

# Implementação de RPA para Automação de processos administrativos em ambiente industrial utilizando a plataforma UiPath

Adriano Mauricio De Bastiani

Rafael Vieira Coelho

Email: adrianocs4@gmail.com

*Engenharia de Controle e Automação*

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – Câmpus Farroupilha  
Farroupilha, Brasil

**Resumo**—A automação de processos repetitivos é uma tendência no ambiente corporativo, especialmente em setores administrativos com grande volume de tarefas padronizadas. Este trabalho propõe o desenvolvimento de uma solução baseada em Robotic Process Automation (RPA), utilizando a plataforma UiPath, voltada para a área de manutenção industrial da empresa Castertech. O objetivo é reduzir a carga de trabalho burocrático da equipe administrativa, aumentar a eficiência dos processos e minimizar erros operacionais. O RPA deve integrar com os sistemas SAP e Office 365, sendo capaz de executar tarefas como coleta e registro de ordens de serviço, emissão de notas fiscais e controle de estoque. Além de apresentar uma arquitetura modular e escalável, o projeto aplica conceitos de low-code para facilitar manutenções futuras e adaptações ao ambiente corporativo.

**Palavras-chave**—RPA, Low-Code, Eficiência Operacional, Automação de Processos

## I. INTRODUÇÃO

Devido à complexidade das operações empresariais, organizações de grande porte tendem a utilizar sistemas ERP(Enterprise Resource Planning) para gerenciar seus recursos [1]. São sistemas integrados de gestão empresarial que reúnem múltiplas áreas e processos da organização, fornecendo rastreamento e visibilidade global da informação de toda a cadeia de suprimento [2], facilitando a tomada de decisões e aumentando a eficiência operacional.

Um dos ERPs mais utilizado é o sistema SAP [3]. Ele possui as ferramentas para realizar a gestão de grande corporações, em alguns casos sendo o único software utilizado. Porém essa gama de ferramentas torna o SAP um sistema extremamente complexo e de difícil utilização para um usuário não capacitado. Além disso, no dia a dia de utilização do sistema, ele é repleto de atividades repetitivas com pouco espaço para erro do usuário.

Esses problemas acabam sendo amplificados no cenário de manutenção industrial onde respostas rápidas e efetivas são extremamente necessárias. Trata-se de uma área onde gestores necessitam balancear seus recursos entre mão de obra técnica e administrativa. Uma solução que esta ganhando força nos últimos anos é a automatização de processos utilizando-se

RPA(Robotic Process Automation). Este é definido como a aplicação de tecnologias e metodologias baseadas em software e algoritmos, com o objetivo de automatizar tarefas repetitivas [4].

Tendo em vista todos os pontos destacados, esse artigo se propõe a projetar um RPA para automatizar processos repetitivos da equipe administrativa de manutenção da empresa Castertech [5], com aproximadamente 1000 funcionários. A tecnologia escolhida foi o software UiPath [6] que possui integração direta com os softwares utilizados (SAP, Office 365) e indireta com sites da internet [7].

## II. OBJETIVO GERAL

Projetar um sistema RPA para automatização processos administrativos em uma grande empresa de grande porte.

### A. Objetivos específicos

- Trabalhar de forma autônoma com o mínimo de atuação humana possível (24 horas por dia);
- Reduzir e padronizar atividades burocráticas da equipe de manutenção.
- Utilizar os sistemas: SAP e Office365 online;
- Copiar tabelas em excel do sistema SAP para geração de relatórios de manutenção;
- Coletar as ordens de manutenção geradas no sistema de manutenção e replicar as informações no sistema SAP;
- Criar requisições de compras no sistema SAP conforme solicitações encontradas em uma tabela do Excel online;
- Gerar Notas Fiscais em PDF para remessas de conserto externo conforme solicitações encontradas em uma tabela do Excel online e deixar as mesmas disponíveis em uma pasta online;
- Revisar a quantidade de peças em estoque e lançar solicitações de compra dos itens faltantes, levando em consideração itens que já estejam em compra.

## III. REVISÃO BIBLIOGRÁFICA

Em [8] o trabalho aborda o processo de implementação de RPA para otimizar a eficiência operacional em organizações.

Em sua implementação eles levantam a necessidade de analisar cada processo existente e identificar alvos específicos para automação dando foco para trabalhos repetitivos e com alto potencial de erro humano. Também levantam a necessidade de avaliar a matriz esforço e impacto da automação de cada processo, levando em consideração 3 itens: complexidade; volume de trabalho; e possíveis benefícios para priorizar a implementação de cada automatização.

O TCC [9] é um estudo da utilização do software UIPath [6] para procurar candidatos aptos a uma vaga de emprego na aplicação web LinkedIn conforme especificações de um agente de recursos humanos. Neste trabalho, ele demonstra como a ferramenta pode ser programada para utilizar navegadores de Internet; abrir páginas da Internet; fazer login em sites; digitar textos em caixas de diálogo; clicar em botões ou *checkboxes*; coletar e armazenar informações da página na forma de variáveis ou tabelas de dados.

O artigo da revista Muhasebe ve Finansman Dergisi [10] faz um estudo do impacto da implementação de RPA em sistemas de contabilidade. Nele são enumeradas diversas tarefas onde um sistema RPA pode ser utilizado como automação de processos de pagamentos e envio automático de e-mails. Os benefícios elencados foram a redução de custos, diminuição do tempo de resposta das instituições e diminuição de erros. Eles comentam que grande parte das atividades financeiras será realizada por RPAs no futuro, especialmente quando se trata de atividades menos eficientes. Isso liberará espaço para os profissionais de contabilidade focarem em operações estratégicas.

Além destes, o artigo [11] apresenta uma pesquisa feita em duas grandes corporações (uma Escandinava e outra Francesa) sobre os motivos, benefícios, riscos e impactos na escolha do modelo de implementação dos RPAs. A primeira utilizando mão de obra terceirizada e a segunda com a criação de um time interno para desenvolvimento e implementação de RPAs.

No primeiro caso, a implementação inicialmente operou conforme o esperado. No entanto, a dependência de uma equipe terceirizada trouxe desafios imprevistos. Toda vez que os sistemas utilizados eram atualizados, surgia um custo adicional para adaptar o RPA. Além disso, a equipe local perdeu conhecimento sobre os processos automatizados já que os funcionários internos deixaram de executar essas funções. Já a segunda empresa, visando preservar a confidencialidade de seus processos, decidiu formar uma equipe interna para desenvolver seus próprios robôs. Essa escolha trouxe melhores resultados a longo prazo. No entanto, a curto prazo, a inexperience da equipe gerou desafios, como decisões equivocadas na abordagem dos robôs. Esses erros resultaram em falhas graves, levando os robôs a pararem completamente ou a operarem de forma incorreta, o que causou danos irreparáveis aos dados de produção. Por fim, o estudo concluiu que a implementação de RPA, em geral, é benéfica, porém os riscos a longo prazo ainda são um tópico pouco explorado.

Por fim o artigo da revista SSRN explica como implementar um RPA robusto e de pouca manutenção em ambientes corpo-

rativos [12]. Nele é explicada a infraestrutura necessária para criar um sistema de RPA que incluiu recuperação de erros, operação contínua e aberto para a ampliação das operações. O estudo fala de conceitos importantes na criação de robôs como filas de trabalho, que se trata de uma lista de tarefas que o robô preenche no começo de uma sessão de trabalho, após preenchida essa lista ele começa a realizar as atividades com os dados contidos nela um item de cada vez. Isso traz vários benefícios como a habilidade do robô recomeçar do último item da fila quando ele para por algum motivo inesperado e a possibilidade de múltiplos robôs utilizarem a mesma fila de trabalho sem se atrapalharem.

Ainda no trabalho de [12] em relação à recuperação de erros, eles comentam sobre os principais tipos, *Business Exceptions*, ocorre quando o robô encontrou um caso que não consegue processar por falta de implementação ou devido à necessidade de julgamento humano para tratar do erro. O outro são *System Exception* que é relacionado a erros de sistema, como sistemas externos que não respondem e interface programada não presente ou diferente do esperado. Quando um desses erros ocorre, é importante implementar maneiras de contorná-los sem causar uma parada total da automação. Para erros comuns é possível deixar pré-programado o processo de correção quando identificado através de alguns gatilhos. Já para os demais erros deve-se implementar meios de avisar os usuários que intervenção humana é necessária para finalização daquele processo específico. Com isso em mente e a utilização de fila de tarefas é possível que o robô ao identificar um erro apenas pule para a próxima tarefa ao invés de parar totalmente.

## IV. FUNDAMENTAÇÃO TEÓRICA

### A. RPA

RPA é uma tecnologia emergente que permite a automação de tarefas repetitivas dentro de processos empresariais [13]. Utilizando softwares chamados de robôs, eles têm capacidade de imitar ações humanas em sistemas digitais, interagindo como se fosse um usuário humano. Ele pode abrir programas e páginas da internet, ler e escrever textos, clicar em campos pré-determinados, utilizar atalhos do teclado, etc. Por operar diretamente sobre a interface dos sistemas, RPAs geralmente dispensam integrações complexas com as aplicações existentes, o que a torna uma solução tecnicamente viável e economicamente acessível para diversas organizações.

A principal vantagem do RPA é o aumento da eficiência operacional, quando tarefas longas e repetitivas são automatizadas colaboradores podem focar em atividades com maior valor agregado [14] como análises estratégicas, tomada de decisões e atuar em áreas com falta de recursos humanos. Além disso a precisão dos robôs minimiza erros humanos, resultando em aumento da qualidade e consistência do processo. Por fim facilita a transformação digital nas empresas, abrindo a possibilidade de conectar vários sistemas diferentes sem a necessidade de realizar grandes estruturas de TI, utilizando interfaces já existentes.

Quanto à implementação, os RPAs podem ser classificados em dois tipos: assistido, que exige a

intervenção do usuário para iniciar a operação ou em determinados momentos da tarefa; e não assistido, que executa suas atividades de forma totalmente autônoma, sem a necessidade de intervenção humana [15]. Cada tipo pode ser adequado a diferentes contextos organizacionais, dependendo da necessidade de automação. Por exemplo, robôs assistidos são boas escolhas para tarefas que são feitas ocasionalmente, mas que demandam alta dedicação do trabalhador ou casos onde existem algumas variáveis pré-determinadas (que são informadas pelo usuário durante o processo de inicialização). Para os demais casos, é recomendado o uso de robôs não assistidos.

### B. Low-code

Low-code refere-se a plataformas de desenvolvimento de software que enfatizam interfaces visuais, possibilitando que pessoas com pouco conhecimento técnico criem aplicativos de negócios de forma simples e intuitiva [16]. Essa abordagem tem como objetivo principal agilizar e diminuir custos durante o desenvolvimento de aplicações em contextos empresariais, eliminando a dependência de especialistas em programação. Com essas ferramentas, usuários sem experiência avançada podem desenvolver aplicativos para dispositivos móveis ou computadores, promovendo a transformação digital nas empresas de maneira eficiente.

A principal característica das plataformas low-code é que a programação é realizada majoritariamente por meio de interfaces gráficas [17]. Os usuários têm à sua disposição uma variedade de blocos pré-definidos com funções padronizadas, que podem incluir menus de opções ou caixas de diálogo com escolhas limitadas para ajustar sua funcionalidade. Em alguns casos, o desenvolvimento ocorre conectando esses blocos em sequência, de forma semelhante a um fluxograma. Por conta dessas características, as plataformas restringem as possibilidades do usuário aos blocos fornecidos pelo desenvolvedor, mas, graças à sua simplicidade, tornam-se excelentes ferramentas para uso em ambientes empresariais.

A Figura 1 é um exemplo de como uma plataforma low-code pode funcionar. Uma sequência de atividades contida em blocos com configuração limitada, como um seletor de função, configurações da função e um espaço para personalização do usuário. O desenvolvimento é realizado escolhendo os blocos de uma lista fornecida pela plataforma, realizando a configuração dos mesmos e distribuindo-os em sequência ou no formato de fluxograma.

### C. Seletores CSS

Uma página de web consiste em diferentes layouts contendo menus, botões, títulos, textos, imagens, etc. Seletores *Cascading style sheets* (CSS) são padrões utilizados para identificar e estilizar elementos específicos em uma página HTML [18], desempenhando um papel fundamental no desenvolvimento web. Eles permitem que os desenvolvedores apliquem regras de estilo com base em elementos, classes, IDs, atributos ou pseudo-classes, garantindo precisão e flexibilidade na formatação visual.

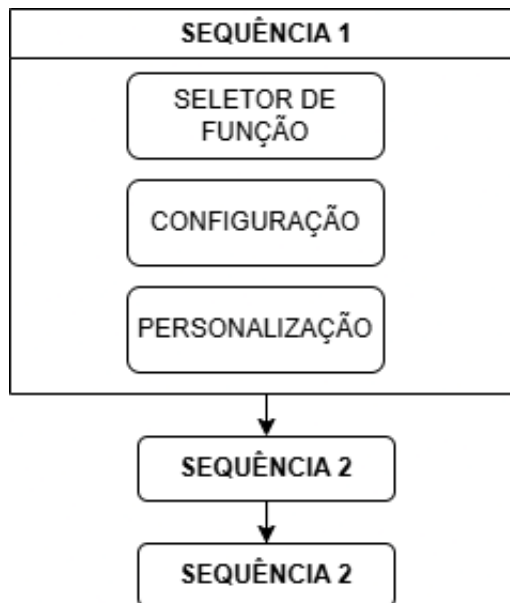


Figura 1: Exemplo Low-Code  
Fonte: Autoria própria

Normalmente seletores CSS são utilizados para impor regras e formatações em elementos específicos de sites e aplicações web. No contexto desse projeto eles terão o papel crucial de isolar elementos pré-determinados de sites para que o robô consiga-os identificar e interagir com os mesmos de forma consistente. Como por exemplo na figura 2 o seletor `div.KxwPGc.iTjxkf` representa o menu de opções do site.

