

Universidade Federal de Santa Catarina
Centro Tecnológico, de Ciências Exatas e Educação
Departamento de Engenharia de
Controle e Automação e Computação



Eduardo Seiti Sonohara

Monitoramento de processos automatizados utilizando RPA

Blumenau

2021

Eduardo Seiti Sonohara

Monitoramento de processos automatizados utilizando RPA

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Santa Catarina como parte dos requisitos necessários para a obtenção do Título de Engenheiro de Controle e Automação.
Orientador: Prof. Dr. Maiquel de Brito

Universidade Federal de Santa Catarina
Centro Tecnológico, de Ciências Exatas e Educação
Departamento de Engenharia de
Controle e Automação e Computação

Blumenau
2021

Eduardo Seiti Sonohara

Monitoramento de processos automatizados utilizando RPA

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Engenheiro de Controle e Automação.

Comissão Examinadora

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

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

Prof. Dr. Mauri Ferrandin
Universidade Federal de Santa Catarina

Blumenau, 29 de setembro de 2021

Agradecimentos

Gostaria de agradecer meus pais, Nelson e Marina, e minha irmã Patricia que me apoiaram durante todo o período de graduação e por toda a educação que me ensinaram e todo o suporte e amor nos momentos que necessitei.

Agradeço também a todos meus professores que me ajudaram e me ensinaram durante toda a graduação para me capacitar profissionalmente nas diversas áreas da engenharia. Em especial, agradeço meu orientador Maiquel por todo o ensinamento, auxílio, suporte, conselho e por ter me aceitado como orientando.

Também agradeço a todos meus amigos que fiz durante esse período e que me motivaram nessa jornada.

Por último, agradeço à Universidade Federal de Santa Catarina por ter me proporcionado o conhecimento para seguir minha jornada profissionalmente.

*"Toda a nossa ciência, comparada com a realidade, é primitiva e infantil - e, no entanto,
é a coisa mais preciosa que temos."
(Albert Einstein)*

Resumo

Atualmente, nas empresas, há vários processos de negócios que geram uma grande quantidade de dados para serem manipulados e processados repetidamente. Geralmente, esses processos são realizados de maneira manual por funcionários que utilizam um computador. Porém essa metodologia de trabalho tornou-se pouco eficiente, e até mesmo pode causar problemas ergonômicos no funcionário em virtude de movimentos repetitivos. Visando melhorar esse cenário e o desempenho nos processos, surgiu a automação de processos robóticos (RPA - Robotic Process Automation). RPA é uma tecnologia capaz de executar processos repetitivos e massivos de forma automatizada utilizando um software de computador. No entanto, os processos robóticos de RPA estão sujeitos a falhas e, por isso, é necessário monitorar o seu funcionamento. Enquanto este monitoramento é usualmente realizado por um operador humano, considera-se adequado torná-lo também automatizado. Assim, nesse trabalho, foi desenvolvido uma aplicação de automação RPA para realizar o processo de monitoramento de processos automatizados de forma automática. Basicamente a solução de automação desenvolvida identifica erros ou falhas que ocorreram nos processos automatizados e informa às partes interessadas. Este processo, anteriormente, era realizado de forma manual. Os resultados obtidos se mostraram satisfatórios, pois com a implementação da solução desenvolvida houve uma redução de 80% no tempo de execução, gerando redução de custo e aumento da eficiência.

Palavras-Chave: 1.Automação de Processos Robóticos. 2.Monitoramento RPA. 3.Blue

Prism.

Abstract

Companies have currently several business processes that generate a large amount of data to be repeatedly manipulated. These processes are performed manually by employees using computer systems. However, this work methodology ended up becoming inefficient and even causing ergonomic problems for the employee due to repetitive movements. Aiming to improve this scenario and the performance in the processes, the automation of robotic processes (RPA - Robotic Process Automation) emerged. RPA is a technology capable of performing repetitive and massive processes in an automated way using computer software. This work was developed using the Blue Prism software to carry out the process of monitoring automated processes automatically. Basically, the developed automation solution identifies errors or failures that occurred in the automated processes and informs the parties, previously performed manually. The results were satisfactory because with the implementation of the developed solution there was a gain in efficiency, cost reduction and also an 80% reduction in process execution time.

Keywords: 1.Robotic Process Automation. 2.RPA Monitoring. 3.Blue Prism.

Lista de figuras

Figura 1 – Previsões da receita mundial de software RPA 2020 e 2021 (milhões de dólares americanos)	13
Figura 2 – Quadrante Mágico da Gartner RPA de 2021	16
Figura 3 – Características do Blue Prism	17
Figura 4 – Arquitetura Blue Prism	18
Figura 5 – Exemplo de dados em forma hierárquica - Conta Corrente	20
Figura 6 – Exemplo de dados em forma de rede - Conta Corrente	21
Figura 7 – Exemplo de dados em forma orientado a objetos - Conta Corrente	21
Figura 8 – Exemplo modelo conceitual	22
Figura 9 – Exemplo modelo lógico	22
Figura 10 – Exemplo de Generalização/Especialização	24
Figura 11 – Recursos Blue Prism	30
Figura 12 – Sessão dos Processos Blue Prism	31
Figura 13 – Agendamentos das automações Blue Prism	32
Figura 14 – Marcado de completo e falha Blue Prism	32
Figura 15 – Tabela Process	34
Figura 16 – Tabela Status	35
Figura 17 – Tabela Resource	35
Figura 18 – Tabela User	35
Figura 19 – Tabela Session	36
Figura 20 – Diagrama de Entidade-Relacionamento	37
Figura 21 – Telegram <i>NewBot</i>	41
Figura 22 – Telegram <i>Chat ID</i>	42
Figura 23 – Telegram método <i>sendMessage</i>	42
Figura 24 – Mensagem quando ocorrer o término da execução do robô	43
Figura 25 – Mensagem quando o robô não iniciar no horário programado	43
Figura 26 – Mensagem quando ocorrer um problema nos recursos	43
Figura 27 – Projeto Blue Prism	45
Figura 28 – Informações execuções Blue Prism	47
Figura 29 – Tempo Médio de Execução	47
Figura 30 – Ação Set Connection Blue Prism	51
Figura 31 – Ação GetCollection Blue Prism	52

Lista de tabelas

Tabela 1 – Exemplo de dados em forma relacional - Conta Corrente	20
Tabela 2 – Exemplo de uma tabela não normalizada	24
Tabela 3 – Tabela Proj - 1FN	24
Tabela 4 – Tabela ProjEmp - 1FN	25
Tabela 5 – Tabela 2FN	26
Tabela 6 – Tabela 3FN	27
Tabela 7 – Notas dos Estudantes	28
Tabela 8 – Notas dos Estudantes	28
Tabela 9 – Idade dos Estudantes	29
Tabela 10 – Junção das Tabelas Notas dos Estudantes e Idade dos Estudantes . . .	29

Sumário

1	INTRODUÇÃO	11
1.1	Objetivo	12
1.1.1	Objetivos Específicos	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	Automação Robótica de Processos (RPA)	13
2.1.1	Tipos de RPA	14
2.1.2	Ferramentas RPA	16
2.2	Banco de dados	19
2.2.1	Normalização	23
2.2.2	Linguagem SQL	25
3	DESENVOLVIMENTO	30
3.1	Monitoramento de RPA não automatizado	30
3.2	Metodologia de solução de automação	32
3.3	Projeto e implementação da solução de automação	33
3.3.1	Identificação das informações necessárias para o monitoramento	33
3.3.2	Mapeamento das informações	34
3.3.3	Desenvolvimento de um mecanismo para obtenção das infor- mações	39
3.3.4	Compartilhamento das informações relevantes	40
3.3.5	Implementação do robô	44
4	RESULTADOS	46
4.1	Tempo de execução	46
4.2	Identificação de falhas	47
4.3	Custo de execução	48
5	CONCLUSÃO	49
A	APÊNDICES	50
A.1	Mecanismo para obtenção das informações	50
	REFERÊNCIAS BIBLIOGRÁFICAS	53

1 Introdução

Nas empresas, hoje em dia, existem vários processos de negócios que geram uma grande quantidade de dados e que envolvem a manipulação e extração de informações contidas em boletos, notas fiscais, bancos de dados entre outros. O processamento dessas informações normalmente é realizado por mão de obra humana com o auxílio de um computador. Entretanto, esta atividade manual está sujeita a problemas como grande quantidade de erros, baixo rendimento, além de desgaste mental e geração de lesões no operador em virtude de movimentos repetitivos na operação do computador [1].

Visando melhorar esse cenário e o desempenho nos processos de negócios, um possível recurso para tratar os problemas mencionados anteriormente é a utilização de sistema robótico de automação de processos (RPA - Robotic Process Automation). Segundo o IEEE (Instituto de Engenheiros Eletricistas e Eletrônicos), RPA pode ser definido como uma tecnologia que utiliza regras predefinidas para a execução autônoma de uma combinação de tarefas, processos ou atividades em um ou mais sistemas de software para fornecer um resultado ou serviço que reproduz o trabalho de um ser humano [2]. Além disso, a implementação de um sistema RPA para automatizar tarefas repetitivas, exaustivas e monótonas pode trazer benefícios como a redução dos custos operacionais e o aumento de performance [3]. Apesar de a ferramenta não ser muito popular no mercado, algumas companhias que adotaram RPA conseguiram bons resultados, até mesmo reduzindo os custos operacionais [4].

Apesar de muitas automações eliminarem a necessidade de um operador humano, elas precisam da presença de um monitor, usualmente humano, para supervisionar o seu correto funcionamento. Problemas podem ocorrer se o monitor estiver realizando outras atividades ou até mesmo ausente e, assim, não detectar as falhas. As principais fontes de problema em uma automação de RPA são: instabilidade no sistema, falhas de desenvolvimento, alterações de informações que não foram comunicadas à equipe de RPA, agendamentos que não foram iniciados devido um recurso estar ocupado ou desligado, que resultam em seu término, impactando a área. Esses problemas tornam-se críticos em dias e horários em que o operador está menos disponível, como finais de semana, feriados e madrugadas.

Para evitar os problemas mencionados anteriormente, e considerando que o monitor realiza o monitoramento de forma manual, identificou-se a necessidade de automatizar este processo. A automação do monitoramento de RPA requer o desenvolvimento de métodos, bem como a implementação de mecanismos, para identificação, obtenção e tratamento de dados a respeito de execuções, de agendamentos de processos que não iniciaram e de falhas de recursos. Especificamente, este trabalho apresenta o desenvolvimento destes elementos

em um projeto para realizar o monitoramento automatizado de RPA, sendo desenvolvido na empresa Bunge - Gaspar/SC [5]. A Bunge é uma empresa global com mais de 100 anos, uma das principais no ramo de agronegócio e alimentos, sendo a maior exportadora em agronegócio [6]. Atualmente, a unidade de Gaspar é organizada a partir de um modelo de Centro de Serviços Compartilhados (CSC), que engloba os serviços: PTP (*Purchase to Pay*), RTP (*Requisition to Pay*), OTC (*Order to Cash*), Fiscal, Contabilidade, Recursos Humanos, Tecnologia da Informação, *Facilities* e Jurídico. Atualmente o time possui mais de 100 automações desenvolvidas, sendo 85 automações em funcionamento e apenas 8 recursos disponíveis, os recursos são máquinas virtuais, onde as automações são executadas.

As principais vantagens esperadas com a automação do monitoramento de RPA são: redução do tempo de processamento e do custo operacional, maior segurança, precisão, disponibilidade da equipe para tarefas mais importantes e menos exaustivas e, consequentemente, maior rendimento.

1.1 Objetivo

O objetivo deste trabalho é desenvolver uma aplicação de automação capaz de realizar o monitoramento de RPA de forma automatizada, informando eventuais problemas às partes interessadas.

1.1.1 Objetivos Específicos

Para alcançar o objetivo são definidos os seguintes objetivos específicos:

- Estudar uma metodologia de solução de automação;
- Projetar uma solução de automação;
- Implementar a automação desenvolvida;
- Analisar e avaliar o desempenho da automação desenvolvida.

2 Fundamentação Teórica

Este capítulo apresenta a fundamentação teórica. A Seção 2.1 trata da automação robótica de processos (RPA) e a Seção 2.2 trata de banco de dados.

2.1 Automação Robótica de Processos (RPA)

A automação robótica de processos (RPA) pode ser definida como uma tecnologia capaz de automatizar atividades que são realizadas por pessoas, de forma manual, como realizar consultas, cadastros, validações, entre outras. Um dos primeiros passos que levaria a criação do RPA foi o aprendizado de máquina (ML), criado em 1959 por Arthur Samuel, um pioneiro na área da inteligência artificial, que na época trabalhava para a empresa de computadores IBM [7]. Com o avanço dos estudos e a progressão da tecnologia, na década de 1990, foi desenvolvido um software de captura de tela e também ocorreram o surgimento de ferramentas de automação de fluxo de trabalho e inteligência artificial. Tudo isso viabilizou a criação da automação de processos robóticos.

Graças a esses desenvolvimentos, no início dos anos 2000 o RPA foi desenvolvido. Porém, somente em 2003, que a empresa Blue Prism lançou o seu primeiro produto completo, na mesma época as empresas UiPath e Automation Anywhere lançaram suas bibliotecas, que eram utilizadas por outras empresas como IBM, Google e Microsoft, que os incorporavam em seus próprios produtos. Essas três empresas foram as pioneiras do mercado e seus produtos são os mais utilizados atualmente.

O principal fator para a utilização do RPA é a capacidade de melhorar a qualidade, velocidade e produtividade do processo, reduzindo seus custos. Segundo Gartner, a receita mundial de software RPA chegou a US \$ 1,9 bilhão em 2020 [8], com um aumento de aproximadamente 38 % em relação a 2019 [9], e superando as expectativas conforme a Figura 1.

	2019	2020	2021
Receita (\$ M)	1.411,1	1.579,5	1.888,1
Crescimento (%)	62,93	11,94	19,53

Fonte: Gartner (setembro de 2020)

Figura 1 – Previsões da receita mundial de software RPA 2020 e 2021 (milhões de dólares americanos)

Além disso, por conta da pandemia devido à COVID-19 e da recessão que se seguiu, aumentou o interesse empresarial em RPA. Gartner prevê que 90% das grandes empresas mundiais, terão adotado o RPA de alguma forma até 2022 [9], pelo fato de que buscam a capacitação digital nos processos de negócios críticos, por meio de resiliência e escalabilidade, enquanto ajustam o trabalho humano e o esforço manual.

