



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
Instituto de Ciência e Tecnologia de Sorocaba

WILLIAN YUDJI SUZUKI SAITO

**DESENVOLVIMENTO DE UM ROBÔ UTILIZANDO RPA (ROBOTIC
PROCESS AUTOMATION) E SOFTWARE DE BI (BUSINESS
INTELLIGENCE) PARA MELHORAR UM PROCESSO NA ÁREA DE
ENGENHARIA DE UMA EMPRESA**

Sorocaba

2024

Willian Yudji Suzuki Saito

DESENVOLVIMENTO DE UM ROBÔ UTILIZANDO RPA (ROBOTIC
PROCESS AUTOMATION) E SOFTWARE DE BI (BUSINESS
INTELLIGENCE) PARA MELHORAR UM PROCESSO NA ÁREA DE
ENGENHARIA DE UMA EMPRESA

Projeto Final de Curso apresentado como requisito parcial para a obtenção do título de Bacharel em Engenharia de Controle e Automação pela Universidade Estadual Paulista "Júlio de Mesquita Filho", Instituto de Ciência e Tecnologia (Câmpus de Sorocaba).

Orientador: Prof. Dr. Márcio Alexandre Marques

Sorocaba

2024

S158d	<p>Saito, Willian Yudji Suzuki</p> <p>Desenvolvimento de um robô utilizando RPA (Robotic Process Automation) e software de BI (Business Intelligence) para melhorar um processo na área de engenharia de uma empresa / Willian Yudji Suzuki Saito. -- Sorocaba, 2024</p> <p>75 p. : il., tabs., fotos</p> <p>Trabalho de conclusão de curso (Bacharelado - Engenharia de Controle e Automação) - Universidade Estadual Paulista (Unesp), Instituto de Ciência e Tecnologia, Sorocaba</p> <p>Orientador: Márcio Alexandre Marques</p> <p>1. Sistemas de computação. 2. Business intelligence. 3. Python (Linguagem de programação de computador). 4. Automação. 5. Software. I. Título.</p>
-------	--

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Ciência e Tecnologia, Sorocaba.

Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

DESENVOLVIMENTO DE UM ROBÔ UTILIZANDO RPA (ROBOTIC PROCESS
AUTOMATION) E SOFTWARE DE BI (BUSINESS INTELLIGENCE) PARA
MELHORAR UM PROCESSO NA ÁREA DE ENGENHARIA DE UMA EMPRESA

WILLIAN YUDJI SUZUKI SAITO

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO
COMO PARTE DO REQUISITO PARA A OBTENÇÃO DO GRAU DE
BACHAREL EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Prof. Dr. Everson Martins
Coordenador

BANCA EXAMINADORA:

Prof. Dr. MÁRCIO ALEXANDRE MARQUES
Orientador/UNESP – Câmpus de Sorocaba

Prof. Dr. EDUARDO VERRI LIBERADO
UNESP – Câmpus de Sorocaba

Prof. Dr. LEOPOLDO ANDRÉ DUTRA LUSQUINO FILHO
UNESP – Câmpus de Sorocaba

Este trabalho é dedicado à minha família e amigos.

AGRADECIMENTOS

Em primeiro lugar, não posso deixar de agradecer aos meus pais, que sempre se esforçaram ao máximo para me proporcionar as melhores condições de vida, dentro do que era possível. Suas abnegações, sacrifícios e amor incondicional foram alicerces fundamentais ao longo da minha trajetória. Eles não apenas me deram suporte, mas também muitas vezes abdicaram seus próprios sonhos e desejos para priorizar os meus. Sua presença constante e inabalável apoio, independente das circunstâncias, foram inspiradores e me impulsionaram a alcançar este importante marco em minha vida.

Aos meus amigos, meu eterno agradecimento. Suas parcerias, apoio, conselhos e os momentos de descontração compartilhados ao longo da minha jornada acadêmica foram inestimáveis. Juntos, enfrentamos desafios, celebramos conquistas e construímos memórias que levo comigo para toda a vida.

Ao meu orientador, Prof. Dr. Márcio Alexandre Marques, expresso minha sincera gratidão por sua orientação, auxílio, paciência, confiança e provocações intelectuais. Seu conhecimento, orientação e apoio foram essenciais para o desenvolvimento deste trabalho, e estou profundamente agradecido por sua dedicação em me ajudar a atingir meus objetivos acadêmicos.

Por fim, agradeço a todos os professores que, de alguma forma, contribuíram para a minha formação pessoal e profissional. Seus ensinamentos, conhecimentos e inspirações moldaram meu percurso acadêmico e me proporcionaram um alicerce sólido para enfrentar os desafios futuros.

Este trabalho não teria sido possível sem o apoio e influência positiva de todos vocês. Agradeço sinceramente por fazerem parte da minha jornada e por me ajudarem a chegar até aqui.

“A excelência é mundana. O desempenho superlativo é, na verdade, uma confluência de dezenas de pequenas habilidades ou atividades, cada uma delas aprendida ou descoberta por acaso, que foram cuidadosamente treinadas até se tornarem um hábito e, em seguida, são encaixadas em um todo sintetizado. Não há nada de extraordinário ou sobre-humano em nenhuma dessas ações; apenas o fato de que elas são feitas de forma consistente e correta e, juntas, produzem excelência.”

(Daniel F. Chambliss, 1989).

RESUMO

O cenário atual destaca a competição empresarial intensificada pela tecnologia digital, com a transformação digital como foco. As empresas de todos os setores estão investindo na digitalização de processos, o que tem levado ao surgimento de soluções como *Robotic Process Automation* (RPA) e *Business Intelligence* (BI) para melhorar a eficiência. A RPA, que usa "robôs" para automatizar tarefas repetitivas, está em ascensão, enquanto o BI coleta, organiza e analisa dados para embasar decisões. Ambas as tecnologias desempenham papéis vitais na evolução e competitividade das instituições. A falta de automação de processos leva a custos operacionais mais altos, erros humanos, processos confusos e complexos. O BI é crucial para lidar com a imensa quantidade de dados gerados na era digital. Investir nessas soluções é essencial para o sucesso e a competitividade no mercado atual. Neste contexto, este trabalho busca otimizar um processo interno de uma empresa, que envolve extração de dados do SAP ERP (sistema integrado de gestão empresarial da SAP SE), filtragem, validação e criação de um relatório no Excel, pois, este processo manual é demorado e suscetível a erros. O objetivo é aplicar uma solução tecnológica, combinando RPA para automatizar a extração de dados e o software de BI para criar um relatório interativo e eficiente. A solução consistiu em desenvolver um "robô" para o SAP ERP e um relatório dinâmico no software de BI, visando economia de tempo, redução de erros e fornecimento de *insights* valiosos para um determinado departamento da empresa. Os resultados obtidos mostram que a implementação da solução RPA substituiu um processo manual diário, gerando uma economia de cerca de 120 horas anuais. Ademais, a solução no Microsoft Power BI simplificou o processo de elaboração de relatórios, o que impactou vários engenheiros em diferentes regiões, resultando em uma economia de aproximadamente 1.656 horas anuais, além de melhorar a visualização e análise dos dados, com gráficos e filtros interativos. Por fim, apesar do avanço tecnológico atual, notou-se a persistência de muitos processos manuais, mesmo em grandes empresas, que podem ser automatizados, economizando tempo e recursos. Conclui-se que a automação é benéfica, independentemente do tamanho do processo.

Palavras-chave: transformação digital; RPA (*Robotic Process Automation*); *Business Intelligence* (BI); automação de processos; eficiência operacional.

ABSTRACT

The current scenario highlights the intensified business competition driven by digital technology, with digital transformation as its focal point. Companies across all sectors are investing in process digitization, leading to the emergence of solutions such as Robotic Process Automation (RPA) and Business Intelligence (BI) to enhance efficiency. RPA, which employs "robots" to automate repetitive tasks, is on the rise, while BI collects, organizes, and analyzes data to inform decision-making. Both technologies play vital roles in the evolution and competitiveness of institutions. The lack of process automation results in higher operational costs, human errors, and convoluted and complex processes. BI is crucial in dealing with the vast amount of data generated in the digital age. Investing in these solutions is essential for success and competitiveness in the current market. In this context, this work aims to optimize an internal process within a company involving data extraction from the SAP ERP (SAP SE's integrated enterprise resource planning system), filtering, validation, and the creation of an Excel report. This manual process is time-consuming and prone to errors. The goal is to apply a technological solution, combining RPA to automate data extraction and BI software to create an interactive and efficient report. The solution involved developing a "robot" for SAP ERP and a dynamic report in BI software, aiming for time savings, error reduction, and providing valuable insights for a specific department of the company. The results obtained indicate that the implementation of the RPA solution replaced a daily manual process, resulting in a savings of approximately 120 hours annually. Furthermore, the solution in Microsoft Power BI simplified the report creation process, impacting various engineers in different regions, resulting in a savings of approximately 1,656 hours annually, in addition to improving data visualization and analysis with interactive charts and filters. Finally, despite the current technological advancements, the persistence of many manual processes was observed, even in large companies, that could be automated, saving time and resources. It is concluded that automation is beneficial, regardless of the size of the process.

Keywords: digital transformation; RPA (Robotic Process Automation); Business Intelligence (BI); process automation; operational efficiency.

LISTA DE ILUSTRAÇÕES

Figura 1 - Processos elegíveis para a aplicação do RPA, considerando a frequência de execução e a complexidade do processo.	19
Figura 2 - Diagrama de espaguete representando o atual processo de obtenção das informações sobre horas trabalhadas e despesas.	30
Figura 3 - Menu lateral do sistema e botão para acessar esse menu.	34
Figura 4 - Botões na aba “Update Database” para abrir o explorador de arquivos.	35
Figura 5 - Botões na aba “Combine Excel Files” para abrir o explorador de arquivos.	36
Figura 6 - Exemplo de mensagem exibida para o usuário com a função “msg” ao iniciar o processo de atualização da base de dados.....	37
Figura 7 - Primeira aba da ferramenta: “Overview”.....	41
Figura 8 - Segunda aba: “Utilization Status”.....	43
Figura 9 - Terceira aba do relatório: “Actual x Available”	44
Figura 10 - Quarta aba do relatório: “Hours on Hold”	45
Figura 11 - Quinta aba do relatório: “SAP Fulfillment”.....	46
Figura 12 - Representação do processo de geração de um relatório antes e depois da solução desenvolvida no Microsoft Power BI.....	49
Figura 13 - Relatório de horas trabalhadas e despesas antes e depois da melhoria.	50

LISTA DE TABELAS

Tabela 1 - Tempo economizado pelos engenheiros com a implementação da aplicação.....	49
--	----

LISTA DE ABREVIATURAS E SIGLAS

BI – *Business Intelligence*

RPA – *Robotic Process Automation*

ERP – *Enterprise Resource Planning*

GUI – *Graphical User Interface*

PM – *Project Manager*

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Contextualização e Desafios	13
1.2	Objetivos e Justificativa	16
1.3	Estrutura deste trabalho	16
2	REVISÃO BIBLIOGRÁFICA E ASPECTOS TEÓRICOS	18
2.1	<i>Robotic Process Automation (RPA)</i>	18
2.2	Business Intelligence	20
2.3	Ferramentas (ou softwares) de Business Intelligence	23
2.3.1	Microsoft Power BI	23
3	MATERIAIS E MÉTODOS.....	24
3.1	Materiais.....	24
3.1.1	Ferramentas utilizadas para o desenvolvimento do sistema de RPA	25
3.1.2	Ferramentas/Plataformas auxiliares utilizadas durante o desenvolvimento do trabalho 27	
3.2	Métodos	28
3.2.1	Introdução.....	28
3.2.2	Desenvolvimento da primeira parte do projeto – robô (solução RPA)	30
3.2.3	Relatório no Microsoft Power BI.....	38
4	RESULTADOS E DISCUSSÕES	41
4.1	Criação dos painéis de exibição de dados.	41
4.1.1	Aba “Overview”	41
4.1.2	Aba “Utilization Status”	43
4.1.3	Aba “Actual x Available”	44
4.1.4	Aba “Hours on Hold”	45
4.1.5	Aba “SAP Fulfillment”	46
4.2	Compartilhamento da solução RPA e publicação do relatório desenvolvido no Microsoft Power BI.	47
4.3	Implementação do projeto e treinamento das partes interessadas	47
4.4	Ganhos com a otimização do processo de extração da base de dados do SAP ERP	48

4.5	Ganhos com a otimização do processo de geração do relatório de horas trabalhadas e despesas.	48
4.6	Melhoria visual do relatório	49
5	CONCLUSÃO.....	51
	REFERÊNCIAS.....	53
	APÊNDICE A - Estrutura da interface gráfica	57
	APÊNDICE B - Código que realiza a extração das bases de dados	60
	APÊNDICE C - Código que concatena as bases de dados	67
	APÊNDICE D - Código para inicializar a interface gráfica e estabelecer a ação de cada botão	68
	APÊNDICE E - Código da função “left menu”	70
	APÊNDICE F - Código das funções auxiliares da página "Update Database"	71
	APÊNDICE G - Código das funções auxiliares da página "Combine Excel Files"	72
	APÊNDICE H - Código da função "msg"	73
	APÊNDICE I - Código da função "resource_path"	74
	APÊNDICE J - Código para inicializar todo o sistema.....	75

1 INTRODUÇÃO

1.1 Contextualização e Desafios

Atualmente, a tecnologia digital é um dos fatores que mais intensifica a competição entre as empresas, onde o grande desafio de seus líderes é conseguir aplicar as novas tecnologias de forma adequada e ágil. Com o passar dos anos, observa-se que a importância dada à implementação de tecnologias digitais dentro das companhias cresceu de uma forma muito intensa. Isso se deu devido a sua alta capacidade de transformação, otimização e automação das organizações como um todo. Este conjunto de mudanças é chamado de transformação digital (Krotov, 2019).

Há pouco tempo, quando se falava em transformação digital, entendia-se que tal fenômeno era ligado apenas aos processos de produção das indústrias ou mesmo à melhoria da qualidade dos serviços ao cliente. Entretanto, nos dias de hoje, observa-se que todas as empresas, independente do setor de atuação, estão investindo fortemente na digitalização de seus processos e de sua parte operacional (Siderska, 2020).

Com o rápido desenvolvimento de novas tecnologias, aliado às constantes mudanças nas demandas do mercado, foram criados conceitos de gerenciamento. Nestes conceitos, soluções de RPA – *Robotic Process Automation* (Automação Robótica de Processos) e *softwares* de *Business Intelligence* (BI) (Inteligência de Negócio), da área da tecnologia da informação, são muito utilizados (Siderska, 2020).

A RPA é uma tecnologia emergente e que atualmente vem ganhando muito espaço dentro das empresas devido, principalmente, à automatização de processos – por meio de “robôs” – que essa tecnologia é capaz de realizar, reduzindo a mão de obra humana nas atividades do negócio, principalmente em tarefas simples e repetitivas (Geyer-Klingeberg *et al.*, 2018).

Segundo Oliveira (2018), a RPA é uma solução tecnológica de automação de processos que se utiliza de software para a criação de um “robô” capaz de reproduzir as ações de um ser humano em uma determinada atividade no computador. Dessa forma, é possível substituir a mão-de-obra humana em atividades muito repetitivas, de baixa importância ou operacionais dentro de uma empresa (Oliveira, 2018).

Devido à constante concorrência entre as organizações no mercado, é primordial que estas estejam sempre buscando ser eficientes, bem como renovar a maneira como são feitas as atividades. Um dos pilares para alcançar tais objetivos é por meio da automação de processos (onde a RPA está incluído). Ao deixar de investir na automação de processos, melhorias e em

novas tecnologias, muitas empresas podem acabar prejudicando a si mesmas diante ao mercado, visto que a implementação desse tipo de tecnologia é uma tendência mundial, correndo um grande risco de ficarem para trás perante outras organizações (Conexos, 2017).

Dentre os principais prejuízos decorridos da falta de investimento em automação de processos, de acordo com a *startup* Otimize Seu Negócio e com a empresa Conexos, tem-se (Otimize Seu Negócio, 2018; Conexos, 2017):

- **Maiores custos de operação:** Quanto menor for o nível de automação de processos, maior será o número de processos manuais a serem executados pelos funcionários. Isso acarreta um maior tempo gasto para a realização dessas atividades e uma menor produtividade, ou seja, um baixo aproveitamento dos recursos da empresa. No final, se todos esses custos não forem repassados para o cliente, estes se traduzirão em uma menor lucratividade para a empresa.
- **Falhas humanas:** Com a baixa automatização de processos, muitas atividades terão de ser executadas por pessoas e de forma manual. Isso se torna um problema a partir do momento em que se sabe que o ser humano está suscetível a erros, independente da sua habilidade e experiência em uma determinada atividade e, dependendo do erro cometido, pode gerar um alto prejuízo para a empresa, o que não ocorreria caso um software estivesse executando a mesma atividade.
- **Processos confusos e complexos:** Devido a execução manual dos processos, os resultados nem sempre são os mesmos. Em muitos casos, não há um padrão a ser seguido para a realização de uma determinada tarefa, tornando o processo confuso e muitas vezes mais complexo do que realmente deveria ser e, quanto maior a complexidade, maior o tempo de execução e, conseqüentemente, os custos envolvidos no mesmo.

Agora, ao estudar sobre *Business Intelligence*, verifica-se que este se trata de um processo que tem como objetivo coletar os dados que são gerados pela companhia, organizá-los e analisá-los de uma forma profunda, de forma a gerar informações reais e concretas que embasarão as tomadas de decisão da empresa. O principal intuito do *Business Intelligence* é conseguir obter novas ideias e melhorar os aspectos bem como os departamentos de uma instituição a partir das informações e dados gerados por ele (Silva, 2021).

Com a transformação digital cada vez mais presente, a quantidade de dados que são gerados por segundo é imensa. Tais dados têm sua origem em diversos dispositivos e locais, além de existirem em muitos formatos e tipos diferentes. Entretanto, de nada adianta possuir

dados e não conseguir extrair informações úteis deles. Neste contexto, o *Business Intelligence* pode ser implementado em qualquer lugar, uma vez que os dados estão presentes em todas as áreas de uma empresa (Romero *et al.*, 2021).

Da mesma forma como ocorre com a automação de processos, ao deixar de investir na utilização de softwares de *Business Intelligence*, a empresa passa a ter alguns prejuízos, tais como (Fia Business School, 2021):

- **O aumento dos riscos nos quais está exposta:** Com a ausência de resultados provenientes das análises de dados realizadas pelo software de BI, as decisões da empresa passam a ser menos embasadas, o que aumentam os riscos de realização de movimentos errados que podem prejudicá-la.
- **Deixará de aumentar o retorno sobre os investimentos:** Sem o auxílio dos softwares de BI, a empresa deixará de obter muitos *insights* que essas ferramentas podem proporcionar, o que acarretará tomadas de ações que não possuem fundamentos muito bem estruturados e que, no futuro, podem ou não trazer resultados positivos, funcionando da mesma maneira como as apostas. Por outro lado, com a utilização inteligente dos softwares de BI, as decisões tomadas são mais bem embasadas e, portanto, mais assertivas, aumentando as chances de retornos positivos sobre os investimentos.
- **Aumento dos custos:** Devido à realização manual das análises de dados, gasta-se muito tempo (e dinheiro) que um software de BI não usaria. Além disso, deve-se levar em consideração que muitas dessas análises podem ser infrutíferas, o que significa um gasto ainda maior.
- **Visão reduzida do desempenho da empresa:** Sem a utilização dos softwares de BI, torna-se mais difícil a visualização dos efeitos de uma determinada ação tomada pela empresa, o que a prejudicará caso essa ação precise sofrer algum tipo de ajuste ou intervenção, pois a agilidade na detecção do erro estará comprometida.

