos de testes de uma plataforma de comércio eletrônico com o auxílio de uma ferramenta de RPA. A utilização do RPA na automação de testes foi motivada pela sua facilidade de aprendizado e sua simplicidade de implementação. Adicionalmente, o emprego da ferramenta RPA neste cenário justifica-se por ser uma estratégia extremamente escalável (permite o aumento do número de operações sem elevar proporcionalmente os custos e o tempo despendido). RPA é uma técnica com enorme potencial, visto que com ela é possível automatizar casos de testes WEB e *mobile*, além de automatizar processos mais amplos, o que permite um retorno sobre o investimento de forma mais rápida e eficiente.

Atualmente, o mercado de trabalho está exigindo analistas de qualidade com conhecimento em ferramentas de automação de casos de testes. Neste sentido, o uso de uma ferramenta de RPA pode trazer grandes benefícios, uma vez que a mesma utiliza fluxos para automatizar os casos de testes, ou seja, sua implementação não está diretamente vinculada ao aprendizado de uma linguagem de programação que depende de linhas de códigos extensos e complexos. Isso permite que os analistas de testes se adaptem mais facilmente ao uso dessa ferramenta, dado que não serão necessárias longas horas de estudos para conseguir aprender a aplicar tal linguagem de programação na solução do problema.

Em suma, com a percepção que o mercado de trabalho carece de analistas com conhecimento em *softwares* específicos capazes de controlar e gerenciar testes e a dificuldade enfrentada pelas empresas para incentivar o estudo de ferramentas de automação entre os funcionários, surgiu a ideia de investigar o emprego de uma ferramenta RPA na automação de casos de testes em uma plataforma e-commerce. O intuito deste trabalho de pesquisa é permitir a melhoria contínua do processo e evitar o erro humano, deixando pouco espaço para erros e anomalia.

## 1.2 OBJETIVOS

São objetivos deste trabalho:

1. Automatizar os casos de testes em uma empresa usando:

- a. Uma ferramenta de automação de testes convencional (Python+Selenium) baseada em uma linguagem de programação de alto nível orientada a objetos (Python) e
  - b. Uma ferramenta RPA (UiPath) com pacote computacional especialmente desenvolvido para automação de testes (Test Suite);
2. Comparar os tempos de desenvolvimento e de execução dos casos de testes usando ambas as ferramentas.
  3. Avaliar a utilização das ferramentas de RPA na automação de testes de software.

### **1.3 ORGANIZAÇÃO DO TRABALHO**

No Capítulo 1, apresentam-se a justificativa e motivação para a realização desse trabalho e os objetivos a serem atingidos. No Capítulo 2 apresenta e discute a revisão bibliográfica. No Capítulo 3 expõe-se a metodologia empregada para o desenvolvimento do trabalho. No Capítulo 4, os resultados do trabalho são apresentados e discutidos. As considerações finais e a conclusão encontram-se no Capítulo 5. O Capítulo 6 lista as referências bibliográficas utilizadas para o desenvolvimento da pesquisa. Por fim, o Anexo traz informações adicionais referentes à programação das ferramentas de automação de testes.



## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 O QUE É ROBOTIC PROCESS AUTOMATION?

Os primeiros passos para o desenvolvimento do conceito de RPA foram dados no início da década de 90 (ANTAS, 2019). Anteriormente havia algumas tecnologias denotadas BPA (do inglês - *Business Process Automation*), as quais foram a base para o desenvolvimento do novo conceito de automação. O RPA começou como uma forma de reduzir custos em terceirização de processos de negócios e passou para execução de serviços compartilhados, terceirização de TI e outras áreas de negócios. Já o termo RPA, propriamente dito, tem sua origem no início dos anos 2000, quando a Blue Prism lançou seu primeiro produto (em 2003), na mesma época em que a UiPath e Automation Anywhere lançaram suas bibliotecas de automação (todas as empresas foram fundadas um pouco antes).

De acordo com (IEEE CORPORATE ADVISORY GROUP, 2017) RPA pode ser definida como:

“Uma instância de *software* pré-configurada que usa regras de negócios e coreografia de atividades predefinidas para completar a execução autônoma de uma combinação de processos, atividades, transações e tarefas em um ou mais sistemas de *software* não relacionados para entregar um resultado ou serviço com gerenciamento de exceção humana.”.

A tecnologia RPA também pode ser descrita como uma abordagem de automação de processos que se utiliza de robôs de *software* para replicar tarefas humanas. Os robôs virtuais são integrados ao *software* existente e podem executar tarefas repetitivas em vários sistemas ao mesmo tempo. Os robôs são extremamente escalonáveis pois podem lidar facilmente com o aumento de volume, resultando em economia de custos. RPA também garante precisão e consistência das tarefas executadas (GEYER-KLINGEBERG *et al.*, 2018).

O objetivo da tecnologia RPA é realizar a modelagem de um robô que executará de forma automática tarefas repetitivas e baseadas em regras, que foram desenvolvidas anteriormente por uma pessoa (desenvolvedor). O robô funciona no nível do usuário, na interface gráfica principal, como se fosse um usuário humano, com suas permissões e seus acessos, igual a um ser humano. Como a ferramenta pode substituir a atividade humana em tarefas repetitivas, estruturadas e cansativas, ela permitirá com que o funcionário tenha mais liberdade e tempo para desenvolver trabalhos mais importantes para os quais sem ela não disporia de tempo hábil (AMMATTIKORKEAKOULU, 2018).

A tecnologia de RPA pode ser dita como de rápido crescimento. Os *softwares* de RPA contêm diversas técnicas úteis e avançadas, dentre as quais merece um destaque especial o OCR (do inglês – *Optical Character Recognition*), que é um recurso de reconhecimento de caracteres em sites, documentos ou imagens, possibilitando a alteração de documentos sem a necessidade de interação humana. Hoje cerca de 70% a 80% das tarefas de TI (Tecnologia da informação) podem ser automatizadas com RPA. Essa tecnologia está crescendo e sendo aplicada em diferentes tipos de companhias e departamentos como seguradoras, centrais médicas, recursos humanos e departamento financeiro (ANTAS, 2019).

## 2.2 FUNCIONALIDADES DAS FERRAMENTAS DE RPA

Usualmente, a tecnologia RPA é empregada para realizar (YATSKIV *et al.*, 2019):

- Tarefas repetitivas, que são executadas regularmente e frequentemente, uma vez que qualquer nível de automação aumenta a eficiência desse tipo de processo;
- Tarefas que requerem o processamento de uma quantidade substancial de dados, pois com o aumento da carga de informação, as pessoas tendem a cometer mais erros, enquanto o robô continua a trabalhar de forma estável;
- Tarefas que são explicitamente descritas por regras estritas de negócio que não requerem uma pessoa na tomada de decisão, nenhum julgamento, acordo, ou uso de dados externos. Por definição, o robô é capaz de tomar decisões em relação ao andamento do processamento desde que as regras sejam fixas e estejam estritamente definidas;
- Tarefas que requerem comunicação com pelo menos um ou mais sistemas eletrônicos, realizando, por exemplo, transferência dados de um sistema para outro.

Além disso, o uso RPA tem se tornado mais frequente em diversos cenários como (VAN DER AALST; BICHLER; HEINZL, 2018):

- **Logística** - (SULLIVAN; SIMPSON; LI, 2021);
- **Recursos humanos** - (PAPAGEORGIOU, 2018) (SCHEPPLER; WEBER, 2020);
- **Departamento financeiro e contábil** - (KABLAN, 2020; MODI *et al.*, 2021; OSHRI; PLUGGE, 2021; SMEETS; ERHARD; KAUSSLER, 2021)(AMINI, 2019);
- **Auditoria** - (HUANG, 2013; MOFFITT; ROZARIO; VASARHELYI, 2018; ROZARIO, 2019);

- **Setor bancário** - (KARIPPUR, 2018; ROMAO; COSTA; COSTA, 2019);
- **Hospitais e centros médicos** - (BHATNAGAR, 2020)(RATIA; MYLLÄRNIEMI; HELANDER, 2018)(SCHMIDER *et al.*, 2018);
- **Seguradoras** - (LAMBERTON; BRIGO; HOY, 2017);
- **Gerenciamento da cadeia de suprimentos** (do inglês - *supply chain*) - (MARUTI TECHLABS, 2021)(MUURINEN, 2019);
- **Automação de testes** - (YATSKIV *et al.*, 2019).

### 2.3 BENEFÍCIOS DAS FERRAMENTAS DE RPA

É importante mencionar que RPA consiste em uma tecnologia que automatiza tarefas pré-existentes não requerendo nenhuma mudança no funcionamento do sistema, ou seja, consiste apenas no treinamento do *software* para executar tarefas iterativas que seriam realizadas por humanos (ANSARI *et al.*, 2019). Apesar dos benefícios óbvios da automação de processos como redução de tempo e eficiência, a introdução do RPA elimina o fator de erro humano garantindo resultados mais homogêneos e permitindo escalabilidade, ou seja, uma vez que o processo certo for identificado e modificado, o mesmo conceito pode ser implementado em grande escala e o resultado será similar (ANAGNOSTE, 2017).

Alguns benefícios das ferramentas RPA são melhor detalhados a seguir (MADAKAM; HOLMUKHE; KUMAR JAISWAL, 2019; MATSUSHITA, 2019; SANTOS, 2019a):

- **Operam 24 horas, 7 dias por semana** - Diferentemente dos humanos, os robôs de RPA podem operar ininterruptamente, mais rapidamente, não precisando interromper as execuções para descanso e com 100% de confiabilidade e precisão (AUTOMATION ANYWHERE, 2021a). Uma vez programados, os robôs operam de forma autônoma, sem exigir que os funcionários os reativem, permitindo um agendamento mais flexível para os processos de negócios;
- **Alta escalabilidade** - A escalabilidade basicamente é a capacidade da empresa crescer e aumentar o número de operações sem elevar proporcionalmente os custos, atendendo demandas flexíveis. Com o RPA, é possível alterar o número de atividades dinamicamente em busca de um objetivo específico, sem a necessidade de contratar novos funcionários ou reduzir pessoal (tendo que lidar com questões trabalhistas), pois os robôs são capazes de executar tarefas repetitivas no lugar dos seres humanos. Além disso, com menos falhas, os gestores podem aumentar o número de operações e manter a qualidade dos produtos e, caso

os processos sejam padronizados, é possível expandir a automação para outras filiais;

- **Diminuição de erros de execução e consequentemente aumento da qualidade** - Com a diminuição de erros de digitação, erros derivados do cansaço, desatenção do trabalhador humano, há um aumento na qualidade do produto final. Além disso há um aumento de precisão e consistência pois os robôs são muito menos propensos a cometer erros do que um trabalhador humano o que se aplica a processos comuns, como configuração ou remoção de contas de usuário, cópia de informações de um sistema para outro, integração e desativação de funcionários ou preenchimento de formulários com base em informações de outro sistema;
- **Melhoria dos padrões de conformidade regulamentares** - Os robôs seguem as instruções configuradas ou decorrente de *Machine Learning* e fornecem um histórico de trilha de auditoria para cada etapa. Sua natureza controlada torna a automação de processos robóticos bem adequada para organizações que desejam automatizar seus processos de negócios de ponta a ponta e, ao mesmo tempo, atender aos rígidos padrões de conformidade;
- **Aumento da eficiência do processo** - Ao utilizar o RPA como parte de seu kit de automação, as organizações podem abrir mais oportunidades para os funcionários dedicarem mais tempo e talento a outro trabalho. Enquanto os robôs preenchem formulários, inserem dados e procuram informações em sites, os funcionários podem se concentrar em projetos de alto impacto que impulsionam ainda mais o crescimento da companhia;
- **Aumento da produtividade** - Um robô de *software* tem capacidade de trabalho maior do que o humano, consequentemente, os tempos de ciclo do processo são mais eficientes e podem ser concluídos em uma velocidade mais rápida em comparação com os processos manuais (ALBERTH; MATTERN, 2017). Estima-se que um robô de *software* consiga aumentar a capacidade de trabalho de 3 a 5 vezes mais que um Full-Time Employee (FTE) (MARTINS, 2018);
- **Barreiras técnicas mínimas:** Há diversas ferramentas que não demandam de habilidades de programação para configurar um robô de RPA. A equipe não técnica pode usar um recurso de gravador de processo para ensinar o robô a executar uma etapa anteriormente manual em um processo de negócios automatizado e integrar a automação baseada em robô em um fluxo de trabalho

automatizado maior usando um designer de processo de arrastar e soltar (do inglês - *drag-and-drop*) ou ferramenta similar;

- **Facilidade de configuração** - As ferramentas RPA são fáceis de configurar. As interfaces RPA funcionam simplesmente arrastando e soltando etapas em um processo. Conforme os usuários arrastam e soltam ícones para automatizar um processo, um código é gerado internamente de maneira automática;
- **Software não invasivo** - O *software* RPA não é invasivo, assenta sobre os sistemas existentes sem a necessidade de criar, substituir ou desenvolver plataformas dispendiosas, ao contrário das integrações tradicionais que podem exigir recursos extensivos do desenvolvedor;
- **Interação semelhante à humana** - A ferramenta RPA acessa outros sistemas de computador da mesma forma que um humano (por meio da interface do usuário), portanto, nenhum *software* é afetado pelo próprio RPA. Os robôs funcionam diretamente a partir de uma interface de usuário, assim como uma pessoa faria. Isso torna a automação de processos robóticos especialmente útil para a integração de sistemas legados, onde as APIs podem não estar imediatamente disponíveis ou em situações em que as organizações não têm tempo ou recursos de funcionários para desenvolver integrações baseadas em código personalizado;
- **Segurança** - O RPA é seguro para empresas pois é uma plataforma robusta construída para atender aos requisitos corporativos de segurança, escalabilidade e auditabilidade. O orquestrador RPA fornece robôs a serem programados em um trabalho contendo múltiplos processos, também fornece auditoria e monitoramento de robôs para um processamento centralizado e sem problemas.
- **Agilidade operacional** - Robôs podem ser “treinados” rapidamente e, portanto, podem responder muito mais rápido às mudanças de requisitos e necessidades de negócios (VISHNU; AGOCHIYA; PALKAR, 2017);
- **Planejamento e previsão de negócios** – A automação pode fazer a coleta, organização e análise de dados de forma mais ágil, ajudando a empresa a se planejar para demandas futuras, tendências e oportunidades (VISHNU; AGOCHIYA; PALKAR, 2017);
- **Eficiência operacional** - a redução de tempo, custo e recursos humanos, redução de tarefas manuais e carga de trabalho e aumento da produtividade e redução do custo operacional reduzido estão no topo da lista de benefícios do RPA. Com base

em medidas quantificáveis, como o número de funcionários equivalentes em tempo integral (do inglês - *Full time Employee* - FTEs) substituídos por robôs, a tecnologia RPA provou cortar o custo dos gastos relacionados com recursos humanos em 20-50%, e pode reduzir o custo de processamento de transações em 30-60%. A redução das tarefas manuais e a redução da carga de trabalho também levaram a eficiências de tempo, conforme evidenciado pela redução significativa (de 30% a 70%) no tempo de ciclo do processo, tempo de manuseio de tarefas, tempo de espera, e assim por diante. O aumento da produtividade é decorrente, em primeiro lugar, do fato de que os robôs podem trabalhar 24 horas por dia, 7 dias por semana, e, em segundo lugar, por liberar recursos humanos de tarefas repetitivas e tediosas e, como resultado participar de atividades de maior valor agregado (SYED *et al.*, 2020);

- **Retorno sobre investimento** (do inglês - *Return over Investment* - ROI) - Os especialistas em RPA prometem um retorno do investimento de até 300% no primeiro ano (THE ROBOTIC WORKFORCE, 2021), mas o cenário pode ser diferente para as empresas que utilizam o RPA como automação assistida, como ferramenta de aceleração do trabalho humano, como robô sem supervisão ou híbrido. O que vale a pena mencionar é que esse retorno advém de um conjunto de benefícios listados anteriormente, como a redução dos custos operacionais, melhorias no processo que executa mais tarefas em menos tempo e é mais flexível, aumento da produtividade, conformidade, qualidade, consistência e permite reimplantação de recursos (humanos ou físicos) em funções de maior valor (ALBERTH; MATTERN, 2017).

## 2.4 DESAFIOS DAS FERRAMENTAS DE RPA

Apesar de a ferramenta de RPA ser mais intuitiva ainda temos alguns desafios na sua implementação na empresa, como:

- **Manutenção dos Robôs** - é preciso manter os robôs de *software* atualizados, isso porque as aplicações sofrem constante atualizações que podem quebrar os processos automatizados anteriormente;
- **Custo da licença quando comparado às ferramentas gratuitas** - a ferramenta de RPA tem um custo, algo em torno de 13.000 dólares por ano (MARTINS, 2018), em contrapartida há diversas ferramentas de automação de testes gratuitas;

- **Limitação da ferramenta** - muitos casos são dificultados pela incapacidade da ferramenta de lidar com formatos não estruturados de forma confiável. O reconhecimento óptico de caracteres está ajudando a melhorar esses casos, junto com Inteligência Artificial e aprendizado de máquina, mas ainda apresentam desafios na ferramenta, como exemplo podemos citar documentos manuscritos. (MULLAKARA, 2021c);
- **Escolha correta da ferramenta de RPA** - A mudança de um sistema de negócios tradicional para uma forma mais avançada, apesar de parecer intuitiva, pode ser menos produtiva se as ferramentas corretas não forem empregadas (ANSARI *et al.*, 2019);
- **Limitações e restrições** - As ferramentas de RPA necessitam de adequação da infraestrutura de TI para alcançar todas as vantagens. Uma crítica comum é que as interações oferecidas pelo RPA são menos robustas do que aquelas que são incorporadas por natureza nos sistemas centrais. Outras limitações identificadas incluem interrupções do *business as usual*, ansiedade dos funcionários, falta de benefícios tangíveis, dificuldades de implementação, incentivo reduzido para otimizações adicionais e crença equivocada de que os sistemas existentes podem lidar com o aumento frequente (SYED *et al.*, 2020);
- **Métricas mais abrangentes** - A forma de medição dos benefícios entregues por uma solução RPA precisa ser reconsiderada, pois normalmente, os benefícios são medidos em termos de reduções de tempo, custo, erro e recursos humanos. Contudo, os benefícios não se limitam a esses resultados diretos e tangíveis. Por exemplo, a retirada de tarefas repetitivas dos funcionários, deixando-os livres para realizar tarefas mais criativas pode levar ao aumento da produtividade e, nesse sentido, a medição de produtividade também deve ser incluída nos benefícios do RPA (SYED *et al.*, 2020);
- **Medidas de contenção** - As ferramentas de RPA exigem suporte cognitivo. O sistema deve ser implementado para eventualmente permitir o aprendizado de máquina, de modo que os robôs possam aprender e tornarem-se mais eficientes com o tempo. Contudo, os robôs têm a capacidade de aprender “maus comportamentos”, pois não têm capacidade de distinção integrada. Consequentemente, é preciso ativar uma proteção e conhecimento do domínio

para mitigar o risco de agravar o comportamento incorreto (VISHNU; AGOCHIYA; PALKAR, 2017).

