



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO, DE CIÊNCIAS EXATAS E EDUCAÇÃO
DEPARTAMENTO DE ENG. DE CONTROLE, AUTOMAÇÃO E COMPUTAÇÃO
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Luiz Eduardo Pontizelli

Automação e consulta do agendamento de cargas utilizando RPA e *Power Apps*

Blumenau
2022

Luiz Eduardo Pontizelli

Automação e consulta do agendamento de cargas utilizando RPA e *Power Apps*

Trabalho de Conclusão de Curso de Graduação em Engenharia de Controle e Automação do Centro Tecnológico, de Ciências Exatas e Educação da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Engenheiro de Controle e Automação.
Orientador: Prof. Orientador, Dr. Maiquel de Brito

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Pontizelli, Luiz Eduardo

Automação e consulta do agendamento de cargas utilizando
RPA e Power Apps / Luiz Eduardo Pontizelli ; orientador,
Maiquel de Brito, 2022.

70 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus Blumenau,
Graduação em Engenharia de Controle e Automação, Blumenau,
2022.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Automação
Agendamento de Cargas. 3. RPA. 4. Blue Prism. 5. Power
Apps. I. Brito, Maiquel de. II. Universidade Federal de
Santa Catarina. Graduação em Engenharia de Controle e
Automação. III. Título.

Luiz Eduardo Pontizelli

Automação e consulta do agendamento de cargas utilizando RPA e *Power Apps*

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Engenheiro de Controle e Automação” e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação.

Blumenau, 28 de Julho de 2022.

Banca Examinadora:

Prof. Dr. Maiquel de Brito
Universidade Federal de Santa Catarina

Prof. Dr. Marcos Vinicius Matsuo
Universidade Federal de Santa Catarina

Prof. Dr. Mauri Ferrandin
Universidade Federal de Santa Catarina

Este trabalho é dedicado aos meus pais, pois graças ao esforço deles posso concluir minha graduação.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, Luiz e Janete, ao meu irmão, Alan, e aos meus avós, que me apoiaram e forneceram todo o suporte necessário ao longo de todo meu trajeto, em especial, durante o período de graduação. Todos os meus valores e princípios vieram dos ensinamentos que me passaram.

Agradeço também a minha namorada Larissa por todos os momentos em que passamos juntos, por dividir seus sonhos comigo, por me apoiar e me fazer querer conquistar todos os objetivos.

Aos meus amigos, em especial Matheus e Uériton, que moraram comigo ao longo da graduação. Agradeço por compartilharem um pouco da experiência que já tiveram, pelos seus conselhos e por me auxiliarem quando precisei.

A todos os meus professores, principalmente ao meu professor orientador Dr. Maiquel de Brito, por repassar os seus conhecimentos, por me auxiliar durante a graduação e por me fazer enxergar o quão importante é a educação. Agradeço também a Universidade Federal de Santa Catarina por proporcionar os ensinamentos para seguir minha jornada profissional.

Por último, a empresa Bunge, pela oportunidade de concretizar este trabalho com as ferramentas e ensinamentos que obtive enquanto estive estagiando.

“Nossa maior fraqueza é a desistência. O caminho mais certo para o sucesso é sempre tentar apenas uma vez mais.”

(Thomas Edison)

RESUMO

O agendamento de cargas, que faz parte do processo logístico, é necessário para controlar a entrada e saída de produtos e materiais nas empresas, garantindo fluxo efetivo de carga e descarga. Embora existam inúmeras formas de automatizar os processos dentro da cadeia de suprimentos, é muito comum que essas tarefas sejam realizadas por operadores humanos. Existe um grande volume de cargas que devem ser agendadas para inúmeros clientes, submetendo os funcionários a tarefas repetitivas com alta frequência. Além disso, o grande volume dessa atividade provoca problemas ergonômicos, desgaste físico e mental, que podem acabar gerando falhas humanas de operação no processo. Uma possível solução para estes problemas é automatizar todo o processo de agendamento de cargas utilizando RPA (Robotic Process Automation), que pode gerenciar e montar as cargas de acordo com os pedidos e produtos, sem intervenção humana. Assim, neste trabalho é desenvolvida uma aplicação com RPA, que é uma das formas de automatizar a atividade e fazê-la mais eficiente, além de padronizar os dados e relatórios do processo. As consultas das cargas e produtos agendados através do RPA são realizadas utilizando um aplicativo (desenvolvido no *Power Apps*) para execução em *smartphones*, no qual é possível consultar e ajustar todos os itens que foram processados pela automação. Os resultados deste trabalho indicam ganhos para a empresa em que a automação foi aplicada, uma vez que houve redução de aproximadamente 66,6% no tempo de execução da tarefa, diminuição de erros e redução de custos com recursos humanos.

Palavras-chave: Agendamento de Cargas; RPA; *Power Apps*.

ABSTRACT

Load scheduling, which is part of the logistical process, is necessary to control the entry and exit of products and materials in companies, ensuring an effective flow of loading and unloading. Although there are numerous ways to automate processes within the supply chain, it is normal that these tasks are done by human operators. There is a large volume of loads that must be scheduled for numerous customers, causing employees to be subjected to repetitive tasks with high frequency. A possible solution to these problems is to automate the entire process of load scheduling using RPA (Robotic Process Automation), which can manage and mount loads according to orders and products, without human intervention. Thus, it is developed in this report an application using RPA, which is a way to automate the activity and make it more efficient, in addition to standardizing the data and reports of the process. The queries of the loads and products scheduled through RPA are performed using an application (developed in Power Apps) to run on smartphones, which it is possible to consult and adjust all the items that were processed by the automation. The results of this project indicate gains for the company in which the automation was applied, once there was a reduction of approximately 66.6% in the timely execution of the tasks, reduction of errors and reduction of costs with human resources.

Keywords: Load Scheduling; RPA; Power Apps.

LISTA DE FIGURAS

Figura 1 – Quadrante mágico de Gartner RPA.	23
Figura 2 – Atributos do botão OK no Application Modeller	24
Figura 3 – Estrutura de desenvolvimento do BP.	25
Figura 4 – Estrutura de um processo no BP.	26
Figura 5 – Tela Inicial.	31
Figura 6 – Fluxograma das atividades no portal.	33
Figura 7 – Planilha de agendamento de cargas do cliente Atacadão.	33
Figura 8 – Página de <i>login</i> do portal Atacadão.	34
Figura 9 – Página inicial do portal Atacadão.	34
Figura 10 – Página montagem e agendamento de carga.	35
Figura 11 – Planilha de agendamentos filtrada de acordo com os pedidos da mesma loja.	36
Figura 12 – Janela de inclusão de pedidos na carga.	37
Figura 13 – Fluxo do processo.	37
Figura 14 – Desenho do processo no Blue Prism.	39
Figura 15 – Página Start Up - Atacadão.	40
Figura 16 – Página Open Cargo.	42
Figura 17 – Página Exclude Unecessary Items.	43
Figura 18 – Página Check Items Cargo.	44
Figura 19 – Página Save.	46
Figura 20 – Inputs subprocesso Update Status SAP.	48
Figura 21 – Inputs subprocesso Update Status Wb.	49
Figura 22 – Fila Filho - Pedidos da Carga.	51
Figura 23 – Página Mark Child Item As Exception.	51
Figura 24 – Check Childs In Exception.	52
Figura 25 – Tabela de Cargas.	54
Figura 26 – Tabela de Produtos.	55
Figura 27 – Tela da Lista de Cargas.	56
Figura 28 – Função Aplicada Sobre a Galeria de Cargas.	56
Figura 29 – Tela de Detalhes da Carga.	57
Figura 30 – Função Aplicada Sobre a Galeria de Produtos.	58
Figura 31 – Tela de Produtos da Carga.	58
Figura 32 – Tela de Edição de Cargas.	59
Figura 33 – Tela de Edição de Produtos.	59
Figura 34 – Fila Pai - Cargas.	60
Figura 35 – Tela Inicial.	62
Figura 36 – Telas da Lista de Cargas e Produtos.	63

Figura 37 – Tela de Detalhes da Carga.	64
Figura 38 – Telas de Edição de Cargas e Produtos.	64
Figura 39 – Estágios do <i>Object Studio</i> e <i>Process Studio</i>	70

LISTA DE QUADROS

Quadro 1 – Distribuição de cargas em veículos.	47
--	----

LISTA DE TABELAS

Tabela 1 – Estágios para desenvolvimento de automações no <i>software</i> BP.	27
---	----

LISTA DE ABREVIATURAS E SIGLAS

BP	Blue Prism
CS	Customer Service
ERP	Enterprise Resource Planning
FTE	Full-Time Equivalent
RPA	Robotic Process Automation
SAP	Systemanalysis Programmentwicklung
SSC	Shared Services Center
VBA	Visual Basic for Applications
VBO	Visual Business Object

LISTA DE SÍMBOLOS

V	Volume anual
T	Tempo de execução
J	Jornada de trabalho anual

SUMÁRIO

1	INTRODUÇÃO	17
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	ROBOTIC PROCESS AUTOMATION (RPA)	19
2.1.1	Funcionamento do RPA	19
2.1.1.1	<i>Work Queues</i>	20
2.1.1.2	<i>Control Room</i>	20
2.1.1.3	<i>Schedules</i>	20
2.1.1.4	<i>Exception Handling</i>	21
2.1.2	Características do RPA	21
2.1.3	Ferramentas RPA	22
2.1.3.1	<i>Blue Prism</i>	22
2.1.3.1.1	VBO (Visual Business Object)	24
2.1.3.1.2	Process	25
2.1.3.1.3	Estágios do BP	27
2.2	AGENDAMENTO DE CARGAS E CADEIA DE SUPRIMENTOS	27
2.2.1	Automação do Agendamento de Cargas	28
2.3	SAP	29
2.3.1	Gerenciamento de Cargas no SAP da Bunge	29
2.4	POWER APPS	30
2.4.1	Estrutura do Power Apps	30
3	DESENVOLVIMENTO	32
3.1	APLICAÇÕES DA AUTOMAÇÃO	32
3.2	FLUXO NO PORTAL ATACADÃO	33
3.3	FLUXO DA AUTOMAÇÃO COM RPA	37
3.4	DESENVOLVIMENTO DO PROJETO DE AUTOMAÇÃO	38
3.4.1	Agendamento no Portal	38
3.4.1.1	<i>Conexão no Portal</i>	38
3.4.1.2	<i>Organização dos Pedidos em Cargas</i>	41
3.4.1.3	<i>Abrir Carga</i>	42
3.4.1.4	<i>Montar Carga</i>	43
3.4.1.5	<i>Salvar Carga</i>	45
3.4.2	Subprocessos	47
3.4.2.1	<i>Subprocesso Update Status SAP</i>	48
3.4.2.2	<i>Subprocesso Update Status WB</i>	49
3.4.3	Tratamento das Exceções	50
3.4.4	Página de Relatórios	53
3.4.5	Estágios Complementares da Página Inicial	53

3.5	POWER APPS	54
3.5.1	Fonte de Dados	54
3.5.2	Desenvolvimento do Aplicativo	55
<i>3.5.2.1</i>	<i>Página Inicial</i>	<i>55</i>
<i>3.5.2.2</i>	<i>Lista de Cargas</i>	<i>55</i>
<i>3.5.2.3</i>	<i>Detalhes da Cargas</i>	<i>57</i>
<i>3.5.2.4</i>	<i>Produtos da Carga</i>	<i>57</i>
<i>3.5.2.5</i>	<i>Editar Carga e Produto</i>	<i>58</i>
4	RESULTADOS	60
4.1	TEMPO DE EXECUÇÃO	60
4.2	NÍVEL DE EXCEÇÃO	61
4.3	REDUÇÃO DE CUSTOS	61
4.4	APLICATIVO DE CONSULTAS	62
5	CONCLUSÃO	65
	REFERÊNCIAS	67
	ANEXO A – Estágios <i>Blue Prism</i>	70

1 INTRODUÇÃO

A cadeia de suprimentos é parte fundamental das atividades desenvolvidas por uma organização. Ela consiste em uma série de tarefas preestabelecidas para que os materiais percorram sua trajetória desde os fornecedores iniciais até os clientes finais (WATERS, 2003). O agendamento de cargas e pedidos é uma etapa fundamental no processo de logística, que por sua vez, é um dos pontos em destaque dentro da cadeia de suprimentos. Portanto, o agendamento de cargas consiste em gerir a entrada e saída de produtos nas empresas, administrar as quantidades de materiais que são despachados, bem como verificar as datas de entrega nas organizações clientes.

O controle de entrada e saída de produtos nas empresas normalmente é realizado de forma manual através de um operador humano, que tem o agendamento de cargas como uma importante tarefa de sua rotina. Contudo, existem situações em que o gerenciamento do agendamento de cargas não é realizado de forma correta e isso pode causar problemas para as empresas, como: gargalos pelo grande volume de cargas, filas de espera, fluxo menos efetivo e etc (TRACKAGE, 2022). Além disso, os funcionários são submetidos a tarefas repetitivas que exigem alto grau de atenção com muita frequência. Essa situação provoca problemas ergonômicos para o trabalhador causando principalmente desgaste mental e por consequência, desgaste físico. Ademais, atividades contínuas repetitivas provocam também muitas falhas humanas de operação, causando uma cadeia de erros que podem acarretar em prejuízo financeiro para a instituição (ORGANISATION, 2010).

Baseado nestas situações, o objetivo deste trabalho é o desenvolvimento de um sistema automatizado que substitua o trabalho manual repetitivo por um robô que desempenhe as mesmas tarefas de um operador humano, implementando a automação do agendamento de cargas diretamente no portal das lojas Atacadão, clientes da empresa Bunge. Além disso, objetiva-se desenvolver um aplicativo para consultas de forma rápida das cargas que a automação agenda. Automatizar esta etapa da cadeia de suprimentos elimina os problemas descritos anteriormente e traz benefícios, como: minimização dos erros, aumento da velocidade de processamento, redução dos custos de um operador humano, entre outros.

O sistema de automação é construído através do RPA (Robotic Process Automation), que é uma ferramenta de automação que interage com as aplicações da mesma forma que faz o usuário, sendo uma automação cujo robô não é físico, mas sim, um *software* que opera outras aplicações em *desktop* e servidores. Os robôs são projetados para seguir diferentes situações e cenários que podem ocorrer ao longo do processo (FANTINA; STOROZHUK; GOYAL, 2021). De forma breve, o RPA segue uma sequência lógica de passos e tarefas que surgem através dos *inputs* do processo. Sendo assim, o robô interage com *softwares* e sistemas de acordo com cada cenário em que a automação é projetada, substituindo portanto o operador em questão que realiza a tarefa de forma manual. O

RPA é aplicado dentro das empresas em processos como extração e processamento de relatórios, processamento de transações, manipulação de dados, disparo de mensagens, entre outros (FANTINA; STOROZHUK; GOYAL, 2021).

A solução descrita neste trabalho é baseada na utilização do *software* BP (Blue Prism) como ferramenta de RPA para automatizar o agendamento de cargas. As atividades desenvolvidas pelo robô são a organização de todos os produtos e pedidos em forma de carga, acesso ao portal Atacadão para ajustar as cargas e montá-las de forma adequada e, por fim, execução do agendamento das cargas para cada loja que solicita os produtos. Além disso, a automação também faz a marcação de todos os produtos processados no sistema de gestão corporativo da empresa, que é o SAP (Systemanalysis Programmentwicklung), e também atualiza a planilha de agendamentos, para melhorar o controle e gerenciamento da área de CS (Customer Service) nesta etapa da cadeia. Para complementar a automação, a solução proposta neste trabalho também inclui desenvolvimento de um aplicativo no Power Apps que objetiva realizar consultas e ajustes sobre todos os produtos, pedidos e cargas atuadas pela automação, sendo esse um meio rápido e prático para buscar todas as informações processadas com sucesso pelo RPA.

A aplicação do RPA neste trabalho é o desenvolvimento de um projeto de automação solicitado pela área de CS do SSC (Shared Services Center) da empresa Bunge, localizada em Gaspar/SC. O SSC de Gaspar reúne serviços Fiscais, Contabilidade, Recursos Humanos, Jurídico, Tecnologia da Informação, Serviço ao Cliente, entre outros. A Bunge em um contexto geral é uma empresa multinacional que opera em mais de 40 países, com sede em St. Louis, Missouri nos Estados Unidos. Suas atividades estão relacionadas ao ramo do agronegócio, alimentos e bioenergia (BUNGE, 2022).

O restante deste documento está estruturado da seguinte forma: o Capítulo 2 apresenta a fundamentação teórica necessária à compreensão deste trabalho; o Capítulo 3 descreve o desenvolvimento da solução proposta; o Capítulo 4 descreve os resultados obtidos; e, por fim, o Capítulo 5 apresenta algumas conclusões sobre o trabalho desenvolvido. Além disso, ao final, são apresentadas as referências bibliográficas em que este trabalho se baseia, bem como um anexo com materiais adicionais.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a fundamentação teórica em que é apoiado este trabalho. As discussões utilizadas como apoio ao projeto podem ser encontradas facilmente nas literaturas citadas na Seção REFERÊNCIAS deste documento.

2.1 ROBOTIC PROCESS AUTOMATION (RPA)

O surgimento do RPA se deu pela necessidade do aumento da eficiência operacional nas atividades de escritório dentro das empresas. A adoção bem sucedida dessa tecnologia traz impactos positivos em metas estratégicas, produtividade e atendimento ao cliente. O RPA é de interesse para as indústrias que tradicionalmente têm a ideia de absorção rápida de tecnologias, visto que a demanda por este recurso está aumentando de forma acelerada por parte das organizações (SYED *et al.*, 2020).

O RPA é uma tecnologia baseada em um *software* que é capaz de emular atividades humanas de forma a interagir e gerenciar outros *softwares* e sistemas do computador. Apoiado nessa premissa, o robô tem o objetivo de replicar exatamente a mesma atividade que um operador humano faz (UIPATH, 2005). Sendo assim, o *software* pode associar o uso de inteligência artificial juntamente com a lógica de programação para tratar de problemas que exigem precisão e velocidade de processamento da tarefa. Além do mais, o robô pode acessar e utilizar todas as ferramentas disponíveis no computador, sendo *off-line* ou *on-line*, desde que o usuário do RPA tenha acesso. Segundo (Figurelli, 2016), “O RPA é o começo da entrada de robôs na gestão dos processos administrativos” (SILVA; BARION, 2018).

O mercado de RPA se mostra consistente com os investimentos das grandes empresas nesta tecnologia, estima-se que cerca de \$1,8 bilhões de dólares foram movimentados em 2021. Empresas como Microsoft, Google e IBM estão apostando fortemente neste setor, tornando-o cada vez mais competitivo e conseqüentemente aumentando seu valor de mercado. Contudo, Blue Prism, UiPath e Automation Anywhere foram pioneiras e até hoje são líderes de mercado desde o início do RPA (HAJJAR, 2021).

2.1.1 Funcionamento do RPA

O funcionamento do RPA tem como premissa replicar uma tarefa realizada por humanos através de ações e interações que o robô tem com *softwares* e sistemas. O projeto de automação com RPA deve ser baseado em atividades repetitivas e padronizadas para que seu funcionamento seja correto (MADAKAM; HOLMUKHE; JAISWAL, 2019).

A execução do robô funciona através de uma série de passos que pode ser representada através um fluxograma, sequência lógica programada pelo desenvolvedor.. O robô realiza uma ação de cada vez até o fim do fluxo do processo. Estas ações podem ser a

leitura de um determinado campo, escrita no campo, pressionamento de botão, entre outras.

Alguns princípios básicos são necessários para entender o funcionamento do RPA. De maneira geral, todo RPA trabalha com *Work Queues*, *Control Room*, *Schedule* e *Exception Handling* podendo variar a nomenclatura de acordo com a empresa desenvolvedora do *software*, mas mantendo o conceito de forma semelhante.

2.1.1.1 *Work Queues*

O RPA realiza as tarefas baseado nas informações da fila de trabalho, que é formada por uma coleção de dados que serão processados pelo robô. Cada item da fila tem propriedades e singularidades associadas a ele, bem como um parâmetro único para identificá-lo chamado de *Item-Key*.

A fila pode ser formada tanto por uma coleção de entrada no *input* do processo como por ações do robô no sistema em que ele está trabalhando. Depois que a fila é formada o robô começa a processar um item por vez, que passa por todo o fluxo do processo e é marcado como completo ou exceção (SCHULER; GEHRING, 2018).

2.1.1.2 *Control Room*

O *Control Room* é uma interface de gerenciamento dos robôs que estão em atividade, fornecendo uma visão unificada para administração centralizada de todas as automações. A funcionalidade do *Control Room* é justamente administrar os robôs em produção, realizar a execução e monitoramento de cada automação (COPES; MELLO, 2020).

Normalmente, cada empresa contratante das licenças de RPA tem uma equipe dedicada ao *Control Room* para dar apoio aos desenvolvedores dos robôs. Em muitos casos, é também função dessa equipe gerenciar usuários e credenciais para os robôs, realizar o agendamento das *schedules* e tratar das exceções.

2.1.1.3 *Schedules*

A programação através das *schedules* é uma forma alternativa de estipular o agendamento de cada automação ao invés de executá-la manualmente. A *schedule* é muito importante pois é uma forma de organização dos processos em produção, além de garantir que o robô inicie suas atividades no horário programado, sem a necessidade de um operador humano para executar as automações. Em muitos casos é fundamental que a automação seja iniciada no momento exato sem a possibilidade de atraso, pois o *output* do processo deve ser imediato.