Neste contexto, existe um processo que é executado dentro da empresa onde o autor trabalha que pode ser otimizado por meio da implementação das tecnologias de RPA e *Business Intelligence*.

Inicialmente, o processo consiste na extração de dados sobre as horas trabalhadas, bem como sobre as despesas dos engenheiros de campo da empresa, a partir de um software utilizado pela companhia, chamado SAP ERP (*Enterprise Resource Planning*). Com os dados em mãos,

é feita uma filtragem para separar apenas os dados de interesse e que podem ser contabilizados. Feito isso, valida-se os mesmos com o intuito de buscar inconsistências e erros. Por fim, utilizando o software Microsoft Excel, é montado um relatório com as informações consolidadas. Todo esse processo é feito de forma manual e envolve muitas pessoas, consumindo muito tempo e, devido à suscetibilidade ao erro humano, o relatório gerado ao final do processo ainda possui muitos dados inconsistentes, além de não possuir um visual muito amigável, por ser apenas uma planilha com milhares de linhas e valores.

1.2 Objetivos e Justificativa

Diante desse contexto, o objetivo deste trabalho foi desenvolver e aplicar uma solução tecnológica – envolvendo RPA e software de BI – no processo supracitado, de forma a reduzir o número de etapas do mesmo, otimizar o tempo despendido, substituir a mão de obra humana e fornecer *insights* valiosos, a partir dos dados analisados e processados pelo software de BI, para o departamento onde este projeto será aplicado.

A solução proposta é constituída por duas partes. A primeira se baseia no desenvolvimento de um “robô” (software) – utilizando RPA – capaz de executar o processo de extração da base de dados no SAP ERP, com o objetivo de substituir o trabalho do funcionário que atualmente executa este processo de forma manual. Já a segunda parte consiste no desenvolvimento de um relatório, em um software de BI, capaz de realizar os devidos tratamentos e transformações dos dados automaticamente e, ao final do processo, fornecer um painel mais amigável – dinâmico, interativo e visual – com todas as informações relevantes sobre as despesas e horas trabalhadas pelos engenheiros de campo pertencentes ao departamento onde o projeto está sendo implementado.

1.3 Estrutura deste trabalho

Esse Projeto Final de Curso está dividido em cinco capítulos:

O primeiro capítulo contextualiza o tema do trabalho, por meio da apresentação dos desafios enfrentados pela área que abrange a aplicação de ferramentas de RPA e *Business Intelligence* dentro da empresa, considerando a transformação digital que têm ganhado bastante foco atualmente. Além disso, apresenta os objetivos e justificativa do projeto.

O segundo capítulo introduz a parte teórica do trabalho, tomando como base trabalhos relacionados da área, descrevendo o que é *Robotic Process Automation* (RPA), sua importância

e como e quando essa tecnologia deve ser aplicada. Explica também o que é o *Business Intelligence* (BI), onde ele pode ser aplicado e quais são seus benefícios. Além disso, traz um exemplo e descreve uma das ferramentas de BI mais utilizadas no mercado, o Microsoft Power BI.

No terceiro capítulo é apresentada toda a metodologia do trabalho para o desenvolvimento tanto da solução RPA, quanto do relatório dentro do Microsoft Power BI, desde o entendimento do processo a ser melhorado, passando pelas etapas de construção das soluções, até a implementação.

O quarto capítulo demonstra quais foram os resultados e benefícios obtidos, com a implementação da solução RPA e do relatório feito no Microsoft Power BI, para a empresa onde este trabalho foi realizado.

Por fim, o quinto capítulo apresenta a conclusão deste trabalho, elencando as contribuições realizadas.

2 REVISÃO BIBLIOGRÁFICA E ASPECTOS TEÓRICOS

Neste capítulo será apresentado o conceito de *Robotic Process Automation* (RPA), como também as métricas que devem ser analisadas nos processos para poder implementar tal tecnologia. Além disso, será abordado o conceito de *Business Intelligence* (BI) e a sua importância dentro das organizações. Adicionalmente, será apresentado um dos principais softwares de BI utilizado no mercado, o Microsoft Power BI.

2.1 *Robotic Process Automation* (RPA)

A RPA é amplamente entendida como sendo uma solução baseada em software, para a automatização de processos do negócio, capaz de executar determinadas tarefas humanas, substituindo esta mão-de-obra. A RPA é normalmente aplicado em tarefas que envolvem muitas análises de dados ou que são muito repetitivas e que obedecem a um determinado padrão (Dialani, 2019).

A abordagem sobre esse assunto, na maioria das vezes, pode ser encontrada de três diferentes formas. Alguns se referem a este como sendo uma tecnologia emergente, outros dizem que se trata de uma ferramenta de software e, por último, também são encontrados autores que definem essa tecnologia com uma abordagem voltada para a automação de processos (Geyer-Klingeberg *et al.*, 2018).

Entretanto, independente da abordagem ou definição utilizada, todos os autores mencionam como resultado da aplicação dessa tecnologia, a redução da carga de trabalho humana, principalmente em tarefas simples e repetitivas, devido à automatização das mesmas proporcionada por essa tecnologia. Na maioria das vezes, essa ferramenta é desenvolvida com lógicas muito simples e é capaz de interagir com múltiplas plataformas existentes nos sistemas. Dessa forma, permite que as pessoas, ou funcionários, utilizem o seu tempo e criatividade em tarefas que agreguem mais valor para o negócio (Geyer-Klingeberg *et al.*, 2018).

A RPA é uma tecnologia que utiliza robôs baseados em softwares para automatizar processos, imitando ações humanas dentro da interface gráfica do sistema. Essa automação é projetada para reduzir a carga de trabalho dos funcionários em tarefas repetitivas, contribuindo para a transformação digital. Pires *et al.* (2020) destacam que a RPA é eficaz na automação de atividades repetitivas e rotineiras, executando tarefas em computadores de forma semelhante a uma pessoa. Ele é uma ferramenta que visa otimizar a execução de tarefas por meio da automação de processos, aumentando a eficiência e reduzindo o tempo necessário para sua conclusão (Bahense, 2018; Pires *et al.*, 2020; Quinn *et al.*, 2013).

Após o conhecimento do conceito do RPA, é possível presumir que qualquer tarefa pode ser automatizada com a utilização dessa tecnologia e, em partes, isso é verdade. Todavia, existe um determinado grupo de tarefas que é elegível à aplicação do RPA, e é de fundamental importância que a organização seja capaz de filtrar essas tarefas antes de empregar a tecnologia.

De acordo com Siderska (2020), uma seleção apropriada dos processos que serão automatizados é muito importante uma vez que, para essa tecnologia, o estágio de concepção da ideia é completamente diferente do estágio de sua implementação no negócio. Portanto, para selecionar o conjunto de atividades em que a RPA será aplicado, é fundamental que seja feita uma análise de todos os processos, considerando a frequência de execução e a complexidade dessas tarefas (Siderska, 2020).

Segundo Sibalija, T., Jovanović, S., Đurić, J. (2018), pode-se dividir os processos em três grupos e classificá-los quanto a implementação do RPA, sendo eles (Sibalija *et al.*, 2018):

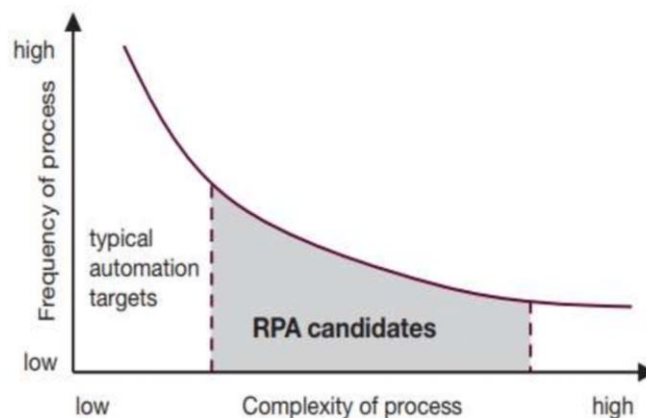
1º grupo: Processos de **baixo** grau de complexidade e com **alta** frequência de execução. Para este grupo, métodos mais simples e tradicionais são capazes e mais indicados para automatizar esses processos.

2º grupo: Processos de grau de complexidade **intermediário** e com **alta** frequência de execução. Esse grupo de processos é o **ideal** para a aplicação do RPA.

3º grupo: Processos de **alto** grau de complexidade e com **baixa** frequência de execução. Neste grupo, não é indicado a utilização de RPA para a automação das atividades.

A Figura 1 apresenta os três grupos de processos e suas respectivas classificações.

Figura 1 - Processos elegíveis para a aplicação do RPA, considerando a frequência de execução e a complexidade do processo.



Fonte: Sibalija *et al.* (2018).

De acordo com Fung (2014), existem quatro critérios básicos para avaliar a possibilidade de aplicação do RPA em um processo, sendo eles (Fung, 2014):

- Baixa complexidade cognitiva. A RPA não conseguirá trabalhar bem em processos muito complexos e que envolvam atividades complicadas.
- Fácil acesso à sistemas e plataformas envolvidas no processo. A RPA normalmente trabalha com aplicações que já existem no sistema e que não requerem nenhum tipo de autenticação ou verificação de segurança.
- Alta frequência de execução. Processos que precisam ser executados com uma alta frequência são ótimos candidatos ao RPA.
- Alta probabilidade de erro humano e com baixa variabilidade do processo.

Ao ter conhecimento sobre o conceito, bem como os requisitos necessários para a adoção do RPA, verifica-se que essa tecnologia pode ser utilizada em diversas áreas de uma empresa e em empresas de todos os tipos.

Atualmente, a RPA é aplicado, por exemplo, nas áreas de recursos humanos, tecnologia da informação, seguros, telecomunicação, educação, bancária, logística, entre outras. Com a implementação desta tecnologia dentro das organizações, pode-se obter inúmeros benefícios, tais como:

1. Redução do custo operacional;
2. Aumento da confiabilidade e conformidade dos processos;
3. Excelência na execução de um processo que envolva diversos sistemas ou plataformas interconectadas;
4. Rápido retorno sobre o investimento na tecnologia;
5. Processos são executados muito mais rapidamente;
6. Funcionários conseguem dedicar seu tempo em outras tarefas mais importantes, que necessitem maior nível de criatividade e agreguem mais valor para o negócio;
7. Aumento da repetibilidade e reprodutibilidade dos processos.

2.2 Business Intelligence

O Business Intelligence (BI) ou Inteligência de Negócios, é um termo dado a um conjunto de processos que utiliza os dados para gerar informações úteis para melhorar a tomada de decisões, independente da área ou departamento dentro de uma empresa.

De acordo com a Tableau (2023) – empresa americana líder no desenvolvimento de softwares para visualização de dados – o Business Intelligence é o nome dado a um conjunto de processos e métodos de coleta, armazenamento e análise de dados referentes às atividades

do negócio, com o intuito de aumentar sua eficiência como também a visão que se tem da empresa, permitindo uma melhor tomada de decisões (Tableau, 2023).

Segundo a Microsoft (2023a), empresa americana que também possui software próprio de visualização de dados, o BI fornece informações para a tomada de decisões estratégicas. As ferramentas de Business Intelligence analisam dados históricos e atuais e exibem os resultados em um formato visual intuitivo (Microsoft, 2023a).

Seguindo uma linha semelhante, para Djerdjouri (2020), o Business Intelligence (BI) é definido como um processo de tomada de decisão baseado nos resultados das análises de dados de um negócio. E, tendo em vista que hoje em dia os dados são gerados a todo momento e são um dos bens mais importantes de um negócio, o BI se tornou um recurso valioso e que pode alavancar o crescimento e o sucesso de uma organização (Djerdjouri, 2020).

De acordo com a definição de Patel (2023), o BI consiste na coleta, na organização, na análise, no compartilhamento e no monitoramento de informações para melhorar a gestão de negócios. Ele diz que essa tecnologia é uma ferramenta que gera *insights* para a tomada de decisão com base nos dados reais do negócio, o que reflete de forma muito positiva por toda a empresa (Patel, 2023).

Dadas as definições de Business Intelligence, fica claro que, de uma maneira geral, o BI é um conjunto de processos que visa coletar dados em sua forma bruta, transformá-los em informações e entregá-las para as pessoas, para que estas tomem decisões mais bem embasadas. Esse conjunto de processos pode ser dividido em quatro etapas:

1. A primeira consiste na coleta e no tratamento de dados provenientes de várias fontes distintas;
2. A segunda é responsável por organizar e analisar todos os dados coletados, com o objetivo de encontrar padrões bem como inconsistências nesses dados;
3. Já na terceira, é o momento em que os dados são apresentados de uma forma visual (sejam através de gráficos, mapas etc.), para facilitar a visualização dos resultados obtidos a partir das análises previamente realizadas;
4. Por último, são tomadas as decisões e as novas medidas são implementadas, ao mesmo tempo em que é feita a avaliação dos resultados alcançados.

Em complemento à esta última etapa, Silva (2021) propõe uma reflexão interessante, dizendo que, além de captar as informações, se utilizar dos dados e implementar as novas ideias, é preciso mensurar o quão positivos foram os resultados gerados, e se os dados realmente ajudaram a empresa de alguma forma (Silva, 2021).

Após a chegada da indústria 4.0, o ambiente organizacional tem se tornado cada vez mais complexo e dinâmico – exigindo tomadas de decisões mais rápidas e eficientes – e, por isso, as empresas adotaram uma mentalidade que visa a constante busca de inovações e novas tecnologias (Ahmad *et al.*, 2020). Neste cenário, os softwares de BI ganharam bastante espaço, devido à alta capacidade de processamento de dados, à geração de informações úteis para o negócio, e a sua alta aplicabilidade nas mais diversas áreas de uma empresa. Dentre os benefícios dessa ferramenta, têm-se a eficiência de informação e o gerenciamento de dados (Negro; Mesia, 2020).

Alguns exemplos de áreas onde o BI pode ser aplicado:

- Na área de vendas e *marketing*: Com o acesso às informações a respeito das necessidades do cliente, bem como seu comportamento perante a um produto ou serviço, pode-se tomar iniciativas que ofereçam ao consumidor exatamente o que ele está procurando, aumentando a receita da empresa. Além disso, é possível fazer o acompanhamento da *performance* da companhia nesta área.
- Na área de Operações da empresa: Com o BI, é possível automatizar as atividades de análise de dados e refinar processos desse departamento, aumentando a eficiência e a produtividade.
- Na área de Finanças: Construindo painéis dinâmicos e personalizados, é possível ter em mãos uma visão abrangente e sempre atualizada sobre a integridade financeira da empresa. Ademais, com o BI, é possível analisar dados históricos, prever tendências e calcular riscos.
- Na área de Estocagem: Pode-se melhorar o gerenciamento do estoque, com informações em tempo real e na palma da mão, além de obter relatórios, otimizar o atendimento e prever necessidades de compra.

Patel (2023) cita alguns benefícios do BI para as empresas, entre eles: o aumento do faturamento do negócio; a ampliação do controle sobre os riscos; o aumento da competitividade da empresa perante as outras; aumento na facilidade na identificação de oportunidades como também de falhas; otimização das rotinas e processos internos; redução de custos; e aumento da eficiência na gestão de informações (Patel, 2023).

Já Silva (2021), aponta benefícios muito semelhantes, como: a descoberta de novas oportunidades para o negócio; geração de ideias para o aprimoramento de processos; auxílio na identificação de detalhes que possam estar gerando gastos excessivos; melhoria na percepção

que se tem dos clientes e fornecedores; e ajuda a empresa a se manter atualizada frente às constantes mudanças do mercado (Silva, 2021).

2.3 Ferramentas (ou softwares) de Business Intelligence

Para a implementação do Business Intelligence existem muitos softwares e plataformas que normalmente são utilizados para facilitar o processo. Essas ferramentas são capazes de se conectar com diversas fontes de dados, coletar e processar esses dados, sendo eles estruturados ou não. Além disso, também têm o papel de facilitar a busca por informações e auxiliar na construção de painéis ou relatórios de visualização de dados. Entre os principais softwares de Business Intelligence, tem-se (Ferreira, 2020):

- a. Google Data Studio
- b. Microsoft Power BI
- c. Tableau
- d. QlikView
- e. Board

2.3.1 Microsoft Power BI

Para o desenvolvimento de parte deste trabalho, foi utilizado o software Microsoft Power BI como ferramenta de Business Intelligence, justamente pelo fato da empresa onde o autor trabalha já possuir a licença de uso desse software.

Como foi citado anteriormente, o Microsoft Power BI é um dos principais softwares de Business Intelligence disponíveis no mercado atual. É uma ferramenta desenvolvida pela Microsoft, e seu principal objetivo é facilitar a análise e a visualização de dados, de forma a fornecer *insights* mais profundos sobre os dados de uma empresa (Fia Business School, 2020).

Um dos pontos mais fortes sobre esse software é a sua semelhança – em termos de interface gráfica – com os outros aplicativos da empresa pertencentes ao famoso pacote Microsoft Office, como o Word, Excel, PowerPoint, entre outros. Isso faz com que os usuários se sintam mais familiarizados com o ambiente dentro do software, facilitando o seu uso. A ferramenta está disponível em três versões (Fia Business School, 2020):

- Power BI Desktop: Assim como qualquer programa instalado nos computadores, incluindo o Word, Excel e PowerPoint, também é possível fazer a instalação do Power BI.
- Serviço do Power BI: Funciona totalmente na nuvem, ou seja, tudo fica armazenado na internet. É caracterizado como um SaaS (software como serviço), onde o usuário utiliza

a ferramenta diretamente pela internet, não precisando instalar nenhum outro componente em sua máquina, exceto o navegador para acessar a internet. Uma das qualidades desta versão, é a velocidade de atualização quando novos dados são inseridos.

- Aplicativo para *mobile* Power BI: Devido ao grande uso de *smartphones* e *tablets*, a empresa também desenvolveu uma versão do Power BI para estes dispositivos. Dessa forma, é possível acessar os relatórios e *dashboards* de qualquer lugar e de uma maneira muito fácil.

O Microsoft Power BI, além das características supracitadas, possui uma interface gráfica bastante intuitiva e que facilita muito a usabilidade do software, permitindo que pessoas que não são especializadas no assunto sejam capazes de utilizar a ferramenta por conta própria, gerando seus próprios relatórios e painéis, podendo ainda compartilhá-los com outras pessoas, aumentando a autonomia e agilizando o processo – esse tipo de ferramenta é chamada de BI de autoatendimento (Palmieri, 2021).

Além disso, o Microsoft Power BI é capaz de transformar os dados em gráficos ou outros tipos de visualizações de dados, em apenas alguns cliques, sem a necessidade de escrever códigos de programação. Dessa forma, é muito fácil visualizar os dados ou criar combinações de dados em um mesmo gráfico, podendo escolher qual o melhor tipo de gráfico para a apresentação dos dados ou até mesmo permitir que o software faça uma recomendação de visualização (Pariz, 2021).

