



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

Isadora Gonçalves de Freitas Vizoli

Automação da Geração de Informações para Pagamento de FGTS

Blumenau
2023

Isadora Gonçalves de Freitas Vizoli

Automação da Geração de Informações para Pagamento de FGTS

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. Dr. Maiquel de Brito

Blumenau

2023

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

Vizoli, Isadora Gonçalves de Freitas
Automação da Geração de Informações para Pagamento de FGTS
/ Isadora Gonçalves de Freitas Vizoli ; orientador,
Maiquel de Brito, 2023.
59 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus Blumenau,
Graduação em Engenharia de Controle e Automação, Blumenau,
2023.

Inclui referências.

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

Isadora Gonçalves de Freitas Vizoli

Automação da Geração de Informações para Pagamento de FGTS

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, 3 de Julho de 2023.

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 à minha mãe, à minha avó, ao
meu marido e à mim.

AGRADECIMENTOS

Gostaria de agradecer aos meus pais, por me apoiarem e por me mostrarem que a educação é importante e nos molda como seres humanos. Agradeço, especialmente, à minha mãe, Eliene, que me deu todo o suporte para enfrentar a jornada de fazer uma graduação em outro Estado. E, também, por no decorrer dos anos, sempre estar ao meu lado me incentivando e me mostrando que eu era capaz de realizar meus sonhos. Agradeço aos meus irmãos e à minha avó por sempre me apoiarem e por me darem bons conselhos. Agradeço ao meu marido, Daniel, por escolher enfrentar comigo essa jornada, abrindo mão de uma vida em São Paulo, para começar outra completamente diferente em Blumenau. Depois de anos de muita luta e aprendizado, conseguimos concluir mais uma etapa importante da nossa vida.

Aos meus amigos de São Paulo e Blumenau que sempre me escutaram e me apoiaram quando precisei. E, também, à minha companheira Lika, que renovava toda a minha energia quando esta parecia se esgotar.

Aos meus professores da Universidade Federal de Santa Catarina por todo ensino compartilhado, em especial, ao meu orientador Maiquel, por sua paciência e orientações na construção deste trabalho.

À Bunge e a todos os colegas com os quais tive contato, por toda ajuda e apoio no dia a dia. E, por fim, agradeço ao meu coordenador, Bruno, pela oportunidade de iniciar minha vida profissional em sua talentosa equipe e pela confiança depositada em mim desde então.

A persistência é o menor caminho do êxito. (Chaplin)

RESUMO

Nos dias atuais, as empresas possuem um grande número de processos administrativos para serem realizados em seu dia a dia. Dentre os diferentes tipos de processos que uma empresa possui, está o pagamento de guias de rescisão relativos aos depósitos do FGTS (Fundo de Garantia de Tempo de Serviço), também conhecidas como guias GRRF. Para realizar este pagamento, é necessário manipular dados provenientes de diferentes arquivos digitais. Este processo é repetitivo e acaba sobrecarregando o responsável pela tarefa quando seu volume é grande. Neste contexto, o presente trabalho apresenta duas soluções para sanar esse problema: a Automação Robótica de Processos (RPA), responsável por substituir o trabalho humano por digital; e as Expressões Regulares (Regex), que em computação, são utilizadas para encontrar dados específicos dentro de um texto. Utilizando-se as duas soluções em conjunto, foi possível reduzir o tempo de execução da tarefa em 50% e a carga de trabalho anual do operador humano em 25%.

Palavras-chave: Guias GRRF; RPA; Regex.

ABSTRACT

Nowadays, companies have a large number of administrative processes to go through on a daily basis. Among the different types of processes that a company has is the payment of termination slips for FGTS deposits, also known as GRRF slips. To make this payment, it is necessary to manipulate data from different digital files. This is a repetitive process and, when the volume is high, it overloads the person responsible for the task. In this context, the present work presents two solutions to solve this problem: the Robotic Process Automation (RPA), responsible for replacing human work by digital work; and the Regular Expressions Regex, which in computing, are used to find specific data within a text. By using both solutions together, it was possible to reduce the task execution time by 50% and the annual workload of the human operator by 25%.

Keywords: GRRF Guides; RPA; Regex.

LISTA DE FIGURAS

Figura 1 – Interface gráfica do Process Studio	19
Figura 2 – Exemplo de um Processo	20
Figura 3 – Exemplo de um Objeto	21
Figura 4 – Estágios adicionais do Object Studio	22
Figura 5 – Páginas padrão do Object Studio	22
Figura 6 – Funcionamento do Application Modeller	23
Figura 7 – Itens de uma fila no Control Room	24
Figura 8 – Metacaractere Ponto	27
Figura 9 – Metacaractere Lista	27
Figura 10 – Metacaractere Lista Negada	27
Figura 11 – Metacaractere Opcional	28
Figura 12 – Metacaractere Asterisco	28
Figura 13 – Metacaractere Mais	29
Figura 14 – Metacaractere Chaves	29
Figura 15 – Metacaractere Circunflexo	29
Figura 16 – Metacaractere Cifrão	30
Figura 17 – Metacaractere Borda	30
Figura 18 – Metacaractere Ou	31
Figura 19 – Metacaractere Grupo	31
Figura 20 – Metacaractere Retrovisor	31
Figura 21 – <i>Action Extract Regex Values</i>	33
Figura 22 – Resultado da regex	34
Figura 23 – Corpo do e-mail	35
Figura 24 – Login no site ADP	36
Figura 25 – Página inicial da ADP	36
Figura 26 – Página Requisições do Usuário	37
Figura 27 – Aba de Arquivos	38
Figura 28 – Página Dados da Requisição	38
Figura 29 – Exemplo de guia GRRF	40
Figura 30 – Exemplo de espelho da guia GRRF	40
Figura 31 – Dados populados na planilha	41
Figura 32 – Diagrama do projeto GRRF's Guides	41
Figura 33 – Regra criada no Outlook	42
Figura 34 – Diagrama da <i>Main Page</i> do processo	44
Figura 35 – Página Populate Queue	45
Figura 36 – Diagrama da página Get Mails	45
Figura 37 – Resultado da Regex REFile	46

Figura 38 – Fluxo para encontrar o número de PIS	47
Figura 39 – Diagrama da página Extract Data from PDF	49
Figura 40 – Diagrama da página Extract Data from PDF	51
Figura 41 – Modo Recovery para Exceptions	52
Figura 42 – Diagrama da página Send Mail	52
Figura 43 – Corpo do e-mail para exceção do PIS	53
Figura 44 – Dashboard do robô	54
Figura 45 – Fila do robô	55

LISTA DE TABELAS

Tabela 1 – Representação dos Metaracteres.	26
Tabela 2 – Exemplo de <i>Named Capturing Groups</i> em duas linguagens de programação.	32

LISTA DE ABREVIATURAS E SIGLAS

ADP	Automatic Data Processing
BP	Blue Prism
BPMN	Notação de Modelagem de Processos de Negócios
CNPJ	Cadastro Nacional da Pessoa Jurídica
FGTS	Fundo de Garantia de Tempo de Serviço
FTE	Full-Time Equivalent
GRRF	Guias de Recolhimento Rescisório do Fundo de Garantia do Tempo de Serviço
PIS	Programa de Integração Social
Regex	Expressões Regulares
RPA	Automação Robótica de Processos
SAP	Systemanalysis Programmentwicklung

LISTA DE SÍMBOLOS

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

SUMÁRIO

1	INTRODUÇÃO	15
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	AUTOMAÇÃO ROBÓTICA DE PROCESSOS	17
2.2	BLUE PRISM	18
2.2.1	Process Studio	18
2.2.2	Object Studio	20
2.2.2.1	<i>Application Modeller</i>	21
2.2.3	Control Room	22
2.3	EXPRESSÕES REGULARES	24
2.3.1	Metacaracteres	26
2.3.1.1	<i>Representantes</i>	26
2.3.1.2	<i>Quantificadores</i>	28
2.3.1.3	<i>Âncoras</i>	29
2.3.1.4	<i>Outros</i>	30
2.3.2	Regex Engine	31
2.3.3	Expressões Regulares no Blue Prism	32
3	DESENVOLVIMENTO	35
3.1	EXECUÇÃO DO PROCESSO NÃO AUTOMATIZADO	35
3.2	AUTOMATIZAÇÃO DO PROCESSO COM BLUE PRISM	41
3.2.1	<i>Main Page</i> do Processo no Blue Prism	43
3.2.2	Página Get Mails	43
3.2.3	Página <i>Download Files from ADP Website</i>	44
3.2.4	Página Extract Data from PDF	46
3.2.4.1	<i>Loop para Manipulação dos Dados</i>	48
3.2.4.2	<i>Bloco Merge Files</i>	49
3.2.5	Tratamento de Exceções	50
4	RESULTADOS	54
4.1	NÍVEL DE EXCEÇÃO	54
4.2	TEMPO DE EXECUÇÃO	55
4.3	ANÁLISE DE CUSTOS	55
5	CONCLUSÃO	57
	REFERÊNCIAS	58

1 INTRODUÇÃO

As empresas, em geral, possuem inúmeros processos administrativos para tratar, que vão desde a gestão de estoques e fornecedores até a gestão de recursos humanos e financeira. Esses processos exigem que os funcionários realizem diversas tarefas e cumpram uma série de prazos, o que pode acabar sobrecarregando-os. Além disso, muitas vezes esses processos são monótonos, repetitivos e burocráticos, o que pode aumentar ainda mais o nível de estresse dos colaboradores e reduzir a eficiência do trabalho. Portanto, faz-se necessário que as empresas encontrem formas de simplificar esses processos, implementando ferramentas e sistemas que facilitem as tarefas administrativas e permitam que os funcionários possam se concentrar em outras atividades que demandam maior atenção. Uma ferramenta que simplifica esses processos é o RPA (Automação Robótica de Processos). O RPA é uma ferramenta de software capaz de imitar as ações que um ser humano executaria ao operar um sistema computacional para transferir informações entre diferentes sistemas de tecnologia. Por meio da automação robótica, o software atua como um operador digital para manipular aplicativos existentes, seguindo os mesmos passos que uma pessoa realizaria para completar uma tarefa (GARTNER, 2022).

Uma tarefa administrativa comum e considerada repetitiva é a extração de dados de ex-colaboradores para pagamentos de Guias de Recolhimento Rescisório do Fundo de Garantia do Tempo de Serviço (GRRF), ou simplesmente guias GRRF. A realização dessa tarefa inclui, além da extração de dados de documentos digitais, a separação dos mesmos documentos de acordo com alguns critérios previamente estabelecidos, todos os dias. Considerando que a tarefa possui um padrão bem definido, é repetitiva e que o volume dos documentos é, em geral, grande, nota-se que a mesma é elegível para ser automatizada através de um sistema de RPA.

Assim, o presente trabalho propõe o desenvolvimento de uma solução automatizada para a tarefa descrita anteriormente. Portanto, o objetivo geral é tornar a tarefa completamente automatizada com o uso de RPA, de forma que nenhum operador humano precise intervir. Para atingir o objetivo geral, define-se os seguintes objetivos específicos:

1. Estudar as ferramentas definidas para a solução da tarefa;
2. Realizar uma revisão de literatura acerca das ferramentas;
3. Extrair os dados dos documentos digitais;
4. Separar os documentos digitais pelo número de CNPJ (Cadastro Nacional da Pessoa Jurídica);
5. Inserir os dados extraídos em uma planilha eletrônica;

6. Entregar as informações para o departamento de Recursos Humanos.

Especificamente, o projeto descrito neste trabalho foi realizado como demanda da área de Recursos Humanos da empresa Bunge, localizada em Gaspar/SC. A Bunge é uma empresa com mais de 200 anos de existência e sua sede é localizada em St.Louis, Missouri (EUA). Com instalações em mais de 40 países, a Bunge é líder mundial no processamento de sementes oleaginosas e na produção e fornecimento de óleos e gorduras vegetais especiais (BUNGE, 2023).

Nos capítulos a seguir, são apresentados respectivamente: a fundamentação teórica na qual a solução do projeto foi baseada (Capítulo 2); o desenvolvimento do projeto, onde a solução utilizada é descrita, se encontra no Capítulo 3; no Capítulo 4 são apresentados os resultados obtidos; e o Capítulo 5 apresenta as conclusões do projeto. Ao final do trabalho, são informadas as referências bibliográficas utilizadas ao longo do texto.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os principais temas que embasam o desenvolvimento do presente trabalho. Estes temas são: Automação Robótica de Processos (RPA), apresentado na Seção 2.1; o software Blue Prism, apresentado na Seção 2.2; e Expressões Regulares, apresentadas na Seção 2.3.

2.1 AUTOMAÇÃO ROBÓTICA DE PROCESSOS

A Automação Robótica de Processos refere-se ao uso de um software para automatizar tarefas repetitivas e baseadas em regras, em processos empresariais e de TI, por meio de scripts que simulam a interação humana com a interface do usuário de uma aplicação. Os softwares mais utilizados no RPA são: *UiPath*, *Automation Anywhere*, *Blue Prism*, *Microsoft* e *NICE* (GARTNER, 2022).

A Automação Robótica de Processos pode ser usada para automatizar processos de negócios em várias áreas, como: finanças, recursos humanos, operações e atendimento ao cliente. Quando o usuário de um software de RPA automatiza um processo, por convenção, diz-se que o processo passa a ser feito por um *bot de software*, ou simplesmente, por um *robô*. Neste contexto, os robôs nada mais são que programas de computador que são criados para realizar tarefas específicas de forma automatizada, sem a necessidade de intervenção humana constante. Os robôs podem ser programados para simular ações humanas, como digitar textos e cliques em botões, ou para realizar tarefas de forma mais eficiente e precisa do que seria possível para uma pessoa (ANYWHERE, 2023).

Além disso, o RPA oferece uma vantagem significativa para as empresas, pois, é capaz de reduzir custos e aumentar a qualidade das tarefas. Outra vantagem do RPA é que sua implementação não requer grandes investimentos ou mudanças significativas nos sistemas e processos das empresas (KROLL, 2016).

Dessa forma, algumas vantagens de usar RPA incluem:

1. **Eficiência:** Os robôs de RPA podem executar tarefas repetitivas e demoradas com maior eficiência e precisão do que os humanos, o que ajuda a reduzir o tempo necessário para concluir as tarefas;
2. **Redução de custos:** Ao automatizar tarefas com RPA, as empresas reduzem a necessidade de mão de obra humana e, portanto, reduzem os custos associados;
3. **Escalabilidade:** Pode-se criar novas automações utilizando-se automações já existentes, sem a necessidade de contratar mais pessoas;

4. Qualidade: Como os robôs de RPA seguem um conjunto predefinido de regras, eles tendem a executar as tarefas com mais precisão do que os humanos, o que pode levar a uma melhoria na qualidade do trabalho.

Apesar de contar com significativas vantagens, o uso do RPA traz algumas desvantagens como: dependência de estruturas e interfaces estáveis; limitações na compreensão de conteúdo não estruturado; necessidade de monitoramento constante; e impacto nas interações humanas, uma vez que a automação de processos pode acarretar em redução de funcionários.

O software de RPA utilizado no presente trabalho é o *Blue Prism*. Este software é o mesmo utilizado para todas as automações de processos da empresa Bunge, onde a solução descrita neste trabalho foi realizada. O funcionamento do software é detalhado a seguir.

2.2 BLUE PRISM

O BP (Blue Prism) é um dos softwares mais populares e o pioneiro da indústria do RPA. Lançado em 2001, tem como cofundadores Alastair Bathgate e Dave Moss (TAULLI, 2020). O Blue Prism possibilita que as empresas desenvolvam robôs programáveis capazes de executar atividades rotineiras, como preenchimento de formulários, processamento de faturas, envio de e-mails e outras tarefas administrativas.

Outro ponto importante a ser mencionado é o centro de segurança de dados que a ferramenta possui, trazendo confiabilidade para seu produto, o que fez grandes companhias como *Siemens*, *Land Rover* e *eBay* aderirem ao software. Como forma de atestar sua confiabilidade, o BP foi a primeira empresa de RPA a receber o mais alto nível de Veracode Verified, plataforma que valida se o software de uma empresa é seguro (TAULLI, 2020).

O Blue Prism tem em sua arquitetura três elementos principais, a saber: *Process Studio*, *Object Studio* e *Control Room*.