## 2.5 PRINCIPAIS FERRAMENTAS RPA DISPONÍVEIS NO MERCADO

Atualmente, existem mais de 50 ferramentas RPA no mercado, além de soluções customizadas de diferentes empresas de consultoria como KPMG, IBM e Delloite. Nesta seção é apresentada uma visão geral das 10 principais ferramentas de RPA disponíveis, as quais são listadas na Tabela 1. Ao final da seção, será dado maior destaque às empresas líderes do mercado (segundo o quadrante de Gartner) e, em seguida, será apresentada uma avaliação comparativa realizada por (MULLAKARA, 2021a), com base na qual justifica-se a escolha da ferramenta RPA empregada neste trabalho.

Tabela 1: 10 Principais ferramentas RPA disponíveis no mercado

Número	Ferramenta
1	Automation Anywhere
2	Microsoft Power Automate
3	Blue Prism
4	Workfusion
5	Pega Systems
6	EdgeVerve
7	Nice
8	Kryon
9	Kofax
10	UiPath

Fonte: Autoria própria

### 2.5.1 *Automation Anywhere (AA)*

A AA é um dos principais provedores de serviços entre as ferramentas RPA. Há 3 módulos dentro da AA, *Bot Creator*, *Control room* e *Bot Runner*. O *Bot Creator* permite o *design* simples e o processo de automação dos robôs. A *Control room* gerencia a execução e programação de robôs, disponibiliza e gerencia credenciais dos robôs, controla possíveis problemas de segurança, permissões e avaliações dos clientes. O *Bot Runner* é usado para executar os robôs e enviar os dados da execução para a *Control room*. A AA disponibiliza três tipos diferentes de robôs chamados: *Task Bot*, *Meta Bot* e *IQ bot*. Os *Task Bots* são robôs utilizados para automatizar tarefas repetitivas, já os *Meta Bots*, são robôs utilizados em blocos de construção, podendo assim serem reutilizados em outros *Task Bots*. Por fim, os *IQ Bots* são robôs equipados com características cognitivas e inteligentes para processamentos de dados não estruturados (KHAN, 2020). A AA é uma ferramenta bem completa, seus recursos são amigáveis aos usuários, tem alta escalabilidade, robustez e uma *bot store*(loja de robôs). Como



desvantagem sua licença apresenta alto custo em comparação com outras ferramentas similares (MULLAKARA, 2021b).

### **2.5.2 Microsoft Power Automate**

A Microsoft é uma das empresas que adentraram ao mundo RPA. Na aquisição da Softomotive, a Microsoft expandiu sua oferta com WinAutomation, ProcessRobot e Robin. O novo pacote da Microsoft foi rebatizado como Microsoft Power Automate (MULLAKARA, 2021b). A ferramenta WinAutomation oferece um conjunto de recursos associados aos processos de automação como, automatização de e-mails, arquivos em vários formatos (PDF e Excel), OCR e outras funcionalidades associadas ao ambiente de trabalho dos funcionários. Além disso, a WinAutomation é voltada para ambientes de *desktop* que possuem processos integrados, *design*, automação de *desktop*, automação WEB, gravação de macro, multitarefa, execução automática de tarefas, automação de teclado, *designer* de interface de usuário, automação de e-mail, planilhas do EXCEL, arquivos e pastas, monitoramento e acionamento, login automático, recursos de segurança embutidos entre outras funções (MICROSOFT POWER AUTOMATE, 2021; RIBEIRO *et al.*, 2021).

### **2.5.3 BluePrism**

A BluePrism é uma das empresas pioneiras no mundo da RPA, disponibilizando uma das ferramentas mais robustas e que tem uma arquitetura orientada a objetos, a qual ajuda a gerenciar melhor os componentes reutilizáveis. Como ponto negativo, ela não possui o botão gravador que facilita os novos desenvolvedores a automatizar com mais rapidez (MULLAKARA, 2021b). A BluePrism é baseada em Java e *.NET framework* e fornece funcionalidade de arrastar e soltar para o desenvolvimento dos fluxos dos robôs. Há 4 componentes na ferramenta: *Process Diagram*, *Process Studio*, *Object Studio* e *Application Modeller*. O *Process Diagram* é constituído por fluxos de trabalho desenvolvidos utilizando os principais conceitos de programação. Essas representações gráficas de fluxos de trabalho são usadas para criar, analisar, modificar e dimensionar recursos de negócios. *Process Studio*, por sua vez, disponibiliza uma plataforma para criar diagramas de processo com várias atividades de arrastar e soltar. O *Object Studio* é usado para criar elementos visuais, objetos básicos que são usados para se comunicar com outros aplicativos. O *Application Modeller* é a funcionalidade usada para criar modelos de aplicativos com *Object Studio* (KHAN, 2020).

### **2.5.4 Workfusion**

O Workfusion é uma plataforma de automação que integra IA e aprendizado de máquina nativamente. O WorkFusion oferece recursos necessários para automatizar várias funções em

uma empresa, integrando várias funcionalidades em uma mesma plataforma, substituindo o trabalho manual pela automação cognitiva e, conseqüentemente, economizando e aumentando a capacidade da empresa (WORKFUSION, 2021). WorkFusion desenvolveu sua ferramenta de RPA com base em ML. Sediada em Nova York, a empresa começou a se envolver com RPA em 2014, tendo como principal foco o autoatendimento com uso de IA para as empresas. A ferramenta oferece algoritmos pré-carregados para extração de dados, classificação e tomadas de decisão não estruturadas (LE CLAIR, 2018).

### 2.5.5 *PegaSystems*

A *PegaSystems* é uma provedora de gerenciamentos de processos de negócios (BPM) que adicionou o RPA com a aquisição da *Openspan*. A *PegaSystem* usa recursos RPA para diminuir a carga de trabalho sobre seus funcionários. Com a automação das tarefas rotineiras, há um aumento no tempo livre e, conseqüentemente, um aumento na produtividade deles e satisfação dos funcionários (MULLAKARA, 2021b). Como muitas vezes as empresas não entendem como a ferramenta de RPA se adaptaria aos processos, a estratégia da *Pegasystems* é direta: forneça valor rápido com RPA com o mínimo de arquitetura, seguindo com o *Workforce Intelligence* (WFI), que pode levar ao BPM, gerenciamentos de casos, aplicativos de atendimento ao cliente ou análises mais avançadas. WFI utiliza robôs que prestam atenção nas atividades do *desktop* e transmite os resultados para um ambiente com ML hospedado no *Amazon Web Services* (AWS). Os relatórios gerados no formato de mapa de calor podem orientar a automação. O *Bot Design* tem uma interface gráfica adequada para o analista de negócios, mas requer conhecimento de programação, como *loops*, para projetar um robô. A ferramenta conta com análise para automação, disponibilizando OCR, visão computacional e IA (LE CLAIR, 2018).

### 2.5.6 *EdgeVerve*

A *EdgeVerve* é uma subsidiária da gigante indiana Infosys Technologies, que tem como seu principal produto a *AssistEdge*. A *EdgeVerve* apresenta uma combinação de serviços com estruturas bem definidas, governança RPA e descoberta de processos. A *AssistEdge* inclui RPA, descoberta de processos, automação de testes e modelos de processos específicos da indústria (MULLAKARA, 2021b). Em janeiro de 2013, a *EdgeVerve* lançou o produto *AssistEdge* e em 2018 alcançou mais de 200 clientes espalhados entre bancos, empresas de manufatura, varejo, telecomunicações e serviços de utilidade pública. *EdgeVerve* combina experiência em serviços com recursos de plataforma. *AssistEdge* possui recursos de arrastar e soltar e gravador de fluxos que os usuários podem utilizar para gravar a área de trabalho e

transpor os resultados para um script de automação. A EdgeVerve está se adaptando rapidamente para se desligar dos serviços da Infosys (LE CLAIR, 2018).

### 2.5.7 Nice

A Nice é uma empresa de origem israelense de *software* empresarial, sendo uma das maiores organizações de tecnologia em Israel. A Nice começou com a tecnologia RPA após a aquisição da E-glue em 2010 (MULLAKARA, 2021b). A Nice abrange robôs assistidos e autônomos, mas a maioria dos clientes valorizam a profundidade de seu departamento de *contact center*. A solução tem como ponto forte sua central de gerenciamento que supervisiona o enfileiramento dos processos e o sistema como um todo. A ferramenta atende a padrões de segurança rígidos e possui um recurso que serve para observar a conectividade, monitorando a Interface do Usuário e captando alterações nos aplicativos. Além disso, a Nice adicionou *Automation Finder*, uma ferramenta que identifica oportunidades de automação usando análise de *desktop* e aprendizado de máquina. Outro recurso importante é a gravação inteligente que está disponível. A Nice tem um amplo portfólio de análises, incluindo um *chatbot* e uma parceria com a *Celaton* para processamento de faturas. Nice é uma escolha segura, mas recomendável especialmente para casos de uso com robôs assistidos (LE CLAIR, 2018).

### 2.5.8 Kryon

A Kryon nasceu em 2008 e possui ferramentas para minerar processos dentro da organização, destacando os processos que podem ser automatizados. Possui integração com ABBY, que é líder em OCR é uma tecnologia para reconhecer caracteres a partir de um arquivo, imagem ou mapa de bits. Kryon usa visão computacional para dar suporte ao Citrix (*software* que integra recursos dos formatos XenApp e XenDesktop os quais visam utilizar aplicações em sistemas de uma forma virtualizada), uma área pouca atendida pelas ferramentas de RPA (KRYON SYSTEMS, 2021). O seu recurso interno de *Bot Design* usa várias linguagens de script vistas como plug-ins, bancos de dados e sistema de arquivos. A visualização do processo baseada em ícones é usada apenas para monitoração do processo, não para o *design*. Sua boa eficiência na implementação dos robôs com gerenciamento inteligente de fila e suporte híbrido (combinando suporte assistido e não assistido) são os pontos fortes do produto. Possui uma arquitetura *Microsoft-based* com requisitos de infraestrutura modestos e requisitos limitados para produtos de terceiros (por exemplo, controle de versão, relatórios ou gerenciamento de credenciais). A ferramenta utiliza um sistema de visão computacional patentado para localização de aplicativo, suporte para sessões Citrix e gravação do *desktop*. A Kryon é uma

solução forte e completa com um bom preço em relação às plataformas que lideram o mercado (LE CLAIR, 2018).

### **2.5.9 Kofax**

A Kofax tem diversos produtos, sendo um deles a Kapow RPA, que foi adquirida pela empresa em 2013, para atender o mercado de RPA. A Kofax tem sido utilizada para aprimorar a inteligência do gerenciamento de conteúdo (MULLAKARA, 2021b). Kofax é uma empresa que desenvolve *softwares* de automação de processos em empresas e organizações. A ferramenta disponibiliza diversos módulos orientados para RPA, orquestração de processos de negócios por meio de fluxos procedimentais de atividades de *software*, reconhecimento de documentos através de OCR e análise avançada de dados. A Kofax não disponibiliza licença de estudante para estudos e aprendizado da ferramenta. Essa ferramenta tem módulos associados à implementação de técnicas e/ou algoritmos de IA, além de suporte ao aprendizado de máquina (RIBEIRO *et al.*, 2021).

### **2.5.10 UiPath**

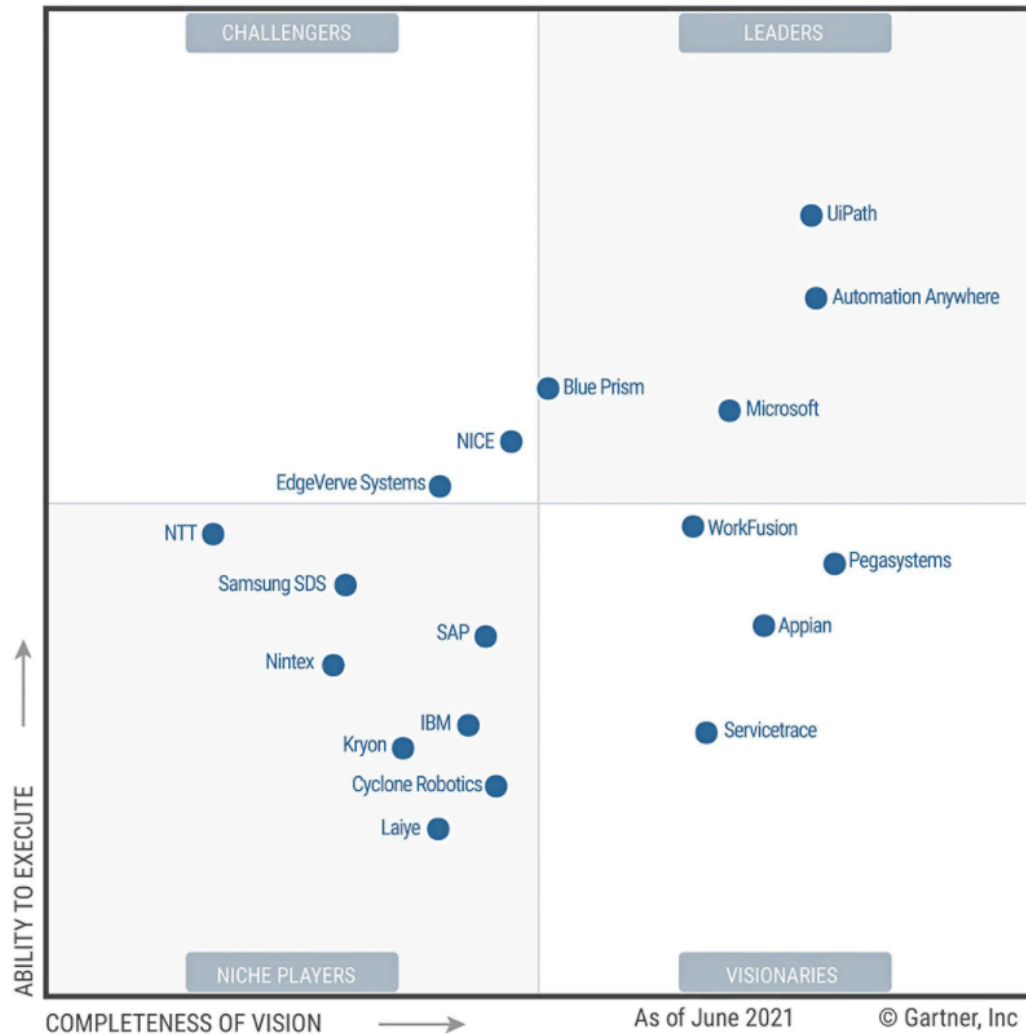
UiPath é uma empresa de *software* global que fornece uma plataforma para desenvolver robôs de *software* para automatizar processos e casos de testes. UiPath possui uma arquitetura baseada em .NET framework. Os principais componentes da ferramenta UiPath são: UiPath studio, UiPath Orchestrator e UiPath Robots. O UiPath Studio disponibiliza diversos espaços de trabalho para projetar e executar os robôs, inclusive Test Suite que é um módulo que permite automatizar casos de testes. Por ser uma ferramenta com abordagem de arrastar e soltar, torna-se mais fácil de usar. O orquestrador permite que o usuário carregue um robô na nuvem e gerencie seus recursos. Há robôs autônomos e assistidos. Os robôs assistidos podem precisar de intervenção humana para finalizar suas tarefas, enquanto o robô autônomo trabalha de forma independente (KHAN, 2020).

### **2.5.11 Quadrante mágico Gartner e escolha da ferramenta RPA**

Tendo em vista a breve discussão apresentada a respeito de cada ferramenta, surge uma questão: como selecionar a que melhor se adequa aos objetivos da empresa?

Na Figura 1 é apresentado o quadrante mágico Gartner (relatório anual disponibilizado pelo Grupo Gartner que dá um panorama das empresas que compõem o mercado tecnológico, apontando líderes, desafiantes, visionários e competidores de nicho) com as principais ferramentas de RPA no ano de 2021.

Figura 1: Quadrante Mágico Gartner 2021 para Robotic Process Automation



Source: Gartner (July 2021)

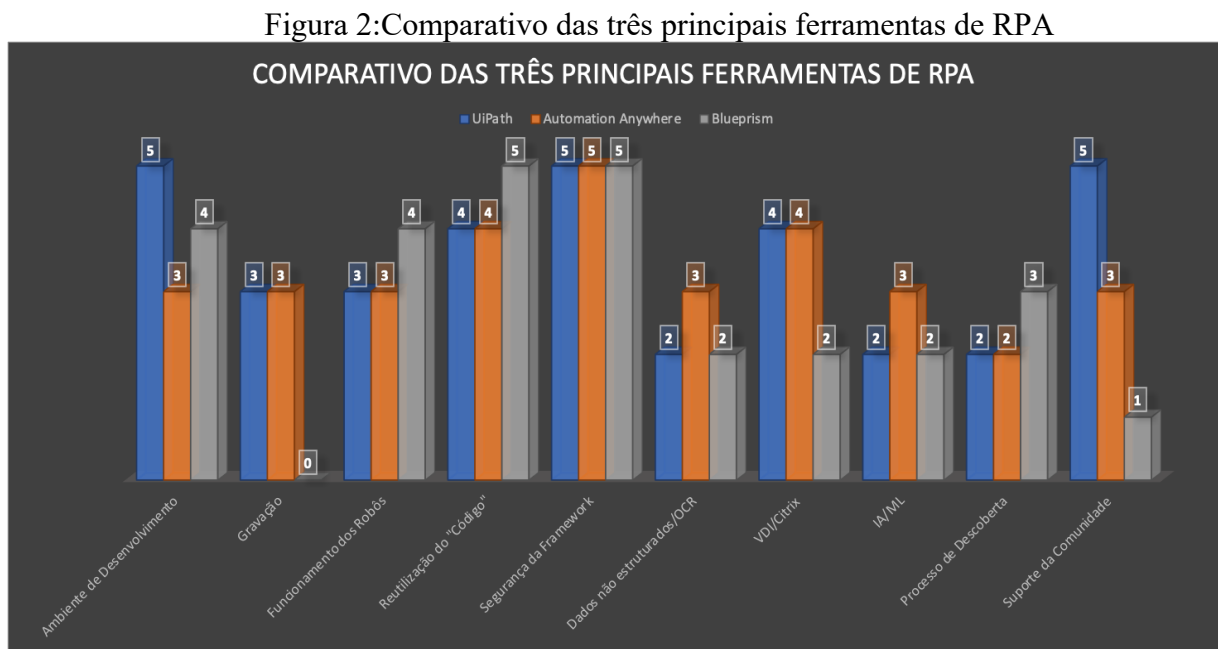
Fonte:(AUTOMATION ANYWHERE, 2021b)

Observe que no quadrante de líderes do mercado destacam-se 4 ferramentas: UiPath, Automation Anywhere, BluePrism e Microsoft. Como apenas as três primeiras são ferramentas especializadas em automação de processos robóticos, em (MULLAKARA, 2021a), essas ferramentas foram comparadas atribuindo-se notas de 1 a 5 para os seguintes critérios:

- 1 - Ambiente de desenvolvimento;
- 2 - Recurso de gravação que a ferramenta disponibiliza;
- 3 - Funcionamento dos robôs;
- 4 - Reutilização dos componentes (código);
- 5 - Segurança;

- 6 - Automação com dados não estruturados/OCR;
- 7 - Utilização de VDI/Citrix,
- 8 - Utilização de Inteligência Artificial e Aprendizado de Máquinas;
- 9 - Processo de descoberta;
- 10 - Suporte da comunidade.