Por último, outra característica do Power BI, é a possibilidade de acesso aos relatórios e painéis criados de qualquer lugar, por meio de qualquer dispositivo móvel. Sendo assim, não é preciso estar no escritório ou com um computador para fazer as análises ou consultas de dados, isso porque os *dashboards* ficam disponíveis na nuvem e podem ser acessados a qualquer momento por meio de um *smartphone* ou *tablet* (Coutinho, 2021).

3 MATERIAIS E MÉTODOS

3.1 Materiais

- Dados utilizados durante o desenvolvimento do projeto:

Todos os dados aqui mostrados foram fornecidos pela empresa onde a plataforma foi implementada. Porém, para a apresentação deste trabalho, estes dados foram modificados de forma a não comprometer ou prejudicar a empresa. Entretanto, as estruturas das bases de dados utilizadas foram mantidas conforme as originais.

- *Hardware e softwares* para a elaboração do projeto:

Com relação ao hardware, foi utilizado um *notebook* da empresa onde o autor trabalha, o qual possui sistema operacional Windows 10, processador Intel Core i5 de 7ª geração, placa de vídeo integrada e 16 gigabytes de memória RAM.

Em termos de softwares, foram utilizados seis, a saber:

- Microsoft Power BI (licença adquirida pela empresa);
- Microsoft Excel (licença adquirida pela empresa);
- Microsoft Sharepoint (licença adquirida pela empresa);
- SAP ERP (*Enterprise Resource Planning*) – licença adquirida pela empresa.
- Pycharm Community Edition (gratuito).
- Qt Designer (gratuito).

3.1.1 Ferramentas utilizadas para o desenvolvimento do sistema de RPA

3.1.1.1 Python

Uma parte constituinte deste trabalho consiste em um sistema para automatizar o processo de extração de uma base de dados do SAP ERP. Os algoritmos deste sistema foram implementados em Python, que é uma linguagem de programação interpretada, orientada a objetos e de alto nível, o que a torna extremamente versátil e, conseqüentemente, permitindo o desenvolvimento de diversas aplicações diferentes (Python Software Foundation, 2023a).

3.1.1.2 PyCharm

O *software* PyCharm está disponível em várias plataformas, incluindo Windows, MacOS e Linux, o que o torna acessível a uma ampla gama de sistemas operacionais. Criado pela renomada empresa JetBrains, o PyCharm oferece uma série de recursos valiosos para os desenvolvedores, como análise de código, um depurador gráfico, sugestões automáticas de código e funcionalidades de navegação, tornando a tarefa de programação mais eficiente e produtiva (Alura, 2023). Neste trabalho, o PyCharm foi utilizado para escrever o código do sistema RPA.

3.1.1.3 Qt Designer

O Qt Designer é a ferramenta do Qt destinada à criação e desenvolvimento de interfaces gráficas de usuário (GUIs) com Qt Widgets. Com ele, é possível compor e personalizar janelas ou diálogos, além de possibilitar a realização de testes utilizando diferentes estilos e resoluções

(The Qt Company, 2023). Neste trabalho, o Qt Designer foi utilizado para desenvolver a interface gráfica do sistema RPA.

3.1.1.4 Bibliotecas utilizadas dentro do código de programação:

- **OS**

O módulo 'os' em Python representa uma biblioteca padrão altamente valiosa no contexto de interação com o sistema operacional. Essa biblioteca oferece uma variedade de recursos para a realização de tarefas específicas, como a navegação em diretórios, criação de novos diretórios, execução de comandos no terminal e a aquisição de informações do sistema (Ramos, 2023).

- **Time**

A biblioteca 'time' fornece ao usuário ferramentas para acessar informações temporais em Python, incluindo o uso de tipos de dados conhecidos como “*timestamps*”, que representam momentos específicos no tempo. Além disso, essa biblioteca permite calcular o tempo de execução de iterações ou funções, o que é particularmente útil para aqueles que desejam comparar os tempos de resposta dos modelos, com o objetivo de aprimorar eficiência e reduzir custos (Barbosa Filho, 2022).

- **Pandas**

O Pandas é uma biblioteca de código aberto para Ciência de Dados, desenvolvida em Python. Essa biblioteca oferece uma abordagem ágil e flexível, fornecendo estruturas sólidas para lidar com dados relacionais (ou dados com rótulos) de maneira simples e intuitiva (Almeida, 2023).

- **PySide6**

O PySide6 é uma biblioteca de interface gráfica do usuário (GUI) para Python, permitindo a criação de aplicativos com interfaces intuitivas e visualmente atraentes. Ele oferece uma ampla gama de recursos e funcionalidades e é construído com base no Qt, um *framework* de desenvolvimento multiplataforma amplamente adotado na indústria de software (Awari, 2023).

- **Sys**

O módulo 'sys' em Python disponibiliza uma série de funções e variáveis utilizadas para gerenciar várias partes do ambiente de execução do Python. Esse módulo permite interagir com o interpretador, pois concede acesso às variáveis e funções que desempenham um papel fundamental na interação com o interpretador (Módulo [...], [ca. 2022]).

- **Qt-material**

O módulo 'qt-material' é uma biblioteca de estilos para PySide6. Com ele, é possível aplicar estilos e *designs* pré-definidos pela biblioteca, em uma interface gráfica desenvolvida com PySide6, de uma forma muito simples e prática.

- **Pyautogui**

O 'pyautogui' é um módulo Python de automação de *GUI* (*Graphical User Interface* ou Interface Gráfica do Usuário). É utilizado para controlar, programaticamente, o *mouse* e o teclado do computador (Python Software Foundation, 2023b).

- **Datetime**

A biblioteca 'datetime' disponibiliza diversas ferramentas para criar e manipular séries temporais em estruturas de dados, como os *dataframes*. Quando usada em conjunto com o Pandas, ela se torna uma poderosa ferramenta, sendo especialmente valiosa para aqueles que trabalham com análise de séries temporais (Barbosa Filho, 2022).

- **Pyperclip**

O Pyperclip é um módulo Python multiplataforma que oferece funções para copiar e colar na área de transferência (Python Software Foundation, 2023c).

3.1.2 Ferramentas/Plataformas auxiliares utilizadas durante o desenvolvimento do trabalho

3.1.2.1 Power Query

O Power Query representa uma solução específica dentro do domínio do Business Intelligence (BI) que desempenha um papel essencial na tarefa de estruturar os dados de acordo com os requisitos específicos do sistema de BI da Microsoft. Isso possibilita a coleta e a transformação de dados anteriormente não estruturados, provenientes de planilhas

desorganizadas e confusas, transformando-os em uma versão de fácil visualização e gerenciamento.

Adicionalmente, essa ferramenta oferece recursos de automação que simplificam as configurações e edições de dados, acelerando o processo de ingestão de dados de maneira eficiente e prática (Restum, 2023).

3.1.2.2 Microsoft Sharepoint

O Microsoft SharePoint é uma plataforma *web* multifuncional desenvolvida pela Microsoft, projetada para promover a colaboração entre grupos de aplicação. Em sua essência, o SharePoint é uma intranet inteligente e acessível por dispositivos móveis, que desempenha um papel vital como um portal corporativo, facilitando a gestão de informações, projetos, fluxos de trabalho e equipes. Suas amplas funcionalidades o destacam como uma ferramenta incrivelmente valiosa para o crescimento e desenvolvimento das empresas (Lattine Group, 2023).

3.1.2.3 Workspace no Power BI

Os Workspaces são ambientes destinados à colaboração com colegas e à criação de coleções que abrangem painéis, relatórios, conjuntos de dados e relatórios paginados (Microsoft, 2023b).

3.2 Métodos

3.2.1 Introdução

Para desenvolver este trabalho, inicialmente realizou-se um estudo acerca do funcionamento do atual processo como um todo, com objetivo de entender as etapas que o compõem, verificar as partes envolvidas bem como suas demandas e “dores” e, eventualmente, já conseguir identificar alguma oportunidade de melhoria.

Durante este estudo, foram feitas algumas reuniões com um dos gerentes de projeto do departamento e com o gerente do departamento.

Assim, foram mapeadas as etapas do processo, composta por 04 etapas:

1. Extração da base de dados por meio do software SAP ERP – processo executado manualmente por um colaborador.
2. Ocorrência de diversas conversas e troca de *e-mails* entre os gerentes de projeto, engenheiros, gerente do departamento e departamento de operações, para mapear – de

forma totalmente imprecisa – as horas trabalhadas bem como as despesas dos engenheiros dentro dos projetos realizados no mês.

3. Com a base de dados em mãos, juntamente com as informações obtidas pelas discussões, os valores são confrontados a fim de verificar sua precisão.
4. Depois dos dados confrontados e muitas conversas, chega-se a um consenso e alguns valores de horas trabalhadas e despesas são definidos para aquele momento.

Ao mesmo tempo em que o processo foi sendo mapeado, foram registradas algumas “dores” enfrentadas pelo departamento, sendo elas:

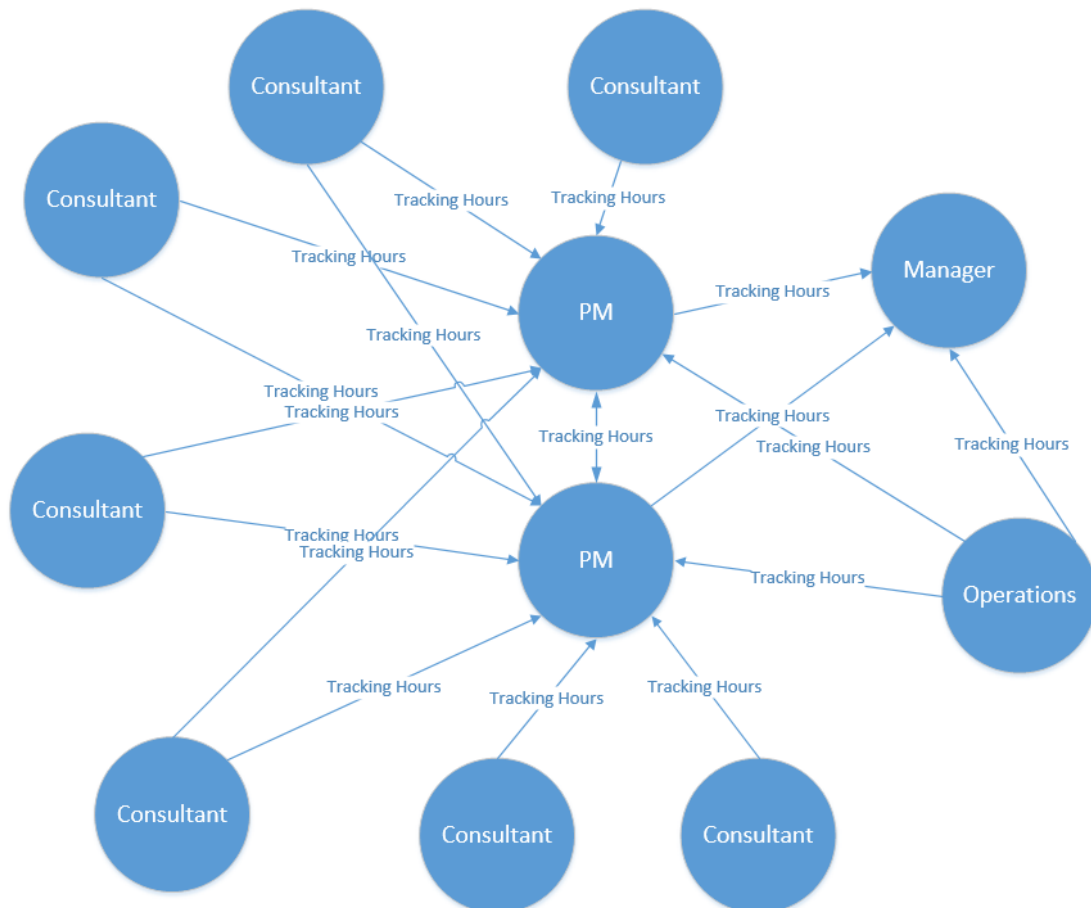
- Esse processo é realizado **sempre** que as informações de horas trabalhadas e despesas de cada projeto são solicitadas, seja pelo gerente do departamento ou por outros colaboradores, ou seja, diversas vezes ao mês – às vezes, mais de uma vez ao dia – o que consome muito tempo.
- Há uma dificuldade na disseminação das informações, uma vez que esta é realizada por meio de *e-mails* e/ou conversas, além de estarem muito dispersas entre os vários colaboradores envolvidos, o que gera certa imprecisão nos dados.
- A base de dados extraída do sistema (SAP ERP) é muito extensa, o que dificulta a obtenção das informações requeridas de forma precisa, visto que a filtragem e o tratamento de dados são feitos manualmente.
- O processo de extração de dados é feito de forma manual, o que consome tempo.
- Ao final do processo, o resultado é sempre um relatório de horas e despesas com valores aproximados, nunca precisos.

As partes envolvidas no processo e a função de cada uma delas é descrita da seguinte forma:

Os gerentes de projeto ficam no centro de tudo, sendo responsáveis por entrar em contato com todos os engenheiros para coletar as despesas feitas e as horas que estes haviam trabalhado e registrado no sistema. Concomitantemente, eles conversam entre si para manter o alinhamento das informações ao mesmo tempo que mantém contato tanto com o departamento de operações quanto com o gerente do departamento, a fim de validar se todos os valores obtidos com os engenheiros estão de acordo com o que está registrado no sistema. Por fim, após diversas validações e discussões, os valores são definidos e enviados para quem os solicitou, podendo ser qualquer colaborador da empresa.

A Figura 2 ilustra, por meio de um diagrama de espaguete, as partes envolvidas durante o processo de obtenção das informações de horas trabalhadas e despesas realizadas, onde a sigla “PM” significa *Project Manager* (Gerente de Projeto).

Figura 2 - Diagrama de espaguete representando o atual processo de obtenção das informações sobre horas trabalhadas e despesas.



Fonte: Rockwell Automation (2023)

3.2.2 Desenvolvimento da primeira parte do projeto – robô (solução RPA)

Para o desenvolvimento do robô (solução RPA) utilizou-se a linguagem de programação Python. Tendo em vista de que uma solução RPA imita as ações humanas (conforme foi descrito na [seção 2.1](#)) desenvolveu-se o algoritmo responsável por controlar as ações do robô “traduzindo” as ações humanas realizadas no processo de extração da base de dados, em código de programação.

Para extrair a base de dados do SAP ERP, é preciso acessar o sistema por meio de um navegador *web* (neste caso, foi utilizado o Google Chrome); preencher 2 (dois) formulários para definir os dados que serão extraídos; gerar dois arquivos em Excel – os quais devem ser salvos

em uma pasta específica do departamento da empresa – e, por fim, concatená-los em um único arquivo. Vale comentar aqui que é possível configurar o *layout* da base de dados que será extraída do SAP ERP, ou seja, é possível selecionar as colunas que irão compor a base de dados gerada pelo sistema. Isso pode ser feito na área de configurações da página que contém o formulário a ser preenchido.

O desenvolvimento da solução RPA foi dividida em 6 (seis) etapas:

- Etapa 1: Construção da interface gráfica da solução.
- Etapa 2: Elaboração do código responsável por acessar o SAP ERP para extração das bases de dados.
- Etapa 3: Desenvolvimento do código para concatenar as bases de dados extraídas do SAP ERP.
- Etapa 4: Implementação do código para interligar os códigos das etapas 2 e 3 com a interface gráfica (etapa 1).
- Etapa 5: Inclusão de dois blocos de código de programação para o bom funcionamento do programa. Um para evitar que um erro conhecido ocorra no momento de conversão do código em um arquivo executável, e outro para executar, exibir, e estabelecer o design do sistema.
- Etapa 6: Conversão do código desenvolvido em um arquivo executável.

3.2.2.1 Desenvolvimento da interface gráfica da solução

Ao desenvolver a interface gráfica da solução, utilizou-se o software Qt Designer, por ser de fácil utilização e por não precisar de código de programação para o desenvolvimento.

A estrutura da interface gráfica é constituída por quatro abas e um menu lateral para navegar entre essas abas ([ver Apêndice A - Estrutura da interface gráfica](#)):

- a) Aba inicial “Home”: É a primeira página que o usuário verá ao abrir o software.
- b) Aba “Update Database”: Esta aba será o local onde o usuário acessará caso queira realizar o processo de atualização da base de dados do relatório. Nela existem dois campos de texto nos quais o usuário entrará com duas informações: a primeira o caminho do local onde ele gostaria de salvar as planilhas vindas do SAP ERP no computador dele, e a segunda o caminho do local onde ele gostaria de salvar a planilha resultante de todo o processo, após a concatenação dos arquivos.
- c) Aba “Combine Excel Files”: Esta é uma aba para o usuário utilizar caso ele queira apenas concatenar as planilhas em Excel, sem necessariamente precisar passar pelo

processo de acessar o SAP ERP e extrair as planilhas (por exemplo, em uma situação em que ele já tenha em mãos as planilhas em Excel e só precisa concatená-las). Nesta aba, o usuário só precisará entrar com duas informações, sendo a primeira o caminho da pasta onde as planilhas estão salvas em sua máquina, e a segunda o caminho do local onde ele gostaria de salvar a planilha resultante do processo, após a concatenação.

d) Aba “About”, contém uma breve descrição do sistema.

O processo de construção da interface gráfica é simples, pois todos os componentes (exemplo: botões, caixas de texto, caixas de entrada de dados etc.) já estão prontos dentro do Qt Designer. Portanto, basta inserir os componentes na área de criação, posicioná-los e dimensioná-los conforme a necessidade.

Na sequência, ao salvar o arquivo no computador, são gerados dois arquivos, um com a extensão “.ui” e outro com a extensão “.qrc”. O primeiro arquivo (“.ui”), é o que contém toda a interface gráfica construída, porém não incluindo as imagens ou ícones utilizados. Já o segundo arquivo (“.qrc”), é o que contém todas as imagens e ícones utilizados na construção da interface gráfica.

Por fim, para fazer com que esses dois arquivos sejam interpretados pelo PyCharm, bem como interajam com os códigos que serão construídos mais adiante, é necessário que ambos os arquivos sejam convertidos para o tipo Python, com a extensão “.py”.

3.2.2.2 Extração das bases de dados

Para o desenvolvimento deste código ([ver Apêndice B - Código que realiza a extração das bases de dados](#)), inicialmente foi criada uma função denominada “*update_database*” e, dentro dela, utilizou-se três módulos do Python, sendo eles o “pyautogui”, “time” e “datetime”, cuja descrição de cada um deles está disponível na [seção 2.4.4](#).

Com esses três módulos, desenvolveu-se um código de programação que fosse capaz de replicar todas as ações humanas necessárias para acessar o software SAP ERP e extrair as bases de dados, tal como um funcionário faria manualmente. Cada um dos módulos foi utilizado da seguinte maneira:

- Pyautogui:

Utilizado para executar todos os processos que envolvem o manuseio do *mouse* e do teclado do computador, como por exemplo, digitar a URL no navegador para acessar o SAP ERP, ou clicar, com o *mouse*, para acessar cada uma das páginas necessárias durante o processo.