A automação robótica de processos (RPA) é uma tecnologia voltada a automatizar processos de negócio para lidar automaticamente com tarefas operacionais, altamente estruturadas, rotineiras e repetitivas, orientadas por regras, eventos ou horários, em que um robô pode assumir o controle e obter a tarefa concluída. O objetivo principal de RPA é imitar as mesmas ações “manuais” realizados por um operador humano, ao fazer uma série de combinações de interações com interface do usuário. Basicamente a RPA pode realizar qualquer tarefa baseada em regras, a partir do mapeamento do processo, para que o “robô” implementado em software possa executá-lo. Possíveis tarefas automatizadas incluem desde copiar e colar arquivos ou até mesmo realizar a cobrança de clientes, atualização de perfis de clientes e funcionários, realizar cálculos complexos, gerenciar banco de talentos de RH, entre outros.

Além disso, as soluções de RPA podem ser executadas em um computador de mesa, com capacidade limitada, ou ainda em servidores empresariais, capazes de suportarem múltiplas tarefas executadas ao mesmo tempo, atendendo as exigências de segurança da empresa. Também há possibilidade de integração com outras ferramentas, como por exemplo plataformas de BPM (Business Process Management) e inteligência artificial.

2.1.1 Tipos de RPA

A tecnologia RPA pode ser classificada em duas categorias: assistida e desassistida. Na assistida, os robôs operam lado a lado com o usuário. Ou seja, as tarefas automatizadas que o robô realiza acontecem no mesmo ambiente em que usuário opera. A principal vantagem da execução por RPA assistida é que o usuário está no controle, decidindo quando o robô deve executar e intervindo quando necessário, conseguindo avaliar quando a automação pode se beneficiar de mudanças ou melhorias [10]. Já na desassistida, os robôs operam em servidores ou em máquinas virtuais, realizando as tarefas que foram programadas de forma contínua sem a intervenção humana. A principal vantagem da execução desassistida de RPA é que remove-se completamente certas tarefas dos usuário, simplificando significativamente os processos de negócios e seus ambientes de trabalho [10].

Alguns fornecedores de RPA dão suporte apenas a uma implementação de tecnologia RPA desassistida ou assistida, mas há fornecedores que dão suporte a ambos os modelos de implementação, de forma combinada.

Além disso, para realizar a implementação de um sistema de RPA, deve ser levado em

consideração alguns pontos para verificar se a automação irá gerar algum ganho para a empresa:

1) Volumetria de processo: O número de repetições que a automação deverá realizar a mesma atividade. Dependendo da volumetria não é viável, pois não haverá redução de custo.

2) Redução do Equivalente a tempo integral (FTE - *full time equivalent*): FTE é o cálculo da média de horas úteis que os profissionais se dedicam às atividades da organização, que pode ser utilizado para realizar uma comparação da redução de FTE com uma solução de automação [11]. Por exemplo um funcionário de uma empresa que trabalhou em tempo integral, 44 horas, por semana possui 1 FTE, enquanto um trabalhador que trabalha 22 horas semanais possui 0,5 FTE. Portanto o cálculo é realizado com a quantidade de horas com que um trabalhador demora para realizar uma determinada atividade, e estimar quanto tempo a automação levará para concluir a mesma atividade. Logo a redução de tempo por atividade realizada pela automação multiplicado pela quantidade de atividades realizadas, fornece a redução do FTE.

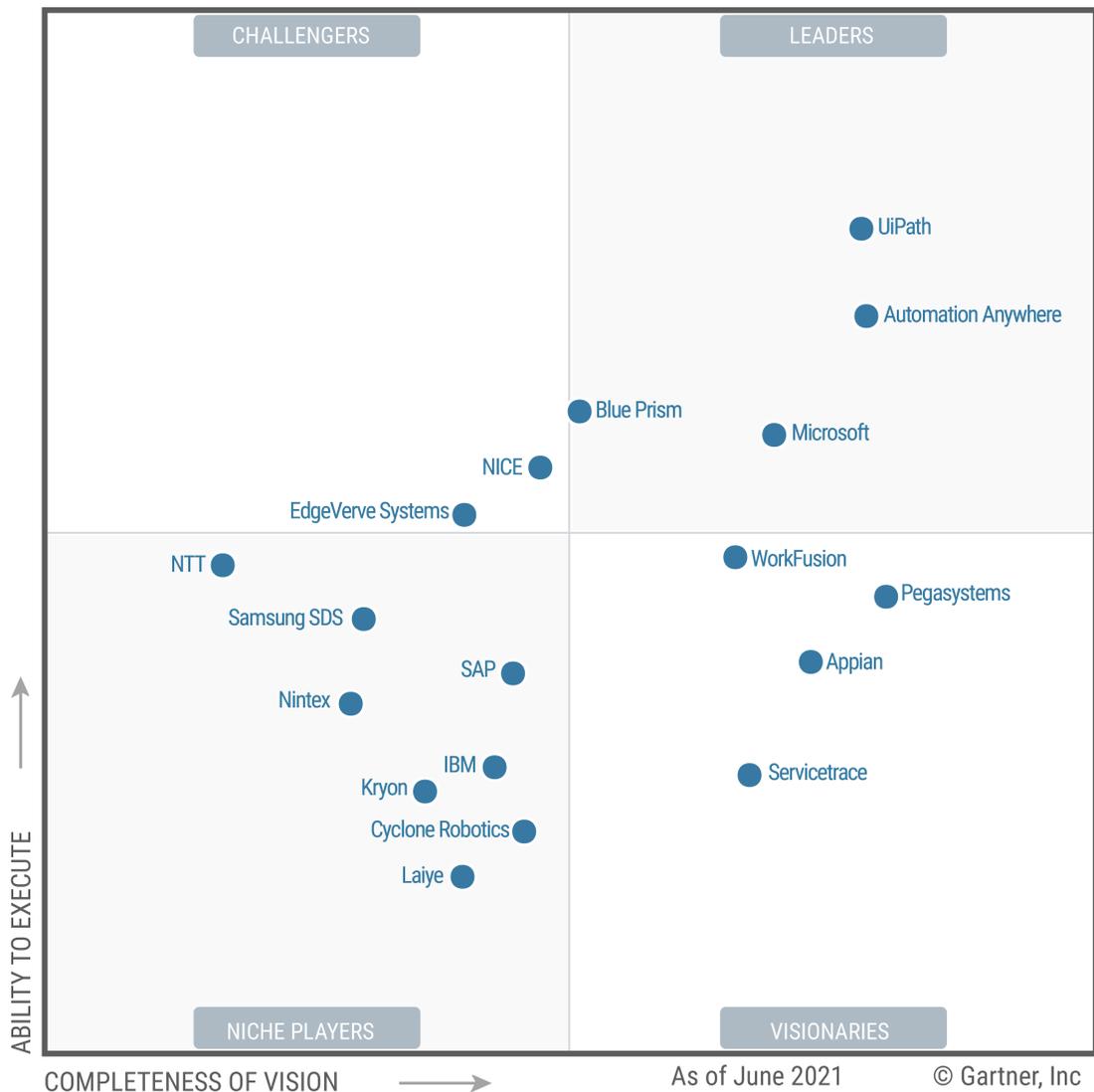
3) Complexidade: Processos muito complexos podem dificultar ou até mesmo impossibilitar a implementação de soluções de RPA

As principais vantagens de uma solução RPA são:

- Ganhos de produtividade rápidos em semanas ou alguns meses, ou seja, cortes de custos;
- Os custos iniciais de investimento e as taxas de licença são pequenos e podem ser calculados de forma confiável;
- Adequado como uma solução tática de redução de custos provisória se a solução técnica estratégica ainda é muito cara;
- Os robôs podem trabalhar 24 horas por dia, 7 dias por semana;
- Poucas mudanças de processo são necessárias, embora a introdução do RPA possa desencadear a implementação de melhorias;
- Nenhuma ou mínima necessidade de mudanças no aplicativo externo, como Excel, sistema integrado de gestão empresarial (ERP), entre outros;
- É escalonável e se beneficia de economias de escala;
- Produção de qualidade melhorada em comparação com trabalhadores humanos, ou seja, menor taxa de falha e riscos.

2.1.2 Ferramentas RPA

Atualmente, as principais ferramentas de automação robótica de processos podem ser visualizadas através do quadrante mágico, criado por Gartner [12], que é uma representação gráfica do mercado de TI, conforme a Figura 2. Dentre tantas ferramentas, o projeto para realizar o monitoramento das automações de forma automatizada, foi implementado através do Blue Prism, por se tratar de uma plataforma que é utilizada pela empresa em que o projeto foi desenvolvido.



Source: Gartner (July 2021)

Figura 2 – Quadrante Mágico da Gartner RPA de 2021

O Blue Prism oferece uma série de ferramentas para a implementação de RPA com a funcionalidade de arrastar e soltar. Além disso, as principais características dessa ferramenta são apresentadas na Figura 3 [13].



Figura 3 – Características do Blue Prism

- Seguro e Preciso: o número de processos que podem ser executados nesta ferramenta é ilimitado. Oferece resultados seguros e precisos para qualquer número de processos que precisem ser automatizados.
- Robusto: tem a capacidade de funcionar apropriadamente em condições anormais e fornece recursos como balanceamento de carga, criptografia de dados e auditoria de ponta a ponta. Assim, toda mudança é auditada e relacionada ao usuário relacionado a ela;
- Escalável e Resiliente: permite escalabilidade com gerenciamento central, ou seja o aumento da produtividade e automatizar um grande número de processos. Assim, todos os processos podem ser automatizados conforme a necessidade e podem ser monitorados de forma centralizada;
- 24/7 Trabalho: a ferramenta foi projetada para funcionar 24 horas por dia sem a necessidade de parar sua execução;
- Analítico: fornece recursos estendidos para configurar painéis de forma que as informações da sessão possam ser redirecionadas para os sistemas de monitoramento;
- Segurança de Dados e Abstração de Dados: todo o processamento é realizado e armazenado em um data center. Isso fornece uma abstração de dados bem definida e segurança de processo;
- Suporte de nuvem: fornece o software como serviço (Saas), disponibilizando o software por meio da internet como um serviço.

- Inteligência de execução: os robôs se conectam a sistemas e reagem dinamicamente às respostas nos dados em vários ambientes.

A Figura 4 ilustra o funcionamento do Blue Prism, onde o processo de negócio automatizado depende de um ou mais objetos de negócios, que correspondem a um único processo a ser realizado. Os objetos de negócios tratam da interação com outras interfaces de aplicativos diferentes. Por exemplo, um objeto pode realizar login em um sistema integrado de gestão empresarial como SAP [14], copiar dados de um campo específico de um determinado aplicativo, selecionar e copiar um valor de uma planilha eletrônica, enviar e-mails de um cliente de e-mail, entre outros.

Os objetos de negócios são programados no ambiente de desenvolvimento Object Studio do Blue Prism. Já os processos de negócios são programados no ambiente Process Studio do Blue Prism, onde descreve-se um conjunto de ações que são realizados pelos objetos nas interfaces das aplicações, conforme a Figura 4.

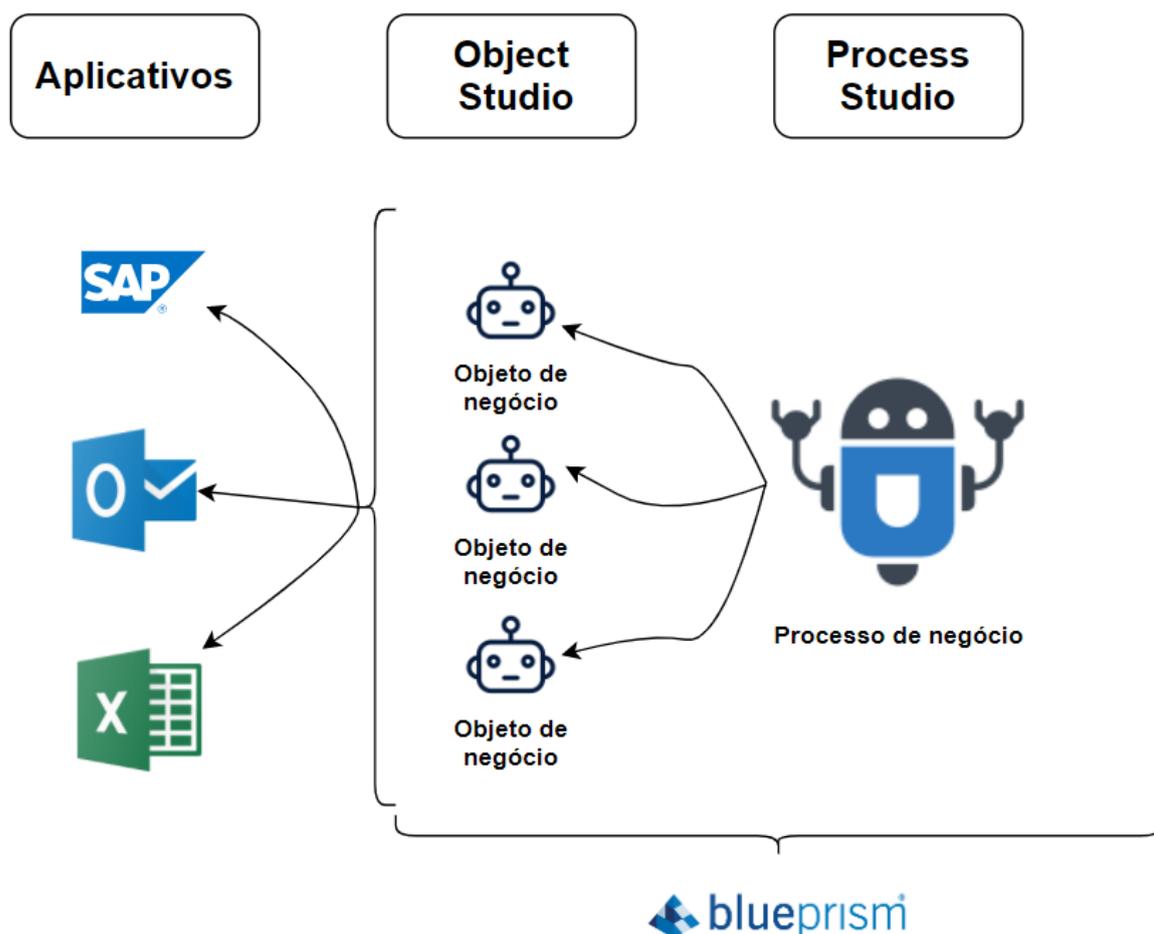


Figura 4 – Arquitetura Blue Prism

Um conceito bastante importante do Blue Prism são as filas de trabalho. Por tratar-se de atividades repetitivas, as filas representam a quantidade de vezes que a automação executará do início ao fim de uma determinada atividade. Ao iniciar-se a execução de uma automação, são carregadas o número de vezes com que essa atividade será executada, esse número de vezes é populado em um fila de trabalho. A fila é responsável por ordenar cada execução de modo que a atividade só é iniciada após o término da execução da atividade anterior. Caso ocorra algum problema quando a atividade é executada, a atividade em questão é finalizada e em seguida a próxima execução da fila é iniciada. Assim, a automação só finaliza sua execução quando não há mais nada na fila, ou quando ocorrer algum problema e a automação finalizar automaticamente.