Essa comparação pode ser melhor visualizada no gráfico da Figura 2.



Fonte:(MULLAKARA, 2021a)

Observe que, dentre os 10 critérios avaliados, a ferramenta UiPath destaca-se com as maiores notas em 5 critérios e soma uma nota total de 35 pontos, contra 33 alcançados pela Automation Anywhere e 28 obtidos pela BluePrism. Além de estar bem avaliada em todos os critérios (recebendo nota máxima em 3 deles), o principal motivador para a escolha da UiPath, foi o seu grande suporte online e sua função gravador integrada que permitem que novos usuários se sintam amplamente amparados no emprego da ferramenta.

É importante mencionar que, atualmente, a UiPath conta com mais de 750.000 usuários pelo mundo, é a numero 3 na Forbes Cloud 100 e 8 das 10 primeiras empresas listadas na Fortune são clientes (UIPATH, 2021).

A UiPath foi a ferramenta utilizada neste trabalho. Com aprendizado fácil e recursos como IA, *machine learning*, *process mining* gratuitos para estudantes, com cursos na sua

plataforma ensinando a utilizar a ferramenta, a UiPath atrai muitos usuários. Adicionalmente, essa ferramenta RPA apresenta um recurso integrado chamado Test Suite totalmente voltado para automação de testes, algo pouco comum nas ferramentas de RPA. Test Suite dá suporte à automação de testes WEB e *mobile* e, por ser uma ferramenta de fácil aprendizado, tem sido muito empregada.

## 2.6 AUTOMAÇÃO DE TESTES

A automação de testes é o uso de *softwares* específicos que automatizam tarefas repetitivas e controlam a execução dos casos de testes automatizados (HIMANSHI; UMESH; SARASWAT, 2015). A automação de testes tornou-se um requisito para a grande maioria dos projetos de softwares. Diversas ferramentas de automação de testes oferecem um recurso que gravam os passos executados nos testes para ajudar o desenvolvedor a automatizar as tarefas (PATIL; TEMKAR, 2017). Em contrapartida, a dependência desses recursos apresenta grandes problemas, como confiabilidade e facilidade de manutenção do código. Os testes sempre precisarão ser regravados quando sofrerem pequenas alterações na aplicação. O uso deste recurso adiciona atividades não desejadas ou pode inclusive gravar de forma errônea algumas atividades (HIMANSHI; UMESH; SARASWAT, 2015).

Automatizar os testes tem vantagens específicas que permitem uma melhoria a longo prazo dos processos de teste, oferecendo (HIMANSHI; UMESH; SARASWAT, 2015):

- Execução frequente de testes de regressão que garantem a qualidade contínua da aplicação;
- Realimentação (*feedback*) contínuo da aplicação, mantendo o desenvolvedor informado a respeito do status atual da mesma;
- Documentação dos casos de testes automatizados;
- Relatórios personalizados dos defeitos encontrados nos testes automatizados;
- Possibilidade de encontrar defeitos negligenciados ou decorrentes de erros humanos nos testes manuais.

### 2.6.1 Principais ferramentas de automação de testes disponíveis no mercado

Assim como foi feito na Seção 2.5, nesta seção será apresentada uma breve introdução às 10 ferramentas de automação de testes mais conhecidas do mercado, as quais são listadas na Tabela 2.

Tabela 2: 10 Principais ferramentas de automação de testes disponíveis no mercado

Número	Ferramentas
1	Selenium

2	TestComplete
3	Robotium
4	Ranorex
5	Robot Framework
6	Cypress
7	Appium
8	UFT
9	Test Studio
10	Capybara

Fonte: Autoria própria

### 2.6.2 Selenium

Selenium é o *software* de automação de testes mais empregado no mercado (PRIME CONTROL, 2021). O Selenium é um conjunto de ferramentas de automação de testes de *software* de código aberto (ALTEXSOFT, 2021) e oferece suporte a várias linguagens de programação e *frameworks* de automação de testes. Algumas das linguagens suportadas incluem Java, Python, C#, PHP, Javascript, Ruby e Perl (SOFTWARE TESTING HELP, 2021). A ferramenta pode ser executada em todas as três plataformas de *desktop* principais: Windows, Mac e Linux. O Selenium IDE é um *add-on* (*plug-in* para navegadores), que permite gravar e reproduzir, facilitando o aprendizado da ferramenta. Na maioria das vezes quando se fala de Selenium, o usuário está se referindo ao Selenium WebDriver, pois a maior parte do desenvolvimento da ferramenta é focada nesse produto. A maneira como a automação de teste funciona no Selenium WebDriver é frequentemente comparada à condução de um táxi. Existem três participantes tanto na condução de um táxi quanto na automação de teste: cliente / engenheiro de teste ou testador, carro / navegador, motorista de táxi / WebDriver. Conforme a analogia, um testador comanda o WebDriver para interagir com os elementos da aplicação de forma equivalente a um cliente dando instruções a um motorista de táxi. Posteriormente, o WebDriver fornece ao navegador (o carro) instruções como: quando o botão for clicável, clique no botão. Logo após, o navegador fornece informações do WebDriver sobre valores e status de elementos da web, que são posteriormente enviados para o script (ALTEXSOFT, 2021). Por ser a ferramenta número 1 das aplicações web e sua alta compatibilidade entre linguagens de programação e sistemas, ela foi escolhida para efeito de comparação com a ferramenta RPA aplicada neste trabalho.

### 2.6.3 TestComplete

*TestComplete* é um *framework* de testes automatizados funcional desenvolvido pela SmartBear Software. O *TestComplete* tem suporte para aplicativos Microsoft Windows, Web,



Android e iOS. É uma ferramenta paga e sua licença pode custar de U\$999,00 a U\$2.000,00. (CAETANO, 2021)

#### **2.6.4 Robotium**

O Robotium é um popular *framework* para automação de testes voltada para o sistema operacional Android que tem suporte para aplicações nativas e híbridas. Como essa ferramenta é baseada em um *framework* exclusivo para Android, demanda um alto nível de conhecimento de desenvolvimento nessa plataforma e, usualmente, não é escolhida quando o projeto depende de diversas tecnologias (PRIME CONTROL, 2021).

#### **2.6.5 Ranorex**

Mais de 4.000 empresas em todo o mundo usam o Ranorex Studio, que é uma ferramenta completa para teste de aplicativos *desktop*, WEB e móvel que possui uma interface *click-and-go* (sem código), mais amigável para iniciantes, mas poderosa para especialistas em automação com um IDE completo (SOFTWARE TESTING HELP, 2021).

#### **2.6.6 Robot Framework**

O Robot Framework é uma ferramenta *open source* (código aberto), com amplo propósito, muito utilizada para testes de aceitação, simplifica o desenvolvimento convencional e traz flexibilidade para os testes, além de ser muito utilizada com o Selenium como uma ferramenta de automação de testes para aplicações WEB (BISHT, 2021).

#### **2.6.7 Cypress**

Cypress é uma ferramenta de automação de teste *end-to-end* para UI usado em aplicações WEB. Os desenvolvedores podem facilmente sincronizar e resolver problemas com o Cypress, que é frequentemente comparada com o Selenium (TAKY, 2009).

#### **2.6.8 Appium**

Appium destina-se principalmente a aplicativos móveis. É uma ferramenta de código aberto. Com um amplo suporte à automação de aplicativos da web nativos, híbridos e móveis desenvolvidos para iOS e Android. Appium é uma ferramenta fácil de instalar e utilizar, que apresenta grande estabilidade e recentemente ganhou grande popularidade sendo considerada como uma das melhores ferramentas de teste de automação móvel (SOFTWARE TESTING HELP, 2021).

A Appium encontra suas raízes no Selenium WebDriver, que é o padrão para automação de navegador, usando-o para executar scripts. Os scripts podem ser escritos em Java, Ruby, C, Python, Perl, o que permite aos programadores a liberdade de escolher qual linguagem de programação será utilizada (SINGH À; GADGIL À; CHUDGOR À, 2014).

### **2.6.9 UFT**

UFT (do inglês - *Unified Functional Testing*), originalmente denominado QuickTest Professional, é uma ferramenta comercial da MicroFocus adquirida pela Hewlett Packard em 2006. Essa ferramenta permite que o usuário execute testes usando palavras-chave e script manual, permite realização de suporte a ferramentas de integração contínua (*CI*, do inglês – *Continuous Integration Tools*), fornece testes de Web, dispositivos móveis e API além de possuir uma função de reconhecimento de objeto inteligente (ALTEXSOFT, 2021).

### **2.6.10 Test Studio**

Telerik Test Studio é uma solução abrangente de automação de testes, adequada para testes de GUI (do inglês - *Graphical User Interface*), desempenho, carga e API. A ferramenta permite testes de aplicativos de *desktop*, móveis e da WEB. Seus principais recursos incluem um gravador de teste de apontar e clicar, suporte para linguagens de codificação reais como C # e VB.NET, um repositório central de objetos e integração contínua com o controle de origem (SOFTWARE TESTING HELP, 2021).

### **2.6.11 Capybara**

*Capybara* é um *framework*/biblioteca de testes de aceitação de código aberto para aplicações WEB escrita na linguagem de programação Ruby e que pode usar uma variedade de drivers de teste. A ferramenta constrói uma API mais abstrata do driver, onde os testes são escritos usando uma linguagem específica, gerenciando entre outras coisas JavaScript assíncrono. Ela foi desenvolvida para simulação e está limitada a realizar as mesmas ações que um usuário executaria em uma aplicação WEB. A *Capybara* vem sendo atualizada continuamente desde sua criação em 2009 (GRAPE, 2016).

## **2.7 TRABALHOS CORRELATOS**

Quando o assunto é RPA e automação de testes, a quantidade de trabalhos científicos publicados ainda é um pouco tímida. Por ser um assunto recente, há poucas pesquisas sobre o tema, no entanto, os principais trabalhos correlatos são listados a seguir. Em (SANTOS, 2019b), o objetivo do autor é testar se é viável usar uma ferramenta de automação de testes (Selenium) para automatizar os processos de negócios de uma empresa e identificar vantagens e desvantagens do uso de uma ferramenta de automação de teste como uma ferramenta RPA. A principal motivação do trabalho é financeira, ou seja, o autor visa encontrar alternativas de baixo custo às ferramentas de RPA para automatizar processos. O trabalho de (SANTOS, 2019a) pode ser considerado uma boa contraposição à presente dissertação que visa realizar

uma análise complementar, verificando as vantagens de se empregar uma ferramenta RPA para automação de testes.

Similarmente ao que está sendo proposto nesta dissertação, em (YATSKIV *et al.*, 2019), foi comparado o emprego de duas ferramentas: *Selenium Web Driver* e *Workfusion* (RPA) na automação de testes de software. Ao final, os autores constataram que a ferramenta *Selenium* demonstrou ter o mesmo tempo de execução para a maioria das operações, porém ela não oferece opções de geração de dados e a possibilidade de utilizar outros programas durante a execução dos testes, além disso demanda um conhecimento técnico na linguagem de programação utilizada. Como a maioria dos fluxos de trabalho demandam outras ferramentas ou *scripts*, os autores consideraram que o uso da ferramenta RPA é uma ótima opção e concluíram que a automação de RPA é mais eficaz quando o usuário precisa interagir com outras ferramentas ou aplicativos durante a execução.

Em (CERNAT; STAIUCU; STEFANESCU, 2020), foi feita uma análise do uso de ferramentas RPA em automação de testes para a interface do usuário. com o objetivo de chamar a atenção da comunidade acadêmica para o potencial das ferramentas RPA nesse cenário. Apesar de ser um trabalho inicial, os autores identificaram vários desafios a serem enfrentados e os caminhos que poderão seguir para tornar essa estratégia um caso de sucesso não só com o emprego da UiPath, mas também replicando o modelo para outros fornecedores RPA.

### 3 METODOLOGIA

Esta pesquisa utiliza como estratégia de investigação um estudo de caso de caráter quantitativo e exploratório, com objetivo de implementar a ferramenta RPA na automação de casos de testes em uma empresa multinacional. Para justificar a escolha do estudo de caso desse estudo, precisamos compreender quais são os tipos de estudo de casos. Segundo (YIN, 2009), o estudo de caso pode ter diversos propósitos, pode ser intrínseco (realizado devido ao seu interesse), instrumental (com um propósito em mente), avaliativo (para verificar se algo está funcionando bem), explicativo (busca uma explicação na forma de uma relação causal) e exploratório (através de um estudo empírico que busca responder questões quando não há conhecimento no assunto).

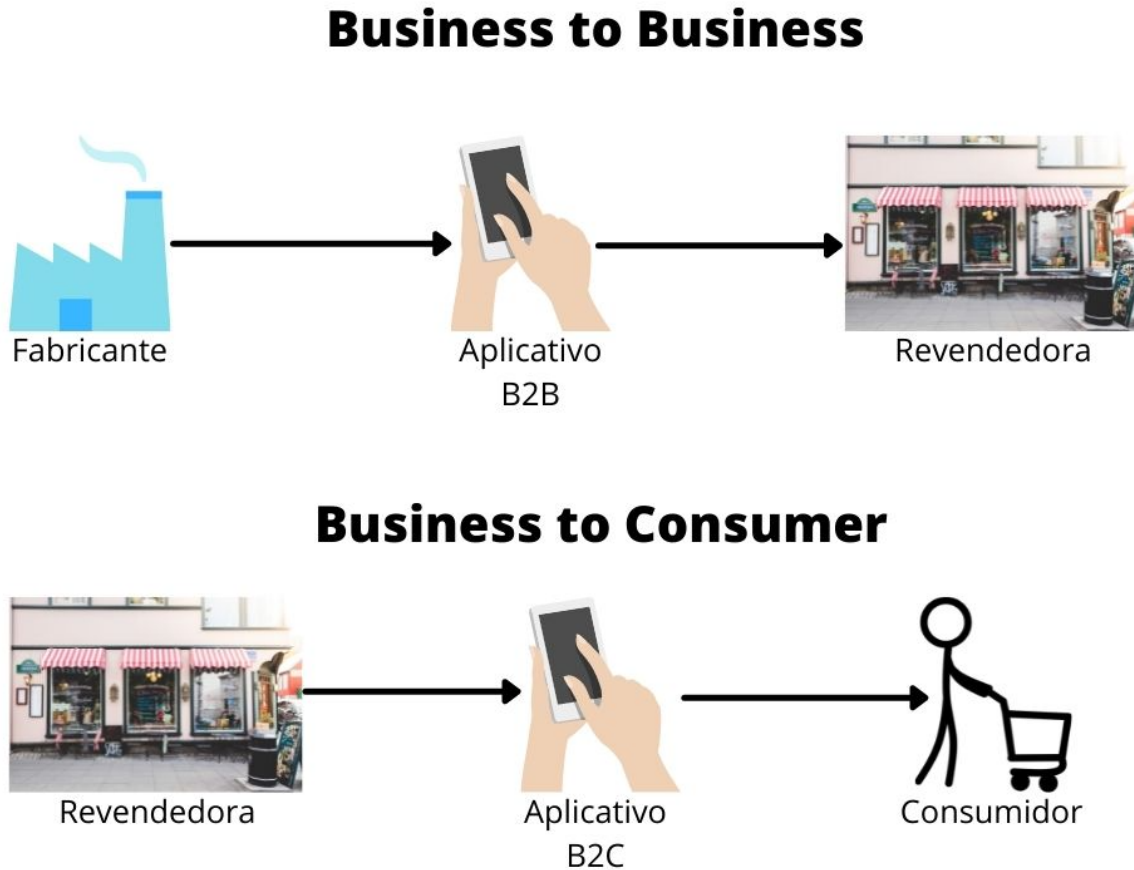
O objeto de estudo desta pesquisa é a automação de testes da plataforma de comércio eletrônico com a tecnologia RPA, em uma empresa que utiliza a Metodologia Ágil Scrum (MAS) para o gerenciamento de projetos. Esta metodologia é definida em (CLAIRE DRUMOND, 2021) como: “uma estrutura que ajuda as equipes a trabalharem juntas e a aprenderem com as experiências, sejam êxitos ou fracassos”. A MAS utiliza *sprints* (períodos limitados) para definir o tempo de desenvolvimento de um determinado conjunto de trabalho (CLAIRE DRUMOND, 2021). Na empresa em questão, são utilizados *sprints* de duas semanas e todas as demandas do projeto são designadas pelo gerente de produto (GP), o qual é responsável por gerar as “histórias” de usuários. No ciclo de desenvolvimento de uma aplicação que utiliza MAS, história ou história de usuário é o processo responsável pela especificação da funcionalidade e da validação de cada recurso de *software*.

A empresa na qual se realiza o presente trabalho é uma multinacional, presente em todos os continentes, que opera em mais de 19 países, como por exemplo, Argentina, Brasil, Bolívia, Canadá, Chile, El Salvador, Honduras, República Dominicana, Panamá, Peru, Colômbia, Uruguai, Paraguai, Estados Unidos, Equador, Tanzânia, México, África do Sul, entre outros. A célula de tecnologia que cuida da aplicação digital da empresa possui mais de 500 funcionários, localizada em Campinas, interior de São Paulo, e atua no desenvolvimento de aplicações para sua matriz multinacional, desenvolvendo aplicações B2B (do inglês, *Business to Business*) e B2C (do inglês, *Business to Consumer*), conforme ilustrado na Figura 3.

B2B é uma aplicação com foco na empresa revendedora e não no consumidor final. Como exemplos de aplicações B2B, pode-se citar a relação entre uma fabricante de eletrodomésticos e uma rede de varejo. Por outro lado, o modelo B2C é uma aplicação com foco no consumidor final, ou seja, representa por exemplo a relação entre uma rede varejista (vendedora) e o consumidor final do produto. Apesar de possuir aplicação B2C, o maior foco

da empresa no momento é o suporte às aplicações B2B, as quais buscam facilitar e agilizar a compra e entrega de produtos para os revendedores.