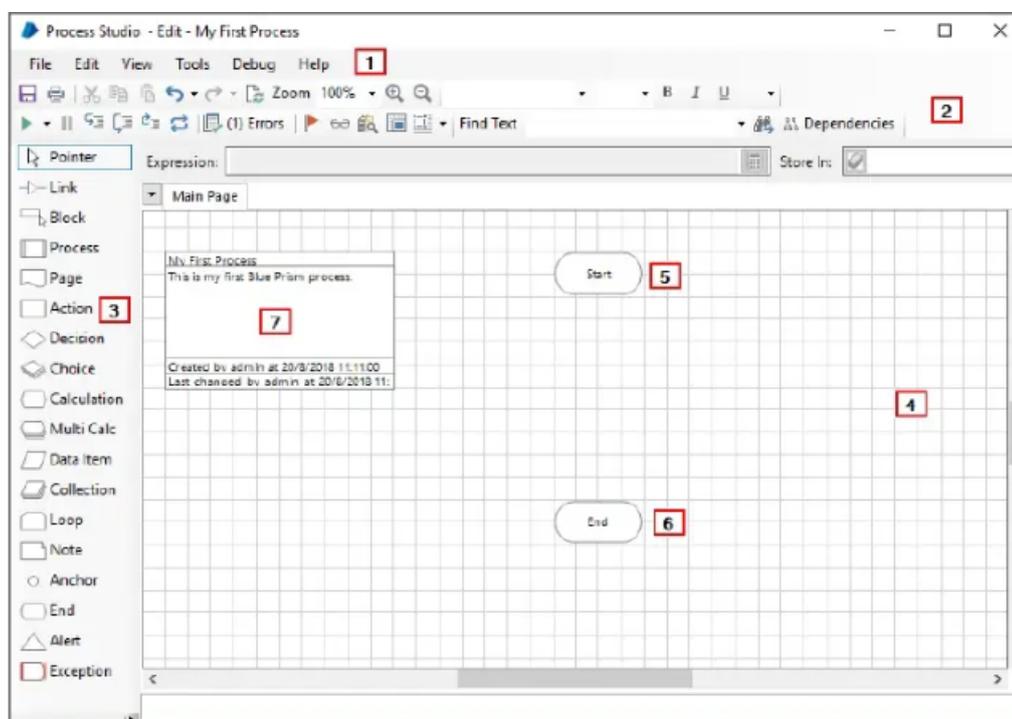
2.2.1 Process Studio

O *Process Studio* é uma interface do Blue Prism que proporciona ao usuário uma forma simples de se construir, editar e testar a lógica de uma tarefa (SIMPLILEARN, 2023). Esta lógica, também chamada de *processo* ou *robô*, é especificada através de um fluxograma. O BP tem como padrão iniciar a especificação de um processo por meio de uma página principal, ou *Main Page*, e a partir dela, fluxogramas podem ser criados de

acordo com uma sequência lógica. Essa interface oferece algumas funcionalidades para a construção do robô, como: chamadas de objetos, *loops*, variáveis, entre outros. A interface do *Process Studio* é ilustrada na Figura 1 e é composta pelos seguintes itens:

1. Menu: onde se encontram os comandos necessários para criar os processos;
2. Barra de ferramentas: onde se encontram os comandos mais utilizados para criar os processos;
3. Caixa de ferramentas: onde se encontram os estágios e funções para se criar a lógica dos processos;
4. Tela: onde o fluxograma do processo é criado;
5. Start: estágio que representa o início de um processo;
6. End: estágio que representa o final de um processo;
7. Bloco de informações: onde se encontram as principais informações do processo, como a descrição do processo, data de criação, nome do criador, entre outros.

Figura 1 – Interface gráfica do Process Studio

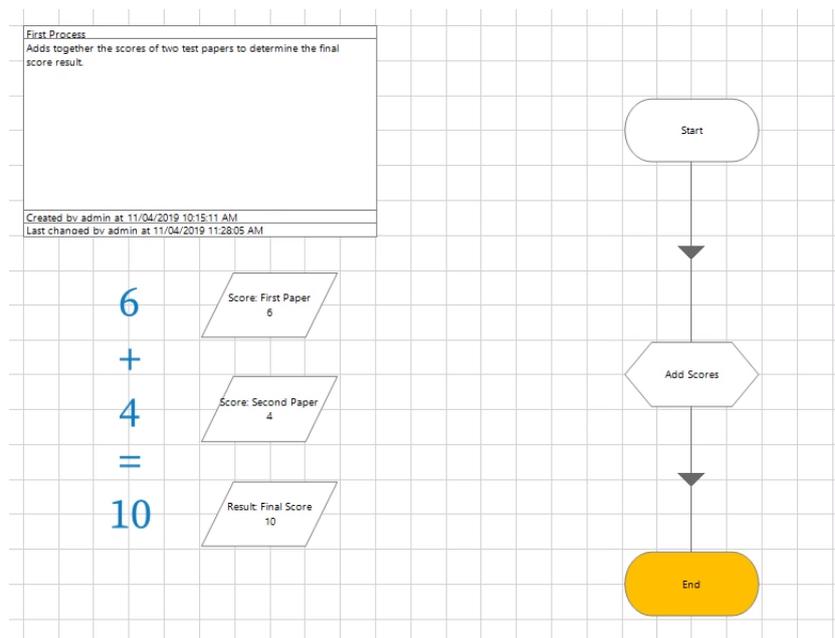


Fonte: Ying (2018)

Na Figura 2 é apresentado um exemplo prático de um processo no *Blue Prism*. À esquerda da figura estão os estágios chamados *data items* que são utilizados para armazenar informações de entrada e saída. À direita da figura está o fluxo do processo. Todo fluxo

começa com o estágio *start* e termina com o estágio *end*. Entre estes dois estágios, está o *calculation*, estágio que utiliza expressões para retornar valores, sejam eles em formato de texto, datas ou somente numéricos. Neste processo, o *calculation* possui uma expressão matemática de soma. A soma é realizada utilizando-se os valores armazenados nos *data items*, que são 6 e 4, e uma vez que o fluxo passa pelo estágio, o resultado da soma é armazenado no *data item* chamado *Result Final Score* e o processo é finalizado.

Figura 2 – Exemplo de um Processo



Fonte: DOCS (2023a)

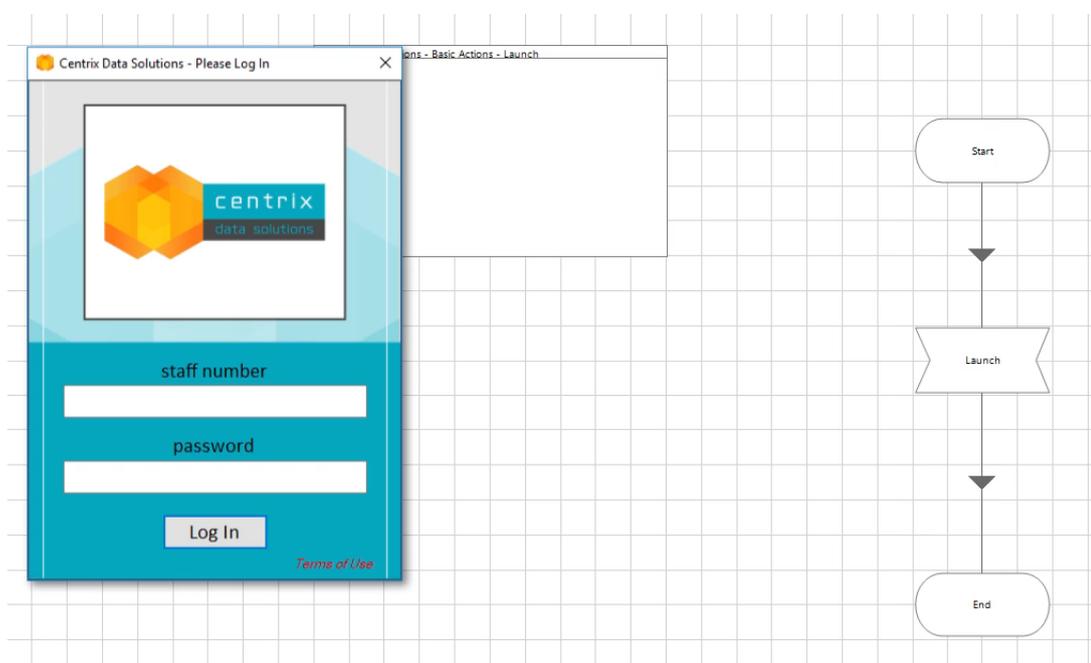
2.2.2 Object Studio

O *Object Studio* também é uma interface do Blue Prism, e é utilizada para a construção de objetos de negócio. Os objetos de negócio oferecem os meios para interagir com os sistemas externos presentes no diagrama de fluxo de processo especificado no *Process Studio* (DOCS, 2023b).

Assim como os processos, os objetos também são criados através de fluxogramas, com a diferença de alguns componentes em sua caixa de ferramentas. Estes objetos são criados com o objetivo de interagir com diferentes aplicações. Essas interações podem ser a inicialização e finalização de uma aplicação, escrita de dados, captura de dados, entre outros. Uma vez criado, o objeto pode ser reaproveitado para diferentes processos.

Objetos nunca são utilizados por conta própria. Eles são construídos como parte da lógica de um processo. Ou seja, a partir de um processo criado, objetos são utilizados para interagir de diferentes formas com aplicações. A Figura 3 exemplifica uma das possíveis interações, em que um objeto foi criado para abrir uma aplicação chamada “Centrix”. Assim como na especificação de processos, o fluxo do objeto começa com o estágio *start* e termina com o estágio *end*. Após o *start*, tem-se um estágio do tipo *navigate* (representado na Figura 3 pelo elemento com a legenda *Launch*) utilizado para abrir e fechar aplicações. Dessa forma, quando o processo chamar o objeto da Figura 3, a aplicação *Centrix Data Solutions* será aberta e o fluxo do objeto será finalizado.

Figura 3 – Exemplo de um Objeto



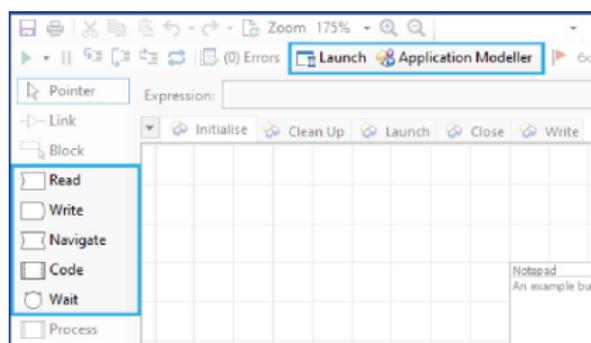
Fonte: DOCS (2023a)

A interface do *Object Studio* é similar à interface do *Process Studio*. Sua principal diferença é a adição dos itens *Read*, *Write*, *Navigate*, *Code* e *Wait*, das páginas *Initialise* e *Clean Up* que são criadas por padrão quando um novo objeto é desenvolvido, e dos comandos *Launch* e *Application Modeller*. As Figuras 4 e 5, ilustram as diferenças citadas.

2.2.2.1 Application Modeller

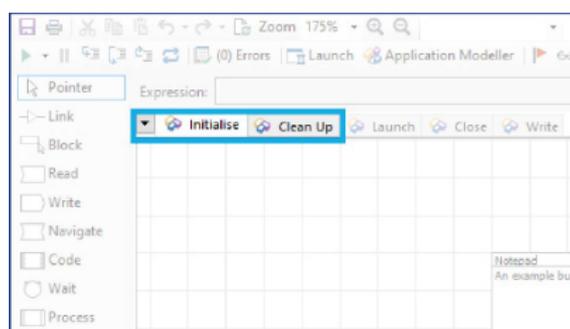
Para que a interação entre um objeto e uma aplicação ocorra, é necessário utilizar um modelo de aplicação. Este modelo de aplicação é desenvolvido utilizando-se o *Application Modeller*, no *Object Studio*. É através do *Application Modeller* que o objeto consegue executar ações como leitura, escrita e captura de dados, dentre outras. Para realizar essas ações em um site, por exemplo, os elementos desse site são identificados

Figura 4 – Estágios adicionais do Object Studio



Fonte: DOCS (2023a)

Figura 5 – Páginas padrão do Object Studio



Fonte: DOCS (2023a)

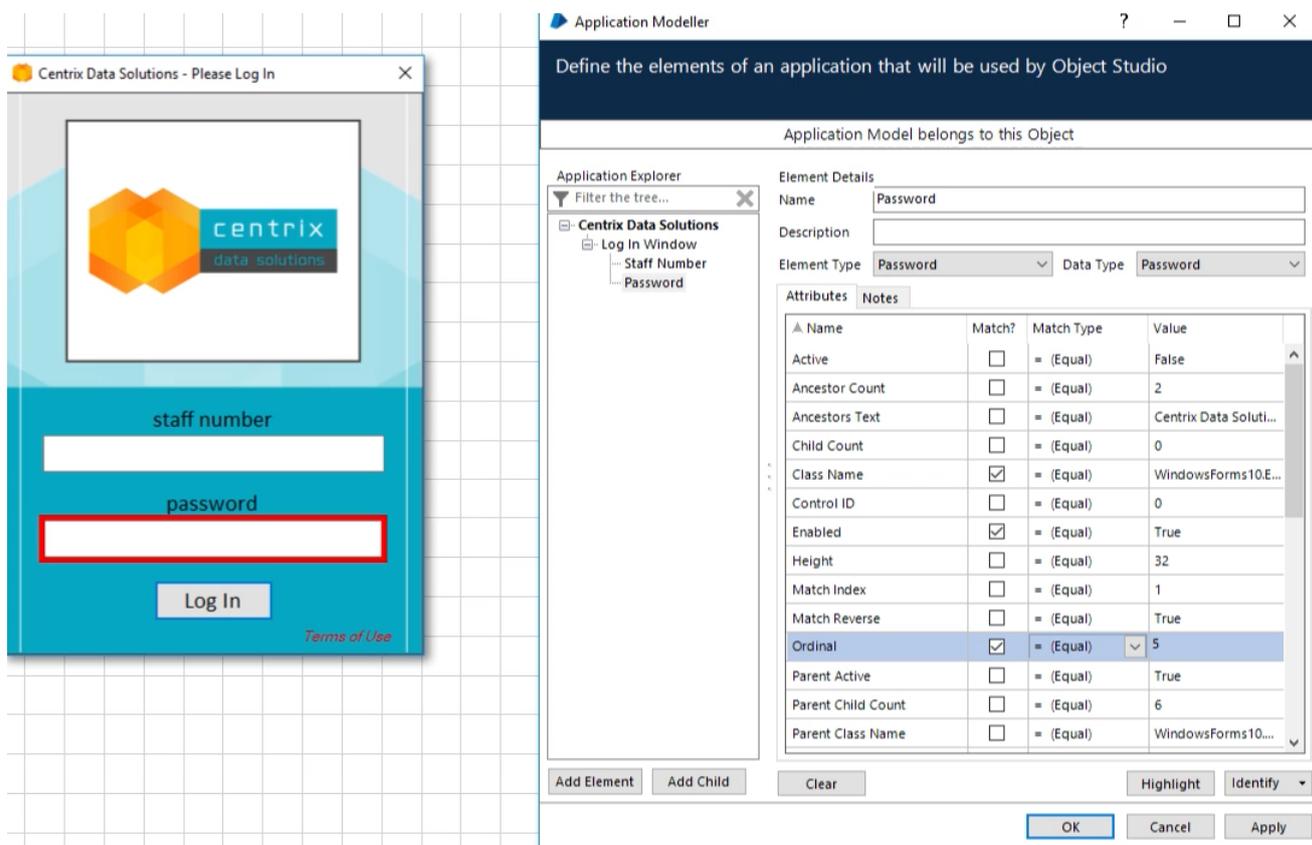
com critérios únicos e capturados no *Application Modeller*. Este método é chamado de “espionagem” ou “mapeamento”. A Figura 6 ilustra o funcionamento do mesmo, em que na aplicação *Centrix Data Solutions*, foi possível fazer o mapeamento e reconhecimento do campo *password* após campos com identificadores únicos serem selecionados na janela de captura do *Application Modeller*.