2.2 Banco de dados

Antigamente, as informações eram registradas em papéis, organizados em arquivos físicos, através de pastas, porém era muito custoso manter esses arquivos organizados. Com o avanço da tecnologia e o surgimento dos computadores, os dados começaram a serem armazenados em arquivos digitais. O termo banco de dados, pode ser definido como um sistema de armazenamento de dados baseados em computador ou seja, um sistema cujo objetivo é registrar e armazenar informações [15].

O primeiro sistema gerenciador de banco de dados (SGBD) surgiu no final da década de 60, com base nos primeiros sistemas de arquivos da época, que armazenavam arquivos em disco, e posteriormente evoluíram para novas estruturas de dados para armazenar informações [15]. Com o passar do tempo, os SGBD sofreram algumas mudanças, e começaram a utilizar diferentes formas para representação da informação, ou seja, foram criados novos modelos de dados para retratar a estrutura das informações contida no banco de dados.

Atualmente, os principais modelos de estrutura de dados utilizados pelos SGBD são: modelo hierárquico, modelo em redes, modelo relacional (amplamente usado) e o modelo orientado a objetos [15].

- Banco de dados hierárquico: os dados são estruturados em árvores ou hierarquias com sentido unidirecional, no qual possuem uma coleção de registros que são conectados uns ao outros por meio de ligações entre nós pai e nós filhos. Além disso, cada registro é uma coleção de campos (atributos), cada um contendo apenas uma informação, conforme mostra a Figura 5.

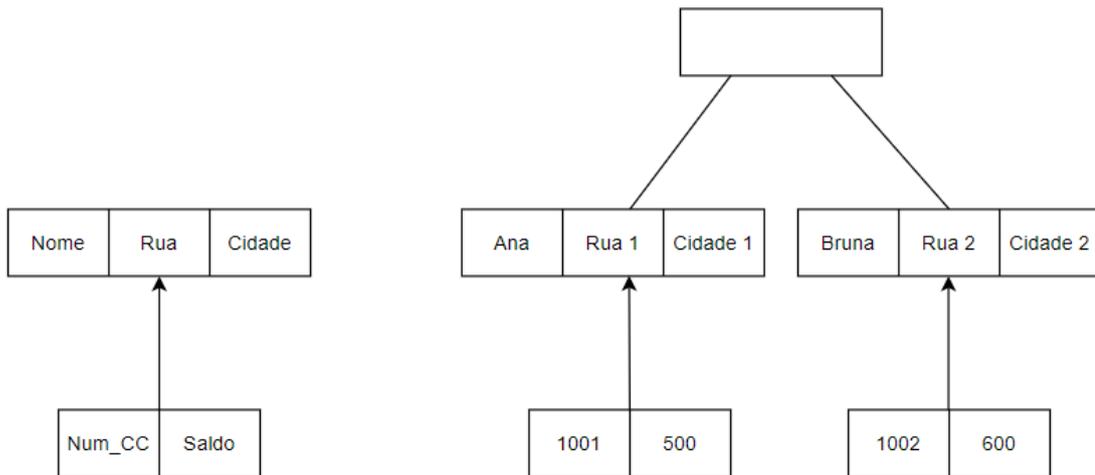


Figura 5 – Exemplo de dados em forma hierárquica - Conta Corrente

- Banco de dados de redes: o modelo surgiu como uma extensão do modelo hierárquico. Neste modelo os dados são representados por registros e inter-ligações, permitindo que uma determinada ocorrência de registro tenha várias associações, eliminando assim o conceito de hierarquia, como mostra a Figura 6.
- Modelo relacional: é um modelo de dados que baseia-se no princípio de que os dados estão armazenados em tabelas, também chamadas de relações. Basicamente uma relação é constituída por um ou mais atributos (campos), que é o tipo de dados a armazenar. A instância de uma relação é o conjunto de linhas, denominadas tuplas ou registros [15], conforme a Tabela 1. Além disso, este modelo surgiu devido a necessidade de aumentar a independência de dados no sistema gerenciador de banco de dados e permitir funções baseado na álgebra relacional para manipulação dos dados.

Tabela 1 – Exemplo de dados em forma relacional - Conta Corrente

Cod_Cliente	Nome	Cidade	Num_CC	Saldo	Cod_Cliente	Num_CC
1	Ana	Cidade1	10001	1000	1	10001
2	Pedro	Cidade2	10002	2000	2	10002
3	Fabio	Cidade3	10003	3000	3	10003

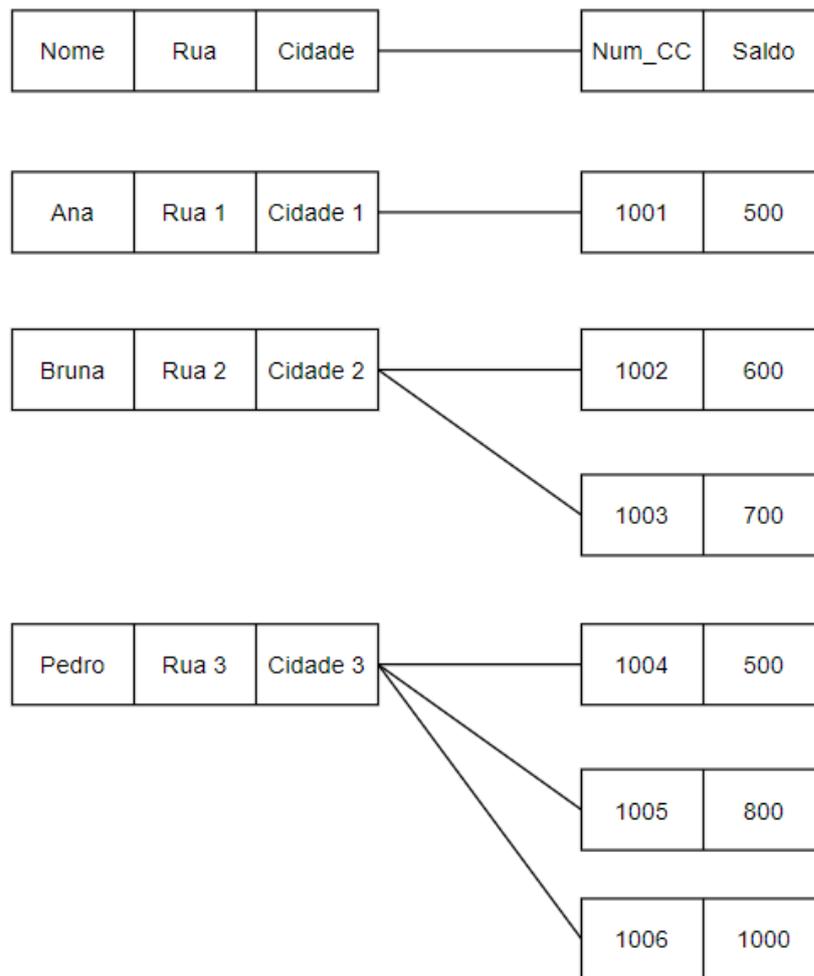


Figura 6 – Exemplo de dados em forma de rede - Conta Corrente

- Banco de dados orientado a objetos: nesse modelo os dados são armazenados na forma de objetos, utilizando o paradigma denominada orientação a objetos. Entretanto, nesse sistema é necessário armazenar as representações das estruturas de dados que usam o armazenamento permanente. Acredita-se que esse modelo de banco de dados deve ser utilizados apenas em aplicações especializados, onde o diagrama UML pode representar o modelo de dados orientados a objetos, conforme a Figura 7.

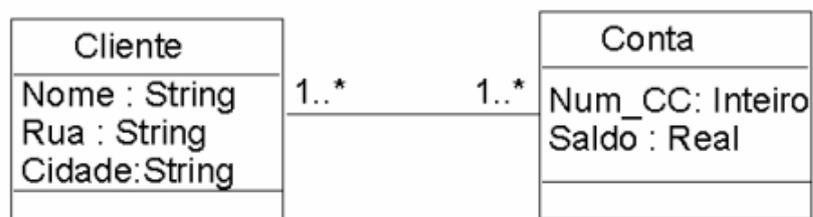


Figura 7 – Exemplo de dados em forma orientado a objetos - Conta Corrente

No projeto de um banco de dados, são levados em consideração dois níveis de abstração de modelos de dados, o modelo conceitual e o modelo lógico. O modelo conceitual é como uma descrição do banco de dados de forma independente de implementação em um SGBD. Nele registra-se os dados que podem aparecer no banco de dados, mas não são armazenados em nível de SGBD [16]. Ou seja, a modelagem trata de aspectos do negócio do cliente e não da tecnologia.

A técnica mais utilizada para realizar a modelagem conceitual é a abordagem entidade-relacionamento (ER), através de diagramas entidade-relacionamento (DER), onde são identificados todas as entidades e os relacionamento entre elas [16]. A Figura 8 representa o DER para um determinado problema.

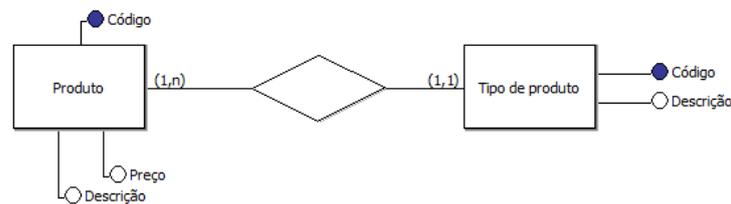


Figura 8 – Exemplo modelo conceitual

O modelo lógico é uma descrição de um banco de dados no nível de abstração visto pelo usuário do SGBD e depende do tipo de estrutura de SGBD que é utilizada [16]. Além disso, precisa considerar aspectos como a adequação de padrão e nomenclatura, normalização, chaves primárias, entre outras. Para realizar a modelagem lógica deve ser projetado a partir do modelo lógico, conforme a Figura 9, que foi criado a partir do modelo conceitual da Figura 8.

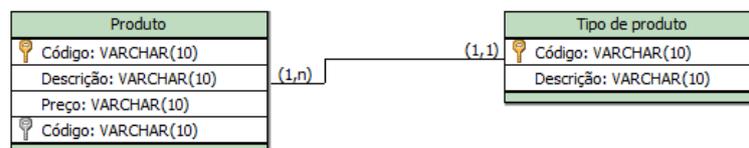


Figura 9 – Exemplo modelo lógico

Para entendermos melhor a respeito da modelagem entidade-relacionamento é necessário definir alguns conceitos, como: entidade, relacionamento, atributo, generalização e entidade associativa.

Entidade pode ser definida como um conjunto de objetos sobre os quais deseja-se manter informações no mundo real, com uma identificação e com um significado próprio. Existem dois tipos de entidades, a física que são as que existem no mundo, como um

cliente, uma pessoa, um empresa, um produto, entre outras. Já entidades lógicas, são aquelas que existem por conta de uma decorrência de interações entre entidades físicas, que não são objetos físicos, como por exemplo uma venda, uma classificação.

Além disso, uma propriedade muito importante de um relacionamento é a cardinalidade, que significa o número de ocorrências de entidades associadas com uma ocorrência da entidade em questão, ou seja, o grau de relação entre duas entidades.

O relacionamento pode ser classificado de três formas: $n:n$, $1:n$ e $1:1$.

- $n:n$ (muitos para muitos): nesse relacionamento, cada entidade de ambos os lados podem referenciar múltiplas unidades da outra, como por exemplo um médico atende muitos pacientes e os pacientes podem ser atendidos por muitos médicos;
- $1:n$ (um para muitos): nesse relacionamento uma das entidades envolvidas pode relacionar-se com várias ocorrências, como por exemplo uma marca possui vários produtos, mas um produto pertence apenas a uma marca;
- $1:1$ (um para um): nesse relacionamento as entidades envolvidas são obrigatoriamente relacionadas a apenas uma unidade da outra, como por exemplo uma pessoa possui apenas um CPF.

Atributo nada mais é que um dado associado a cada ocorrência de uma entidade, ou seja, é uma característica de descreve uma entidade, como um cliente (entidade), possui nome, endereço e telefone.

Além de relacionamentos e atributos, podem ser atribuídas propriedades em uma entidade através do conceito de generalização/especialização. Esse conceito baseia-se na ideia de herança de propriedades, no qual é possível atribuir propriedades especiais a um subconjunto das ocorrências de uma entidade genérica, como pode ser visto na Figura 10, que expressa um exemplo de generalização/especialização em que Cliente é a entidade genérica, resumidamente toda pessoa física possui atributos nome, CPF, sexo e é identificada através do código, da mesma forma que toda pessoa jurídica, possui atributos nome, CNPJ, tipo de organização e é identificada pelo código.

2.2.1 Normalização

O processo de normalização tem o objetivo de modelar o banco de dados projetando a forma com que os dados serão armazenados, para eliminar e minimizar a redundância do banco que possa existir e reagrupar as informações de uma forma que permita obter um modelo conceitual [17]. Esse processo baseia-se no conceito de forma normal que é um conjunto de regra que devem ser obedecidas.