2.1.1.4 *Exception Handling*

O fluxo da automação é normalmente desenhado para seguir um caminho exato, onde todas as ações e decisões tomadas pelo robô são consequências dos parâmetros de entrada e da fila de trabalho (MADAKAM; HOLMUKHE; JAISWAL, 2019). Contudo, em alguns casos, o robô se depara com situações que não foram mapeadas pelo desenvolvedor, situações que não fazem parte das regras de negócio do processo ou até mesmo situações que estão fora do escopo da automação. Estes casos são tratados como exceção do robô e podem variar de acordo com cada condição (SCHULER; GEHRING, 2018).

O *Exception Handling* é a estratégia utilizada para o tratamento de exceções da automação, que deve ser cuidadosamente programada para que se entenda a natureza das exceções e o correto tratamento para cada ocorrência. Primeiramente deve-se entender os tipos de exceções e motivos pelos quais elas podem acontecer. Pode-se categorizar as exceções da seguinte forma:

- *Business Exceptions*: São exceções relacionadas às regras de negócio do processo, nas quais o robô trava por informações de negócios incompletas ou ausentes. Ou seja, existem situações em que o robô interrompe sua atividade devido aos erros das regras de negócio definidas no projeto.
- *System/Application Exception*: São exceções sistêmicas que podem ocorrer devido a algum tipo de falha no sistema ou aplicação em que o robô está interagindo e assim, ele não consegue prosseguir adiante no fluxo do processo.
- *Internal Exception*: São exceções internas e acontecem no próprio *software* de RPA em casos onde há problemas que o robô não está programado para tratar.
- *Out of Scope*: São casos de exceções já previstas que o robô não deve prosseguir trabalhando, pois essa situação está fora do escopo da automação.

Conhecendo o contexto do problema que ocasiona a exceção, o tratamento pode ser realizado pelo próprio robô. Nestes casos é comum uma nova tentativa de realizar a atividade em que houve a exceção ou até mesmo seguir um novo fluxo alternativo do processo. Contudo, uma segunda alternativa é que o robô marque o item da fila em que está trabalhando com o tipo de exceção para que seja tratada manualmente de forma adequada.

2.1.2 Características do RPA

A implementação de uma automação utilizando RPA traz uma série de vantagens para quem utiliza essa tecnologia, agregando valores como: eficiência, qualidade, controle e vantagens competitivas (ACOBÁ *et al.*, 2019). Dentre esses valores, pode-se evidenciar:

- Jornada de trabalho: o robô pode operar durante o dia todo sem pausas;

- Redução de tempo: o RPA pode processar uma atividade com velocidade superior quando comparado com a velocidade de trabalho humano;
- Ganho de desempenho: quantidade de erros é muito baixa;
- Padronização das tarefas: a automação padroniza as tarefas tornando a atividade mais consistente e precisa;
- Segurança de dados: dados tratados de maneira segura sem risco de vazamento de informação;
- Confiabilidade de dados: dados tratados e manipulados de maneira correta;
- Foco em atividades: a automação permite que os funcionários foquem em atividades de maior expressão;
- Controle de processos: o RPA fornece maior controle sobre o processo;
- Retorno financeiro: alto potencial de retorno financeiro a curto prazo;
- Indicadores: melhor análise de indicadores devido à padronização de dados.

Além disso, RPA trabalha muito bem com grande volume de dados e consegue lidar de maneira eficiente com processos de baixo e médio nível de complexidade.

2.1.3 Ferramentas RPA

O mercado da tecnologia está em constante crescimento desde o surgimento de ferramentas de automação de processos. Isso se tornou ainda mais evidente com o surgimento das grandes empresas desenvolvedoras de RPA quando lançaram seus produtos e bibliotecas por volta de 2003 (HAJJAR, 2021).

Com o passar do tempo, as desenvolvedoras foram aprimorando seus *softwares* com novas tecnologias e atualizações, sendo que a cada nova versão, surgiam mais recursos. Hoje em dia, com o RPA já consolidado, as empresas líderes de mercado vislumbram grande possibilidade da integração do RPA com grandes tópicos da tecnologia atual, como computação cognitiva, *big data*, indústria 4.0 e aprendizado de máquina (OSTDICK, 2016).

As melhores e mais completas ferramentas de RPA são entregues pelas principais líderes de mercado. Segundo Gartner, conforme a última pesquisa (2021), essas empresas são UiPath, Automation Anywhere, Blue Prism e Microsoft. A Figura 1 mostra a representação gráfica do cenário das principais empresas de acordo com o mercado de RPA atual.

2.1.3.1 Blue Prism

O *software* da empresa BP (Blue Prism) se destaca por ser completo e exclusivo em vários aspectos. Ele engloba inúmeras ferramentas, além de bibliotecas e ambientes

Figura 1 – Quadrante mágico de Gartner RPA.



Fonte: Gartner (2021).

para desenvolvimento e produção dos projetos. Este é o *software* escolhido para o desenvolvimento do projeto por ser a principal plataforma de RPA utilizada pela empresa Bunge.

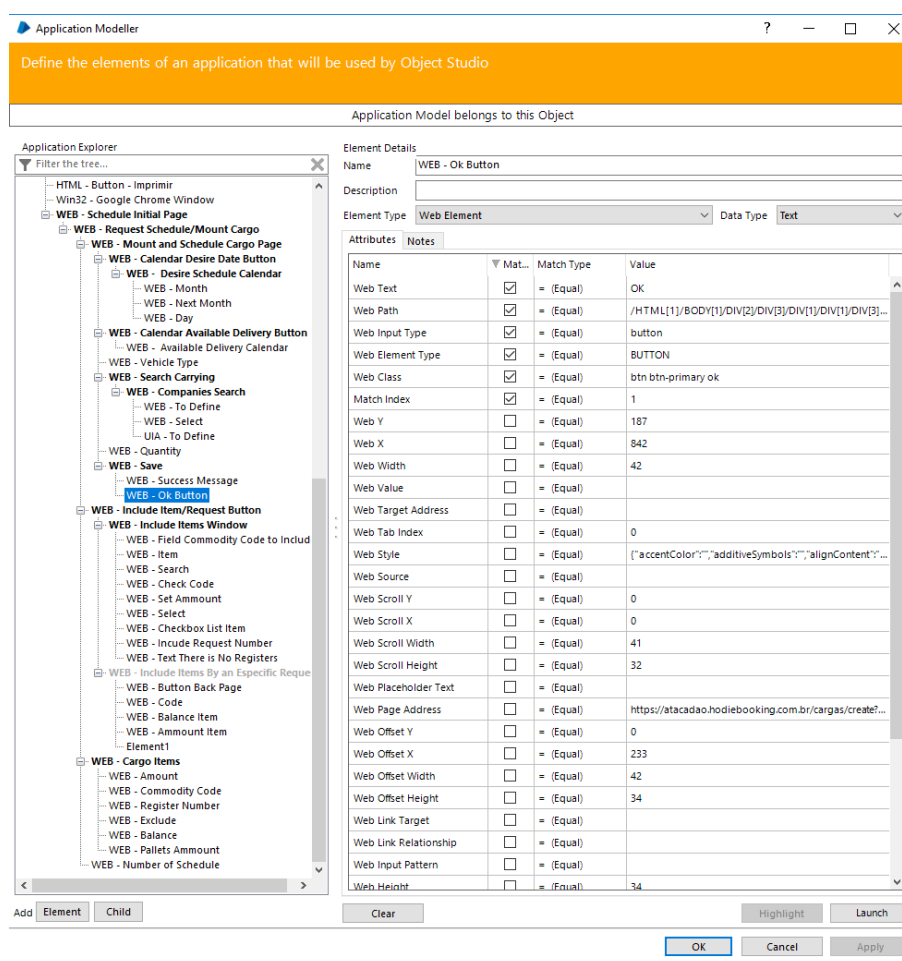
O BP tem uma interface gráfica e programável que funciona através de um fluxograma lógico de passos. Os blocos deste fluxograma são chamados de estágios e têm a função de realizar alguma operação do robô através da execução de *scripts* (CHAPPELL, 2017).

Existem inúmeros estágios disponíveis para que o desenvolvedor possa desenvolver uma automação robusta e eficiente. Os principais são os estágios de lógica que têm um arsenal de operações como *decision*, *calculation* e *loop*, estágios utilizados para interação com as aplicações como *Navigate*, *read* e *write* e estágios para armazenamento de informação como *data item* e *collection*, além de outros.

É importante mencionar que existe uma peculiaridade nos estágios utilizados para

a interação, no que diz respeito a como o robô consegue interagir com um *software* ou uma aplicação em geral. O *Application Modeller* é a ferramenta utilizada para modelar cada elemento (botão, campo, título, figura e etc.) na interface de usuário da aplicação. Essa modelagem funciona através de uma lista de atributos que cada elemento da interface de usuário possui, tornando cada elemento único e distinguível, conforme a Figura 2. Sendo assim, o robô consegue diferenciar um elemento de outro através destes atributos e portanto é capaz de comunicar-se com cada elemento desta aplicação (CHAPPELL, 2017).

Figura 2 – Atributos do botão OK no Application Modeller



Fonte: Elaborado pelo autor.

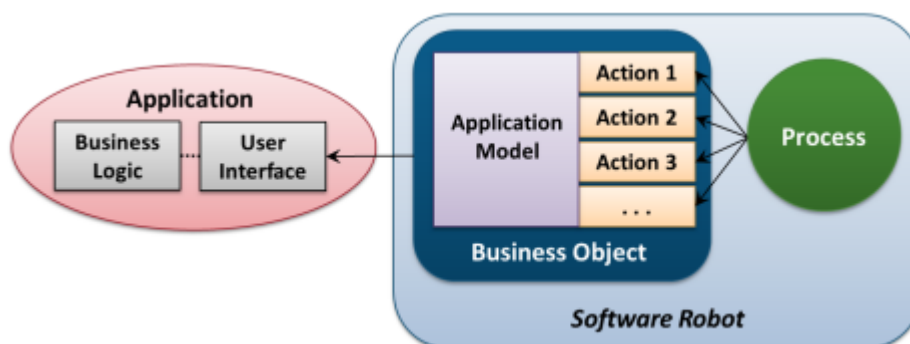
Para entender melhor o funcionamento do BP é muito importante definir e compreender dois conceitos: objeto e processo.

2.1.3.1.1 VBO (Visual Business Object)

O VBO (Visual Business Object) ou simplesmente objeto, nada mais é do que o conjunto de ações que o robô utiliza para fazer uma atividade dentro da aplicação.

Portanto, pode-se pensar que a construção do RPA funciona através de uma cadeia de passos, onde primeiramente é necessário modelar cada elemento da aplicação com seus atributos únicos e distinguíveis, desenvolver as ações que o robô toma com cada elemento e, por fim, desenvolver a atividade que o robô realiza com cada ação, como pode ser observado na Figura 3.

Figura 3 – Estrutura de desenvolvimento do BP.



Fonte: Chappell & Associates (2017).

O desenvolvedor pode contar com uma série de ferramentas em um ambiente exclusivo para desenvolver os objetos chamado de *Object Studio*. Todas as funções e estágios já mencionados, assim como o próprio *Application Modeller* estão integrados neste ambiente.

Finalmente, depois que os elementos são mapeados através do *Application Modeller* se pode criar as ações, que são tarefas em específico, como por exemplo: fazer *login* em uma página *web*, fazer *download* de documentos, entre outros. As ações são compostas por uma série de etapas, como: escrever em um campo, ler um campo, apertar um botão, etc. Cada interação é feita, neste caso, através dos estágios *write*, *read* e *navigate* respectivamente.

Portando, é necessário fornecer os parâmetros de entrada para que o estágio execute um código raiz, que é responsável por fazer a interação direta com os elementos mapeados da aplicação. Todos os estágios disponíveis para desenvolvimento da automação no *software Blue Prism* podem ser visualizados no ANEXO A deste documento.

As linguagens de programação dos códigos que cada estágio executa para interagir com a aplicação pode variar de acordo com o objeto. Os estágios de um objeto *web* por exemplo, podem executar códigos cuja linguagem de programação é Selenium, enquanto estágios de objeto Excel podem utilizar VBA (Visual Basic for Applications).

2.1.3.1.2 Process

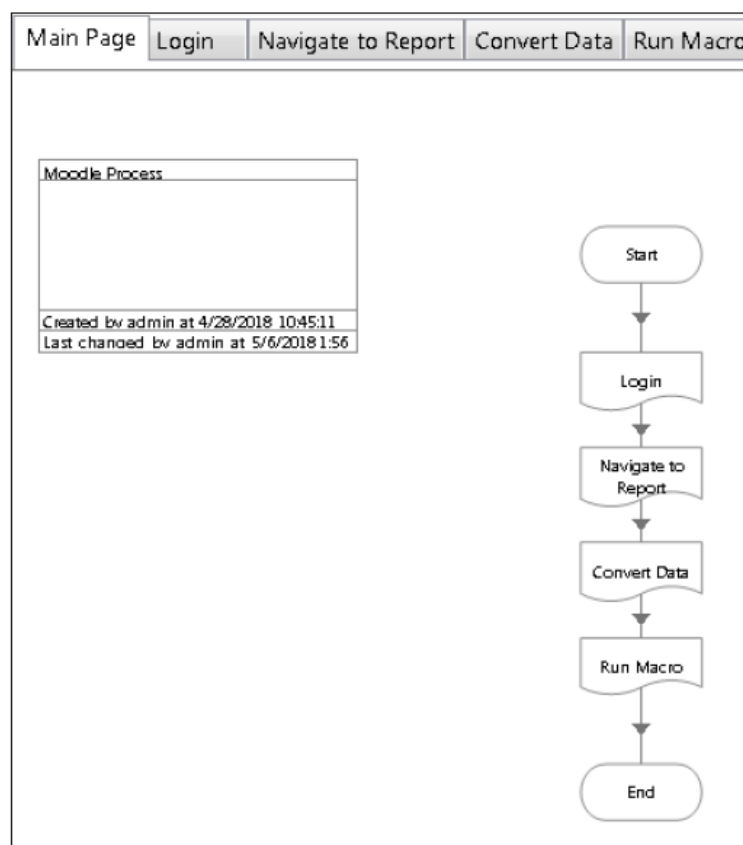
O processo por sua vez, é uma única atividade que deve ser realizada (com início e fim definidos), sendo que essa atividade pode ser feita através de um conjunto de tarefas

independentes. Assim, um processo é justamente a atividade em que se quer automatizar, que pode ser composta por interagir com vários aplicativos, *softwares* e sistemas, como também realizar uma série de tarefas para concluir a atividade por completo (CHAPPELL, 2017).

A abordagem que o BP utiliza separando um processo de um objeto é interessante e diferente da maioria das empresas desenvolvedoras de RPA. Isso porque cada objeto, como já foi explicado, tem funções únicas e genéricas, portanto cada processo pode chamar várias ações de objetos diferentes para realizar a automação. Por exemplo, um objeto para salvar arquivos do Google pode ser reutilizado por vários processos em que hajam a necessidade de fazer *download* de arquivos em algum momento da automação.

Essa definição fica ainda mais clara observando o processo da Figura 4. As duas primeiras páginas do processo fazem *login* em um *website* e navegam até a área de relatórios para fazer um *download*, portanto dentro delas devem haver várias ações de um objeto *web*. A terceira página converte o tipo de dado em que foi feito o *download*, portanto utiliza ações de um objeto desenvolvido para gerenciar arquivos. Por fim, a última página executa uma macro e portanto, dentro dela devem haver ações de objeto Excel (SHVETS, A.; SHVETS, V., 2018).

Figura 4 – Estrutura de um processo no BP.



2.1.3.1.3 Estágios do BP

O *software* Blue Prism conta com uma série de estágios para que o desenvolvedor possa criar a automação eficiente e robusta para processos de todas as áreas. Todos os estágios disponíveis no *software* podem ser visualizados de acordo com a Figura 39 no ANEXO A deste documento. Além disso, a Tabela 1 resume a funcionalidade de cada estágio.

Tabela 1 – Estágios para desenvolvimento de automações no *software* BP.

Estágios	Funcionalidade
<i>Pointer</i>	Ponteiro
<i>Link</i>	Conectar estágios
<i>Block</i>	Organizar em blocos
<i>Read</i>	Ler campo
<i>Write</i>	Escrever no campo
<i>Navigate</i>	Navegar
<i>Code</i>	Executar códigos
<i>Wait</i>	Esperar
<i>Process</i>	Chamar um processo
<i>Page</i>	Chamar uma página
<i>Action</i>	Chamar uma ação de algum objeto
<i>Decision</i>	Tomar decisão
<i>Choice</i>	Escolher entre opções
<i>Calculation</i>	Calcular expressões lógicas
<i>Multi Calculation</i>	Calcular múltiplas expressões lógicas
<i>Data Item</i>	Guardar variável
<i>Collection</i>	Guardar uma coleção de variáveis
<i>Loop</i>	Fazer ciclo dentro de uma <i>collection</i>
<i>Note</i>	Fazer anotações
<i>Anchor</i>	Âncora
<i>End</i>	Final da página
<i>Exception</i>	Exceção
<i>Recover</i>	Restaurar exceção
<i>Resume</i>	Retomar fluxo

Fonte: Elaborado pelo autor.

2.2 AGENDAMENTO DE CARGAS E CADEIA DE SUPRIMENTOS

A cadeia de suprimentos é definida como o conjunto de todas as atividades relacionadas com o fluxo e transformação de mercadorias desde o estágio da matéria-prima até o usuário final, bem como os respectivos fluxos de informação (BALLOU, 2009). Assim como no conceito de automação, o controle e gerenciamento da cadeia de suprimentos busca oportunidades para redução de custos e melhoria de serviços ao consumidor.

Além disso, existem alguns benefícios que acrescentam valor ao produto nas etapas de gerenciamento logístico dentro da cadeia de suprimentos (GOMES; RIBEIRO, 2020), como:

- Diminuir prazos de entregas;

- Aumentar confiabilidade dos prazos e quantidade;
- Reduzir problemas de qualidade, bem como aumentar a fidelidade dos dados e produtos;
- Ser eficaz.

Administrar o agendamento de cargas com eficiência é fazer a gestão do fluxo de entrada e saída dos materiais, controlar a solicitação de cada produto e também transferi-los de um ponto a outro nas mesmas condições.

Sendo assim, o agendamento se faz necessário para evitar gargalos do grande volume de carga e descarga no mesmo período para lojas, garantindo o fluxo de produtos muito mais efetivo. Da mesma forma, fazendo o agendamento das cargas se tem o controle de solicitações, alerta das cargas e indicadores do número de entradas e saídas dos produtos. Além disso, fazer o agendamento das cargas é vantajoso para a empresa pois isso permite o controle do horário de chegada a saída dos produtos, reduz ou até elimina a fila de espera entre a carga e descarga de produtos de diferentes remetentes (TRACKAGE, 2022).

2.2.1 Automação do Agendamento de Cargas

A tarefa de fazer o agendamento das cargas em muitos casos demanda uma pessoa focada a maior parte do tempo nessa atividade. Isso acontece por dois fatores principais: o grande volume de cargas que são agendadas e o tempo da verificação dos dados por item da carga. Além de que, esse processo feito de forma manual leva tempo para ser completado e traz despesas desnecessárias para a empresa.

Fazer o agendamento de cargas de forma manual requer excessiva atenção devido ao grande volume de dados em que se trabalha e portanto, frequentemente acontecem erros neste processo. Por isso, a automação nesta etapa da cadeia de suprimentos agrega uma série de benefícios tanto para a empresa quanto para as lojas clientes (ESALES, 2021). Alguns desses benefícios originados da automação são:

- Velocidade no processamento de itens para cada agendamento;
- Minimização de erros nos dados dos itens de cada carga;
- Redução de custos;
- Padronização na forma de agendar, nos relatórios e dados do processo;
- Fornecimento de indicadores;
- Otimização do fluxo e do processo.

Sendo assim, a automação no agendamento de cargas é muito válida para as empresas por se tratar de um processo repetitivo e linear, sendo estes os principais aspectos que possibilitam a utilização de RPA como forma de automação. Logo, a iniciativa da automação neste processo agrega valores e promove inúmeras vantagens para empresa.

2.3 SAP

Fundada em 1972, a empresa alemã SAP tem o objetivo de centralizar o gerenciamento de dados das empresas e companhias de uma forma mais efetiva através de um sistema ERP (Enterprise Resource Planning).

O ERP é um sistema para gerenciamento de negócios que contempla um conjunto de ferramentas e aplicativos para todas os setores administrativos dentro da empresa, como: contabilidade, vendas, recursos humanos, gerenciamento de materiais, planejamento de produção, entre outras. Sendo assim, um sistema ERP facilita o fluxo de informações de todos os processos dentro de uma organização, incluindo também os processos da cadeia de suprimentos (SHEHAB *et al.*, 2004).

O *software* da empresa SAP gerencia e integra todas as funções de negócio da companhia em uma só plataforma, centralizando o gerenciamento de dados, ele fornece uma única visão de todos os processos em execução. A centralização dos processos é vantajosa pois isso auxilia o melhor gerenciamento das atividades de negócio mais complexas, além de facilitar o acesso de diferentes colaboradores a todos os processos da empresa. A utilização do SAP pode acelerar os fluxos de trabalho, aumentar produtividade e melhorar a experiência dos colaboradores e clientes (SAP, 2022).