2.2.3 Control Room

O terceiro componente da arquitetura do Blue Prism, é o Control Room. O Control Room funciona como um centro de comando, onde é possível gerenciar diversas atividades no Blue Prism. As funcionalidades mais importantes do *Control Room* são: a lista de robôs disponíveis para execução; os recursos disponíveis para execução dos robôs; os detalhes do robô a ser executado; agendamento da execução do robô; e os *logs* do robô.

Outra funcionalidade importante do *Control Room* é a possibilidade de acompanhar os itens de uma fila. Os robôs criados no Blue Prism têm seu funcionamento baseado

Figura 6 – Funcionamento do Application Modeller



Fonte: DOCS (2023a)

em uma fila de trabalho. Ou seja, o robô popula uma fila de trabalho com informações externas, que podem ser provenientes de formulários, planilhas, bancos de dados. A seguir, o robô seleciona o primeiro item da fila e executa sua lógica. Ao fim da execução, o robô seleciona o próximo item para executar sua lógica e isso se repete até que não se tenha mais nenhum item para ser executado. Dado que não se tenham mais itens na fila, o robô encerra sua execução. Quando uma fila de determinado robô é selecionada, todos os itens executados são exibidos no *Control Room* com suas respectivas informações, como identificador do item, prioridade, status, servidor, entre outras. A Figura 7, ilustra itens de uma fila no *Control Room*.

O símbolos que se encontram à esquerda da Figura 7 possuem o seguinte significado: o símbolo de três pontos indica que o item está pendente, ou seja, à espera de alguma ação; o símbolo de “check” no item indica que o mesmo foi concluído com sucesso; e o símbolo de bandeira indica que houve erro na execução do item, gerando uma exceção. No Blue Prism, existem vários tipos de exceções que podem ser usadas para capturar e lidar com erros durante a execução de processos de automação. Os diferentes tipos de exceções são importantes para que os erros possam ser entendidos e tratados de maneira adequada

específico de caractere, mas sim, a quaisquer caracteres dentro de um texto, sejam eles alfabéticos, numéricos ou especiais.

Para Fitzgerald (2012, p.16): “Expressões regulares são cadeia de caracteres de texto especialmente codificadas, utilizadas como padrões para corresponder a conjuntos de cadeia de caracteres.”

E, para Jargas (2016, p.19):

Uma Expressão Regular (Regex) é um método formal de se especificar um padrão de texto. É uma composição de símbolos, caracteres com funções especiais, chamados “metacaracteres” que, agrupados entre si e com caracteres literais, formam uma sequência, uma expressão. Essa expressão é testada em textos e retorna sucesso caso esse texto obedeça exatamente a todas as suas condições. Diz-se que o texto “casou” com a expressão.

Acredita-se que o primeiro uso de expressões regulares foi feito pelo cientista computacional Ken Thompson, em seu artigo publicado em 1968, chamado “Regular Expression Search Algorithm”. Após a publicação do artigo, o uso das expressões regulares se popularizou no mundo da programação, sendo encontradas no editor de texto *QED* e tornando-se parte crucial do conjunto de ferramentas que surgiram a partir do sistema operacional Unix - como o *ed*, *sed*, *AWK*, entre outros (FITZGERALD, 2012).

As expressões regulares completas são compostas de dois tipos de caracteres. Os caracteres especiais são chamados de metacaracteres, enquanto os demais são chamados de caracteres literais, ou de texto normal (FRIEDL, 2016). Como exemplo, uma expressão regular de caracteres literais seria “mundo”. Neste caso, a expressão regular irá corresponder exatamente à sequência “mundo” e não corresponderá a nenhuma outra sequência de caracteres. Uma expressão comumente utilizada para dizer que um resultado foi encontrado por uma expressão regular é o *match*. Dessa forma, para a frase “Olá, mundo!” o *match* será dado na cadeia de caracteres “mundo”. Um outro exemplo do funcionamento de uma expressão regular pode ser dado com o uso de metacaracteres, como a seguinte: “\w*”. Para essa expressão regular, o *match* será dado em qualquer cadeia de caracteres que consista apenas de caracteres alfanuméricos.

Segundo Friedl (2016, p.5), uma analogia utilizada para facilitar a compreensão do funcionamento das expressões regulares é considerá-las sua própria linguagem. Ou seja, os caracteres literais funcionam como as palavras, e os metacaracteres funcionam como a gramática.

Um ponto importante sobre o funcionamento das expressões regulares é que, se em uma cadeia de caracteres houver mais de um resultado ou *match*, o que será escolhido será

aquele que estiver mais próximo da esquerda. Em outras palavras, aquele que começar antes tem preferência sobre aqueles que comecem depois. Como exemplo, tem-se a seguinte frase: “É encantador o modo como ele canta”. Se, para esta frase, tem-se uma expressão regular para encontrar a sequência de caracteres “canta”, o resultado apresentado será a cadeia de caracteres “canta” que se encontra dentro da palavra “encantador” em vez de retornar a palavra “canta” no fim da frase. Esse resultado se dá pela regra de retornar o resultado que estiver mais à esquerda.

Além disso, também é importante destacar que as expressões regulares possuem um vasto conteúdo e possibilidades de uso. Contudo, no presente trabalho, serão apresentadas apenas as operações básicas utilizadas no desenvolvimento da solução automatizada.

2.3.1 Metacaracteres

Os metacaracteres, ou caracteres especiais, são caracteres não alfabéticos que possuem função específica quando utilizados nas expressões regulares (IBM, 2022). Na Tabela 1, encontram-se os metacaracteres e seus respectivos nomes e categorias.

Tabela 1 – Representação dos Metacaracteres.

Metacaractere	Nome	Categoria
. [] [^...]	Ponto Lista Lista negada	Representantes
? * + { }	Opcional Asterisco Mais Chaves	Quantificador
^ \$ \b	Circunflexo Cifrão Borda	Âncora
\ () \1	Escape Ou Grupo Retrovisor	Outros

Fonte: Do Autor

Como visto na Tabela 1, os metacaracteres podem ser divididos em quatro categorias, de acordo com as características comuns entre eles. As categorias foram definidas como “Representantes”, “Quantificadores”, “Âncoras” e “Outros” (JARGAS, 2016).

2.3.1.1 Representantes

Os metacacteres do tipo “Representantes” são aqueles que dão *match* com algum caractere. Os metacacteres desse tipo são: ponto, lista e lista negada.

O metacaractere “ponto” dá *match* com qualquer caractere, incluindo ele próprio. Na Figura 8, temos o exemplo da expressão regular “e.tendido”, essa expressão pode retornar resultados como: “entendido”, “extendido”, entre outros (JARGAS, 2016).

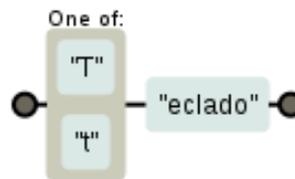
Figura 8 – Metacaractere Ponto



Fonte: devfuria (2022)

O metacaractere “lista” permite à expressão regular dar *match* com qualquer caractere que esteja dentro dos colchetes. Ou seja, qualquer caractere que esteja listado, como mostra a Figura 9, em que a expressão “[Tt]eclado” pode retornar: “Teclado” ou “teclado” (JARGAS, 2016).

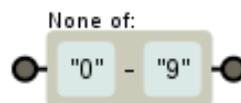
Figura 9 – Metacaractere Lista



Fonte: devfuria (2022)

Já, o metacaractere “lista negada” é exatamente o oposto da “lista”: o *match* ocorre com qualquer caractere que não esteja listado. A Figura 10 traz como exemplo a expressão regular “[^0-9]” que retorna qualquer caractere que não seja um número entre 0 e 9 (JARGAS, 2016).

Figura 10 – Metacaractere Lista Negada



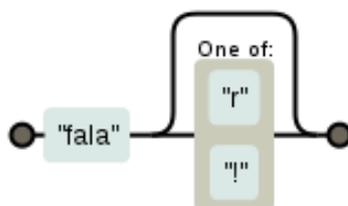
Fonte: devfuria (2022)

2.3.1.2 Quantificadores

Os metacaracteres do tipo “Quantificadores” são: opcional, asterisco, mais e chaves. Esse tipo de metacaracteres serve para indicar quantas vezes o caractere anterior será repetido.

O metacaracteres “opcional” funciona como o nome sugere: ele pode dar *match* com uma palavra contendo ou não determinado caractere. Como mostra a Figura 11, a expressão regular “fala[r!]?” pode ter como resultado: “fala”, “falar” ou “fala!” (JARGAS, 2016).

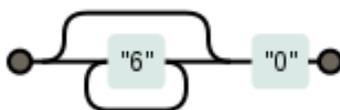
Figura 11 – Metacaracteres Opcional



Fonte: devfuria (2022)

O metacaracteres “asterisco” é utilizado para retornar o elemento anterior em qualquer quantidade, incluindo zero, porém, ele sempre tentará a maior quantidade possível. A Figura 12 exemplifica isso com a expressão regular “6*0”, esta expressão pode retornar: 0, 60, 660, 6666660, entre outros (JARGAS, 2016).

Figura 12 – Metacaracteres Asterisco

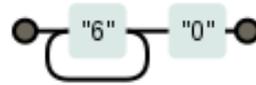


Fonte: devfuria (2022)

O metacaracteres “mais” funciona de maneira semelhante ao “asterisco”, tendo como diferença o fato de que o “mais” retorna, no mínimo, uma repetição do elemento anterior. Como exemplo, na Figura 13, tem-se a expressão “6+0” que pode retornar: 60, 660, 66660, entre outros (JARGAS, 2016).

O metacaracteres “chaves” permite especificar quantas vezes o elemento anterior deve ser repetido. Na Figura 14, tem-se os possíveis resultados da expressão regular “n{1,4}” que são: n, nn, nnn e nnnn. Ou seja, esta expressão regular permite que o *match* seja

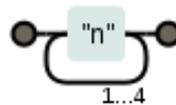
Figura 13 – Metacaractere Mais



Fonte: devfuria (2022)

desde um “n” até quatro “n” (JARGAS, 2016).

Figura 14 – Metacaractere Chaves



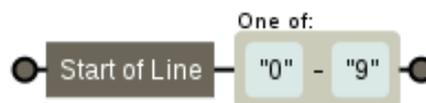
Fonte: devfuria (2022)

2.3.1.3 Âncoras

Os metacaracteres do tipo “Âncora” marcam uma posição específica de uma linha onde o *match* ocorrerá. São eles: circunflexo, cifrão e borda.

O “circunflexo” é utilizado para marcar o começo de uma linha. Como ilustrado na Figura 15, a expressão regular “ $^[0-9]$ ” dá *match* com qualquer número entre 0 e 9 que estiver no começo da linha (JARGAS, 2016).

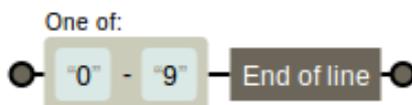
Figura 15 – Metacaractere Circunflexo



Fonte: devfuria (2022)

O “cifrão” tem funcionamento similar ao “circunflexo”: ele dá *match* com caracteres no fim de uma linha. Na Figura 16, tem-se os possíveis resultados para a expressão regular “[0-9]\$”. Ou seja, esta expressão regular retorna números entre 0 e 9 que se encontrem no fim de uma linha. Uma observação importante é que o \$ só funciona se colocado ao final da expressão regular (JARGAS, 2016).

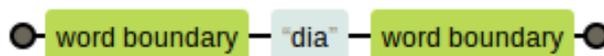
Figura 16 – Metacaractere Cifrão



Fonte: devfuria (2022)

O metacaractere “borda” marca, como o nome sugere, um limite. Ou seja, onde uma palavra começa e onde uma palavra termina. A Figura 17 mostra como a expressão regular “\b dia\b” funciona. Se uma palavra for colocada entre o metacaractere “borda”, como, neste caso, “dia”, pode-se ter resultados como: diafragma, melodia, entre outros (JARGAS, 2016).

Figura 17 – Metacaractere Borda



Fonte: devfuria (2022)

2.3.1.4 Outros

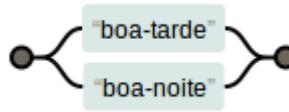
Os metacaracteres do tipo “Outros” são aqueles que têm funções específicas e não tem relação entre si. São eles: escape, ou, grupo e retrovisor.

O metacaractere “escape” tem como funcionalidade tirar a função de um metacaractere. Ou seja, é utilizado quando precisa-se que um metacaractere funcione como um caractere literal. Como exemplo, na expressão regular “[0-9]\.[0-9]”, o ponto funciona como um caractere literal e não um metacaractere, e retorna resultados como: 9.6, 9.9, entre outros.

Já, o metacaractere “ou” tem como função retornar ou um *match* ou outro. No exemplo abaixo, Figura 18, temos que para a expressão regular “boa-tarde|boa-noite” retorna-se uma das duas palavras: ou boa-tarde ou boa-noite (JARGAS, 2016).

O metacaractere “grupo” tem como funcionalidade agrupar. Em um grupo pode-se ter um ou mais caracteres, um ou mais metacaracteres e outros grupos. A Figura 19 exemplifica os possíveis resultados da expressão regular “boa-(tarde|noite)”, que podem

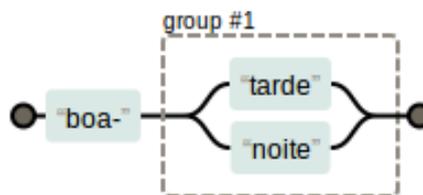
Figura 18 – Metacaractere Ou



Fonte: devfuria (2022)

ser: boa-tarde ou boa-noite (JARGAS, 2016).

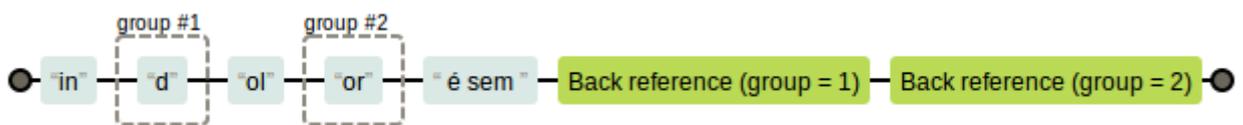
Figura 19 – Metacaractere Grupo



Fonte: devfuria (2022)

Por fim, tem-se o metacaractere “retrovisor”, ele pode retornar parte de um valor que foi retornado pela expressão regular. Por exemplo, na Figura 20, tem-se que para a expressão regular “in(d)ol(or) é sem \1\2” o resultado é “indolor é sem dor” (JARGAS, 2016).

Figura 20 – Metacaractere Retrovisor



Fonte: devfuria (2022)

2.3.2 Regex Engine

Segundo Goyvaerts (2007, p.3) uma engine de expressão regular “regex engine” é um pedaço de software que pode processar expressões regulares, tentando combinar a expressão regular com uma dada cadeia de caracteres.

Dessa forma, mesmo que as expressões regulares sejam consideradas uma linguagem própria, as linguagens de programação, através das *engines*, possuem diferentes sintaxes para descrevê-las dependendo de seu uso. Isso pode ser exemplificado pelo uso de *named capturing groups*. *Named capturing groups* (ou grupos de captura com nome) são uma extensão das expressões regulares que permitem nomear um grupo de captura dentro da expressão regular. O uso do *named capturing groups* facilita a leitura e a manutenção de expressões regulares complexas, pois é possível referir-se aos grupos pelo nome em vez de pelo índice numérico. Além disso, alguns recursos de processamento de expressão regular, como a substituição, permitem referências de grupo por nome em vez de número, tornando o código mais legível e menos propenso a erros. O *named capturing groups* é suportado em algumas linguagens de programação que utilizam expressões regulares, como o *Perl*, *Python*, *Ruby* e *.NET*.

Em *Python*, o *named capturing groups* possui a seguinte estrutura: $(?P<nome>grupo)$. Onde, “nome” é uma referência que deve ser uma sequência alfanumérica começando por uma letra e o “grupo” pode ser qualquer expressão regular. Outra linguagem que também suporta *named capturing groups* é o *.NET*. Em *.NET*, o *named capturing groups* é estruturado como: $(?'nome'grupo)$ ou $(?<nome>grupo)$. Onde, o “nome” também significa uma referência e o “grupo” qualquer expressão regular. Comparando as estruturas das duas linguagens, não há “P” na sintaxe do *.NET* e pode-se usar tanto apóstrofo quanto colchetes angulares para o nome de referência (GOYVAERTS, 2007). Na Tabela 2, o uso do *named capturing groups* para as duas linguagens é exemplificado:

Tabela 2 – Exemplo de *Named Capturing Groups* em duas linguagens de programação.