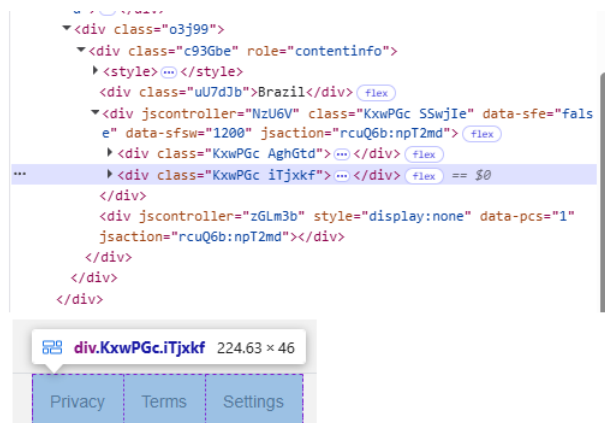


Figura 2: Exemplo CSS selector  
Fonte: Autoria própria

### D. Uipath

Uipath é uma poderosa plataforma de automação de processos desenvolvida pela Uipath Inc., uma empresa fundada em 2005 na Romênia [6]. O Uipath permite a criação de RPA, como explicado anteriormente que simula ações humanas em sistemas digitais. Ele é composto por três módulos principais: Uipath Studio, onde é desenvolvido a automação, UiPath

Orchestrator, que gerencia as automações criadas e UiPath Robot que executa as tarefas criadas no Studio [19].

1) *UiPath Studio*: Plataforma de low-code usada para criar os robôs que são controlados pelos outros 2 módulos. O desenvolvimento é feito criando processos e adicionando atividades em sequências ou fluxogramas dentro desses processos. Cada atividade é um bloco com uma função padronizada, podendo ser ela clicar em um botão da tela, extrair um texto, criar uma tabela de dados, etc. Para inserir uma atividade, basta arrastá-la para a tela principal e preencher as informações necessárias nas propriedades do bloco utilizando a linguagem C#. Caso o usuário não tenha conhecimento o *software* possui uma inteligência artificial embutida que auxilia no preenchimento das informações.

Em atividades onde é necessário interagir com um elemento da tela, o studio detecta-o via um *selector* indicado pelo usuário que é dividido em vários estágios de detecção. Inicialmente, ele tenta detectar o elemento de maneira restrita em que todas as informações daquele ponto da tela sejam idênticas ao indicado pelo usuário. Caso não consiga identificar ele busca os elementos com informações semelhantes que estejam próximos do elemento de ancora. Se este método não funciona, a detecção parte para a visão do computador, que é um modelo de inteligência artificial que procura elementos parecidos com o indicado pelo usuário na tela. Por fim, ele tenta fazer a busca por imagem, procurando elementos visualmente parecidos com os indicados pelo usuário. Caso nenhum desses métodos funcione ele retorna um erro que caso não seja tratado pelo usuário pode bloquear o processo.

2) *UiPath Robot*: O UiPath Robot é a ferramenta responsável por executar os processos de automação desenvolvidos no UiPath Studio. Após a publicação do processo no Orchestrator (ambiente de gerenciamento centralizado da plataforma), o robô pode ser acionado de duas formas: manualmente, por meio do aplicativo instalado na máquina local; ou remotamente, por meio do Orchestrator. Quando acionado, o UiPath Robot inicia a execução do processo de forma autônoma, interagindo com as interfaces dos sistemas envolvidos como se fosse um usuário humano. Durante a execução, o robô assume o controle do computador, utilizando teclado e mouse de forma programada, garantindo a reprodução fiel das atividades automatizadas. Ao final do processo, o Robot também pode gerar registros de log e erros, que auxiliam no monitoramento e análise da execução [7].

3) *UiPath Orchestrator*: É a plataforma online que organiza as atividades dos robôs. Nela são organizados as filas de atividades de todos os robôs ativos. Também pode guardar informações como senhas de acesso a outros sistemas utilizadas pelos robôs, evitando a necessidade de ter informações sigilosas no código fonte dos mesmos.

A integração entre *Robotic Process Automation* (RPA), plataformas *low-code* e a ferramenta UiPath representa uma convergência estratégica para a automação de processos empresariais de forma ágil e acessível. O RPA permite a automação de tarefas repetitivas por meio da imitação de interações humanas em interfaces digitais, dispensando integrações complexas e

promovendo eficiência operacional e redução de erros. Já as plataformas *low-code* facilitam o acesso ao desenvolvimento ao oferecer ambientes visuais baseados em blocos pré-configurados, possibilitando que usuários com baixo conhecimento técnico construam soluções robustas. Nesse contexto, o UiPath se posiciona como uma solução consolidada que une ambos os conceitos: seu módulo *UiPath Studio* opera como uma plataforma *low-code* para a criação de fluxos de automação, enquanto os módulos *Robot* e *Orchestrator* garantem a execução autônoma e o gerenciamento centralizado dos robôs. Essa sinergia potencializa a transformação digital, permitindo que organizações implementem automações escaláveis com menor dependência de equipes de TI especializadas e maior foco em atividades estratégicas.

## V. PROPOSTA DE PROJETO

O diagrama apresentado na Figura 3 ilustra a sequência das principais operações que irão compor a arquitetura do RPA, a ser desenvolvida na plataforma UiPath. As etapas planejadas são: inicialização (A); obtenção de dados de transação (B); processamento (C); e finalização (D).

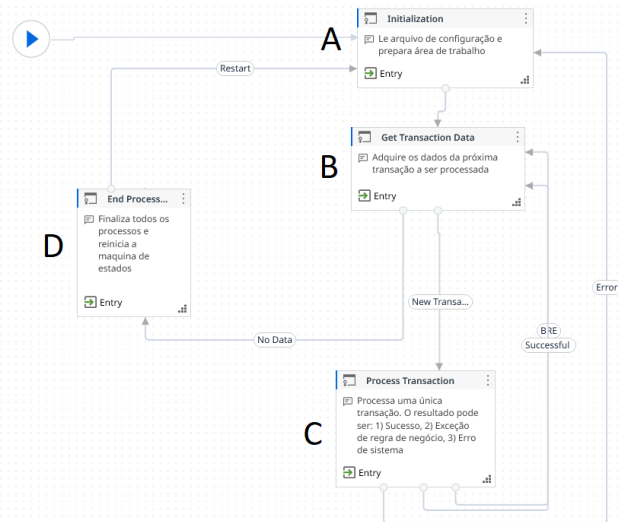


Figura 3: Fluxograma do funcionamento do RPA

Fonte: UiPath

A primeira etapa, denominada **inicialização**, consiste na execução de atividades essenciais para garantir o funcionamento correto do robô nas fases seguintes. Inicialmente, devem ser carregadas informações como: localizações de pastas, URLs dos sites a serem acessados, nome da fila de trabalho, endereços de e-mail, entre outros. Esses dados são armazenados em uma variável do tipo dicionário, permitindo que o robô os consulte sempre que necessário.

Uma prática comum consiste em utilizar uma planilha eletrônica contendo essas informações e, por meio de uma atividade de leitura, carregar cada uma de suas linhas em uma variável do tipo dicionário. A principal vantagem desse método reside em sua flexibilidade, pois, caso algum parâmetro necessite de alteração, o usuário pode modificar o conteúdo da

planilha, sem que haja necessidade de mudanças no código-fonte do robô.

A segunda atividade da etapa de inicialização envolve a preparação do ambiente de trabalho. Para isso, o robô deve executar uma sequência de comandos destinados a encerrar todos os processos dos softwares utilizados, como Excel, SAP e navegadores. Essa prática visa prevenir falhas na execução, como a tentativa de abrir aplicações que já estejam em uso e evitar que múltiplas cópias do mesmo software fiquem abertas ao mesmo tempo, afetando o resto da operação.

Por fim, a última ação da inicialização é o carregamento das tarefas na fila de trabalho hospedada na nuvem do Orchestrator. Essa subetapa é denominada **coletor**. Nela, o robô acessa os sistemas da empresa, identifica as atividades pendentes para execução automatizada, extrai os dados necessários e cria um item na fila para cada tarefa. Esses itens, denominados **transações** segundo a nomenclatura adotada pela documentação oficial do UiPath, pode ser acompanhados por meio do menu de filas do Orchestrator [7].

A segunda operação, denominada **obter dados de transação** (B), consiste em acessar a fila de trabalho, identificar a próxima transação a ser executada, extrair suas informações e enviar os dados de execução para a próxima etapa. Em seguida, o robô atualizar o status da transação para “em progresso”, prática comum em soluções de automação, com o objetivo de evitar que a mesma transação seja acessada simultaneamente por mais de um robô, caso múltiplos agentes estejam operando em paralelo. Essa atualização também permite a retomada do processo em caso de falhas e viabiliza o monitoramento em tempo real por parte dos usuários. Caso não existam transações disponíveis, o robô segue diretamente para a etapa de finalização.

A terceira operação é o **processamento** (C). Nessa fase, o robô deve simular o comportamento de um usuário humano, executando as tarefas conforme um roteiro previamente definido. Para isso, é necessário que a equipe da empresa forneça o detalhamento completo de cada processo a ser automatizado, incluindo a ordem das ações, os dados exigidos em cada etapa, os critérios de conclusão e as tratativas para exceções. Esse detalhamento pode ser documentado por meio de gravações em vídeo, fluxogramas, relatórios descritivos ou uma combinação desses métodos, dependendo da complexidade do processo. Esse conjunto de instruções é conhecido na indústria como *Process Design Document* (PDD) [7].

O desenvolvimento da etapa de processamento segue uma abordagem modular, na qual cada atividade é implementada em um módulo independente, criado conforme o roteiro estabelecido no PDD. A execução de cada módulo depende das informações recebidas da etapa anterior. Essa estratégia tem como objetivo evitar a necessidade de criar múltiplos robôs distintos para cada tipo de atividade, permitindo que um único robô seja capaz de lidar com diferentes tarefas de forma flexível e escalável.

Exemplos de tarefas passíveis de automação incluem: abertura de ordens de serviço, emissão de notas fiscais, geração de planilhas com dados empresariais, entre outros. Após concluir

uma transação, o robô atualiza o status do item correspondente na fila com o resultado obtido e retorna à etapa anterior para verificar a existência de novas transações. Caso não haja, ele avança para a etapa final (D).

A última operação, **finalização** (D), compreende o encerramento das aplicações utilizadas pelo robô, a exclusão de arquivos temporários gerados durante a execução e, por fim, a entrada em modo de hibernação por um período estimado de uma hora. Após esse intervalo, o sistema retorna automaticamente à etapa de inicialização, reiniciando o ciclo de operação.

## VI. DESENVOLVIMENTO

O primeiro passo do projeto consiste na criação da máquina de estados conforme ilustrado na Figura 3, seguindo o sequenciamento definido na seção anterior. O processo envolve a criação de um bloco de estado para cada etapa na plataforma *UiPath*, com a adição de transições condicionais entre eles. A Figura 4 exemplifica a transição após a etapa “Adquirir dados de transação”: caso o RPA colete um item da fila de trabalho, o fluxo é direcionado à etapa de processamento; caso contrário, prossegue para o estado de finalização.

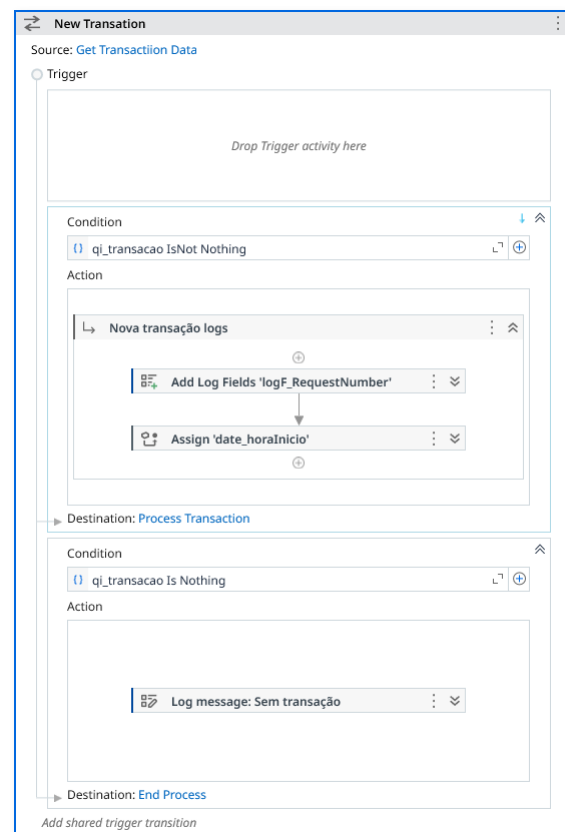


Figura 4: Transição de Estados

Fonte: UiPath

### A. Ambiente do Orchestrator

O ambiente do *Orchestrator* tem duas funções importantes nesse projeto, guardar dados sigilosos como senhas de logins e

gerenciar a fila de trabalho do robô. Para a executar a primeira função são criados *Assets* na plataforma, mostrado na figura 5. Cada um contém um usuário e senha que será utilizado conforme a necessidade dos processos.