Figura 3: Diagrama ilustrativo dos modelos B2B e B2C



Fonte: Autoria Própria

A empresa, cujo objeto de estudo foi implementado nesta pesquisa, não nasceu como uma empresa de tecnologia, no entanto, diversificou-se, criando uma célula interna destinada a este novo ramo, com o intuito de atender da melhor forma possível seu cliente e continuar crescendo como corporação e como marca. Com a pandemia do novo coronavírus (COVID-19), a maior parte das organizações investiu no desenvolvimento de aplicações WEB para atender seu público-alvo sem que ele precisasse sair de casa. Adicionalmente, para evitar a exposição de seus funcionários ao perigo da pandemia, várias empresas adaptaram alguns postos de trabalho presenciais em um modelo não presencial (também conhecido como *home office*), o que fortaleceu o uso de aplicações digitais. Com isso, observou-se que, ao utilizar uma aplicação digital intermediária, é possível permitir uma maior segurança sanitária (no que concerne à pandemia), além de reduzir o número de funcionários. A empresa em que foi implementado o objeto de estudo desta pesquisa, é mais uma das que investiu de forma significativa em aplicações digitais durante a pandemia e obteve sucesso.

Recentemente, observou-se um crescimento exponencial da célula de tecnologia da empresa, a qual começou de forma conservadora, com poucos funcionários e um único país e hoje conta com mais de 500 funcionários atendendo mais de 19 países. Com esse crescimento também aumentaram as demandas, sendo uma delas a automação dos testes da aplicação. Desde o surgimento da célula de tecnologia, iniciou-se a implementação de automação de testes com diversas ferramentas gratuitas. Contudo, cada linguagem de programação empregada apresenta peculiaridades próprias e desafios. Cada vez mais, os funcionários de QA (analista de testes), encontram dificuldade na definição de uma linguagem única para estudar e implementar seus casos de testes, de forma a estarem sempre preparados para atender novas demandas. Essa dificuldade decorre da falta de padronização das ferramentas e da variação da curva de aprendizagem associada a cada linguagem de programação usada nos departamentos de QA de diferentes instituições. Com o intuito de facilitar a implementação da automação, a metodologia adotada neste trabalho foi um estudo de caso na empresa multinacional descrita anteriormente utilizando uma ferramenta de RPA chamada UiPath. Os principais motivos para escolha dessa ferramenta em particular, são: (i) possui um pacote (Test Suite) especialmente desenvolvido para automação/validação de casos de testes, (ii) um ambiente de desenvolvimento mais amigável, (iii) uma comunidade extensa e suporte on-line, o que facilita com que um novo usuário aprenda do zero a implementar casos de testes mais rapidamente.

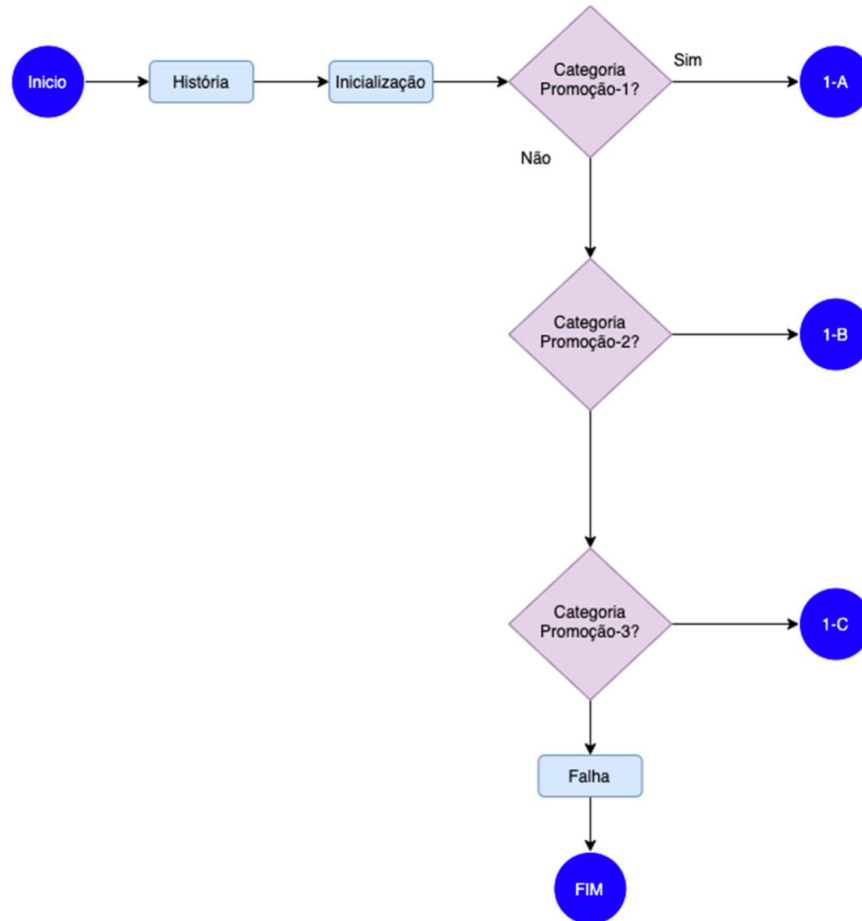
Como mencionado anteriormente, este estudo de caso é exploratório, pois explora o emprego de uma ferramenta de RPA em automação de testes, visto que a quantidade de pesquisas sobre esse assunto ainda é escassa na literatura especializada. Este estudo de caso também pode ser classificado como quantitativo, pois são comparados os desempenhos temporais de execução e desenvolvimento obtidos com diferentes tecnologias e execução manual.

Para a implementação da automação de testes neste trabalho utilizaram-se as ferramentas UiPath e Python. Os testes de validação são voltados para promoções, as quais pertencem a três categorias denominadas respectivamente: Promoção-1, Promoção-2 e Promoção-3. De posse da história, o analista cria os casos de testes para validar a história implementada pelo desenvolvedor de cada tecnologia. Frequentemente, essa validação ocorre de forma manual na sua primeira execução e, para manter a qualidade da aplicação, os casos de testes devem ser validados periodicamente, o que torna a automação dos testes atrativa em termos de produtividade. Na seção seguinte será descrita a modelagem dos casos de testes que foram automatizados nesta pesquisa.

### 3.1 MODELAGEM DOS TESTES DE AUTOMAÇÃO

Na Figura 4, iniciado o processo, conhece-se a história, que fornecerá informação quanto aos requisitos da promoção (tipo de promoção, demandas do GP etc.). Na sequência, realiza-se a etapa de inicialização com o login na aplicação (com e-mail e senha) e o direcionamento para a página de promoções, antes de se testar o tipo de promoção, que foi definido na história.

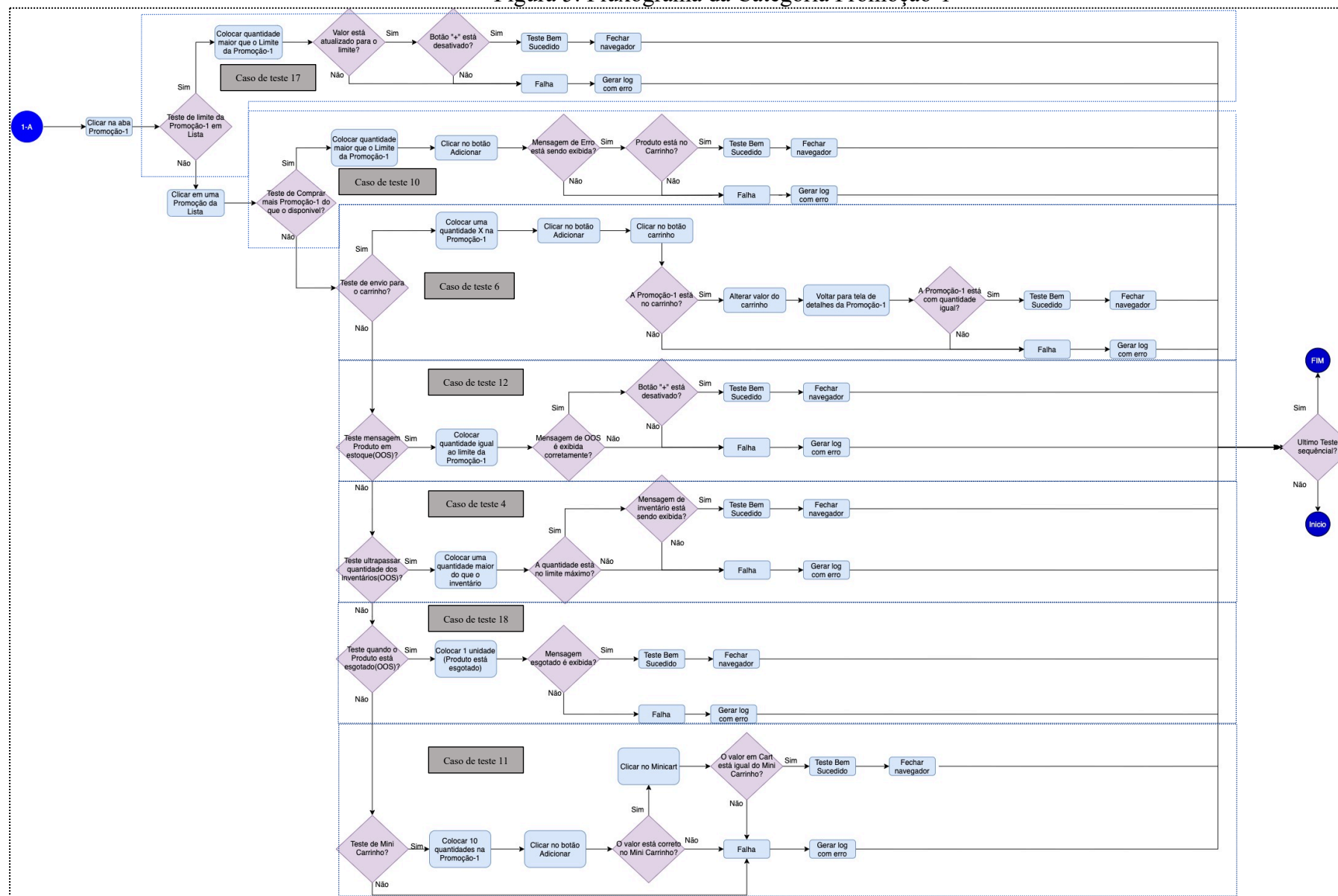
Figura 4: Fluxograma inicial



Fonte: Autoria própria

A Figura 5 refere-se à demanda que envolve “Promoção-1”, portanto, deve-se “Clicar na Aba Promoção-1” e, então, verificar se há demanda para o “Teste de limite da Promoção-1 em Lista”, o que consiste em verificar se a quantidade limite da Promoção-1 é respeitada.

Figura 5: Fluxograma da Categoria Promoção-1



Fonte: Autoria própria



Se houver um limite máximo para a Promoção-1 (caso de teste 17) e demanda para realização desse primeiro teste, se digitado, no campo da quantidade, um valor maior do que o limite da promoção, este deverá ser corrigido automaticamente para o valor máximo da Promoção-1. Caso isso não aconteça, haverá o registro de falha em um arquivo que será encaminhado para o desenvolvedor. Ao continuar esse teste, haverá uma segunda validação, relativa ao botão “+” que incrementa a quantidade da promoção. Esse botão deverá estar desabilitado ao se atingir o valor máximo da Promoção-1, nesse caso o teste será bem-sucedido, a validação encerrada e o navegador finalizado.

Na ausência de demanda para o “Teste de limite da Promoção-1 em lista (caso de teste 17)”, será escolhida uma das promoções disponíveis (todas serão testadas), que será aberta em sua página de detalhes. Nesta segunda validação da Promoção-1, se houver demanda para o "Teste de comprar mais da promoção 1 do que o disponível" referente ao caso de teste 10, será selecionada uma quantidade maior que o limite da promoção, a qual será adicionada ao carrinho após se acionar o botão "Adicionar". Nessa etapa, uma mensagem de erro deve ser exibida, caso isso não aconteça, haverá a devolução de um registro de falha ao desenvolvedor. Se a mensagem de erro for exibida, em seguida, verifica-se se o produto está no carrinho para definir se o teste foi bem-sucedido ou se um relatório de falha deve ser gerado.

Na ausência de demanda para o "Teste de comprar mais promoção 1 do que o disponível" e demanda para o "Teste de envio para o carrinho" referente ao caso de teste 6, verifica-se se a Promoção-1 está no carrinho com a quantidade ‘X’ adicionada na página de detalhes, pois, caso contrário, haverá indicação de falha do teste, com o registro em arquivo. No entanto, caso a primeira validação seja bem-sucedida, na sequência, a quantidade selecionada da Promoção-1, deve ser alterada no carrinho e, então verifica-se se a quantidade no carrinho é a mesma na tela de detalhes. Neste caso, a validação terá sido bem-sucedida e será registrado o êxito do teste em um arquivo. Essa validação estará encerrada.

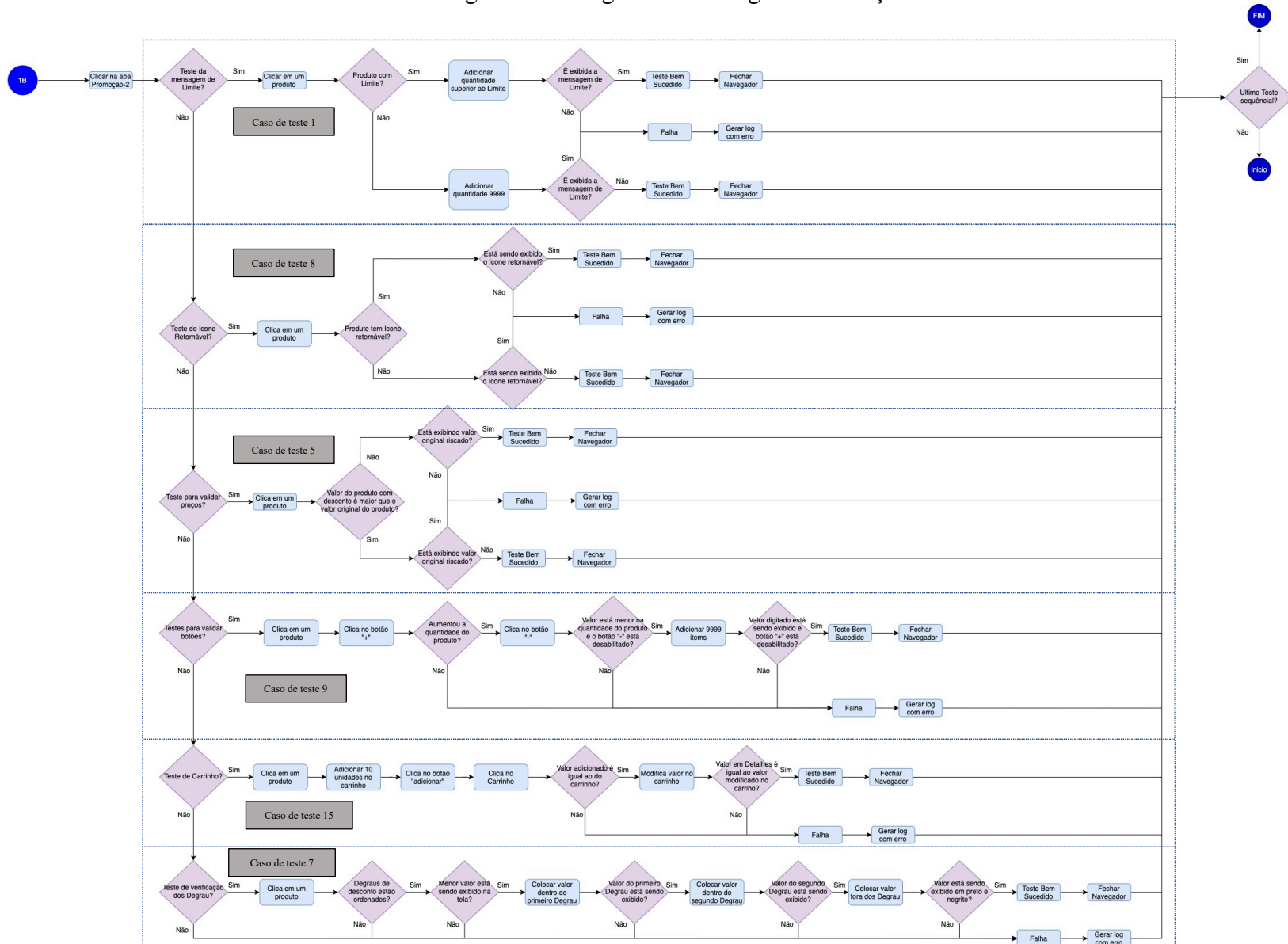
Caso haja demanda para o “Teste mensagem Produto em estoque (OOS)” referente ao caso de teste 12, em que OOS é uma sigla em inglês que significa *out of stock* (fora de estoque), primeiramente, é colocada a quantidade limite de produtos da Promoção-1. Em seguida, é feita a validação da exibição da mensagem “OOS” dos produtos que compõem a Promoção-1. Caso a mensagem não seja exibida, haverá indicação de falha do teste, com o registro em arquivo. Porém, se a mensagem for exibida corretamente, haverá a segunda verificação, isto é, o botão “+” deverá estar desabilitado. Caso esteja desabilitado, a validação foi bem-sucedida e haverá o registro de êxito do teste em arquivo e essa validação será encerrada. Caso contrário, haverá indicação de falha, com o registro em arquivo.

Com a demanda para a validação de “Teste ultrapassar quantidade dos inventários (OOS)?” referente ao caso de teste 4, primeiramente, deve-se adicionar uma quantidade maior que a em inventário. Em seguida, é verificado se o valor da quantidade está no valor máximo do inventário, caso positivo, o teste será validado se a mensagem de inventário está sendo exibida. Caso a validação tenha sido bem-sucedida, será registrado o êxito do teste em um arquivo e a validação será encerrada. Caso contrário, haverá indicação de falha do teste, com o registro em arquivo.

A próxima validação será “Teste quando o Produto está esgotado (OOS)?” referente ao caso de teste 18. Nesse caso, uma unidade da promoção deve ser adicionada, preenchendo o campo quantidade e clicando no botão “Adicionar”. O produto dentro da promoção está esgotado, por isso será feita a validação da exibição da mensagem de esgotado. Caso a validação tenha sido bem-sucedida, será registrado o êxito do teste em um arquivo. Caso contrário, haverá indicação de falha do teste, com o registro em arquivo.

No último caso de teste da Promoção-1, quando houver demanda para a validação de “Teste de Mini Carrinho” que se refere ao caso de teste 11, primeiramente serão adicionadas 10 unidades da promoção. Em seguida, deve-se preencher o campo de quantidade e clicar no botão “Adicionar”. Na primeira validação a ser feita, deve-se verificar se a quantidade adicionada da promoção é condizente com a quantidade exibida ainda no Mini Carrinho (Ícone de Carrinho, com a soma total dos produtos presentes no carrinho), que se encontra na tela de detalhes. Posteriormente, deve-se clicar no botão “Mini Carrinho”, sendo direcionado ao carrinho. A seguir é verificado se o valor exibido no Mini Carrinho está igual ao exibido na tela de carrinho, nesse caso a validação terá sido bem-sucedida e será registrado o êxito do teste em um arquivo, finalizado com sucesso. Quando as validações anteriores não forem satisfeitas, haverá indicação de falha do teste, com o registro em arquivo.