O SAP funciona através de transações, que são códigos alfanuméricos utilizados para executar um processo dentro do sistema. Ou seja, as transações funcionam como *links* que efetuam operações e atividades de negócio, como: telas de cadastro, relatórios de ordens de compra e venda, tabela de produtos, etc.

2.3.1 Gerenciamento de Cargas no SAP da Bunge

É muito interessante utilizar o SAP para o gerenciamento da cadeia de suprimentos pelas inúmeras vantagens já citadas neste documento. Em especial, o gerenciamento de cargas para todos os clientes da empresa Bunge, incluindo o Atacadão, é feito através do SAP.

O sistema SAP é utilizado para fazer o controle e gerenciamento de toda a parte da cadeia de suprimentos que envolve as cargas e o carregamento de produtos da empresa, desde a parte do cadastro de produtos, a solicitação e retorno de confirmação do agendamento das cargas, a saída dos veículos, etc. A transação utilizada pela área de CS para fazer o gerenciamento das cargas é chamada de ZSGCML0291. Ela contém tabelas para controle de produtos e pedidos, além de conter também campos de *status* do agendamento, horário da carga agendada, código da loja que solicitou os produtos, entre outros.

O SAP se faz uma ferramenta muito intuitiva com relação ao uso e aplicação dentro da empresa, além de ser realmente completo com o cadastro e fornecimento de todos os dados e relatórios importantes para o gerenciamento de uma atividade. Com essa tecnologia, os setores da empresa responsáveis pelo gerenciamento de cargas têm total

controle sobre essa etapa da cadeia de suprimentos.

2.4 POWER APPS

O avanço da tecnologia tem possibilitado o desenvolvimento de aplicativos no ambiente corporativo e se mostrado relevante em soluções para serviços e tarefas, além de trazer bons indicadores de resultados para as empresas. Em 2018 a Microsoft lançou a primeira versão do conjunto de ferramentas Power Platform com o objetivo de inovar com relação aos métodos tradicionais de desenvolvimento de aplicativos e serviços.

O Power Apps é um dos ambientes do conjunto Microsoft Power Platform que tem como atributos um grupo de serviços e conectores, que servem para o desenvolvimento rápido de aplicativos personalizados de acordo com a necessidade da aplicação. O Power Apps oferece funcionalidades avançadas para transformar processos manuais em sistemas automatizados, que podem ser executados tanto por um navegador em *desktop* ou até mesmo em dispositivos móveis (VIVEK, 2022).

O desenvolvimento de aplicativos no Power Apps funciona com base em fontes de dados, que podem ser tabelas ou ações, que são fornecidas por conectores. Os conectores por sua vez, são os aplicativos e serviços sobre os quais o Power Apps consegue se interligar e atuar. Alguns exemplos de conectores são o Office Package, SharePoint, SQL Server, Youtube, Twitter, entre outros, sendo que alguns deles são pagos e outros gratuitos (DELANO, 2022).

Portanto, quando o Power Apps se conecta com um aplicativo ou serviço cuja fonte de dados é uma tabela por exemplo, ele recupera as informações desta tabela e salva dentro do próprio aplicativo para que o desenvolvedor possa gerenciar todos os elementos. O Power Apps tem uma linguagem própria com sintaxe definida e oferece várias funções para que se possa manipular os dados fornecidos da fonte. Algumas dessas funções são globais conhecidas por desenvolvedores de qualquer linguagem de programação e outras criadas justamente para a utilização do Power Apps (DELANO, 2022).

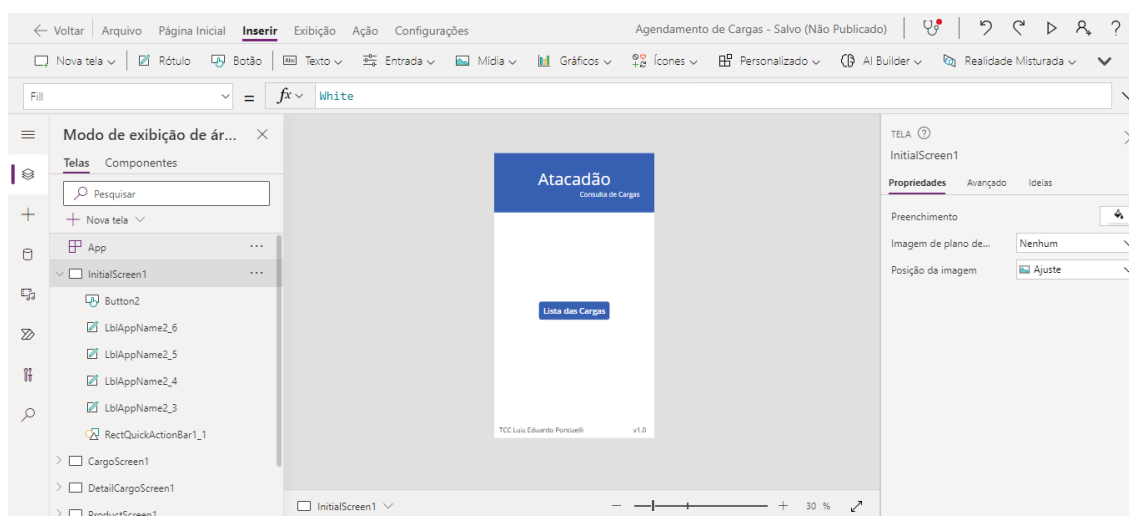
Por fim, o Power Apps mostra os dados retirados da fonte através de uma galeria ou formulário, que são ferramentas dentro do próprio aplicativo para a visualização e interação das informações. Ele ainda oferece opções de botões, rótulos, entradas de texto, mídia, ícones, gráficos e muitos outros instrumentos para que se possa desenvolver um aplicativo robusto e intuitivo o suficiente para todos os tipos de aplicações.

2.4.1 Estrutura do Power Apps

A estrutura de desenvolvimento de aplicativos no *Power Apps* é simples e intuitiva, pois ele mostra todas as ferramentas disponíveis para a construção do *app* na janela principal, conforme ilustrado na Figura 5. Pode-se notar a distribuição das ferramentas para desenvolvimento na parte superior da imagem, os recursos e telas na parte esquerda,

o menu de configurações e parâmetros dos objetos na parte direita e, finalmente, a pré-visualização de cada tela do aplicativo na parte central. Além disso, ainda na parte superior existe uma barra da fórmulas que é utilizada para inserir as equações lógicas de cada tela, rótulo, botão, etc. O funcionamento e desempenho do aplicativo é totalmente baseado nas fórmulas aplicadas em cada componente, que têm ação direta nas duas fontes de dados selecionadas ao criar o *app*.

Figura 5 – Tela Inicial.



Fonte: Elaborado pelo autor.

3 DESENVOLVIMENTO

Este Capítulo apresenta o desenvolvimento do projeto de automação do processo de agendamento de cargas no portal Atacadão, utilizando RPA como tecnologia. Além de apresentar também o desenvolvimento do aplicativo de consultas no Power Apps. Ademais, as Subseções detalham o fluxo da atividade de agendamento dentro do portal Atacadão, o fluxo da automação e as aplicações que o robô interage. Estas seções são importantes para entender como é feita a atividade de agendamento de cargas de forma manual e como a automação atua sobre as aplicações deste processo.

O projeto é desenvolvido com as plataformas Blue Prism e Power Apps, cujas licenças para utilização dos *softwares* são disponibilizadas pela empresa Bunge para projetos internos do setor de RPA.

3.1 APLICAÇÕES DA AUTOMAÇÃO

Para fazer a automação de agendamento de cargas por completo o robô deve interagir com três aplicações: Microsoft Excel, Google Chrome (portal Atacadão) e SAP.

A primeira aplicação que o robô interage é a planilha de pedidos e produtos para agendamento no portal Atacadão fornecida pela área de CS. Essa planilha é disponibilizada em formato ‘.xlsx’ e pode ser visualizada através do editor de planilhas da Microsoft chamado Excel. Essa ferramenta é globalmente utilizada por grande parte das empresas para tratar informações através de planilhas eletrônicas, pois é a plataforma mais completa. Além disso, o setor de RPA da Bunge já possui inúmeras ações para o objeto Excel, tornando o desenvolvimento da automação mais simples.

O robô também interage com o portal Atacadão (*website* para gerenciar todas as cargas e pedidos solicitados pelas lojas Atacadão) através do navegador Google Chrome utilizando a URL indicada na nota de rodapé desta página ¹. O portal contém todas as informações detalhadas das cargas, produtos e pedidos solicitados pelas lojas Atacadão para todos os fornecedores. Sendo assim, toda a etapa da automação do agendamento de cargas é feita através desta página *web*.

Além disso, a última aplicação que faz parte do escopo desta automação é o SAP. O robô interage com o sistema SAP para marcar os pedidos processados corretamente na transação ‘ZSGCML0291’, além de alterar o *status* e horário de agendamento dos pedidos.

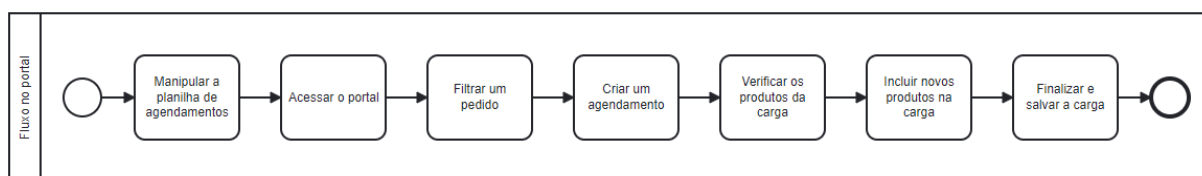
Para complementar o projeto, é desenvolvido também um aplicativo através da ferramenta Microsoft Power Apps para fazer a consulta e ajuste das cargas processadas com sucesso pela automação. Esse aplicativo possibilita fazer uma busca avançada das cargas e produtos agendados de forma rápida e intuitiva, sem a necessidade de procurar esses dados na planilha de agendamento, que é repleta de informações e visivelmente mais poluída.

¹ <https://atacado.hodiebooking.com.br/>

3.2 FLUXO NO PORTAL ATACADÃO

O fluxo de operações realizadas neste processo requer tarefas combinadas e simultâneas no *website* do portal Atacadão e na planilha para que a atividade de agendamentos de cargas no portal seja completada, conforme a Figura 6. Todo o processo do agendamentos é baseado em uma planilha eletrônica que contém as informações de todas as cargas, pedidos e produtos fornecidos pela empresa Bunge para as lojas Atacadão. A dinâmica utilizada para manipular a planilha é: um número de produtos formam um pedido, enquanto um número de pedidos formam uma carga.

Figura 6 – Fluxograma das atividades no portal.



Fonte: Elaborado pelo autor.

O procedimento para o agendamento de cargas se inicia com o tratamento da planilha de agendamentos através de classificações e filtros dos pedidos. Primeiramente é necessário selecionar os pedidos cujo status de agendamento é ‘Solicitar’, sendo assim, trabalha-se somente com os itens que são necessários fazer o agendamento no portal. Depois disso é necessário ordenar a planilha pelo código emissor da ordem para agrupar os pedidos que serão fornecidos para cada loja, conforme a Figura 7.

Figura 7 – Planilha de agendamento de cargas do cliente Atacadão.

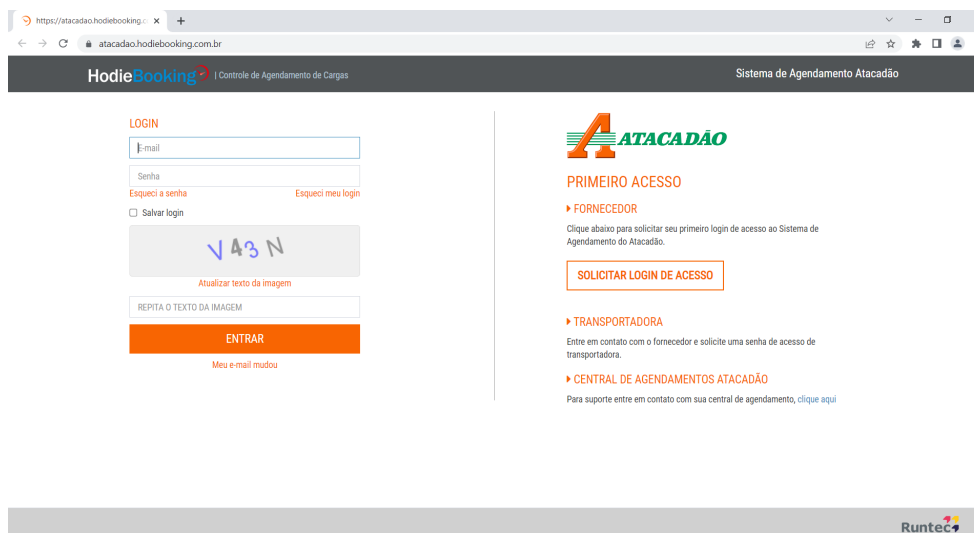
	K	L	Z	AA	AC	AD	AP	CZ	DP	EF	EG
	Documento SD	Nº do pedido	Código do Emissor	Nome	Cidade do Emissor da	Est	Qtd a fornece	Qtde_Pallet	Codigo_Atacadao	Status OV	Data Solicitac
4	19017694	834310	1004050	ATACADA0 S.A	CUIABA	MT	1512	24	3423	Solicitar	05/05/2022
5	19025653	933776	1004063	ATACADA0 S.A	ARACAJU	SE	1764	5	3423	Solicitar	05/05/2022
6	19025654	933776	1004063	ATACADA0 S.A	ARACAJU	SE	1764	10	3423	Solicitar	05/05/2022
7	19240649	951090	1004065	ATACADA0 S.A	JUAZEIRO	BA	1764	28	11300	Solicitar	05/05/2022
8	19240644	951087	1059250	ATACADA0 S.A	LAURO DE FREITAS	BA	1764	5	11300	Solicitar	05/05/2022
9	19240645	951088	1059260	ATACADA0 S.A	FEIRA DE SANTANA	BA	1764	28	11300	Solicitar	05/05/2022
10	19240650	951091	1059268	ATACADA0 S.A	FEIRA DE SANTANA	BA	1764	28	11300	Solicitar	05/05/2022
11	18967566	508002	1059270	ATACADA0 S.A	JUAZEIRO DO NORTE	CE	1764	7	3423	Solicitar	05/05/2022
12	19240651	951092	1059278	ATACADA0 S.A	EUNAPOLIS	BA	1764	13	11300	Solicitar	05/05/2022
13	19025989	933781	1059279	ATACADA0 S.A	VITORIA DA CONQUISTA	BA	1764	28	3423	Solicitar	06/05/2022
14	19025990	933781	1059279	ATACADA0 S.A	VITORIA DA CONQUISTA	BA	1764	15	3423	Solicitar	06/05/2022
15	19240654	951095	1059295	ATACADA0 S.A	FEIRA DE SANTANA	BA	1764	28	11300	Solicitar	05/05/2022
16	18999499	330413	1851218	ATACADA0 S.A	MANAUS	AM	1512	24	3423	Solicitar	05/05/2022
17	18999503	330413	1851218	ATACADA0 S.A	MANAUS	AM	1413	22	3423	Solicitar	05/05/2022
18	18999510	330413	1851218	ATACADA0 S.A	MANAUS	AM	99	1	3423	Solicitar	05/05/2022
19	18753420	327282	1717327	ATACADA0 S.A	BELEM	PA	1512	24	9735	Solicitar	05/05/2022
20	18865650	330204	1717327	ATACADA0 S.A	BELEM	PA	1512	5	3423	Solicitar	05/05/2022
21	18865703	330204	1717327	ATACADA0 S.A	BELEM	PA	1512	24	9735	Solicitar	05/05/2022
22	19024221	933780	1717327	ATACADA0 S.A	BELEM	PA	1512	4	9735	Solicitar	05/05/2022

Fonte: Disponibilizado pela área de CS.

Depois de tratar a planilha, pode-se abrir o portal atacadão inserindo o usuário, senha e o *captcha* que estão localizados na página *web* de *login* conforme ilustrado na

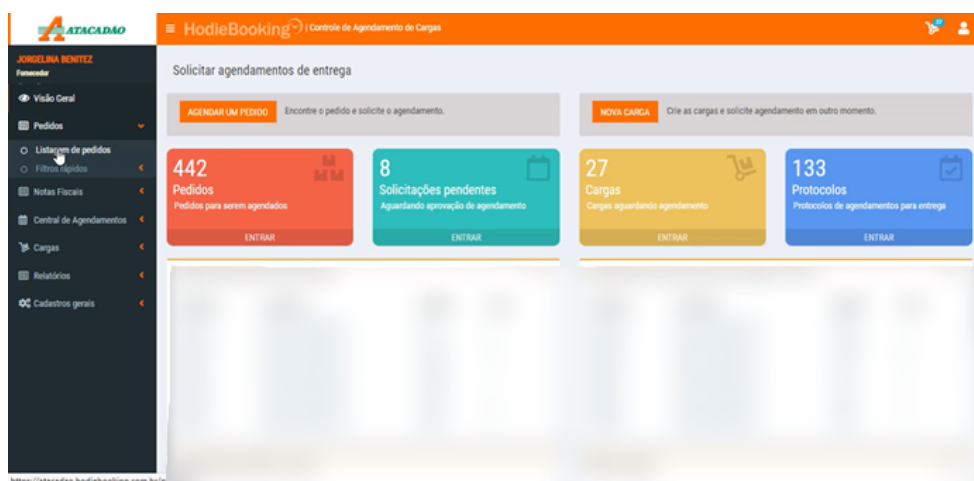
Figura 8. Logo depois que o *login* é realizado, o usuário é redirecionado para a página *web* inicial do portal (Figura 9) que contém uma série de informações, como: o número de pedidos para serem agendados, solicitações pendentes, cargas e protocolos. Além disso, há também uma barra de navegação no canto esquerdo da Figura 9 que mostra uma visão geral das opções de pesquisa e navegação dentro do portal.

Figura 8 – Página de *login* do portal Atacadão.



Fonte: Elaborado pelo autor.

Figura 9 – Página inicial do portal Atacadão.



Fonte: Elaborado pelo autor.

Para criar o agendamento de uma carga é necessário pressionar o botão 'Listagem de pedidos' dentro da opção 'Pedidos' da barra de navegação. Feito isso, o portal abre

uma janela com o histórico da lista de todos os pedidos que já foram agendados e também dos pedidos que foram registrados e que devem ser agendados.

Dentro desta mesma janela existe o botão de ‘Filtros’, que possibilita fazer um filtro que retorna um só pedido da coluna ‘Nº do pedido’, mostrado na Figura 7. Sendo assim, é possível criar um agendamento para cada pedido em específico pressionando o botão de ‘Solicitar Agendamento/Montar carga’. Este botão só aparece disponível para pedidos que ainda não foram agendados.

Depois de criar o agendamento para um pedido em específico, o portal redireciona o usuário para a página da ‘Montagem e agendamento da carga’, de acordo com a Figura 10. O próprio sistema do portal já inclui todos os produtos de um pedido que a loja Atacadão solicita, como se pode notar na parte inferior da Figura 10, porém nem sempre esses pedidos serão formados pelos materiais que o portal indica. Isso pode acontecer por diversos fatores, como por exemplo indisponibilidade em estoque para entrega do material. Portanto, deve-se formar a carga com base na planilha de agendamentos, pois a área de CS é responsável por controlar todos os produtos que serão agendados.

Figura 10 – Página montagem e agendamento de carga.

Código	Mercadoria	Pedido	Embalagem	Saldo	Qtde. alocada	Qtde. Paletes
50322	FARINHA DE TRIGO PREDILETA 25KG	34241	SCD	360	360	8,182
9493	FARINHA TRIGO B. PRO PASTEL 5KG	34241	FDD	60	60	1,5
6960	FARINHA TRIGO B. PRO PIZZA 5KG	34241	FDD	60	60	1,5
70077	FARINHA TRIGO PRIMOR C/FERMENTO 1KG	34241	ENV	100	100	1
47853	FARINHA TRIGO PRIMOR TIPO 1 1KG	34241	FDD	800	800	8
12025	FARINHA TRIGO SUPREMA 25KG	34241	SCD	80	80	1,818

Fonte: Elaborado pelo autor.

Ao criar o agendamento para um pedido, deve-se inicialmente verificar se os produtos que o portal indica estão de acordo com os produtos da planilha de agendamentos, conforme a Figura 11. Isto é, verificar os códigos dos produtos de acordo com a coluna ‘Codigo_Atacadao’ e a quantidade de produtos a fornecer de acordo com a coluna ‘Quantidade a fornecer’. Também deve-se tomar cuidado para não exceder o número máximo de *pallets* de uma carga.

Se houverem divergências na quantidade de itens que serão fornecidos de acordo com cada produto, basta corrigir o valor no portal de acordo com a planilha. Em contrapartida,

Figura 11 – Planilha de agendamentos filtrada de acordo com os pedidos da mesma loja.