- Time:

Utilizado para efetuar as pausas ou tempo de espera durante a execução do código, como por exemplo, o tempo de espera para abrir o navegador ou o tempo de espera para abrir a página do SAP ERP. Além disso, esse módulo também foi utilizado para medir o tempo de execução do código.

- Datetime:

Utilizado para efetuar todas as operações que envolvem valores de data, como por exemplo, o cálculo da data do primeiro dia do mês corrente ou da data do primeiro dia do mês anterior. Esses são dois dos campos que precisam ser preenchidos nos formulários dentro do SAP ERP para a extração das bases de dados.

3.2.2.3 Concatenar as bases de dados extraídas do SAP ERP.

Para desenvolver esta parte da solução ([ver Apêndice C - Código que concatena as bases de dados](#)), foi criada uma outra função denominada “*combine_excel_files*” e, dentro dela, foram utilizados dois módulos do Python, o módulo “os” e o módulo “pandas” ([ver seção 2.4.4](#)).

A função do código desenvolvido nesta seção é concatenar todas as planilhas no formato Excel extraídas do SAP ERP. Para isso, os módulos foram utilizados da seguinte maneira:

- OS:

Utilizado para fazer a manipulação do sistema de arquivos do computador como, por exemplo, trazer uma lista com todos os arquivos presentes em uma determinada pasta ou mesmo concatenar duas ou mais cadeias de caracteres e transformar o resultado dessa operação em um “caminho” para um local do computador.

- Pandas:

Usado para executar três funções, sendo a primeira a de criar uma tabela vazia, a qual posteriormente é preenchida com os dados vindos de todas as planilhas extraídas do SAP ERP. A segunda função é justamente concatenar os dados de todas as planilhas. Por último, o Pandas realiza a transformação da tabela que contém todos os dados das planilhas em um único arquivo Excel (.xlsx).

3.2.2.4 Interligar os códigos das etapas 2 e 3 com a interface gráfica.

Para interligar os códigos das etapas anteriores com a interface gráfica e, finalmente, atribuir uma ação para cada um dos botões da mesma, é necessário criar uma função

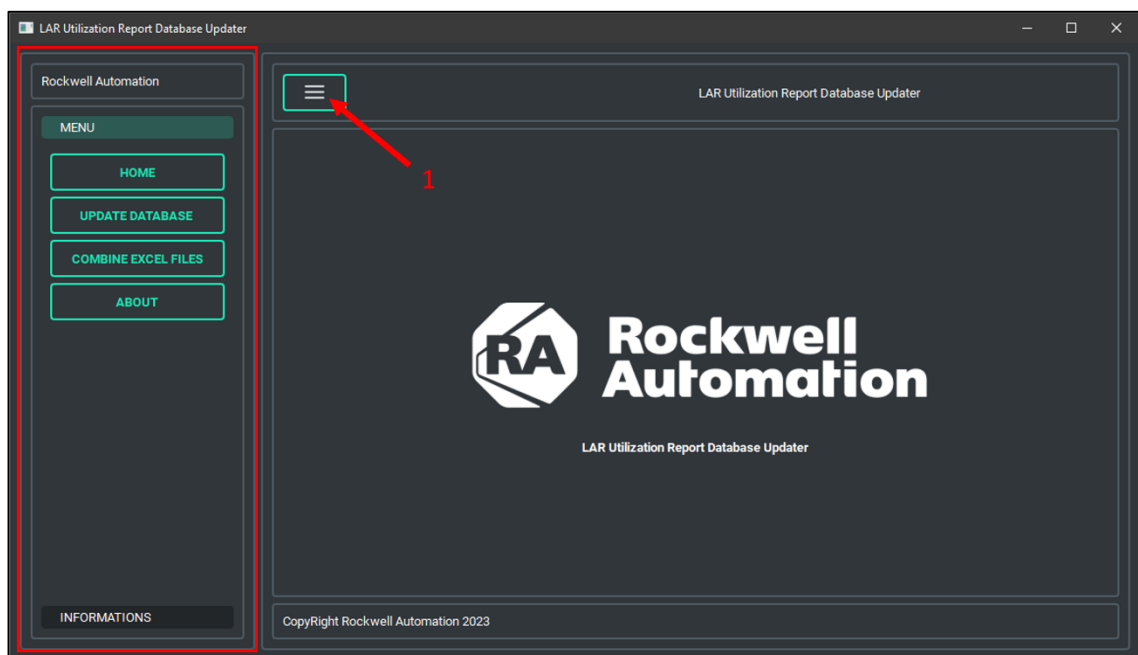
(“*__init__*”) no início do código, a qual será executada assim que o programa é inicializado. Ela tem como objetivo inicializar a interface gráfica, bem como estabelecer a ação de cada um dos botões presentes na mesma ([ver Apêndice D - Código para inicializar a interface gráfica e estabelecer a ação de cada botão](#)).

Feito isso e, considerando que já foram desenvolvidas as principais funções do sistema (“*update_database*” e “*combine_excel_files*”), é necessário criar algumas funções auxiliares que irão interagir com o usuário. Por exemplo, quando o usuário clicar em um botão, esse botão deve acionar uma função que abrirá o explorador de arquivos do computador do usuário, e solicitará que ele selecione uma pasta onde ele quer salvar o arquivo que será gerado pelo sistema.

Vale ressaltar que as funções “*update_database*” e “*combine_excel_files*” são atribuídas aos botões “Update Database” (na aba “Update Database”) e “Combine Files” (na aba “Combine Excel Files”), respectivamente.

A primeira função auxiliar foi chamada de “*left menu*” ([ver Apêndice E - Código da função "left menu"](#)), ela está presente no botão do tipo “sanduíche” na página “Home” da interface gráfica (indicado com o número “1” na Figura 3) tem como objetivo abrir e fechar o menu lateral do programa (onde estarão os botões para navegar pelas outras abas).

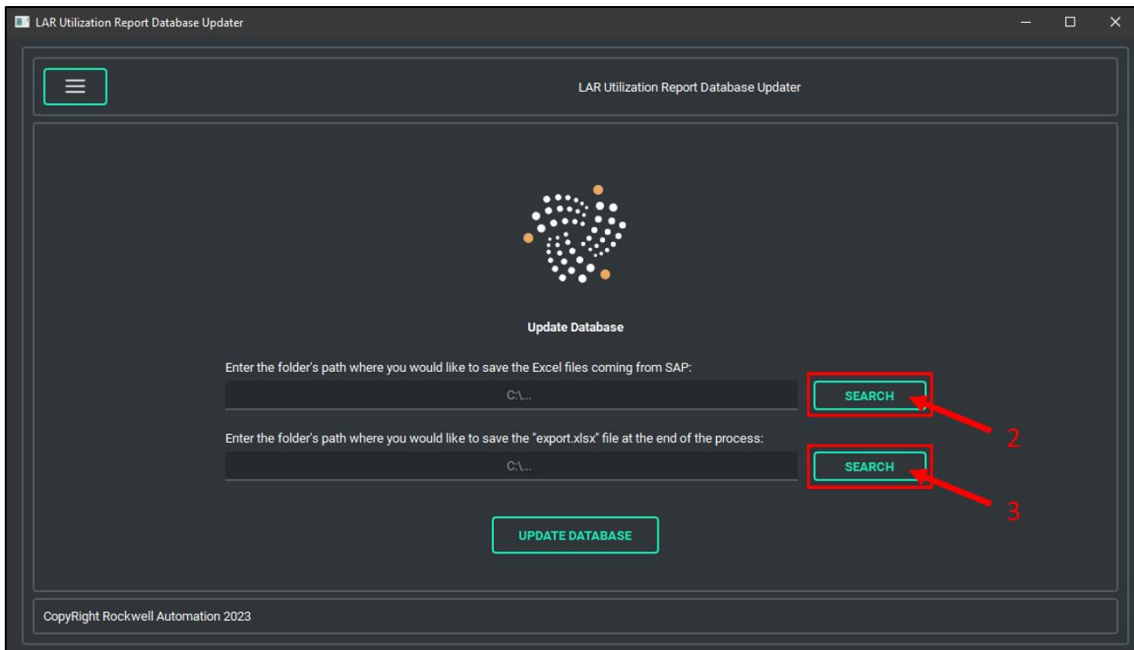
Figura 3 - Menu lateral do sistema e botão para acessar esse menu.



Fonte: Autoria própria.

Já as funções auxiliares da página “Update Database” ([ver Apêndice F - Código das funções auxiliares da página "Update Database"](#)) estão presentes nos botões “Search”, indicados como 2 e 3 na Figura 4.

Figura 4 - Botões na aba “Update Database” para abrir o explorador de arquivos.



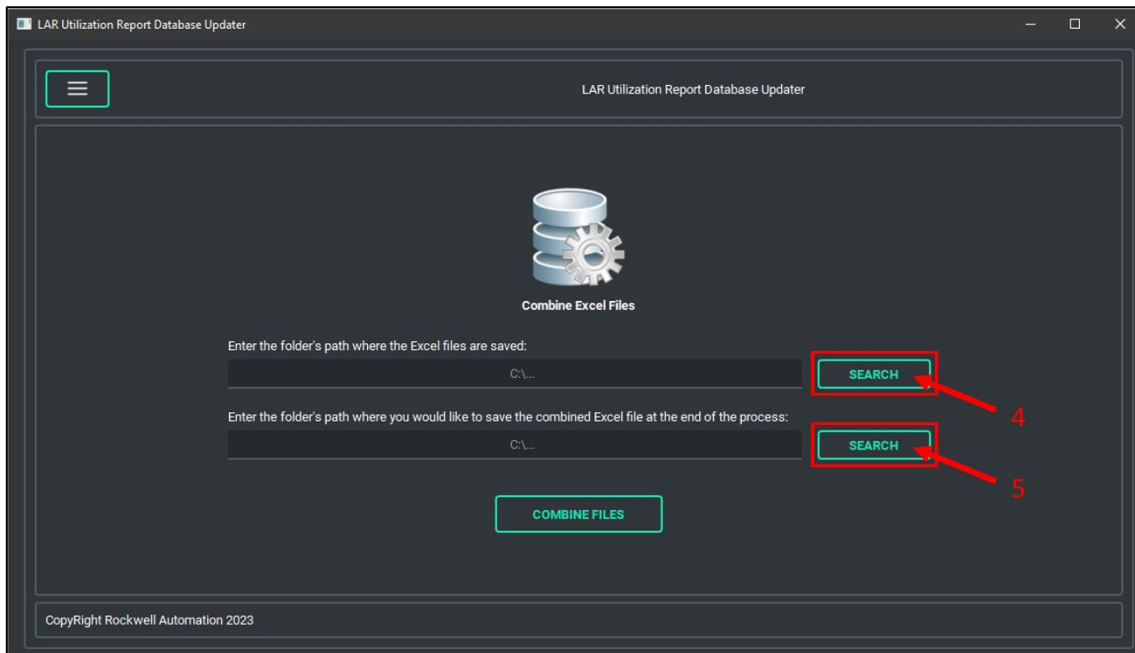
Fonte: Autoria própria.

A função do botão indicado com o número “2” (denominada, no código, como “*search_path_SAVE_SAP_files*”) na Figura 4, tem como finalidade abrir o explorador de arquivos da máquina do usuário, para que este possa indicar o local onde ele gostaria de salvar os arquivos em Excel que serão extraídos do SAP ERP.

Na Figura 4 também se encontra o botão indicado com o número “3” e cuja função denominada como “*search_path_SAVE_export*”, possui a finalidade semelhante ao botão 2, com a diferença de que o usuário deve indicar o local onde ele gostaria de salvar o arquivo final, resultante de todo o processo.

Na página “Combine Excel Files” da interface gráfica (Figura 5), estão presentes as funções auxiliares 4 e 5 ([ver Apêndice G - Código das funções auxiliares da página "Combine Excel Files"](#)).

Figura 5 - Botões na aba “Combine Excel Files” para abrir o explorador de arquivos.



Fonte: Autoria própria.

A função do botão indicado com o número “4” (denominada, no código, como “*search_path_Excel_files*”) na Figura 5, tem como finalidade abrir o explorador de arquivos da máquina do usuário, para que este possa indicar o local onde estão salvos os arquivos em Excel que ele gostaria de concatenar.

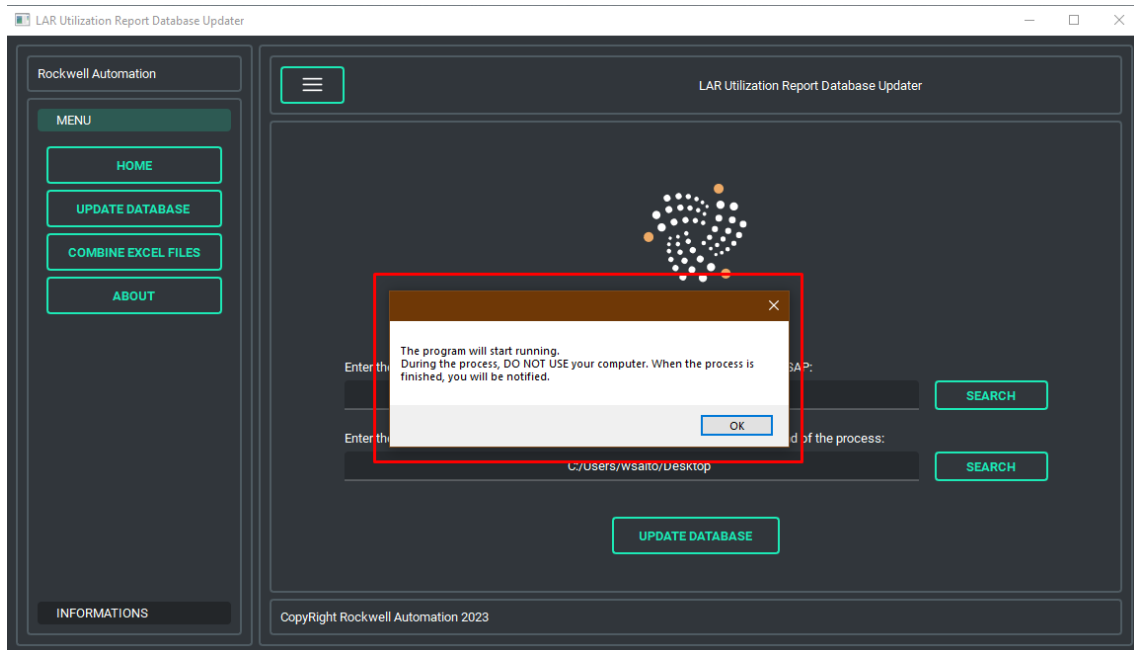
Da mesma forma, a função do botão indicado com o número “5” (denominada, no código, como “*search_path_SAVE_combined_file*”) na Figura 5, possui uma finalidade semelhante ao botão 4, com a diferença de que o usuário deve indicar o local onde ele gostaria de salvar o arquivo em Excel resultante do processo, ou seja, o arquivo que conterá todas as informações concatenadas.

A função auxiliar 6 denominada “msg” ([ver Apêndice H - Código da função "msg"](#)), diferentemente das citadas anteriormente, não é atribuída diretamente como uma ação de um botão. Isso porque a sua função é somente avisar o usuário – por meio de uma caixa de texto – sobre qualquer evento importante para seu conhecimento. Por exemplo, quando o sistema terminar de executar algum processo, é interessante avisar o usuário de que o processo foi executado com sucesso.

Isso também vale para quando ocorrer algum erro durante a execução de algum processo. A Figura 6 mostra um exemplo da função “msg” sendo executada ao clicar no botão

“Update Database”, alertando o usuário de que o programa irá iniciar um processo, e solicitando para que a máquina não seja utilizada durante a sua execução.

Figura 6 - Exemplo de mensagem exibida para o usuário com a função “msg” ao iniciar o processo de atualização da base de dados.



Fonte: Autoria própria.

3.2.2.5 Inclusão de dois blocos de código de programação para o correto funcionamento do programa.

Por fim, para finalizar a parte de programação em linguagem Python, deve-se incluir dois blocos de código no programa (Apêndice I - Código da função "resource_path" e Apêndice J - Código para inicializar todo o sistema).

O bloco de código presente no [Apêndice I - Código da função "resource_path"](#) é uma função chamada “resource_path”, e seu objetivo é impedir que um erro aconteça quando se faz a conversão do código para um arquivo executável. Não é necessária a atribuição desta função a nenhum componente da interface gráfica, pois ela será executada automaticamente ao inicializar o sistema.

Já o bloco de código presente no [Apêndice J - Código para inicializar todo o sistema](#) tem o objetivo de executar o sistema, bem como estabelecer o seu *design* (cores dos botões, fonte, cor da fonte etc.) e exibir a janela do sistema na tela da máquina do usuário.

3.2.2.6 Arquivo executável.

Após escrever todo o código de programação, é necessário fazer a conversão desse código em um único arquivo executável, para que o usuário possa executar o programa em sua máquina. Para isso, basta executar o seguinte comando no terminal do PyCharm:

```
pyinstaller.exe -onefile -windowed -icon=icone.ico main.py
```

O processo levará alguns minutos e, assim que for finalizado, uma pasta chamada “dist” será criada no mesmo diretório onde o código está salvo e, dentro dela, um único arquivo executável do código será gerado. Este é o arquivo que o usuário utilizará para executar o programa e que também pode ser compartilhado com outras pessoas.

3.2.3 Relatório no Microsoft Power BI.

Com a base de dados em mãos e conhecidas as necessidades para a construção do relatório ([seção 3.2.1](#)), desenvolveu-se os painéis de exibição dos dados, utilizando o Microsoft Power BI.

Para fins de organização, dividiu-se a construção do relatório em seis etapas:

1. Definição do tipo e da formatação dos dados de cada uma das colunas presentes na base de dados principal (neste trabalho, intitulada como “export”).
2. Criação de colunas calculadas na base de dados “export”.
3. Criação da tabela intitulada como “Forecast”.
4. Criação da tabela intitulada como “Report Employees”.
5. Estabelecimento dos relacionamentos entre as tabelas (export, Forecast e Report Employees).
6. Criação dos painéis para a exibição de dados.

O primeiro passo para o desenvolvimento de qualquer painel no Microsoft Power BI é a definição do tipo e da formatação de dados para cada uma das colunas presentes na base de dados. Isso pode ser feito por meio da aba “Visualização de tabela” no Microsoft Power BI. Dessa forma, para cada uma das colunas da base de dados principal “export”, definiu-se o tipo e a formatação de dados apropriada.

O segundo passo, ainda na base de dados “export”, foi a criação de algumas colunas calculadas para a obtenção de dados que não existiam nos dados originais e, assim, atender

todas as demandas do negócio. Essas colunas podem ser criadas por meio do Power Query ou por meio de uma fórmula.

Feitos os tratamentos de dados e a criação das colunas calculadas na tabela “export”, foi necessário criar novas tabelas para complementar as suas informações.

A primeira tabela criada foi intitulada como “Forecast”, cujo objetivo era estabelecer a meta de horas a serem trabalhadas por cada engenheiro em cada mês do ano, pois, na tabela “export”, há apenas os registros das horas efetivamente trabalhadas. Para fins de gerenciamento e obtenção de indicadores, é importante ter um parâmetro (meta) de horas trabalhadas em cada mês.