Figura 6: Fluxograma da Categoria Promoção-2



Fonte: Autoria própria

Na Figura 6, os testes envolvem “Promoção-2”, portanto, deve-se “Clicar na Aba Promoção-2” e, então, verificar se há demanda para o “Teste da mensagem de limite” que se refere ao caso de teste 1, que habilita a exibição de uma mensagem se o limite de quantidade comprada for extrapolado. Se houver demanda para esse teste, será escolhido um produto (clicar em um produto) e será verificado se o produto possui limite máximo ou não. Caso haja um limite de produtos que possam ser adicionados a essa promoção, um valor maior do que o limite será adicionado no campo da quantidade. Se a mensagem não for exibida, o teste terá falhado e um arquivo reportando o erro será devolvido ao desenvolvedor. Por outro lado, um teste bem-sucedido fechará o navegador. Se o produto não possuir limite máximo associado a essa promoção, mesmo com a adição de 9999 produtos, a mensagem de limite não deve ser exibida, com isso, o teste será bem-sucedido e o navegador será fechado. Por outro lado, se a mensagem for exibida, um erro será reportado ao desenvolvedor.

Ainda na Figura 6, caso haja demanda para o caso de teste 8 “Teste de ícone retornável”, então, será escolhido um produto (clicar no produto) e em seguida serão investigadas duas possibilidades de falha: o produto deve possuir ícone retornável, que não está sendo exibido, ou o produto não possui ícone retornável e este é exibido de maneira errônea. Nesses dois casos, o teste de ícone retornável devolverá um relatório de falha para o desenvolvedor. Caso contrário, um arquivo será gerado indicando sucesso e o navegador será fechado.

Quando não houver demanda para os testes anteriores, mas houver demanda para o caso de teste 5, “Teste para validar os preços”, então seleciona-se um produto, o que envolverá duas validações. A primeira ocorre quando o valor promocional do produto é maior do que seu valor original e mesmo assim o valor original (mais baixo) foi exibido riscado na plataforma, juntamente ao valor mais alto. A segunda ocorre quando o produto tem um valor promocional menor que o valor original, no entanto, o valor original riscado não foi exibido na plataforma juntamente ao valor com desconto para indicar a promoção. Em ambos os casos, um relatório de falha é enviado ao desenvolvedor, caso contrário o teste será considerado bem-sucedido e um arquivo com o sucesso é gerado e o navegador fechado.

Se não houver demanda para os testes anteriores, mas houver demanda para o caso de teste 9, “Teste para validar botões”, então, seleciona-se um produto, em seguida, o primeiro passo é o teste de adição, ou seja, clica-se no botão “+” e verifica-se se o produto teve sua quantidade incrementada. Se o primeiro passo for bem-sucedido, testa-se o segundo passo: de retirada de produtos, ou seja, clica-se no botão “-” e verifica-se se a quantidade do produto é decrementada e o botão “-” é desabilitado por ter atingido o limite inferior (0 produtos). Se ambos os passos anteriores forem bem-sucedidos, o terceiro passo corresponde ao teste do

limite superior, ou seja, adiciona-se 9999 ao campo quantidade do produto e, então, verifica-se se o botão “+” está desabilitado. Caso os três passos tenham produzido simultaneamente resultados positivos, o teste de validação de botões foi bem-sucedido e gera-se um arquivo que registra o sucesso e o navegador finalizado. Caso qualquer uma das três validações falhe, consequentemente um relatório de erro é gerado e enviado para o desenvolvedor.

Quando houver demanda para o “Teste de Carrinho” que se refere ao caso de teste 15, então, seleciona-se um produto, e adicionam-se 10 unidades ao campo da quantidade. Na sequência, clica-se no botão “Adicionar” para enviar os itens para o carrinho. Finalmente, clica-se no botão “Carrinho” para visualizar a quantidade de itens da compra. Nesse ponto, é realizado o primeiro teste a fim de se verificar se o valor adicionado (10 unidades) está presente no carrinho. Caso tenha sido adicionado corretamente, o valor é modificado, e um segundo teste é realizado a fim de se verificar se a alteração realizada também acontece na tela de detalhes. Se ambas as validações forem bem-sucedidas, um arquivo de êxito é gerado e o navegador fechado. Caso algum dos testes falhe, um arquivo com o erro é enviado ao desenvolvedor.

Caso haja demanda para o caso de teste 7, “Teste de verificação dos Degraus”, nesse caso, seleciona-se um produto e o primeiro passo consiste em se verificar se os degraus da promoção estão ordenados. A promoção em degraus consiste na aplicação de descontos progressivos, conforme a quantidade de produtos comprados for incrementada. A priori, antes de acrescentar produtos ao carrinho, o usuário deverá visualizar o preço mínimo (com o maior desconto possível) e, em seguida, de acordo com a quantidade de produtos selecionados, os descontos são aplicados. Caso o primeiro passo seja validado (ou seja, os degraus estejam ordenados), passa-se para o segundo passo, que consiste em testar se o valor exibido na tela corresponde ao de maior desconto (menor valor unitário). Se o segundo passo também for validado, coloca-se uma quantidade de itens dentro da primeira faixa de descontos (degrau da promoção) e verifica-se se o valor está de acordo com o previsto pela promoção. Em seguida, caso o teste anterior tenha sido validado, repete-se o procedimento para as duas próximas faixas (segundo e terceiro degraus de promoção). Por fim, coloca-se uma quantidade que extrapola os limites dos degraus da promoção, a fim de verificar se o valor exibido na tela está em negrito. Caso todas as verificações anteriores sejam válidas, o teste é considerado bem-sucedido e um arquivo com o registro do êxito é salvo. Caso contrário, o teste falhou e um arquivo com a descrição do erro é gerado e enviado para o desenvolvedor.

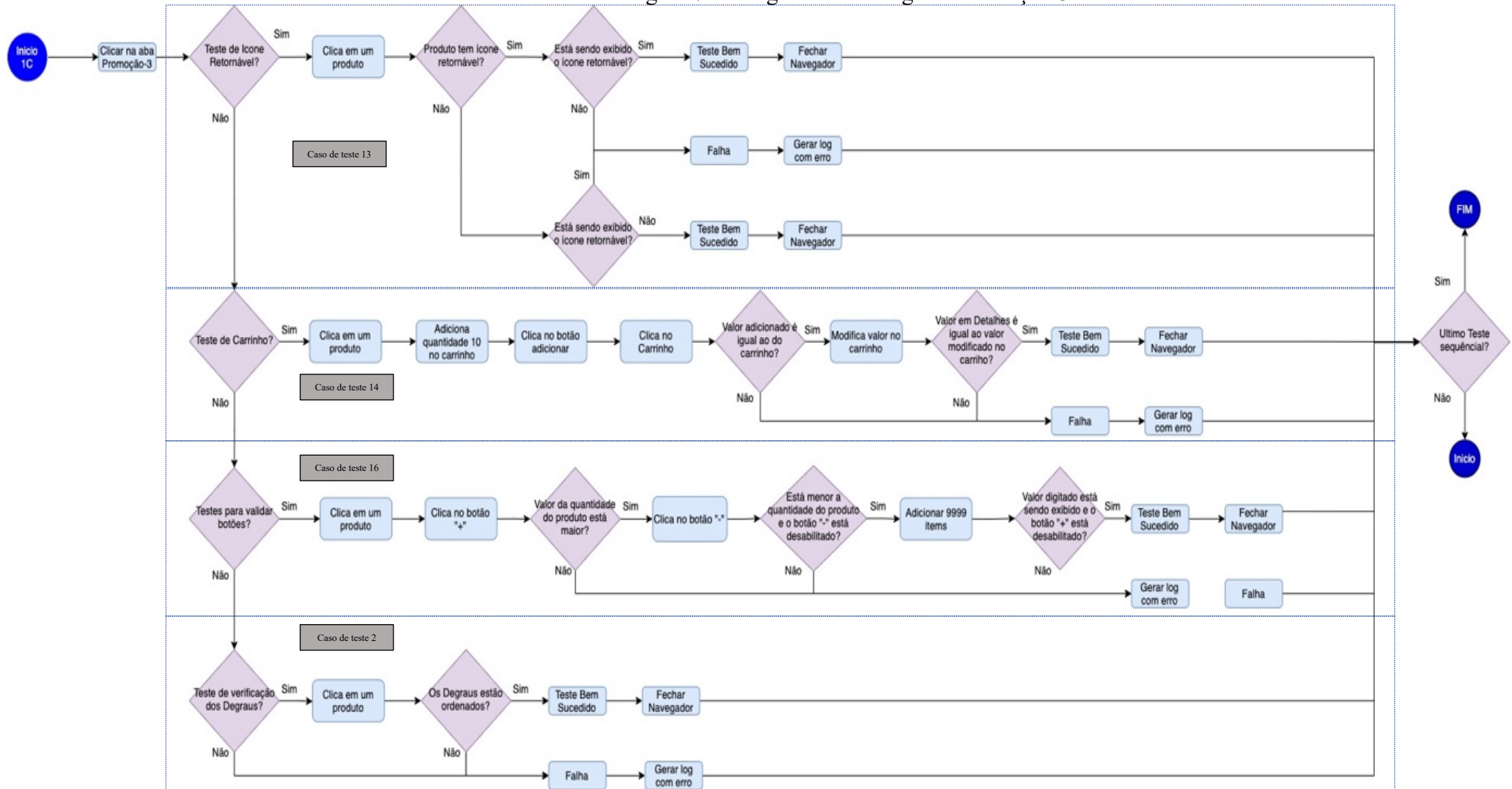
Na Figura 7, os testes envolvem a “Promoção-3”, então, clica-se na aba Promoção-3. Caso haja demanda para o caso de teste 13, “Teste de ícone retornável”, então, será escolhido um produto (clicar no produto) e em seguida serão investigadas duas possibilidades de falha: o produto deve possuir ícone retornável (Ícone Retornável simboliza que o produto precisa de devolução do casco), que não está sendo exibido, ou o produto não possui ícone retornável e este é exibido de maneira errônea. Nesses dois casos, o teste de ícone retornável devolverá um relatório de falha para o desenvolvedor. Caso contrário, quando o teste for bem-sucedido, um arquivo será gerado com o sucesso e o navegador será fechado.

Quando houver demanda para o “Teste de Carrinho” referente ao caso de teste 14, então, seleciona-se um produto, e adicionam-se 10 unidades ao campo da quantidade. Na sequência, clica-se no botão “Adicionar” para enviar os itens para o carrinho. Finalmente, clica-se no botão “Carrinho” para visualizar a quantidade de itens da compra. Nesse ponto, é realizado o primeiro teste a fim de se verificar se o valor adicionado (10 unidades) está presente no carrinho. Caso tenha sido adicionado corretamente, o valor é modificado e um segundo teste é realizado a fim de se verificar se a alteração realizada também acontece na tela de detalhes. Se ambas as validações forem bem-sucedidas um arquivo de êxito é gerado e o navegador fechado. Caso algum dos testes falhe, um arquivo com o erro é enviado ao desenvolvedor.

Se não houver demanda para os testes anteriores, mas houver demanda para o caso de teste 16, “Teste para validar botões”, seleciona-se um produto e, em seguida, o primeiro passo é clicar no botão “+” e verificar se o produto teve sua quantidade incrementada. Se o primeiro passo for bem-sucedido, testa-se o segundo passo: de retirada de produtos, ou seja, clica-se no botão “-” e verifica-se se a quantidade do produto é decrementada e o botão “-” é desabilitado por ter atingido o limite inferior (0 produtos). Se ambos os passos anteriores forem bem-sucedidos, o terceiro passo corresponde ao teste do limite superior, ou seja, adiciona-se 9999 ao campo quantidade do produto e, então, verifica-se se o botão “+” está desabilitado. Caso os três passos tenham produzido sequencialmente resultados positivos, o teste de validação de botões foi bem-sucedido e gera-se um arquivo que registra o sucesso e o navegador finalizado. Caso qualquer uma das três validações falhe, consequentemente um relatório de erro é gerado e enviado para o desenvolvedor.

Caso haja demanda para o caso de teste 2, “Teste de verificação dos Degraus”, nesse caso, seleciona-se um produto, valida-se se os degraus da promoção estão ordenados. Com a confirmação dos degraus ordenados, então o teste foi bem-sucedido e o navegador foi finalizado. Caso contrário, o teste falha e um arquivo é gerado e enviado para o desenvolvedor.

Figura 7: Fluxograma da Categoria Promoção-3



Fonte: Autoria própria

### 3.2 MATERIAIS E FERRAMENTAS COMPUTACIONAIS

Para a automação e execução dos casos de testes descritos na seção anterior, foi utilizado um computador com as seguintes características: Processador Intel i7 4ª geração com 8gb de RAM, HD de 1Tb, SSD de 256Gb e placa de vídeo GTX 765M 2Gb. Nas subseções seguintes serão detalhadas as ferramentas de automação utilizadas nesta pesquisa.

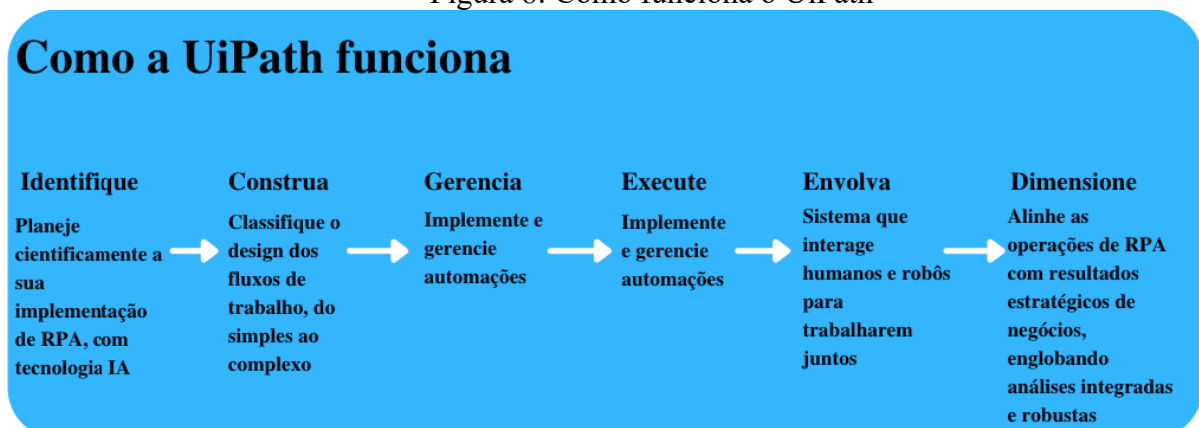
Para efeitos de comparação serão usadas duas estratégias: (i) UiPath e (ii) Python com Selenium. O Selenium é uma ferramenta empregada para automatização de testes de sistemas que permite ao usuário reproduzi-los rapidamente no ambiente real da aplicação, devido a sua integração direta com o navegador. Por meio de uma extensão, é possível criar os scripts de testes de forma simples usando diversas linguagens de programação (SOUSA, 2021).

#### 3.2.1 UiPath

De acordo com (STATION, 2019), a UiPath lidera a era da “automação” - defendendo um robô para cada pessoa e possibilitando que os robôs aprendam novas habilidades por meio de IA (inteligência artificial) e aprendizado de máquina, enquanto outras ferramentas de RPA apostam em um único robô por licença. A ferramenta disponibiliza em seu site uma área de formação (UiPath Academy) que disponibiliza cursos gratuitos e aberto a todos. A sua interface amigável facilita o desenvolvimento de fluxos de trabalho para automatização. A ferramenta conta também com um gravador embutido, que permite gravar tudo que o usuário está executando na interface o que, conseqüentemente, auxilia na automatização dos fluxos de trabalho. Esta ferramenta possibilita automatizar processos no módulo principal testes de validação em um pacote especializado chamado Test Suite.

Para automatizar processos em geral, ou casos de testes, é preciso primeiramente compreender o que se quer automatizar. A Figura 8 apresenta uma descrição geral de como é feito esse passo-a-passo.

Figura 8: Como funciona o UiPath



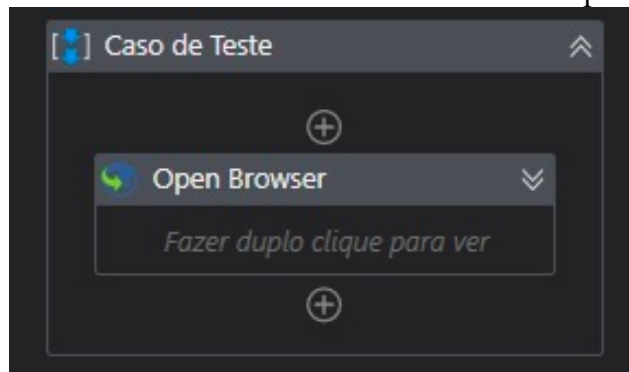
Fonte: (UIPATH, 2021)



Primeiramente é feita a identificação do que deve ser automatizado no caso de testes analisado. Em seguida, é implementado o caso de teste na ferramenta UiPath. Na sequência, é feita a gestão das automações implementadas e, posteriormente, executam-se os casos de testes. As duas etapas finais são referentes à interação, manutenção e organização dos casos de testes.

Na Figura 9, podemos ver o primeiro passo da criação de um caso de teste na ferramenta UiPath, que se baseia na função *Open Browser*. Esta função é responsável por abrir o navegador escolhido pelo desenvolvedor e ir para a URL pré-definida na configuração.

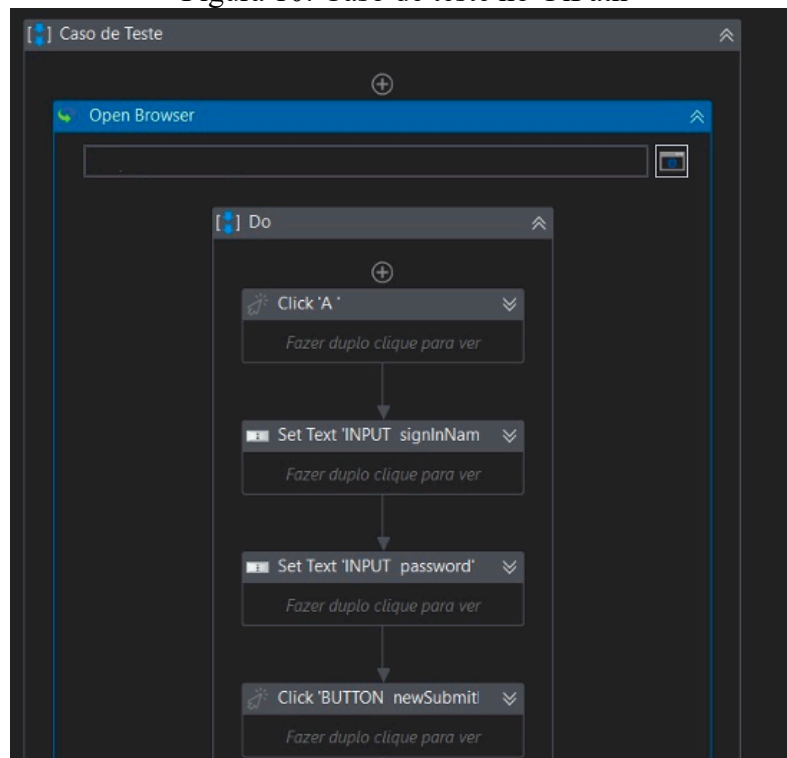
Figura 9: Vista de um caso de teste minimizado no Open Browser



Fonte: Autoria própria

Na Figura 10, é possível ver as instruções implementadas dentro do *Open Browser*. Note que a programação é feita na forma de fluxos, sem códigos ou linguagem de programação complexa, de maneira simples e visual.