	K	L	Z	AA	AC	AD	AP	CZ	DP	EF	EG
4	Documento SD	Nº do pedido	Código do Emissor	Nome	Cidade do Emissor da	Est	Qtd a fornece	Qtde_Pallet	Codigo_Atacadac	Status O	Data Solicitac
277	19034207	32241	2140511	ATACADAO S.A.	SANTAREM	PA	360	8,2	50322	Solicitar	06/05/2022
278	19034210	32241	2140511	ATACADAO S.A.	SANTAREM	PA	60	1,5	9893	Solicitar	06/05/2022
279	19034211	32241	2140511	ATACADAO S.A.	SANTAREM	PA	60	1,5	6969	Solicitar	06/05/2022
282	19034222	32241	2140511	ATACADAO S.A.	SANTAREM	PA	100	1	70077	Solicitar	06/05/2022
285	19034223	32241	2140511	ATACADAO S.A.	SANTAREM	PA	800	8	47853	Solicitar	06/05/2022
287	18867777	32241	2140511	ATACADAO S.A.	SANTAREM	PA	80	1,8	12925	Solicitar	06/05/2022
290	19240802	340827	2140511	ATACADAO S.A.	SANTAREM	PA	1	0,1	7393	Solicitar	06/05/2022
291	19240803	340827	2140511	ATACADAO S.A.	SANTAREM	PA	1	0,1	6416	Solicitar	06/05/2022
292	19240804	340827	2140511	ATACADAO S.A.	SANTAREM	PA	1	0,1	6415	Solicitar	06/05/2022
293	19269846	340827	2140511	ATACADAO S.A.	SANTAREM	PA	1	0,1	6385	Solicitar	06/05/2022
294	19269847	340827	2140511	ATACADAO S.A.	SANTAREM	PA	1	0,1	9618	Solicitar	06/05/2022

Fonte: Elaborado pelo autor.

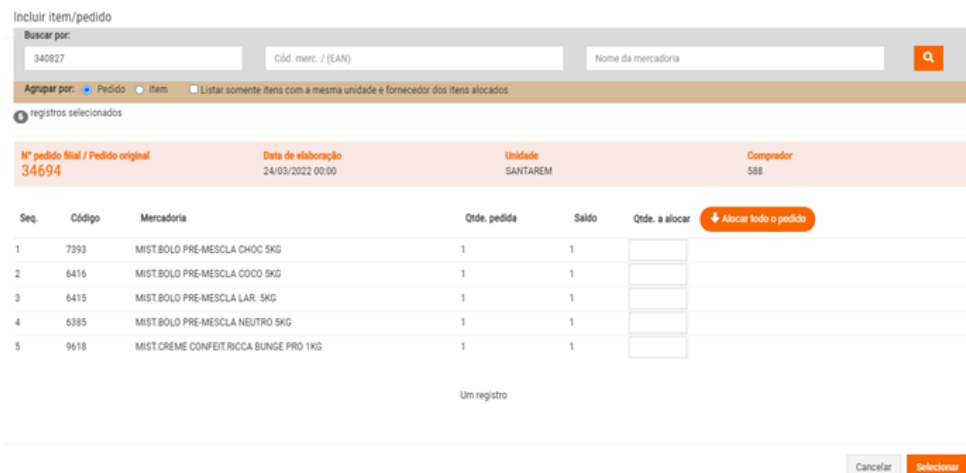
existem duas possibilidades se houverem divergências nos códigos dos produtos: a primeira delas é excluir o produto da carga no portal se ele não consta na planilha de agendamentos ou, a segunda que é adicionar um produto na carga se ele aparece na planilha mas não no portal.

É possível haver vários pedidos que devem ser agendados para a mesma loja no mesmo dia, conforme ilustrado na Figura 11. Sendo assim, dependendo do volume da carga pode-se tomar duas decisões: adicionar mais um pedido na mesma carga ou fechar a carga com os pedidos atuais. Essa decisão é baseada na quantidade total de *pallets* da carga, uma vez que é possível adicionar mais pedidos pressionando o botão ‘INCLUIR ITEM/PEDIDO’, de acordo com a Figura 10, quando a quantidade total de *pallets* não ultrapassa o máximo permitido para uma carga. Por outro lado, se isso não for verdade, deve-se fechar a carga atual, e abrir uma nova para os demais pedidos.

Ao pressionar o botão para inclusão de pedidos na carga, o portal abre uma segunda janela de acordo com a Figura 12, onde existe um campo para buscar o número do pedido que se quer adicionar. Quando essa pesquisa é realizada, o portal já indica os produtos do pedido que podem ser adicionados. Portanto, basta conferir cada item de acordo com o código de cada produto da planilha, e adicioná-los com a quantidade correta a ser fornecida.

Quando todos os pedidos solicitados pela mesma loja em um mesmo dia forem tratados, pode-se salvar a carga e fazer o agendamento. Para fazer o agendamento, basta voltar na página de montagem e agendamento de carga (Figura 10), inserir as datas de agendamento nos campos corretos, escolher o tipo do veículo de transporte que varia de acordo com a quantidade de *pallets* da carga e, por fim, definir a transportadora. Feito isso, pode-se pressionar no botão ‘SALVAR’ para finalizar o processo. Logo na sequência, o portal retorna uma mensagem de registro com sucesso juntamente com o código da carga, quando todo o processo foi feito de forma correta.

Figura 12 – Janela de inclusão de pedidos na carga.



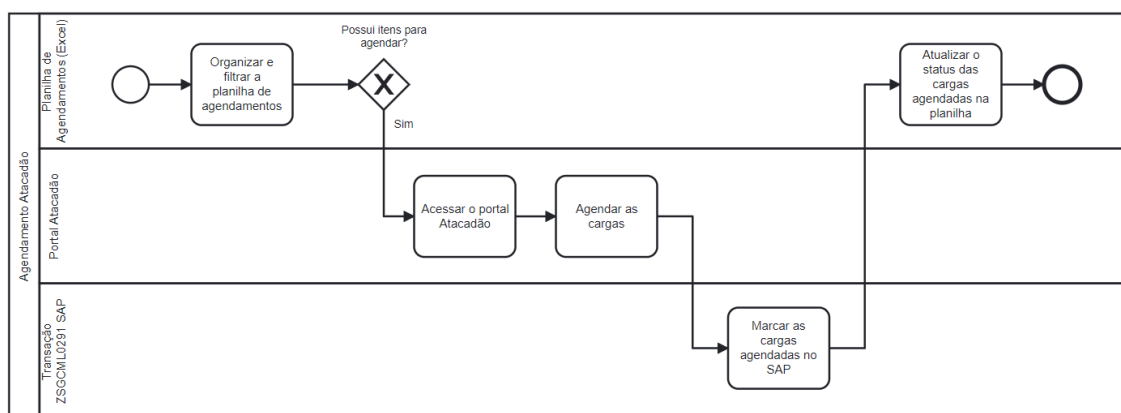
Fonte: Elaborado pelo autor.

3.3 FLUXO DA AUTOMAÇÃO COM RPA

Esta seção relata de forma breve as etapas de cada atividade que o robô utiliza para automatizar o processo descrito na seção anterior.

O fluxo da automação deste projeto é simples e linear, como mostrado na Figura 13. Inicialmente, o setor CS da Bunge disponibiliza uma planilha em Excel com o todos os pedidos de agendamento de cargas para o cliente Atacadão em uma pasta que o robô tem acesso. O robô manipula a planilha filtrando os pedidos que serão agendados no momento da execução e separa-os em cargas para cada loja Atacadão.

Figura 13 – Fluxo do processo.



Fonte: Elaborado pelo autor.

Com todos os pedidos definidos e organizados, o robô quebra o *captcha* e acessa o portal Atacadão com o usuário e senha fornecidos pelo setor CS, faz o agendamento

das cargas para cada loja em específico e verifica o código da carga gerado pelo próprio portal para assegurar que o agendamento foi feito corretamente. Finalmente, o robô volta a planilha do início do processo e muda o status de cada pedido para solicitado agendamento. Depois disso, refaz o mesmo fluxo novamente até que todos os pedidos quem devem ser agendados nesta execução sejam concluídos.

O fluxograma do processo desenvolvido no BP segue a mesma sequência já descrita, porém é dividido em etapas, referenciadas como ‘páginas’, para que seja mais claro a visualização de cada tarefa que o robô realiza, conforme a Figura 14. Sendo assim, cada página contém uma sequência de tarefas que em conjunto complementam a atividade de agendamento por completo. Esse desenho serve também para representar que o robô executa o conjunto de ações que estão dentro de cada página do fluxograma demonstrado na figura, no qual ele executa cada tarefa do processo de agendamento de cargas de acordo com o fluxo das páginas.

No canto esquerdo de cada página são posicionados todas as variáveis do processo, organizadas em blocos conforme sua utilização. Os blocos de variáveis podem ser observados de acordo com a Figura 14.

Por fim, a automação sempre é executada através de uma *schedule*. Em raras exceções o time de *control room* executa a automação por alguma razão em específico. A *schedule* inicia o processamento do robô todos os dias úteis nos horários de 08:40, 09:40, 10:40, 12:30, 14:30 e 16:30.

3.4 DESENVOLVIMENTO DO PROJETO DE AUTOMAÇÃO

Nesta seção é descrito todo o desenvolvimento do projeto de automação com RPA, detalhando o desenvolvimento do robô e como ele realiza cada atividade do processo. Cada etapa do projeto é separada em subseções, que dizem respeito às operações que o robô realiza sobre as aplicações do processo de agendamento de cargas, de acordo com as páginas da Figura 14.

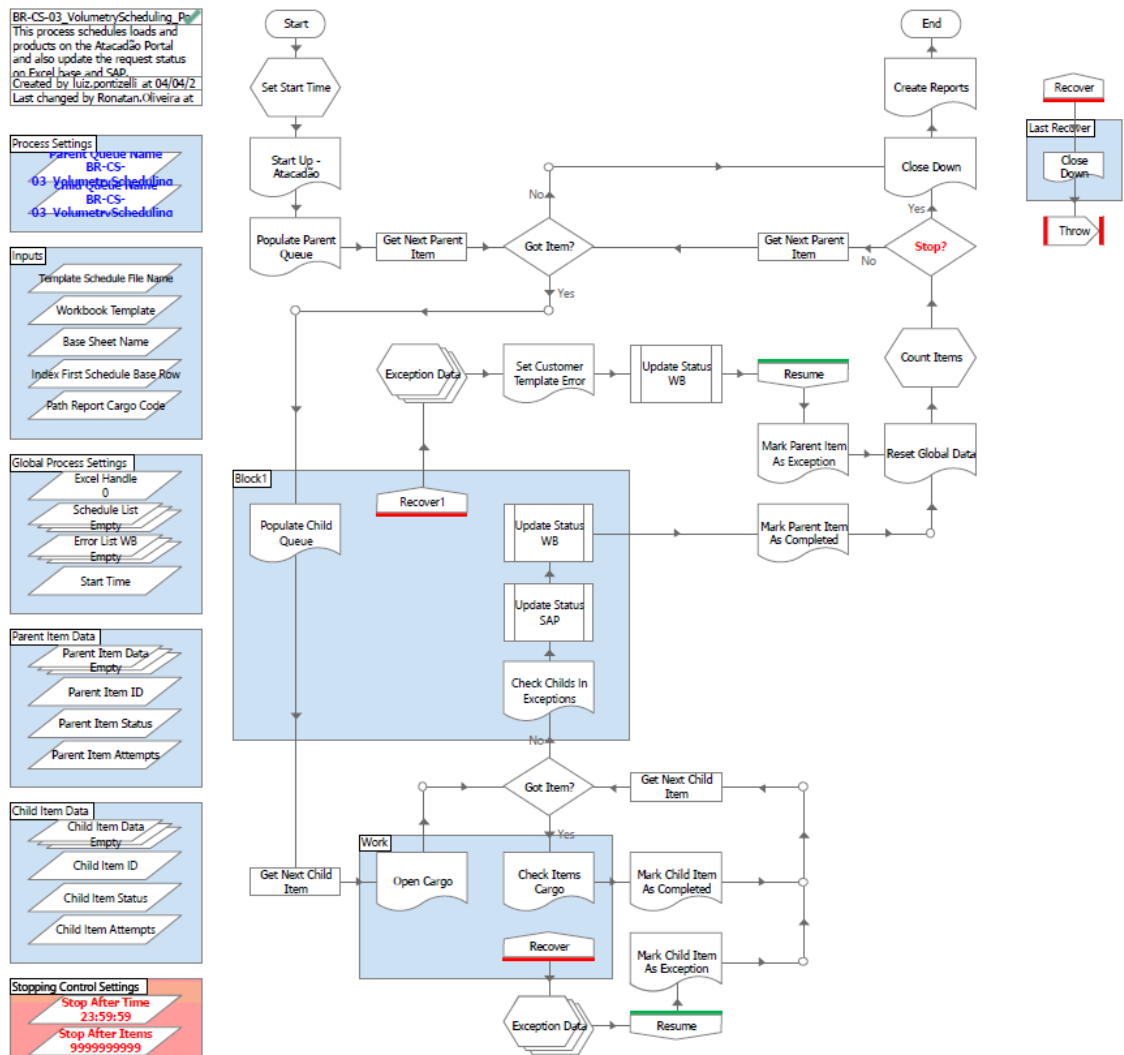
3.4.1 Agendamento no Portal

Esta subseção detalha a automação do processo de agendamento das cargas dentro do portal Atacadão, incluindo as tarefas de fazer o *login*, organizar os pedidos, criar, montar e salvar as cargas.

3.4.1.1 Conexão no Portal

A primeira interação que o robô tem com o portal Atacadão acontece na página *Start Up - Atacadão*, cujo fluxo da automação pode ser observado na Figura 15. Pode-se verificar que a página é dividida em duas partes. No início o foco é buscar as credenciais de acesso do portal e depois, quebrar o *captcha*.

Figura 14 – Desenho do processo no Blue Prism.

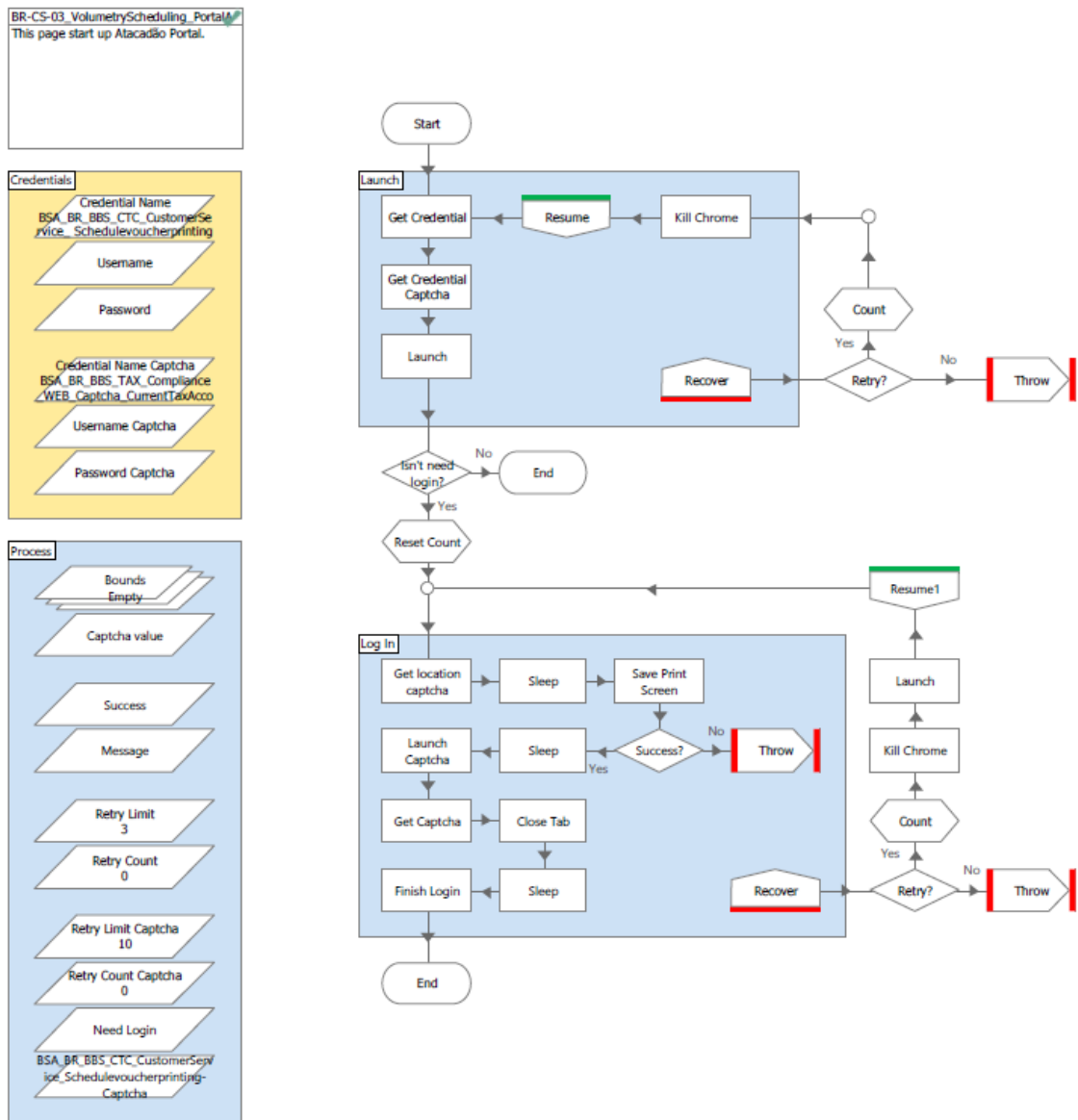


Fonte: Elaborado pelo autor.

As duas primeiras *actions* da página (*Get Credential* e *Get Credential Captcha*) se resumem em buscar os valores das credenciais de acesso ao portal Atacadão e credencias do site que quebra o *captcha* do portal. Essas credenciais são guardadas dentro de variáveis em um ambiente especial no *software* Blue Prism e portanto são sigilosas, sendo que somente o *key-user* (operador do processo) conhece o acesso.

Logo na sequência, a *action Launch* inicializa o Google Chrome (navegador em que foi desenvolvido a VBO do portal Atacadão) na página de *login* do portal. Em algumas situações a automação abre o site já conectado na conta do usuário, sendo assim, não é necessário fazer o *login*. Portanto, o bloco *decision* direciona o fluxo para o final da página, de acordo com a Figura 15. Caso contrário, o fluxo de execução segue na vertical quando é necessário entrar na conta do usuário, sendo que o primeiro passo é fazer a quebra do *captcha* através de uma série de ações. Para fazer isso, a *action Get Location Captcha*

Figura 15 – Página Start Up - Atacadão.



Fonte: Elaborado pelo autor.

busca na tela de *login* a localização exata e retorna a posição do *captcha* para que, logo na sequência, a *action Save Print Screen* capture a tela baseada na posição e nos limites do ícone do *captcha*. Uma exceção é lançada caso esse processo da captura de tela não seja executado corretamente.

Na sequência, a combinação das *actions Launch Captcha* e *Get Captcha* abrem o site mencionado em uma segunda guia do navegador, onde são inseridos as credencias do site e também é feito o *upload* da imagem do *captcha* desejado. Sendo assim, quando essas informações são inseridas, o site processa a imagem enviada e retorna o valor do *captcha*. Depois disso, a guia com o site é fechada.

Depois de todo esse procedimento tem-se todas as informações necessárias para

fazer o *login* no site do portal através da *action Finish Login*, cujos parâmetros de entrada são exatamente esses dados colhidos. Essa *action* tem a função de inserir os parâmetros nos campos corretos e fazer o *login* no portal Atacadão.

Se em algum momento acontecer uma exceção dentro dos blocos azuis na Figura 15, o fluxo das ações é interrompido imediatamente e desviado para o *Recover*. Este *stage*, combinado com *Resume*, serve para tentar realizar a tarefa novamente e por isso faz parte de um fluxo em formato de *loop*.

Dentro deste *loop* é utilizado o *decision Retry?* para extinguir a possibilidade de um *loop* infinito, cujo parâmetro de decisão é a comparação dos *Data Items Retry Limit* e *Retry Count*. Além disso, o *Calculate Count* incrementa justamente a variável *Retry Count*. Portanto, toda vez que o fluxo segue em direção ao *loop*, a sequência lógica é verificar se ainda existe a possibilidade de tentar refazer a atividade novamente ou se a variável já ultrapassou o limite, não permitindo uma nova tentativa.

A *action Kill Chrome* fecha todas as abas no navegador e é utilizada também dentro do *loop* antecedente à *Launch*. Assim, será restabelecido a aplicação do portal Atacadão com as configurações e parâmetros iniciais quando as *actions* forem executadas novamente. Por fim, os *stages* localizados à esquerda da Figura 15 são variáveis locais, que são utilizadas durante a execução desta página.