Linguagem de Programação	Regex	Texto	Match
Python	$(?P<ano>[0-9]{4})$	1999-12-31	1999
.NET	$(?'ano'[0-9]{4})$	1999-12-31	1999

Fonte: Do Autor

2.3.3 Expressões Regulares no Blue Prism

Para utilizar as expressões regulares no software Blue Prism, seus desenvolvedores criaram um objeto chamado “Extract Regex Values”. Como explicado na Seção 2.2, cada objeto pode ser utilizado quando chamado por uma *action*. Para a *action* que utiliza o objeto “Extract Regex Values”, tem-se os seguintes valores de entrada:

1. *Target string* - Este campo espera receber o texto em que o usuário deseja utilizar a expressão regular;

2. *Regex Pattern* - Este campo espera receber uma expressão regular que utiliza *named capturing groups*. Esta expressão regular é empregada no texto passado no campo *Target string*;
3. *Named Values* - Este campo espera receber uma coleção de variáveis contendo dois campos, sendo estes: “Name” e “Value”. No campo “Name”, o usuário deve inserir um nome que identifique o resultado da expressão regular; já o campo “Value” deve ser deixado vazio, pois é onde o resultado será alocado.

Um exemplo do uso da *action Extract Regex Values* pode ser observado na Figura 21. Onde, utiliza-se a regex “(?<Números>(\d+))” para se obter somente os dígitos do texto “Teste: 1234”.

Figura 21 – *Action Extract Regex Values*

The screenshot shows the 'Action Properties' dialog box for the 'Extract Regex Values' action. The 'Name' field is 'Regex Teste' and the 'Business Object' is 'Utility - Strings'. The 'Action' is 'Extract Regex Values'. The 'Inputs' tab is active, showing a table with the following data:

Name	Data Type	Value
Regex Pattern	Text	"(?<Números>(\d+))"
Target String	Text	"Teste: 1234"
Named Values	Collection	[Result]

At the bottom, there are settings for 'Stage logging' (Enabled) and 'Warning threshold' (System Default, 5 minutes). The 'Group' section on the right is expanded to 'Numbers'.

Fonte: Do Autor

Na Figura 22, pode-se observar o resultado da regex. No campo “Name” tem-se o nome do grupo, chamado “Números”. E, no campo “Value” tem-se o resultado obtido da regex.

Este objeto possui algumas características que devem ser levadas em consideração durante seu uso. A característica mais importante é a que ele foi criado utilizando-se as classes de expressões regulares disponibilizadas pelo framework Microsoft .NET. As

expressões regulares da Microsoft .NET utilizam uma sintaxe compatível com a sintaxe das expressões regulares da linguagem de programação PERL 5 (DOCS, 2022). Dessa forma, caso a expressão regular passada dentro da action que utiliza o objeto “Extract Regex Values” possua a sintaxe de qualquer outra linguagem de programação que não a PERL 5, a action retornará um erro de sistema, ou *system exception*.

Figura 22 – Resultado da regex

Fields	Initial Values	Current Values
Name (Text)		Value (Text)
Números		1234

Fonte: Do Autor

Uma vez concluída a revisão de literatura acerca das ferramentas essenciais para este trabalho, incluindo o funcionamento do *software Blue Prism* e as diferentes operações realizadas com as expressões regulares, é possível aplicar os conhecimentos obtidos na implementação do projeto de automação do presente trabalho.

3 DESENVOLVIMENTO

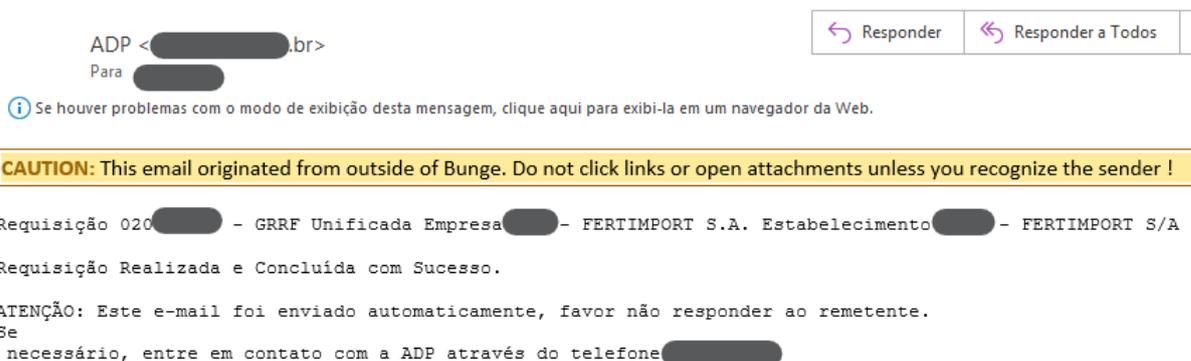
Esta seção descreve o desenvolvimento de um projeto de automação realizado na área de Recursos Humanos (RH) da empresa Bunge. O projeto é separado em duas partes: a primeira parte consiste em realizar a extração de dados de um documento em formato digital e inserir os mesmos em uma planilha eletrônica; e a segunda parte consiste em separar o documento por CNPJ, ou seja, aquelas páginas que possuem o mesmo CNPJ são separados em outros documentos, que são enviados ao departamento financeiro. Estes documentos possuem informações de pagamento de Fundo de Garantia do Tempo de Serviço (FGTS) de ex-colaboradores das empresas Bunge e Fertimport, e são conhecidos como Guia de Recolhimento Rescisório de FGTS, ou simplesmente “guia GRRF”. Para a realização do projeto, foi utilizado o Microsoft Excel para o desenvolvimento da planilha, o navegador Google Chrome, utilizado para acessar o site da ADP eXpert, empresa que gerencia diferentes processos de RH e onde se encontram as guias GRRF e, por fim, para a automatização do projeto foi utilizado o software Blue Prism.

3.1 EXECUÇÃO DO PROCESSO NÃO AUTOMATIZADO

Esta seção descreve brevemente o processo não automatizado, realizado por um operador humano. Como primeiro passo do processo, o operador humano deve identificar o recebimento de uma notificação via e-mail enviado pela ADP eXpert, informando o número de requisição da Guia de Recolhimento Rescisório do FGTS (GRRF) a ser processada. O formato da notificação pode ser visto na Figura 23.

Figura 23 – Corpo do e-mail

ADP: GRRF Unificada



Fonte: Do Autor

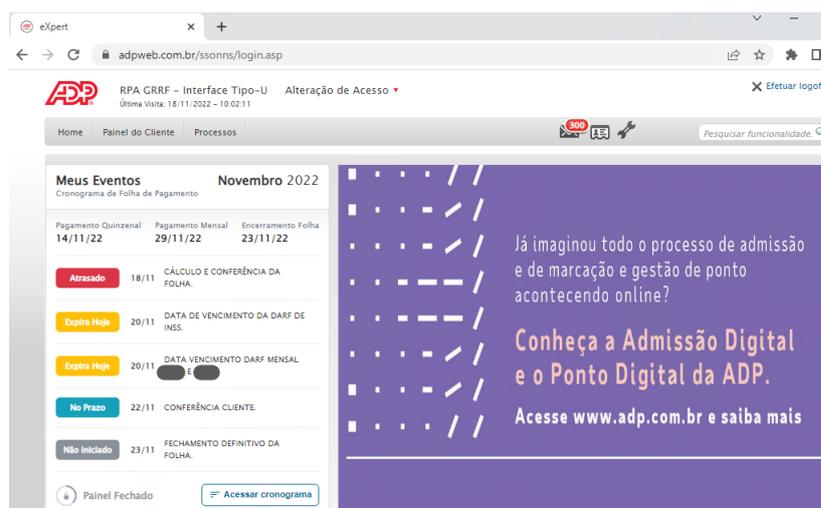
Uma vez que o número de requisição é obtido, é necessário abrir o navegador, acessar o sistema da ADP eXpert e realizar o processo conhecido como *login* (Figura 24). Após fazer o *login*, o operador é redirecionado para a página inicial do site (Figura 25), onde o mesmo pode realizar diferentes ações. Para ter acesso aos arquivos, o usuário deve acessar a página “Painel de Cliente” e clicar na opção ”Verificar Requisições”, que irá redirecionar o usuário para a página “Requisições de Usuário”.

Figura 24 – Login no site ADP



Fonte: Do Autor

Figura 25 – Página inicial da ADP

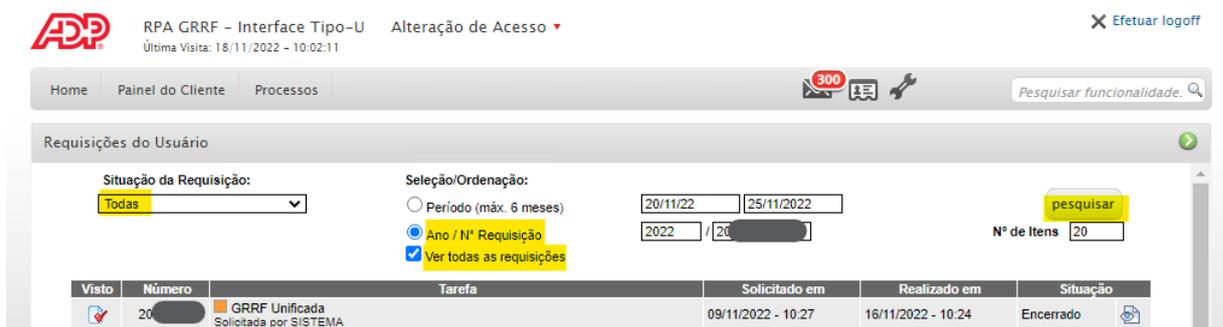


Fonte: Do Autor

Na página “Requisições de Usuário”, Figura 26, o usuário deve realizar as seguintes ações: selecionar a opção “Todas” no campo “Situação da Requisição”; clicar no item

“Ano/Nº Requisição” e adicionar o número de requisição; após, deve clicar no item “Ver todas as requisições” e clicar no botão “Pesquisar”. Em seguida, a página irá carregar o número da requisição pesquisada com opções de visualizar e baixar os arquivos que a acompanham. Os arquivos com extensão “.re” e “.csv” se encontram em uma aba chamada “Arquivos”. Para encontrá-la, o usuário deve clicar na coluna “Situação”, e em seguida, clicar no ícone ao lado para ser redirecionado à página “Solicitações de Processamento”. Nesta página, ao clicar no ícone da coluna “Visto”, a página é atualizada com diversas abas, dentre elas a aba “Arquivos”. O usuário então clica na aba “Arquivos” e são exibidos dois arquivos que devem ser baixados, conforme ilustrado na Figura 27. O primeiro é o arquivo com extensão “.re”, que deve ser armazenado em uma pasta definida pela área de RH. Este arquivo é utilizado para comunicar o governo sobre os recolhimentos feitos e para que a Caixa Econômica Federal faça a distribuição na conta do FGTS dos ex-colaboradores. E, o segundo arquivo é o de extensão “.csv” que é salvo na pasta *Downloads* e contém informações dos ex-colaboradores.

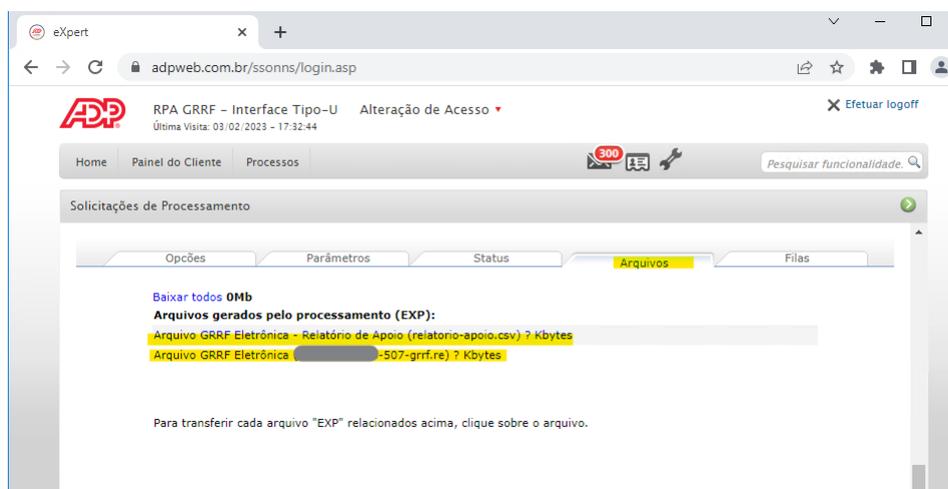
Figura 26 – Página Requisições do Usuário



Fonte: Do Autor

Após o download dos arquivos, deve-se clicar no botão “cancelar” para voltar à página “Requisições de Usuário”, onde encontra-se a requisição pesquisada. Nesta página, deve-se clicar no ícone da coluna “Visto”, que redireciona o usuário à página “Dados da Requisição”. Na página “Dados da Requisição”, Figura 28, encontram-se as guias GRRF, em arquivos em formato PDF, bem como duas informações pertinentes ao processo, que são: o tipo de processo realizado, ou seja, se é uma rescisão de contrato ou uma rescisão complementar de contrato; e para qual empresa o funcionário trabalhava. Essas informações são utilizadas em uma planilha eletrônica onde serão alocadas outras informações dos ex-colaboradores.

Figura 27 – Aba de Arquivos



Fonte: Do Autor

Figura 28 – Página Dados da Requisição



Fonte: Do Autor

A segunda parte do processo consiste na manipulação das guias GRRF. As guias GRRF constituem um arquivo em formato de pdf que possui a seguinte estrutura: uma página chamada de “guia” e uma página chamada de “espelho”. Para cada empresa tem-se uma guia. E essa guia possui o valor total a ser pago pela empresa a n beneficiários, entre outras informações, como: nome do pagador, data de vencimento e CNPJ. Para cada beneficiário tem-se um espelho. E este espelho contém as informações do receptor, como: nome, número do PIS (Programa de Integração Social) e valor a ser recebido.

Este arquivo pdf pode ser disponibilizado em dois formatos:

- a) o primeiro formato consiste em uma página contendo a guia com o valor total a ser pago por determinada empresa (Figura 29), e sequencialmente estão os espelhos de cada beneficiário com seu respectivo valor a ser recebido (Figura 30). Como exemplo, neste formato, o arquivo gerado pela ADP eXpert teria como primeira página uma guia com o valor total de R\$ 100,00 a ser pago pela empresa *A* para dois funcionários. Dessa forma, as duas páginas seguintes seriam dois espelhos contendo o valor de R\$ 70,00 para um beneficiário e o valor de R\$ 30,00 para o outro;
- b) o segundo formato tem a mesma estrutura do primeiro: uma página com o valor total a ser pago por determinada empresa e seus respectivos espelhos de beneficiários, com a diferença de que essas páginas não vêm mais com a sequência “guia-espelho”. Neste formato, as páginas vêm em ordem aleatória. Ou seja, em um arquivo que contém guias de pagamento das empresas *A* e *B*, os espelhos dessas empresas estarão desordenados, misturados entre si.

O arquivo não deve ser enviado ao departamento financeiro em ordem aleatória nem contendo guias e espelhos de diferentes empresas no mesmo arquivo. Os espelhos e guias de uma mesma empresa são identificados através de seu CNPJ. Portanto, deve-se separar os espelhos e guias que contém o mesmo CNPJ e formar um novo arquivo. Dessa forma, o departamento financeiro recebe dois arquivos pdfs: o primeiro contendo somente a guia de pagamento; e o segundo, contendo a mesma guia de pagamento com os seus respectivos espelhos.

Uma vez que os arquivos foram devidamente separados por CNPJ, os mesmos são armazenados em uma pasta cujo nome é a data de validade da guia de pagamento. Nesta mesma pasta o arquivo “.re” baixado anteriormente é armazenado. Em seguida, é criada uma planilha no Excel onde serão inseridas as informações dos ex-colaboradores. Para tal, são utilizados os arquivos baixados anteriormente no site da ADP eXpert, chamados “relatorio-apoio” e a guia GRRF. Da guia GRRF são extraídas as seguintes informações: valor recebido por funcionário, data do pagamento, valor total da guia e competência; do relatório de apoio são extraídos nome e matrícula do beneficiário; e do site da ADP, é extraído o tipo de rescisão. Todas estas informações são colocadas na planilha e o arquivo é salvo.