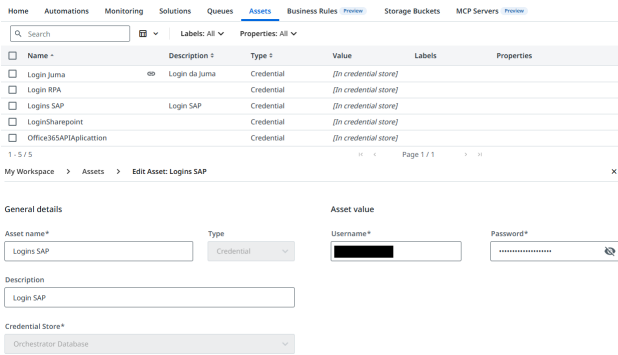


Figura 5: Assets Orchestrator  
Fonte: UiPath

Sempre que o robô for acessar um sistema, que requiera login, ele utilizara o *Asset* correspondente para fazer o mesmo, os dados são guardados de forma segura e protegida na nuvem. Dessa maneira evita-se que qualquer senha esteja escrita na *hard code* do RPA e quando houver a necessidade do usuário final necessitar trocar uma senha ele poderá fazer o mesmo pela plataforma do *Orchestrator*, sem a necessidade de alteração direta no código, aumentando a segurança e facilidade da operação.

Quanto a fila de trabalho a mesma também é criada na nuvem do *Orchestrator*, conforme a figura 6. O papel dela é gerenciar a ordem dos processos do RPA, assim como possibilitar ao usuário acompanhar o estado das operações, alterar as prioridades quando necessário e verificar se ocorreu algum erro durante as mesmas.

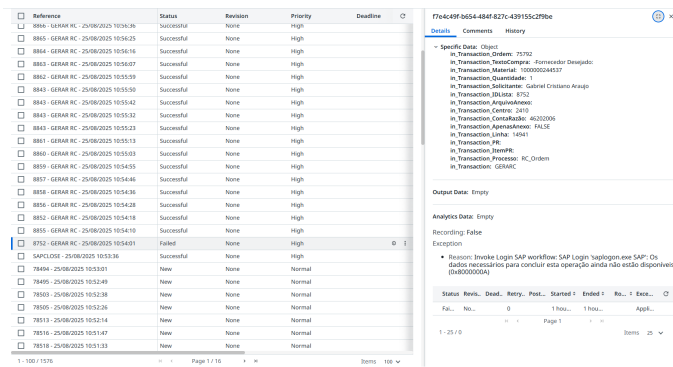


Figura 6: Fila de trabalho  
Fonte: UiPath

Cada linha representa uma operação de trabalho para ser executada pelo robô e contém os dados necessários para executar o processo em específico; os horários de criação, execução e termino da operação; status da operação(Criada, em progresso, finalizada, falha ou excluída); e em casos de

falha o *log* de erro do robô. A informação inicial de cada item é carregada através da atividade do coletor explicada a seguir e após alterada durante a execução dos processos específicos.

### B. Preparação do ambiente de trabalho

O primeiro passo da inicialização é verificar se essa é a primeira execução do dia e caso sim carregar as configurações iniciais do robô, para isso é acessado a planilha de configurações guardada junto ao executável do RPA e transformado a mesma em uma variável dicionario, nela se encontram os dados necessários para acessar a nuvem do *Orchestrator*, filas de trabalho e executáveis utilizados. Após é acessado a nuvem do *Orchestrator* e carregado os dados contidos nela e também transforma-la em uma variável dicionario para ser acessada quando necessário. A figura 7 demonstra o procedimento para carregar os dados de *Login* do servido *Sharepoint Online* da empresa: é criado a variável dicionario(A), após utilizado a função *Get Credential*(B) para coletar as senhas e salvo o valor nas variável criada anteriormente(C).

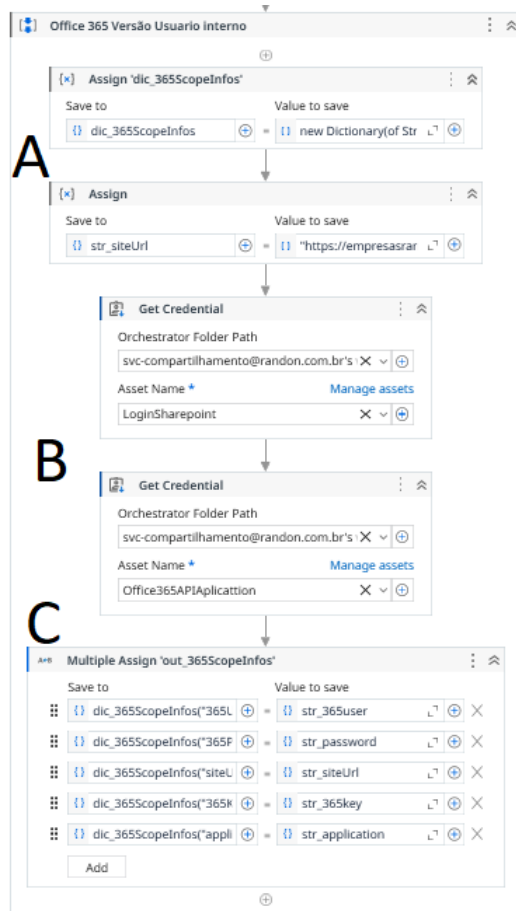


Figura 7: Carregar Assets *Office365*  
Fonte: UiPath

O segundo passo é preparar o ambiente de trabalho, para isso é utilizado um *loop* passando por cada executável contido na variável dicionario de configurações e a função *Kill Process*

que fecha o processo informado nela, conforme mostrado na figura 8.

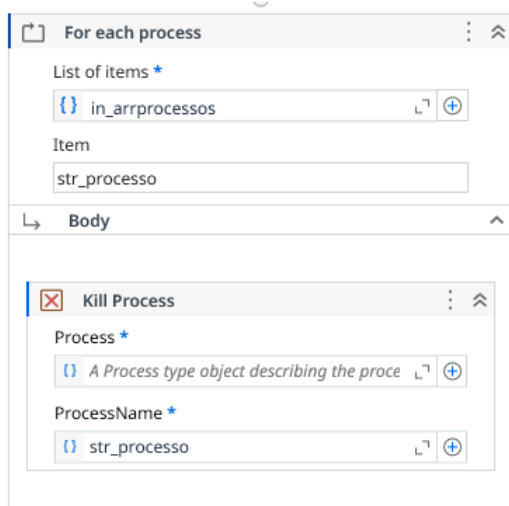


Figura 8: Fechar executáveis

Fonte: UiPath

O terceiro passo é fazer o *login* no sistema SAP utilizando a função *SAP Login*, mostrado na figura 9, e os dados coletados na nuvem do *Orchestrator*. O último passo é verificar se existem itens pendente na fila de trabalho e, caso não, ativar o Coletor.

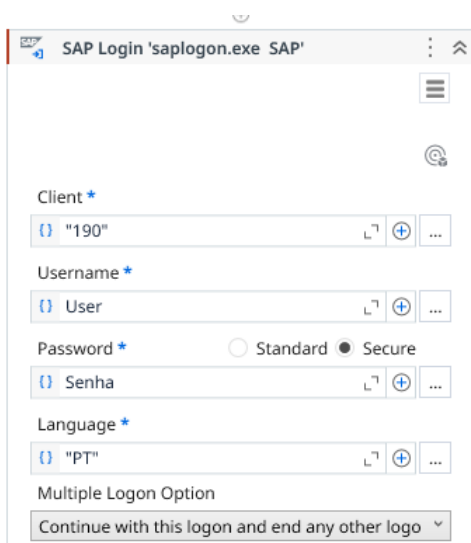


Figura 9: Função SAP Login

Fonte: UiPath

### C. Coletor

A função do coletor é inserir todas as atividades que o RPA deve executar na fila de trabalho, carregando, juntamente, os dados necessários para sua execução. Em contato com a equipe da empresa, foi decidido que o RPA realizaria a carga inicial automaticamente em horários específicos. Para

isso, foi criada uma atividade que, a cada início de ciclo de operação, verifica se não há itens pendentes na fila de trabalho, conforme a figura 10, e, utilizando um bloco de comando *Switch*, mostrado na figura 11, verifica se o horário atual corresponde aos especificados. Caso positivo, novas atividades são adicionadas à fila por meio de uma série de blocos *Add Queue Item*. Nesses blocos, são informados o nome da fila de trabalho e os dados a serem carregados no campo *Item Information*, conforme ilustrado em 12.

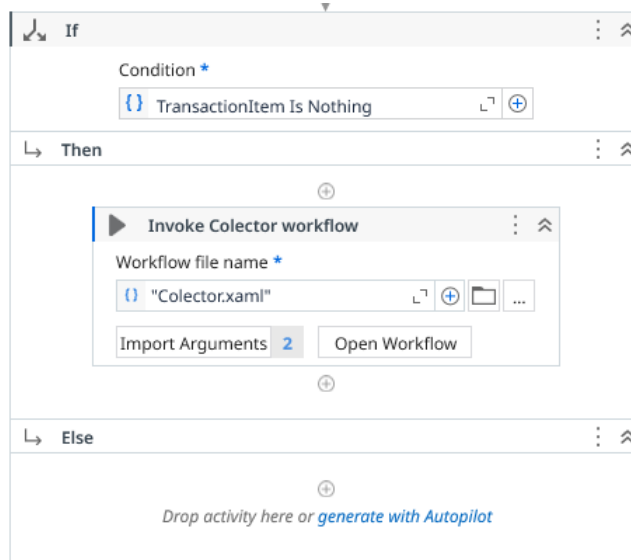


Figura 10: Acionamento condicional do Coletor

Fonte: UiPath

O método descrito acima é aplicado aos itens que não apresentam variação em sua execução, como a coleta de relatórios e a verificação de estoques. Para os demais objetivos do projeto, as informações necessárias estão contidas em planilhas online geradas pelos sistemas de chamados da empresa, como apresentado na planilha 13.

Nesses casos o robô acessa a planilha diretamente, fazendo *Login* na plataforma *Office365* por meio da atividade *Microsoft Office 365 Scope* utilizando as credenciais de autenticação salvas na figura 7 e informando-os nas propriedades da função. Dentro do escopo dessa atividade, é utilizada a função *Find files and Folders* para acessar o arquivo e, em seguida, a atividade *Read Range* para convertê-lo em uma tabela de dados interpretável pelo robô, conforme mostrado na figura 14. Em seguida são identificadas as atividades pendentes para execução, definidas pelas linhas com a coluna "STATUS" em vazio, coletados os dados relevantes, adicionados à fila de trabalho e, por fim, atualizada a coluna "STATUS" dos itens adicionados com o texto "IN QUEUE". Essa marcação evita a adição duplicada do mesmo item na fila caso o coletor seja executado novamente antes da conclusão do processo, o que geraria duplicidades indesejadas. Essa mesma marcação será substituída, durante a execução dos processos, pelo resultado obtido, que pode ser uma mensagem de erro ou o número do item gerado no sistema SAP.

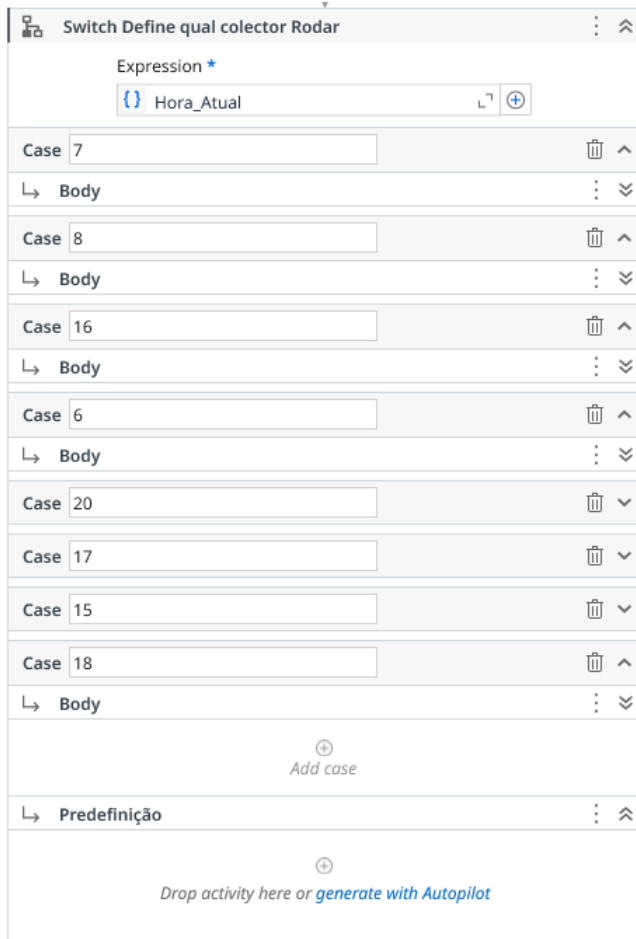


Figura 11: Acionamento *Switch Coletor*  
Fonte: UiPath

### D. Adquirir dados de transação

Nesse ponto do processo o RPA acessa a fila de trabalho na nuvem do *Orchestrator* e tenta coletar o item de maior prioridade da mesa utilizando a função *Get transaction item* mostrado na figura 15. Em caso de sucesso, os dados coletados são transformados em uma variável do tipo dicionário, contendo as informações enviadas pelo coletor, e encaminhados para a etapa de processo, onde a operação será realizada. Caso contrário, o sistema prossegue diretamente para a etapa de finalização.

