



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

Patrick Mulinari

**Sistema para integração de dados e geração de indicadores de vendas e
comissionamento em uma empresa de transformadores**

Blumenau
2024

Patrick Mulinari

**Sistema para integração de dados e geração de indicadores de vendas e
comissionamento em uma empresa de transformadores**

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

Blumenau
2024

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Mulinari, Patrick

Sistema para integração de dados e geração de indicadores de vendas e comissionamento em uma empresa de transformadores / Patrick Mulinari ; orientador, Mauri Ferrandin, 2024.

77 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Campus Blumenau, Graduação em Engenharia de Controle e Automação, Blumenau, 2024.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Integração de dados. 3. Indicadores. I. Ferrandin, Mauri. II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Controle e Automação. III. Título.

Patrick Mulinari

**Sistema para integração de dados e geração de indicadores de vendas e
comissionamento em uma empresa de transformadores**

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

Blumenau, 13 de dezembro de 2024.

Banca Examinadora:

Prof. Mauri Ferrandin, Dr.
Instituição Universidade Federal de Santa Catarina

Prof. Carlos Roberto Moratelli, Dr.
Instituição Universidade Federal de Santa Catarina

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

Este trabalho é dedicado aos meus colegas de classe e aos
meus queridos pais.

AGRADECIMENTOS

Gostaria, primeiramente, de expressar minha mais profunda gratidão à minha família, que nunca mediu esforços para me ajudar a alcançar os meus objetivos. Um agradecimento mais que especial ao meu pai, cuja dedicação e ensinamentos marcaram profundamente minha trajetória de vida. Sou imensamente grato por todo o esforço que ele dedicou ao longo de sua vida, trabalhando incansavelmente na lavoura, dia após dia, até seus últimos momentos em vida, para garantir que eu tivesse uma educação de qualidade e, com isso, fosse aberto o caminho para que eu chegasse até onde cheguei.

Expresso também minha sincera gratidão a todos os professores que, ao longo da minha jornada acadêmica, contribuíram para minha formação com dedicação, conhecimento e inspiração. Em especial, quero agradecer ao meu orientador, que me guiou até aqui, tornando possível a realização deste trabalho.

Quero também expressar minha gratidão aos meus colegas de trabalho, que estiveram ao meu lado ao longo dessa jornada. Seu apoio, tanto no meu desenvolvimento profissional quanto pessoal, foi fundamental para a realização deste projeto. A colaboração, troca de experiências e incentivo que recebi de vocês foram essenciais para que eu pudesse superar desafios e alcançar os objetivos traçados.

Por fim, não poderia deixar de agradecer aos meus amigos, que sempre estiveram ao meu lado, oferecendo apoio e motivação ao longo dessa jornada. Vocês foram fundamentais, seja nos momentos de celebração ou nos desafios enfrentados.

RESUMO

O presente trabalho aborda o desenvolvimento de um sistema automatizado para a coleta, organização e visualização de indicadores de desempenho, com foco em otimizar processos de vendas e comissionamento. O objetivo principal é avaliar a eficácia das ferramentas utilizadas, como Python, VBA e Power BI, na integração de dados provenientes de diferentes fontes, incluindo sistemas como SAP e tabelas do Excel. Inicialmente, foram desenvolvidas soluções para automatizar a coleta de dados de cotações, entrada de pedidos, ROL e carteira, utilizando scripts para acessar e manipular grandes volumes de informações. Em seguida, os dados foram organizados e consolidados em bases estruturadas, permitindo a criação de dashboards interativos que facilitam a tomada de decisão. A integração de dados, embora desafiadora devido à diversidade de formatos e à necessidade de sincronização em tempo real, proporcionou ganhos significativos em termos de agilidade, confiabilidade e precisão das informações disponíveis. Resultados obtidos demonstram a eficácia do sistema proposto em reduzir o tempo de processamento, minimizar erros humanos e fornecer uma visualização clara e dinâmica dos principais indicadores da empresa, contribuindo para uma gestão mais eficiente e estratégica.