Por fim, posto que a planilha está preenchida (Figura 31) e os arquivos estão adequadamente organizados, a área de RH realiza uma verificação acerca das informações presentes nos arquivos. Uma vez que a verificação foi bem-sucedida, envia-se um e-mail para o departamento financeiro contendo os arquivos de extensão “.pdf” para efetuação do pagamento e o arquivo Excel é enviado para o software de gerenciamento de processos

de negócios, SAP.

Figura 29 – Exemplo de guia GRRF

FGTS FUNDO DE GARANTIA DO TEMPO DE SERVIÇO

GRRF - Guia de Recolhimento Rescisório do FGTS
Gerada em 15/01/2023 21:09:13

Versão do Aplicativo: 3.3.17 - 24/01/2020

01 - Razão social/Nome BUNGE ALIMENTOS S A		02 - CNPJ/CEI 84.046.101/0247-00	
03 - Endereço (logradouro, nº, andar, apartamento) BR 364 S N KM 20		04 - Contato/DDD/telefone 600	
05 - CEP 78.746-060			
06 - Bairro/distrito PARQUE INDUSTRIAL VE	07 - Município RONDONOPOLIS	08 - UF MT	09 - FPAS 507
10 - Simples 1		14 - Qtd de Trabalhadores 1	

11- Identificador: 01537

12- Total a Recolher: 10,41

13- Data de Validade = 20/01/2023

Atenção: não receber após Validade

Código de Barras: [] [] [] 303404610122

Autenticação mecânica

Via Empresa

Fonte: Do Autor

Figura 30 – Exemplo de espelho da guia GRRF

Dados do Trabalhador

Nome: JONATHAN

PIS/PASEP: Admissão: Categoria: 01

Data Nascimento: Data Opção: CTPS:

Movimentação: Aviso Prévio: Dissídio/Acordo:

Informações Financeiras

	Mês Anterior a Rescisão	Mês Rescisão	Aviso Prévio Indenizado	Multa Rescisória
Remuneração/Saldo	0,00	500,11	2.473,70	1.800,89
Depósito	0,00	40,00	197,89	720,35
JAM	0,00	0,00	0,00	0,00
Encargos	0,00	0,00	0,00	0,00
Contrib.Social	0,00	0,00	0,00	0,00

Valor Trabalhador: 958,24 Valor Devido pela Empresa: 958,24

Cabe ao trabalhador os valores de Depósito e JAM

Fonte: Do Autor

Figura 31 – Dados populados na planilha

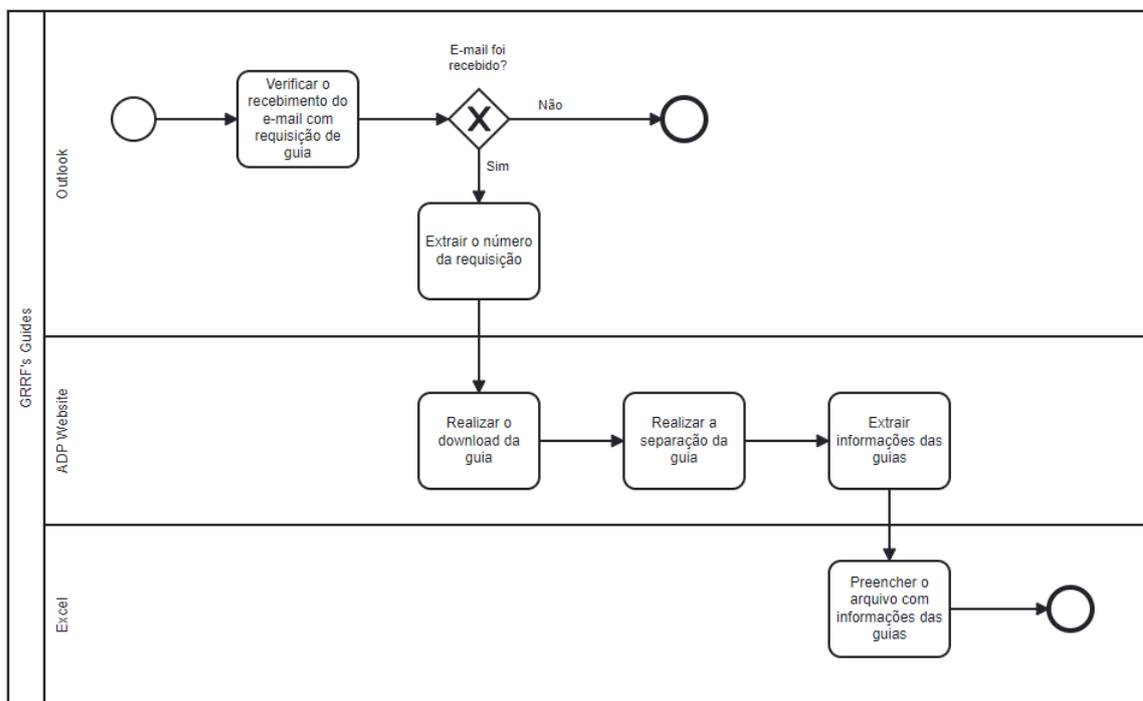
	A	B	C	D	E	F	G
1	ID	NOME	VALOR GRRF por func	DATA PAGTO	Valor Total	Competencia	Tipo
2	25	Franciscd	45,09	23/01/2023	1.003,33	23/jan	Rescisão
3	25	Jonathan	958,24	23/01/2023	1.003,33	23/jan	Rescisão

Fonte: Do Autor

3.2 AUTOMATIZAÇÃO DO PROCESSO COM BLUE PRISM

Nesta seção são relatados os passos necessários para a automação do processo descrito na Seção 3.1 com uso do software Blue Prism. Na Figura 32, encontra-se o diagrama do projeto, criado com a ferramenta *bpmn.io*. Esta ferramenta possibilita a criação de diagramas de notação BPMN. BPMN, também conhecido como Notação de Modelagem de Processos de Negócio, é uma notação baseada em fluxograma para definir processos de negócios, desde os mais simples até os modelos mais complexos necessários para a execução dos processos (WHITE; MIERS, 2008).

Figura 32 – Diagrama do projeto GRRF's Guides

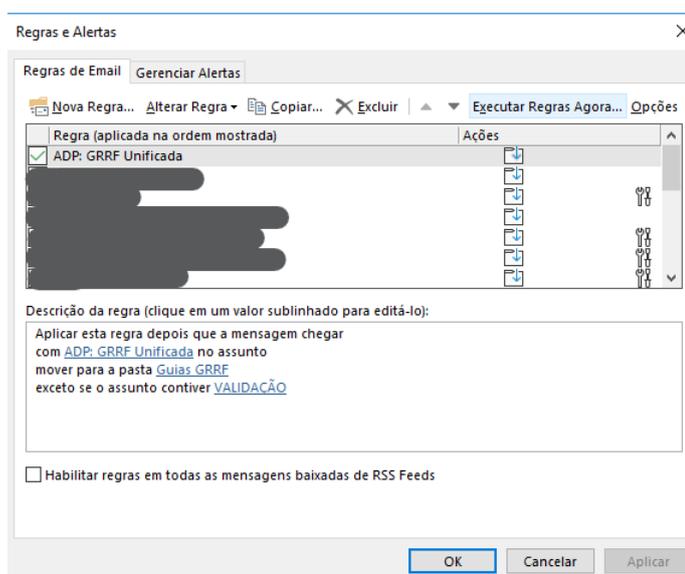


Fonte: Do Autor

A finalidade do fluxograma é estruturar as ações a serem realizadas dentro do Blue Prism. Dessa forma, pode-se observar que o fluxo da automação é dividido em três etapas. A primeira etapa acontece no software de gerenciamento de e-mails *Outlook*. Os itens da fila do processo são os números das requisições que são enviados no e-mail da área de RH.

Como o e-mail utilizado é o mesmo e-mail em que chegam diversos assuntos relacionados ao RH, foi realizada a configuração de uma regra no *Outlook* para que todo e-mail que chegar na caixa de entrada com o assunto “ADP:GRRF Unificada” seja enviado para a pasta “Guias GRRF”, criada unicamente para auxiliar no processo, a fim de evitar que o robô processe um e-mail que não envolva as guias GRRF. A ADP envia sempre dois e-mails ao mesmo tempo, um com o assunto “ADP:GRRF Unificada” que possui em seu corpo de texto o número de requisição que interessa ao processo, e outro com o assunto “ADP:GRRF Unificada VALIDAÇÃO” que não possui número de requisição. Dessa forma, foi adicionada uma exceção à regra, ou seja, se o assunto do e-mail recebido contiver a frase de interesse junto com palavra “VALIDAÇÃO”, o e-mail não deve ser movido para a pasta “Guias GRRF”. A Figura 33 ilustra a regra criada no *Outlook*.

Figura 33 – Regra criada no Outlook



Fonte: Do Autor

Com a regra criada, o robô verifica se um e-mail da ADP eXpert contendo um número de requisição foi recebido. Em seguida, tem-se a seguinte decisão: caso o e-mail não tenha sido recebido, o robô encerra sua atividade; caso o e-mail tenha sido recebido, o robô extrai o número de requisição e o salva. Esta condição leva o fluxo para a segunda etapa, realizada no sistema ADP eXpert. Nesta etapa, o robô realiza o login no site e insere o número de requisição para fazer o *download* dos arquivos utilizados no processo. Após, o robô abre o arquivo das guias GRRF e separa as guias em novos arquivos de acordo com seus respectivos CNPJ's. E, ao fim desta etapa, as informações das guias são extraídas e salvas. Por fim, a última etapa consiste em utilizar o software Excel para alocar as informações extraídas anteriormente da guia e o fluxo é terminado.

Com o desenho do processo definido através do fluxograma, o robô pode ser desenvolvido no Blue Prism. Como visto na Seção 2.2, o processo de execução dos robôs desenvolvidos no Blue Prism é dividido em páginas para que seu entendimento e visualização sejam mais fáceis e o processo fique devidamente organizado. Dessa forma, o processo realizado neste trabalho possui como principais páginas: *Main Page*, *Get Mails*, *Extract Data from PDF*, *Download Files from ADP Website* e *Send Mail*. O funcionamento de cada página é detalhado nas seções a seguir.

3.2.1 *Main Page* do Processo no Blue Prism

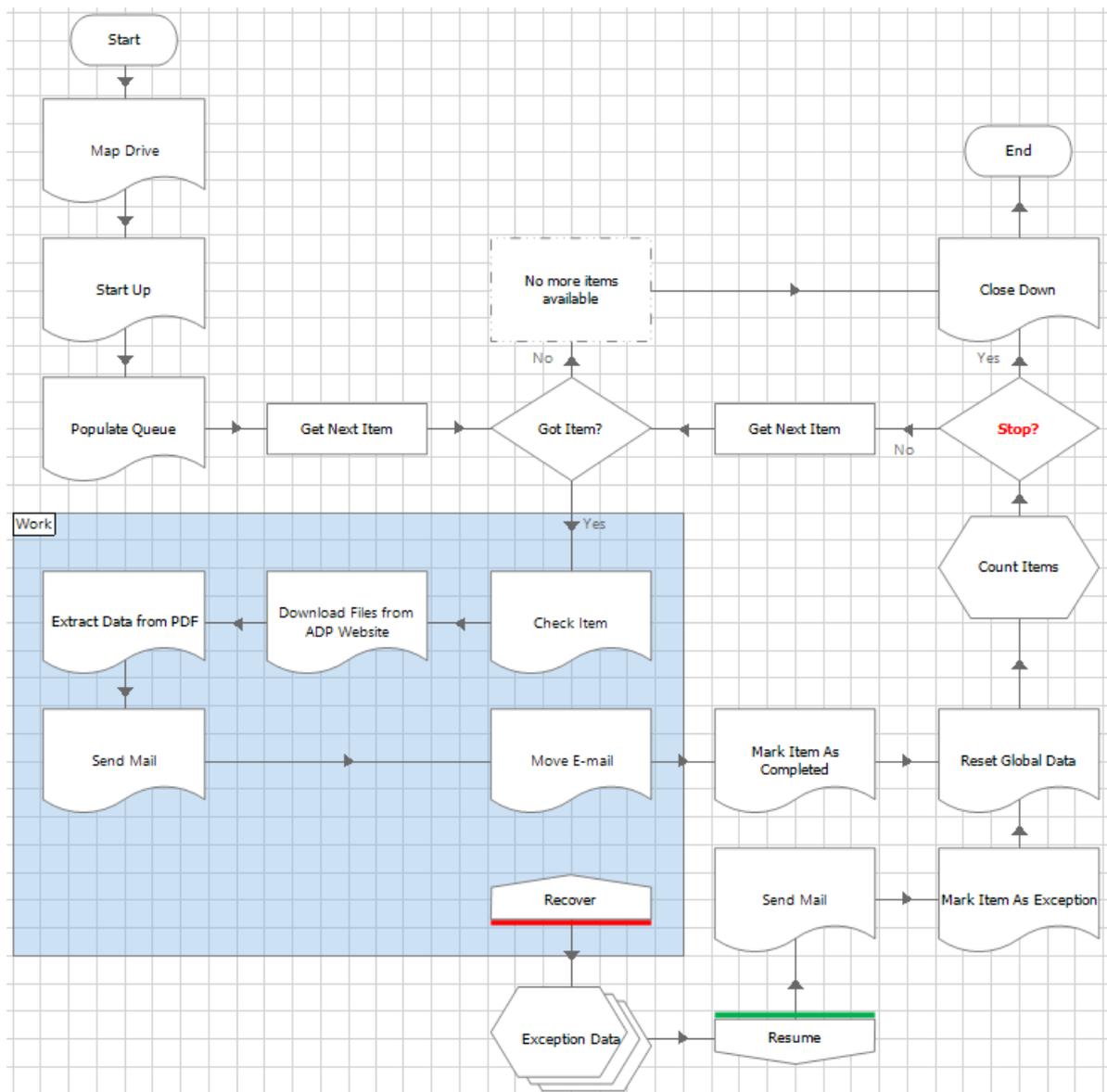
A *Main Page*, é a página principal do processo, onde toda sua lógica acontece. Esta página é criada, pois, à medida que o processo fica mais complexo e volumoso, é importante que os fluxogramas estejam bem organizados e claros. Dessa forma, o processo é dividido por pequenas ações que são alocadas em páginas individuais e são chamadas na página principal (*Main Page*), de acordo com a lógica estabelecida para o processo. Essa estrutura também facilita a identificação e solução de erros, pois, o usuário saberá exatamente em qual página o erro aconteceu. A *Main Page* do processo do presente trabalho foi feita com base no template disponibilizado pela Bunge, conforme ilustra a Figura 34, onde pode-se observar que o processo segue uma sequência de páginas para que o item seja executado com sucesso, sendo as principais: *Start Up*, *Populate Queue*, *Download Files from ADP Website*, *Extract Data from PDF* e *Send Mail*.

A página “Start Up” inicia o *Outlook*, abre o site da ADP no navegador *Google Chrome* e faz o login com as credenciais da área de RH. Em seguida, a página *Populate Queue* começa a executar sua lógica. A página *Populate Queue*, Figura 35, é uma página de funcionamento simples que apenas popula a fila do processo com os itens indicados. Para tal, é feita uma chamada para outra página chamada “Get Mails”, que é discutida com detalhes a seguir.

3.2.2 Página Get Mails

A página “Get Mails” tem como função principal adequar os itens da fila para que estes sejam populados na página “Populate Queue”. Os itens da fila são os números de requisição que são enviados pela ADP por e-mail. Dessa forma, o primeiro passo é listar os e-mails recebidos na pasta “Guias GRRF”, através da *action List Mails*, que retorna todos os e-mails recebidos da pasta indicada e os coloca em uma *collection*. Esta *collection* é colocada dentro de um *loop* para que de cada e-mail seja extraído o número de requisição, como mostra a Figura 36.