### E. Processos

Essa atividade recebe as informações das etapas anteriores e decide qual módulo deve ser ativado para concluir a tarefa solicitada. Para isso, todos os itens enviados à fila de trabalho possuem uma variável denominada "in\_Transaction\_Processo" exibida na aba *Details* da figura 6. Essa variável é extraída e utilizada em um bloco de comando *Switch*, que seleciona o processo correspondente, conforme ilustrado na figura 16.

Executado a atividade solicitada, o RPA atualiza o status do item na fila de trabalho com o resultado obtido. Caso não

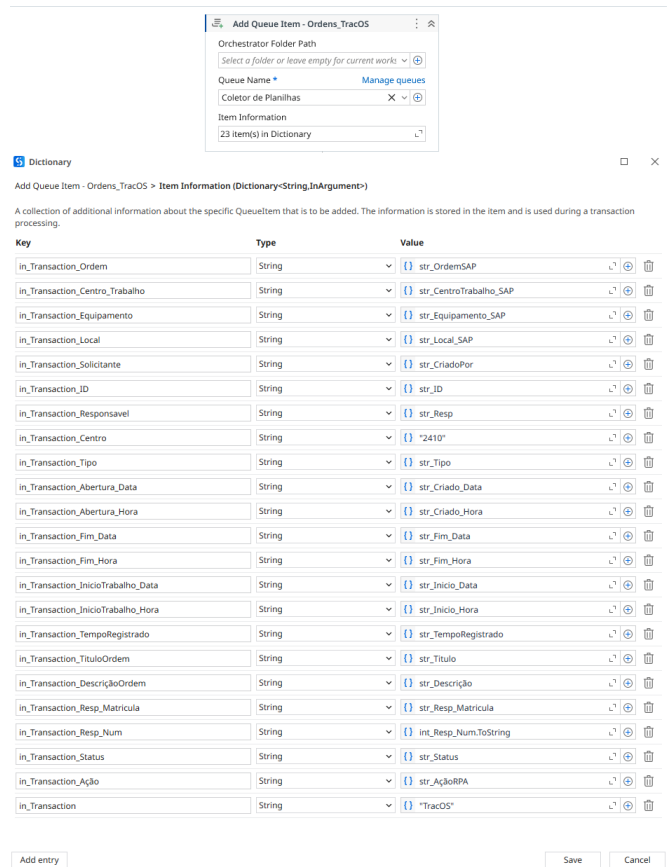


Figura 12: Dados carregados no Coletor  
Fonte: UiPath

Ordem	RC	Item	STATUS	Materiais	Quantidade	Texto da Compra	Solicitante
15100	23647554	27712349	10	RC Gerada	1000000322655	200	Compra reposição
15110	23647555	27712440	10	ERRO	ERRO NA CRIAÇÃO	10	Estoque
15111	23647556	27712440	10	RC Gerada	1000000283701	1	Compra reposição
15112	23647557	27712441	10	RC Gerada	1000000244602	200	Compra reposição
15113	23647560	27712442	10	RC Gerada	1000000105985	4	Compra reposição
15114	23647621	27712444	10	RC Gerada	1000000064958	1	Compra reposição
15115	23647622	27712445	10	RC Gerada	1000000005966	6	Compra reposição
15116	23647623	27712446	10	RC Gerada	1000000039673	1	Compra reposição
15117	23647624	27712447	10	RC Gerada	1000000109587	2	Compra reposição
15118	23647625	27712829	10	RC Gerada	1000000100022	8	Compra reposição
15119	23647626	27712830	10	RC Gerada	1000000148038	2	Compra reposição
15120	23647627	27712831	10	RC Gerada	1000000277563	1	Compra reposição
15121	23647629	27712832	10	RC Gerada	1000000237538	3	Compra reposição
15122	23647630			REVISAR	1000000282960	1	Compra reposição
15123	23647632	27712835	10	RC Gerada	1000000082121	1	Compra reposição
15124	23647633	27712837	10	RC Gerada	1000000109651	4	Compra reposição

Figura 13: Planilha de Compras Pendentes  
Fonte: UiPath

ocorra nenhum erro o fluxo retorna à etapa "Adquirir dados de transação" para coletar os dados do próximo item da fila de trabalho. Se houver erros imprevistos ao invés retorna-se para a etapa inicial. Esse procedimento é feito para retornar a área de trabalho ao estado padrão, evitando que um erro inesperado se alastre para as próximas atividades.

A tabela I apresenta cada processo implementado e os respectivos resultados esperados. A seguir, será detalhado o funcionamento de cada módulo executado nessa atividade.

1) *Geração de relatórios*: Esse processo tem como objetivo a criação de arquivos em Excel, a partir da extração de dados relevantes do sistema SAP para o setor de manutenção. O

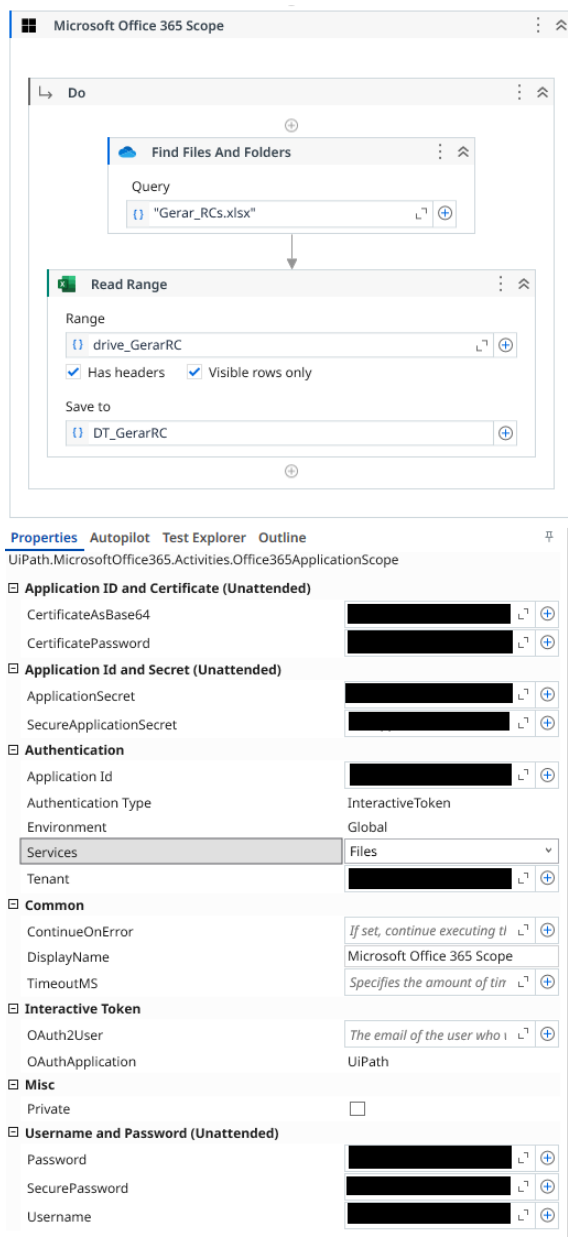


Figura 14: Acesso as planilhas online  
Fonte: UiPath

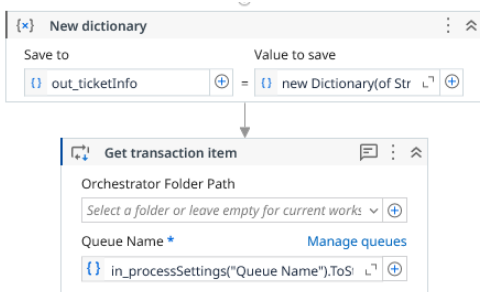


Figura 15: Adquirir dados de transação  
Fonte: UiPath

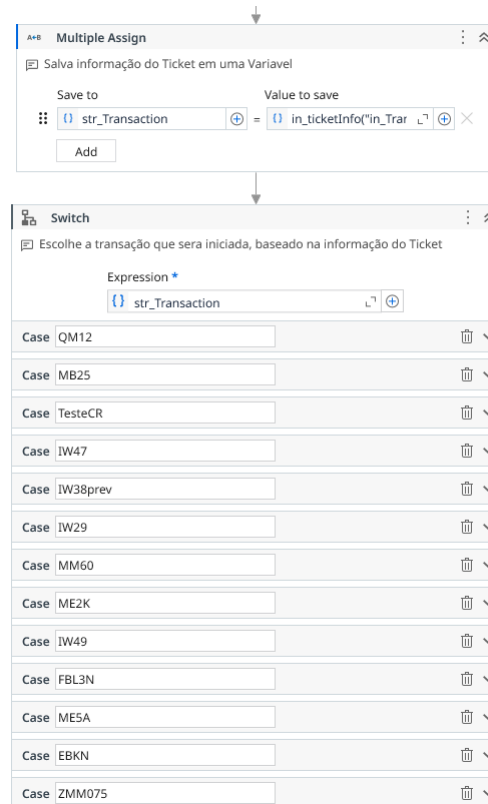


Figura 16: Seleção de processos  
Fonte: UiPath

Tabela I: Processos e Benefícios Esperados

Processo	Entrada	Saída	Benefício Esperado
Geração de relatórios	Automático	Arquivos em Excel	Economia de tempo, Atualização automática de dashboards com informações atualizadas
Requisições de compras	Planilha Online	Número da requisição criada	Economia de tempo, Padronização e controle unificado
Geração de notas fiscais	Planilha Online	Arquivo da NFe em PDF	Economia de tempo, Padronização e controle unificado
Revisão de estoques	Automático	Solicitar Compra de Itens Faltantes	Fazer ponte entre estoque no sistema Tractian e compras no sistema SAP simulando um sistema MRP
Replicar ordens de manutenção	Sistema Tractian	Número da Ordem de manutenção criada	Mantém o histórico de manutenção no sistema SAP. Fazer a ponte das ordens de manutenção entre o sistema Tractian e SAP

fluxograma apresentado na Figura 17 ilustra o passo a passo para a extração manual dos dados da transação *IW38*, a qual contém uma lista de informações referentes a cada ordem de manutenção. Ao todo, são extraídos 12 relatórios do sistema utilizando esse método, variando apenas o nome da transação e o campo *Layout*.

Para replicar essa atividade, são utilizados blocos de comandos como o ilustrado na Figura 18. O primeiro bloco corresponde à atividade *Call Transaction* (A), que chama diretamente a transação *IW38* no SAP, conforme mostrado na Figura 19. Em seguida, são executados blocos *Type Into* (B), responsáveis por inserir diretamente os valores desejados nos campos identificados. Após essa etapa, utiliza-se o bloco *Click Toolbar Button* (C) para simular a ação do usuário ao pressionar a tecla de avanço, resultando no relatório apresentado na Figura 20. Por fim, o bloco *Click Toolbar Button* é novamente utilizado para acionar o comando de extração do relatório, seguido por outros blocos *Type Into* (B), que definem o local

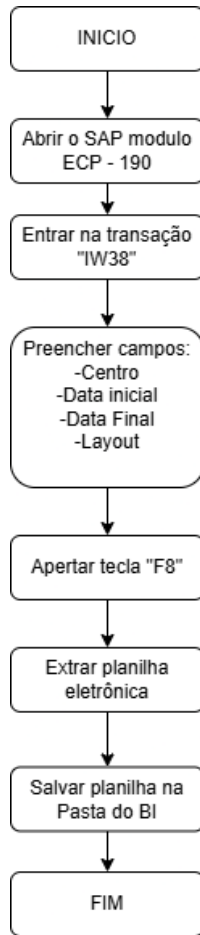


Figura 17: Fluxograma geração de relatório IW38  
Fonte: Autoria própria

onde o arquivo será salvo.

Um possível problema identificado nessa etapa é que alguns campos a serem preenchidos, bem como o diretório onde os arquivos devem ser salvos, podem sofrer alterações ao longo do tempo devido a mudanças em sua utilização. Para evitar a necessidade de modificar esses valores diretamente no código, foi criada uma planilha de referência, conforme apresentado na Figura 21. Ao invés de os valores estarem inseridos diretamente no código do robô, ele passa a extrair e utilizar os dados presentes no arquivo, facilitando para o usuário final realizar alterações sempre que necessário.

2) *Requisições de compras*: Esse módulo do processo tem por objetivo gerar requisições de compras no sistema SAP com base nas informações contidas na planilha online disponibilizada pela equipe da empresa, apresentada na figura 13. Após a conclusão do processo, o número da requisição de compra criada é informado na planilha original. O processo de coleta de informações da planilha é feito pelo coletor, como explicado anteriormente, e, a partir das informações carregadas na fila de trabalho, é iniciado esse módulo.