A primeira forma normal (1FN) é atingida quando todos os atributos contém apenas um valor correspondente, e não permitem grupos de atributos repetidos ou compostos.

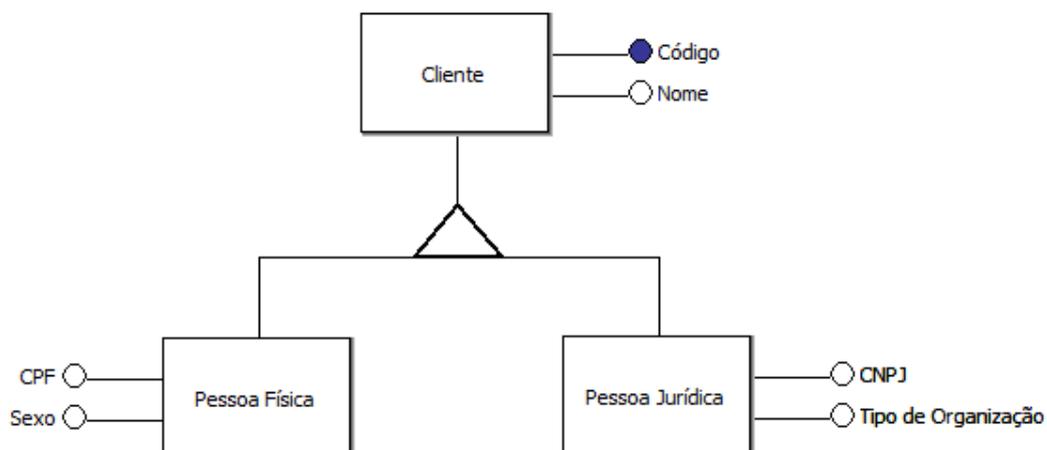


Figura 10 – Exemplo de Generalização/Especialização

Tabela 2 – Exemplo de uma tabela não normalizada

CodProj	Tipo	Descr	Emp					
			CodEmp	Nome	Cat	Sal	DataIni	TempAl
LSC001	Desenvolvimento	Sistema de Estoque	2146	João	A1	4	01/11/91	24
			3145	Silvio	A2	4	02/10/91	24
			6126	José	B1	9	03/10/92	18
			1214	Carlos	A2	4	04/10/92	18
			8191	Mário	A1	4	01/11/92	12
PAG02	Sustentação	Sistema de RH	8191	Mário	A1	4	01/05/93	12
			4112	João	A2	4	04/01/91	24
			6126	José	B1	9	01/11/92	12

Tabela 3 – Tabela Proj - 1FN

CodProj	Tipo	Descr
LSC001	Desenvolvimento	Sistema de Estoque
PAG02	Sustentação	Sistema de RH

Percebe-se que, a Tabela 2 não está na primeira forma normal, para isso é necessário decompor nas Tabelas 3 e 4, no qual a coluna CodProj é a chave primária da Tabela 3, já a chave primária da Tabela 4 é composta pelas colunas CodProj e CodEmp. Isto ocorre pelo fato que o mesmo trabalhador pode estar em vários projetos.

A segunda forma normal (2FN) é atingida quando atende-se a todos os requisitos da primeira forma normal e tem o objetivo de eliminar dados de redundância, quando contém dependências funcionais parciais, ou seja, possui colunas que dependem apenas

Tabela 4 – Tabela ProjEmp - 1FN

CodProj	CodEmp	Nome	Cat	Sal	DataIni	TempAl
LSC001	2146	João	A1	4	01/11/91	24
	3145	Silvio	A2	4	02/10/91	24
	6126	José	B1	9	03/10/92	18
	1214	Carlos	A2	4	04/10/92	18
	8191	Mário	A1	4	01/11/92	12
PAG02	8191	Mário	A1	4	01/05/93	12
	4112	João	A2	4	04/01/91	24
	6126	José	B1	9	01/11/92	12

de uma parte da chave primária. Exemplificando, a Tabela 4 contém dados referentes aos empregados (Nome, Cat e Sat) que estão redundantes, pois há empregados que trabalham em mais de um projeto.

Para isso é necessário criar uma nova tabela Emp, com os atributos Nome, Sal, Cat e chave primária CodEmp, conforme a Tabela 5 para eliminar a redundância. Também os atributos DataIni e TempAl da tabela ProjEmp são dependentes funcionais do CodProj.

Na terceira forma normal (3FN), elimina-se outro tipo de redundância, quando uma coluna não chave primária depende de outra coluna ou combinação de colunas não chave primária. Para exemplificar, considera-se que o salário (coluna Sal) de um empregado é determinado pela sua categoria (coluna Cat), conforme tabela Emp da Tabela 5. Então para esse caso, nota-se que ocorre uma redundância, pois o salário é pago a uma categoria e existem vários funcionários que possuem a mesma categoria. Para solucionar esse problema, é necessário criar uma nova tabela Cat com o atributo Sal e chave primária Cat, conforme a Tabela 6.

A quarta forma normal, ocorre quando ela estiver na terceira forma normal e não existir dependências multi-valoradas, ou seja, campos repetidos em relação a chave primária, no qual geram redundâncias. Caso existir, deve-se fragmentar essa relação com o objetivo de não restar mais dependências funcionais de gênero [17].

2.2.2 Linguagem SQL

O SQL (*Structured Query Language*) é uma linguagem de banco de dados desenvolvida pela IBM com o intuito de padronizar os comandos de manipulação de dados no banco de dados relacional. Com ela pode-se realizar instruções chamadas de *Queries* para construir ou consultar tabelas, colunas, permissões de usuários entre outras funcionalidades [18].

O comando *Select* permite recuperar os dados de uma linha ou uma seleção de linhas do banco de dados. A sintaxe utilizada para esse comando é:

```
Select <nome_coluna>
```

Tabela 5 – Tabela 2FN

Proj:

CodProj	Tipo	Descr
LSC001	Desenvolvimento	Sistema de Estoque
PAG02	Sustentação	Sistema de RH

ProjEmp:

CodProj	CodEmp	DataIni	TempAl
LSC001	2146	01/11/91	24
	3145	02/10/91	24
	6126	03/10/92	18
	1214	04/10/92	18
	8191	01/11/92	12
PAG002	8191	01/05/93	12
	4112	04/01/91	24
	6126	01/11/92	12

Emp:

CodEmp	Nome	Cat	Sal
2146	João	A1	4
3145	Silvio	A2	4
6126	José	B1	9
1214	Carlos	A2	4
8191	Mário	A1	4
8191	Mário	A1	4
4112	João	A2	4
6126	José	B1	9

O comando *Where* é utilizada para recuperar dados que atentem a uma condição, ou seja, possui a função de um filtro. A sintaxe utilizada para esse comando é:

`Where <condição>`

O comando *Order By* permite que o dados sejam ordenados por um ordem específica. A sintaxe utilizada para esse comando é:

`Order By <nome_coluna>`

Para exemplificar, considere-se as informações das notas dos estudantes da escola Geração, conforme a Tabela 7.

Para descobrir os estudantes que possuem a nota maior que 8, construiu-se a com a seguinte *query*:

Tabela 6 – Tabela 3FN

Proj:

CodProj	Tipo	Descr
LSC001	Desenvolvimento	Sistema de Estoque
PAG02	Sustentação	Sistema de RH

ProjEmp:

CodProj	CodEmp	DataIni	TempAl
LSC001	2146	01/11/91	24
	3145	02/10/91	24
	6126	03/10/92	18
	1214	04/10/92	18
	8191	01/11/92	12
PAG002	8191	01/05/93	12
	4112	04/01/91	24
	6126	01/11/92	12

Emp:

CodEmp	Nome	Cat
2146	João	A1
3145	Silvio	A2
6126	José	B1
1214	Carlos	A2
8191	Mário	A1
8191	Mário	A1
4112	João	A2
6126	José	B1

Cat:

Cat	Sal
A1	4
A2	4
B1	9

Tabela 7 – Notas dos Estudantes

Matricula	Nome	Nota
1	Alex	9
2	Bruna	8
3	Carlos	7
4	Davi	8
5	Eduardo	9
6	Fábio	6
7	Giovana	8

```

Select *
From Notas_dos_Estudantes
Where Nota > 8
Order By Notas_dos_Estudantes.Matricula

```

O resultado da *query* é representado conforme a Tabela 8.

Tabela 8 – Notas dos Estudantes

Matricula	Nome	Nota
1	Alex	9
5	Eduardo	9

Além disso, existe o comando *Join* que é utilizado para juntar duas ou mais tabelas na mesma pesquisa, com base em um relacionamento entre colunas específicas dessas tabelas. A sintaxe utilizada para esse comando é:

```

Select <nome_coluna>
From<nome_tabela_1>
Join <nome_tabela_2>
On (<nome_coluna_1>=<nome_coluna_2>)

```

Para encontrar a idade de cada estudante é necessário fazer a junção da Tabela 7 com a Tabela 9. Utilizando o comando *Join* constrói-se a seguinte *query*:

```

Select *
From Notas_dos_Estudantes
Join Idade_dos_Estudantes
On (Notas_dos_Estudantes.Matricula = Idade_dos_Estudantes.Matricula)
Order By Notas_dos_Estudantes.Matricula

```

O resultado da junção das duas tabelas é representado na Tabela 10.

Tabela 9 – Idade dos Estudantes

Matricula	Idade
1	10
2	11
3	13
4	14
5	17
6	12
7	9

Tabela 10 – Junção das Tabelas Notas dos Estudantes e Idade dos Estudantes

Matricula	Nome	Nota	Idade
1	Alex	9	10
2	Bruna	8	11
3	Carlos	7	13
4	Davi	8	14
5	Eduardo	9	17
6	Fábio	6	12
7	Giovana	8	9

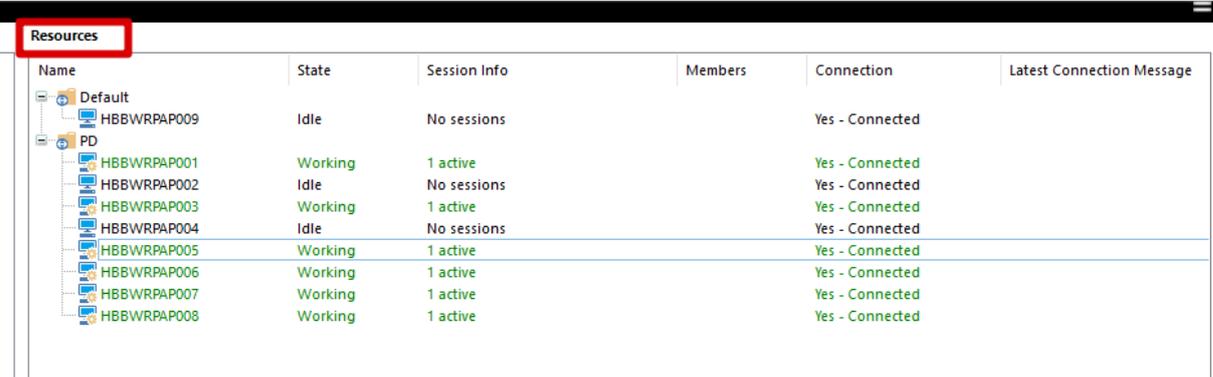
3 Desenvolvimento

Este capítulo apresenta o sistema de RPA desenvolvido para realizar o monitoramento das automações (robôs - RPA) de forma automatizada na empresa Bunge. Nele são descritas cada etapa do processo realizada pela automação desenvolvida.

3.1 Monitoramento de RPA não automatizado

Esta seção explica como era realizado o monitoramento de RPA de forma manual na empresa. O processo de monitoramento de RPA consiste em monitorar os robôs e recursos para identificar possíveis falhas que possam comprometer a conclusão de uma determinada atividade executada pelos robôs. Recursos são ambientes virtuais onde os robôs são executados, como por exemplo máquinas virtuais ou servidores virtuais. Esse monitoramento é, atualmente, feito por operadores humanos que utilizam informações de um painel de controle chamado Control Room, disponibilizado pela ferramenta Blue Prism. Esse painel de controle fornece uma série de informações a respeito dos recursos e dos robôs, como por exemplo seu estado, se está em execução, ou seja se o robô está realizando as tarefas que foi programado para executar, e também as agendas dos robôs que foram programados para serem executados em uma determinada data.

A janela de recursos ilustrada na Figura 11 mostra várias informações de todos os recursos disponíveis, exibindo o nome, o status, a informação da seção, membros, conexão e a última mensagem de conexão. O status do recurso pode ser: *Working*, indicando que um robô está sendo executado no recurso; *Idle* é quando não possui nenhum robô sendo executado no recurso; e *Offline* indicando que o recurso perdeu a conexão com o Blue Prism. A Figura 11 mostra que os recursos HBBWRPAP002, HBBWRPAP004 e HBBWRPAP009 estão ociosos, enquanto os recursos HBBWRPAP001, HBBWRPAP003, HBBWRPAP005 ao HBBWRPAP008 estão sendo utilizados, ou seja, há robôs em execução nesses recursos.



Name	State	Session Info	Members	Connection	Latest Connection Message
Default					
HBBWRPAP009	Idle	No sessions		Yes - Connected	
PD					
HBBWRPAP001	Working	1 active		Yes - Connected	
HBBWRPAP002	Idle	No sessions		Yes - Connected	
HBBWRPAP003	Working	1 active		Yes - Connected	
HBBWRPAP004	Idle	No sessions		Yes - Connected	
HBBWRPAP005	Working	1 active		Yes - Connected	
HBBWRPAP006	Working	1 active		Yes - Connected	
HBBWRPAP007	Working	1 active		Yes - Connected	
HBBWRPAP008	Working	1 active		Yes - Connected	

Figura 11 – Recursos Blue Prism

A janela *Environment* exibe as informações da execução de cada robô, mostrando o status, ID (atribuído um número de sessão de uma execução de um robô), o recurso, usuário que iniciou sua execução, data de início e fim, como mostra a Figura 12. O status pode ser *completed*, indicando que a execução de um robô concluiu com sucesso; *running* quando o robô está em execução; e o *terminated* quando ocorre algum erro durante a execução do robô resultando no término de sua execução.