Figura 34 – Diagrama da *Main Page* do processo

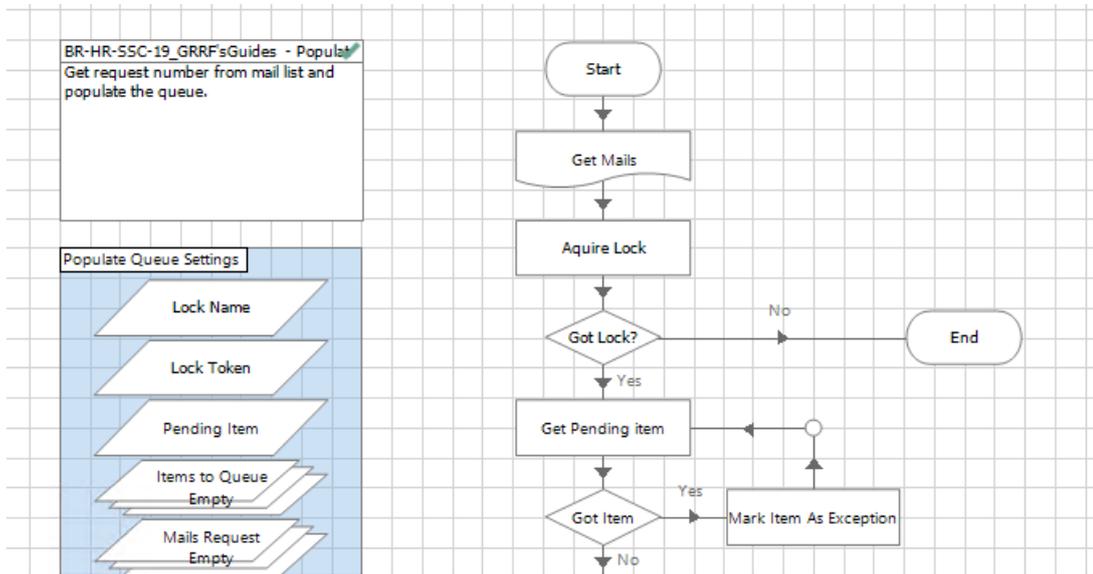


Fonte: Do Autor

3.2.3 Página *Download Files from ADP Website*

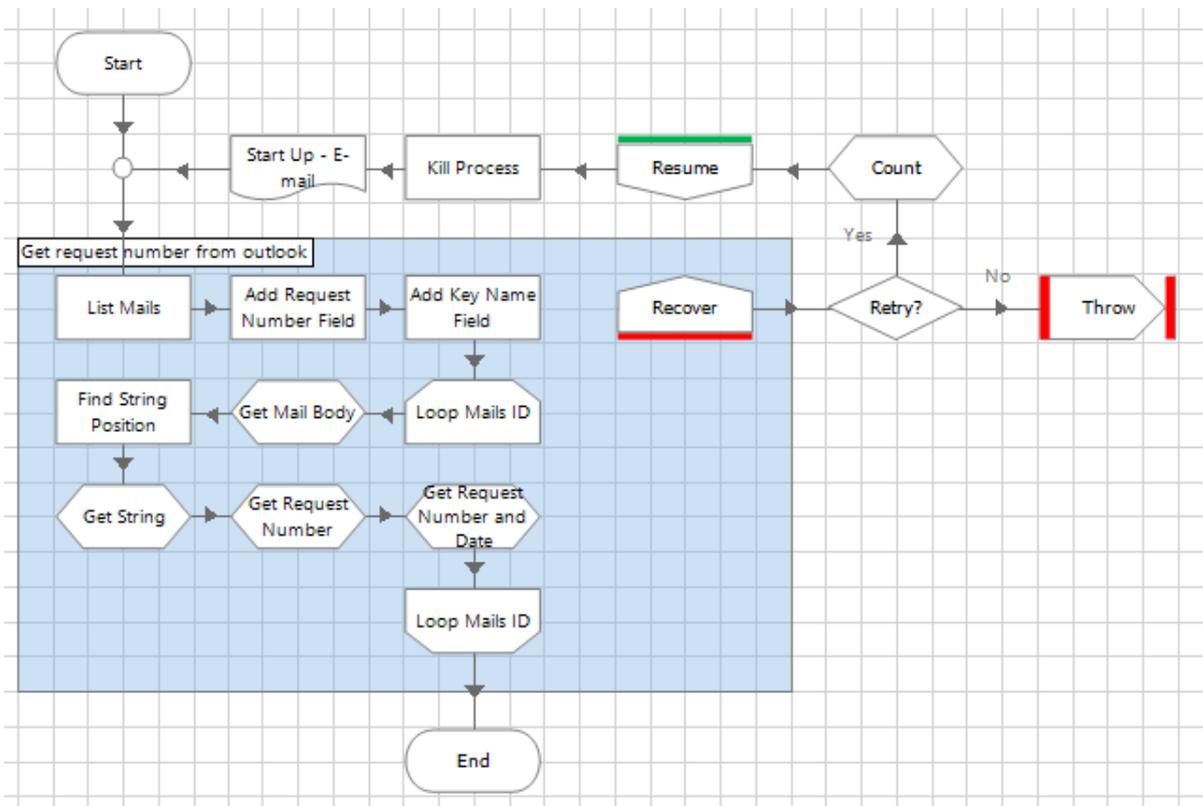
Uma vez que a fila do processo foi populada, tem-se o número de requisição e é possível realizar o download das guias GRRF, do arquivo “.re” e do relatório de apoio no site da ADP eXpert que pertencem a essa requisição. Para que o Blue Prism consiga interagir com o site da ADP, é necessário que as páginas utilizadas no site sejam mapeadas dentro de um objeto. Por isso, foi criado o objeto chamado “Set Request Parameters” que possibilita que o robô realize as ações necessárias no site. Este objeto recebe como parâmetro de entrada: o número de requisição disponível na fila do processo. Os parâmetros de saída são: o nome do arquivo das guias GRRF, o nome do arquivo de apoio “csv” e o nome do arquivo

Figura 35 – Página Populate Queue



Fonte: Do Autor

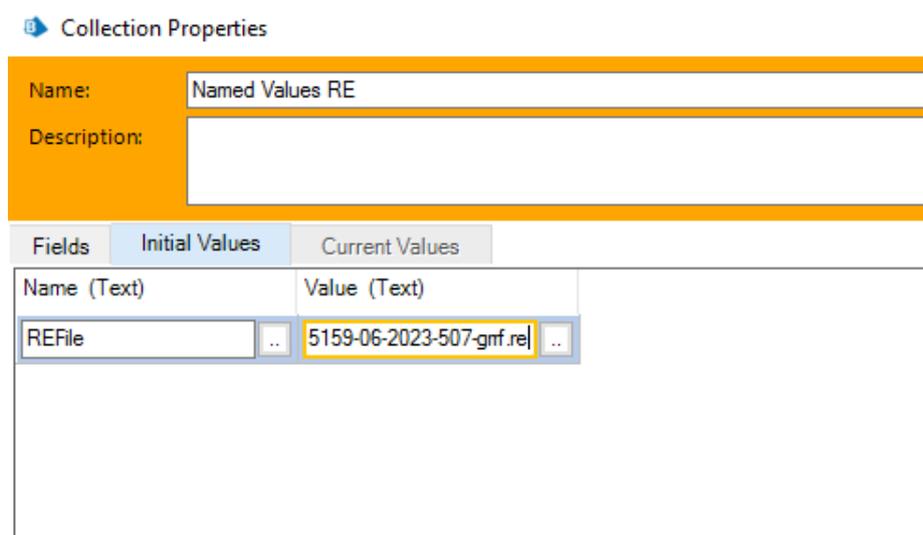
Figura 36 – Diagrama da página Get Mails



Fonte: Do Autor

“re”. Como o nome do arquivo “re” é dinâmico, faz-se o uso de uma expressão regular para obtê-lo. Como visto na Seção 2.3.3, o Blue Prism, por padrão, utiliza o metacaractere *grupo* em suas expressões regulares. Dessa forma, a expressão regular criada tem a seguinte estrutura: (?<REFile>(\d*[-]\d*[-]\d*[-]\d*[-]\w{4}[\.] \w{2})) onde, o lado esquerdo da expressão refere-se ao nome do grupo, chamado de “REFile” e o restante da expressão procura no nome do arquivo, uma quantidade de números quaisquer separados por quatro traços e acompanhados de quatro letras quaisquer separadas por um ponto e seguido de duas letras quaisquer. O resultado pode ser visto na Figura 37:

Figura 37 – Resultado da Regex REFile



The screenshot shows the 'Collection Properties' window in Blue Prism. The 'Name' field is 'Named Values RE'. Below it, there is a table with columns 'Fields', 'Initial Values', and 'Current Values'. The table has two rows: one for 'Name (Text)' with 'Value (Text)' and another row where the 'Fields' column contains 'REFile' and the 'Current Values' column contains '5159-06-2023-507-grf.rel'.

Fields	Initial Values	Current Values
Name (Text)		Value (Text)
REFile		5159-06-2023-507-grf.rel

Fonte: Do Autor

3.2.4 Página Extract Data from PDF

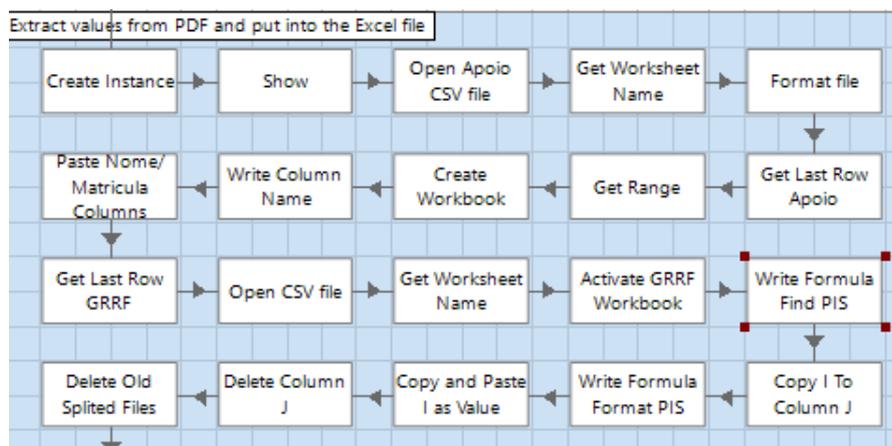
A página “Extract Data from PDF” é a página mais complexa do processo. Por isso, sua lógica foi dividida em dois blocos principais: o bloco “Extract values from PDF and put in the Excel file” e o bloco “Merge files”.

O bloco “Extract values from PDF and put in the Excel file”, como ilustra a Figura 38, inicia o fluxo da página abrindo três planilhas eletrônicas: a planilha “relatorio-apoio”, uma planilha em branco e uma planilha chamada “Relatório de PIS”. Nesta etapa do processo, o caminho feito pelo *key-user* (usuário-chave do processo) e pelo robô muda. Isso acontece porquê no processo não-automatizado o *key-user* compara visualmente o nome do ex-colaborador que está na planilha “relatorio-apoio” com os nomes que estão nas guias GRRF até encontrá-lo. Quando o nome é encontrado na guia, o *key-user* copia todas as informações necessárias da guia e as insere em uma nova planilha. Entretanto, o mesmo não pode ser feito pelo robô, pois, os nomes tem uma grande variação de acentuação e o

Blue Prism não reconhece nem acentos e nem cedilha. Conseqüentemente, quando o robô tenta fazer a comparação de nomes entre os arquivos, o Blue Prism sempre retorna o valor “false”, mesmo que os nomes sejam iguais. Dessa forma, como alternativa, foi utilizado o número do PIS de cada ex-colaborador para fazer a comparação. Como o número do PIS só está presente na guia GRRF, foi solicitado para a área um arquivo que contivesse nome, matrícula e número de PIS. Esse arquivo, chamado “Relatório de PIS”, foi criado unicamente para auxiliar no processo e deve ser atualizado pelo RH mensalmente em uma pasta que o robô também tem acesso.

De volta ao fluxo do bloco “Extract values from PDF and put in the Excel file”, após a abertura dos arquivos, o robô copia os nomes e matrículas dos ex-colaboradores que estão no “relatorio-apoio” e os insere na planilha em branco. É necessário que o número de PIS do ex-colaborador também esteja nesta planilha para que o robô seja capaz de fazer a comparação entre os números e possa colocar as informações extraídas das guias na linha correta. Para isso foi utilizada uma função do Excel chamada “PROCV”, que procura em uma tabela ou em um intervalo, o valor desejado com base em um parâmetro da mesma linha. O parâmetro utilizado foi a matrícula do ex-colaborador. A função “PROCV” então encontra a mesma matrícula no arquivo “Relatório de PIS” e retorna o número de PIS que está na mesma linha. Após o número do PIS ser encontrado, o mesmo ainda é formatado para o mesmo padrão que está no arquivo das guias GRRF, ou seja, sem traços e pontos. Em seguida, foi criado um *loop* para que os dados fossem manipulados e inseridos na planilha criada anteriormente. A próxima seção aborda o funcionamento desse *loop*.

Figura 38 – Fluxo para encontrar o número de PIS



Fonte: Do Autor

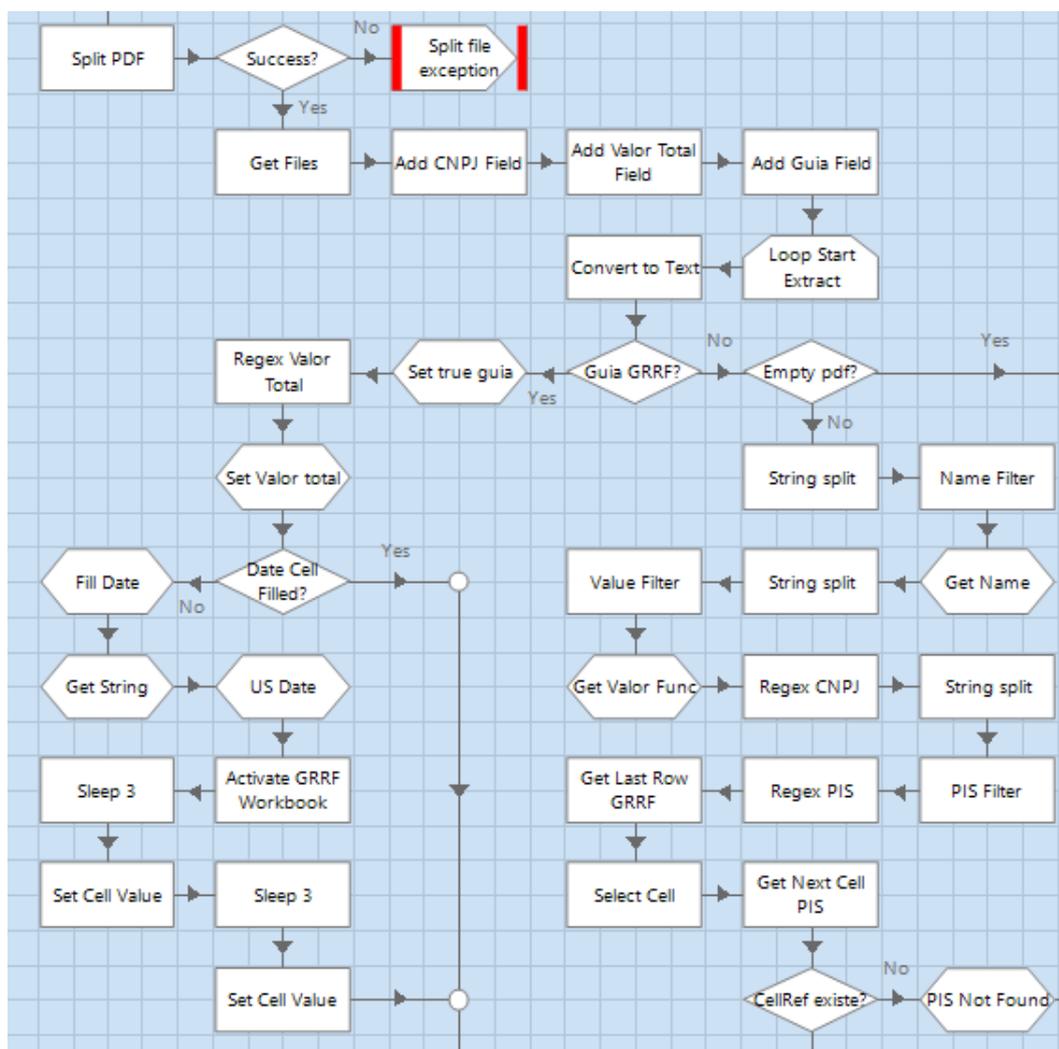
3.2.4.1 Loop para Manipulação dos Dados

Após a criação da planilha eletrônica, onde foi colocado o nome, matrícula e número de PIS dos ex-colaboradores, deve-se popular a planilha com as informações presentes nas guias GRRF, que são: o valor total da guia, a data de validade e o valor devido pela empresa. Para isso, como pode ser observado na Figura 39, o primeiro passo do fluxo é separar o arquivo das guias GRRF em páginas individuais dentro de uma *collection*, ou tabela, chamada “Splited Files”. Após a separação, o *loop* é iniciado com a conversão da página de formato pdf para um arquivo de texto para que o Blue Prism consiga extrair as informações de interesse. O *loop* é separado em dois fluxos: um fluxo utilizado para extrair informações do arquivo caso ele seja uma guia, e o segundo fluxo é utilizado para extrair informações do arquivo caso este seja um espelho.