Para criar essa tabela, foram necessárias duas etapas:

- i) Da mesma forma que a tabela “export” é um arquivo em Excel, criou-se um arquivo em Excel com o nome “Metas.xlsx” e, dentro desse arquivo, foi criada a tabela “Forecast” contendo 5 (cinco) colunas, sendo elas: Nome do engenheiro, Time ao qual pertence, País de origem, Mês de referência e Horas a serem trabalhadas.
- ii) Importar o arquivo Excel criado dentro do Power BI e criar algumas colunas calculadas com base nos dados das colunas existentes, a fim de agregar todas as informações necessárias nesta tabela.

Após a conclusão da tabela “Forecast”, foi necessária a criação de uma segunda tabela de dados, a qual foi intitulada como “Report Employees”. Da mesma forma como ocorreu com a tabela “Forecast”, esta foi criada no Microsoft Excel e posteriormente importada para o Microsoft Power BI.

Essa tabela serve apenas para identificar o nome dos engenheiros presentes no relatório, bem como seus respectivos países de origem e o time ao qual pertencem. Além disso, ela contém uma coluna denominada “Status”, cujo objetivo é apenas indicar se um determinado engenheiro já foi inserido no relatório.

Por fim, para que os dados presentes nas tabelas criadas possam “conversar” entre si, é necessário estabelecer o relacionamento entre as tabelas, assim como o tipo desse relacionamento (um para um, um para muitos ou muitos para muitos). Isso pode ser feito na aba “Visualização de Modelo” dentro do Microsoft Power BI.

Estabelecer os relacionamentos faz com que o Microsoft Power BI saiba como os dados de cada tabela estão interligados. Isso é importante para a construção dos painéis de dados, pois existem momentos, por exemplo, em que são colocados – em um mesmo gráfico – dados

provenientes de tabelas distintas e, para que esses dados tenham um sentido lógico, é necessário que haja relacionamentos entre essas tabelas.

4 RESULTADOS E DISCUSSÕES

4.1 Criação dos painéis de exibição de dados.

Após coletar todos os dados e concluir todas as manipulações e associações entre eles, os painéis de visualização dos dados foram criados no Microsoft Power BI.

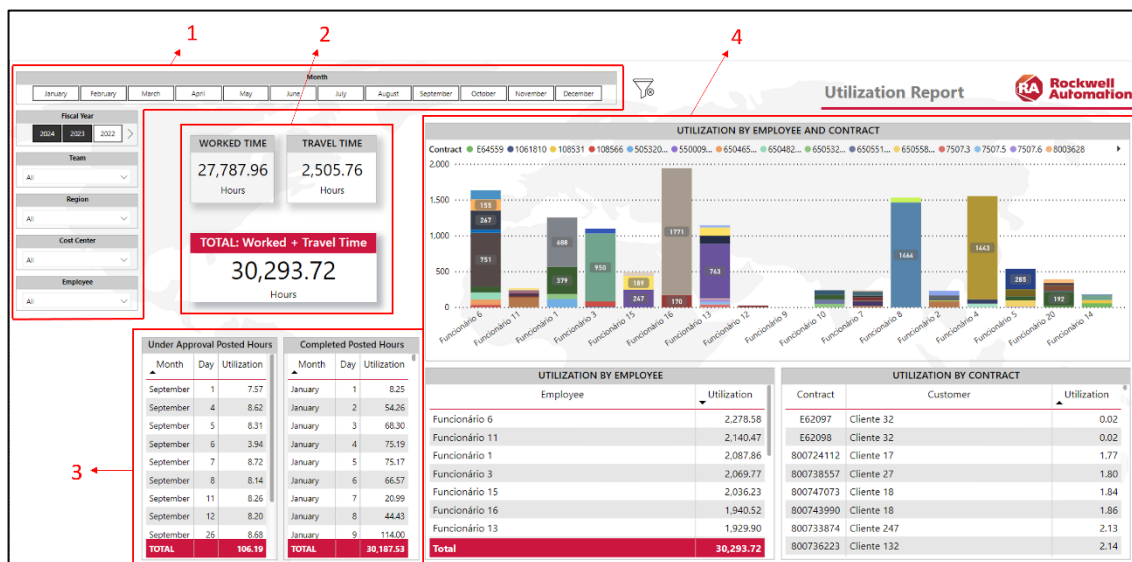
Para isso, foram elaboradas cinco abas, cada uma com um propósito diferente. O conjunto de todas elas recebeu o nome “Utilization Report” (“Relatório de Utilização”) e foram nomeadas como:

1. Overview
2. Utilization Status
3. Actual x Available
4. Hours on hold
5. SAP Fulfillment

4.1.1 Aba “Overview”

A Figura 7 apresenta a imagem da primeira aba da ferramenta, chamada “Overview”.

Figura 7 - Primeira aba da ferramenta: “Overview”.



Fonte: Autoria própria.

O intuito desta aba é fornecer uma visualização geral dos dados relacionados às horas trabalhadas por cada funcionário/engenheiro, em cada projeto.

Na parte superior esquerda do painel – indicado pelo número “1” na Figura 7, foram colocados alguns filtros de dados interativos, para que os usuários pudessem filtrar os dados que são exibidos nos gráficos e tabelas presentes nesta aba.

Dentre os filtros disponíveis, destacam-se:

- Filtro de meses do ano.
- Filtro de anos fiscais.
- Filtro de time.
- Filtro de região.
- Filtro de centro de custo.
- Filtro para o nome do funcionário.

Os três cartões – indicado pelo número “2” na Figura 7, mostram o somatório das horas trabalhadas (cartão “Worked Time”), o somatório das horas de viagem (cartão “Travel Time”) e o somatório total, que são as horas trabalhadas somadas com as horas de viagem (cartão “Total: Worked + Travel Time”).

Na parte inferior esquerda do painel, há duas tabelas: “Under Approval Posted Hours” e “Completed Posted Hours” – indicadas pelo número “3” na Figura 7. O objetivo de ambas as tabelas é apresentar a quantidade e a data de registro das horas trabalhadas. Assim, é possível saber em que momento os engenheiros registraram suas horas trabalhadas.

A diferença entre as duas tabelas é que, enquanto a tabela “Under Approval Posted Hours” apresenta os registros que ainda não foram aprovados pelo sistema, a tabela “Completed Posted Hours” apresenta todos os registros já aprovados e contabilizados. Vale ressaltar que a coluna “Utilization” de ambas as tabelas representa a quantidade de horas trabalhadas.

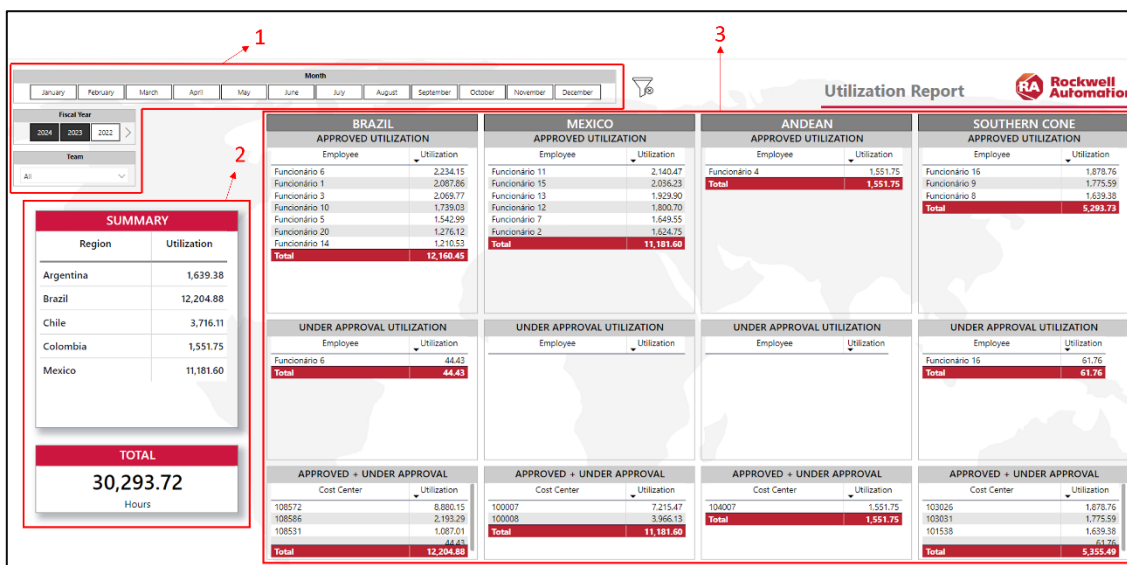
Na região central desta aba – indicado pelo número “4” na Figura 7, tem-se, na parte superior, um gráfico de barras, cujo eixo *x* representa cada um dos engenheiros, o eixo *y* a quantidade de horas trabalhadas e as cores representam os diversos projetos onde os engenheiros trabalharam.

Nessa mesma região, logo abaixo do gráfico, observa-se a presença de duas tabelas. A tabela do lado esquerdo, denominada “Utilization by Employee”, apresenta as horas trabalhadas por cada engenheiro – os valores da coluna “Utilization” são medidos em horas. Já a tabela à direita, “Utilization by Contract”, apresenta a quantidade de horas trabalhadas em cada contrato/cliente.

4.1.2 Aba “Utilization Status”

A Figura 8 ilustra a segunda aba desenvolvida na ferramenta, denominada “Utilization Status”. O objetivo dela é apresentar a quantidade de horas trabalhadas por cada engenheiro, bem como o *status* dos registros de horas trabalhadas.

Figura 8 - Segunda aba: “Utilization Status”



Fonte: Autoria própria.

Assim como na aba “Overview”, tem-se, na parte superior esquerda – indicado pelo número “1” – alguns filtros de dados interativos para auxiliar o usuário durante a análise das informações do relatório.

Logo abaixo dos filtros – indicado pelo número “2” – encontram-se uma tabela e um cartão. A tabela chamada “Summary” apresenta os valores de horas trabalhadas em cada país e o cartão o somatório de todas essas horas.

Na região central desta aba – indicado pelo número “3” – observa-se quatro colunas, cada uma representando uma região (Brasil, México, Região Andina e Cone Sul) e sobrepostas a elas, aparecem ainda três tabelas.

A primeira tabela, intitulada “Approved Utilization”, apresenta os registros de horas trabalhadas que já estão aprovadas e contabilizadas no sistema.

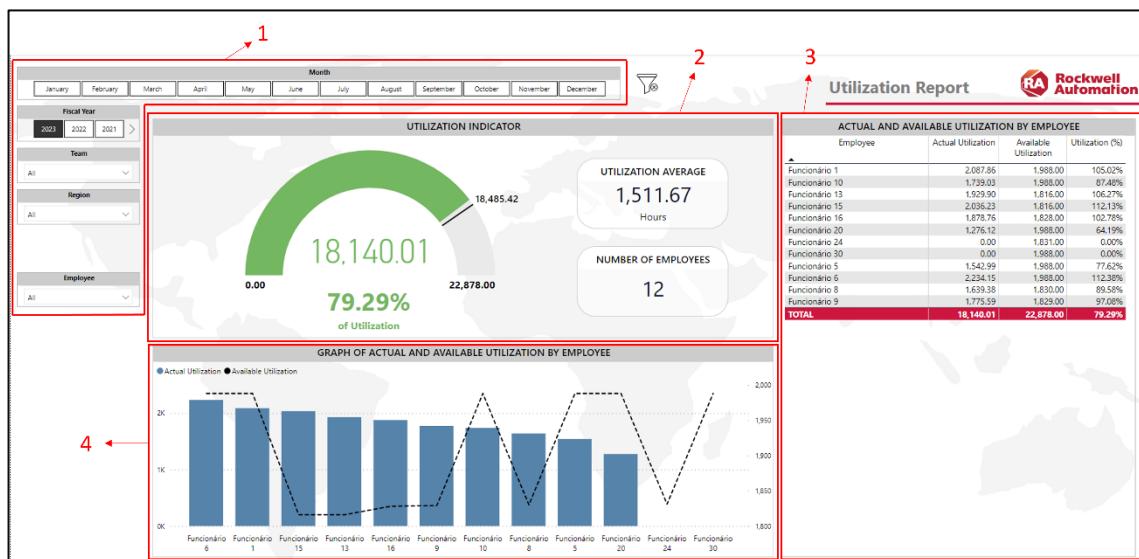
A segunda tabela, “Under Approval Utilization”, por sua vez, exibe os registros de horas trabalhadas que ainda não foram aprovadas pelo sistema.

E por último, tem-se a tabela “Approved + Under Approval”, a qual mostra o somatório das horas já aprovadas e as que estão sob análise, classificadas por centro de custo (*cost center*).

4.1.3 Aba “Actual x Available”

A Figura 9 apresenta a terceira aba do relatório, denominada “Actual x Available”, cujo objetivo é analisar a *performance* de cada engenheiro, comparando as horas efetivamente trabalhadas e já registradas no sistema, com a meta de horas a serem trabalhadas durante o período de análise, o qual varia de acordo com o filtro aplicado.

Figura 9 - Terceira aba do relatório: “Actual x Available”



Fonte: Autoria própria.

Assim como nas abas anteriormente apresentadas, na parte superior esquerda tem-se – indicado com o número “1” – alguns filtros de dados para auxiliar o usuário durante a análise das informações (filtros de mês, ano fiscal, time, região e nome do funcionário).

Na região indicada com o número “2” existem três indicadores de dados: um do tipo velocímetro e dois cartões. O indicador do tipo velocímetro apresenta a quantidade de horas trabalhadas tanto no formato percentual quanto em número. Além disso, ele também mostra – por meio de uma linha na cor preta – o valor da meta de horas trabalhadas, para auxiliar a análise. Um pouco mais a direita, tem-se um cartão intitulado “Utilization Average”, que apresenta a média de horas trabalhadas por cada engenheiro dentro do período selecionado. E logo abaixo deste último cartão, tem-se o cartão “Number of Employees”, o qual mostra o número de engenheiros que estão sendo considerados no momento da análise. Vale lembrar que todos os valores variam conforme a aplicação dos filtros de dados.

Já na região indicada pelo número “3”, tem-se uma tabela com quatro colunas. A primeira chamada de “Employee”, refere-se ao nome do engenheiro. A segunda - “Actual

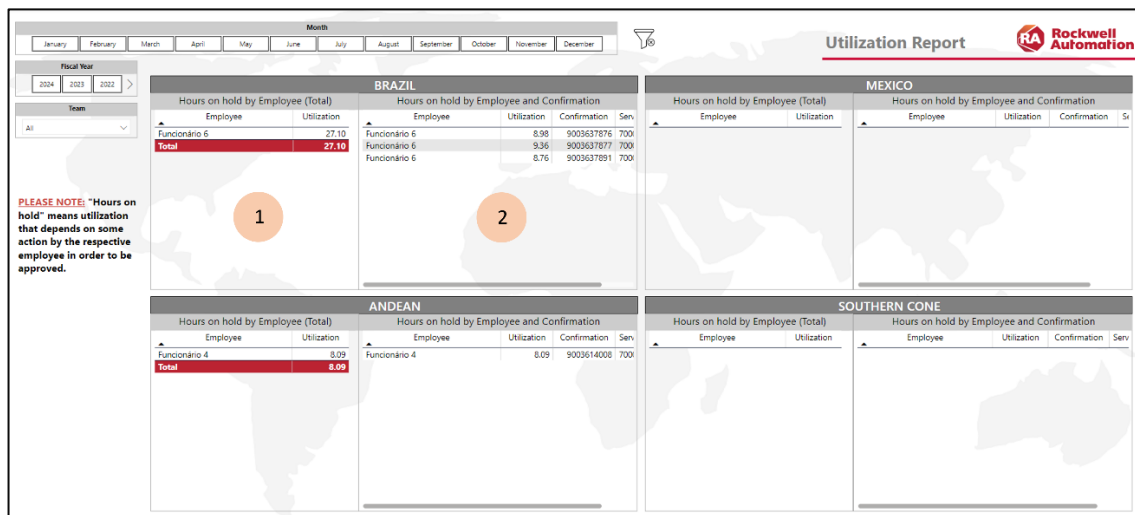
Utilization”, representa a quantidade de horas já trabalhadas e registradas pelo respectivo engenheiro. Já a terceira - “Available Utilization”, mostra a meta de horas a serem trabalhadas pelo respectivo engenheiro dentro do período analisado. Por fim, na última coluna, denominada de “Utilization (%)”, tem-se o percentual de horas trabalhadas em relação a meta.

Por último, na região indicada com o número “4”, tem-se um gráfico de barras verticais onde o eixo **x** representa os engenheiros e o eixo **y** do lado direito, a quantidade de horas trabalhadas (representado pelas barras na cor azul). Também no eixo **y**, mas agora ao lado esquerdo, tem-se a meta de horas a serem trabalhadas dentro do período considerado, nesse caso, representada pela linha tracejada na cor preta.

4.1.4 Aba “Hours on Hold”

A Figura 10 apresenta a quarta aba do relatório, intitulada “Hours on Hold”, cujo objetivo principal é apresentar todos os registros de horas trabalhadas que tiveram algum problema durante o processo de aprovação e foram reprovados. Os registros que aparecem nesta aba são revisados pelos respectivos engenheiros e depois reenviados para o processo de aprovação.

Figura 10 - Quarta aba do relatório: “Hours on Hold”



Fonte: Autoria própria.

Uma das primeiras coisas que são observadas na Figura 10 é que os mesmos quadros indicados pelos números 1 e 2 são repetidos quatro vezes, uma vez para cada região (Brasil, México, Região Andina e Região do Cone Sul).

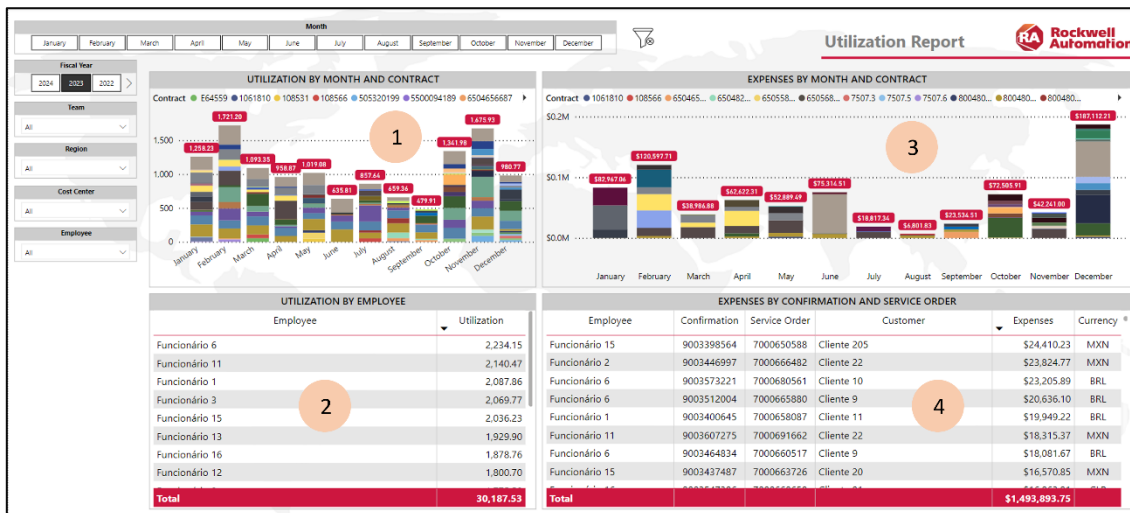
O quadro indicado pelo número “1”, denominado “Hours on Hold by Employee (Total)”, apresenta o total de horas reprovadas por cada engenheiro.