A Figura 12 mostra que os robôs BSA_BR_BBS_PTP_Invoice_Receipt ARG Invoice, BSA_BR_BBS_PTP_Invoice_ValidationOfBasicData_Services e BSA_BR_BB_PTP_Invoice_ValidationOfBasicData_Material estão em execução, respectivamente, nos recursos HBBWRPAP006, HBBWRPAP003 e HBBWRPAP001. Porém o robô BSA_BR_BBS_OE_BackOffice_04-01-52 apresentou o status de *terminated* que representa um erro durante sua execução. Os demais robôs executaram com sucesso.

ID	Process	Resource	User	Status	Start Time	End Time	Latest Stage	Stage Started
120925	BSA_BR_BBS_PTP_AccountsPayable_BackOffice_01-03-03_PaymentToSu...	HBBWRPAP008	[Scheduler]	Completed	19/10/2020 10:26:37	19/10/2020 10:27:31		
120924	BSA_BR_BBS_PTP_AccountsPayable_BackOffice_01-02-03_01-02-08_Anti...	HBBWRPAP008	[Scheduler]	Completed	19/10/2020 10:20:02	19/10/2020 10:26:35		
120923	BSA_BR_LOGISTICS_LogisticsPlanning_LogisticsPlanningGrains_Grains...	HBBWRPAP007	[Scheduler]	Completed	19/10/2020 10:20:01	19/10/2020 10:22:19		
120922	BSA_BR_BBS_TAX_State_RegisterRO_ExtractReport	HBBWRPAP004	[Scheduler]	Completed	19/10/2020 10:18:04	19/10/2020 10:18:45		
120921	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Terminated	HBBWRPAP008	[Scheduler]	Completed	19/10/2020 10:17:12	19/10/2020 10:17:30		
120920	BSA_BR_BBS_OE_MD_BackOffice_04-01-52_SupplierChangeNFEACredit...	HBBWRPAP008	[Scheduler]	Terminated	19/10/2020 10:15:55	19/10/2020 10:17:09		
120919	BSA_BR_BBS_TAX_State_StatusVIM	HBBWRPAP004	[Scheduler]	Completed	19/10/2020 10:13:00	19/10/2020 10:18:02		
120917	BSA_BR_BBS_TAX_State_StatusRO	HBBWRPAP004	[Scheduler]	Completed	19/10/2020 10:06:25	19/10/2020 10:12:57		
120916	BSA_BR_BBS_TAX_State_RegisterRO	HBBWRPAP004	[Scheduler]	Completed	19/10/2020 10:00:03	19/10/2020 10:06:22		
120915	BSA_BR_BBS_PTP_Invoice_Receipt ARG Invoice	HBBWRPAP006	[Scheduler]	Running	19/10/2020 10:00:03		Move	19/10/2020 10:32:53
120914	BSA_BR_BBS_PTP_Invoice_BackOffice_01-18-03_ReceiptInvoiceTicket	HBBWRPAP008	[Scheduler]	Completed	19/10/2020 09:54:19	19/10/2020 10:15:53		
120913	BSA_BR_BBS_PTP_AccountsPayable_BackOffice_01-03-03_PaymentToSu...	HBBWRPAP008	[Scheduler]	Completed	19/10/2020 09:53:20	19/10/2020 09:54:17		
120912	BSA_BR_BBS_PTP_Invoice_ValidationOfBasicData_Services	HBBWRPAP003	[Scheduler]	Running	19/10/2020 09:49:58		Error ABAP	19/10/2020 10:32:52
120911	BSA_BR_BBS_PTP_Invoice_ValidationOfBasicData_Material	HBBWRPAP001	[Scheduler]	Running	19/10/2020 09:49:58		Hide Options	19/10/2020 10:32:38
120910	BSA_BR_BBS_PTP_AccountsPayable_BackOffice_01-02-03_01-02-08_Anti...	HBBWRPAP008	[Scheduler]	Completed	19/10/2020 09:46:24	19/10/2020 09:53:17		

Figura 12 – Sessão dos Processos Blue Prism

A janela *Recenty Activity* exibe o histórico das execuções dos agendamentos dos robôs, mostrando o nome do agendamento, o status, o horário de início e fim, conforme a Figura 13. Um agendamento pode ter dois tipos de status: execução bem sucedida, indicado pelo marcador conforme a Figura 14 (a) e falha no agendamento, indicado pelo marcador como mostra a Figura 14 (b).

Schedule	Start Time	End Time	Server
Migrado - Monthly - BSA_BR_BBS_RTR_Recon_ProfitCenterApportionment 00:00* - P002	25/10/2020 00:00:00	25/10/2020 00:02:47	HBBWAPLP176:8199
Migrado - Daily - BSA_BR_BBS_PTP_Invoice_Receipt ARG Invoice - 14:30 - P007	25/10/2020 00:30:00	25/10/2020 00:30:43	HBBWAPLP176:8199
Migrado - Daily - CN_SSC_RTR_PriceAdjustment_Create 03:00 - P002	25/10/2020 03:00:00	25/10/2020 03:00:35	HBBWAPLP176:8199
Once - Hourly / Minutely - BSA_BR_BBS_PTP_Invoice_ValidationOfBasicData_Services- P001 - 24/10 e 25/10	25/10/2020 09:00:00	25/10/2020 09:05:57	HBBWAPLP176:8199
Once - Hourly/Minutely - BSA_BR_BBS_PTP_Invoice_ValidationOfBasicData_Material - P005 - 24/10 e 25/10	25/10/2020 09:00:01	25/10/2020 09:03:06	HBBWAPLP176:8199
Migrado - Monthly - BSA_BR_Procurement_Procurement_Scorecard_CreateProject_CheckPossibilities 11:50* - P002	25/10/2020 11:50:00	25/10/2020 12:00:42	HBBWAPLP176:8199
Migrado - Weekends - BSA_BR_BBS_PTP_Invoice_Receipt ARG Invoice - 16:00 - P003	25/10/2020 16:00:00	25/10/2020 16:00:03	HBBWAPLP176:8199

Figura 13 – Agendamentos das automações Blue Prism

Para realizar o monitoramento de RPA, o operador humano deve ficar continuamente analisando as janelas de recursos, *Environment* e *Recent Activity* afim de identificar a ocorrência de falhas durante a execução de um robô, um robô não iniciar conforme o horário programado ou alguma falha nos recursos. Na próxima seção é descrita uma solução para realizar o monitoramento de RPA de forma automatizada.

3.2 Metodologia de solução de automação

Um dos grandes problemas na equipe de RPA dentro da empresa é o monitoramento dos robôs. Como se trata de um processo realizado manualmente e pelo grande número de robôs ativos, o operador humano não consegue detectar todas as falhas que ocorrem nos robôs, por estar realizando outras atividades pelas quais é responsável. Essas falhas podem acarretar uma série de problemas para um usuário, cujas tarefas dependem da finalização correta dos processos realizados pelos robôs. O trabalho do usuário será prejudicado caso ocorra alguma falha do robô, podendo até mesmo ser necessário realizar as atividades do robô manualmente.



(a) Marcador de completo Blue Prism



(b) Marcado de falha Blue Prism

Figura 14 – Marcado de completo e falha Blue Prism

Por conta dessas questões, estudou-se uma forma de realizar o monitoramento dos robôs de forma automática. Para isso foi preciso entender quais eram as informações essenciais para identificar se ocorreu algum problema com o robô. Tendo como base a ferramenta Blue Prism, verificou-se que as principais maneiras de detectar se houve alguma falha no robô, seriam realizar a identificação dos robôs que não iniciaram conforme o horário programado, identificar se ocorreu algum erro durante a execução de um determinado robô e identificar problemas com o ambiente onde o robô é executado.

3.3 Projeto e implementação da solução de automação

Basicamente, o projeto consiste em monitorar os robôs de forma automatizada, consultando informações essenciais no banco de dados para realizar o monitoramento e, caso algum problema seja identificado, comunicar os responsáveis. Para isso o projeto foi dividido nos seguintes tópicos:

- Identificação das informações necessárias para o monitoramento;
- Mapeamento das informações;
- Desenvolvimento de um mecanismo para obtenção das informações;
- Compartilhamento das informações relevantes;
- Implementação do robô.

Os tópicos citados acima são tratados nas Seções 3.3.1 até 3.3.5.

3.3.1 Identificação das informações necessárias para o monitoramento

Primeiramente, foi necessário identificar quais informações seriam usadas para realizar o monitoramento. Para isso levantou-se as informações necessárias para a identificação das falhas que poderiam acontecer com o robô com base nas informações que o painel de controle da ferramenta Blue Prism fornece. Essas informações são:

- *nome do robô*, para identificar robôs;
- *status do robô*, para identificar se o robô está em execução ou não;
- *horário de início do robô*, para identificar o horário que o robô foi iniciado;
- *horário de término do robô*, para identificar o horário que o robô finalizou sua execução;

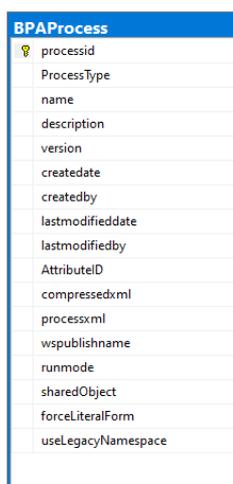
- *nome do recurso*, para identificar e diferenciar o nome do ambiente onde os robôs são executados;
- *status do recurso*, para verificar se o recurso está sendo utilizado ou não e verificar se está funcionando corretamente;
- *usuário que iniciou o robô*, para verificar se o robô foi iniciado manualmente, por um operador humano, ou se foi iniciado conforme programado.

3.3.2 Mapeamento das informações

As informações identificadas anteriormente são armazenadas em um banco de dados relacional, estruturadas em tabelas. Essas tabelas são organizadas em colunas, cada coluna armazena um tipo de informação. Então o próximo passo, foi identificar quais tabelas continham essas informações. Para isso utilizou-se o software SQL Server Management Studio, que permitiu a visualização das tabelas do banco de dados do Blue Prism.

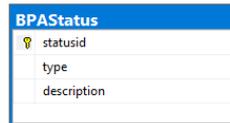
Não havia informação ou documentação a respeito de como o banco de dados e sua estrutura foram definidos e o modo com que as tabelas foram relacionadas. Por conta disso, foi preciso analisar manualmente cada tabela do banco de dados. Foram cerca de 180 tabelas analisadas para encontrar as informações identificadas anteriormente.

A informação com o nome do robô foi encontrada na tabela BPAProcess, que possui a chave primária processid, conforme a Figura 15. A informação com o status foi encontrada na tabela BPAStatus, que possui a chave primária statusid, conforme a Figura 16. A informação com o nome do recurso foi encontrada na tabela BPAResource, que possui a chave primária resourcid, conforme a Figura 17. A informação com o usuário foi encontrada na tabela BPAUser, que possui a chave primária userid, conforme a Figura 18.



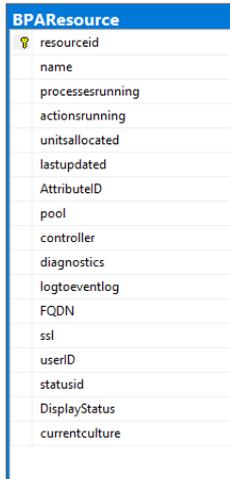
BPAProcess	
processid	ProcessType
name	
description	
version	
createdate	
createdby	
lastmodifieddate	
lastmodifiedby	
AttributelD	
compressedxml	
processxml	
wspublishname	
runmode	
sharedObject	
forceLiteralForm	
useLegacyNamespace	

Figura 15 – Tabela Process



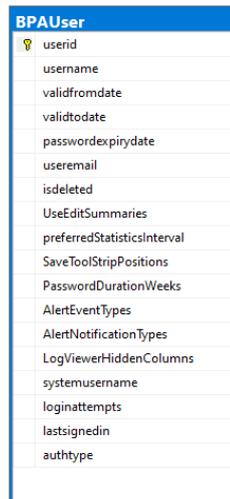
BPASStatus	
statusid	
type	
description	

Figura 16 – Tabela Status



BPAResource	
resourceid	
name	
processesrunning	
actionsrunning	
unitsallocated	
lastupdated	
AttributelD	
pool	
controller	
diagnostics	
logtoeventlog	
FQDN	
ssl	
userID	
statusid	
DisplayStatus	
currentculture	

Figura 17 – Tabela Resource



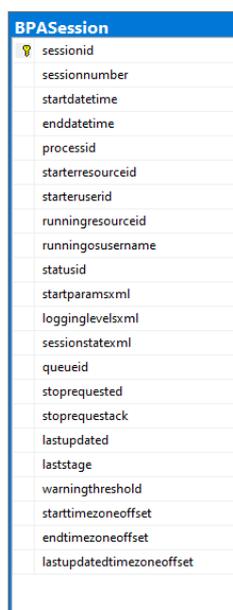
BPAUser	
userid	
username	
validfromdate	
validtodate	
passwordexpirydate	
useremail	
isdeleted	
UseEditSummaries	
preferredStatisticsInterval	
SaveToolStripPositions	
PasswordDurationWeeks	
AlertEventTypes	
AlertNotificationTypes	
LogViewerHiddenColumns	
systemusername	
loginattempts	
lastsignedin	
authtype	

Figura 18 – Tabela User

Em um modelo relacional, a forma de relacionar uma tabela com outra é através das informações comuns existentes em ambas. Ou seja, há informações especiais chamadas de chaves estrangeiras para fazer a ligação com outras tabelas, em que essas mesmas informações são utilizados para identificar cada umas das linhas, chamada de chaves primárias.

Sabendo disso, o próximo passo foi descobrir como as tabelas se relacionam. Para isso descobriu-se que existe uma tabela em específico responsável pela conexão entre todas as tabelas. A tabela encontrada foi a BPASession, como mostrado na Figura 19.

Essa tabela possui as chaves estrangeiras que referenciam as chaves primárias das tabelas BPAResource, BPAStatus, BPAUser e BPAProcess, conforme a Figura 20.