O fluxo para as guias é o mais simples, onde do arquivo de texto são extraídas duas informações: a data de validade e o valor total da guia. A data de validade é extraída do texto utilizando-se uma *action* chamada “calculate” que permite ao usuário utilizar diferentes funções para manipulação de variáveis. A principal função utilizada para encontrar a data foi a chamada “InStr”, que retorna a posição de uma palavra dentro de um texto. Dessa forma, a palavra passada foi “Data” e fazendo-se outras pequenas manipulações, foi possível obter a data de validade da guia. A segunda informação, o valor total da guia, foi extraída através de uma expressão regular. A expressão regular utilizada tem a seguinte estrutura: (?<ValorTotal>(\d*[.]\d*[,][0-9]{2}|\d*[,][0-9]{2})). Esta expressão utiliza o metacactere “ou” para retornar ou um valor que possua ponto e vírgula ou um valor que só possua vírgula.

O segundo fluxo começa com uma verificação para saber se a página está em branco. Em caso positivo, nada deve ser feito e o fluxo deve seguir. Em caso negativo, o fluxo segue para encontrar os seguintes valores: o valor devido pela empresa, o CNPJ e o PIS. O valor devido pela empresa é encontrado da mesma forma que a data foi encontrada no primeiro fluxo: utilizando-se a função de texto “InStr” dentro da *action* “calculate”. Para encontrar os valores de CNPJ e PIS foram utilizadas expressões regulares. Para o CNPJ, tem-se a seguinte expressão: (?<CNPJ>([0-9]{2}[0-9]{3}[0-9]{3}[0-9]{4}[0-9]{2})). Esta expressão retorna para o grupo “CNPJ” um valor com a estrutura típica de um CNPJ, separado por pontos, traços e barra. Já, a expressão para o PIS é mais simples, pois, aparece no arquivo somente com números. Dessa forma, para o PIS, tem-se a seguinte expressão regular: (?<PIS>([0-9]{11})). Após coletar as informações da página do arquivo e popular a planilha, o robô filtra os CNPJ’s de mesmo valor e os coloca em uma *collection* chamada “CNPJ’s” que será usada posteriormente, e reinicia o *loop* para todas as páginas que restarem. Uma vez que não se tenha mais páginas, o *loop* é encerrado e o robô segue o fluxo para o bloco “Merge Files”.

Figura 39 – Diagrama da página Extract Data from PDF



Fonte: Do Autor

3.2.4.2 Bloco Merge Files

O bloco “Merge Files” inicia com a criação de uma pasta nomeada com o dia, mês e ano correntes para que os arquivos possam ser devidamente alocados e organizados. Após a criação da pasta, é iniciada a lógica do bloco com um *loop* dentro de outro *loop*. O primeiro *loop* passa por todos os elementos da *collection* “CNPJ’s”, criada no bloco “Extract values from PDF and put in the Excel file”. Este *loop* é responsável por separar as páginas que possuírem o mesmo valor de CNPJ e colocá-las em uma nova *collection* chamada “Same CNPJ Files”. O segundo *loop* opera sobre esta nova estrutura de dados e tem a função de mover uma página por vez para uma pasta temporária. O *loop* ainda conta com uma verificação para saber se a página é uma guia GRRF ou um espelho. Caso seja uma guia GRRF, esta além de ser movida para a pasta temporária, deve ser copiada para a pasta de guias GRRF da área de RH e renomeada com o seu nome original e o valor total da guia. Caso a

página não seja a guia GRRF, o *loop* é reiniciado até que se encerrem as páginas de mesmo CNPJ. Uma vez que as páginas de mesmo CNPJ foram movidas para a pasta temporária, o fluxo segue para a execução da action “Merge Files”, responsável por unir todas as páginas da pasta temporária em um único arquivo pdf e enviá-las a pasta criada no início do bloco. O *loop* então recomeça até que não se tenha mais CNPJ’s na collection “CNPJ’s”.

Ao final do *loop* os arquivos pdf e .re baixados no site da ADP eXpert são movidos para a pasta nomeada com a data corrente e a planilha eletrônica é salva com o número da respectiva requisição e data corrente. O fluxo da página então é finalizado e o robô passa a seguir o fluxo da *Main Page*. Na *Main Page* o robô segue para a página “Send Mail” que envia um e-mail para o key-user do processo informando que o item da fila foi executado com sucesso. O e-mail que corresponde a esse item da fila é movido para a pasta de itens processados no outlook e o item é marcado como completo na fila do Blue Prism. Dessa forma, o robô reinicia todas as variáveis globais e seleciona o próximo item da fila para repetir o mesmo processo até que não existam mais itens na fila e o processo é encerrado. A Figura 40 ilustra a estrutura do bloco “Merge Files”.

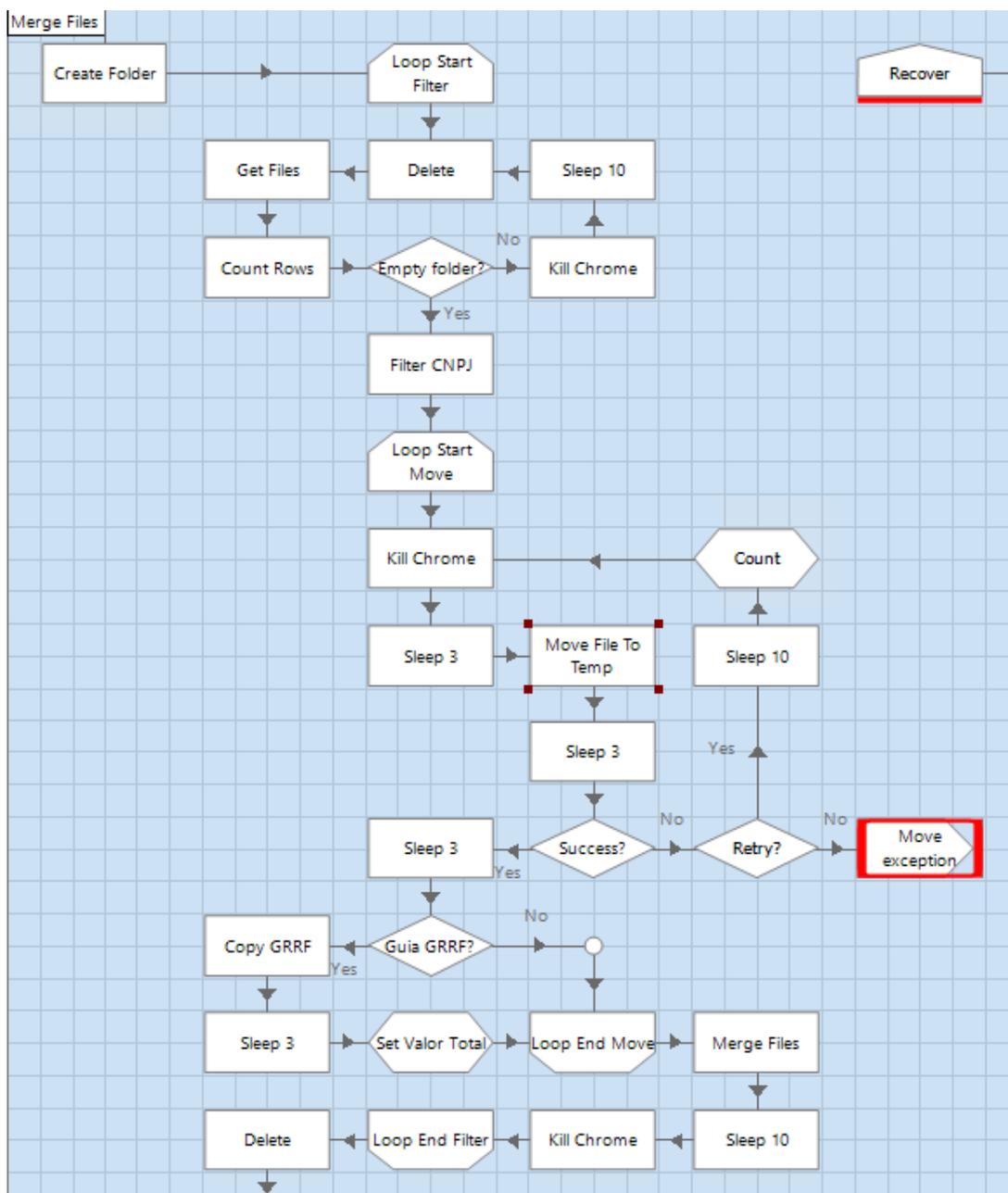
3.2.5 Tratamento de Exceções

O tratamento de exceções é um tópico importante no que diz respeito à automação de processos robóticos. Isto se deve ao fato de que é muito improvável existir um processo que possua somente “caminhos bem-sucedidos”. Portanto, se faz necessário criar caminhos alternativos caso o processo não saia como planejado. Pensando nisso, foram elencados possíveis obstáculos que o robô pudesse encontrar e como o robô deveria lidar com essas situações.

Como visto na Seção 2.2.3.1, um processo pode ter diferentes tipos de exceções. No processo do presente trabalho, as exceções de maior relevância são as *System Exceptions* e as *Business Exceptions*. Quando o download dos arquivos no site da ADP eXpert falhar ou quando não for possível mover o arquivo de uma pasta para outra, erros como esses serão capturados e classificados como *System Exceptions*. O robô então entra no modo *recovery*, conforme ilustrado na Figura 41, e segue para uma decisão que verifica o tipo da exceção. Se for uma exceção interna ou uma *System Exception*, o robô incrementa um contador e tenta realizar a ação novamente. Caso continue dando erro e o contador seja igual a três, o robô executa o chamado “bubbling” que é basicamente, “carregar” a exceção até que chegue na página principal, *Main Page*, e dentro da *Main Page* o fluxo é continuado. Neste caso, o robô carrega as informações da exceção, envia um e-mail para a área do RH informando o motivo da exceção e segue para o próximo item da fila.

O outro tipo de exceção que deve ser tratada com atenção são as *Business Excep-*

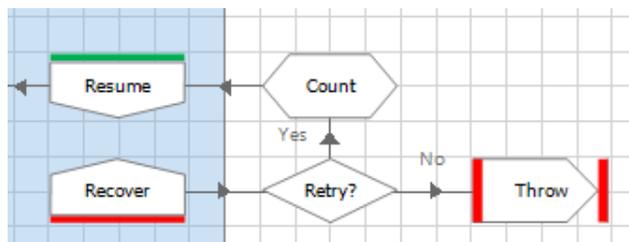
Figura 40 – Diagrama da página Extract Data from PDF



Fonte: Do Autor

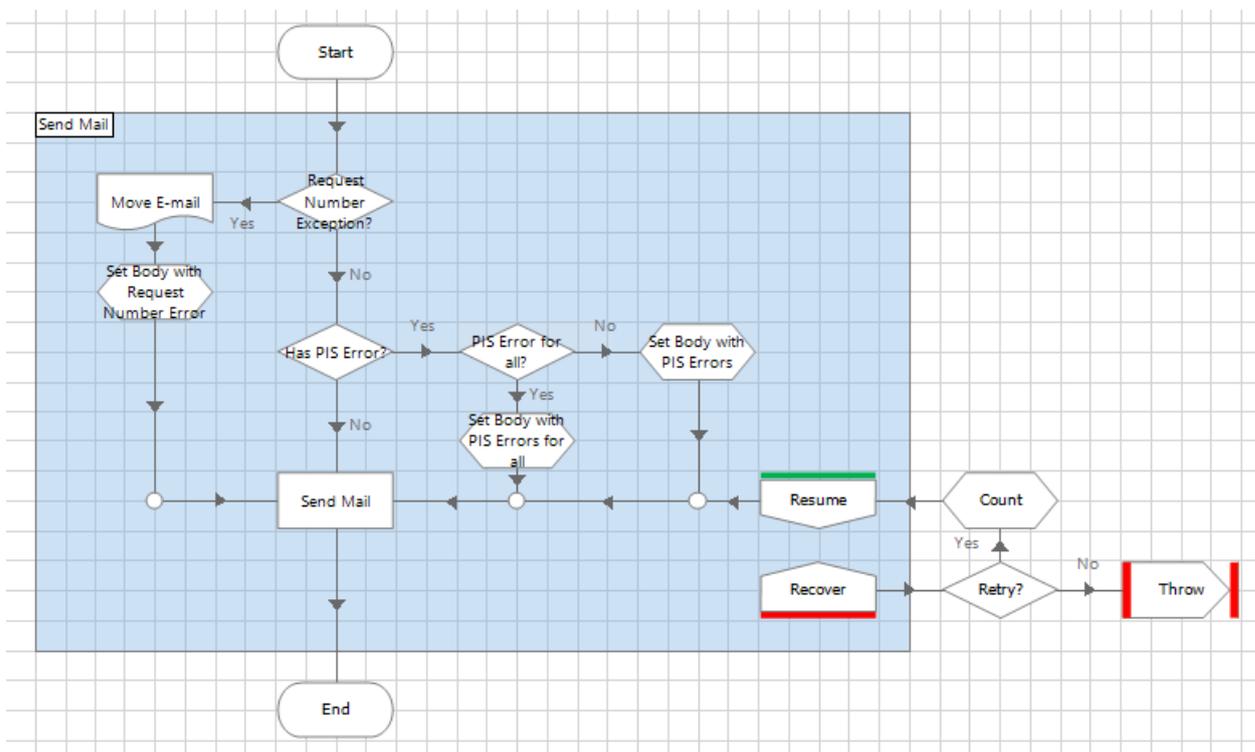
tions. Neste processo, tem-se duas principais *Business Exceptions*: quando a guia GRRF possuir somente uma pessoa e a mesma está sem o PIS na planilha auxiliar; e quando a requisição não estiver disponível no site da ADP eXpert. Em ambos os casos, como a exceção se trata de uma falha na regra de negócio, não há nada que o robô consiga fazer para voltar a funcionar. Dessa forma, a única ação é enviar um e-mail personalizado para que a área entenda exatamente o motivo de determinada exceção. Isto pode ser visto na Figura 42, que mostra o fluxo da página *Send Mail*.