Figura 10: Caso de teste no UiPath



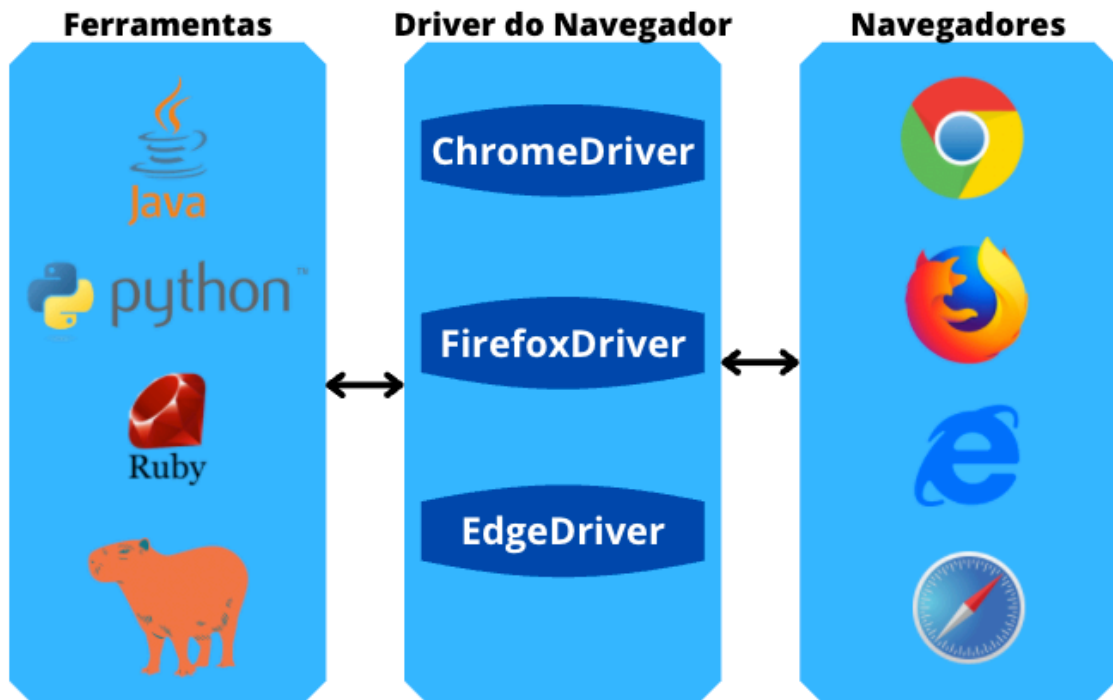
Fonte: Autoria Própria

### 3.2.2 Selenium

O Selenium Webdriver, em particular, foi escolhido nesta dissertação como tecnologia para comparar os casos de testes implementados via RPA. Tal ferramenta consiste em um recurso adicionado recentemente ao *framework* da API do Selenium, permitindo a execução direta de testes via código fonte no servidor, traduzindo o código de servidor para o código de cliente e executando testes automatizados na interface gráfica dos diversos navegadores suportados (SOUSA, 2021).

Uma vantagem do Selenium é a possibilidade que os desenvolvedores têm para criar testes unitários e atrelá-los aos testes automatizados, dentro de loops que abrem várias instâncias dos drivers dos navegadores, permitindo testar diferentes recursos das aplicações de forma muito mais rápida e performática. Neste trabalho, o Selenium Webdriver foi utilizado como uma biblioteca do Python. Python é uma linguagem de programação *Open-Source* (código aberto) de amplo espectro, muito utilizada em *data science*, *machine learning*, desenvolvimento WEB, desenvolvimento de aplicativos, automação de *scripts* e outras aplicações. O Python emprega o Selenium Webdriver para se comunicar com o navegador/aplicação, utilizando uma arquitetura como a representada na Figura 11. Na Figura 11, o primeiro retângulo representa a ferramenta que será utilizada, como por exemplo, o Python, neste ambiente será feita toda a parte de programação. Já no segundo retângulo, faz-se menção aos *drivers* que poderão ser utilizados para que os navegadores se comuniquem com os navegadores, um dos exemplos seria o Chromedriver e por fim temos os navegadores no último retângulo, representando a interface visual a ser utilizada.

Figura 11: Arquitetura do Selenium Webdriver



Fonte: Autoria própria

O Selenium é um conjunto de componentes e cada componente tem seu propósito na automação (PATIL; TEMKAR, 2017).

- Selenium IDE – é um *plugin* para Firefox (navegador), que permite aos analistas de testes registrar suas ações.
- Selenium Remote Control – é a primeira ferramenta automatizada de teste WEB que permite que os usuários escrevam testes de UI em várias linguagens de programação.
- Selenium Webdriver – é uma ferramenta de estrutura de automação que suporta vários navegadores e aplicativos Ajax e implementa uma abordagem mais moderna e estável em automatizar as ações do navegador. Tem bom controle em páginas dinâmicas onde os elementos de uma página da internet mudam sem que a própria página seja recarregada.
- Selenium Grid – É uma ferramenta usada para executar testes paralelos em várias máquinas em diferentes navegadores de forma simultânea o que resulta em economia de tempo de execução.

Para automatizar um único caso de testes, o fluxo do *framework* (Selenium-Python) desenvolvido é um pouco mais complexo do que aquele apresentado na Figura 10 com implementação via Test Suite do UiPath.

Para a implementação são necessárias quatro “páginas” integradas que se comunicam para a execução do caso de testes. Primeiramente é desenvolvida uma página com extensão “.feature” na qual é feita uma descrição em BDD (do inglês *Behavior Driver Development* - Desenvolvimento orientado por comportamento). BDD é uma técnica de desenvolvimento de *software* ágil que utiliza a linguagem Gherkin (CUCUMBER, 2021) como apoio, possuindo palavras-chave necessárias para a criação do cenário de teste com seus respectivos passos (LEAL, 2019). Na Figura 12 é possível ver a primeira página a ser criada no Python, nela é feita a escrita dos casos de testes em Gherkin.

Figura 12: Primeira página do fluxo do caso de testes em Python

```
@discount
Feature: Discount Prices
  As a logged user in the app
  The user goes to deals page
  The user goes to discount details page
  The user should be able to see the different prices

Background:
  Given the user is logged in
  And he goes to the deals page

@web @uat @ar @br @co @do @ec @pe
Scenario: Validate Prices in discount details page
  When he clicks in discount details page
  Then he should be able to see different prices
```

Fonte: Autoria própria

A segunda página, ilustrada na Figura 13, consiste na página que contém as classes, salva em formato “.py”. Nesta página, é possível ver as classes sendo chamadas na sequência e com a lógica necessária para ser executada corretamente.

Figura 13: Segunda página para implementação do fluxo do caso de testes em Python

```
from pytest_bdd import scenarios, given, when, then

scenarios('./features/discount.feature')

@given('the user is logged in')
def the_user_is_logged_in(driver):
    user = driver.data['browser'][driver.session_info.environment]['user']
    password = driver.data['browser'][driver.session_info.environment]['password']
    driver.pages.landing.access_the_login_screen()
    driver.pages.login.log_in(user, password)
```

Fonte: Autoria própria

A terceira página consiste no desenvolvimento das funções chamadas na página 2. Um exemplo ilustrativo da terceira página pode ser encontrado na Figura 14, na qual tem-se as funções desenvolvidas e a lógica para a execução dos casos de testes.

Figura 14: Terceira página para implementação do fluxo do caso de testes em Python

```

from automation_core import page
from selenium.webdriver.support.color import Color

@page(name='deals', components=['discount_product_list', 'combo_product_list', 'free_good_product_list'])
class Deals:

    def is_on_focus(self):
        return self.driver.actions.is_visible(self.mapping.TAB_DEALS)

    def select_tab_by_name(self, name):
        self.driver.actions.wait_until_elements_visible(self.mapping.TAB_DEALS)
        tabs = self.driver.actions.wait_until_elements_present(self.mapping.TAB_DEALS)
        tab_name_list = ['combo', 'discount', 'freegood']
        if name not in tab_name_list:
            raise ValueError(
                f'The "name" parameter must be one of the following: {"", ".join(tab_name_list)}')
        for tab in tabs:
            if '/' + name + '/' in tab.get_attribute('href'):
                tab.click()
                break

```

Fonte: Autoria própria

Por fim, a última etapa consiste em desenvolver uma página que guarda os *mappings* (localizadores dos elementos a serem chamados). Os localizadores são encontrados ao inspecionar a página do navegador e podem ser definidos por ID ou XPath, permitindo selecionar e testar os componentes. Um exemplo de desenvolvimento da quarta página pode ser encontrado na Figura 15, a qual, representa todos os *mappings* (referente ao endereço dos componentes da página a ser testada, podendo ser por XPATH ou ID). Todas as quatro páginas apresentadas da Figura 12 até a Figura 15 interagem entre si para executar um único caso de testes.

Figura 15: Quarta página para implementação do fluxo do caso de testes em Python

```

from application.web.common.mappings.header import HeaderMapping
from automation_core import mapping
from selenium.webdriver.common.by import By

@mapping(page='deals')
class DealsMapping(HeaderMapping):
    TXT_SCREEN_DEALS = (By.XPATH, '//span[@data-ui-id="page-title-wrapper"]')
    TAB_DEALS = (By.XPATH,
                '//*[@class="top-toolbar"]/descendant::div[contains(@class,"-product-page-toolbar-filter")]//a')
    TAB_DISCOUNT = (By.XPATH, '//a[contains(@href,"discount/")]')
    DISCOUNT_PRODUCT = (By.XPATH, '//div//div[@class="{...}-product-title"]')
    TXT_SCREEN_DISCOUNT = (By.XPATH, '//div[@class="{...}-product-title loading"]')
    PRICE_DISCOUNT = (By.XPATH, '//*[@class="price highlight"]')
    ORIGINAL_PRICE_DISCOUNT = (By.XPATH, '//*[@class="original-price"]')
    AVAILABLE_DISCOUNT_PRODUCT_CARD_LIST = (
        By.XPATH,
        '//*[@class="{...}-product-item type-discount']'
        '[not(./input[contains(@class,"disabled")])]'
        '[not(./button[contains(@class,"button-qty-plus disabled")])]'
    )

```

Fonte: Autoria própria

Com as explicações das ferramentas utilizadas nas seções 3.2.1 e 3.2.2, foi possível verificar que a ferramenta Selenium-Python tem um processo de desenvolvimento mais trabalhoso e complexo do que a ferramenta Test Suite – UiPath. No capítulo seguinte referente aos resultados será demonstrado através do tempo de desenvolvimento que devido ao seu processo mais complexo para desenvolver um caso de teste, a ferramenta Selenium-Python demandou mais tempo de desenvolvimento do que o UiPath, conforme era esperado.

#### 4 RESULTADOS

Conforme descrito anteriormente, este estudo de caso foi conduzido em uma empresa multinacional que atua em diversos países (no intervalo de tempo em que este projeto piloto foi desenvolvido as operações WEB e *mobile* da empresa alcançavam oficialmente 8 países, embora haja uma estimativa de que até o ano de 2022 esse número seja expandido para os 19 países que a empresa atende). Considera-se que a automação precisa ser validada em cada um dos países. Atualmente, o time de promoções no qual foi realizado o estudo, é responsável pela validação de 110 casos de testes em duas tecnologias (WEB e *mobile*). Considerando os 8 países multiplicados pelos 110 casos de testes, computa-se um total de 880 validações necessárias. Devido às suas particularidades, em um estudo realizado a priori, determinou-se que, dentre os 110 casos de testes disponíveis, a automação pode ser considerada vantajosa em apenas 80 deles, os quais são executados com maior frequência. Os 80 casos de testes viáveis para automação, podem ser divididos em 3 diferentes categorias: WEB, Android e iOS. Isso resulta em aproximadamente 26 casos de testes por tecnologia. Neste estudo de caso, em particular, foram automatizados apenas os casos de testes relacionados à tecnologia WEB.

Dentre os 26 disponíveis foram selecionados um conjunto de 18 casos de testes (6 dos mais importantes por categoria de promoção, totalizando aproximadamente 69% dos casos de testes WEB automatizados), considerados os mais relevantes presentes no repositório de testes. Dessa forma, foi possível visualizar as dificuldades, vantagens, desafios a serem enfrentados e colher todas as informações necessárias para comparar as tecnologias investigadas neste trabalho.

A Tabela 3 lista os casos de teste de validação implementados neste trabalho.

Tabela 3: Listagem dos casos de testes

Numeração	Nome do caso de teste	Descrição dos Casos de Teste
1	Teste da mensagem de Limite	Validar se o Limite diário/mensal que a Promoção-2 tem na tela de detalhes, adicionando quantidade para estourar o limite diário e mensal.
2	Teste de verificação dos Degraus em Promoção-3	Consiste em testar se os degraus da Promoção-3 estão ordenados em ordem numérica.
3	Teste para validar preços em Promoção-3	Validação se o preço da Promoção-3 está na coloração certa e se está exibindo somente o valor Original sem qualquer outro tipo de Promoção.
4	Testes ultrapassar quantidade dos inventários (OOS)	Ultrapassar a quantidade dos produtos e verificar se a mensagem é exibida e o valor volta ao máximo que o estoque está dizendo que tem.
5	Teste para validar preços Promoção-2	Validar os preços na Promoção-2, se a cor está correta e se está vindo os preços originais e com desconto.
6	Teste de envio para o carrinho Promoção-1	Validar se o produto em questão está sendo enviado para o carrinho ao clicar no botão adicionar, na quantidade correta.
7	Teste de verificação dos Degraus na Promoção-2	Verificar se os degraus da Promoção-2 estão ordenados em ordem numérica e ao adicionar a quantidade escolhida de um determinado degrau o campo Preço está exibindo o valor correspondente.
8	Teste de Ícone Retornável na Promoção-2	Verificar se o Ícone Retornável está sendo exibido nas telas correspondentes, com a coloração e a posição correta.
9	Teste para validar Botões na Promoção-2	Validar se os botões "+", "-", "Adicionar" e "Atualizar" estão funcionando conforme o esperado
10	Teste de comprar mais Promoção-1 do que o disponível	Tela de Detalhes da Promoção-1, o campo quantidade volta a ser o maior valor permitido, ao ser adicionado mais quantidade do que a disponível e uma mensagem é exibida.
11	Teste de Mini carrinho Promoção-1	Validar se após adicionar uma determinada quantidade no carrinho e verificar se o mini carrinho presente na mesma tela está com o valor/quantidade correta.
12	Teste mensagem Produto em estoque (OOS)	Validar se quando o Produto está com uma determinada quantidade de produto em estoque então a mensagem de somente x produtos disponíveis está presente, permitindo adicionar somente até este limite.
13	Teste de Ícone Retornável em Promoção-3	Verificar se o Ícone Retornável está sendo exibido nas telas correspondentes, com a coloração e a posição correta.
14	Teste de carrinho Promoção-3	Verificar se a interação entre a tela de detalhes e o carrinho estão sincronizados, ao adicionar pela tela de detalhes a quantidade no carrinho deve ser atualizada e o mesmo deve ocorrer quando atualizado o valor no carrinho.
15	Teste de carrinho Promoção-2	Verificar se a interação entre a tela de detalhes e o carrinho estão sincronizados, ao adicionar pela tela de detalhes a quantidade no carrinho deve ser atualizada e o mesmo deve ocorrer quando atualizado o valor no carrinho.
16	Teste para validar Botões em Promoção-3	Validar se os botões "+", "-", "Adicionar" e "Atualizar" estão funcionando conforme o esperado.
17	Teste de limite da Promoção-1 em lista	Verificar se o Limite diário/mensal que a Promoção-1 tem na tela de detalhes, adicionando quantidade para estourar o limite diário e mensal na tela de lista dos produtos.
18	Teste quando o produto está esgotado (OOS)	Validar se quando o Produto está esgotado a mensagem de fora de estoque está presente e o produto fica desabilitado para comprar.

Fonte: Autoria própria



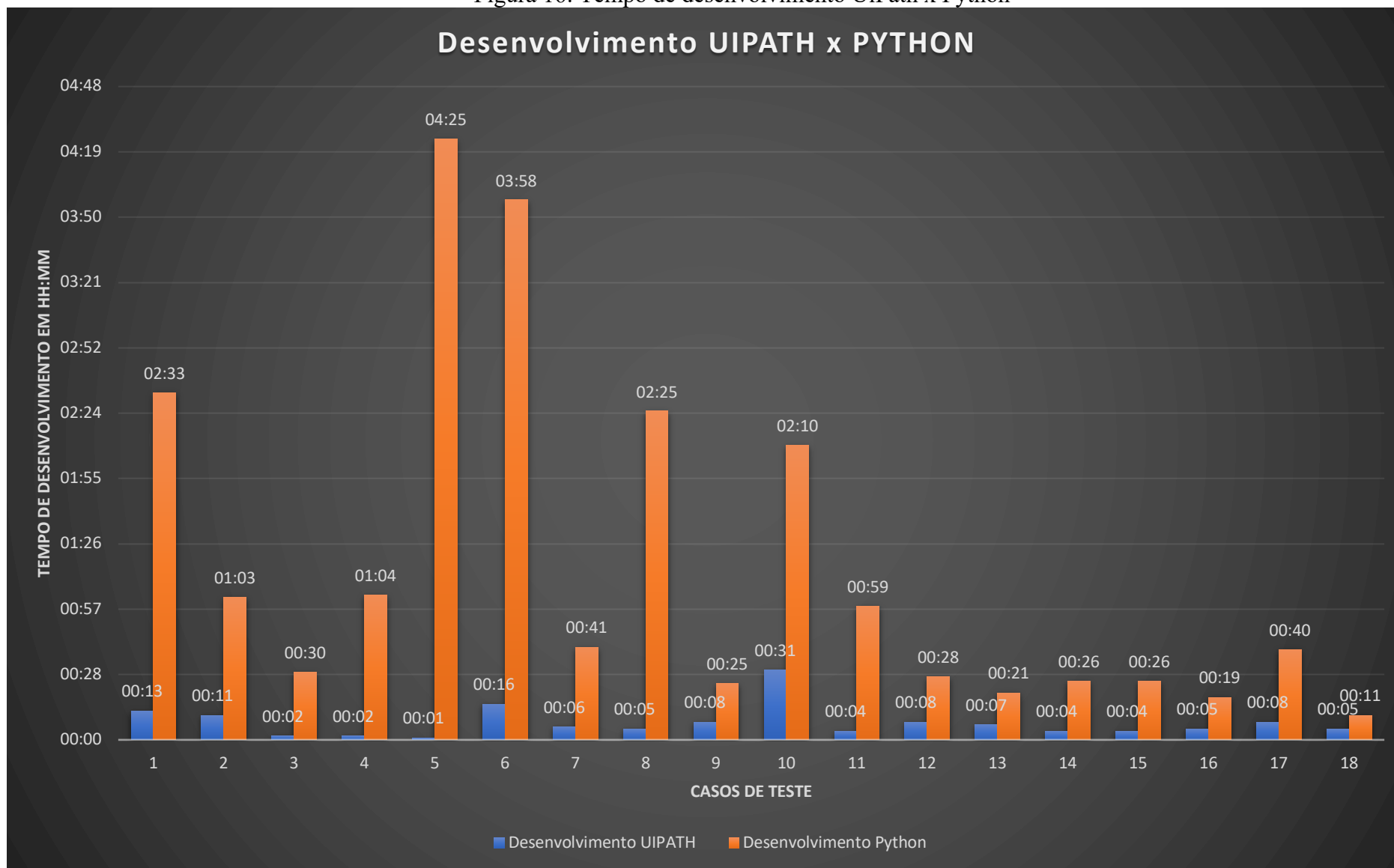
A empresa, na qual se realizou o estudo, autorizou que o autor dedicasse cerca de 10 horas semanais para o projeto de automação de testes dentre as 80 horas úteis de uma *sprint*, o que representa 12,5% do tempo útil da *sprint*. Como já mencionado, dos 80 casos de testes passíveis de automação, 18 deles foram automatizados utilizando as ferramentas: UiPath e Selenium-Python. Os resultados obtidos foram analisados para avaliar os benefícios trazidos.