Já o segundo quadro, indicado pelo número “2” e intitulado como “Hours on hold by Employee and Confirmation”, apresenta, em detalhes, os registros de horas trabalhadas que precisam ser revisados. Dessa forma, por meio da coluna “Confirmation” – que é o código do registro – o engenheiro consegue saber exatamente quais são os registros que estão com problema.

4.1.5 Aba “SAP Fulfillment”

A Figura 11 apresenta a quinta e última aba do relatório intitulada “SAP Fulfillment”. O objetivo desta aba é apresentar dados referentes às horas trabalhadas, bem como às despesas feitas por cada engenheiro e em cada projeto.

Figura 11 - Quinta aba do relatório: “SAP Fulfillment”



Fonte: Autoria própria.

O gráfico indicado pelo número “1” é um gráfico de barras verticais, onde o eixo x representa os meses do ano, o eixo y a quantidade de horas trabalhadas, e cada cor nas barras representa um projeto/contrato.

A tabela indicada pelo número “2” apresenta as horas trabalhadas por cada engenheiro onde, na coluna “Employee”, tem-se o nome do engenheiro e na coluna “Utilization”, a quantidade de horas trabalhadas por ele.

O gráfico de barras verticais – indicado pelo número “3” – apresenta as informações referentes às despesas em cada mês e em cada projeto. O eixo x representa os meses do ano e o eixo y os valores das despesas. Cada cor presente nas barras verticais representa um projeto/contrato distinto. Assim, é possível obter, de uma forma visualmente mais intuitiva, o tamanho das despesas feitas por projeto e por mês.

Por fim, a tabela indicada pelo número “4” apresenta, em detalhes, as despesas feitas por cada engenheiro, mostrando o seu nome (coluna “Employee”), o código do registro (coluna “Confirmation”), o código da ordem de serviço (coluna “Service Order”), o nome do cliente (coluna “Customer”), o valor da despesa (coluna “Expenses”) e a moeda utilizada na despesa (coluna “Currency”).

Por meio desta aba, é possível obter uma visão e uma relação entre os valores de horas trabalhadas e as despesas feitas por cada engenheiro em cada projeto.

4.2 Compartilhamento da solução RPA e publicação do relatório desenvolvido no Microsoft Power BI.

Após a conclusão de ambas as partes deste projeto, o software com a solução RPA foi disponibilizado em uma pasta compartilhada pelo “time”, dentro do Microsoft SharePoint. Já o relatório desenvolvido no Microsoft Power BI foi publicado em um *Workspace* próprio da empresa – onde existem outros relatórios desenvolvidos com o mesmo software – e compartilhado com todas as partes interessadas.

4.3 Implementação do projeto e treinamento das partes interessadas

Logo após o compartilhamento do projeto com todas as partes interessadas, foram realizadas três reuniões com toda a equipe do departamento para a sua implementação. A primeira reunião tinha como objetivo a contextualização do “time”, realizando uma apresentação geral do projeto, desde a problemática inicial até o desenvolvimento da solução. O segundo encontro teve o intuito de apresentar, na prática, o funcionamento do projeto e como utilizá-lo da maneira correta. Por fim, após um período de aproximadamente duas semanas desde o início do uso da solução, ocorreu a terceira reunião, com o objetivo de sanar as dúvidas que apareceram durante esse período.

Depois de algum tempo desde a implementação do projeto, outros dois departamentos da empresa – um da América do Norte e outro da Europa – se interessaram pela solução e demonstraram interesse em implantá-la em seus “times”. Assim, de forma semelhante como ocorreu com a equipe local, foram realizadas duas reuniões com cada um dos “times” interessados para a implementação da solução e treinamento sobre o correto uso da mesma.

4.4 Ganhos com a otimização do processo de extração da base de dados do SAP ERP

Com a implementação da solução RPA no processo de extração da base de dados do SAP ERP, houve uma grande economia de tempo e de trabalho para o funcionário responsável por esta atividade.

Diariamente, eram gastos cerca de 30 minutos para executar esse processo de forma totalmente manual. Com a utilização da solução RPA, o funcionário passou a não usar mais o seu tempo com esta atividade, pois o robô possui a capacidade de executar todo o processo de forma autônoma. Normalmente, o funcionário coloca o robô para funcionar em horários em que ele não está utilizando o computador como, por exemplo, no horário de almoço, dessa forma, ao retornar, o robô já terá executado todo o processo. Anualmente, são economizadas aproximadamente 120 horas, apenas com esta automação de processo.

Além disso, um outro ponto importante a ser ressaltado é a redução do tempo de execução do processo pelo robô com relação ao funcionário. Enquanto o funcionário levava cerca de 30 minutos para executar o processo, o robô leva em torno de 20 minutos, o que significa uma redução de 33% no tempo de execução.

4.5 Ganhos com a otimização do processo de geração do relatório de horas trabalhadas e despesas.

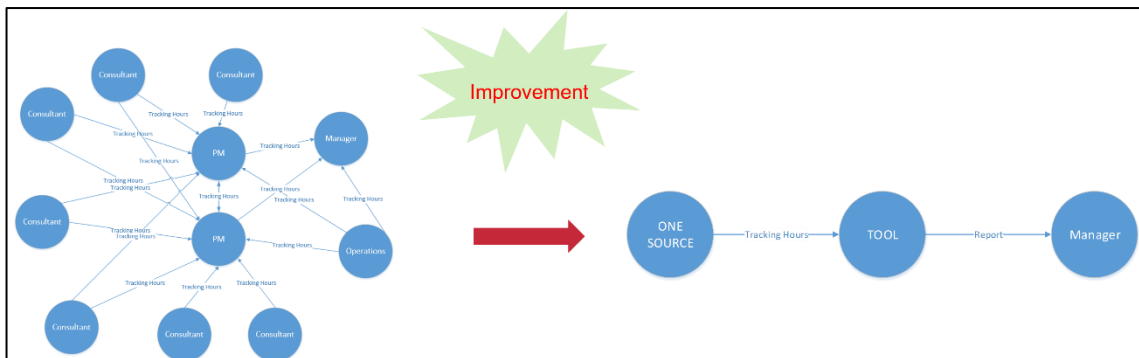
Além dos benefícios descritos na [seção 4.4](#), constatou-se uma segunda melhoria significativa relacionada ao tempo gasto no processo de geração do relatório de horas trabalhadas e despesas.

Anteriormente à implementação do relatório desenvolvido no Microsoft Power BI, o processo de geração de um relatório era bastante complexo, manual, grande e envolvia muitas pessoas. Com a implementação da solução, esse processo se tornou simples, automático, muito menor, e com praticamente nenhum funcionário envolvido, exceto aquele responsável por executar o robô para extrair os dados do SAP ERP.

Uma representação de como era o processo anterior à implantação do relatório feito no Microsoft Power BI – descrito na [seção 3.2.1](#), pode ser visto no lado esquerdo da Figura 12.

Já no lado direito da Figura 12, encontra-se a representação do processo após a implementação da solução, onde existe apenas uma única fonte de dados, a qual tem seus dados tratados de forma automática pelo Microsoft Power BI, gerando um relatório mais intuitivo e visualmente agradável para o gerente responsável ou para outros colaboradores.

Figura 12 - Representação do processo de geração de um relatório antes e depois da solução desenvolvida no Microsoft Power BI.



Fonte: Rockwell Automation (2023).

Considerando todos os engenheiros impactados nas regiões onde a solução foi aplicada, foi feito um levantamento sobre o tempo economizado no processo com a implementação do relatório feito no Microsoft Power BI (Tabela 1). Constatou-se uma economia de aproximadamente 138 horas mensais, o que significa 1.656 horas por ano.

Tabela 1 - Tempo economizado pelos engenheiros com a implementação da aplicação.

Região dos engenheiros impactados	Número de engenheiros	Horas gastas por mês	Total de horas gastas por mês
América do Norte	29	2	58
Europa	19	2	38
América Latina	21	2	42
TOTAL			138

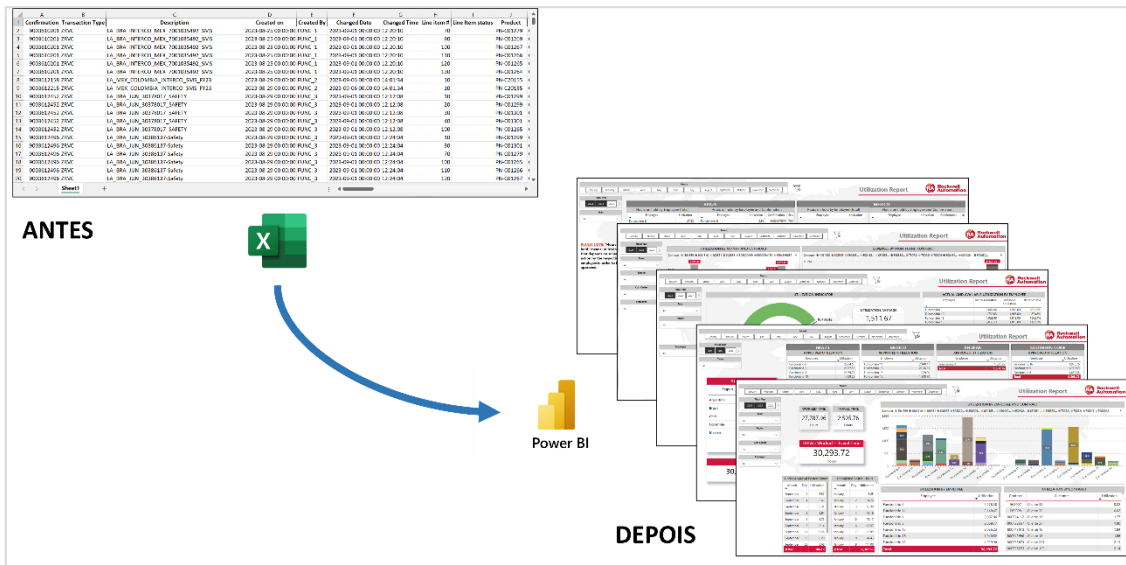
Fonte: Rockwell Automation (2023).

4.6 Melhoria visual do relatório

Por fim, é importante citar a melhoria do relatório em termos visuais. Anteriormente, o relatório era entregue em uma planilha do Microsoft Excel, sem nenhum tipo de visualização gráfica ou tratamento visual dos dados.

Com a utilização do Microsoft Power BI, foi possível criar um relatório que apresenta visualizações mais intuitivas e estéticas dos dados, além de não mostrar informações irrelevantes como ocorria no relatório anterior. Isso facilitou a análise das informações, a qual foi impulsionada com a inclusão de filtros interativos e intuitivos no relatório. A Figura 13 apresenta os relatórios antes e depois da melhoria.

Figura 13 - Relatório de horas trabalhadas e despesas antes e depois da melhoria.



Fonte: Autoria própria.

5 CONCLUSÃO

O desenvolvimento dessa aplicação mostrou que a linguagem de programação Python permite automatizar processos manuais executados no computador, muitas vezes envolvendo diversas plataformas ou programas não correlacionados. O sistema implementado em Python foi uma solução eficaz para a obtenção de dados com um mínimo de intervenção manual. Isso foi necessário devido à falta de uma conexão direta entre o Microsoft Power BI e o banco de dados da empresa.

Além disso, o uso de softwares de Business Intelligence como o Microsoft Power BI para o tratamento e visualização de dados também comprovou que esses softwares, além de terem a capacidade de processar maiores volumes de dados e de forma mais rápida, eles possuem diversos tipos de visualização de dados prontos para uso, o que facilita a construção de painéis. Consequentemente, auxiliam na obtenção de *insights*, aumentando a agilidade para tomadas de decisão dentro da empresa.

Outra vantagem do uso do Microsoft Power BI, está relacionado à apresentação dos dados. Anteriormente, o relatório era entregue em uma planilha do Excel sem gráficos ou tratamento visual dos dados. Agora, o relatório apresenta visualizações mais intuitivas e atraentes, eliminando informações irrelevantes, facilitando a análise e compreensão das informações.

Antes da solução RPA, diariamente eram gastos, pelo funcionário, aproximadamente 30 minutos nesse processo, e agora, o robô é capaz de executar o processo de forma autônoma, liberando o funcionário desta tarefa, e gerando uma economia anual de cerca de 120 horas. Com isso, houve uma redução de 33% no tempo de execução do processo em relação ao tempo de execução de uma pessoa.

Conclui-se que o relatório desenvolvido no Microsoft Power BI trouxe uma drástica simplificação e automação ao processo de geração de relatórios, anteriormente complexo e manual. O impacto abrangeu vários engenheiros em diferentes regiões, resultando em uma economia de aproximadamente 138 horas por mês, equivalente a 1.656 horas por ano. Com isso, houve uma economia significativa tanto financeira como de tempo, o que permitiu, aos colaboradores envolvidos nos processos, a possibilidade de alocar seus esforços em outras atividades que agregassem mais valor para o negócio.

Trabalhos Futuros

Para futuros trabalhos, é recomendável incorporar testes unitários e de integração ao sistema desenvolvido em Python. Os testes unitários possibilitam isolar e avaliar cada unidade

de código de maneira independente, assegurando que cada uma execute suas tarefas conforme o esperado. Por outro lado, os testes de integração ajudam a identificar problemas que podem surgir quando módulos individuais da solução são combinados, garantindo uma integração eficaz entre eles e certificando-se de que o sistema funcione conforme planejado. Isso reduz o risco de falhas decorrentes de problemas de comunicação ou integração entre diferentes partes do software.

Além disso, uma ideia valiosa seria redesenhar a solução para torná-la aplicável em diversas situações, como em outros departamentos da empresa. Isso envolve uma adaptação automática a diferentes estruturas de banco de dados e locais de extração de dados, mantendo as mesmas funcionalidades da solução inicial.

REFERÊNCIAS

- AHMAD, Sumera; MISKON, Suraya; ALABDAN, Rana; TLILI, Iskander. Towards Sustainable Textile and Apparel Industry: Exploring the Role of Business Intelligence Systems in the Era of Industry 4.0. **Sustainability**, mdpi, v. 12, ed. 7, p. 2632, 26 mar. 2020. DOI: <https://doi.org/10.3390/su12072632>. Available from: <https://www.mdpi.com/2071-1050/12/7/2632>. Access in: 4 January 2023.
- ALMEIDA, Marcus. **Pandas Python: o que é, para que serve e como instalar: O que é Pandas?**. [S. l.]: Alura, 2023. Disponível em: <https://www.alura.com.br/artigos/pandas-oque-e-para-que-serve-como-instalar>. Acesso em: 18 out. 2023.
- ALURA. **Introdução ao PyCharm**. [S. l.], 2023. Disponível em: <https://www.alura.com.br/apostila-python-orientacao-a-objetos/introducao-ao-pycharm>. Acesso em: 18 out. 2023.
- AWARI. **Aprenda a Programar com Pyside6: Tudo sobre Python e Desenvolvimento de Aplicações**. [S. l.], 2023. Disponível em: https://awari.com.br/aprenda-a-programar-com-pyside6-tudo-sobre-python-e-desenvolvimento-de-aplicacoes/?utm_source=blog&utm_campaign=projeto+blog&utm_medium=Aprenda%20a%20Programar%20com%20Pyside6:%20Tudo%20sobre%20Python%20e%20Desenvolvimento%20de%20Aplica%C3%A7%C3%B5es. Acesso em: 18 out. 2023.
- BAHIENSE, Felipe. **O que é o Robotic Process Automation (RPA) e como os processos de negócios podem ganhar com a tecnologia?** Rio de Janeiro: Neomind, 2018. Disponível em: <http://www.neomind.com.br:81/blog/robotic-process-automation-rpa/>. Acesso em: 13 jan. 2023.
- BARBOSA FILHO, Luiz Henrique. **Lidando com datas no Python: As bibliotecas time e datetime**. Rio de Janeiro: Análise Macro, 2022. Disponível em: <https://analisemacro.com.br/datascience/python/lidando-com-datas-no-python-as-bibliotecas-time-e-datetime/>. Acesso em: 18 out. 2023.
- CONEXOS. **Falta de automação: Descubra como isso pode afetar seu negócio em 6 aspectos**. [S. l.], 2017. Disponível em: <https://www.conexos.com.br/5-prejuizos-que-a-falta-de-automacao-gera-para-a-sua-empresa/#:~:text=Um%20dos%20principais%20preju%C3%AAdzos%20percebidos,gasto%20para%20a%20conclus%C3%A3o%20delas>. Acesso em: 15 abr. 2023.
- COUTINHO, Thiago. **Conheça quais são os benefícios do Power BI para as empresas!**. [S. l.], 2021. Disponível em: <https://www.voitto.com.br/blog/artigo/53epositor53-do-power-bi-para-as-empresas>. Acesso em: 16 jan. 2023.
- DIALANI, Priya. **4 Robotic Process Automation Trends for 2020**. [S. l.], 13 dez. 2019. Disponível em: <https://www.analyticsinsight.net/4-robotic-process-automation-trends-for-2020/>. Acesso em: 4 jan. 2023.
- DJERDJOURI, Mohamed. Data and Business Intelligence Systems for Competitive Advantage: prospects, challenges, and real-world applications. **Mercados y Negocios**,

redalyc, v. 41, p. 5-18, 2020. Disponível em:
<https://www.redalyc.org/articulo.oa?id=571861494009>. Acesso em: 4 jan. 2023.

FERREIRA, Kellison. **Conheça as 12 melhores ferramentas de Business Intelligence para implementar em seu negócio.** [S. l.], 2020. Disponível em:
<https://rockcontent.com/br/blog/ferramentas-de-business-intelligence/>. Acesso em: 21 fev. 2023.

FIA BUSINESS SCHOOL. **BI: o que é, importância, ferramentas e dicas de implementação.** [S. l.], 2021. Disponível em: <https://fia.com.br/blog/bi/>. Acesso em: 15 abr. 2023.

FIA BUSINESS SCHOOL. **Power BI: o que é, funcionalidades e benefícios da ferramenta.** [S. l.], 2020. Disponível em: <https://fia.com.br/blog/power-bi/>. Acesso em: 21 fev. 2023.

FUNG, Han Ping. Criteria, Use Cases and Effects of Information Technology Process Automation (ITPA). **Advances in Robotics & Automation**, v. 3, 2014. Available from: <https://ssrn.com/abstract=2588999>. Access in: 16 January 2023.

GEYER-KLINGEBERG, Jerome; NAKLADAL, Janina; BALDAUF, Fabian; VEIT, Fabian. Process Mining and Robotic Process Automation: A Perfect Match. **International Conference on Business Process Management**, [s. l.], 2018. Available from: <https://www.semanticscholar.org/paper/Process-Mining-and-Robotic-Process-Automation%3A-A-Geyer%E2%80%90Klingeberg-Nakladal/8dc732b3e8c8c5d454a3ca528423a6217847360a>. Access in: 4 January 2023.