3.4.1.2 Organização dos Pedidos em Cargas

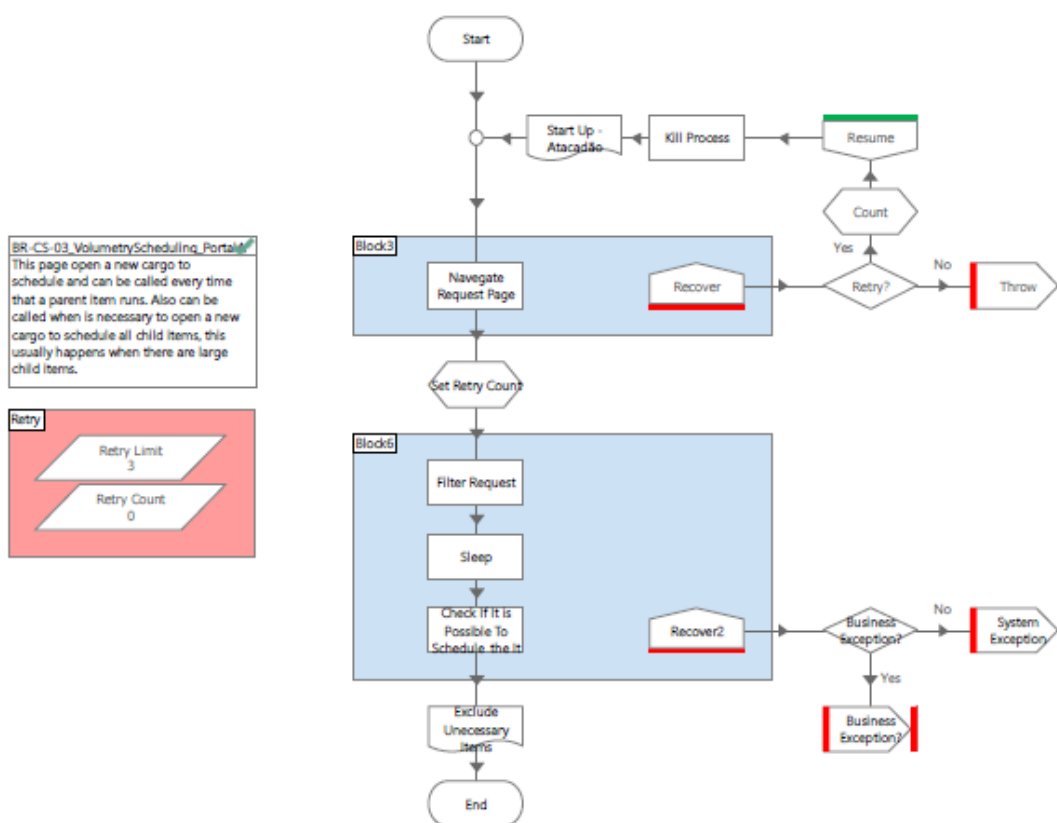
Depois que o *login* é realizado através da página *Start Up - Atacadão*, inicia-se a atividade da verificação de cada pedido, de acordo com a planilha de agendamentos. Primeiramente o robô abre e manipula a planilha que contém toda a carteira de cada loja cliente Atacadão, ou seja, todos os produtos e materiais que são fornecidos para as lojas. A dinâmica das ações do robô é ordenar os produtos que formam um pedido e logo depois agrupar os pedidos que formam uma carga e que serão entregues para a mesma loja. Assim, é preciso filtrar os produtos para os quais deve-se solicitar agendamento e então ordenar a planilha pelo código que representa cada loja Atacadão, de acordo com a Figura 11.

Depois disso, são preparadas e populadas as filas de trabalho para a execução do robô. A fila pai é composta pelas cargas que devem ser agendadas e a fila filho pelos pedidos e produtos de cada carga. Essas duas filas são preparadas respectivamente durante a execução das páginas *Populate Parent Queue* e *Populate Child Queue* de acordo com a Figura 14. As filas funcionam em conjunto, de forma hierárquica, sendo que cada item pai tem pelo menos um ou mais itens filhos. Em cada ciclo de execução, cria-se um agendamento para cada loja, representando um item pai. Na sequência, são incluídos todos os pedidos desta carga, que representam os itens filho. Esse processo se repete a cada ciclo de execução até que todos pedidos e cargas sejam alocados nas filas de trabalho.

3.4.1.3 Abrir Carga

Após a geração das filas a partir da planilha de agendamentos, o robô pode criar uma nova carga no portal através da página *Open Cargo* com o objetivo de fazer as etapas iniciais de agendamento. Dentro dessa página, conforme a Figura 16, a primeira interação efetuada é abrir a janela de pedidos no portal utilizando as informações do item da fila pai, ou seja, informações da carga. Cria-se um agendamento de carga com o número do pedido que é referente à loja Atacadão correspondente ao item da fila pai. Então, ao inserir o número do pedido e depois selecioná-lo, o próprio portal redireciona o usuário para uma segunda página *web*, que é referente à loja em que foi feita esta requisição de produtos. Essas duas interações descritas correspondem às *actions* dos estágios *Navigate Request Page* e *Filter Request* de acordo com a Figura 16.

Figura 16 – Página Open Cargo.



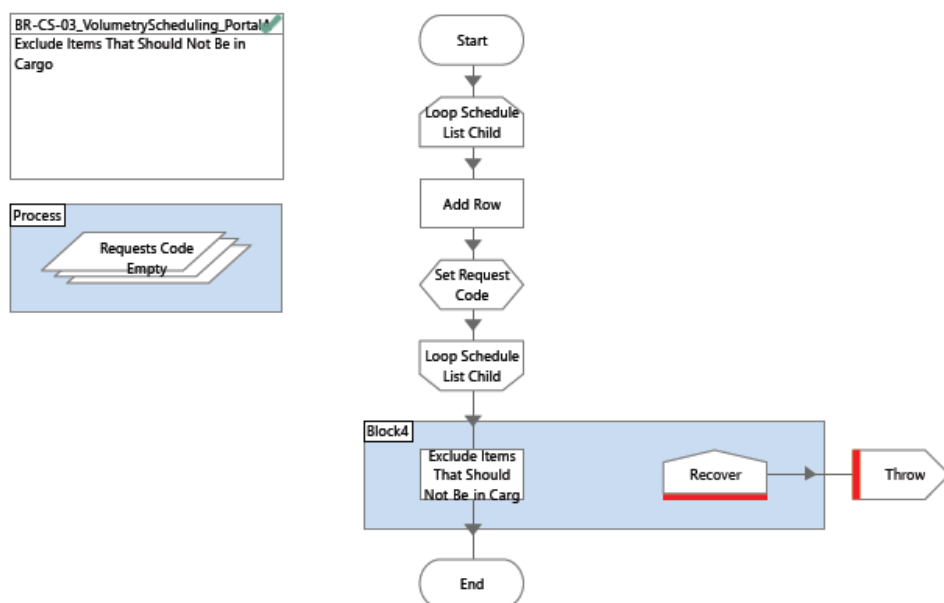
Fonte: Elaborado pelo autor.

Em alguns casos, pode acontecer de que a carga com os pedidos já foi agendada no portal por algum membro da área de CS e não foi atualizado na planilha que o robô se baseia, tornando este um caso de *Business Exception*. A *action Check If It Is Possible to Schedule The Item* verifica se essa situação aconteceu e lança uma exceção caso isso seja verdade. Ao contrário, o fluxo segue normalmente para a página subsequente.

Conforme descrito anteriormente, quando acontece uma exceção em algum estágio dentro dos blocos em azul na Figura 16, o fluxo segue imediatamente para o *recover* para que a exceção seja tratada na própria automação. O estágio *recover* desta mesma figura tem o fluxo para um *loop* com o objetivo de uma segunda tentativa em caso de exceção, cuja variável *Retry Count* é incrementada como um sistema de segurança para que este *loop* não seja infinito e siga para a *Exception Throw*. Por outro lado, exceções que caem no *recover2* são marcadas diretamente como *Business* ou *System Exceptions*.

Logo na sequência, a subpágina *Exclude Unecessary Items*, mostrada na Figura 17, tem como propósito excluir eventuais produtos que não devem estar no pedido e preservar os demais produtos que devem estar.

Figura 17 – Página Exclude Unecessary Items.



Fonte: Elaborado pelo autor.

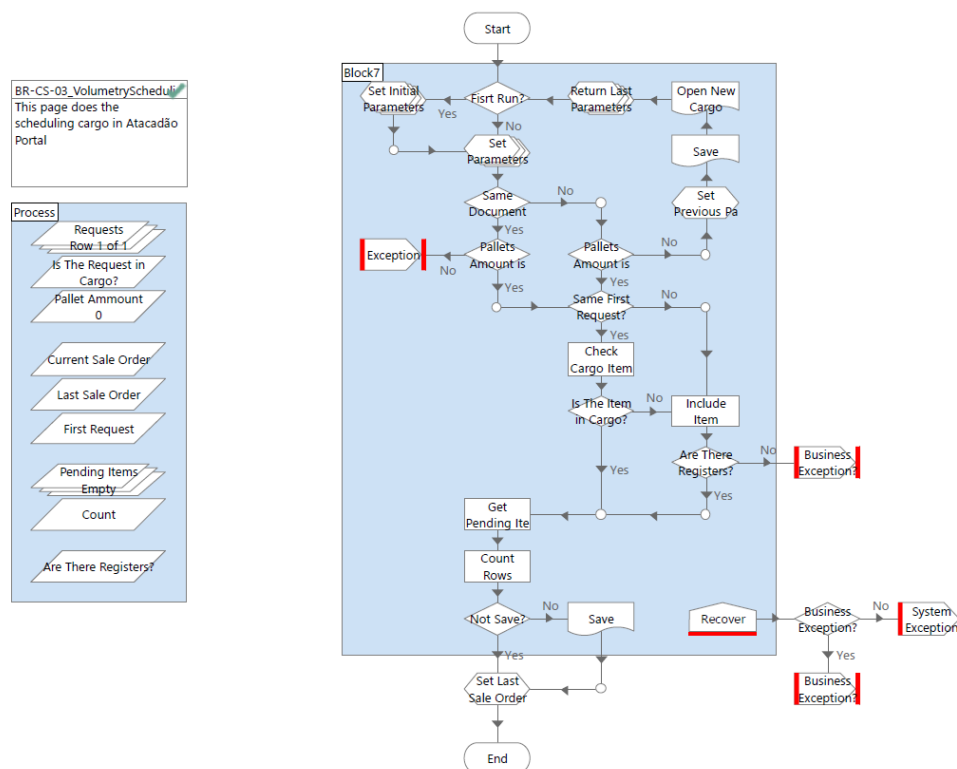
O estágio *Exclude Items That Should Not Be in Cargo* é responsável pela interação do robô com o portal para excluir estes pedidos não desejados. Essa *action* tem como parâmetros de entrada a *collection Requests Code*, que é totalmente baseada nos códigos dos produtos da planilha de agendamentos. Sendo assim, a dinâmica desenhada para o robô nesta *action* é verificar todos os produtos registrados em *Requests Code* e excluir produtos selecionados no portal que não estão dentro desta lista.

3.4.1.4 Montar Carga

Depois que a etapa anterior do processo foi realizada, segue-se para a página principal do projeto *Check Items Cargo* de acordo com a Figura 18. É nesta página em

que se define a maioria das regras de negócio do projeto, além de checar e montar a carga para fazer o agendamento.

Figura 18 – Página Check Items Cargo.



Fonte: Elaborado pelo autor.

A página é composta por uma série de regras de negócio representadas pelos *decisions* e *calculate stages* no início do fluxo da Figura 18. Esses estágio são importantes para o funcionamento correto da automação com relação à manipulação de dados que o robô faz e também para o gerenciamento das informações feito pela área de CS. Por isso, serão descritos brevemente com o intuito de focar nas interações que a automação têm com o portal Atacadão.

Toda a página é baseada em duas regras de negócio que são essenciais para o correto funcionamento da automação: a quantidade de produtos máxima de uma carga e a divisão correta dos pedidos nas cargas. A primeira regra é apoiada na quantidade máxima de produtos que cabem em uma só carga, cuja unidade de medida é a quantidade de *pallets* dos pedidos. A segunda regra diz respeito à divisão dos pedidos em cada carga, sendo que pedidos diferentes com a mesma ordem de venda (documento que define os detalhes e produtos de uma venda), obrigatoriamente devem ser agendados na mesma carga.

Sabendo disso, os primeiros estágios *multicalculations*, assim como o *First Run?*, servem para definir os parâmetros iniciais da execução, que são obtidos através da planilha

de agendamentos. Logo, na sequência, o *decision Same document?* direciona o pedido de acordo com a ordem de venda da última execução. Portanto, quando se trata do mesmo documento da execução anterior a automação obrigatoriamente deve manter o pedido na mesma carga e se isso não for possível, o fluxo leva diretamente para uma exceção de acordo Figura 18.

Por outro lado, quando não se trata do mesmo documento, é possível agendar os pedidos na mesma carga ou em cargas diferentes, nos casos em que a atual já esteja cheia. Nesta situação, o fluxo da automação se torna um *loop* que passa por alguns *calculations stage* para definir parâmetros e variáveis, passando também pela página *Save* que será discutida posteriormente e finalmente, pela página *Open New Cargo* que já foi discutida.

O ponto em comum dos possíveis fluxos citados anteriormente é o *decision Same First request?* cujo parâmetro de decisão é o número do pedido da última execução. A partir daí, existem dois caminhos possíveis que são determinados por este *decision*. Se o pedido é o mesmo da execução anterior, o fluxo direciona para a *action Check Cargo Item* que verifica e ajusta a quantidade de *pallets* dos produtos deste pedido e ainda retorna uma *flag* caso o produto não esteja na carga, para que seja possível adicioná-lo. Caso contrário, quando é necessário incluir um novo pedido na carga, o fluxo correto segue para a *action Include Item* que abre um novo pedido e inclui a quantidade do produto na carga. Ao tentar incluir um produto que não existe na carga, a *action* retorna uma *flag* que direciona o fluxo da execução diretamente para a exceção depois do *decision Are There Registers?*.

Por fim, as últimas etapas na página da Figura 18 dizem respeito aos produtos restantes que ainda devem ser inseridos na mesma carga. Sendo assim, a *action Get Pending Items* identifica itens pendentes que ainda não foram executados na fila filho (fila que contém as informações de cada produto) e retorna uma *collection* com a quantidade de produtos que ainda devem ser agendados na carga. A quantidade de itens pendentes serve como parâmetro de decisão no estágio *Not Save?*, que deve seguir o fluxo no sentido do *Yes* quando ainda há produtos para agendar na mesma carga e sentido *No* quando não há mais produtos, sendo assim, deve-se finalizar o agendamento da carga.

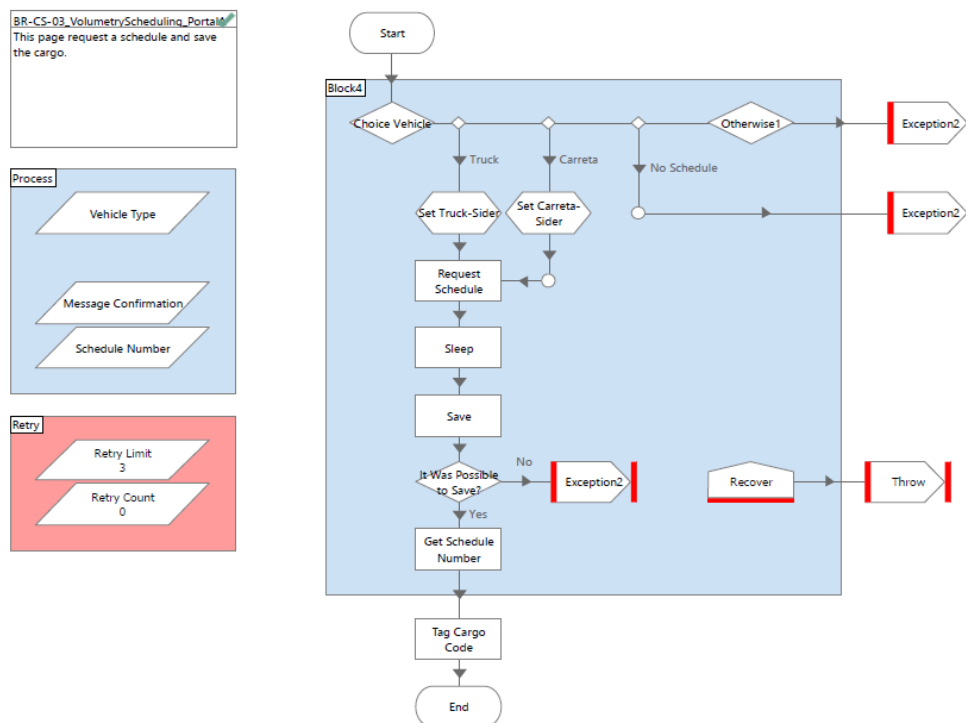
Se existir alguma exceção no fluxo do robô antes de finalizar o agendamento da carga, ele segue diretamente para o estágio *recover* e é marcada pela própria automação como sendo *System* ou *Business Exception*, além de trazer também os detalhes e o motivo causador da exceção.

3.4.1.5 Salvar Carga

A página da Figura 19 é a etapa final do processo de agendamento no portal, onde fecha-se a carga e faz-se a requisição do agendamento. Ela é iniciada com um *choice stage* nomeado por *Choice Vehicle*, que serve como uma regra de negócio solicitada pela área de CS para selecionar o tipo de veículo que transportará a carga agendada, cujo critério

de escolha é dado pelo tamanho da carga, de acordo com o Quadro 1.

Figura 19 – Página Save.



Fonte: Elaborado pelo autor.

Cargas inferiores a 10 *pallets* não são agendadas pois são consideradas cargas pequenas. Deve-se então aguardar para que a loja faça mais pedidos e aí sim, fazer o agendamento com a automação. Da mesma forma, cargas superiores a 28 *pallets* também não devem ser agendadas por serem cargas grandes demais para fazer o transporte e descarregar no local da entrega. Essas duas situações normalmente não acontecem pois é a própria área de CS que alimenta a planilha de agendamento e define o status de cada pedido, sendo assim, eles conseguem controlar quais cargas devem e não devem ser agendadas, além de controlar também o tamanho das cargas de acordo com os pedidos inseridos em cada uma. Contudo, se por um acaso acontecer uma dessas situações na execução do robô, o fluxo da automação é direcionado diretamente para as *exceptions*, de acordo com a Figura 19.

Por outro lado, cargas entre 10 e 16 *pallets* devem ser agendadas e transportadas por um *Truck-sider*, veículo de transporte médio. Cargas entre 16 e 28 *pallets* devem ser transportadas por *Carreta-Sider*, que é um veículo maior, mais robusto e aguenta transportar maiores pesos.

Depois da definição do veículo de transporte, faz-se a requisição do agendamento através da *action Request Schedule*, cujas interações com o portal são justamente selecionar o tipo de veículo definido pela regra de negócio mencionada, definir a transportadora e

Quadro 1 – Distribuição de cargas em veículos.

Tamanho da Carga (X <i>Pallets</i>)	Veículo
$X < 10$	Não Agendar
$10 \leq X < 16$	Truck-Sider
$16 \leq X \leq 28$	Carreta-Sider
$X > 28$	Não Agendar

Fonte: Elaborado pelo autor.

inserir as datas para o agendamento da carga que são obtidas através da planilha. Logo após, a *action sleep* serve para aguardar um período de tempo até a próxima interação.

Por fim, a *action Save* salva o agendamento da carga no portal, que imediatamente retorna um *pop up* com a mensagem de confirmação que o agendamento da carga foi registrado com sucesso. Sendo assim, a *action* retorna essa mensagem que é utilizada logo na sequência como parâmetro para o *decision It Was Possible to Save?*, que direciona o fluxo da automação para a próxima interação em casos de sucesso ao salvar a carga ou imediatamente lança uma exceção nos casos em que não foi possível concluir o agendamento.

Logo depois de receber a mensagem de confirmação do agendamento bem sucedido, o usuário é redirecionado para a página *web* da carga que acaba de ser agendada. Lá é possível verificar todas as informações que foram utilizados para fazer o agendamento, como também é definido um código identificador da carga. A *action Get Schedule Number* retorna justamente esse código identificador que é guardado dentro de uma variável e posteriormente será utilizado pela automação para gerar relatórios de todas as cargas agendadas. Além disso, a *action Tag Cargo Code* adiciona esse código na *tag* do item da fila pai (fila com as informações de cada carga).

3.4.2 Subprocessos

Ainda há alguns subprocessos que não foram detalhados neste documento porque não fazem parte do escopo da automação no portal Atacadão. Contudo são muito importantes e até mesmo indispensáveis para o processo como um todo, tornando-o mais robusto e completo. Estes tópicos são tratados de forma breve nesta subseção.

Existem dois subprocessos dentro do projeto de agendamento Atacadão que são dedicados para marcar os pedidos agendados no SAP e também na própria planilha que alimenta a automação. Estes subprocessos são genéricos e podem ser utilizados em outros processos para fazer as marcações de itens na planilha e no SAP, dado os *inputs* corretos para cada um.

3.4.2.1 Subprocesso Update Status SAP

Esse subprocesso consiste em atualizar o status dos produtos que foram agendados nas cargas, na transação ‘ZSGCML0291’ do SAP. Portanto ele também pode ser utilizado em outros projetos que fazem a marcação de status na mesma transação. Contudo ele só funciona de forma efetiva se os *inputs* do processo estão corretos e não possuem nenhum tipo de erro.

Figura 20 – Inputs subprocesso Update Status SAP.

Process Properties

Name: Update Status SAP

Description:

Process: BSA_BR_BBS_OTC_CustomerService_SchedulingUpdateStatusSAP

Group: Page Data Type View All Items

Name	Data Type	Value
Reason Description	Text	"SOLICITADO AGENDA"
Description Note	Text	
Schedule List	Collection	[Schedule List Child]
Description Item Key	Text	"*ATACADAO"
Close Sap	Flag	False

Binaries

- Collections
- Dates
- DateTimes
- Flags
- Images
- Numbers
- Passwords
- Text
- Times
- TimeSpans

Stage logging: Enabled

Warning threshold: System Default Number of minutes 5 (0 to disable)

OK Cancel

Fonte: Elaborado pelo autor.