Figura 41 – Modo Recovery para Exceptions



Fonte: Do Autor

Figura 42 – Diagrama da página Send Mail

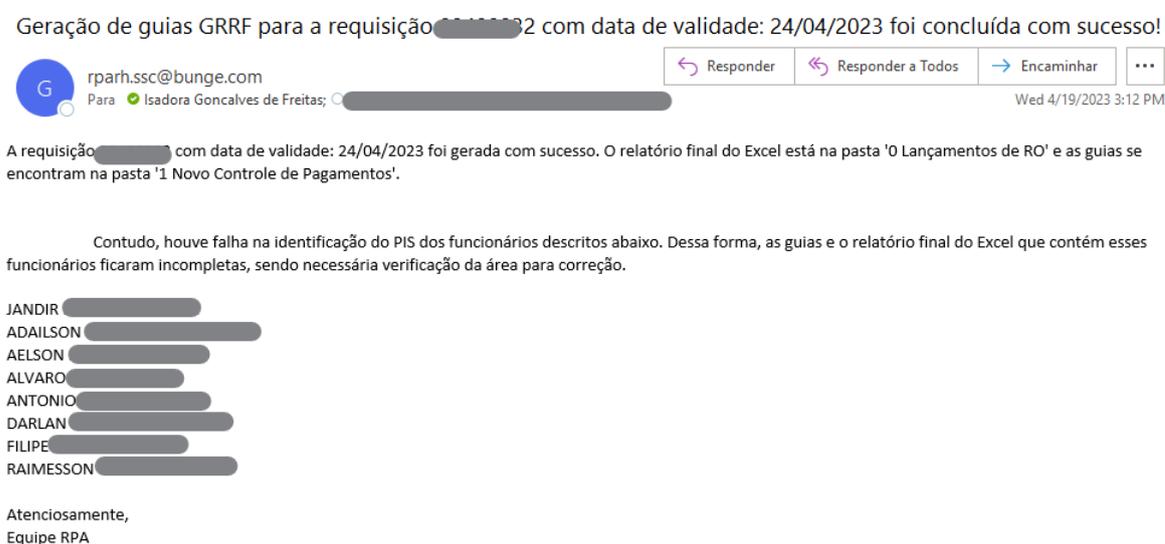


Fonte: Do Autor

A página *Send Mail* inicia seu fluxo com o estágio *decision* para verificar se houve a exceção chamada *Request Number Exception*, que acontece quando o número requisição não está disponível no site. Em caso positivo, o robô move o e-mail para a pasta “Processados” do *Outlook* para que, dessa forma, o item não volte a ser processado em uma nova execução do robô. E um e-mail é enviado informando que a ADP disponibilizou um número de requisição inválido. Caso não seja esse tipo de exceção, o fluxo segue para um novo *decision* para verificar se há *Business Exception* relacionada ao PIS. Em caso negativo, o robô envia um e-mail de sucesso ou outro tipo de exceção, a depender do caminho iniciado na *Main*

Page. Caso a exceção seja relacionada ao PIS, o robô passa por sua última verificação a fim de saber se a exceção do PIS corresponde a uma pessoa ou a mais de uma pessoa. Se nessa exceção o arquivo utilizado contiver apenas uma pessoa, o robô envia um e-mail informando que não foi possível concluir o processo, marca o item como exceção na fila e segue para o próximo item. Porém, se no arquivo utilizado contiver pessoas com o número de PIS na planilha e apenas uma ou algumas pessoas sem o número de PIS, o robô segue o fluxo do processo normalmente e apenas ignora esses funcionários sem o PIS. Ao fim do processo, o robô envia um e-mail informando que o processo foi realizado, contudo, de forma incompleta e será necessária verificação e correção manual da área. O corpo do e-mail pode ser observado na Figura 43 abaixo:

Figura 43 – Corpo do e-mail para exceção do PIS



Fonte: Do Autor

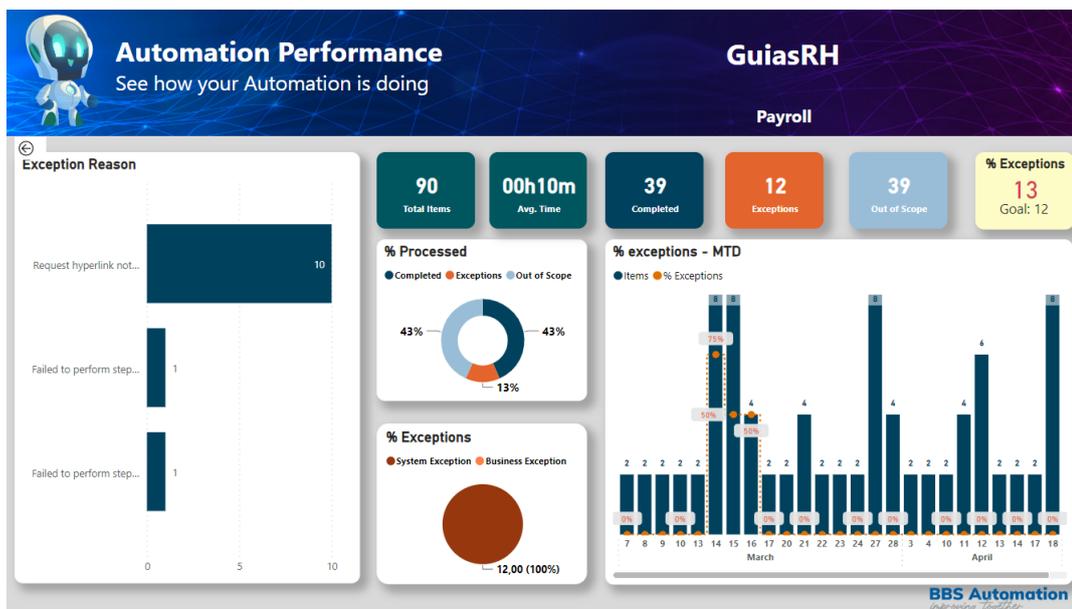
4 RESULTADOS

Neste capítulo, são apresentados os resultados obtidos com a implementação do robô na empresa. O capítulo é dividido em três seções que visam apresentar os principais indicadores da funcionalidade e sucesso do robô, sendo eles: o nível de exceção que o robô gerou, o tempo de execução do robô e os custos reduzidos com a implementação do robô.

4.1 NÍVEL DE EXCEÇÃO

No RPA, um indicador fundamental para qualquer tipo de robô é o nível de exceção. Dessa forma, faz-se necessário realizar um acompanhamento desse nível, a fim corrigir os erros e manter o robô funcional. Para tal, foi desenvolvido pelo time de RPA da Bunge, um *dashboard* no software *Power BI*. Neste *dashboard* é possível acompanhar toda a performance do robô, como: total de execuções, tempo médio de execução, tipos de exceção e porcentagem de exceções. A Figura 44 ilustra todas essas informações correspondentes ao robô do presente trabalho. No *dashboard*, é possível observar que, no mês de março de 2023, houve somente *system exceptions*. Essas exceções foram causadas pelo fato do site da ADP ter atualizado seu *layout*, sendo necessário refazer o mapeamento do mesmo, pois, o robô tentava seguir o fluxo de execução com informações do *layout* antigo e como não as encontrava, retornava exceções.

Figura 44 – Dashboard do robô



Fonte: Do Autor

4.2 TEMPO DE EXECUÇÃO

O tempo de execução de um robô também é um indicador importante para o RPA, pois, dentre outros indicadores, ajuda a justificar o uso de um robô no lugar de um operador humano. Para realizar a comparação de tempo de execução da tarefa que foi automatizada, a área de RH realizou um cálculo baseado no número de guias GRRF que os operadores humanos recebem diariamente. Como o número de guias GRRF em um arquivo varia constantemente, pois, depende de quantos funcionários rescindiriam contrato com a empresa, a área de RH utilizou para o cálculo de execução da tarefa, o número de 20 guias GRRF em um arquivo. Portanto, para um arquivo com 20 guias GRRF, o operador humano necessita de seis minutos para realizar toda a tarefa. Já, o robô, para a mesma quantidade de guias GRRF, precisou de cerca de três minutos. Com uma simples regra de três, nota-se que o robô reduziu o tempo de execução da tarefa pela metade. A Figura 45 ilustra os variados tempos de execução do robô:

Figura 45 – Fila do robô

Item Key	Status	Tags	Resource	Created	Last Updated	Completed	Total Work Time
07890968 - 20/04/2022	Out of Scope	Business Exception	HBWRPAP016	20/04/2023 10:02:42	20/04/2023 10:09:50		00:14
07890968 - 20/04/2022	Completed		HBWRPAP016	20/04/2023 10:02:42	20/04/2023 10:09:36	20/04/2023 10:09:36	06:54
08754321 - 19/04/2022	Out of Scope	Business Exception	HBWRPAP016	19/04/2023 15:02:47	19/04/2023 15:17:32		00:13
08754321 - 19/04/2022	Completed		HBWRPAP016	19/04/2023 15:02:47	19/04/2023 15:17:19	19/04/2023 15:17:19	04:42
08400832 - 19/04/2022	Out of Scope	Business Exception	HBWRPAP016	19/04/2023 15:02:47	19/04/2023 15:12:37		00:13
08400832 - 19/04/2022	Completed		HBWRPAP016	19/04/2023 15:02:47	19/04/2023 15:12:24	19/04/2023 15:12:24	09:37
08593379 - 18/04/2022	Out of Scope	Business Exception	HBWRPAP016	18/04/2023 15:02:50	18/04/2023 15:16:14		00:13
08593379 - 18/04/2022	Completed		HBWRPAP016	18/04/2023 15:02:50	18/04/2023 15:16:00	18/04/2023 15:16:00	03:50
08972494 - 18/04/2022	Out of Scope	Business Exception	HBWRPAP016	18/04/2023 15:02:50	18/04/2023 15:12:10		00:13
08972494 - 18/04/2022	Completed		HBWRPAP016	18/04/2023 15:02:50	18/04/2023 15:11:57	18/04/2023 15:11:57	03:38
08983073 - 18/04/2022	Out of Scope	Business Exception	HBWRPAP016	18/04/2023 15:02:50	18/04/2023 15:08:19		00:14
08983073 - 18/04/2022	Completed		HBWRPAP016	18/04/2023 15:02:50	18/04/2023 15:08:05		05:15

Fonte: Do Autor

4.3 ANÁLISE DE CUSTOS

Outro indicador importante para justificar o uso de um robô é o custo que o mesmo pôde reduzir. Para realizar este cálculo, a empresa utiliza o *FTE (Full-Time Equivalent)*. O FTE corresponde à unidade de medida que equivale à carga horária um colaborador. (CFI, 2023)

Portanto, o FTE calcula quantas horas de trabalho uma tarefa totalmente automatizada equivale. Para realizar este cálculo são necessários os seguintes valores: volume anual da tarefa (V), tempo em minutos de execução da tarefa (T) e jornada de trabalho anual (J) do colaborador. O FTE é calculado utilizando-se a Equação (1). Os valores para a realização do cálculo foram fornecidos pela área de RH, sendo eles: $V = 5280$ guias anuais, considerando que a tarefa é realizada diariamente e que por tarefa são processadas 20 guias; $T = 6$ minutos, para execução da tarefa; e, $J = 2112$ horas, para jornada de

trabalho anual. Transformando o tempo de execução da tarefa em horas, tem-se o seguinte resultado:

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

$$FTE = V \frac{\frac{6}{60}}{2112} \quad (2)$$

$$FTE = 0.25 \quad (3)$$

Sabendo-se que 1 FTE equivale a 2112 horas de trabalho, 0.25 corresponde a 528 horas de trabalho. Portanto, a automatização da tarefa possibilitou à empresa a economia de 528 horas trabalhadas no ano. Essas horas equivalem a 25% da carga horária anual do colaborador que podem ser destinadas a outras tarefas da empresa.

5 CONCLUSÃO

O presente trabalho apresentou uma solução utilizando RPA para automatizar uma tarefa realizada manualmente pela equipe de Recursos Humanos da empresa Bunge. A tarefa é a extração de dados de ex-colaboradores advindas de guias GRRF que são enviadas pelo sistema ADP. A ADP enviava por e-mail o número de requisição a ser consultado em seu site. O operador humano deveria utilizar este número para acessar o sistema da ADP e baixar os arquivos para realizar a extração dos dados e inserção dos mesmos em uma planilha, e após, realizar a separação das guias para enviar ao departamento financeiro. Contudo, o arquivo com as guias era recebido com frequência de forma desordenada, fazendo com que o operador humano empregasse mais tempo na tarefa, pois, além realizar a extração e inserção dos dados, deveria ordenar as guias para tal. Neste contexto, foi proposto o uso de um software de RPA, chamado Blue Prism, para automatizar esta tarefa.

Ao longo do trabalho, foi visto que a solução proposta foi bem-sucedida, pois, após o estudo das ferramentas e sua posterior utilização, a tarefa pôde ser realizada com sucesso. Além disso, o uso do RPA reduziu o tempo de execução da tarefa em 50%. Dessa forma, utilizando-se o cálculo de FTE, observou-se que a automação da tarefa economizou 44 horas mensais do operador humano, o que equivale a 5 dias úteis. Com essa economia de horas, além de reduzir o estresse de um funcionário ao lidar com uma tarefa repetitiva e muitas vezes longa, possibilitou ao funcionário a realização de outras funções dentro da empresa.

Para trabalhos futuros, pretende-se automatizar a parte do processo em que a área do RH disponibiliza o relatório de PIS de forma mensal, visto que é possível extrair as informações contidas neste relatório através de algumas manipulações no site da ADP eXpert.

REFERÊNCIAS

ANYWHERE, Automation. **O que são Bots de software?** [S.l.], 2023. Disponível em: <https://www.automationanywhere.com/br/rpa/software-bots>. Acesso em: 4 mai. 2023.

BUNGE. **Somos Bunge | Nossa História.** [S.l.], 2023. Disponível em: <https://www.bunge.com.br/en/Somos-Bunge/Nossa-Historia>. Acesso em: 6 jun. 2023.

CASSANDRAS, Christos G.; LAFORTUNE, Stéphane. **Introduction to Discrete Event Systems.** 2. ed. [S.l.]: Springer, 2008.

CFI. **Full Time Equivalent (FTE).** [S.l.], 2023. Disponível em: <https://corporatefinanceinstitute.com/resources/management/full-time-equivalent-fte/>. Acesso em: 11 jun. 2023.

DEVFURIA. **Expressões Regulares: Metacaracteres.** [S.l.], 2022. Disponível em: <http://devfuria.com.br/regex/metacaracteres/>. Acesso em: 11 nov. 2022.

DOCS, SS&C Blue Prism |. **Blue Prism® Foundation Training.** [S.l.], 2023. Disponível em: <https://blue-prism.docebosaa.com/learn/course/180/Blue%5C%2520Prism%5C%25C2%5C%25AE%5C%2520Foundation%5C%2520Training>. Acesso em: 19 mai. 2023.

DOCS, SS&C Blue Prism |. **Objetos.** [S.l.], 2023. Disponível em: <https://bpdocs.blueprism.com/bp-6-10/es-la/helpBusinessObjects.htm>. Acesso em: 27 abr. 2023.

DOCS, SS&C Blue Prism |. **Queue management.** [S.l.], 2023. Disponível em: <https://bpdocs.blueprism.com/bp-6-7/en-us/control-queues.html>. Acesso em: 27 abr. 2023.

DOCS, SS&C Blue Prism |. **Regular Expressions.** [S.l.], 2022. Disponível em: <https://bpdocs.blueprism.com/bp-7-1/en-us/helpRegularExpressions.htm>. Acesso em: 14 out. 2022.

FITZGERALD, Michael. **Introdução às Expressões Regulares.** 1. ed. São Paulo: Novatec, 2012.

FRIEDL, Jeffrey E.F. **Mastering Regular Expressions.** 3. ed. [S.l.]: O'Reilly, 2016.

GARTNER. **Magic Quadrant for Robotic Process Automation**. [S.l.], 2022.

Disponível em:

<https://www.gartner.com/doc/reprints?id=1-2AOPUPBE&ct=220727&st=sb>. Acesso em: 7 jun. 2023.

GOYVAERTS, Jan. **Regular Expressions: The Complete Tutorial**. 1. ed. [S.l.]: Regular-Expressions.info, 2007.

IBM. **Metacacteres**. [S.l.], 2022. Disponível em:

<https://www.ibm.com/docs/pt-br/netcoolomnibus/8.1?topic=library-metacharacters>. Acesso em: 10 jan. 2023.

JARGAS, Aurelio Marinho. **Expressões Regulares**. 4. ed. [S.l.]: Novatec, 2016.

KROLL, Christian *et al.* **Robotic Process Automation: Robots conquer business processes in back offices**. 1. ed. [S.l.]: Capgemini Consulting, 2016.

SIMPLILEARN. **What Is Blue Prism: Architecture, Components, Features and More**. [S.l.], 2023. Disponível em:

<https://www.simplilearn.com/tutorials/rpa-tutorial/what-is-blue-prism>. Acesso em: 2 mar. 2023.

TAULLI, Tom. **The Robotic Process Automation Handbook: A Guide to Implementing RPA Systems**. 1. ed. [S.l.]: Apress, 2020.

WHITE, Stephen A.; MIERS, Derek. **BPMN Modeling and Reference Guide: Understanding and using BPMN**. 1. ed. [S.l.]: Future Strategies, 2008.

YING, Lim Mei. **Robotic Process Automation with Blue Prism Quick StartGuide**. 1. ed. Birmingham: Packt, 2018.