KROTOV, Vlad. Predicting the future of disruptive 54epositor5454s: The method of alternative histories. **Business Horizons**, [s. l.], v. 62, ed. 6, p. 695-705, 2019. DOI: <https://doi.org/10.1016/j.bushor.2019.07.003>. Available from: <https://sci-hub.se/10.1016/j.bushor.2019.07.003>. Access in: 4 January 2023.

LATTINE GROUP. **O que é Microsoft SharePoint e quais são seus benefícios?** [S. l.], 2023. Disponível em: <https://lattinegroup.com/compartilhamento/o-que-e-microsoft-sharepoint/>. Acesso em: 19 out. 2023.

MICROSOFT. **O que é Business Intelligence?** [S. l.], 2023a. Disponível em: <https://powerbi.microsoft.com/pt-br/what-is-business-intelligence/>. Acesso em: 16 jan. 2023.

MICROSOFT. **Workspaces no Power BI.** [S. l.], 2023b. Disponível em: <https://learn.microsoft.com/pt-br/power-bi/collaborate-share/service-new-workspaces>. Acesso em: 19 out. 2023.

MÓDULO Python Sys. [S. l.]: Acervo Lima, [ca. 2022]. Disponível em: <https://acervolima.com/modulo-python-sys/>. Acesso em: 18 out. 2023.

NEGRO, Andrés R.; MESIA, Ron. The Business Intelligence and Its Influence on Decision Making. **Journal of Applied Business & Economics**, [S. l.], v. 22, ed. 2, 2020. Available from: http://www.digitalcommons.www.na-businesspress.com/JABE/JABE22-2/12_Negro_22_2_.pdf. Access in: 13 January 2023.

OLIVEIRA, Wallace. **Entenda o que é RPA (Robotic Process Automation) e como**

implantar em sua empresa. [S. l.], 2018. Disponível em: <https://www.heflo.com/pt-br/55positor-processos/rpa-robotic-process-automation/>. Acesso em: 13 jan. 2023

OTIMIZE SEU NEGÓCIO. **5 Problemas Que Você Evita Automatizando Sua Empresa.** [S. l.], 2018. Disponível em: <https://otimizeunegocio.com/55epos-financeira/5-problemas-que-voce-evita-automatizando/>. Acesso em: 15 abr. 2023.

PALMIERI, Fernanda. **10 benefícios do Power BI para a sua empresa.** [S. l.], 2021. Disponível em: <https://blog.simbiox.com.br/10-beneficios-do-power-bi-para-a-sua-empresa>. Acesso em: 16 jan. 2023.

PARIZ, Lorena. **Benefícios do Power BI na eficiência das tomadas de decisões da sua empresa.** [S. l.], 2021. Disponível em: <https://www.tiqs.com.br/55positor55-do-power-bi/>. Acesso em: 16 jan. 2023.

PATEL, Neil. **Business Intelligence: O Que É, Importância, Benefícios e Como Adotar.** [S. l.], 2023. Disponível em: <https://neilpatel.com/br/blog/business-intelligence/>. Acesso em: 16 jan. 2023.

PIRES, E. A. N.; GIRARD, C. D. T.; GIRARD, C. M. T.; SANTANA, S. R. Proposta do rpa aplicado ao contexto da biblioteconomia. **Biblionline**, v. 16, n. 1, p. 18-29, 2020. DOI: 10.22478/ufpb.1809-4775.2020v16n1.54681. Disponível em: <https://periodicos.ufpb.br/ojs2/index.php/biblio/article/view/54681>. Acesso em: 13 jan. 2023.

PYTHON SOFTWARE FOUNDATION. **What is Python? Executive Summary.** [S. l.], 2023a. Available from: <https://www.python.org/doc/essays/blurb/>. Access in: 16 August 2023.

PYTHON SOFTWARE FOUNDATION. **PyAutoGUI.** [S. l.], 2023b. Disponível em: <https://pypi.org/project/PyAutoGUI/>. Acesso em: 18 out. 2023.

PYTHON SOFTWARE FOUNDATION. **Pyperclip.** [S. l.], 2023c. Disponível em: <https://pypi.org/project/pyperclip/>. Acesso em: 18 out. 2023.

QUINN, Martin; STRAUSS, Erik. **The Routledge Companion to Accounting Information Systems.** [S. l.], 2018. Available from: <http://repository.gctu.edu.gh/items/show/765>. Access in: 13 January 2023.

RAMOS, Vinícius. **A biblioteca OS do Python.** [S. l.], 2023. Disponível em: <https://pythonacademy.com.br/blog/a-biblioteca-os-do-python>. Acesso em: 18 out. 2023.

RESTUM, Guilherme. **O que é o Power Query e por que é importante no Power BI?** [S. l.], 2023. Disponível em: https://www.hashtagtreinamentos.com/o-que-e-power-query-power-bi?gad=1&gclid=CjwKCAjwp8OpBhAFEiwAG7NaEmMt2JfwIbx2oNR9USjHUWQwxTB1ac55NUSFgwhFTcq-CWsvsejdihoC5WcQAvD_BwE. Acesso em: 19 out. 2023.

ROMERO, Carlos Andrés Tavera; ORTIZ, Jesús Hamilton; KHALAF, Osamah Ibrahim; PRADO, Andrea Ríos. Business Intelligence: Business Evolution after Industry 4.0. **Sustainability**, [s. l.], v. 13, n. 10026, ed. 18, 2021. DOI:

<https://doi.org/10.3390/su131810026>. Available from: <https://www.mdpi.com/2071-1050/13/18/10026>. Access in: 16 January 2023.

SIBALIJA, Tatjana; JOVANOVIĆ, Stefan; ĐURIĆ, Jelena S. Robotic Process Automation: Overview and Opportunities. **International Journal Advanced Quality**, [S. l.], v. 46, p. 33-39, 2019. Available from: https://www.researchgate.net/publication/332970286_ROBOTIC_PROCESS_AUTOMATION_OVERVIEW_AND OPPORTUNITIES. Access in: 16 January 2023.

SIDERSKA, Julia. Robotic Process Automation: a driver of digital transformation? **Engineering Management in Production and Services**, Sciendo, v. 12, ed. 2, p. 21-31, 2020. DOI: <https://doi.org/10.2478/emj-2020-0009>. Available from: <https://sciendo.com/article/10.2478/emj-2020-0009>. Access in: 4 January 2023.

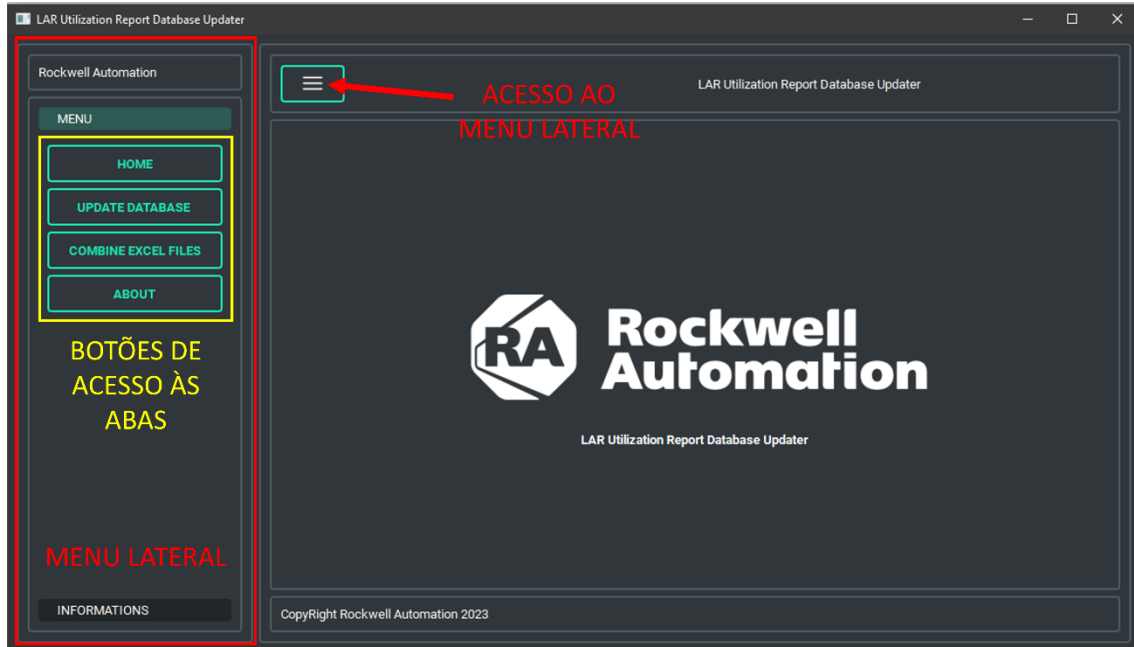
SILVA, Douglas da. **O que é Business Intelligence, para que serve + exemplos**. [S. l.], 11 fev. 2021. Disponível em: <https://www.zendesk.com.br/blog/o-que-e-business-intelligence-para-que-serve/>. Acesso em: 16 jan. 2023.

TABLEAU. **O que é business intelligence?** Seu guia sobre o BI e por que ele é importante. [S. l.], 2023. Disponível em: <https://www.tableau.com/pt-br/learn/articles/business-intelligence>. Acesso em: 16 jan. 2023.

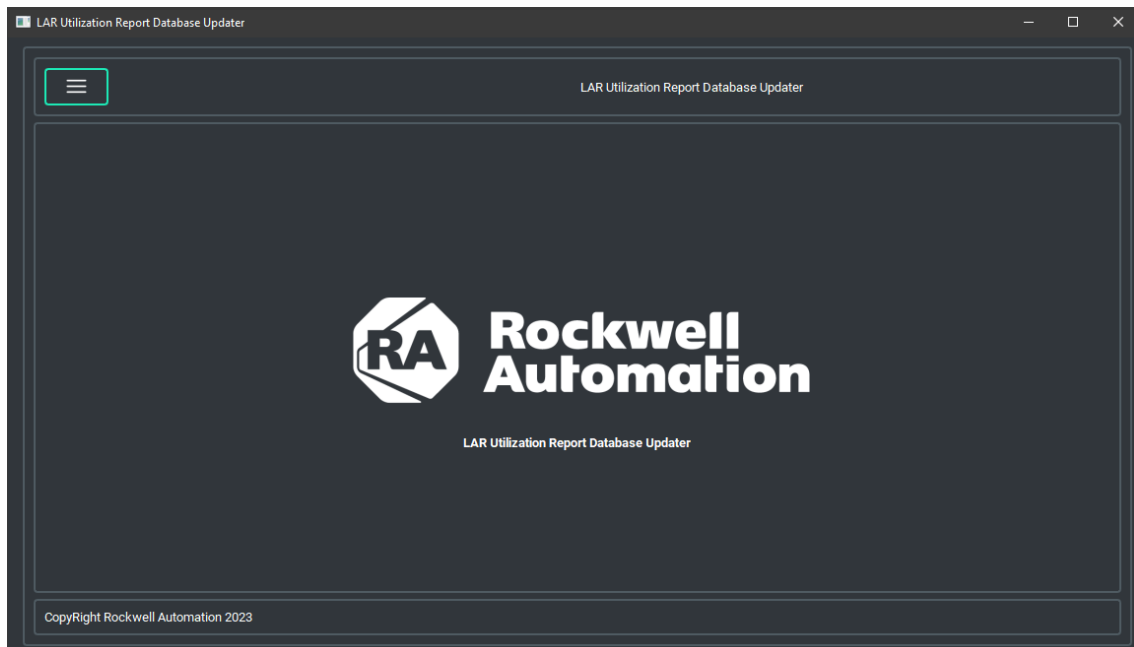
THE QT COMPANY. **Qt Designer Manual, Qt Documentation**. Qt 6.6. [S. l.], 2023. Available from: <https://doc.qt.io/qt-6/qt designer-manual.html>. Access in: 18 October 2023.

APÊNDICE A - Estrutura da interface gráfica

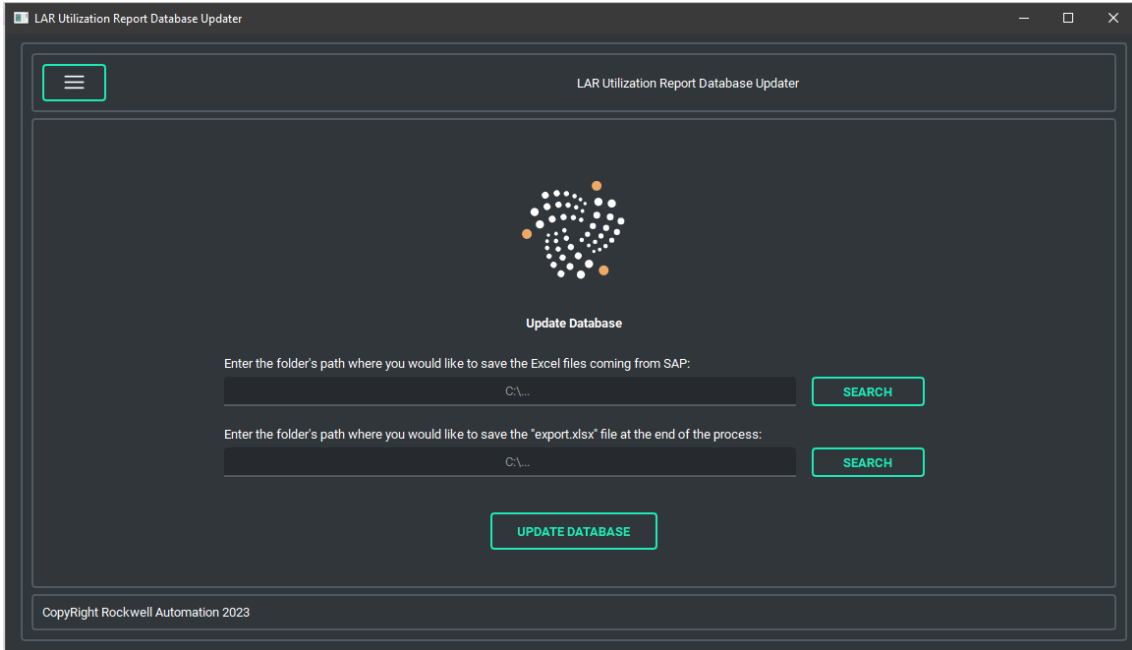
Menu lateral com os botões de acesso para as outras abas do sistema



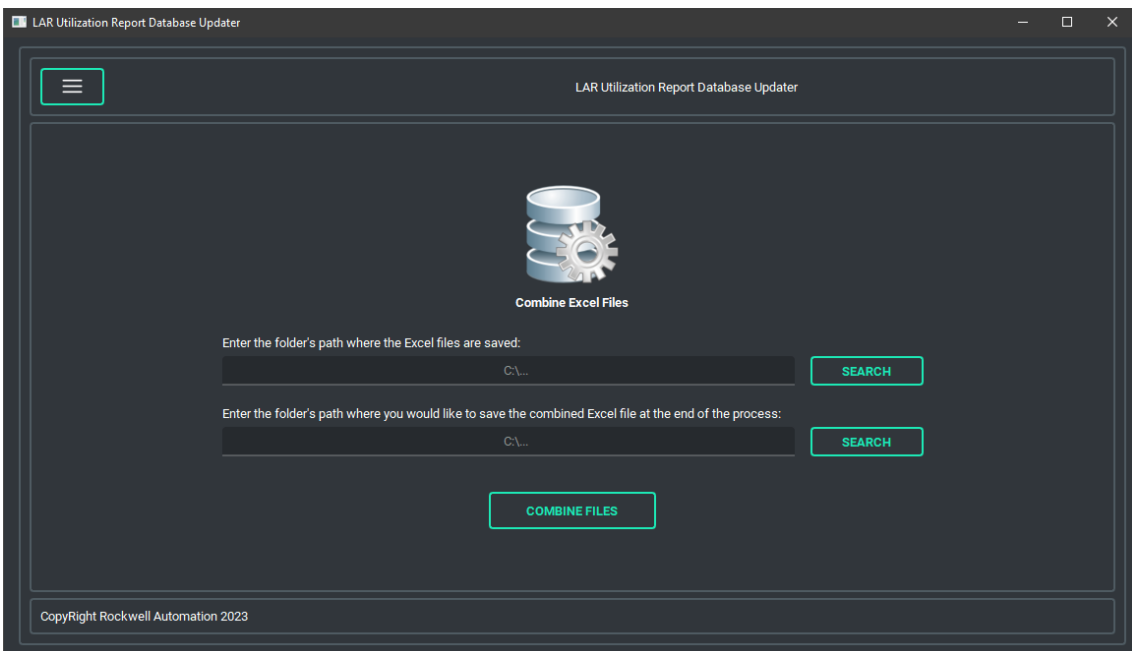
Aba "Home"



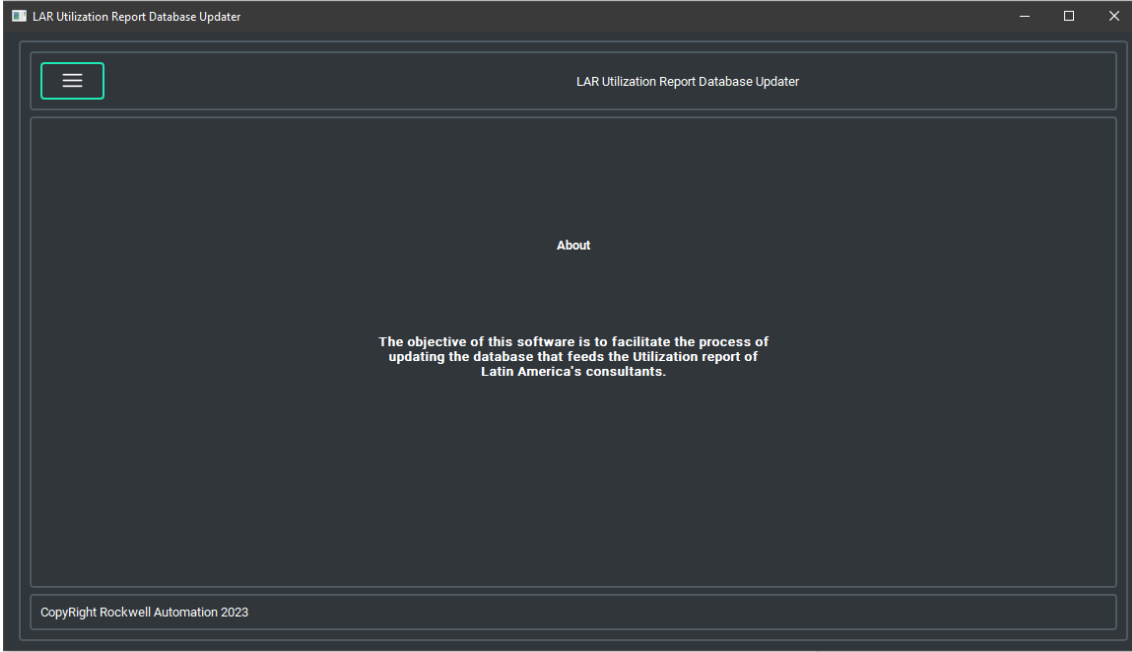
Aba "Update Database"



Aba "Combine Excel Files"



Aba "About"