Os *inputs* fornecidos pelo processo de agendamento são a *collection Schedule List Child*, dois parâmetros de texto e uma *flag* como pode ser observado na Figura 20. A *collection Schedule List Child* é justamente todo o relatório contendo os dados e informações de cada pedido para agendamento nas lojas clientes Atacadão. Dentro desta *collection* existe uma coluna que indica justamente o número de cada pedido feito pelas lojas, sendo que este número é exatamente o mesmo registrado no SAP da Bunge. Portanto, o subprocesso de atualização de status busca o número de todos os pedidos que foram agendados no portal e trabalha com um de cada vez através de uma fila. Dessa forma, a automação busca o número do pedido na transação ‘ZSGCML0291’ e marca este pedido como ‘SOLICITADO AGENDA’ dentro do SAP.

O parâmetro *Description Item Key* da Figura 20 é utilizado somente na fila de trabalho do processo *Update Status SAP* para distinguir os itens da fila que foram criados

a partir do processo de agendamento Atacadão. Além disso, a *flag Close Sap* tem o parâmetro de *false* para certificar que o processo de atualização não feche o SAP após o final do seu processamento, sendo assim, no próximo ciclo de execução o SAP já estará aberto ganhando desempenho para o conjunto dos processos.

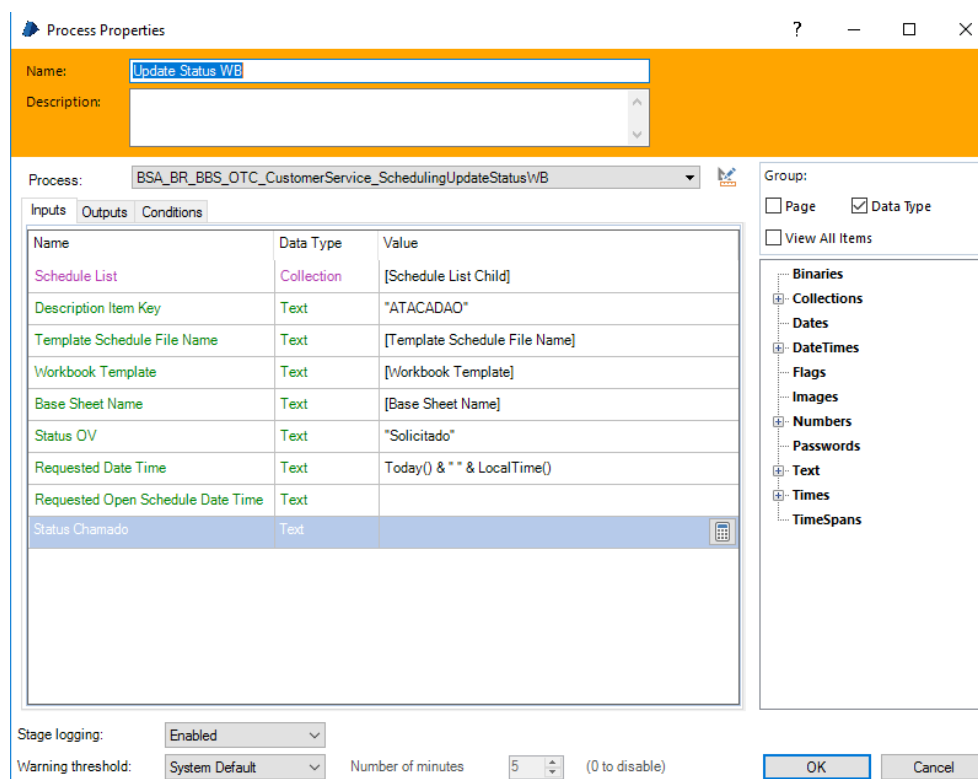
Este subprocesso só é iniciado pelo processo de agendamento quando todos os pedidos da carga foram processados corretamente e a carga já foi agendada.

3.4.2.2 Subprocesso Update Status WB

O subprocesso *Update Status WB* é parecido com o anterior, sendo que o intuito deste fluxo é marcar na planilha de agendamentos todos os pedidos processados pela automação no portal do Atacadão.

Da mesma forma, a *collection Schedule List Child* mostrada na Figura 21, contém todas as informações dos pedidos agendados com sucesso e inclusive as informações dos pedidos que não foram processados por completo devido a algum tipo de exceção. Essas exceções são relacionadas e marcadas com cada pedido dentro da *collection* através da página *Set Customer Template Error* que será descrita posteriormente.

Figura 21 – Inputs subprocesso Update Status Wb.



Fonte: Elaborado pelo autor.

Além disso, quando são populadas as filas e também a *collection Schedule List Child*, a própria automação grava o *index* (número da linha) de cada produto da planilha

que será trabalhado em uma coluna dentro desta *collection*. Sendo assim, através do *index* fica fácil retornar a planilha de agendamentos e localizar cada produto para marcá-lo de acordo com seu status.

Portanto, os *inputs* do subprocesso são o nome e caminho da planilha de agendamentos, além do dia e da hora em que está sendo realizado a execução deste subprocesso, de acordo com a Figura 21. Ademais, a *collection Schedule List Child* contém todos os pedidos que serão processados como já explicado anteriormente, sendo que a fila tem sua *tag* marcada por ‘ATACADAO’ para distinguir a automação que chama o subprocesso *Update Status WB*.

Por fim, o ‘status da OV’ é o status que será marcado em cada pedido de acordo com o seu processamento. Quando a execução do pedido ocorre sem nenhum tipo de falha ele é marcado como *Completed* na fila de trabalho e ‘Solicitado’ na planilha, Figura 21. Por outro lado, quando existe algum tipo de erro no processamento do pedido ele é marcado como *Exception* na fila de trabalho e ‘Exceção’ na planilha de agendamentos. Para isso, existem dois caminhos possíveis que chamam o subprocesso *Update Status WB* de acordo com a Figura 14, onde o caminho em que houve o processamento correto dos pedidos e cargas tem como status o *input* de ‘Solicitado’, enquanto o outro caminho tem como status o *input* de ‘Exceção’.

3.4.3 Tratamento das Exceções

Marcar os itens que não processaram corretamente como exceção nas filas é muito importante para o gerenciamento e indicadores de resultados, feito pelo time de *control room*. Sendo assim, é essencial definir uma lógica para o processamento e gerenciamento das exceções no processo. A página *Mark Parent Item As Exception* funciona estabelecendo o status dos itens da fila pai (cargas), distingue o tipo e define o detalhe da exceção. Da mesma forma funciona a página *Mark Child Item As Exception* que marca os itens filhos (produtos) como exceção. Contudo, existe uma pequena diferença no desenvolvimento destas páginas para ganhar desempenho durante a execução.

A lógica por trás da página *Mark Child Item As Exception* é simples e é baseada na premissa em que não se deve fazer o agendamento de uma carga se um dos pedidos desta carga não foi incluído com sucesso. Sendo assim, se existir algum pedido que foi marcado como exceção em uma determinada carga, a própria automação nem chega a processar os demais pedidos desta mesma carga e marca-os diretamente como exceção na fila de trabalho. Essa ideia de marcação pode ser observada de acordo com a Figura 22, cujos dois primeiros pedidos da carga foram incluídos com sucesso, o terceiro pedido foi marcado como uma exceção do tipo *internal* devido a algum erro na inclusão. Portanto, o robô nem chega a processar os demais itens e marca-os como exceção de forma direta.

Essa lógica foi implementada através de um *loop* de acordo com a Figura 23. Inicialmente, a sequência das *actions Update Status, Tag Item e Mark Exception* verifica

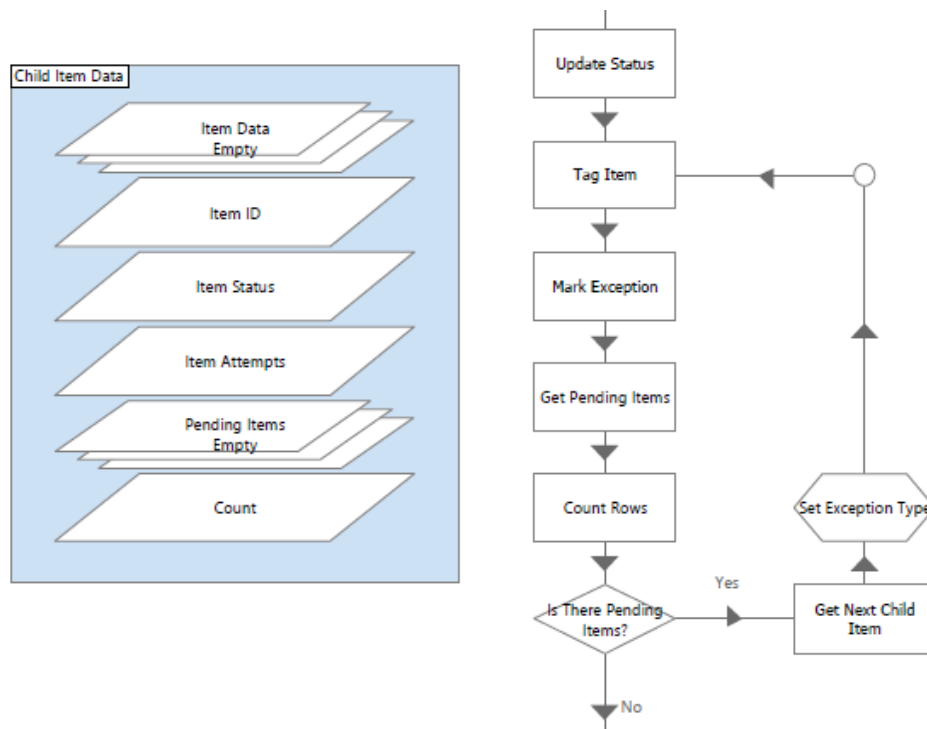
Figura 22 – Fila Filho - Pedidos da Carga.

▶	19566456 - 6969	0	Pending	1059251 - 02/06/2022	HBBWRPAP004	1	30/05/2022 20:13:47	30/05/2022 20:14:28
▶	19566456 - 9693	0	Pending	1059251 - 02/06/2022	HBBWRPAP004	1	30/05/2022 20:13:47	30/05/2022 20:14:28
▶	19566456 - 12925	0	Pending	1059251 - 02/06/2022	HBBWRPAP004	1	30/05/2022 20:13:47	30/05/2022 20:14:28
▶	19566456 - 9692	0	Internal	1059251 - 02/06/2022	HBBWRPAP004	1	30/05/2022 20:13:47	30/05/2022 20:14:28
✓	19566456 - 47853	0	Completed	1059251 - 02/06/2022	HBBWRPAP004	1	30/05/2022 20:13:47	30/05/2022 20:14:28
✓	19566456 - 9295	0	Completed	1059251 - 02/06/2022	HBBWRPAP004	1	30/05/2022 20:13:47	30/05/2022 20:14:19

Fonte: Elaborado pelo autor.

o produto que não foi incluído com sucesso na carga e modifica o status deste produto na fila filho de acordo com o tipo e detalhe da exceção que ocorreu durante o processamento. Logo após, o conjunto das *actions* subsequentes (*Get Pending Items* e *Count Rows*) salva na *colletion Pending Items* todos os demais produtos da carga que ainda não foram processados para que também sejam marcados como e exceção. Portanto, a lógica em *loop* utiliza os itens da *colletion* e faz a marcação de exceção de todos os produtos da mesma carga de uma só vez.

Figura 23 – Página Mark Child Item As Exception.



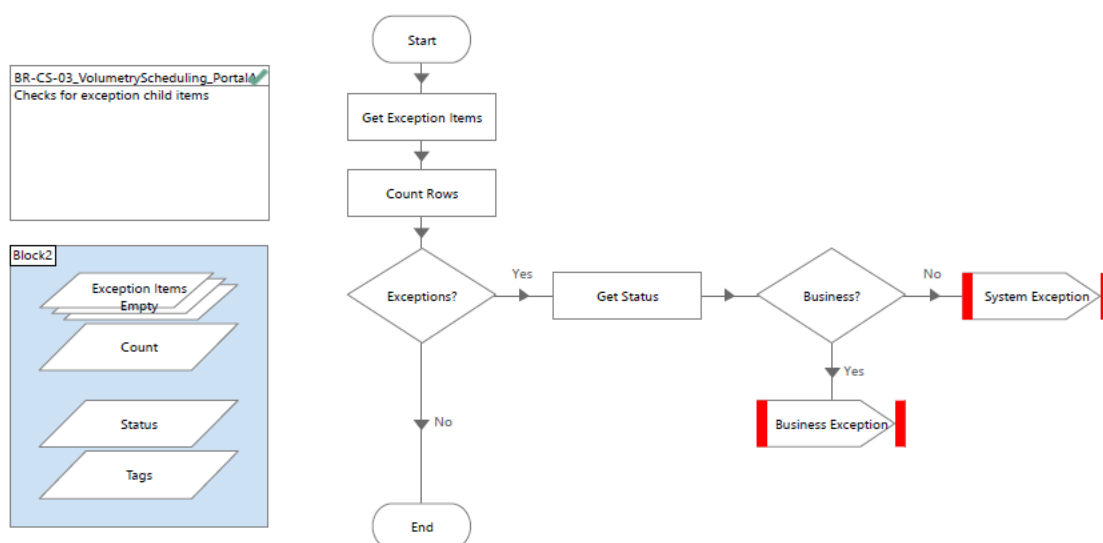
Fonte: Elaborado pelo autor.

A ideia de marcar todos os produtos restantes da carga de uma só vez como exceção foi implantada justamente nessa página para que no próximo ciclo de execução o robô não encontre mais itens pendentes na fila. Sendo assim, não se perde tempo tentando incluir os demais produtos na carga pois ela já estará imprópria para o agendamento.

A página *Check Child in Exceptions*, mostrada na Figura 24, também faz parte do sistema de tratamento das exceções e é responsável por checar se existe algum item da fila filho que não foi processado corretamente. Se isso acontecer, o robô imediatamente marca também o item da fila pai, como exceção. Ou seja, se existir algum pedido ou produto que não foi incluído corretamente na carga por algum motivo, o robô marca toda essa carga como exceção e não faz o agendamento.

Para fazer isso, inicialmente a *action Get Exception Itens* busca todos os itens que foram marcados como exceção na fila filho e salva-os na *collection Exception Items*, que por sua vez é verificada pelos estágios subsequentes. Nas situações onde não há exceções nos itens da fila filho, o fluxo da página segue em direção ao *end* e a automação se encaminha normalmente. Contudo, se existirem exceções o fluxo caminha em direção a *action Get Status* que traz algumas informações como o tipo e detalhe da exceção. Essas informações servem como parâmetros para o *decision Business?* disparar a *Exception* que ocorreu na execução dos itens filhos.

Figura 24 – Check Childs In Exception.



Fonte: Elaborado pelo autor.

Quando ocorre alguma exceção dentro desta página o fluxo da automação não vai diretamente para o subprocesso *Update Status SAP* como normalmente faria, mas é direcionado imediatamente para o *Recover1* de acordo com a Figura 14. Esse novo fluxo se encaminha para a página *Mark Parent Item As Exception*, que marca a carga total como exceção, como já foi explicado.

Por fim, a página *Set Customer Template Error* serve justamente para definir o detalhe e tipo da exceção, isto é, o erro que aconteceu durante a execução do item. Sendo assim, ela é responsável por especificar as exceções que aconteceram em cada pedido da

carga e gravá-los em uma *collection*, que é utilizada como *input* do subprocesso *Update Status WB*, que já foi explicado anteriormente.

3.4.4 Página de Relatórios

A página *Create Reports* serve para atualizar um relatório com algumas informações de todas as cargas que foram agendadas com sucesso no portal, conforme a Figura 25. Além disso, ela modifica também um segundo relatório com as informações dos pedidos de cada carga, conforme a Figura 26. Ambos os relatórios são salvos em um diretório de rede compartilhado. A página funciona inicialmente filtrando todas as cargas marcadas como completas na fila pai, que são gravadas dentro de uma *collection*. Cada carga possui um elemento identificador (*item-key*) na fila de trabalho que pode ser utilizado para buscar todas as informações contidas neste item. Estas informações são o *status* do item, *tag*, tipo e detalhe da exceção, data e hora que o item foi completa, item ID, entre outras.

Após buscar todas as informações, a automação atualiza ambos os relatórios com todos os itens processados na execução, pontuando todas essas informações descritas. Além disso, busca-se em especial a informação mais importante, que é o número da carga que foi gerado logo após o final do agendamento no portal.

Por fim, as cargas do primeiro relatório (Figura 25) estão ligadas aos produtos e pedidos do segundo relatório (Figura 26) através de um elemento identificador comum. Sendo assim, pode-se confrontar as informações das cargas com os detalhes dos pedidos. Ou seja, é possível relacionar cada carga com todos os produtos e pedidos referentes a ela através do elemento identificador comum das duas tabelas. Esses dados serão utilizados em conjunto no aplicativo.

3.4.5 Estágios Complementares da Página Inicial

O conjunto da *action Get Next Item* e do *desicion Got Item?*, mostrados na Figura 14, são utilizados respectivamente para coletar o primeiro ou o próximo item pendente de uma fila, bem como todos os dados referente a este item. Portanto, utiliza-se esse conjunto para que seja possível trabalhar com todas as informações disponíveis e também para verificar se uma fila está populada ou está vazia.

É possível verificar também na Figura 14 alguns *calculations* e *multicalculations* que são utilizados para definir parâmetros importantes para o funcionamento do processo. Algumas destas funções são a hora de início do processo, um contador que limita o número de tentativas de execução e um indicador dos dados de exceções, como o tipo e detalhe da exceção.

A última página do processo *Close Down* mostrada na Figura 14, é utilizada simplesmente para fechar as aplicações que o robô interage, isto é, o navegador com portal Atacadão, a planilha de agendamentos e o SAP.

3.5 POWER APPS

Esta seção é dividida em subseções que mostram algumas etapas do desenvolvimento do aplicativo utilizado para consultar e ajustar as cargas que foram agendadas através da automação com RPA. Além disso, demonstram-se as fontes de dados e explica-se como o Power Apps faz a relação entre as cargas e produtos.

3.5.1 Fonte de Dados

O desenvolvimento do aplicativo de consulta e ajuste das cargas é baseado nos dois relatórios que o RPA entrega no final do processamento através da página *Create Reports* (descrita na Seção 3.4.4), que podem ser vistos nas Figuras 25 e 26. Um dos relatórios é chamado de ‘CargasAtacadao’ e contém as informações das cargas que foram solicitadas no portal. O segundo relatório é chamado de ‘ProdutosAtacadao’ e contém todos os produtos e pedidos das cargas.

Figura 25 – Tabela de Cargas.

ChildItem	CódigoCarga	CódigoEmissorOrdem	CidadeEmissorOrdem	Estado	Nome	DataAgendar	DataSolicitada
1	2022051739609	2140511	SANTAREM	PA	ATACADAO S.A.	23/05/2022	17/05/2022 18:50:34
2	2022052032495	2040540	MANAUS	AM	ATACADAO S.A.	26/05/2022	20/05/2022 13:22:10
3	2022051739650	1004050	CUIABA	MT	ATACADAO S.A.	27/05/2022	20/05/2022 13:19:03
4	2022051739730	1004063	ARACAJU	SE	ATACADAO S.A.	27/05/2022	20/05/2022 13:21:00
5	2022052006893	1004063	ARACAJU	SE	ATACADAO S.A.	26/05/2022	20/05/2022 13:22:10
6	2022052007504	1004065	JUAZEIRO	BA	ATACADAO S.A.	25/05/2022	20/05/2022 13:24:04
7	2022052007890	1059250	LAURO DE FREITAS	BA	ATACADAO S.A.	30/05/2022	20/05/2022 13:25:12
8	2022052032363	1059260	FEIRA DE SANTANA	BA	ATACADAO S.A.	30/05/2022	20/05/2022 13:26:25
9	2022052032550	1059268	FEIRA DE SANTANA	BA	ATACADAO S.A.	25/05/2022	20/05/2022 13:28:14
10	2022052032630	1059270	JUAZEIRO DO NORTE	CE	ATACADAO S.A.	30/05/2022	20/05/2022 13:30:49
11	2022052032703	1059278	EUNAPOLIS	BA	ATACADAO S.A.	25/05/2022	20/05/2022 18:41:32
12	2022052032789	1059279	VITORIA DA CONQUISTA	BA	ATACADAO S.A.	30/05/2022	21/05/2022 10:09:20
13	2022052032940	1059279	VITORIA DA CONQUISTA	BA	ATACADAO S.A.	26/05/2022	21/05/2022 10:10:21
14	2022052033025	1059295	FEIRA DE SANTANA	BA	ATACADAO S.A.	25/05/2022	23/05/2022 14:55:55

Fonte: Elaborado pelo autor.

Estes relatórios são formatados pela automação de agendamento que é responsável por definir o elemento identificador comum que relaciona cada item da tabela de cargas com os itens da tabela de produtos, de acordo com a coluna A na Figura 25 e coluna B na Figura 26. Portanto, observando as duas figuras, pode-se perceber que cada item da carga pode ter um ou mais itens de produtos.

Além disso, o Power Apps faz a leitura de uma tabela e distingue cada linha através de um código único que não pode se repetir dentro da mesma tabela em nenhuma hipótese. Estes códigos estão na primeira coluna de cada tabela de acordo com as Figuras 25 e 26. O código tem a funcionalidade de um índice para o aplicativo e se faz necessário para que o Power Apps consiga se conectar com cada linha da tabela.