O primeiro passo consiste em separar os dados recebidos da fila de trabalho em variáveis do tipo *string*, conforme a

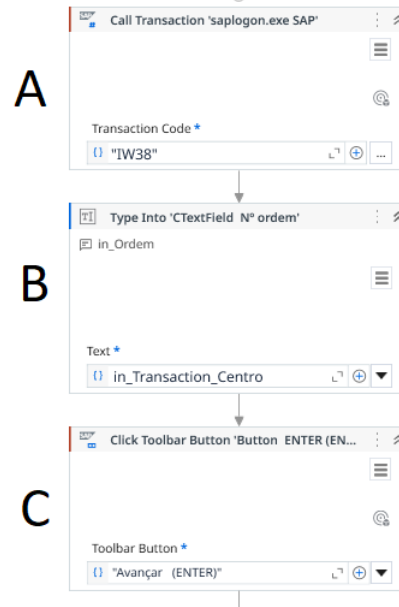


Figura 18: Sequência inicial para geração do relatório IW38  
Fonte: Uipath

Modificar ordens PM: Seleção de ordens

Receptor de apropriação de custos MAP

Status da ordem: pendente em pronto Concluído histórico Esq.seleção  Endereço

Seleção de ordens

Ordem		até		
Tipo de ordem		até		
Local de instalação		até		
Equipamento		até		
Nota		até		
Cen.Trab.principal		até		
Centro p/cen.trab.		até		
Período	01.01.2024	até	31.12.2025	
Parcs.				

Seleção de ordem de serviço de manutenção

ID de documento de serviço		até		
ID de documento de serviço		até		
Produto de serviços		até		
Emissor da ordem		até		
Organização de vendas		até		
Canal de distribuição		até		
Divisão		até		

Dados gerais/dados administrativos

Criado por		até		
Data-base do início		até		
Data-base do fim		até		

Dados locais./classif.contábil

Centro de manutenção	2410	até		
Tp.atvd.manut.		até		
Centro custo		até		
Empresa		até		
Grp.planej.manutenção - ordem		até		

Outros

Status de paging		até		
Layout	/ORDENS			
Campo ref.para monitor				

Figura 19: Transação IW38  
Fonte: SAP

Figura 20: Relatório IW38  
Fonte: SAP

A	C	D	E	F	G	H	
1	TransacaoSAP	Centro	Layout	Arquivo	Arquivo Copia	LocalDeUpload	PastaDestino
2	TesteCR	2410 /39633		TesteCR.XLSX	TesteCR.XLSX	\\randon.com.br\empr\C\SharepointPCV	
3	MB25	2410 /39633		MB25.XLSX	MB25.XLSX	\\randon.com.br\empr\C\SharepointPCV	
4	IP24	2410 /39633		IP24.XLSX	IP24.XLSX	\\randon.com.br\empr\C\SharepointPCV	
5	IP17	2410 /39633		IP17.XLSX	IP17.XLSX	\\randon.com.br\empr\C\SharepointPCV	
6	QM12	2410 /39633		QM12.XLSX	QM12.XLSX	\\randon.com.br\empr\C\SharepointPCV	
7	IW38prev	2410 /39633PBI		iw38prev.csv	IW38.csv	\\randon.com.br\empr\C\SharepointPCV	
8	IW47	2410 /39633PBI		IW47.XLSX	IW47.XLSX	\\randon.com.br\empr\C\SharepointPCV	
9	IW29	2410 /PBI_CASTER		IW29.CSV	IW28.CSV	\\randon.com.br\empr\C\SharepointPCV	
10	MM60	2410		MM60.CSV	MM60.CSV	\\randon.com.br\empr\C\SharepointPCV	
11	MESA	2410 /39633		MESA.XLSX	MESA.XLSX	\\randon.com.br\empr\C\SharepointPCV	
12	IW49	2410 /39633PBI		IW49.CSV	IW49.CSV	\\randon.com.br\empr\C\SharepointPCV	
13	FBL3N	2400 /PRESET_USIN		fbl3n.XLSX	fbl3n.XLSX	\\randon.com.br\empr\C\SharepointPCV	
14	ME2K	2410 /39633		me2k.XLSX	me2k.XLSX	\\randon.com.br\empr\C\SharepointPCV	
15	ZMM075	2410 /39633		zmm075.XLSX	zmm075.XLSX	\\randon.com.br\empr\C\SharepointPCV	
16	ME2L	2410 /39633		ME2L.XLSX	ME2L.XLSX	\\randon.com.br\empr\C\SharepointPCV	
17	EBKN	2410		EBKN.XLSX	EBKN.XLSX	\\randon.com.br\empr\C\SharepointPCV	
18	MB51	2410 /PRESET		MB51.XLSX	MB51.XLSX	\\randon.com.br\empr\C\SharepointPCV	
19	MB52	2410 /IRT_2		mb52.xlsx	mb52.xlsx	\\randon.com.br\empr\C\SharepointPCV	

Figura 21: Planilha de referencia relatórios  
Fonte: Autoria própria

figura 22, utilizando a função *Multiple Assign*. Em seguida, verifica-se se a compra solicitada possui um arquivo em anexo, por exemplo, um orçamento ou desenho técnico e, em caso afirmativo, salva-se o arquivo no computador onde o RPA está operando, utilizando a sequência de funções mostradas na figura 23. O bloco *Find Files and folder*(A) procura o arquivo com o nome informado no diretório da nuvem e devolve o resultado da *Query* em uma variável do tipo *drive*; a função *Get Environment Folder*(B) identifica o local da pasta de documentos padrão do *Windows*; por fim, a função *Download File*(C) recebe as duas informações geradas anteriormente e baixa o arquivo na pasta indicada, para uso nas atividades posteriores.

O próximo passo é criar a requisição de compra. Para isso, é utilizado uma sequência de blocos de comandos para abrir a transação IW32 no SAP, preencher o número de Ordem e clicar no botão componentes, abrindo a tela da figura 24. Nesse ponto é utilizado uma sequência de atividades *Type Into* para preencher as informações recebidas na lista de componentes, pressionar o botão "enter" e preencher os dados na segunda janela da figura 25. Após isso, é usado a função *Click Toolbar Button* para pressionar o botão salvar.

Feito esses passo, a próxima tarefa é reabrir a ordem, copiar o número da requisição gerada, inserir o texto descritivo da compra e, caso necessário, adicionar o arquivo em anexo. Para isso, o robô utiliza novamente a função *Click Toolbar Button* para reabrir a ordem clicando no botão componentes

Figura 22: Separando variáveis da fila de trabalho  
Fonte: UiPath

Figura 23: Baixar Arquivo em Anexo  
Fonte: UiPath

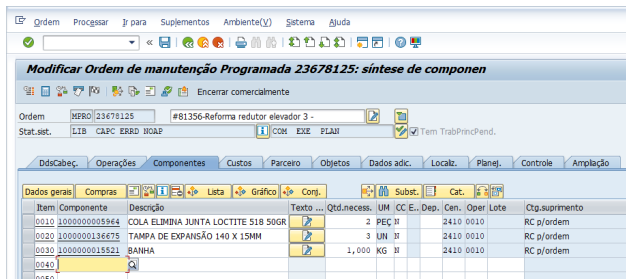


Figura 24: Transação IW32: Componentes parte 1  
Fonte: SAP

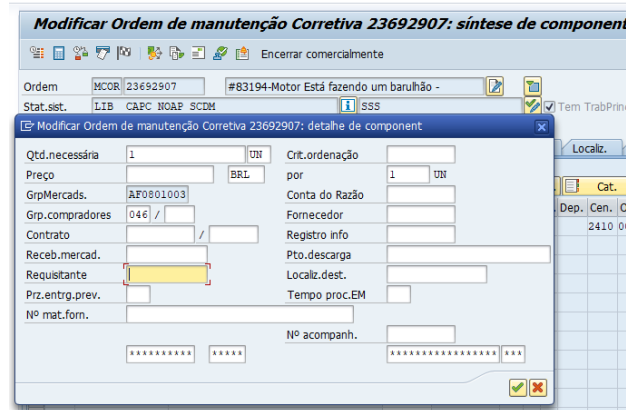


Figura 25: Transação IW32: Componentes parte 2  
Fonte: SAP

e, em seguida, no botão "COMPRAS", mostrado na figura 24, utilizando a função *Click*, abrindo a janela da figura 26. Em seguida, seleciona a opção de pular para ultima requisição, utiliza a função *Type into* no campo de descrição, escrevendo o texto descritivo, copia o número da requisição utilizando a função *get text* e, novamente, a função *Click* para dar dois cliques nesse número, abrindo uma nova janela onde é adicionado o arquivo em anexo. Para isso, utiliza-se a função *Click Toolbar Button* para selecionar a opção anexo, duas vezes a função *Type into* para escrever o local e o nome do arquivo e, após, salvar as alterações.

Caso todo o processo acima ocorra sem erros, o robô acessa a planilha de compras da figura 13 e informa o número da requisição utilizando a função *Write Cell* com o número gravado anteriormente, preenchendo o campo de status com o texto "RC Gerada". Caso ocorra algum erro durante a operação, ao invés disso, ele informa o erro ocorrido nesse campo. Para realizar a detecção de erros, é utilizado a função *Try Catch* envolvendo todo o processo, juntamente com uma variável de status que altera seu valor conforme o passo atual do processo. A função *Try Catch* opera monitorando o processo contido na mesma e, quando ocorre um erro, ao invés de interromper o RPA com uma mensagem de erro necessitando de intervenção humana, ela pula a execução do processo e ativa uma rotina alternativa para contenção do erro. Nesse caso, a contenção consiste em preencher o campo de

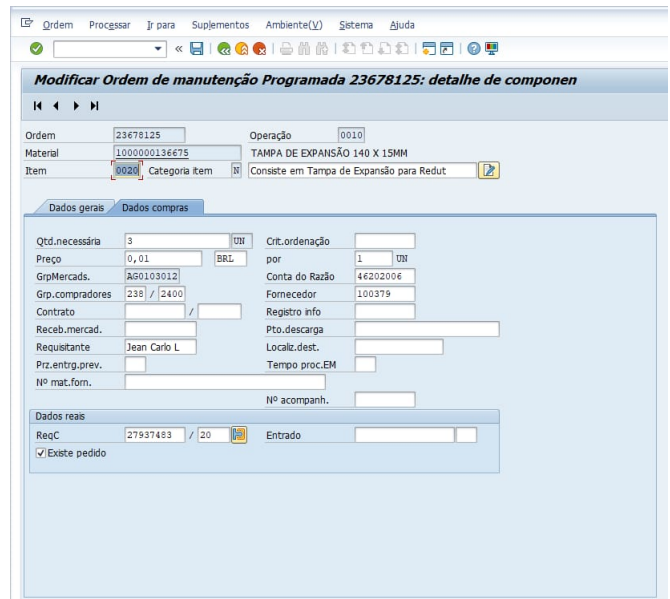


Figura 26: Transação IW32: Componentes 3  
Fonte: SAP

status da planilha com o erro ocorrido, pular esse item da fila de trabalho e retornar o processo para o ponto de inicialização, onde será preparado novamente a área de trabalho.

3) *Geração de notas fiscais*: Esse módulo tem por objetivo gerar notas fiscais de remessas para conserto em fornecedores externos a partir de informações contidas em uma planilha online, conforme mostrado na figura 27, e salvar o arquivo gerado em uma pasta de rede no formato PDF. O processo de coleta de informações é realizado pelo coletor, como explicado anteriormente, e, a partir das informações carregadas na fila de trabalho, é iniciado esse módulo.

№	Equipamento	Objeto	Material	Quantidade	Valor	Unidade
№ 00001811	PR0007	12143076	18 BOMBA	18	100	1
№ 00001812	PR0007	12143076	20 BOMBA BOMBA	20	100	1
№ 00001813	PR0007	12143076	40 BOMBA	40	100	1
№ 00001814	PR0007	12143076	10 BOMBA BOMBA	10	100	1
№ 00001815	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001816	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001817	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001818	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001819	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001820	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001821	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001822	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001823	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001824	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001825	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001826	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001827	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001828	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001829	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001830	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001831	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001832	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001833	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001834	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001835	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001836	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001837	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001838	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001839	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001840	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001841	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001842	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001843	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001844	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001845	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001846	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001847	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001848	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001849	PR0007	12143076	2 BOMBA BOMBA	2	100	1
№ 00001850	PR0007	12143076	2 BOMBA BOMBA	2	100	1

Figura 27: Planilha Notas Fiscais  
Fonte: Autoria própria

O passo inicial é criar uma requisição de compra, processo idêntico ao módulo anterior, e salvar o número gerado em uma variável. Em seguida, é aberto a transação "VA01" no SAP da figura 28, preenchidas as informações contida na planilha utilizando uma sequência de funções *Type into* e *Click* para navegar pelos menus, clicar no botão fornecer e, após, salvar. Durante esse processo, também é informado o número da requisição de compra gerada nas informações de rodapé da nota fiscal, embora não seja uma informação obrigatória para

gera-la, é considerado boa prática pela equipe da empresa e, por isso, foi incluída no processo do robô. Finalizando esse processo, é ativado um temporizador de 7 minutos, tempo necessário para o sistema SAP gere o documento da nota fiscal eletrônica junto ao sistema da receita federal do Brasil [20].

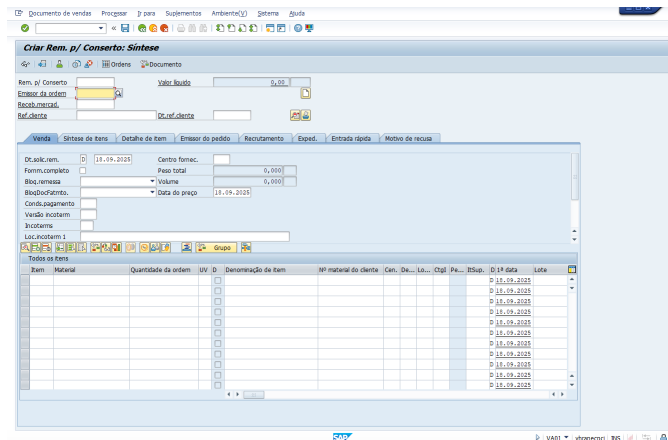


Figura 28: Transação VA01  
Fonte: SAP

Finalizado o temporizador, é utilizada a função *Call transaction* para chamar a transação "J1BNFE", com o intuito de pesquisar o número da última nota fiscal criada pelo usuário, salvar o número da mesma em uma variável, chamar a transação "ZSD068", informar o número, clicar em *Download PDF*, selecionar a pasta informada na planilha online e salvar. Após, é aberta a requisição de compra criada no início e informado na mesma o número da nota fiscal gerada.