APÊNDICE B - Código que realiza a extração das bases de dados

```

def update_database(self):
    try:
        # Definindo uma variável para começar a contar o tempo
        st = time.time()

        # Verifica qual é o caminho do local onde está armazenado o
        programa
        caminho = os.getcwd()

        # Abre o SAP
        pyautogui.alert('The program will start running. \n'
                        'During the process, DO NOT USE your computer. When
the process is finished, you will be notified.')

        pyautogui.hotkey('win', 'd')
        time.sleep(1.5)

        # Aperta a tecla "windows" do teclado
        pyautogui.press('win')
        time.sleep(0.5)

        # Busca pelo navegador Google Chrome
        pyautogui.write('Google Chrome')
        time.sleep(0.5)

        # Aperta a tecla "Enter"
        pyautogui.press('enter')
        # Espera o navegador abrir
        while not pyautogui.locateOnScreen(os.path.join(caminho,
r'images\1.png')):
            time.sleep(0.5)

        # Digita o endereço do SAP no Google Chrome
        pyautogui.write(r'https://portal.ra.rockwell.com/irj/portal')
        # Aperta enter
        pyautogui.press('enter')

        # Espera a página inicial do SAP carregar
        while not pyautogui.locateOnScreen(os.path.join(caminho,
r'images\2.png')):
            time.sleep(0.5)

        # Localiza o botão "Customer Relationship Mgmt"
        x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\3.png'))
        # Clica no botão "Customer Relationship Mgmt"
        pyautogui.click(x + largura / 2, y + altura / 2)

        time.sleep(4)

        # Localiza o botão "Field Service Engineer"
        x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\4.png'))
        # Clica no botão "Field Service Engineer"
        pyautogui.click(x + largura / 2, y + altura / 2)

        # Espera aparecer o botão "Service Confirmation Lifecycle"
        while not pyautogui.locateOnScreen(os.path.join(caminho,
r'images\5.png')):

```

```

        time.sleep(0.5)

        # Localiza o botão "Service Confirmation Lifecycle"
        x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\5.png'))
        # Clica no botão "Service Confirmation Lifecycle"
        pyautogui.click(x + largura / 2, y + altura / 2)

        # -----
        -----

        # Espera abrir o formulário
        while not pyautogui.locateOnScreen(os.path.join(caminho,
r'images\6.png')):
            time.sleep(0.5)

        # Aperta a tecla "Tab" até chegar no campo de Sales Org
        for x in range(12):
            pyautogui.press('tab')
            time.sleep(0.1)

        # Digita o código do PRIMEIRO PAÍS DO EMEA
        pyautogui.write('O 50031025')
        # Vai para o próximo campo
        pyautogui.press('tab')
        # Digita o código do ÚLTIMO PAÍS DO EMEA
        pyautogui.write('O 50038950')

        # Calcula a data do primeiro dia do mês atual
        data_primeiro_dia_mes = datetime.today().replace(day=1)
        data_primeiro_dia_mes = data_primeiro_dia_mes.strftime('%m/%d/%Y')
        # Calcula a data atual
        data_atual = datetime.today()
        data_atual = data_atual.strftime('%m/%d/%Y')

        # Aperta a tecla "Tab" até chegar no campo "Completed Date"
        for x in range(2):
            pyautogui.press('tab')
            time.sleep(0.1)

        # Digita a data do primeiro dia do mês atual
        pyautogui.write(f'{data_primeiro_dia_mes}')
        # Vai para o próximo campo
        pyautogui.press('tab')
        # Digita a data atual
        pyautogui.write(f'{data_atual}')

        # Aperta a tecla "Tab" até chegar no campo "Transaction Type"
        for x in range(4):
            pyautogui.press('tab')
            time.sleep(0.1)

        # Digita "ZRVC"
        pyautogui.write('ZRVC')

        # Aperta a tecla "Tab" até chegar no campo de "Created Date"
        for x in range(6):
            pyautogui.press('tab')
            time.sleep(0.1)

        # Calcula data do primeiro dia do mês anterior

```

```

data_primeiro_dia_mes_anterior = datetime.today().replace(day=1)
data_primeiro_dia_mes_anterior =
data_primeiro_dia_mes_anterior.replace(month=datetime.today().month - 1)
data_primeiro_dia_mes_anterior =
data_primeiro_dia_mes_anterior.strftime('%m/%d/%Y')

# Digita a data do primeiro dia do mês anterior
pyautogui.write(f'{data_primeiro_dia_mes_anterior}')
# Vai para o próximo campo
pyautogui.press('tab')
# Digita a data atual
pyautogui.write(f'{data_atual}')
#####
# Avança dois campos
pyautogui.press('tab')
time.sleep(0.1)
pyautogui.press('tab')
#####

#####
#####
# Aguarda até encontrar o radio button "Service Confirmation"
while not pyautogui.locateOnScreen(os.path.join(caminho,
r'images\6.png')):
    time.sleep(0.5)
# Localiza o radio button "Service Confirmation"
x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\6.png'))
# Clica no radio button "Service Confirmation"
pyautogui.click(x + largura / 2, y + altura / 2)

#####
#####
# Localiza a caixa de seleção "Completed"
x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\7.png'))
# Clica na caixa de seleção "Completed"
pyautogui.click(x + largura / 2, y + altura / 2)

# Localiza o botão para confirmar o envio do formulário
x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\8.png'))
# Clica no botão
pyautogui.click(x + largura / 2, y + altura / 2)

# Mantém o programa aguardando até que o SAP gere a tabela com as
Confirmations
while not pyautogui.locateOnScreen(os.path.join(caminho,
r'images\9.png')):
    time.sleep(0.5)

# Localiza o cabeçalho da tabela de Confirmations
x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\10.png'))
# Clica com o botão direito no cabeçalho da tabela
pyautogui.click(x + largura / 2, y + altura / 2, button='right')
time.sleep(1.5)

# Localiza a opção "spreadsheet"
x, y, largura, altura =

```

```

pyautogui.locateOnScreen(os.path.join(caminho, r'images\11.png'))
    # Clica na opção "spreadsheet"
    pyautogui.click(x + largura / 2, y + altura / 2)
    time.sleep(2)

    # Localiza o botão para gerar o arquivo em Excel
    x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\12.png'))
    # Clica no botão para gerar o arquivo em Excel
    pyautogui.click(x + largura / 2, y + altura / 2)

    # Espera aparecer a janela de seleção de local para salvar o
arquivo
    while not pyautogui.locateOnScreen(os.path.join(caminho,
r'images\13.png')):
        time.sleep(0.5)

    # Localiza o campo onde se insere o caminho para salvar o arquivo
    x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\13.png'))
    # Clica sobre esse campo
    pyautogui.click(x + largura / 2, y + altura / 2)

    # Escreve o caminho onde irá ser salvo a base de dados para as
Confirmations COMPLETED
    ano_atual = datetime.today().strftime('%Y')
    mes_atual = datetime.today().strftime('%m')
    nome_arquivo = '{}-{} - COMPLETED.xlsx'.format(ano_atual,
mes_atual)
    pyperclip.copy(os.path.join(self.txt_caminho_arq_SAP.text(),
nome_arquivo).replace('/', '\\'))
    pyautogui.hotkey('ctrl', 'v')
    # Aguarda um pouco
    time.sleep(0.5)
    # Pressiona a tecla "enter"
    pyautogui.press('enter')

    # Aguarda um pouco
    time.sleep(0.5)

    # Clica na opção "yes" para confirmar a substituição do arquivo
    pyautogui.press('left')
    time.sleep(0.1)
    pyautogui.press('enter')

    # Aguarda até que apareça uma janela do SAP com três opções:
"Allow", "Deny" e "Help"
    while not pyautogui.locateOnScreen(os.path.join(caminho,
r'images\14.png')):
        time.sleep(0.5)

    # Localiza o botão "Deny"
    x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\14.png'))
    # Clica nesse botão
    pyautogui.click(x + largura / 2, y + altura / 2)

    # Aguarda um pouco
    time.sleep(2)

    # Avança para o botão para finalizar o processo

```

```

pyautogui.press('tab')
time.sleep(1)
# Aperta a tecla "enter"
pyautogui.press('enter')
# Aguarda um pouco
time.sleep(1)

# -----
-----

# Localiza o botão para voltar para o formulário inicial
x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\16.png'))
# Clica nesse botão
pyautogui.click(x + largura / 2, y + altura / 2)
# Aguarda um momento
time.sleep(2)

# Avança até os campos de "Completed Date" e apaga os valores de
ambos os campos
for x in range(15):
    pyautogui.press('tab')
    time.sleep(0.1)
pyautogui.press('delete')
pyautogui.press('tab')
pyautogui.press('delete')

#####
# Avança três campos
for x in range(3):
    pyautogui.press('tab')
#####

# Localiza a caixa de seleção "Completed"
x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\17.png'))
# Desmarca essa caixa de seleção
pyautogui.click(x + largura / 2, y + altura / 2)

# Localiza a caixa de seleção "Open"
x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\18.png'))
# Marca essa caixa de seleção
pyautogui.click(x + largura / 2, y + altura / 2)

# Localiza o botão para confirmar o envio do formulário
x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\19.png'))
# Clica no botão
pyautogui.click(x + largura / 2, y + altura / 2)

# Mantém o programa aguardando até que o SAP gere a tabela com as
Confirmations
while not pyautogui.locateOnScreen(os.path.join(caminho,
r'images\20.png')):
    time.sleep(0.5)

# Localiza o cabeçalho da tabela de Confirmations
x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\21.png'))
# Clica no cabeçalho da tabela com o botão direito

```

```

pyautogui.click(x + largura / 2, y + altura / 2, button='right')
time.sleep(1.5)

# Localiza a opção "spreadsheet"
x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\22.png'))
# Seleciona a opção "spreadsheet"
pyautogui.click(x + largura / 2, y + altura / 2)
time.sleep(2)

# Localiza o botão para gerar o arquivo em Excel
x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\23.png'))
# Clica no botão para gerar o arquivo em Excel
pyautogui.click(x + largura / 2, y + altura / 2)

# Aguarda até que apareça a tela para escolher onde salvar o
arquivo
while not pyautogui.locateOnScreen(os.path.join(caminho,
r'images\24.png')):
    time.sleep(0.5)

# Localiza o campo onde se insere o caminho da pasta onde será
salvo o arquivo
x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\24.png'))
# Clica nesse campo
pyautogui.click(x + largura / 2, y + altura / 2)

# Escreve o caminho onde será salvo a base de dados para as
Confirmations com status OPEN
ano_atual = datetime.today().strftime('%Y')
mes_atual = datetime.today().strftime('%m')
nome_arquivo = '{}-{} - OPEN.XLSX'.format(ano_atual, mes_atual)
pyperclip.copy(os.path.join(self.txt_caminho_arq_SAP.text(),
nome_arquivo).replace('/', '\\'))
pyautogui.hotkey('ctrl', 'v')
# Aguarda um pouco
time.sleep(0.5)
# Pressiona a tecla "enter"
pyautogui.press('enter')

# Aguarda um pouco
time.sleep(0.5)

# Aperta o botão "yes" para confirmar a substituição do arquivo
pyautogui.press('left')
time.sleep(0.1)
pyautogui.press('enter')

# Aguarda até que apareça uma janela do SAP com três opções:
"Allow", "Deny" e "Help"
while not pyautogui.locateOnScreen(os.path.join(caminho,
r'images\25.png')):
    time.sleep(0.5)

# Localiza o botão "Deny"
x, y, largura, altura =
pyautogui.locateOnScreen(os.path.join(caminho, r'images\25.png'))
# Clica nesse botão
pyautogui.click(x + largura / 2, y + altura / 2)

```

```

# Aguarda um pouco
time.sleep(2)

# Avança para o botão para finalizar o processo
pyautogui.press('tab')
time.sleep(1)
# Aperta a tecla "enter"
pyautogui.press('enter')
time.sleep(1)

# ----- Código para atualizar o arquivo
export -----
path = self.txt_caminho_arq_SAP.text()
files = os.listdir(path)

combined_file = pd.DataFrame()

for file in files:
    full_path = os.path.join(path, file).replace('/', '\\')
    if file.endswith('.XLSX') or file.endswith('.xlsx'):
        combined_file = pd.concat([combined_file,
pd.read_excel(full_path)], ignore_index=True)

    combined_file.to_excel(os.path.join(self.txt_caminho_export.text(),
'export.xlsx').replace('/', '\\'), index=False)
# -----
-----

elapsed_time = time.time() - st

pyautogui.hotkey('win', 'd')
pyautogui.alert('Done! The database has been successfully updated!
\n'
                'Now you can use your computer. \n\n'
                f"Running Time: {time.strftime('%H:%M:%S',
time.gmtime(elapsed_time))}")

except Exception as e:
    self.msg('erro', "Ops! An error occurred. \n\n"
            "Error: {}".format(e))

```

APÊNDICE C - Código que concatena as bases de dados

```
def combine_excel_files(self):
    try:
        # Definindo uma variável para começar a contar o tempo
        st = time.time()

        path = self.txt_caminho_arquivos_excel.text()
        files = os.listdir(path)

        combined_file = pd.DataFrame()

        for file in files:
            full_path = os.path.join(path, file).replace('/', '\\')
            if file.endswith('.XLSX') or file.endswith('.xlsx'):
                combined_file = pd.concat([combined_file,
pd.read_excel(full_path)], ignore_index=True)

        combined_file.to_excel(os.path.join(self.txt_caminho_arq_combinado.text(),
'export.xlsx').replace('/', '\\'), index=False)

        elapsed_time = time.time() - st

        self.msg('ok', "The files were successfully merged! \n\n"
                f"Running Time: {time.strftime('%H:%M:%S',
time.gmtime(elapsed_time))}")

    except Exception as e:
        self.msg('erro', "Ops! An error occurred. \n\n"
                "Error: {}".format(e))
```

APÊNDICE D - Código para inicializar a interface gráfica e estabelecer a ação de cada botão

```

class MainWindow(QMainWindow, Ui_MainWindow):
    def __init__(self):
        super(MainWindow, self).__init__()
        self.setupUi(self)
        self.setWindowTitle("LAR Utilization Report Database Updater")

        # Definindo o icone do sistema
        app_icon = QIcon('/:icons/app-icon.png')
        self.setWindowIcon(app_icon)

        # ===== CONFIGURAÇÃO DA AÇÃO DE CADA BOTÃO
        # DO SISTEMA =====

        # Definindo a função do botão menu (sanduíche)
        self.btn_menu.clicked.connect(self.left_menu)

        # Definindo as funções dos botões do menu
        self.btn_menu_home.clicked.connect(lambda:
self.Pages.setCurrentWidget(self.pg_inicio))
        self.btn_menu_atualizar_base.clicked.connect(lambda:
self.Pages.setCurrentWidget(self.pg_atualizar_base))
        self.btn_menu_combinar_excel.clicked.connect(lambda:
self.Pages.setCurrentWidget(self.pg_combinar_excel))
        self.btn_menu_sobre.clicked.connect(lambda:
self.Pages.setCurrentWidget(self.pg_sobre))

        # ----- Página "Update Data Base" -----
        # Botão "Search" - Caminho para salvar os arquivos vindos do SAP
self.btn_caminho_arquivos_SAP.clicked.connect(self.search_path_SAVE_SAP_files)

        # Botão "Search" - Caminho para salvar o "export" ao final do
        # processo
self.btn_caminho_export.clicked.connect(self.search_path_SAVE_export)

        # Botão "Update Database"
self.btn_atualizar_base.clicked.connect(self.update_database)
        # -----

        # ----- Página "Combine Excel Files" -----
        # Botão "Search" - Caminho onde estão salvas todas as planilhas
        # vindas do SAP
self.btn_caminho_arquivos_excel.clicked.connect(self.search_path_Excel_files)

        # Botão "Search" - Caminho para salvar o arquivo Excel combinado ao
        # final do processo
self.btn_caminho_arq_combinado.clicked.connect(self.search_path_SAVE_combined_file)

        # Botão "Combine Files"

```

```
self.btn_combinar_arquivos.clicked.connect(self.combine_excel_files)
```

APÊNDICE E - Código da função “left menu”

```
def left_menu(self):
    width = self.left_frame.width()

    if width == 0:
        new_width = 240
    else:
        new_width = 0

    self.animation = QtCore.QPropertyAnimation(self.left_frame,
b'maximumWidth')
    self.animation.setDuration(500)
    self.animation.setStartValue(width)
    self.animation.setEndValue(new_width)
    self.animation.setEasingCurve(QtCore.QEasingCurve.InOutQuart)
    self.animation.start()
```

APÊNDICE F - Código das funções auxiliares da página "Update Database"

```
def search_path_SAVE_SAP_files(self):
    path = QFileDialog.getExistingDirectory(self, str("Enter the folder's
    path where you would like to save the Excel files coming from SAP:"),
    "/home", QFileDialog.ShowDirsOnly | QFileDialog.DontResolveSymlinks)
    path = str(path)
    self.txt_caminho_arq_SAP.setText(path)

def search_path_SAVE_export(self):
    path = QFileDialog.getExistingDirectory(self, str("Enter the folder's
    path where you would like to save the 'export.xlsx' file at the end of the
    process:"), "/home", QFileDialog.ShowDirsOnly |
    QFileDialog.DontResolveSymlinks)
    path = str(path)
    self.txt_caminho_export.setText(path)
```

APÊNDICE G - Código das funções auxiliares da página "Combine Excel Files"

```
def search_path_Excel_files(self):
    path = QFileDialog.getExistingDirectory(self, str("Enter the folder's
path where the Excel files are saved:"),
                                          "/home",
QFileDialog.ShowDirsOnly | QFileDialog.DontResolveSymlinks)
    path = str(path)
    self.txt_caminho_arquivos_excel.setText(path)

def search_path_SAVE_combined_file(self):
    path = QFileDialog.getExistingDirectory(self, str("Enter the folder's
path where you would like to save the combined file at the end of the
process:"), "/home", QFileDialog.ShowDirsOnly |
QFileDialog.DontResolveSymlinks)
    path = str(path)
    self.txt_caminho_arq_combinado.setText(path)
```

APÊNDICE H - Código da função "msg"

```
def msg(self, tipo, message):  
  
    msgbox = QMessageBox()  
  
    if tipo.lower() == 'ok':  
        msgbox.setIcon(QMessageBox.Information)  
    elif tipo.lower() == 'erro':  
        msgbox.setIcon(QMessageBox.Critical)  
    elif tipo.lower() == 'aviso':  
        msgbox.setIcon(QMessageBox.Warning)  
  
    msgbox.setText(message)  
    msgbox.exec()
```

APÊNDICE I - Código da função "resource_path"

```
def resource_path(relative_path):  
    """ Get absolute path to resource, works for dev and for PyInstaller  
    """  
    try:  
        # PyInstaller creates a temp folder and stores path in _MEIPASS  
        base_path = sys._MEIPASS  
    except Exception:  
        base_path = os.path.abspath(".")  
  
    return os.path.join(base_path, relative_path)
```

APÊNDICE J - Código para inicializar todo o sistema

```
if __name__ == '__main__':  
    app = QApplication(sys.argv)  
    apply_stylesheet(app, theme='dark_teal.xml')  
  
    window = MainWindow()  
    window.show()  
    app.exec()
```