Figura 26 – Tabela de Produtos.

ItemKey	ParentItem	CodigoProduto	TextoBreveMaterial	QuantidadeFornecer	QuantidadePallets	DocumentoSD	NumPedido
1	1	147745	FT TIPO 1 PRIMOR PLAS TRAV 10X1KG BR	360	8,2	19034207	32241
2	2	151089	FT SUPREMA ESPECIAL PAP 25KG BR	60	1,5	19034210	32241
3	3	152479	FT PREDILETA 25KG PAP	60	1,5	19034211	32241
4	4	135086	FT PIZZA SUPREMA PLA 5X5KG BR	100	1	19034222	32241
5	5	135123	FT PASTEL SUPREMA PLA 5X5KG BR	800	8	19034223	32241
6	6	147723	FT TIPO 1 FERM PRIMOR PLAS TRA 10X1KG BR	80	1,8	18867777	32241
7	7	147745	FT TIPO 1 PRIMOR PLAS TRAV 10X1KG BR	50	5	19240802	340827
8	8	151089	FT SUPREMA ESPECIAL PAP 25KG BR	15	8	19240803	340827
9	9	152479	FT PREDILETA 25KG PAP	35	7	19240804	340827
10	10	503534	FT LASANHA SUPREMA PLA 5X5KG BR	100	5	19269846	340827
11	11	135086	FT PIZZA SUPREMA PLA 5X5KG BR	20	0,67	19017694	834310
12	12	135123	FT PASTEL SUPREMA PLA 5X5KG BR	40	1,33	19025653	933776
13	13	147723	FT TIPO 1 FERM PRIMOR PLAS TRA 10X1KG BR	300	3	19025654	933776
14	14	147745	FT TIPO 1 PRIMOR PLAS TRAV 10X1KG BR	400	4	19240649	951090
15	15	152479	FT PREDILETA 25KG PAP	44	1,1	19240644	951087
16	16	153309	FT ESPECIAL ASSAI CHEF PAP 25KG BR	300	3	19240645	951088
17	17	147745	FT TIPO 1 PRIMOR PLAS TRAV 10X1KG BR	750	7,5	19240650	951091
18	18	147723	FT TIPO 1 FERM PRIMOR PLAS TRA 10X1KG BR	350	3,5	18867566	508002
19	19	135123	FT PASTEL SUPREMA PLA 5X5KG BR	100	3,33	19240651	951092
20	20	135086	FT PIZZA SUPREMA PLA 5X5KG BR	120	4	19025989	933781
21	21	135086	FT PIZZA SUPREMA PLA 5X5KG BR	120	4	19025990	933781
22	22	135086	FT PIZZA SUPREMA PLA 5X5KG BR	120	4	19240654	951095
23	23	135123	FT PASTEL SUPREMA PLA 5X5KG BR	180	6	18866460	330413

Fonte: Elaborado pelo autor.

3.5.2 Desenvolvimento do Aplicativo

A função do aplicativo é fazer a consulta e ajuste das cargas que a automação processa de forma rápida. Sendo assim, os funcionários da área de CS podem fazer as consultas mais facilmente utilizando o *app* ao invés da planilha de agendamentos. O aplicativo é baseado em telas que são programadas para mostrar e editar as informações das fontes de dados utilizadas no projeto. O desenvolvimento destas telas será descrito a seguir.

3.5.2.1 Página Inicial

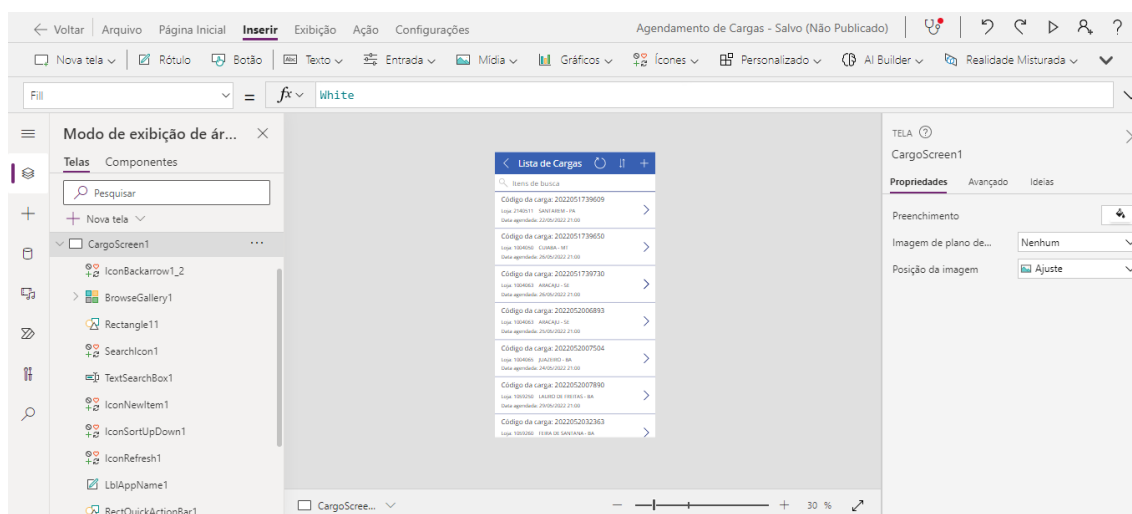
A tela inicial do aplicativo, mostrada na Figura 5, tem um conjunto de componentes básicos como um retângulo e uma série de rótulos, que servem para apresentar algumas informações do aplicativo, como: o nome, a versão e o desenvolvedor do *app*. Além disso existe ainda um botão chamado Lista das Cargas que tem como função interna navegar para a tela subsequente quando é pressionado.

3.5.2.2 Lista de Cargas

Logo na sequência, a tela da Figura 27 tem uma série de componentes para a interação do usuário com as cargas que foram agendadas pela automação. Toda a tabela de cargas é mostrada nessa tela através da galeria *BrowseGallery1*, que pode ser vista no canto esquerdo da figura.

Dentro da galeria, é possível inserir rótulos que apresentam as informações das cargas disponíveis para a visualização do usuário, como pode se observar na tela central da Figura 27. Sendo assim, ao inserir a galeria nessa tela, o Power Apps requisita a fonte

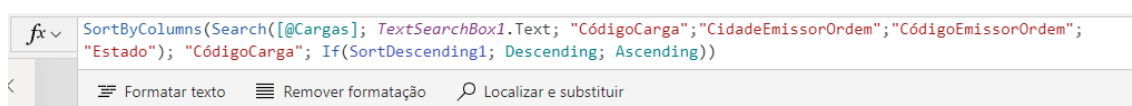
Figura 27 – Tela da Lista de Cargas.



Fonte: Elaborado pelo autor.

de dados que é colhida as informações e os campos que deverão ficar visíveis. O *layout* da galeria é organizado de acordo com o código da carga, cidade e estado da loja destino, além da data em que foi feito o agendamento. Essa tela ainda tem algumas funções como pesquisar os itens da galeria (cargas agendadas) de acordo com o texto de entrada escrito pelo usuário na barra de pesquisa, além de ordenar os itens conforme o código da carga. Essas funcionalidades são possíveis por conta função da Figura 28 aplicada sobre a galeria.

Figura 28 – Função Aplicada Sobre a Galeria de Cargas.



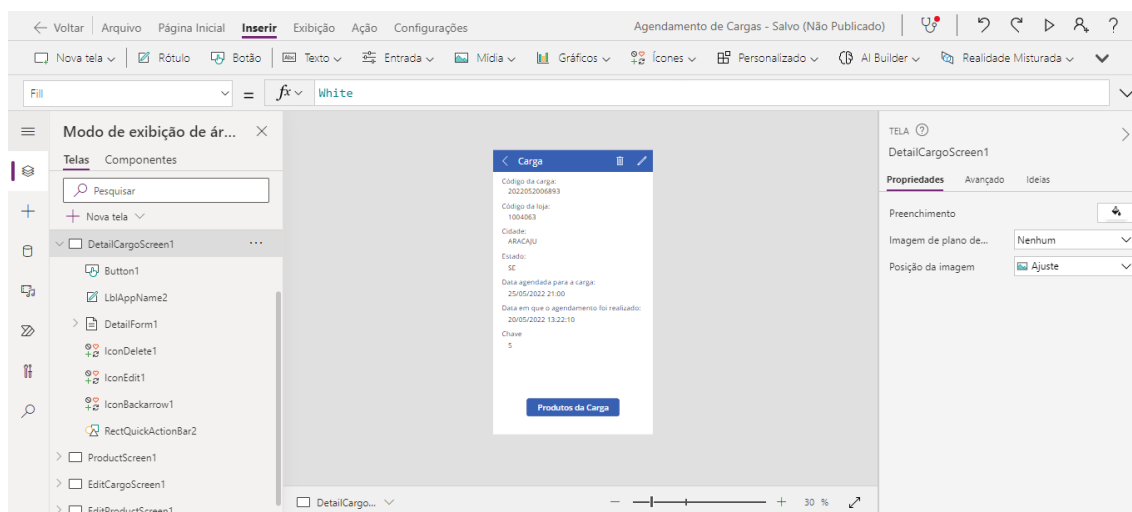
Fonte: Elaborado pelo autor.

Ainda é possível atualizar a galeria através da fórmula *Refresh* aplicada sobre o botão de recarregar e retornar para a tela inicial do aplicativo pressionando o botão de voltar. Para adicionar uma nova carga, é acrescentado um botão com a função *NewForm* que abre um novo formulário para adicionar os itens, além de conter ao mesmo tempo a função para navegar até a página Editar Cargas que será descrita posteriormente. Por fim, existe ainda o botão para verificar os detalhes de cada carga, que redireciona o usuário para a página Cargas quando é pressionado.

3.5.2.3 Detalhes da Cargas

A tela que mostra os detalhes da carga tem como principais componentes um formulário com todas as informações da carga selecionada na página anterior. Esses dados são posicionados de acordo com o *layout* da Figura 29.

Figura 29 – Tela de Detalhes da Carga.



Fonte: Elaborado pelo autor.

Existe a opção de retornar para a tela anterior através do botão de voltar que tem a função de *Navigate*, sendo que, essa função também está presente juntamente com *EditForm* no botão de editar. O botão de excluir tem a função *Remove*, cujo parâmetro é a carga selecionada dentro da galeria *BrowseGallery1*. Ou seja, quando pressionado, a função do botão remove a carga da galeria e conseqüentemente da fonte de dados. Por fim, o botão *Produtos da Carga* navega para a página subsequente que será explicada a seguir.

3.5.2.4 Produtos da Carga

A tela de produtos da carga é baseada em uma galeria chamada *BrowseGallery2* que contém toda a tabela de dados do relatório de produtos. Sendo assim, é necessário filtrar a galeria para que nessa página apareça somente as informações dos produtos que fazem parte da carga selecionada anteriormente, por isso é necessário utilizar a função *Filter* e fornecer como parâmetro o elemento identificador comum dos produtos que pertencem à carga escolhida.

Além disso, é feito uma combinação da função *Filter* com a função *Search* para facilitar a busca de um pedido em específico dentro da galeria. Esse conjunto pode ser visto de acordo com a Figura 30.

É possível ainda editar ou excluir cada produto dentro da galeria pressionando os respectivos botões que aparecem em cada item. Além de que, nessa página também existem

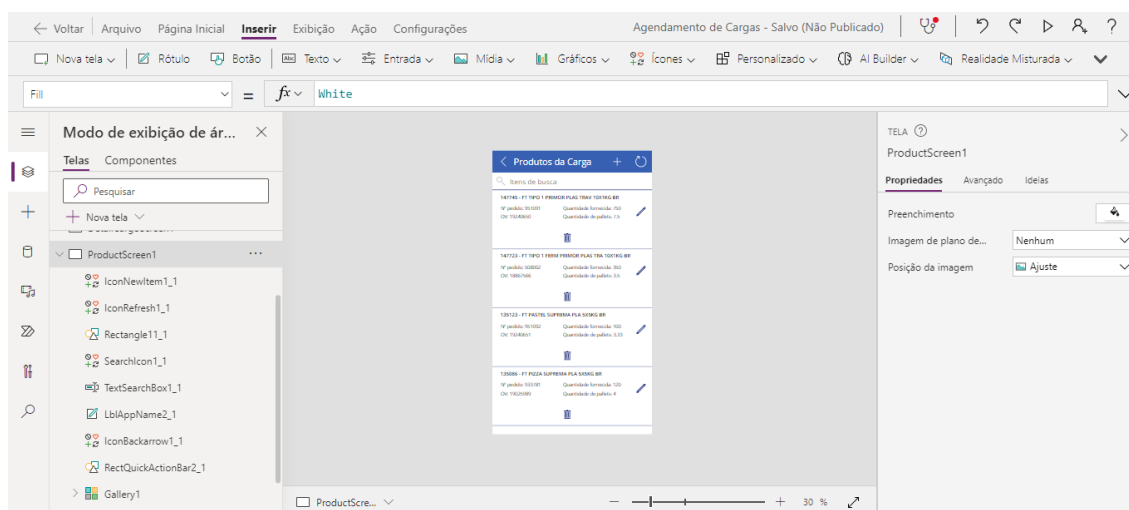
Figura 30 – Função Aplicada Sobre a Galeria de Produtos.

```
fx Search(Filter(Produtos;ParentItem = Var1); TextSearchBox1_1.Text; "NumPedido")
```

Fonte: Elaborado pelo autor.

os botões de voltar, recarregar a página e adicionar um novo produto na carga, de acordo com a Figura 31. As funções para executar cada tarefa já foram mencionadas na subseção Lista de Cargas. Porém agora, todas as funções são aplicadas sobre a *BrowseGallery2*, a galeria que representa fonte de dados dos produtos.

Figura 31 – Tela de Produtos da Carga.



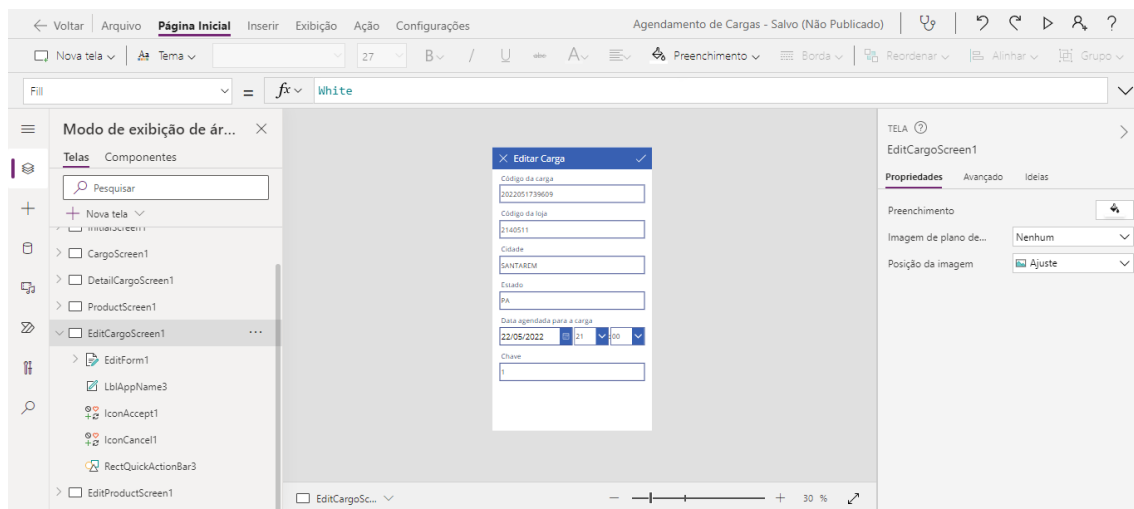
Fonte: Elaborado pelo autor.

3.5.2.5 Editar Carga e Produto

As últimas telas são de edição das cargas e produtos que são muito parecidas, pois ambas tratam-se de um formulário que apresenta algumas informações das duas fontes de dados utilizadas nesse projeto, de acordo com as Figuras 32 e 33.

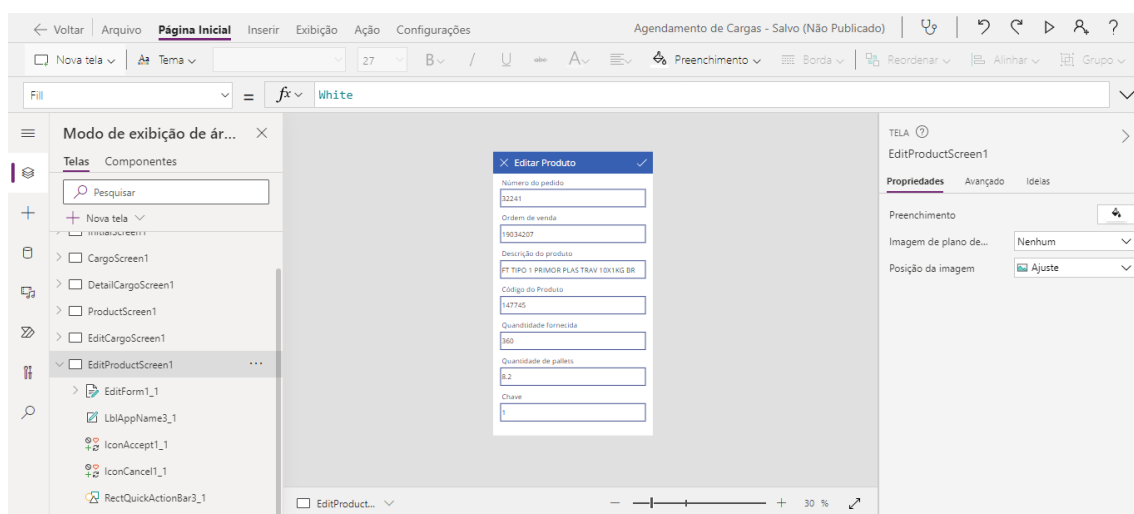
A tela Edição de Cargas apresentada na Figura 32 mostra um formulário que contém as colunas da planilha e traz as informações deste item, como: os códigos da carga e da loja, cidade, estado, data de agendamento e elemento identificador. Sendo assim, se for necessário alterar alguma dessas informações, pode-se corrigir os dados nos campos e pressionar o botão de enviar. Fazendo isso, o *Power Apps* localiza a carga na tabela da fonte de dados e modifica as colunas que foram alteradas. A tela Edição de Produtos apresentada Figura 33 funciona da mesma forma, porém ela mostra as informações dos produtos e não das cargas.

Figura 32 – Tela de Edição de Cargas.



Fonte: Elaborado pelo autor.

Figura 33 – Tela de Edição de Produtos.



Fonte: Elaborado pelo autor.

Essas duas telas de edição também podem servir para inserção de novas cargas e novos produtos quando o usuário do aplicativo pressiona o botão adicionar, que tem como função *NewForm*. Sendo assim, os campos que o aplicativo mostra são os mesmos das Figuras 32 e 33, porém eles aparecem sem nenhuma informação, com exceção do elemento identificador comum (mostrado através dos campos ‘Chave’ no *app*). Este elemento foi programado para aparecer automaticamente pois ele deve ser único para cada carga e comum somente para os produtos da mesma carga. Sendo assim, não é correto deixar o usuário definir o valor do elemento pois ele pode determinar um valor já existente, o que provocaria um erro duplicidade nas cargas e produtos de todo o aplicativo de consultas.

4 RESULTADOS

Este capítulo apresenta o resultado da solução desenvolvida para a automação do agendamento de cargas dentro do portal Atacadão. Além disso, é apresentado também a análise comparativa entre o desempenho do RPA e a execução da tarefa de forma manual feita por um operador humano. Por fim, apresenta também o resultado do aplicativo de consultas.

4.1 TEMPO DE EXECUÇÃO

Para comparar o tempo médio de execução da atividade de agendamento de cargas dentro do Portal Atacadão realizada por um operador humano, com o tempo de execução da mesma tarefa feita através RPA, foram feitas algumas avaliações diretamente com o time de CS responsável pela atividade. Essas avaliações trouxeram valores um pouco variáveis dependendo do operador que realiza a atividade e também da quantidade de pedidos e produtos que formam cada carga. Contudo, ainda assim foi possível calcular o tempo médio de aproximadamente 3 minutos para fazer a atividade por completo em cada carga agendada.

Este valor contabiliza o tempo de abrir a planilha de agendamentos e tratá-la para que fique organizada de acordo com as cargas para cada loja. É contabilizado também todo o processo de agendamento dentro do portal, que incluem atividades como: verificação dos produtos de cada pedido, adição de pedidos e agendamento das cargas. Por fim, é também contabilizado o tempo de execução do processo de atualização de status no SAP e o tempo de execução do processo de atualização da planilha de agendamentos.

O tempo médio da execução das mesmas tarefas realizadas pelo RPA é 57 segundos, de acordo com a coluna *Total Work Time* da Figura 34. Aproximando o valor para 1 minuto, chega-se a uma redução de aproximadamente 66,6% no tempo de execução da mesma atividade realizada através do RPA.

Figura 34 – Fila Pai - Cargas.