4) *Revisão de estoques*: O objetivo desse processo é criar um relatório sobre a situação dos materiais armazenados em estoques controlados no sistema *Tractian*, abaixo dos valores de controle definidos pela equipe da manutenção, verificar quais materiais já estão em compras no sistema SAP, cruzar essas informações e solicitar a compra dos itens em falta, semelhante a um sistema *MRP* (*Material requirements planning*). Nesse caso, segundo a equipe da empresa, o processo manual nunca foi executado de maneira efetiva, razão pela qual optou-se por não replicá-lo, mas sim desenvolver um método específico para execução do RPA.

O primeiro passo é verificar a situação atual do estoque no sistema *Tractian*. Para isso, o RPA utiliza a atividade *Use Application/Browser*, da figura 29. Ela desempenha dois papéis importantes: abrir a página da internet onde se encontram os dados do inventário e fazer a delimitação inicial para os seletores utilizados dentro da mesma, indicando que suas interações devem se limitar à página ou aplicativo especificado na atividade [7].

Aberto a página da figura 30, é utilizada uma sequência de atividades *Click* para clicar nos pontos indicados na figura na sequência: Selecionar Filtros salvos(A), Selecionar filtro "RPA ESTOQUE"(B), Baixar dados(C).

Para configurar os seletores das atividades *Click*, é selecionada a opção *Indicate target on screen*, mostrada na figura 29,

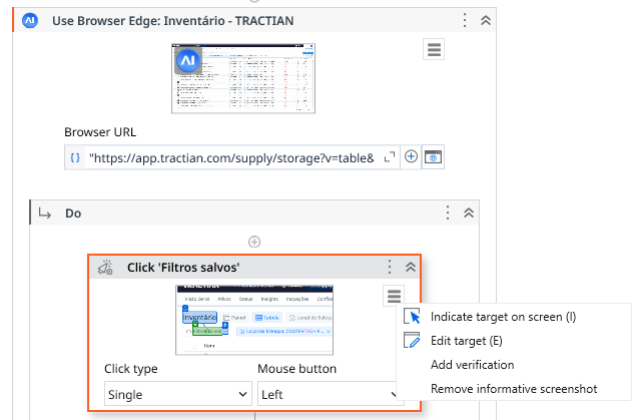


Figura 29: Use Application/Browser  
Fonte: Autoria própria

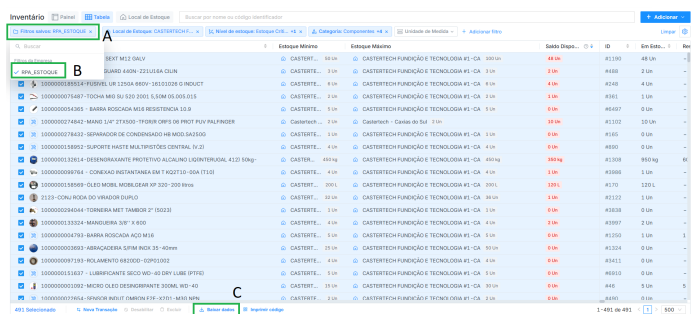


Figura 30: Inventário Tractian  
Fonte: Autoria própria

que abre a ferramenta do seletor, ilustrada na figura 31. Nela, é indicado o alvo a ser clicado, destacado em verde, e duas âncoras destacadas em azul, que servem para orientar o robô a selecionar o alvo apenas se ele estiver próximos das duas âncoras. Neste caso, estão sendo utilizados três métodos de seleção diferentes, explicados nas seções anteriores: o seletor HTML, visão de computador e por imagem. Isso é feito pois, caso o primeiro seletor falhe na identificação do alvo durante a execução, o RPA utilizará o segundo e, em seguida, o terceiro método antes de gerar um erro e cancelar a operação.

Completando essa sequência de operações é baixada uma planilha eletrônica com os detalhes dos materiais em estoque. O próximo passo é verificar a situação de cada material no sistema SAP. Para isso, a planilha é transformada em uma *Data Table* na memória do robô e utilizada a atividade *For Each Row in Data Table* da figura 32, semelhante a um *loop for* de outras linguagens de programação. Essa atividade executa as operações contidas nela uma vez para cada linha da tabela utilizada, criando uma variável dinâmica chamada *Row*, com as informações da linha atual da tabela extraída.

Dentro dessa atividade, é feita a verificação no SAP de cada linha da tabela extraída. Para isso, é extraído o número do material contido na variável "Row" e inserido o valor na transação "MD04" do SAP, utilizando os blocos de comando da figura 33: *Call Transaction*(A), *Type Into*(B) e *Click Toolbar*

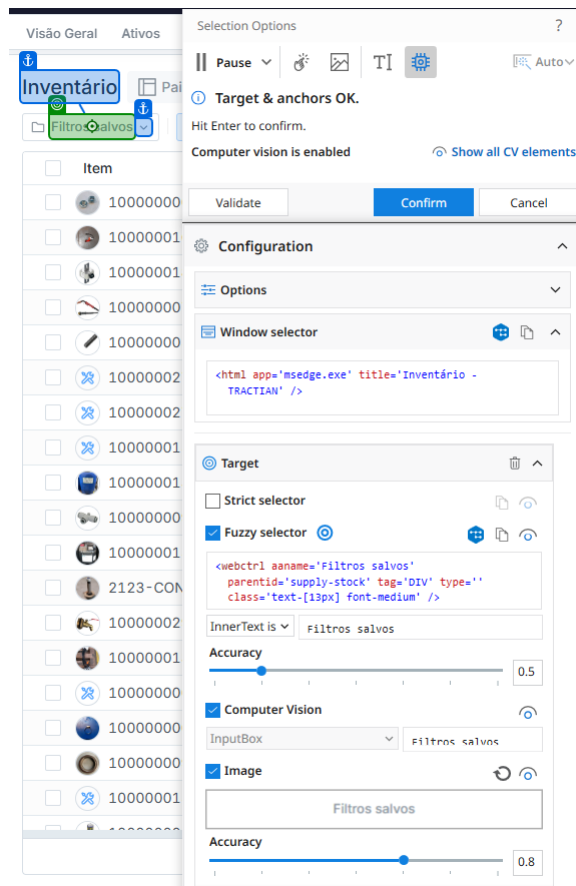


Figura 31: Seletor  
Fonte: Autoria própria

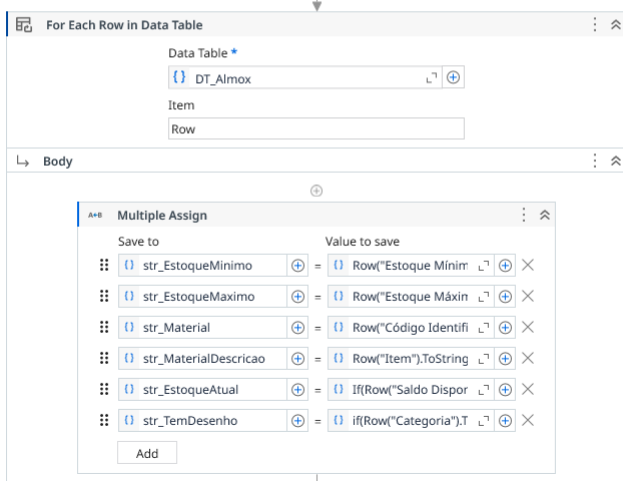
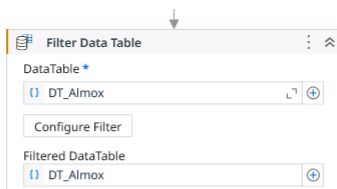


Figura 32: Função *For Each Row*  
Fonte: UiPath

*Button(C)*, configurado para selecionar a opção avançar. Essa transação exibe, de forma atualizada, o que está previsto de entrada e saída do material especificado em formato de lista dinâmica [21]. Para obter essa informação, é utilizada a função *Table Extraction* da figura 35, que extrai o conteúdo da lista em formato de tabela de dados que pode ser lida pelo RPA. Em seguida, utiliza-se uma sequência de funções *Assign*, para atribuir valores a variáveis, combinando as informações da variável *Row* com a tabela de dados extraída, gerando informações relevantes para o processo, tais como:

- Quantidade de peças em estoque;
- Quantidade de peças em compras ou pendentes para chegada;
- Quantidade de peças em falta;
- Necessidade de lançar uma nova compra de material.

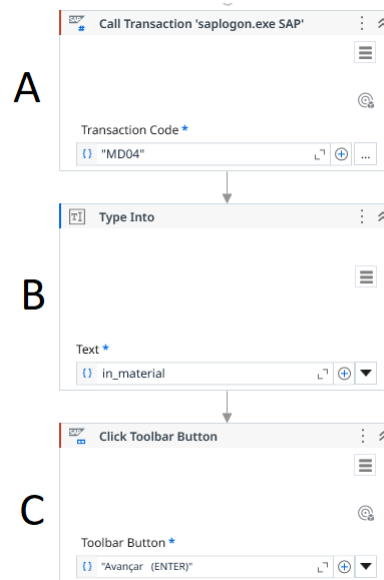


Figura 33: Acessar transação MD04  
Fonte: UiPath

Terminado o *loop for*, as informações são compiladas em uma tabela de dados, salva como um arquivo nos diretórios de rede da empresa, conforme a figura 36, e os itens definidos como necessários para compra são adicionados na planilha de compras pendentes mostrada na figura 13.

5) *Replicar ordens de manutenção*: O objetivo desse processo é replicar as informações de ordens de manutenção do sistema *Tractian* para o sistema SAP, servindo como uma ponte conectando os dois sistemas em direção única. Para evitar a repetição de informações copiadas, é utilizado a planilha online da figura 37, que contém todas as ordens já duplicadas pelo RPA, se ocorreram erros durante o processo e em qual estágio elas se encontram, sendo esses "Aberta" ou "Encerrada".

Neste caso, o coletor funciona de maneira diferente dos demais. As ordens de manutenção são baixadas de forma semelhante à demonstrada na figura 30, alterando-se apenas a página para a de Ordens. Em seguida, é realizada a