BPASession	
sessionid	
sessionnumber	
startdatetime	
enddatetime	
processid	
starterresourceid	
starteruserid	
runningresourceid	
runningosusername	
statusid	
startparamsxml	
logginglevelxml	
sessionstatexml	
queueid	
stoprequested	
stoprequestack	
lastupdated	
laststage	
warningthreshold	
starttimezoneoffset	
endtimezoneoffset	
lastupdatedtimezoneoffset	

Figura 19 – Tabela Session

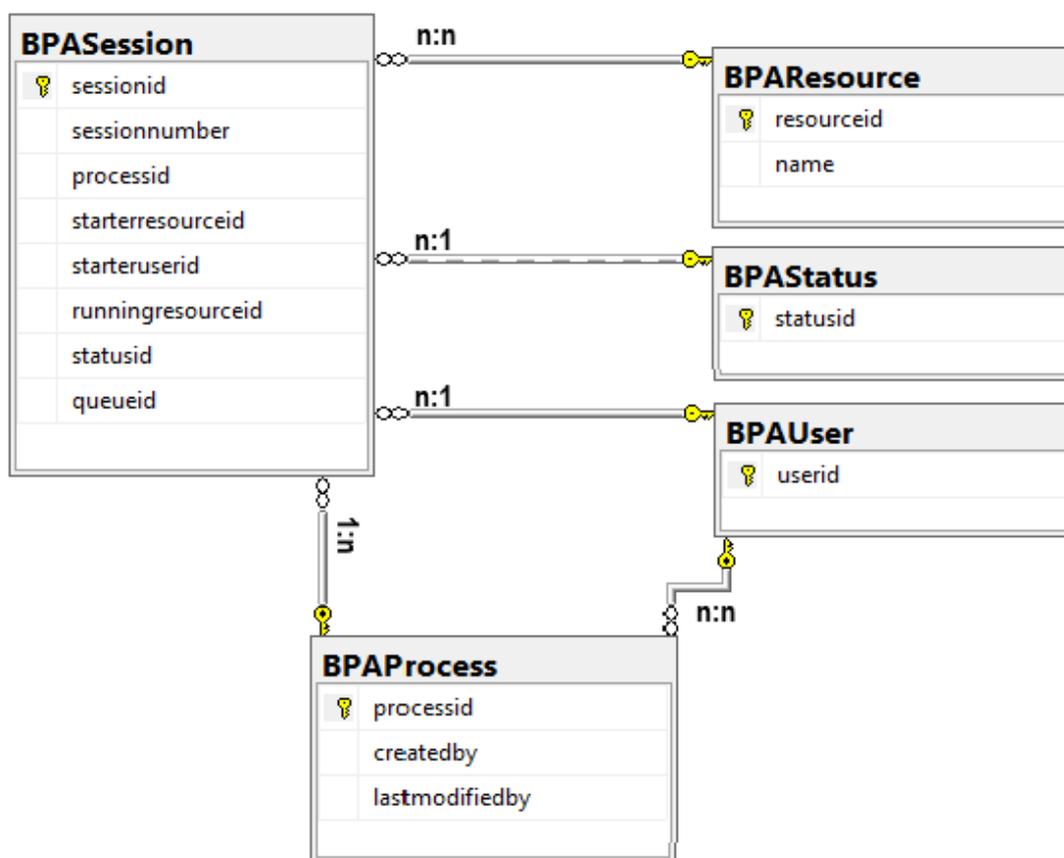


Figura 20 – Diagrama de Entidade-Relacionamento

Após compreender a maneira com que as tabelas se relacionam, é possível criar a consulta para obter as informações necessárias para o monitoramento. Para isso foi utilizado a linguagem SQL, que é uma linguagem textual que permite a formulação da consulta por meio de um expressão textual, que possui uma sintaxe e um conjunto de regras definida.

Para esse trabalho foram criadas três consultas (query) para obter as informações necessárias para realizar o monitoramento.

1. Query para o monitoramento das execuções das automações

A query para o monitoramento das execuções dos robôs foi criada para obter os dados necessários para identificar se ocorreu algum problema durante a execução do robô, através do valor do status, como mostra o código abaixo:

```
select
    BPASession.sessionnumber as sessionnumber,
    BPAProcess.name as processname,
    BPAStatus.description as status,
    BPASession.[startdatetime] as [startdatetime],
```

```

    BPASession.[enddatetime] as [enddatetime],
    DATEPART(MINUTE,enddatetime-startdatetime) AS duração,
    BPAUser.username as starterusername,
BPASession.name as resourcename
from [BLUEPRISM65-PRD].[dbo].[BPASession]
    join BPAProcess on BPASession.processid = BPAProcess.processid
    join BPAStatus on BPASession.statusid = BPAStatus.statusid
    join BPAResource on BPASession.starterresourceid = BPAResource.resourceid
    join BPAUser on BPASession.starteruserid = BPAUser.userid
where
st.description = 'Terminated' and enddatetime >
dateadd(MINUTE,-"&[TempoSubtraidoDePesquisa(Minute)]&",Current_timestamp)

```

Basicamente na consulta são selecionados os campos `sessionnumber`, `processname`, `status`, `startdatetime`, `enddatetime`, `starterusername` e `resourcename` da junção da tabela `BPASession` com as tabelas `BPAProcess`, `BPAStatus`, `BPAResource` e `BPAUser`, respeitando a condição de que o valor do `status` deve ser igual ao valor `Terminated` e o valor `enddatetime` deve ser maior que o valor do tempo atual subtraído um determinado valor de tempo.

2. Query para o monitoramento das falhas de agendamento

A query para o monitoramento das falhas de agendamento foi criada com o objetivo de consultar os dados necessários para identificar a não inicialização de um robô em seu horário programado. Para isso foi desenvolvido o seguinte código:

```

SELECT
BPVAnnotatedScheduleLog.scheduleid,
    BPASchedule.name,
    BPVAnnotatedScheduleLog.starttime as starttime,
    BPATaskSession.resourcename,
    BPVAnnotatedScheduleLog.endreason,
    BPVAnnotatedScheduleLog.endtype

FROM [BLUEPRISM65-PRD].[dbo].[BPVAnnotatedScheduleLog]
    inner join [BLUEPRISM65-PRD].[dbo].[BPASchedule] on
    BPVAnnotatedScheduleLog.scheduleid = BPASchedule.id
    inner join [BLUEPRISM65-PRD].[dbo].[BPATaskSession] on
    BPASchedule.initialtaskid = BPATaskSession.taskid
where
dateadd(hour,-3,starttime) >

```

```
dateadd(MINUTE,-"&[TempoSubtraidoDePesquisa(Minute)]&",Current_timestamp)
```

Nesse código são selecionados os campos `scheduleid`, `name`, `starttime`, `resourcename` e `endreason` da união da tabela `BPVAnnotatedScheduleLog` com as tabelas `BPASchedule` e `BPATaskSession`, respeitando a condição de que o valor `starttime` deve ser maior que o valor do tempo atual subtraído um determinado valor de tempo.

3. Query para o monitoramento das falhas dos recursos

A query para o monitoramento dos recursos foi criada para identificar a ocorrência de algum problema com o ambiente onde os robôs são executados. Para isso foi criado o seguinte código:

```
SELECT
    BPAResource.name,
    BPAResource.DisplayStatus as status
FROM [BLUEPRISM65-PRD].[dbo].[BPAResource] a
where
(status = 'Offline' or status = 'Missing') and
((a.name = 'HBBWRPAP001') or (a.name = 'HBBWRPAP002')
or (a.name = 'HBBWRPAP003') or (a.name = 'HBBWRPAP004')
or (a.name = 'HBBWRPAP005') or (a.name = 'HBBWRPAP006')
or (a.name = 'HBBWRPAP007') or (a.name = 'HBBWRPAP008'))
order by a.name
```

Nesse código pode ser visto que são selecionados apenas os campos `name` e `status` da tabela `BPAResource`, respeitando a condição de que o valor do `status` deve ser igual ao valor `Offline` ou `Missing`. Isso quer dizer que ocorreu alguma falha no recurso e também de que o valor do campo `name` deve ser igual ao nomes dos recursos: `HBBWRPAP001` ou `HBBWRPAP002` ou `HBBWRPAP003` ou `HBBWRPAP004` ou `HBBWRPAP005` ou `HBBWRPAP006` ou `HBBWRPAP007` ou `HBBWRPAP008`.

3.3.3 Desenvolvimento de um mecanismo para obtenção das informações

Para a obtenção das informações, foi desenvolvido um mecanismo que realiza algumas tarefas básicas no banco de dados, como acessar e obter dados. Para isso criou-se uma ação chamada de `GetCollection` no objeto do Blue Prism para realizar consultas no SQL Server

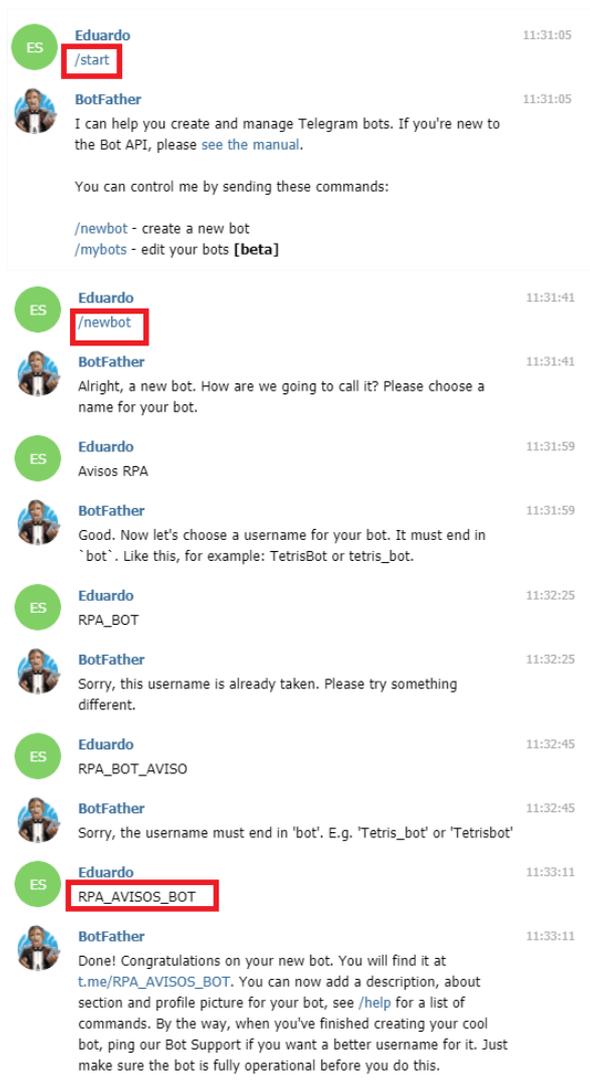
afim de obter informações para realizar o monitoramento das execuções das automações, o monitoramento das falhas de agendamento e o monitoramento das falhas dos recursos.

Para o funcionamento desse mecanismo, é necessário passar um parâmetro de entrada, que seria uma das instruções SQL (query) desenvolvidas na seção Mapeamento das informações no qual retornariam as informações da consulta. Além disso, foi criada uma agenda de execuções para que o mecanismo seja executado a cada 10 minutos de forma automática para obter as informações mais recentes para o monitoramento.

3.3.4 Compartilhamento das informações relevantes

Para realizar o compartilhamento das informações obtidas através do mecanismo desenvolvido, optou-se usar um aplicativo de mensagens instantâneas. Então foi necessário escolher o aplicativo para enviar mensagens contendo essas informações para a equipe de RPA. O aplicativo escolhido pela equipe de RPA foi o Telegram [19], por ser um serviço de mensagens instantâneas baseado na nuvem em que os integrantes da equipe utilizam frequentemente e por ter uma API aberta.

Inicialmente, criou-se um bot do Telegram. Bot é uma espécie de robô virtual programado para realizar determinadas funções. Com o bot é possível realizar solicitações HTTP para receber notificações com as informações obtidas a respeito de falhas dos robôs ou recursos. Para isso é necessário trocar mensagens com o BotFather e digitar `/start`, após digitar `/newbot` e depois digitar o nome do usuário do bot, como ilustrado na Figura 21.

Figura 21 – Telegram *NewBot*

As solicitações enviadas ao Bot API requerem a informação de um token, que pode ser obtido utilizando o comando `/token`. O próximo passo foi utilizar o Webservices REST fornecido pela própria Blue Prism, que é um utilitário que tem a capacidade de se conectar a APIs de serviço da web baseadas em REST para encontrar o chatID do bot. No modelo REST, uma requisição HTTP corresponde a uma chamada de um método em um recurso que é indicado na URL da requisição. O chatID é necessário para realizar o envio de mensagens, pois é o ID de identificação do chat do bot. Para isso foi realizada uma requisição HTTP usando a ação GET do objeto Webservices REST afim de obter o chatID, como parâmetro de entrada foi inserido a requisição HTTP no formato de URL `https://api.telegram.org/bot&[Token]/getUpdates`, como mostrado na Figura 22, então a requisição retornou o seguinte JSON conforme o Listing 1.3, onde o 'id' representa o chatID.

```
1 { "ok": true, "result": [ { "update_id": 242682897,
```

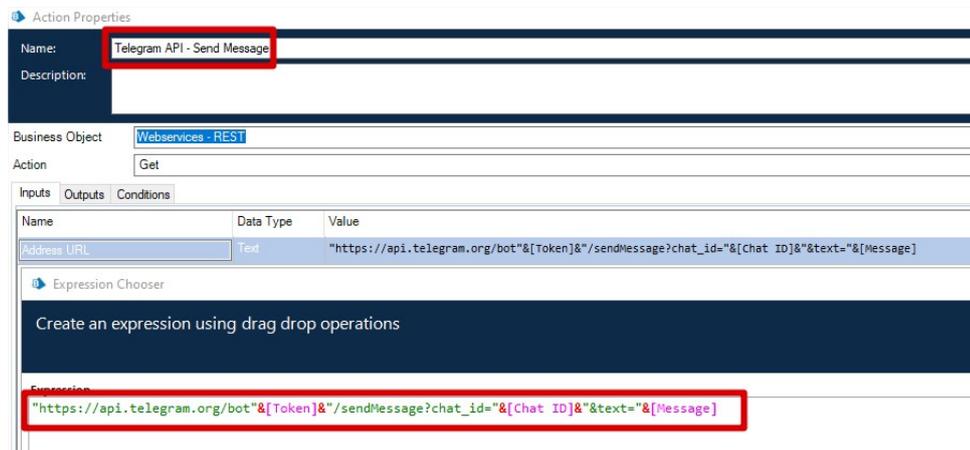
```
2 "message":{ "message_id":1, "from":{" id":-1001426511625, "is_bot":false, "
  first_name": "Eduardo", "last_name": "Sonohara", "language_code": "pt-br"}, "
  chat":{" id":-1001426511625, "first_name": "Eduardo", "last_name": "Sonohara"
  , "type": "private"}, "date":1606000288, "text": "/start", "entities":[{" "
  offset":0, "length":6, "type": "bot_command"}]}}
```

Listing 3.1 – JSON



Figura 22 – Telegram Chat ID

A comunicação entre o sistema de monitoramento e o Telegram é feita através de um objeto no Blue Prism chamado Telegram. Este objeto executa uma ação chamada SendMessage para realizar o envio de mensagens de texto utilizando o método sendMessage fornecido pela API do telegram, que tem o formato de uma URL para realizar a requisição HTTP: `https://api.telegram.org/bot\"[Token]\"/sendMessage?chat_id = \"[ChatID]\"text = \"[Message]`, como mostrado na Figura 23.