A primeira análise que foi realizada refere-se ao tempo de desenvolvimento dos casos de testes requerido por cada ferramenta. O resultado para os 18 casos de testes relatados na Figura 16 na forma de um gráfico de barras, cuja unidade de medida do tempo está em horas:minutos. As automatizações com a ferramenta Test Suite do UiPath levaram em média 7 min para serem desenvolvidas, um tempo máximo de aproximadamente 31 min e mínimo de 2 min. O tempo médio de desenvolvimento com a ferramenta em Selenium-Python foi de 1h e 17 min, com tempo máximo de aproximadamente 4 h e 25 min e mínimo de 11 min. A discrepância desses resultados é evidenciada visualmente na Figura 16.

Esse resultado corrobora a discussão que foi apresentada na Seção 3.2.1 e na Seção 3.2.2, quando se verifica que complexidade de implementação em Selenium-Python é bem maior do que a exigida pelo Test Suite do UiPath, que tem sua interface e usabilidade mais amigável que a linguagem de programação Python.

Na Figura 16, alguns casos de testes como 1, 5, 6, 8 e 10 merecem destaque pois suas diferenças temporais destoam dos demais casos. A principal justificativa para essa diferença temporal é a necessidade de validar muitos testes em sequência. Por exemplo, no caso de teste 5, quanto automatizado, é validado se os componentes estão presentes na tela, se ambos os preços originais e com desconto estão conforme o esperado (moeda do país), se as cores estão corretas (preço original em preto e riscado e preço com desconto em verde). Todas essas validações no Selenium-Python demandam várias linhas de programação, enquanto no Test Suite do UiPath, por utilizar recursos de verificação visual e comparativa dos componentes, essa implementação é muito mais fácil e rápida.

Figura 16: Tempo de desenvolvimento UiPath x Python



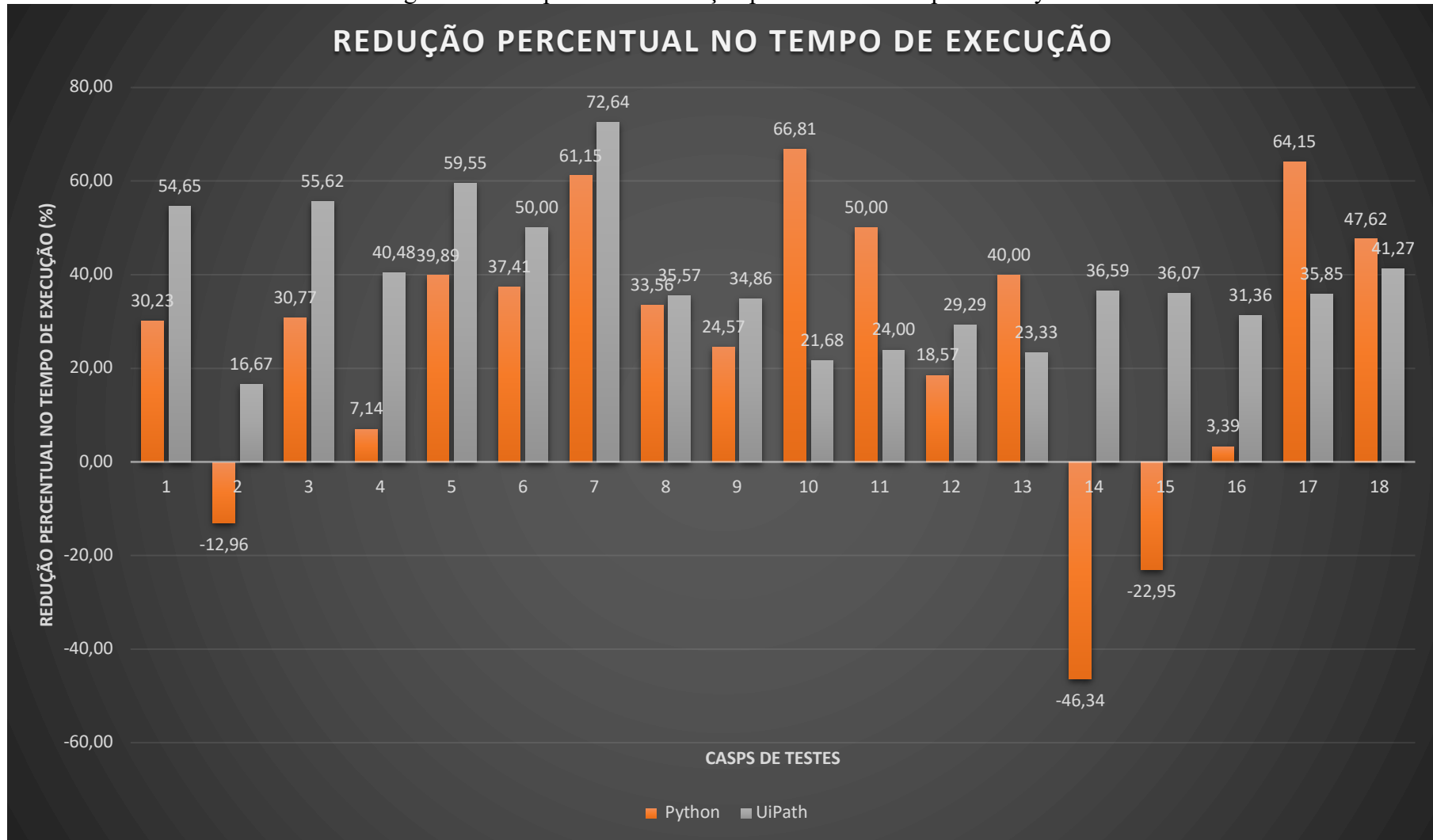
Fonte: Autoria própria

A segunda análise realizada consiste em uma comparação da redução percentual do tempo de execução de cada um dos casos de testes de validação automatizados com sua execução manual por um analista de testes experiente.

Na Figura 17, é apresentada uma comparação com a execução dos testes usando a ferramenta Selenium-Python e UiPath. Ao observar essa figura, chamam a atenção: os casos de testes 2, 14 e 15 em Selenium-Python, pois levaram mais tempo em sua execução automática do que em sua execução manual, obteve uma "redução" negativa do tempo de execução. Após uma análise detalhada, verificou-se que esse acréscimo temporal se deu pelo Python ser uma linguagem de alto nível e sua digitação dinâmica. Como, em Python, as variáveis e tipos de argumento não são declarados (só existem a partir do momento em que a elas se atribui algum valor), devido a sua poderosa lista polimórfica e tipos de dicionário, isso torna-se uma vantagem no desenvolvimento da aplicação, no entanto, quando comparados a outras linguagens de programação, como Java por exemplo, os programas em Python podem demorar mais tempo para executar. Na média, a ferramenta Python obteve redução percentual do tempo de execução em torno de 26%, com redução máxima de 67% e acréscimo no tempo de execução de até 46%.

Ainda na Figura 17, analisou-se a redução percentual do tempo de execução de cada um dos casos de testes de validação automatizados via ferramenta Test Suite do UiPath que indicou uma redução média em torno de 39%, porém obteve-se redução máxima de 73% e mínima de 17%. Diferentemente do que foi observado com o emprego da ferramenta baseada em Python, em todos os casos de testes, a automação via ferramenta RPA mostrou-se benéfica em termos de economia de tempo de execução.

Figura 17: Comparativo da redução percentual no tempo entre Python e UiPath

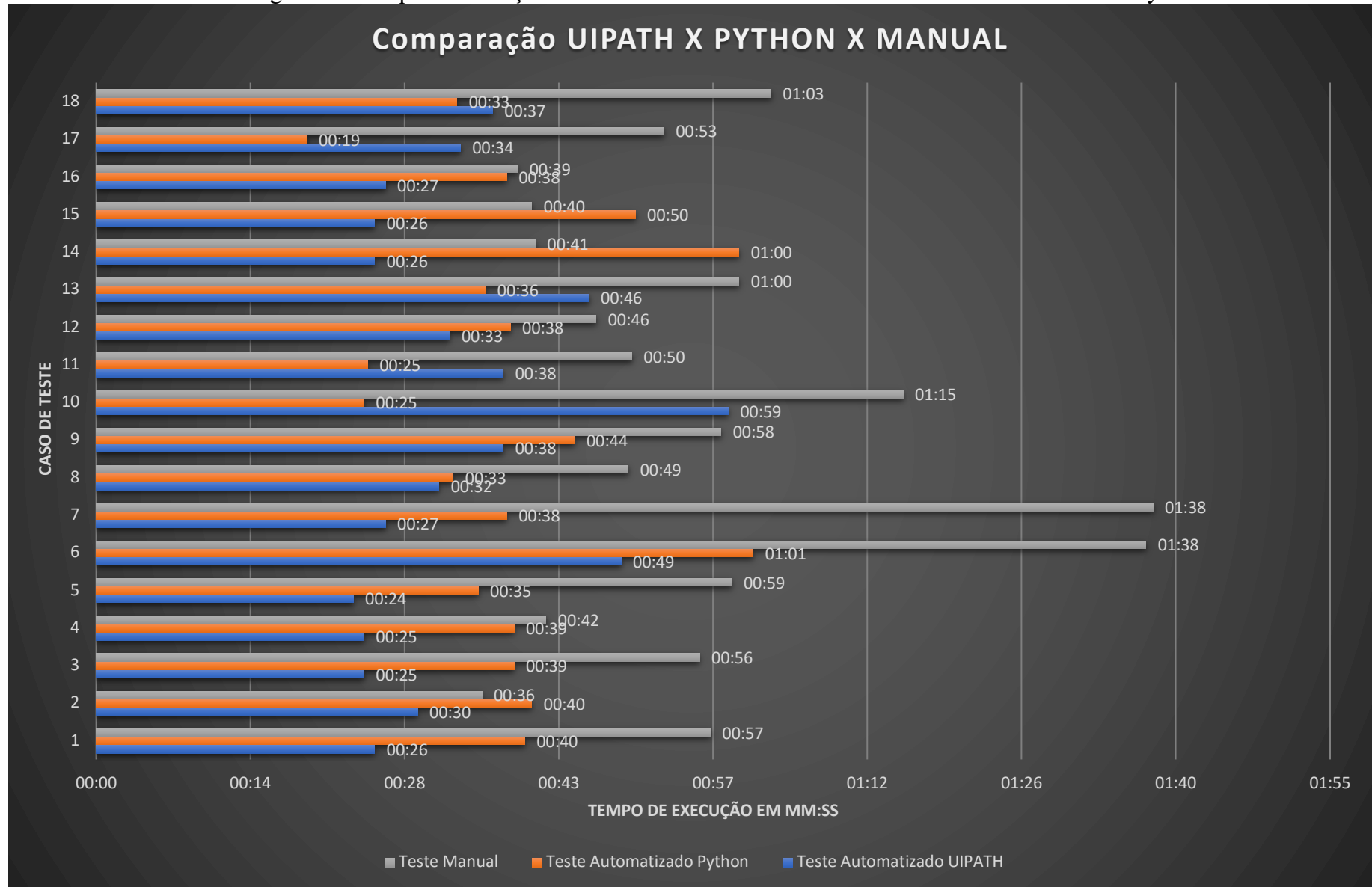


Para uma análise comparativa mais completa, apresentam-se em um mesmo gráfico de barras (Figura 18) os tempos de execução dos testes de validação obtidos de forma manual e automatizada via Test Suite-UiPath e Selenium Python.

Assim como já havia sido mostrado com o auxílio da Figura 17, a Figura 18 permite reiterar que, em nenhum momento, o tempo de execução da ferramenta UiPath é superior ao tempo gasto pelos testes manuais. Em contrapartida, em alguns casos de testes, a ferramenta RPA mostrou-se menos eficiente que a ferramenta baseada em linguagem de programação Python. Após uma análise detalhada, verificou-se que essa "anomalia" encontrada nos casos de testes 10, 11, 13, 17 e 18, se deu pela necessidade do uso de comparadores de imagem na ferramenta UiPath, em vez da identificação de componentes feita pela Selenium-Python (na forma XPath ou ID, conforme descrito na Subseção 3.2.2). Uma vez que estes recursos demandam maior capacidade de processamento, o tempo de execução também sofreu um incremento considerável.

Para finalizar, foi feita uma estimativa de que a validação manual dos 80 casos de testes nos 8 países (640 testes) levaria no mínimo 10 horas e 3 minutos para ser realizada. É importante destacar que essa estimativa não leva em consideração problemas ou erros inerentes aos testes manuais nem paradas para descanso. Por outro lado, fazendo uma extrapolação dos resultados obtidos para os 18 casos de testes e calculando uma estimativa do tempo que seria gasto para executar a validação automática dos 640 testes obtém-se o seguinte resultado: a ferramenta RPA Test Suite-UiPath permitiria a realização dos testes em um tempo de 3 horas e 2 minutos, enquanto a ferramenta Selenium-Python levaria cerca de 6 horas e 54 minutos para concluir os testes. Note que a economia de tempo gerada pela automação, proporciona ao validador tempo adicional para realizar outras tarefas da *Sprint*.

Figura 18: Tempo de execução dos testes manuais x automatizados UiPath x automatizados Python



Fonte: Autoria própria

## 5 CONCLUSÕES E CONSIDERAÇÕES FINAIS

Neste trabalho foi realizada a automatização de 18 testes de validação de uma plataforma de comércio eletrônico de uma empresa multinacional, que possui 80 casos de testes passíveis de automação. Para esse objetivo foram usadas duas tecnologias de automação de testes. A primeira é uma ferramenta RPA conhecida como UiPath, uma tecnologia amigável e flexível baseada em programação por fluxos a qual poderia ser aplicada em outros setores da empresa carentes de automação de processos repetitivos como, por exemplo, departamento financeiro ou de recursos humanos. A segunda ferramenta é um recurso chamado Selenium Webdriver, o qual foi desenvolvido especialmente para automação de testes, no entanto é baseado em linguagem de programação de alto nível Python. Portanto, avaliou-se a eficiência de desenvolvimento e execução dos testes de validação automatizados, comparando-os com os tempos de execução manual.

Em média, a redução do tempo de execução dos testes no UiPath ficou em 39%, porém, obtiveram-se reduções entre de 73% e 17% enquanto no Python a média da redução do tempo de execução foi de 26%, produzindo até mesmo tempos de execução maiores que os realizados por um analista de testes de maneira manual. Ao analisar os dados referentes aos testes sequenciais (execução de todos os testes implementados em sequência) a economia de tempo foi de 69% com UiPath e 33% com Python, se comparada à execução manual. Adicionalmente, uma das principais vantagens da ferramenta RPA é seu tempo de aprendizagem e o tempo de desenvolvimento das aplicações. Nesse contexto, as automatizações com UiPath levaram em média 7 min para serem desenvolvidas, com situações de tempo máximo e mínimo de cerca de 31 min e 2 min, respectivamente. Por outro lado, as automatizações desenvolvidas em Python levaram em média 1h e 17min, com situações de tempo máximo de 4h e 25min e mínimo de 11min.

Neste projeto piloto, a versão da ferramenta RPA utilizada é uma versão gratuita disponibilizada para estudantes. No entanto, o autor acredita que os resultados alcançados indicam que a compra de uma licença comercial pela empresa seria justificada tendo em vista os benefícios que podem ser alcançados com a escalabilidade da ferramenta, a possibilidade de replicação dos testes nas aplicações em tecnologias *mobile* (Android e iOS). Além disso, como já mencionado, essa tecnologia é versátil e possui recursos que

poderiam ser empregados em outros setores da empresa como: recursos humanos, departamento financeiro e de comunicação, o que produziria uma economia de tempo e consequente economia de dinheiro, liberando os colaboradores para realizarem tarefas que exigem um maior nível de criatividade.



## 6 REFERÊNCIAS

ALBERTH, MARKUS; MATTERN, MICHAEL. Understanding robotic process automation (RPA). **the Capco Institute Journal of Financial Transformation**, v. 46, p. 54–61, 2017. .

ALTEXSOFT. The Good and the Bad of Selenium Test Automation Software. 2021. Available at: <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-selenium-test-automation-tool/>. Acesso em: 26 out. 2021.

AMINI, Borhan. **Robotic Process Automation: Implementation within an organization**. 2019. 1–71 f. JAMK University of Applied Sciences, 2019.

AMMATTIKORKEAKOULU, Metropolia. Oskar Ylönen Blue Prism-ja UiPath-vertailu ohjelmistorobotii-kassa. 2018. Available at: [https://www.theseus.fi/bitstream/handle/10024/146100/Ylonen\\_Oskar.pdf?sequence=1](https://www.theseus.fi/bitstream/handle/10024/146100/Ylonen_Oskar.pdf?sequence=1).

ANAGNOSTE, Sorin. Robotic Automation Process - The next major revolution in terms of back office operations improvement. **Proceedings of the International Conference on Business Excellence**, v. 11, n. 1, p. 676–686, 2017. <https://doi.org/10.1515/picbe-2017-0072>.