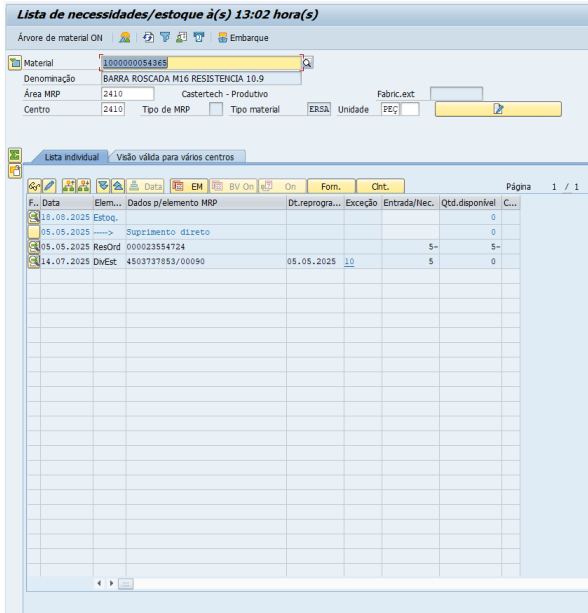


Figura 34: transação MD04  
Fonte: SAP



Figura 35: Extract Data Table  
Fonte: Uiopath

Material	Descrição	Data Da		Qtd. Em	UoM	Documento de Origem	Estoque			Qtd. Para
		compra	Existente				Atual	Estoque	Estoque	
1000000183308	1000000183308 - HASTE DO CILINDRO DA TAMPA DA PROGETA			0	PEÇ		0	1	1	1
100000071158	100000071158 - FUSÍVEL, PLACA FANUC GP50 SA A8000010245	01.08.2025		1	PEÇ	4503823240/00010	1	2	2	0
1000000102726	1000000102726 - CORREIA 3Vx 850 DENTADA	16.07.2025		3	PEÇ	4503813096/00010	0	3	3	0
1000000130969	1000000130969 - B0312M-PS 1Vx 14-C SENSOR OPTICO	10.06.2025		2	PEÇ	4503791020/00020	0	2	2	0
1000000128024	1000000128024 - FRENDOH INDUTIVO TL-W1808M1 DMIRON	23.06.2025		2	PEÇ	4503797810/00010	0	2	2	0
1000000320304	1000000320304 - CORPO FIT ANDRIZ C39P-2 SMC	23.06.2025		2	PEÇ	4503778815/00010	0	2	2	0
1000000195312	1000000195312 - CORPO PARA FILTRO SMC CASP	16.06.2025		2	PEÇ	4503778815/00010	0	2	2	0
1000000320333	1000000320333 - CORPO FIT ANDRIZ C39F3A SMC	23.07.2025		2	PEÇ	4503778815/00010	0	2	2	0
1000000022754	1000000022754 - ESPRINHO RELETR P SENSOR OPT	14.08.2025		4	UN	4503780000/00010	0	4	4	0
1000000130421	1000000130421 - K02L12-03AS CONEXAO INST EM L	04.06.2025		10	PEÇ	4503765862/00010	0	10	10	0
1000000020811	1000000020811 - BUCHA REDUÇÃO 1 X 1/2 GALVANIZADA	04.04.2025		6	PEÇ	4503688336/00010	0	4	10	0
1000000244732	1000000244732 - BUCHA REDUÇÃO 1 X 1/2 GALVANIZADA	15.08.2025		5	PEÇ	4503125223/00010	0	5	5	0
1000000151637	1000000151637 - LUBRIFICANTE SECO WD-40 DRY LUBE (PTFE)	13.06.2025		5	UN	4503760273/00010	0	5	5	0
1000000012580	1000000012580 - CABO FLEXÍVEL PP 3 X 1,5MM	09.06.2025		40	M	4503156409/00010	1	20	40	0
1000000100333	1000000100333 - FITA DURA SERVO FINA FORTE 3M	29.07.2025		8	UN	4503682796/00010	0	4	4	0
1000000322655	1000000322655 - ÓLEO HIDRA EQUIVIS ZS 32 TOTAL ENERGIES			0	L		0	200	200	200
1000000134719	1000000134719 - ÓLEO AZULLA ZS 10	14.08.2025 /		100	L	2803103083/4503835379 **Em Trânsi	120	200	200	0
1000000112114	1000000112114 - AMILHA AÇÚCAR 23 L20	18.08.2025		3	PEÇ	4503804410/00010	7	10	10	0
1000000045197	1000000045197 - AMILHA PARA TUBO 10mm	16.04.2025		20	PEÇ	4503700407/00010	0	10	10	0
1000000127938	1000000127938 - UNHÃO 10 MM	10.07.2024		10	PEÇ	4503676702/00010	0	4	4	0
1000000010048	1000000010048 - GRAXA KILBER GOFIX NEW 15 SPINDLE	15.08.2025		1	KG	4503112176/00010	0	1	1	0
1000000312434	1000000312434 - CONTADOR 3P B0305M0 3PT20251884 SIEMENS	10.07.2025 /		3	PEÇ	4503826618/00010 / 4503844200/000	0	3	3	0
1000000018826	1000000018826 - FILTRO MANTA 3M F130	25.10.2024 /		6	PEÇ	4503836711/00010 / 4503841783/000	0	1	1	0
1000000174370	1000000174370 - ESPRINHO 1/2" X PRIMO ROSCA 1/4"	0		0	UN		0	10	10	10
1000000286141	1000000286141 - PINO ENGATE RÁPIDO ROS 3/8 IMAP	29.05.2025		10	UN	4503751189/00010	0	10	10	0
1000000136357	1000000136357 - AMEL ELÁSTICO E 28	28.04.2025		10	PEÇ	0029442024/00010 *	0	10	10	0
1000000019616	1000000019616 - ÓLEO HIDRÁULICO OM COSMOLLUBRIC BAF OLIAKER	13.08.2025 /		4000	L	3803102544/4503849457 **Em Trânsi	0	2000	4000	0
1000000283701	1000000283701 - BOMBA PNEUM OIAFR PRO 1/2 666051-344	0		0	UN		0	1	1	1
1000000006034	1000000006034 - CONEXÃO MPPE GALV 1/2"	11.08.2025		10	PEÇ	4503826285/00010	0	10	10	0
1000000000601	1000000000601 - BOTELO SCHNEIDER B848421	15.08.2025 /		4	PEÇ	4503171270/00010 / 4503818304/000	0	4	4	0
100000044602	100000044602 - ÓLEO DE BARRAMENTO DROSENA MS 68	13.08.2025		400	L	3803102015/4503844229 **Em Trânsi	0	600	600	200
100000028590	100000028590 - CABO FLEX PP 3X2,5 MM 70C 750V	24.07.2025		40	M	4503761551/00010	0	20	40	0

Figura 36: Relatório Estoque  
Fonte: Autoria própria

ID	Tracos	Ordem	Tipo	Status	Ação RPA
81403	23678604	MCOR	Ordem Encerrada	Abre e Encerra	
81402	23678608	MCOR	Ordem Encerrada	Abre e Encerra	
81400	23678610	MCOR	Ordem Encerrada	Abre e Encerra	
81399	23678611	MCOR	Ordem Encerrada	Abre e Encerra	
81390	23678612	MCOR	Ordem Encerrada	Abre e Encerra	
81388	23678613	MCOR	Ordem Encerrada	Abre e Encerra	
81066	23667141	MPRV	Ordem Encerrada	Apenas Encerra	
80876	23660588	MPRV	Ordem Encerrada	Apenas Encerra	
80394	23627490	MPRV	Ordem Encerrada	Apenas Encerra	
81444	23678625	MCOR	Ordem Encerrada	Encerra Existente	
81419	23678626	MCOR	Ordem Encerrada	Encerra Existente	
81417	23678627	MPRO	Ordem Aberta	Apenas Abre	
81416	23678628	MPRO	Ordem Aberta	Apenas Abre	
81415	23678629	MPRO	Ordem Aberta	Apenas Abre	
81414	23678630	MPRO	Ordem Aberta	Apenas Abre	
81412	23678631	MPRO	Ordem Aberta	Apenas Abre	
81410	23678718	MCOR	Ordem Aberta	Apenas Abre	
81408	23678633	MCOR	ERRO ENCERRAMENTO	Encerra Existente	
81404	23678717	MPRO	Ordem Encerrada	Encerra Existente	
81401	23678720	MCOR	Ordem Encerrada	Encerra Existente	
81398	23678741	MPRO	Ordem Aberta	Apenas Abre	
81389	23678742	MPRO	Ordem Aberta	Apenas Abre	
81471	23678743	MCOR	Ordem Encerrada	Abre e Encerra	
81470	23678744	MCOR	Ordem Encerrada	Abre e Encerra	
81455	23678745	MCOR	Ordem Encerrada	Abre e Encerra	
81454	23678746	MCOR	ERRO ENCERRAMENTO	Encerra Existente	
81453	23678747	MCOR	Ordem Encerrada	Abre e Encerra	

Figura 37: Planilha de Referência: Ordens de Manutenção  
Fonte: Autoria própria

comparação da tabela baixada com a tabela de referência e adicionados à fila de trabalho os itens que não constam na mesma ou que estejam no estágio "Aberto" na planilha de referência e "Encerrado" na plataforma Tractian, juntamente com as informações necessárias para a abertura das ordens no SAP.

A execução desse módulo é dividido em duas partes: abertura e encerramento, podendo cada uma ser executada separadamente ou em sequência, dependendo do que for definido pelo coletor. A atividade de abertura inicia recebendo os dados da fila de trabalho e, em seguida, abrindo a transação IW21 do SAP, ilustrado na figura 38, utilizando a função Call Transaction e uma sequência de funções Type Into para preencher as informações. Após, é clicado no botão criar ordem e salvar. O número de ordem gerado aparece no campo inferior, onde é utilizado a função Get Text para capturar o número criado, que é enviado para ser salvo na planilha de referência. Caso ocorrer um erro nesse passo, o robô envia uma mensagem de erro no lugar e pula a tarefa de encerramento.

A atividade de Encerramento pode ser ativada de maneira independente utilizando o número da ordem gerado a partir da planilha de referência ou recebendo o valor diretamente da atividade anterior. O primeiro objetivo é capturar as informações das ordens que não constam na planilha baixada durante o coletor. Para isso, é aberto novamente a plataforma Tractian, digitado o ID na barra de pesquisa utilizando a função Type Into, aberto a ordem e utilizado dois comandos Keyboard Shortcuts para pressionar as teclas "Ctrl+A" (selecionar toda a pagina) e "Ctrl+C" (copiar o texto selecionado), como mostrado na figura 39. Em seguida, as partes relevantes são separadas e salvas em uma variável.

Finalizado a captura de dados é aberto a transação IW41 no SAP, como mostrada na figura 40, informado o número da ordem, preenchido os dados com uma sequência de atividades Type Into, incluindo o material capturado anteriormente,

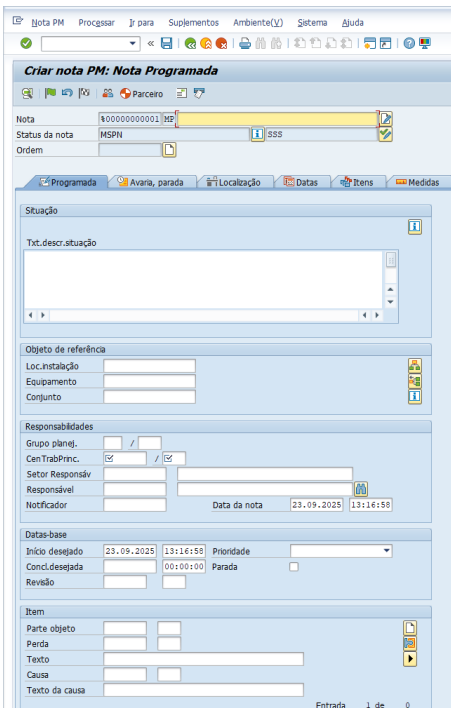


Figura 38: Transação IW21  
Fonte: SAP

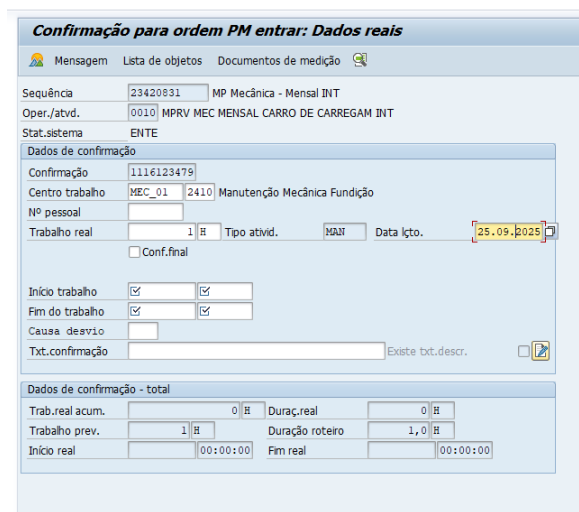


Figura 40: Transação IW41  
Fonte: SAP

de setembro e 9 de outubro, com algumas interrupções devido a quedas de energia ou correções no programa. Os arquivos do projeto foram executados a partir das pastas de rede da empresa, possibilitando realizar alterações no código utilizando o notebook da figura 41, diminuindo a necessidade de pausar a execução do mesmo para realizar correções. Também foi dado acesso à fila do *Orchestrator* para os analistas da empresa alterarem as prioridades de execução conforme necessário ou solicitar a repetição de atividades após uma falha.

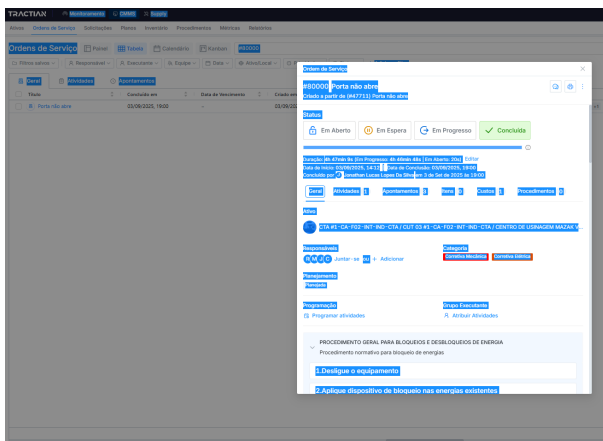


Figura 39: Inventário Tractian  
Fonte: <https://app.tractian.com/>

clicado em salvar e enviados o resultado para a tabela de referência online.

### F. Finalização

Nesse ponto, o RPA repete o *loop* da figura 8, fechando todos os processos em execução, entra em hibernação por uma hora e retorna à inicialização, encerrando o ciclo de operação.

## VII. IMPLEMENTAÇÃO E RESULTADOS

O sistema foi instalado em um computador *DELL OptiPlex 3000* dedicado exclusivamente à sua operação mostrado na figura 41, funcionando de forma contínua por 30 dias, entre 10

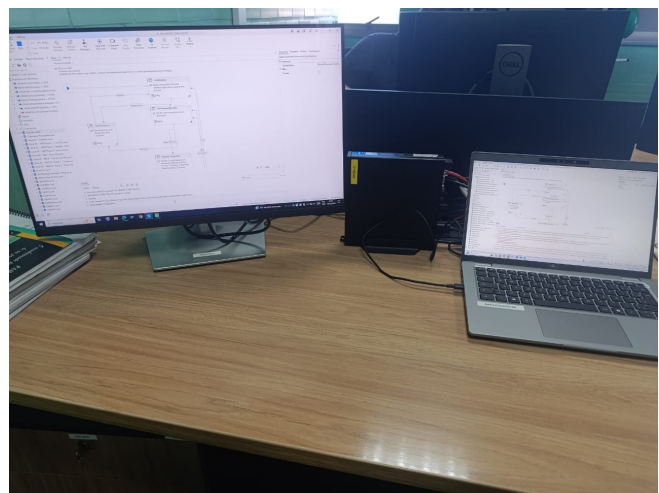


Figura 41: Instalação  
Fonte: Autoria própria

Ao longo do período de testes, o sistema executou um total de 5.683 operações distintas, incluindo operações de testes. Os resultados de execução foram extraídos em formato de tabela como mostrado na figura 42 a partir do banco de dados da nuvem e utilizados para realizar a análise estatística dos resultados. As informações coletadas incluem o status final, tempo de execução, dados utilizados entre outros.