Figura 23 – Telegram método `sendMessage`

Caso a automação de monitoramento identifique algum problema durante a execução de um robô e que acabe resultando no término de sua execução, a mensagem enviada via telegram irá conter as informações como o título *Process Terminated*, o *Process Name* indicando o nome do robô, o status do robô, *Startdatetime* e *Enddatetime* indicando o horário de início e fim da execução do robô, *Starterusername* indicando o usuário que

iniciou o robô e *Resource* indicando o ambiente em que o robô foi executado, como mostra a Figura 24.

Caso a automação de monitoramento identifique alguma falha de agendamento de um robô, a mensagem enviada via telegram possuirá as informações como o título *Schedule Fail*, o *Schedule Name* indicando o nome do agendamento do robô, o status do robô, *Starttime* indicando o horário do agendamento do robô e *Resource* indicando o ambiente em que o robô foi programado para executar, como mostra a Figura 25.

Caso a automação de monitoramento identifique algum problema com os recursos, a mensagem enviada via telegram conterá as informações como o título *Resource Offline or Missing*, o status do recurso e *Resource* indicando nome do ambiente, como mostra a Figura 26.

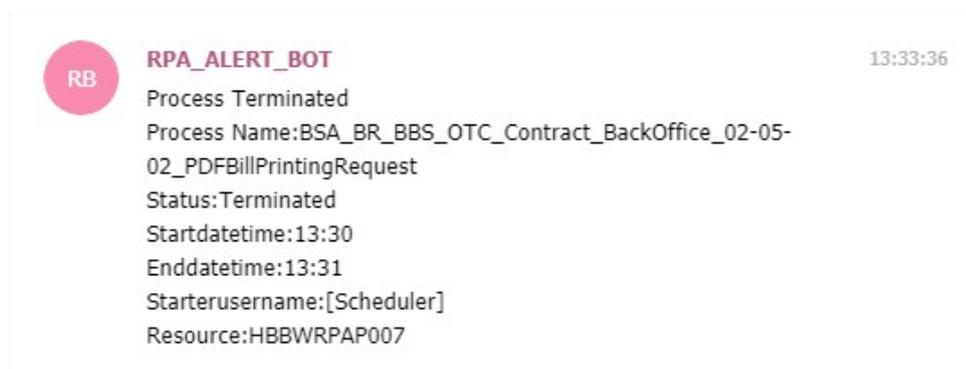


Figura 24 – Mensagem quando ocorrer o término da execução do robô

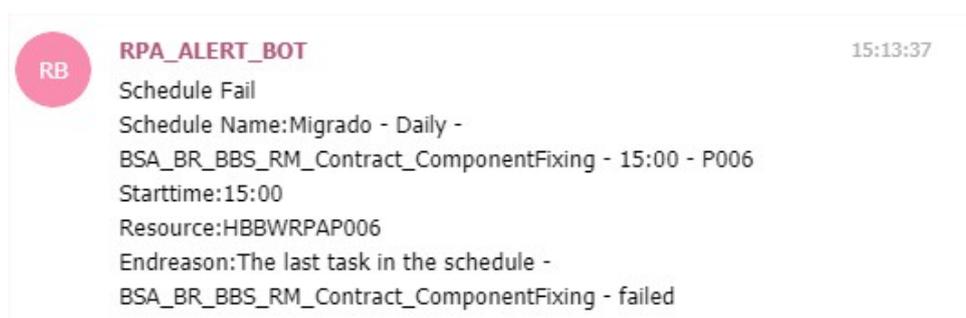


Figura 25 – Mensagem quando o robô não iniciar no horário programado



Figura 26 – Mensagem quando ocorrer um problema nos recursos

3.3.5 Implementação do robô

Após a identificação das informações necessárias para o monitoramento, o mapeamento das informações, o desenvolvimento de um mecanismo para obtenção das informações e o compartilhamento das informações relevantes, foi possível desenvolver um robô no Blue Prism para realizar o monitoramento de RPA de forma automatizada.

O projeto do robô desenvolvido pode ser visualizado na Figura 27. Basicamente o robô abre uma planilha de excel onde serão armazenadas as informações coletadas na consulta ao banco de dados. Na sequência é realizada a conexão com o banco de dados Microsoft SQL Server e executado as instruções SQL através das queries que foram criadas para obter as informações necessárias para o monitoramento. Então as informações adquiridas na consulta são manipuladas pelo robô para popular a fila de trabalho. Basicamente essas informações representam as falhas que foram identificados na consulta, como por exemplo um robô que não iniciou no horário programado. Em seguida o robô adquire o primeiro item da fila de trabalho para realizar o compartilhamento das informações através do Telegram e o preenchimento dessas informações na planilha de excel. Após completado essas tarefas, o robô adquire o próximo item e executa as mesmas tarefas até não existir mais itens na fila de trabalho.

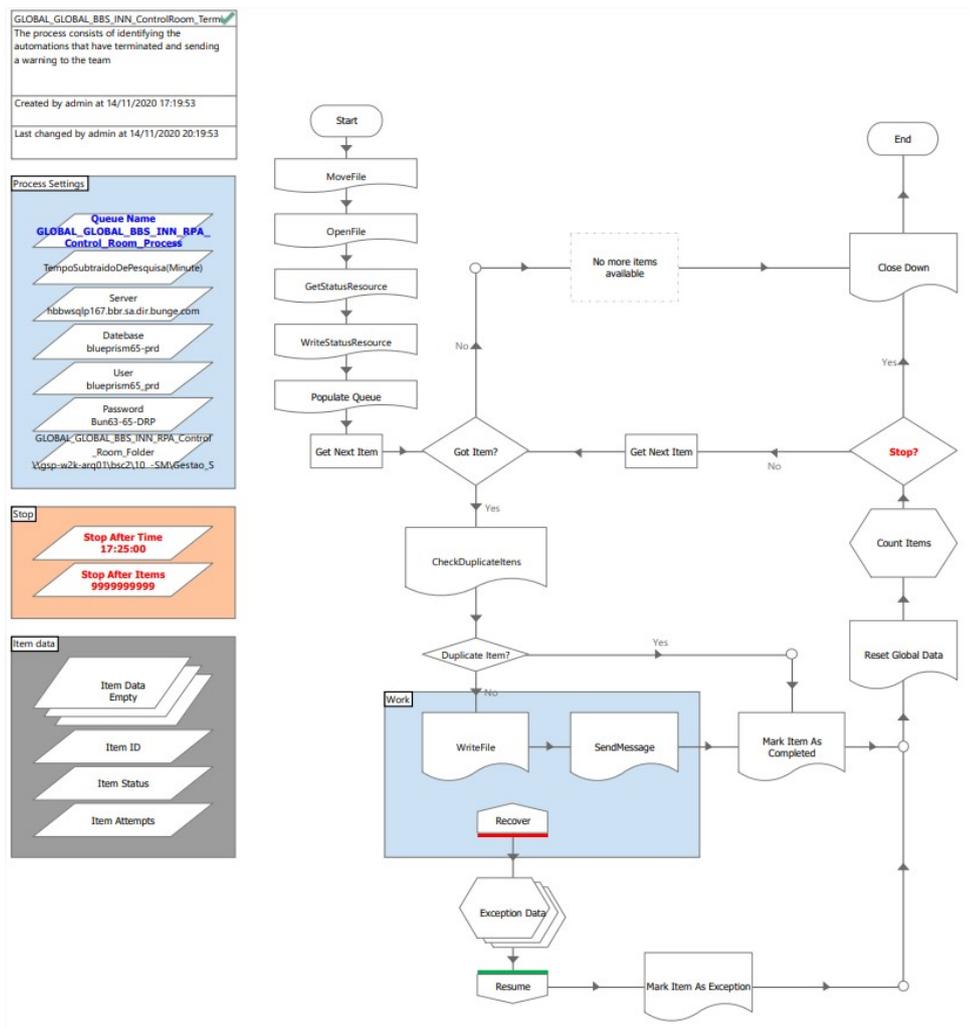


Figura 27 – Projeto Blue Prism

4 Resultados

Este capítulo apresenta o resultado da solução de automação para o monitoramento dos robôs RPA de forma automatizada após sua inserção em ambiente de produção. Também é apresentada uma análise de desempenho. Vale ressaltar que os dados quantitativos dos resultados foram omitidos, pois são informações sigilosas da empresa em foi desenvolvido o projeto. Os indicadores para a avaliação da solução desenvolvida são:

1. Tempo de execução;
2. Identificação de falhas;
3. Custo de execução.

Esse indicadores são utilizados pela empresa para mensurar a eficiência de um robô RPA.

Um dos benefícios de realizar a solução de RPA é que a a mesma possui uma função de medição de alguns indicadores, pois a própria fornece uma visão analítica do processo. A ferramenta Blue Prism possui uma base de dados de registros para cada robô que é executado na ferramenta, conseqüentemente os dados armazenados podem ser utilizados para a medição dos indicadores.

Para realizar a análise de desempenho foi preciso obter os dados das medições dos indicadores em duas etapas, sendo elas: antes da implementação de RPA e depois da implementação RPA.

4.1 Tempo de execução

Para realizar a análise do tempo de execução necessitou-se fazer uma análise com os funcionários da equipe de RPA para levantar o tempo médio que a atividade de monitoramento era realizada, obtendo-se um tempo médio de execução de aproximadamente 60 segundos.

A Figura 28 mostra as últimas execuções da automação de monitoramento durante o período do último mês. Ela possui os dados necessários para obter o tempo médio de execução. Então encontrou-se uma média de tempo de execução de 12 segundos.

Conclui-se que houve uma redução de 48 segundos de processamento, ou seja uma redução de 80% como mostrado na Figura 29. Essa redução significativa no tempo de execução está relacionada com a velocidade de manipulação e identificação dos dados com que o software Blue Prism realiza essas atividades, pois geralmente são realizados em segundos ou até mesmos milisegundos.

ID	Process	Resource	User	Status	Start Time	End Time
170036	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 23:30:39	30/04/2021 23:35:58
170021	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 20:30:52	30/04/2021 20:36:12
170015	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 20:03:57	30/04/2021 20:09:33
170010	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 19:56:53	30/04/2021 20:02:30
170003	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 19:43:48	30/04/2021 19:49:06
170000	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 19:37:04	30/04/2021 19:42:22
169993	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 19:23:48	30/04/2021 19:29:09
169990	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 19:17:02	30/04/2021 19:22:21
169983	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 19:04:25	30/04/2021 19:09:45
169978	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 18:57:06	30/04/2021 19:02:26
169969	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 18:34:17	30/04/2021 18:40:07
169964	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 18:27:03	30/04/2021 18:32:53
169956	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 18:12:56	30/04/2021 18:18:50
169951	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 18:05:32	30/04/2021 18:11:25
169941	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 17:25:27	30/04/2021 17:30:45
169938	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 17:23:34	30/04/2021 17:23:57
169927	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 16:55:49	30/04/2021 16:56:29
169924	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 16:53:41	30/04/2021 16:54:18
169908	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 16:00:50	30/04/2021 16:01:30
169903	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 15:58:40	30/04/2021 15:59:22
169894	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 15:21:52	30/04/2021 15:22:32
169890	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 15:19:34	30/04/2021 15:20:16
169880	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 15:08:35	30/04/2021 15:09:13
169877	GLOBAL_GLOBAL_BBS_INN_ControlRoom_Schedule	HBBWRPAP008	[Scheduler]	Completed	30/04/2021 15:06:21	30/04/2021 15:07:01

Figura 28 – Informações execuções Blue Prism

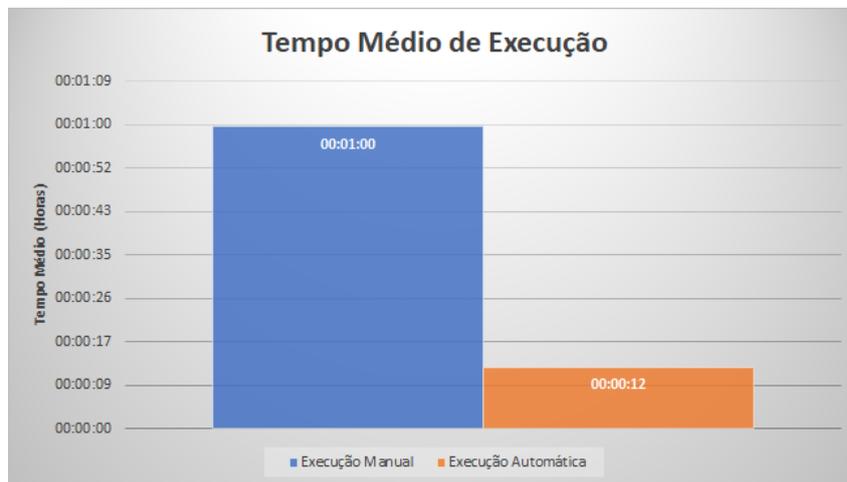


Figura 29 – Tempo Médio de Execução

4.2 Identificação de falhas

Com a implementação da solução de RPA para realizar o monitoramento de identificação de possíveis erros ou falhas nos robôs de RPA tornam-se mínimas as falhas dessas identificações. Pelo fato de que a consulta realizada pela automação é diretamente com o banco de dados do Blue Prism, permitindo a confiabilidade dos dados e uma melhor precisão das informações. Além disso, a solução desenvolvida sempre estará sendo executada repetidamente em intervalos de tempo pré-definidos que cobrirá as 24 horas do dia, e assim não permitindo falhas de identificação de erros ou falhas nos robôs de RPA. Diferentemente de um funcionário, que em certas ocasiões não identificará algumas falhas por estar realizando outras atividades pelas quais é responsável.