ANSARI, Wasique Ali; DIYA, Paritosh; PATIL, Sahishnu; PATIL, Sunita. A Review on Robotic Process Automation - The Future of Business Organizations. **SSRN Electronic Journal**, 2019. <https://doi.org/10.2139/ssrn.3372171>.

ANTAS, Mikael. **ROBOTIC PROCESS AUTOMATION FOR BUSINESS CASES – A REVIEW**. 2019. 1–22 f. Resultados da pesquisa Resultado da Web com links de sites Arcada University of Applied Sciences, 2019.

APPIUM. 2021. Available at: <https://appium.io/>. Acesso em: 1 out. 2021.

AUTOMATION ANYWHERE. AUTOMAÇÃO ROBÓTICA DE PROCESSOS (RPA). 2021a. Available at: <https://www.automationanywhere.com/br/rpa/robotic-process-automation>. Acesso em: 12 out. 2021.

AUTOMATION ANYWHERE. Gartner - Automation Anywhere. 2021b. Available at: <https://www.automationanywhere.com/images/mq-rpa-graphic-2021.jpg>. Acesso em: 10 nov. 2021.

BHATNAGAR, Nitu. **Role of Robotic Process Automation in Pharmaceutical Industries**. [S. l.]: Springer International Publishing, 2020. v. 921, . DOI 10.1007/978-3-030-14118-9\_50. Available at: [http://dx.doi.org/10.1007/978-3-030-14118-9\\_50](http://dx.doi.org/10.1007/978-3-030-14118-9_50).

BISHT, Sumit. **Robot Framework Test Automation**. [S. l.: s. n.], 2021. Available at: [https://books.google.com.br/books?id=\\_RO8AQAAQBAJ&printsec=frontcover&hl=pt-BR#v=onepage&q=selenium&f=false](https://books.google.com.br/books?id=_RO8AQAAQBAJ&printsec=frontcover&hl=pt-BR#v=onepage&q=selenium&f=false).

CAETANO, Cristiano. TestComplete. 2021. Available at: <https://www.devmedia.com.br/artigo-clube-delphi-102-testcomplete/11758>. Acesso em: 21 dez. 2021.

CERNAT, Marina; STAIUCU, Adelina Nicoleta; STEFANESCU, Alin. Improving UI test automation using robotic process automation. **ICSOF 2020 - Proceedings of the 15th International Conference on Software Technologies**, , p. 260–272, 2020. <https://doi.org/10.5220/0009911202600267>.

CHACÓN-MONTERO, J.; JIMÉNEZ-RAMÍREZ, A.; ENRÍQUEZ, J. G. Towards a method for automated testing in robotic process automation projects. **Proceedings - 2019 IEEE/ACM 14th International Workshop on Automation of Software Test, AST 2019**, , p. 42–47, 2019. <https://doi.org/10.1109/AST.2019.00012>.

CLAIRE DRUMOND. O que é SCRUM? 2021. **ATLASSIAN**. Available at: <https://www.atlassian.com/br/agile/scrum>. Acesso em: 1 out. 2021.

CUCUMBER. Gherkin. 2021. Available at: <https://cucumber.io/docs/gherkin/reference/>. Acesso em: 10 out. 2021.

CYPRESS. 2021. Available at: <https://www.cypress.io/>. Acesso em: 1 out. 2021.

GEYER-KLINGEBERG, Jerome; NAKLADAL, Janina; BALDAUF, Fabian; VEIT, Fabian. Process Mining and Robotic Process Automation: A Perfect Match Process Mining as Enabler for RPA Implementation. **16th International Conference on Business Process Management**, v. i, n. July, 2018. .

GRAPE, Victor. Comparing Costs of Browser Automation Test Tools with Manual Testing. , p. 52, 2016. Available at: <http://www.ep.liu.se/>.

HIMANSHI, Himanshi; UMESH, Nitin; SARASWAT, Amar. Automation Testing: An Introduction to Selenium. **International Journal of Computer Applications**, v. 119, n. 3, p. 49–51, 2015. <https://doi.org/10.5120/21051-3797>.

HUANG, Feiqi. Three Essays on Emerging Technologies in Accounting. **Journal of Chemical Information and Modeling**, v. 01, n. 01, p. 1689–1699, 2013. .

IEEE CORPORATE ADVISORY GROUP. Guide for Terms and Concepts in Intelligent Process Automation (IEEE). **IEEE Std 2755-2017**, , p. 14, 2017. .

KABLAN, Ali. **Dark Factories from an Industry 4.0 Perspective: Its Effects on Cost Accounting and Managerial Accounting**. [S. l.: s. n.], 2020. [https://doi.org/10.1007/978-3-030-29739-8\\_24](https://doi.org/10.1007/978-3-030-29739-8_24).

KARIPPUR, Nanda Kumar. A study of the impact on customer experience in retail banking industry. **Journal of Internet Banking and Commerce**, v. 23, n. 2, 2018. Available at: <http://www.icommercecentral.com>.

KHAN, Sameera. Comparative Analysis of Rpa Tools-Uipath, Automation Anywhere and Blueprism. **International Journal of Computer Science and Mobile Applications**, v. 8, n. 11, p. 1–6, 2020. <https://doi.org/10.47760/ijcsma.2020.v08i11.001>.

KRYON SYSTEMS. Kryon. 2021. Available at: <https://www.kryonsystems.com/about-us/>. Acesso em: 11 out. 2021.

LAMBERTON, Chris; BRIGO, Damiano; HOY, Dave. Impact of Robotics, RPA and AI on the insurance industry: challenges and opportunities. **Journal of Financial Perspectives**, v. 4, n. 1, p. 8–20, 2017. .

LE CLAIR, Craig. The Forrester Wave™: Robotic Process Automation, Q2 2018 - The 15 Providers That Matter Most And How They Stack Up. **Forrester Research, Inc.**, 2018. .

LEAL, Fábio Barros. Uma abordagem usando features BDD e Modelo de Objetivos para o desenvolvimento ágil de software. 2019. Available at: <https://repositorio.unb.br/handle/10482/37458>.

MADAKAM, Somayya; HOLMUKHE, Rajesh M.; KUMAR JAISWAL, Durgesh. The Future Digital Work Force: Robotic Process Automation (RPA). **Journal**

**of Information Systems and Technology Management**, v. 16, p. 1–17, 30 jan. 2019. DOI 10.4301/S1807-1775201916001. Available at: <http://jistem.tecsi.org/index.php/jistem/article/view/3077/707>.

MARTINS, Ricardo José Salvado. **BENEFÍCIOS DA IMPLEMENTAÇÃO DE UM CENTRO DE EXCELÊNCIA DE ROBOTIC PROCESS AUTOMATION**. 2018. 1–50 f. Universidade de Lisboa, 2018.

MARUTI TECHLABS. RPA in Supply Chain - The Key to SCM Success. 2021. Available at: <https://marutitech.com/rpa-in-supply-chain/>. Acesso em: 1 nov. 2021.

MATSUSHITA, FREDERICO MASSARU ABE. **Desafios na Implementação de projetos RPA Uma Abordagem Exploratória em Instituição Financeira**. 2019. 1–50 f. FUNDAÇÃO GETULIO VARGAS, 2019.

MICROSOFT POWER AUTOMATE. WinAutomation. 2021. Available at: <https://www.winautomation.com/>. Acesso em: 5 nov. 2021.

MODI, Diviyansh; WAZIR, Aaliya; KAUR, Bisman; AGARWAL, Chinmay. RPA In Finance: Acceptance & Implication of RPA Based Algorithm in Indian Eco-Space. **SPAST Abstracts**, v. 1, 2021. Available at: <https://spast.org/techrep/article/view/1611>.

MOFFITT, Kevin C.; ROZARIO, Andrea M.; VASARHELYI, Miklos A. Robotic Process Automation for Auditing. **Journal of Emerging Technologies in Accounting**, v. 15, n. 1, p. 1–10, 1 jul. 2018. DOI 10.2308/jeta-10589. Available at: <https://meridian.allenpress.com/jeta/article/15/1/1/9413/Robotic-Process-Automation-for-Auditing>.

MULLAKARA, Nandan. Selecting the right RPA tools for your use case. 2021a. Available at: <https://nandan.info/selecting-the-right-rpa-tools-for-your-use-case/>. Acesso em: 1 nov. 2021.

MULLAKARA, Nandan. Top Intelligent Automation. 2021b. Available at: <https://nandan.info/top-robotic-process-automation-rpa-tools-list/>. Acesso em: 10 out. 2021.

MULLAKARA, NANDAN. Top Seven RPA challenges you need to be aware of. 2021c. Available at: <https://nandan.info/robotic-process-automation-rpa-challenges/>.

Acesso em: 14 out. 2021.

MUURINEN, Juho. **Robotic process automation as an automation tool for improving purchasing processes – Case study**. 2019. 1–84 f. LUT University, 2019.

OSHRI, Ilan; PLUGGE, Albert. Introducing RPA and automation in the financial sector: Lessons from KAS Bank. **Journal of Information Technology Teaching Cases**, 2021. <https://doi.org/10.1177/2043886921994828>.

PAPAGEORGIU, Dimitris. Transforming the HR Function Through Robotic Process Automation: EBSCOhost. **Benefits Quarterly**, v. 34, p. 27–30, 2018. Available at: <http://web.b.ebscohost.com/ehost/detail/detail?vid=0&sid=42d055c4-f261-42f9-b02a-cbaabe2718b8%40sessionmgr102&bdata=JnNpdGU9ZWhvc3QtbGl2ZSZzY29wZT1zaXRl#AN=129603839&db=bth>.

PATIL, Renu; TEMKAR, Rohini. Intelligent Testing Tool : Selenium Web Driver. **International Research Journal of Engineering and Technology(IRJET)**, v. 4, n. 6, p. 1920–1923, 2017. Available at: <https://irjet.net/archives/V4/i6/IRJET-V4I6599.pdf>.

PRIME CONTROL. 10 ferramentas de automação de testes mais usadas. 2021. Available at: <https://www.primecontrol.com.br/10-ferramentas-de-automacao-de-testes-mais-usadas/>. Acesso em: 26 out. 2021.

RATIA, M.; MYLLÄRNIEMI, J.; HELANDER, N. Robotic process automation - Creating value by digitalizing work in the private healthcare? **ACM International Conference Proceeding Series**, , p. 222–227, 2018. <https://doi.org/10.1145/3275116.3275129>.

RIBEIRO, Jorge; LIMA, Rui; ECKHARDT, Tiago; PAIVA, Sara. Robotic Process Automation and Artificial Intelligence in Industry 4.0 - A Literature review. **Procedia Computer Science**, v. 181, n. 2019, p. 51–58, 2021. DOI 10.1016/j.procs.2021.01.104. Available at: <https://doi.org/10.1016/j.procs.2021.01.104>.

ROBOT FRAMEWORK. Robot Framework. 2021. Available at: <https://robotframework.org/>. Acesso em: 1 out. 2021.

ROMAO, Mario; COSTA, Joao; COSTA, Carlos J. Robotic process automation:

A case study in the banking industry. **Iberian Conference on Information Systems and Technologies, CISTI**, v. 2019-June, n. June, p. 1–6, 2019. <https://doi.org/10.23919/CISTI.2019.8760733>.

ROZARIO, Andrea M. Three essays on audit innovation using social media information and disruptive technologies to enhance audit quality. n. May, p. 218, 2019. .

SANTOS, Filipa da Silva. Using a test automation tool for robotic process automation: an empirical study. n. July, 2019a. Available at: <http://hdl.handle.net/10071/20188>.

SANTOS, Filipa da Silva. Using a test automation tool for robotic process automation: an empirical study. n. July, 2019b. .

SCHEPPLER, Björn; WEBER, Christian. Robotic Process Automation & Human Resources. **Informatik-Spektrum**, v. 43, n. 2, p. 152–156, 2020. <https://doi.org/10.1007/s00287-020-01263-6>.

SCHMIDER, Juergen; KUMAR, Krishan; LAFOREST, Chantal; SWANKOSKI, Brian; NAIM, Karen; CAUBEL, Patrick M. Innovation in Pharmacovigilance: Use of Artificial Intelligence in Adverse Event Case Processing. **Clinical Pharmacology and Therapeutics**, 2018. <https://doi.org/10.1002/cpt.1255>.

SELENIUM. 2021. Available at: <https://www.selenium.dev/about/>. Acesso em: 1 out. 2021.

SINGH À, Shiwangi; GADGIL À, Rucha; CHUDGOR À, Ayushi. Automated Testing of Mobile Applications using Scripting Technique: A Study on Appium. **International Journal of Current Engineering and Technology India Accepted**, v. 362744, n. 55, p. 3627–3630, 2014. Available at: <http://inpressco.com/category/ijcet>.

SMEETS, Mario; ERHARD, Ralph; KAUSSLER, Thomas. **Robotic Process Automation (RPA) in the Financial Sector**. [S. l.: s. n.], 2021. <https://doi.org/10.1007/978-3-658-32974-7>.

SOFTWARE TESTING HELP. Top 20 Best Automation Testing Tools In 2021. 2021. Available at: <https://www.softwaretestinghelp.com/top-20-automation-testing-tools/>. Acesso em: 26 out. 2021.

SOUSA, Sueila. Dominando o Selenium Web Driver na prática. 2021. Available at: <https://www.devmedia.com.br/dominando-o-selenium-web-driver-na-pratica/34183>. Acesso em: 1 out. 2021.

STATION, IT Central. Robotic Process Automation (RPA) Buyer's Guide and Reviews. **Buyer's Guide and Reviews**, n. September, 2019. Available at: <https://www.itcentralstation.com/categories/robotic-process-automation-rpa>.

SULLIVAN, Mac; SIMPSON, Walter; LI, Wesley. The Role of Robotic Process Automation (RPA) in Logistics. **The Digital Transformation of Logistics**, , p. 61–78, 2021. <https://doi.org/10.1002/9781119646495.ch5>.

SYED, Rehan; SURIADI, Suriadi; ADAMS, Michael; BANDARA, Wasana; LEEMANS, Sander J.J.; OUYANG, Chun; TER HOFSTEDÉ, Arthur H.M.; VAN DE WEERD, Inge; WYNN, Moe Thandar; REIJERS, Hajo A. Robotic Process Automation: Contemporary themes and challenges. **Computers in Industry**, v. 115, p. 103162, 2020. DOI 10.1016/j.compind.2019.103162. Available at: <https://doi.org/10.1016/j.compind.2019.103162>.

TAKY, Malika Tasnim. AUTOMATED TESTING WITH CYPRESS. , p. 1–11, 2009. .

THE ROBOTIC WORKFORCE. RPA ROI. 2021. Available at: <http://www.theroboticworkforce.com/resources/rpa-roi>. Acesso em: 1 out. 2021.

UIPATH. Como a UiPath funciona. 2021. Available at: <https://www.uipath.com/pt/>. Acesso em: 1 out. 2021.

UIPATH - TEST SUITE. 2021. Available at: <https://www.uipath.com/pt/product/test-suite>. Acesso em: 1 out. 2021.

VAN DER AALST, Wil M. P.; BICHLER, Martin; HEINZL, Armin. Robotic Process Automation. **Business & Information Systems Engineering**, v. 60, n. 4, p. 269–272, 14 ago. 2018. DOI 10.1007/s12599-018-0542-4. Available at: <http://link.springer.com/10.1007/s12599-018-0542-4>.

VISHNU, Sandeep; AGOCHIYA, Vipul; PALKAR, Ranjit. Data-centered dependencies and opportunities for robotics process automation in bankin. **Journal of Financial Transformation**, v. 45, p. 68–76, 2017. DOI <https://doi.org/10.1108/BIJ-01->

2021-0033. Available at: <https://www.emerald.com/insight/content/doi/10.1108/BIJ-01-2021-0033/full/html>.

WORKFUSION. Workfusion. 2021. Available at: <https://www.workfusion.com/>. Acesso em: 10 out. 2021.

YATSKIV, Solomiya; VOYTYUK, Iryna; YATSKIV, Nataliia; KUSHNIR, Oksana; TRUFANOVA, Yuliia; PANASYUK, Valentyna. Improved Method of Software Automation Testing Based on the Robotic Process Automation Technology. **2019 9th International Conference on Advanced Computer Information Technologies, ACIT 2019 - Proceedings**, , p. 293–296, 2019. <https://doi.org/10.1109/ACITT.2019.8780038>.

YIN, Robert K. **Case Study Research: Design and Methods**. 4th ed. [S. l.]: SAGE Publications, 2009.



## 7 ANEXOS

Abaixo temos um pseudocódigo gerado pelo UiPath de um dos casos de testes.

```

1.1 Caso de Teste (Sequence)
  Private = False
  Activities
    1.2 Open Browser (OpenBrowser)
      Url = https://xxxxxxxxxxxxxxxxx.dev
      Private = True
      NewSession = True
      BrowserType = Chrome
      Private = False
      Body
        1.3 Do (Sequence)
          Private = False
          Activities
            1.38 Click 'A' (Click)
              KeyModifiers = None
              CursorPosition
                Posição = Center
              ClickType = CLICK_SINGLE
              MouseButton = BTN_LEFT
              Target
                Selector=<webctrl aaname=' Entrar' tag='A' />
              Private = False
            1.34 Set Text 'INPUT signInName' (SetValue)
              Text=teste@mailinator.com
              Target
                Selector=<webctrl tag='INPUT' type='text' />
              Private = False
            1.30 Set Text 'INPUT password' (SetValue)
              Text = teste123
              Target
                Selector=<webctrl tag='INPUT' type='password' />
              Private = False
            1.25 Click 'BUTTON newSubmitBtn' (Click)
              KeyModifiers = None
              CursorPosition
                Posição = Center
              ClickType = CLICK_SINGLE
              MouseButton = BTN_LEFT
              Target
                Selector=<webctrl tag='BUTTON' type='button' />
              Private = False
            1.15 Click 'A' (Click)
              KeyModifiers = None
              CursorPosition
                Posição = Center
              ClickType = CLICK_SINGLE
              MouseButton = BTN_LEFT
              Target
                Selector = <webctrl parentid='ui-id-5' tag='SPAN' />
              Private = False
            1.15 Click 'A' (Click)
              KeyModifiers = None
              CursorPosition
                Posição = Center
              ClickType = CLICK_SINGLE
              MouseButton = BTN_LEFT
              Target
                Selector = <webctrl aaname='Combo'idx='1' parentid='maincontent'
tag='A' />
              Private = False
            1.10 Click 'A https://xxx-xxx...' (Click)
              KeyModifiers = None
              CursorPosition
                Posição = Center
              ClickType = CLICK_SINGLE
              MouseButton = BTN_LEFT
              Target
                Selector=<webctrl aaname=' DM-61984 type free good '
parentid='maincontent' tag='A' />
              Private = False
            1.8 Element Exists 'IMG' (UiElementExists)
              Target
                Selector=<webctrl tableRow='1' tag='IMG' />
              ResultType = System.Boolean
              Private = False

```