	A	B	C	K	
1	Status	Revisid	Reference	Transaction Execution Time	Specific Data
2	Successful	None	1010 - GERAR RC - 02/10/2025 08:54:12		36 ("DynamicProperties"
3	Successful	None	1010 - GERAR RC - 08/10/2025 16:42:42		52 ("DynamicProperties"
4	Successful	None	1010 - GERAR RC - 15/09/2025 10:56:48		63 ("DynamicProperties"
5	Successful	None	1010 - GERAR RC - 17/09/2025 18:47:48		27 ("DynamicProperties"
6	Successful	None	1010 - GERAR RC - 24/09/2025 16:26:29		31 ("DynamicProperties"
7	Successful	None	47140 - GERAR RC - 06/10/2025 14:31:08		52 ("DynamicProperties"
8	Successful	None	7164 - GERAR RC - 06/10/2025 14:29:36		69 ("DynamicProperties"
9	Successful	None	7523 - GERAR RC - 06/10/2025 14:29:58		51 ("DynamicProperties"
10	Successful	None	7524 - GERAR RC - 06/10/2025 14:29:47		51 ("DynamicProperties"
11	Successful	None	7553 - GERAR RC - 06/10/2025 14:30:10		49 ("DynamicProperties"
12	Successful	None	7554 - GERAR RC - 06/10/2025 14:30:24		48 ("DynamicProperties"

Figura 42: Banco de dados  
Fonte: Autoria própria

Dentre todas execuções, 157 apresentaram falhas do tipo *System Exception*, cuja distribuição está apresentada na Tabela II junto com o motivo de falha mais comum de cada categoria. A quantidade de erros total é elevada pelo fato de quando a uma ocorrência todos os itens pendentes da fila de trabalho podem ser comprometidos e considerados como uma falha.

Tabela II: Principais *System Exception* registradas durante o período de testes

Ocorrências	Categoria	Principal motivo
48	Autenticação	Erro licença Office
29	Erro de seletor	Alteração de layout do sistema
67	Sistema Indisponível	Quedas de rede
13	Abandonos	Orchestrator em manutenção

Dentre essas a única categoria que necessitou de alteração direta no código para retornar a normalidade do RPA foram as falhas da categoria erro de seletor, a mesma ocorre quando o robô não identifica corretamente um ponto de interação no layout dos sistemas utilizados e não consegue executar a operação desejada. Mesmo assim o sistema trabalhou de maneira ininterrupta durante a ocorrência de falhas. Na maioria dos casos foi possível executar a correção do erro a partir de outro computador, sem a necessidade de pausar o RPA ou apenas fazendo a alteração das senhas e parâmetros pela nuvem.

O outro tipo de erro que pode ocorrer é denominado *Business Exceptions*. Esse tipo de exceção acontece quando a operação do RPA é executada corretamente, porém os dados fornecidos para sua execução apresentam inconsistências que impedem a conclusão da tarefa. Por exemplo, se for solicitada a compra de um material cujo código não esteja cadastrado no sistema SAP, a execução da tarefa resultará em uma *Business Exceptions*. Devido à forma como os dados foram registrados e constantemente modificados pela equipe da empresa durante a operação do RPA, não foi possível contabilizar com exatidão a quantidade de ocorrências desse tipo de erro durante o período de testes. No entanto, estima-se que aproximadamente 1% dos casos apresentaram essa falha.

Em paralelo foram cronometrados a execução das mesmas atividades quando realizadas por funcionários da empresa com o objetivo de calcular o tempo economizado em cada processo utilizando a formula 1. Foram cronometrados 20 execuções para cada processo e utilizado o valor de tempo médio das mesmas como referência, exceto no caso do processo "Relatório de estoque" pois a execução humana foi considerada

como não viável pela equipe da empresa. A tabela 43 demonstra as comparações dos resultados com os valores de tempo em segundos junto com o ganho obtido em cada processo.

$$Ganho\ de\ tempo\ (\%) = \frac{T_{manual} - T_{RPA}}{T_{manual}} \times 100 \quad (1)$$

Processo	Media RPA(s)	σ(s)	Amostras	Media Manual	Ganho(%)
Notas fiscais	591	30	72	1253	52.83%
Requisições de compra	54	22	455	194	72.16%
Atualizar Relatórios	4587	673	30	7300	37.16%
Relatório Estoque	4782	405	26	N/A	N/A
Criação de ordens	108	49	3873	210	48.57%

Figura 43: Ganhos de eficiência

Como observado houve uma economia de tempo significativa em todos os processos, com destaque nas requisições de compras, essa diferença se deve ao fato de ser um processo que embora rápido requer extensiva memorização e pesquisa de variáveis no sistema SAP para preencher os campos mostrados na figura 25. Como o robô já possui todos os dados memorizados e os preenche de maneira instantânea fica limitado apenas ao tempo de resposta do sistema SAP para executar a operação, reduzindo consideravelmente o tempo utilizado.

Para o cálculo do ganho total de eficiência do projeto, foi utilizada a equação 2, cujos resultados são apresentados no gráfico da Figura 44. O cálculo considerou as amostras coletadas ao longo dos 30 dias do período de testes, permitindo a extrapolação do impacto mensal da automação. O processo "Replicar ordens de manutenção" destacou-se com 109 horas de trabalho economizadas, apresentando o maior ganho entre todas as automações implementadas, resultado atribuído ao elevado volume de operações executadas em comparação aos demais processos.

$$Ganho\ total\ (Horas) = \frac{T_{manual} - T_{RPA}}{3600} \times n_{amostras} \quad (2)$$

Ganho Total por Processo (Horas)

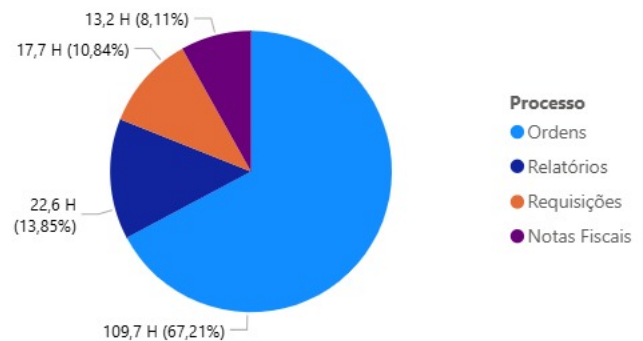


Figura 44: Ganho total de eficiência Mensal  
Fonte: Autoria própria

Durante os 30 dias de testes do RPA apresentado neste projeto, aproximadamente 330 horas de trabalho humano foram reduzidas para 163 horas de trabalho realizado pelo robô, equivalente à carga de trabalho de dois funcionários. Além do aumento da eficiência dos processos, houve outros benefícios, como a possibilidade de executar tarefas fora do horário de expediente. Por exemplo, a atualização dos relatórios durante a madrugada permitiu que os indicadores da fábrica estivessem atualizados no início do primeiro turno, possibilitando à gestão realizar análises com dados recentes. Outra vantagem foi a padronização dos processos e diminuição de erros humanos que passavam despercebido, o RPA sempre executa as tarefas da mesma maneira, diminuindo erros por falta de atenção a detalhes e esquecimento. Por fim, também foi viabilizada a criação de novos processos que não existiam anteriormente, como o relatório de compras em estoque.

A recepção da equipe da empresa ao projeto foi positiva, havendo uma redução considerável na carga de trabalho, o que permitiu realizar alterações na estrutura organizacional e dedicar mais atenção a outras atividades que estavam em atraso. Além disso, a gestão da empresa solicitou a implementação do sistema RPA nas filiais assim como a automação de outros processos administrativos, contribuindo para o resultado positivo obtido.

Um dos principais impedimentos para aumentar a eficiência do projeto foi a estabilidade da conexão com a internet da empresa. Por exemplo, no processo de replicação de ordens, são utilizados 60 segundos para coletar os dados de encerramento na plataforma *Tractian*. Entretanto, em testes paralelos com uma banda de internet superior, foi possível reduzir esse tempo em 40 segundos, o que melhoraria consideravelmente o desempenho do sistema. Também existem várias oportunidades para aumentar a eficiência do projeto executando pequenos ajustes nos tempos de resposta de cada operação ou realizando aprimoramentos na própria lógica de operação do robô. Outra melhoria solicitada pelos analistas da empresa foi a possibilidade de acionar o coletor sob demanda, em vez de ficar restrito apenas aos horários específicos definidos no projeto.

Um erro persistente acontece quando há uma queda no sistema SAP e devido ao método de comunicação entre a plataforma UiPath e SAP *script* o robô fica preso na operação esperando uma resposta até um usuário clicar na mensagem de erro na tela, após realizada a interação o RPA volta a operar sem outras intervenções. A solução utilizada para esse problema em sistemas mais complexos, onde são utilizados múltiplas automações simultaneamente, é a criação de um RPA secundário dedicado a monitorar o funcionamento dos robôs principais e reestabelecer o funcionamento correto quando detecta que uma tarefa está demorando muito além do esperado para ser concluída. Como é uma solução custosa, não foi implementada nesse protótipo, porém é uma melhoria a ser considerada em futuros projetos.

- [1] T. C. C. P. e Fernando Augusto Silva Marins, “Sistemas erp: características, custos e tendências,” *Revista Acadêmica de Tecnologia da Informação*, Ano desconhecido.
- [2] S. Chopra and P. Meindl, “Gerenciamento da cadeia de suprimentos: estratégia, planejamento, e operação,” in *Gerenciamento da Cadeia de Suprimentos: Estratégia, Planejamento, e Operação*, 2003, pp. 465–465.
- [3] A. L. Capelli, E. Stork, F. Schunski, M. Toazza, and T. Leoni, “Implementação e avaliação do sistema erp-sap na empresa john deere brasil - fábrica de tratores,” *CAD*, vol. 8, no. 1, pp. 38–49, jan-dez 2014.
- [4] L. Ivančić, D. Suša Vugec, and V. Vuksic, *Robotic Process Automation: Systematic Literature Review*, 08 2019, pp. 280–295.
- [5] Castertech, “Sobre nós - castertech,” 2025, acesso em: 26 jun. 2025. [Online]. Available: <https://www.castertech.com.br/>
- [6] UiPath, “About us - uipath,” 2024, acesso em: 24 fev. 2025. [Online]. Available: <https://www.uipath.com/about-us>
- [7] UiPath, “UiPath - documentation,” 2024, acesso em: 3 Mar. 2025. [Online]. Available: <https://docs.uipath.com/>
- [8] G. H. F. da Silva, “Aplicação do rpa nas organizações: Proposta de automação de processos,” Master’s thesis, Universidade Federal de Pernambuco, Recife/PE, 2023, curso de Gestão da Informação, Centro de Artes e Comunicação.
- [9] L. L. L. do Prado Ribeiro, “Contornando a barreira técnica: Um estudo de caso utilizando uipath para a automação de processos,” CUBATÃO, SP, 2023.
- [10] C. T. Kaya, M. Türkyılmaz, and B. Birol, “Impact of rpa technologies on accounting systems,” *Muhasebe ve Finansman Dergisi*, no. 82, 2019.
- [11] P. Marciniak and R. Stanislawski, “Internal determinants in the field of rpa technology implementation on the example of selected companies in the context of industry 4.0 assumptions,” *Information*, vol. 12, no. 6, p. 222, 2021.
- [12] J. Schuler and F. Gehring, “Implementing robust and low-maintenance robotic process automation (rpa) solutions in large organisations,” *Available at SSRN 3298036*, 2018.
- [13] A. B. de Moraes, “Aplicação de rpa (robot process automation) na criação de roteiros de fabricação em produtos tipo eto,” in *XLII Encontro Nacional de Engenharia de Produção (ENEGEP)*. Foz do Iguaçu, Paraná, Brasil: Associação Brasileira de Engenharia de Produção (ABEPRO), outubro 2022.
- [14] S. Madakam, R. M. Holmukhe, and D. K. Jaiswal, “The future digital work force: robotic process automation (rpa),” *JISTEM-Journal of Information Systems and Technology Management*, vol. 16, p. e201916001, 2019.
- [15] “Ieee guide for terms and concepts in intelligent process automation,” *IEEE Std 2755-2017*, pp. 1–16, 2017.
- [16] R. Sanchis, Ó. García-Perales, F. Fraile, and R. Poler, “Low-code as enabler of digital transformation in manufacturing industry,” *Applied Sciences*, vol. 10, no. 1, p. 12, 2019.
- [17] A. C. Bock and U. Frank, “Low-code platform,” *Business & Information Systems Engineering*, vol. 63, pp. 733–740, 2021.
- [18] E. Uzun, “A regular expression generator based on css selectors for efficient extraction from html pages,” *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 28, no. 6, pp. 3389–3401, 2020.
- [19] A. M. Tripathi, *Learning Robotic Process Automation: Create Software robots and automate business processes with the leading RPA tool—UiPath*. Packt Publishing Ltd, 2018.
- [20] R. F. do Brasil. (n.d.) Página 488 — sped/rfb. SPED – Sistema Público de Escrituração Digital, Receita Federal do Brasil. [Online]. Available: <http://sped.rfb.gov.br/pagina/show/488>
- [21] SAP, “Sap help portal,” 2025, acesso em: 31 Ago. 2025. [Online]. Available: <https://help.sap.com/docs/>