Item Key	Pr	Status	Tag	Re.	At	Created	Last Updated	Completed	Total Work Time	Exception Reason
1059290 - 17/06/2022	0	Business Exception	Business Exception:Exception c	HBB	1	14/06/2022 173552	14/06/2022 173631		0038	14 Exception children: The button to schedule the request is not available
1671532 - 20/06/2022	0	Completed	Código da carga: 2022061409628	HBB	1	14/06/2022 094721	14/06/2022 094842	14/06/2022 09484	0119	
1953041 - 08/06/2022	0	System Exception	Exception:Exception children: failed to pe	HBB	1	06/06/2022 095152	06/06/2022 095236		0044	06 Exception children: failed to perform step 1 in Navigate Stage 'Select To Define' on page 'Request Schem
2131811 - 09/06/2022	0	Completed	Código da carga: 2022060401356	HBB	1	04/06/2022 084932	04/06/2022 090537	04/06/2022 09055	0056	
2121621 - 08/06/2022	0	Completed	Código da carga: 2022060401313	HBB	1	04/06/2022 084932	04/06/2022 090501	04/06/2022 09050	0055	
2121235 - 10/06/2022	0	Completed	Código da carga: 2022060401275	HBB	1	04/06/2022 084932	04/06/2022 090406	04/06/2022 09040	0057	
2035152 - 10/06/2022	0	Completed	Código da carga: 2022060401267	HBB	1	04/06/2022 084932	04/06/2022 090309	04/06/2022 09030	0055	
2023320 - 09/06/2022	0	Completed	Código da carga: 2022060401232	HBB	1	04/06/2022 084932	04/06/2022 090214	04/06/2022 09021	0056	
1848022 - 08/06/2022	0	Completed	Código da carga: 2022060401216	HBB	1	04/06/2022 084932	04/06/2022 090118	04/06/2022 09011	0118	
1848022 - 10/06/2022	0	System Exception	Exception:Exception children: It was not p	HBB	1	04/06/2022 084932	04/06/2022 090000		0046	04 Exception children: It was not possible to schedule the cargo.
1671532 - 10/06/2022	0	Business Exception	Business Exception:Exception:Exception c	HBB	1	04/06/2022 084932	04/06/2022 085914		0030	04 Exception children: The button to schedule the request is not available
1059299 - 09/06/2022	0	Completed	Código da carga: 2022060401127	HBB	1	04/06/2022 084932	04/06/2022 085844	04/06/2022 08584	0055	
1059299 - 08/06/2022	0	Completed	Código da carga: 2022060401089	HBB	1	04/06/2022 084932	04/06/2022 085749	04/06/2022 08574	0058	
1059291 - 10/06/2022	0	Completed	Código da carga: 2022060401046	HBB	1	04/06/2022 084932	04/06/2022 085551	04/06/2022 08555	0059	
1059284 - 09/06/2022	0	Completed	Código da carga: 2022060401003	HBB	1	04/06/2022 084932	04/06/2022 085552	04/06/2022 08555	0056	
1059275 - 08/06/2022	0	Completed	Código da carga: 2022060400961	HBB	1	04/06/2022 084932	04/06/2022 085456	04/06/2022 08545	0139	
1059266 - 09/06/2022	0	Completed	Código da carga: 2022060400937	HBB	1	04/06/2022 084932	04/06/2022 085317	04/06/2022 08531	0057	
1059258 - 13/06/2022	0	Completed	Código da carga: 2022060400929	HBB	1	04/06/2022 084932	04/06/2022 085220	04/06/2022 08522	0055	
1059258 - 10/06/2022	0	Completed	Código da carga: 2022060400910	HBB	1	04/06/2022 084932	04/06/2022 085125	04/06/2022 08512	0037	
1059236 - 09/06/2022	0	Completed	Código da carga: 2022060400899	HBB	1	04/06/2022 084932	04/06/2022 085028	04/06/2022 08502	0056	

Fonte: Elaborado pelo autor.

4.2 NÍVEL DE EXCEÇÃO

A confiabilidade da automação do projeto é um dos pontos principais para que o processo seja autorizado a entrar em execução. Portanto, a confiabilidade deste projeto diz respeito à segurança das informações processadas pelo robô, além do processamento correto e eliminação de erros.

A automação confere um nível adicional de confiabilidade pois (i) o robô coleta todos os dados necessários para o processamento das cargas diretamente da planilha de agendamentos da própria área de CS sem alterá-los e (ii) detecta possíveis erros nestes dados e faz a indicação deles através das exceções, conforme a Figura 34, evitando propagação de erros.

Dentre as últimas execuções houveram apenas 2 exceções de negócio e duas exceções sistêmicas, sendo assim, apenas 4 cargas não foram agendadas, conforme a Figura 34. As exceções de negócios remetem a cargas que já foram agendadas por algum operador humano e portanto, não é necessário processar esta carga novamente. Por outro lado, as exceções sistêmicas ocorreram durante a interação entre o robô e o portal, sendo uma falha da automação que acontece em algumas situações e portanto, devem ser corrigidas. Contudo, a área de CS considera esses números satisfatórios de acordo com os *feedbacks* feitos após a entrega do projeto. Isso porque, o número de exceções sistêmicas da amostra colhida (20 cargas) é de somente 10%, conforme a Figura 34.

4.3 REDUÇÃO DE CUSTOS

Outro indicador de resultado muito importante é a redução de custos que a implementação da automação do agendamento de cargas traz. O cálculo é baseado no FTE (Full-Time Equivalent), que é a métrica utilizada para mensurar a necessidade de recursos humanos para realizar uma tarefa (SARI; HARDIANSÁ; SURYOPUTRO, 2018).

O cálculo do FTE é baseado no volume anual de cargas (V), tempo de execução por carga (T) e jornada de trabalho anual (J). Estes valores foram cedidos pela área de CS para o desenvolvimento do projeto, sendo: $V = 9483$, $T = 3$ min e $J = 2080$ horas.

$$FTE = V \cdot \frac{T}{60} \quad (1)$$

$$FTE = 9483 \cdot \frac{3}{2080} \quad (2)$$

$$FTE = 0,23 \quad (3)$$

Considerando que 1 FTE equivale em média às 2080 horas anuais de trabalho de um colaborador, a implementação da automação de agendamento de cargas com RPA evita um custo de aproximadamente 478 horas anuais da carga de trabalho deste mesmo

colaborador. Ou seja, a automação do agendamento de cargas elimina 23% da carga horária de um colaborador. O valor monetário não pode ser divulgado pois é uma informação sigilosa da empresa e portanto não pode ser exposta. Contudo, utilizando a automação se tem uma redução de aproximadamente 40 horas mensais do funcionário, que é uma redução significativa. Sendo assim, esse tempo pode ser utilizado para outra atividade lucrativa dentro da empresa.

4.4 APLICATIVO DE CONSULTAS

O aplicativo de consulta das cargas processadas pela automação foi desenvolvido para detalhar e consultar com mais facilidade todas as cargas, pedidos e produtos que o robô de agendamentos Atacadão processou. Sendo essa uma forma mais rápida e eficaz para visualizar os agendamentos sem ter que consultar a planilha, que é menos intuitiva.

O aplicativo foi desenvolvido para apresentar 5 telas, sendo elas: tela inicial, lista das cargas, lista dos produtos, detalhe da carga, editar carga e editar produto. A tela inicial mostrada na Figura 35, contém apenas um botão para redirecionar o usuário para a lista das cargas.

Figura 35 – Tela Inicial.



Fonte: Elaborado pelo autor.

As telas lista de cargas e lista de produtos da Figura 36 mostram as informações das fontes de dados das duas planilhas que o robô produz no final do processamento. A imagem da esquerda exibe as informações principais de cada carga que foi agendada, como por exemplo: códigos da carga, número da loja em que foi agendado, cidade e data de agendamento. A imagem da direita indica uma lista com todos os produtos que foram inseridos em uma específica carga, além de revelar alguns dados de cada produto, como:

descrição e código do produto, quantidade fornecida e quantidade de *pallets*, número da ordem de venda e número do pedido.

Essas informações são muito importantes para o controle e gerenciamento de todo o agendamento de cargas por parte da área de CS. Além disso, com essas informações é possível fazer um levantamento da quantidade de cargas que foram agendadas para determinadas lojas, os produtos que foram mais solicitados, etc. Sendo assim, é possível apresentar indicadores para a empresa Bunge da quantidade de produtos que saem do estoque e os produtos dos quais se deve aumentar ou diminuir a produção.

Figura 36 – Telas da Lista de Cargas e Produtos.



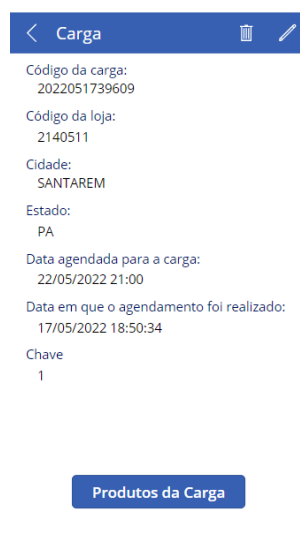
Fonte: Elaborado pelo autor.

Além disso, na Figura 36 nota-se os ícones dos botões de inclusão, exclusão e edição das cargas e produtos para eventuais correções. Essas funções são úteis principalmente quando a área de CS agenda ou corrige cargas de forma manual no portal Atacadão, sendo possível fazer a modificação diretamente no aplicativo.

Ao selecionar uma carga na tela lista de cargas, o usuário é redirecionado para os detalhes da carga, mostrada na Figura 37. Essa tela exhibe todas as informações da carga, inclusive a data e hora em que o RPA fez o agendamento no portal. Nessa tela, também estão disponíveis as funções de editar e excluir para possíveis ajustes. Além disso, pode-se pressionar o botão no canto inferior para exibir a tela da lista de produtos que já foi anunciada anteriormente.

As duas últimas telas do aplicativo, ilustradas na Figura 38, são utilizadas para edição de cargas e produtos. Estão disponíveis e editáveis todos campos em ambas as telas para fazer quaisquer correções necessárias. As duas telas também são utilizadas para adicionar novas cargas e novos produtos. Contudo, nessa situação, todos os campos da Figura 38 aparecem vazios, para a inserção dos dados e informações, com exceção dos

Figura 37 – Tela de Detalhes da Carga.



Detalhes de uma carga em uma interface de usuário. O cabeçalho contém um ícone de seta para voltar, o título 'Carga', um ícone de lixeira e um ícone de lápis. O conteúdo principal apresenta as seguintes informações:

- Código da carga: 2022051739609
- Código da loja: 2140511
- Cidade: SANTAREM
- Estado: PA
- Data agendada para a carga: 22/05/2022 21:00
- Data em que o agendamento foi realizado: 17/05/2022 18:50:34
- Chave: 1

Abaixo das informações, há um botão azul com o texto 'Produtos da Carga'.

Fonte: Elaborado pelo autor.

campos 'Chave', que são os elementos identificadores de cada item.

Figura 38 – Telas de Edição de Cargas e Produtos.



Dois formulários de edição lado a lado. O formulário à esquerda, intitulado 'Editar Carga', contém os seguintes campos:

- Código da carga: 2022051739609
- Código da loja: 2140511
- Cidade: SANTAREM
- Estado: PA
- Data agendada para a carga: 22/05/2022 21:00
- Chave: 1

O formulário à direita, intitulado 'Editar Produto', contém os seguintes campos:

- Número do pedido: 32241
- Ordem de venda: 19034207
- Descrição do produto: FT TIPO 1 PRIMOR PLAS TRAV 10X1KG BR
- Código do Produto: 147745
- Quantidade fornecida: 360
- Quantidade de pallets: 8.2
- Chave: 1

Fonte: Elaborado pelo autor.

5 CONCLUSÃO

Este trabalho apresentou o desenvolvimento e resultados de uma automação utilizando RPA para solucionar o problema do agendamento de cargas das lojas Atacadão, clientes da empresa Bunge. O robô de RPA desenvolvido através do *software* Blue Prism teve como objetivo automatizar por completo o agendamento de cargas, que é uma das etapas que fazem parte da logística, dentro da cadeia de suprimentos.

O robô desenvolvido interage com três aplicações pra fazer a atividade de agendamento por completo, sendo: Microsoft Excel, SAP e Google Chrome. No início do processamento da automação, o robô trabalha na planilha de agendamentos fornecida pela área de CS, que contém todas as informações de agendamento dos produtos e pedidos. Ele manipula a planilha de modo a organizar os pedidos em formato de cargas que serão direcionadas para as lojas Atacadão. Além disso, o robô é capaz de entrar no portal Atacadão através do Google Chrome e fazer o agendamento de todas as cargas de forma rápida, minimizando todos os possíveis erros de negócios. Por fim, ele ainda acessa o sistema SAP para marcar os pedidos processados por completo e gera relatórios com os itens que foram executados com sucesso.

O resultado da automação com RPA foi satisfatório, pois ela pode substituir o trabalho manual e repetitivo de um operador humano, fazendo toda a atividade do mesmo modo que o funcionário faria. A automação traz vários benefícios, como segurança e confiabilidade das informações, velocidade e eficiência. Além disso, os resultados deste trabalho são compatíveis com a proposta do projeto, visto que houve uma redução de aproximadamente 66,6% no tempo de execução da atividade quando ela foi automatizada através do RPA. Ademais, o nível de exceções durante a execução do robô é extremamente baixo para um sistema automatizado, tornando-o eficiente. Por fim, houve também uma redução de custos com a implementação do RPA para realizar o processo de agendamento de cargas, visto que a automação substitui uma jornada de 40 horas mensais de um operador humano.

Além disso, foi desenvolvido também um aplicativo com Power Apps para consulta e ajuste das cargas que foram agendadas através da automação com RPA. O aplicativo é uma forma rápida e intuitiva da área de CS fazer as consultas e checar como está o andamento dos agendamentos de carga feito através da automação. Além disso, o aplicativo também mostra alguns indicadores de resultados para a empresa Bunge, como os produtos mais solicitados e a quantidade de produtos que saem do estoque.

Para trabalhos futuros pretende-se fazer melhorias contínuas na automação do agendamento de cargas, além de incluir mais funcionalidades tanto no processo em RPA quanto no aplicativo. Uma segunda alternativa seria fazer o levantamento dos indicadores de números e quantidades de produtos com base nos pedidos agendados, além de fazer as análises gráficas e quantitativas para apontar de forma automática os produtos que mais

tem movimentações no estoque. Dessa forma, esse projeto apresentaria os indicadores de resultados como também atenderia a demanda de um sistema de alerta de estoque para a empresa Bunge.

REFERÊNCIAS

ACOBA, Francisco; LEVINE, Abby; TOUSAIN, Alina; KAPLAN, David. Corporate Real Estate and Facilities Management in the digital world Part Two: Robotic process automation benefits and adoption journey. **Deloitte.**, p. 2, 2019.

BALLOU, Ronald H. **Gerenciamento da Cadeia de Suprimentos-: Logística Empresarial.** 5. ed. [S.l.]: Bookman editora, 2009. P. 28.

BUNGE. **Locations.** 2022. Disponível em:
<https://www.bunge.com/who-we-are/locations>. Acesso em: 4 jul. 2022.

CHAPPELL, David. Introducing Blue Prism Robotic Process Automation for the Enterprise. **Chappell & Associates**, p. 12, 2017.

COPEES, Mariano; MELLO, Caio Serra. RPA (ROBOTIC PROCESS AUTOMATION): CRITÉRIOS PARA ESCOLHA DE FERRAMENTAS E PROCESSOS. **Universidade Federal de Fluminense**, p. 23, 2020.

DELANO, Lance. **Visão geral de conectores para aplicativos de tela.** 2022. Disponível em:
<https://docs.microsoft.com/pt-br/power-apps/maker/canvas-apps/connections-list>. Acesso em: 5 jul. 2022.

ESALES. **Agendamento de cargas e descargas: saiba como fazer esse controle.** 2021. Disponível em: <https://esales.com.br/blog/agendamento-de-cargas/>. Acesso em: 31 mai. 2022.

FANTINA, Robert; STOROZHUK, Andriy; GOYAL, Kamal. **Introducing Robotic Process Automation to Your Organization: A Guide for Business Leaders.** 1. ed. [S.l.]: Apress, 2021. ISBN 1484274156,9781484274156.

GOMES, Carlos Francisco Simões; RIBEIRO, Priscilla Cristina Cabral. **Gestão da cadeia de suprimentos integrada à tecnologia da informação.** [S.l.]: Editora Senac Rio, 2020.

HAIJAR, Alamira Jouman. **RPA Market Size and Popular Vendors in 2022.** 2021. Disponível em: <https://research.aimultiple.com/rpa-market/>. Acesso em: 19 mai. 2022.

- MADAKAM, Somayya; HOLMUKHE, Rajesh M.; JAISWAL, Durgesh Kumar. THE FUTURE DIGITAL WORK FORCE: ROBOTIC PROCESS AUTOMATION (RPA). **Journal of Information Systems and Technology Management**, p. 5, 2019.
- ORGANISATION, International Labour. **Ergonomic Checkpoints: Practical and Easy-to-Implement Solutions for Improving Safety, Health and Working Conditions**. second edition. [S.l.]: International Labour Office, 2010. ISBN 9221226662,9789221226666.
- OSTDICK, Nick. **Looking Forward, Looking Back: Five Key Moments in The History of RPA**. 2016. Disponível em: <https://www.uipath.com/blog/rpa/looking-forward-looking-back-five-key-moments-in-the-history-of-rpa>. Acesso em: 24 mai. 2022.
- SAP. **O que é a SAP?** 2022. Disponível em: <https://www.sap.com/brazil/about/company/what-is-sap.html>. Acesso em: 1 jun. 2022.
- SARI, Amarria Dila; HARDIANSANA, Fajri; SURYOPUTRO, Muhamad Ragil. Workload assessment on foundry SME to enhance productivity using full time equivalent. **MATEC Web Conf.**, v. 154, p. 01081, 2018.
- SCHULER, Juerg; GEHRING, Florian. Implementing Robust and Low-Maintenance Robotic Process Automation (RPA) Solutions in Large Organisations. **Social Science Research Network**, p. 29, 2018.
- SHEHAB, E.M; SHARP, M.W; SUPRAMANIAM, L.; SPEDDING, Trevor A. Enterprise resource planning: An integrative review. **Business process management journal**, Emerald Group Publishing Limited, 2004.
- SHVETS, Anna; SHVETS, Valentyna. Cognitive and technological aspects of e-learning in context of robotization. v. 3, p. 65, jun. 2018.
- SILVA, Arthur Marcos da; BARION, Michele Cristiani. Automação Robótica de Processos (RPA): Estudo de Caso Através da Tarefa Administrativa Contas a Pagar. **Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP)**, p. 2, 2018.
- SYED, Rehan *et al.* Robotic Process Automation: Contemporary themes and challenges. **Computers in Industry**, v. 115, p. 103162, 2020. ISSN 0166-3615.

TRACKAGE, Blog da. **Agendamento de carga e descarga**. 2022. Disponível em: <https://trackage.com.br/blog/agendamento-de-carga-e-descarga/>. Acesso em: 31 mai. 2022.

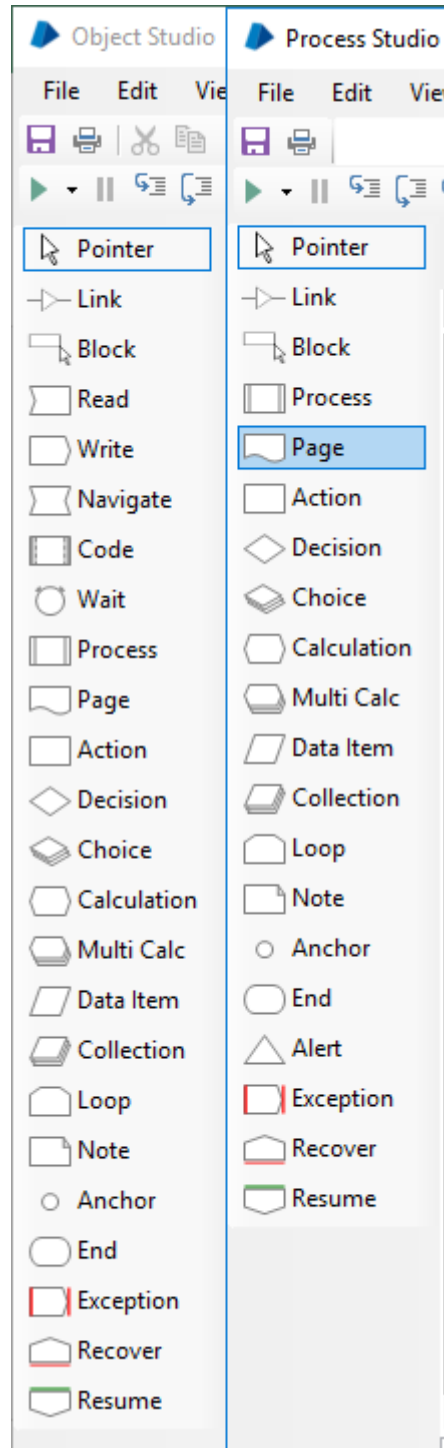
UIPATH. **What is Robotic Process Automation - RPA Software**. 2005. Disponível em: <https://www.uipath.com/rpa/robotic-process-automation>. Acesso em: 18 mai. 2022.

VIVEK, Kumar. **What is Power Apps?** 2022. Disponível em: <https://docs.microsoft.com/en-us/power-apps/powerapps-overview>. Acesso em: 20 jul. 2022.

WATERS, Donald. **Logistics: An Introduction to Supply Chain Management**. [S.l.]: Palgrave Macmillan, 2003. ISBN 9780333963692,0333963695.

ANEXO A – Estágios *Blue Prism*

Figura 39 – Estágios do *Object Studio* e *Process Studio*.



Fonte: Elaborado pelo autor.