4.3 Custo de execução

A análise do custo de execução é feita com base no cálculo do FTE (Equivalente de tempo completo), que é uma métrica para calcular a média de horas úteis com que os trabalhadores se dedicam às suas atividades dentro da empresa. Logo a métrica representa o quanto que a solução da automação RPA trará de redução de custo para a empresa.

Para calcular o FTE são necessários o volume anual de execuções (V), tempo de execução (T) e a carga de trabalho do funcionário anual (C). Então para o monitoramento dos robôs de RPA levantaram-se os dados: V: 16200, T: 1 minuto e C: 2080 horas.

Logo, pode-se calcular o FTE com a seguinte equação:

$$FTE = V * (T/60)/C \quad (4.1)$$

$$FTE = 16200 * (1/60)/2080 \quad (4.2)$$

$$FTE = 0,13 \quad (4.3)$$

Portanto o FTE anual para realizar o monitoramento dos robôs de RPA é de aproximadamente 0,13. Levando em consideração que 1 FTE equivale a 260 dias úteis, esse valor representa que com a implementação da solução de RPA para essa atividade, trará um custo evitado de 34 dias úteis da carga de trabalho integral de um funcionário da equipe de RPA. Vale ressaltar que foi levado em consideração a carga de trabalho do funcionário pois o custo evitado trata-se de uma informações sigilosas da empresa.

5 Conclusão

O trabalho proposto teve como objetivo o desenvolvimento de uma solução para realizar o monitoramento dos robôs de RPA de forma automatizada. Esses robôs de RPA desempenham inúmeras atividades que eram realizadas pelos funcionários da empresa Bunge de diferentes áreas. O monitoramento buscou identificar erros durante a execução de um robô, falhas de agendamentos dos robôs e falhas nos ambientes onde os robôs são executados. Para isso necessitou-se estudar e levantar quais seriam as informações essenciais que pudessem identificar tais erros e falhas. Essa etapa foi um grande desafio pois essas informações eram desconhecidas.

Após identificado quais eram essas informações, foi possível desenvolver um mecanismo no software Blue Prism com o objetivo de fazer a comunicação com a API do Telegram em que permitiu realizar requisições HTTP. E assim foi possível compartilhar essas informações com as áreas interessadas de que houve algum erro ou falha.

O sistema de automação RPA desenvolvido foi satisfatório, pois do ponto de vista da força de trabalho, foi capaz de satisfazer as necessidades da equipe de RPA e suprir as tarefas monótonas, possibilitando os funcionários realizarem outras tarefas mais relevantes.

Além disso, pode-se observar que a utilização do RPA na execução da atividade mostrou-se ser mais eficiente do que se comparado com a atividade realizada por um operador humano. Como a redução no tempo de execução da atividade e conseqüentemente, foi possível aumentar a quantidade de execuções e também gerou uma redução de custo para a empresa.

Para trabalhos futuros, visando o aperfeiçoamento e melhoria da atividade de monitoramento dos robôs de RPA, sugere-se a implementação de um chatbot, que simula um ser humano na conversação com outras pessoas, com o objetivo de auxiliar o cliente a obter as informações do robô RPA de forma automatizada e em tempo real. Como por exemplo a consulta do status do robô se já foi iniciada no horário programado, a última e a próxima execução do robô RPA, entre outras funcionalidades que podem ser programadas.

A Apêndices

A.1 Mecanismo para obtenção das informações

O primeiro passo foi criar um novo objeto na ferramenta Blue Prism e importar o namespaces (System.Data.SqlClient), pois é necessário quando utilizado banco de dados em páginas ASP.NET. Para realizar a conexão com o banco de dados foi utilizado a classe SqlConnection fornecido pelo ADO.NET e assim criado o seguinte código:

```

1 Success = True
2 Message = " "
3
4 Try
5     //Realiza a conexao com o banco de dados
6     moConnection = New SqlConnection(ConnectionString)
7     //A conexao aberta via metodo Open
8     moConnection.Open()
9
10 Catch ex As Exception
11     Success = False
12     Message = ex.Message
13
14 End Try

```

Listing A.1 – Conexão com banco de dados

Após criado o código de conexão, foi criado a ação SetConnection no objeto do Blue Prism para realizar a conexão com o banco de dados. Para isso foi utilizado o code stage, que utilizada a linguagem VisualBasic. Está ação possui parâmetros de entrada database, data source, Server, user id e password, que é utilizada na string ConnectionString, como mostra a Figura 30.

O próximo passo foi encontrar uma forma para obtenção dos dados no banco de dados, a alternativa encontrada foi utilizar a classe SqlCommand que é responsável por executar instruções SQL, a classe SqlTransaction que tem a função de reverter a transação caso ocorrer qualquer erro, a classe DataSet que representa um conjunto de dados em memória e também a classe SQLDataAdapter que tem a função de realizar a ponte entre o banco de dados e o DataSet, então foi criado o seguinte código:

```

1 Try
2     Using cmd As New SqlCommand()
3         cmd.Connection = moConnection
4         cmd.Transaction = moTransaction
5         cmd.CommandText = SQL
6

```

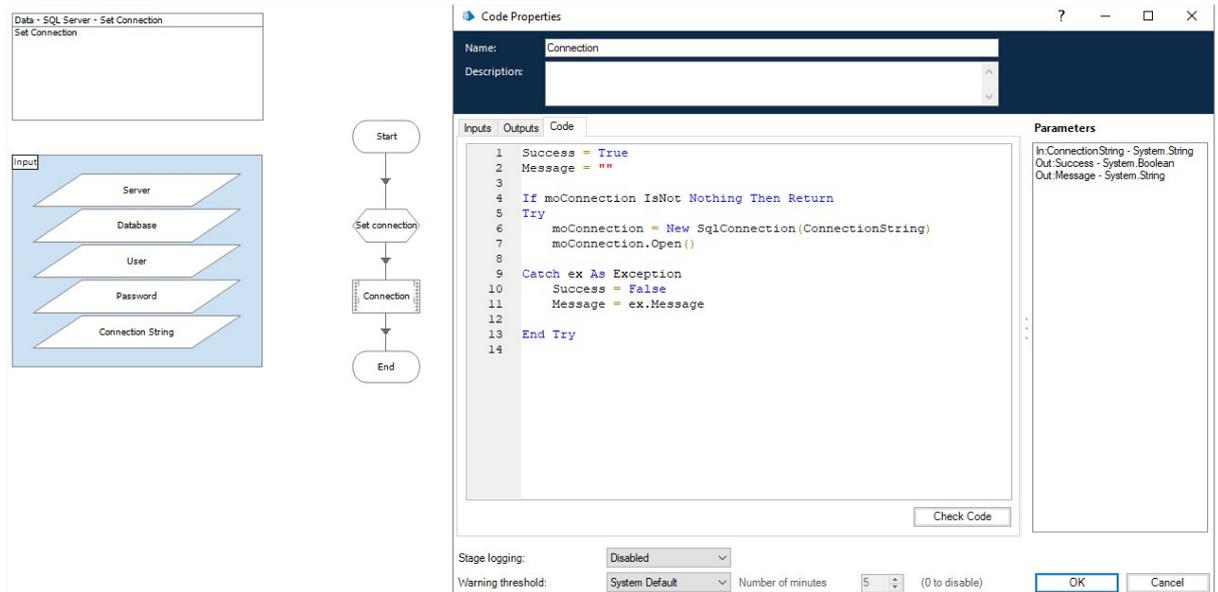


Figura 30 – Ação Set Connection Blue Prism

```

7 Using adapter As New SqlDataAdapter()
8     adapter.SelectCommand = cmd
9     Dim ds As New DataSet()
10    adapter.Fill(ds)
11    Results = ds.Tables(0)
12 End Using
13 End Using
14
15 Catch ex As Exception
16     Success = False
17     Message = ex.Message
18
19 End Try

```

Listing A.2 – Obtenção de dados

No código de obtenção de dados será realizada a conexão com o banco de dados e executada as instruções descritas na string SQL. Após, é criado um DataSet chamado ds para armazenar os dados e inicializado a conexão do DataSet com o bando de dados. Então é usado o método SqlDataAdapter.Fill para armazenar os dados no DataSet ds.

Além disso, criou-se a ação GetCollection no objeto do Blue Prim. Foi utilizado o code stage para inserir o código de obtenção de dados, essa ação possui a string SQL como entrada, que contém as instruções SQL que foram obtidas na seção Mapeamento das informações, e possui a saída Results que será o resultado dessas instruções, como mostrado na Figura 31.

The screenshot displays the 'Code Properties' window for the 'Get Data Table' action in Blue Prism. On the left, a flowchart shows the process flow: Start -> Open -> Get Data Table -> Raise Exception If Appropriate -> End. A decision diamond 'Success?' follows the 'Open' step, with a 'Yes' path leading to 'Get Data Table' and a 'No' path leading to 'Raise Exception If Appropriate'. Below the flowchart are two blocks: 'Block1' containing 'SQL' and 'Results Empty', and 'Block2' containing 'Success False' and 'Message'. The main window features a code editor with the following C# code:

```

1
2 ' Assume success
3 Success = True
4 Message = ""
5
6 Try
7     Using cmd As New SqlCommand()
8         cmd.Connection = moConnection
9         cmd.Transaction = moTransaction
10        cmd.CommandText = SQL
11
12        Using adapter As New SqlDataAdapter()
13            adapter.SelectCommand = cmd
14            Dim ds As New DataSet()
15            adapter.Fill(ds)
16            Results = ds.Tables(0)
17        End Using
18    End Using
19
20 Catch ex As Exception
21     Success = False
22     Message = ex.Message
23
24 End Try
    
```

At the bottom of the window, there are settings for 'Stage logging' (Disabled) and 'Warning threshold' (System Default, 5 minutes). The 'Parameters' section on the right lists: In: SQL - System.String, Out: Success - System.Boolean, Out: Message - System.String, and Out: Results - System.Data.DataTable.

Figura 31 – Ação GetCollection Blue Prism

Referências Bibliográficas

- 1 BRASIL, M. da S. *Lesões por Esforços Repetitivos – LER*. 2021. Acesso em: 20 de ago. de 2021. Disponível em: <https://bvsmms.saude.gov.br/bvs/dicas/174_ler.html>. 11
- 2 IEEE Guide for Terms and Concepts in Intelligent Process Automation. *IEEE Std 2755-2017*, p. 1–16, 2017. 11
- 3 FORRESTER. *Building A Center Of Expertise To Support Robotic Automation Preparing For The Life Cycle Of Business Change*. [S.l.], 2014. 11
- 4 LACITY M.; WILLCOCKS, L. P. C. A. Robotic process automation at telefónica o2. the outsourcing unit working research paper series, 15/02. *Blue Prism*, 2015. 11
- 5 BUNGE. *Bunge | Agronegócio | Alimentos e Bioenergia*. Acesso em: 20 de ago. de 2021. Disponível em: <<https://www.bunge.com.br/Default.aspx>>. 12
- 6 BUNGE. *A Bunge: Perfil*. Acesso em: 20 de ago. de 2021. Disponível em: <<https://www.bunge.com.br/Bunge/Perfil.aspx>>. 12
- 7 WEISS, E. Biographies: Eloge: Arthur lee samuel (1901-90). *IEEE Annals of the History of Computing*, v. 14, n. 3, p. 55–69, 1992. 13
- 8 GARTNER. *Market Share Analysis: Robotic Process Automation, Worldwide, 2020*. Acesso em: 20 de ago. de 2021. Disponível em: <<https://www.gartner.com/en/documents/4001926/market-share-analysis-robotic-process-automation-worldwi>>. 13
- 9 GARTNER. *Gartner Says Worldwide Robotic Process Automation Software Revenue to Reach Nearly \$2 Billion in 2021*. Acesso em: 20 de ago. de 2021. Disponível em: <<https://www.gartner.com/en/newsroom/press-releases/2020-09-21-gartner-says-worldwide-robotic-process-automation-software-revenue-to-reach-nearly-2->>. 13, 14
- 10 OSWAL, P. J. S. Recent trends in automation: Robotic process automation. In: *International Conference on Recent Trends of Engineering and Science 2020 (ICRTES2020)*. [S.l.: s.n.], 2020. p. 1–8. 14
- 11 KENOBY. *FTE: O que é e como calcular o Full Time Equivalent da empresa?* Acesso em: 20 de ago. de 2021. Disponível em: <<https://kenoby.com/blog/fte>>. 15
- 12 ANYWHERE, A. *Líder no Quadrante Mágico para RPA | Automation Anywhere*. Acesso em: 20 de ago. de 2021. Disponível em: <<https://www.automationanywhere.com/br/lp/gartner-magic-quadrant>>. 16
- 13 ACOBA ABBY LEVINE, A. T. F.; KAPLAN, D. Corporate real estate and facilities management in the digital world. *deloitte*, 2019. 17
- 14 SAP. *Sobre | a SAP Brasil*. Acesso em: 20 de ago. de 2021. Disponível em: <<https://www.sap.com/brazil/about.html>>. 18

- 15 FERREIRA, O. K. T. I. C. I. J. E. *INTRODUÇÃO A BANCO DE DADOS*. [S.l.], 2005. 19, 20
- 16 HEUSER, C. A. *Projeto de Banco de Dados*. [S.l.]: Bookman, 2008. (6ª edição). ISBN 8577803821. 22
- 17 MACHADO, D. *Normalização em Bancos de Dados*. Acesso em: 20 de ago. de 2021. Disponível em: <<https://medium.com/@diegobmachado/normalizaç~ao-em-banco-de-dados-5647cdf84a12>>. 23, 25
- 18 HALVORSEN, H.-P. *Structured Query Language*. [S.l.], 2017. 25
- 19 TELEGRAM. *Telegram*. Acesso em: 20 de ago. de 2021. Disponível em: <<https://telegram.org/>>. 40