Palavras-chave: Indicadores de desempenho; integração de dados; painéis interativos.

ABSTRACT

This work addresses the development of an automated system for collecting, organizing, and visualizing performance indicators, focusing on optimizing sales and commissioning processes. The main objective is to evaluate the effectiveness of tools such as Python, VBA, and Power BI in integrating data from various sources, including systems like SAP and Excel sheets. Initially, solutions were developed to automate the data collection for quotations, order intake, net operating income, and backlog, utilizing scripts to access and manipulate large volumes of information. Subsequently, the data was organized and consolidated into structured databases, enabling the creation of interactive dashboards that facilitate decision-making. The data integration, although challenging due to the diversity of formats and the need for real-time synchronization, provided significant gains in terms of agility, reliability, and accuracy of the available information. The results obtained demonstrate the effectiveness of the proposed system in reducing processing time, minimizing human errors, and providing a clear and dynamic visualization of the company's key performance indicators, contributing to more efficient and strategic management.

Keywords: Performance indicators; data integration; dashboards.

LISTA DE FIGURAS

Figura 1 – Simbologia de um DFD estilo Gane-Sarson.	18
Figura 2 – Estrutura de um DFD separado por níveis.	19
Figura 3 – Simbologia de um fluxograma.	20
Figura 4 – Etapas do desenvolvimento incremental de um sistema automatizado. .	20
Figura 5 – Gravação de <i>scripts</i> por meio do SAP.	21
Figura 6 – Diagrama DFD da célula de vendas.	28
Figura 7 – Fluxograma do processo de aquisição de dados do histórico de cotações.	29
Figura 8 – Trecho da tabela de cotações com os dados obtidos através do <i>script</i> em Python.	30
Figura 9 – Fluxograma do processo de atualização da base de dados de cotações. .	31
Figura 10 – Formatação dos dados de entrada, ROL e carteira.	33
Figura 11 – Fluxograma do processo de manipulação dos dados de entrada, ROL e carteira.	34
Figura 12 – Trecho da tabela de dados de entrada.	35
Figura 13 – Trecho da tabela de dados de ROL/Carteira.	35
Figura 14 – Diagrama de fluxo de dados da célula de comissionamento.	37
Figura 15 – Fluxograma do processo de aquisição de dados das atas do PCP. . . .	38
Figura 16 – Trecho da tabela com os dados das atas do PCP.	39
Figura 17 – Formatação dos dados da carteira de vendas.	39
Figura 18 – Fluxograma do processo de aquisição de dados da carteira de vendas de equipamentos.	40
Figura 19 – Trecho da tabela com os dados da carteira de vendas.	41
Figura 20 – Desenvolvimento incremental do executável de aditivos.	43
Figura 21 – Visualização dos dados do arquivo DB.	44
Figura 22 – Visualização dos dados do arquivo JSON.	45
Figura 23 – Janela inicial do executável.	46
Figura 24 – Janela de edição de dados.	48
Figura 25 – Janela para inserção de comentários.	49
Figura 26 – Janela do histórico dos aditivos.	51
Figura 27 – Campos para filtragem de dados.	52
Figura 28 – Janela do SAP GUI para exibição de ordens de serviço.	54
Figura 29 – Campo onde está localizado o valor orçado da OS.	54
Figura 30 – Tabela de dados no SAP onde está localizado o valor de contratação. .	55
Figura 31 – Janela com as informações do extrato.	56
Figura 32 – Modelo de genérico do extrato.	57
Figura 33 – Planilha em Excel contendo a interface de comando do sistema. . . .	58
Figura 34 – Caixas para inserção das datas de início e data final.	59

Figura 35 – Janela inicial da transação IW73 no SAP.	59
Figura 36 – Janela inicial da transação KOB1 no SAP.	60
Figura 37 – Importação direta de dados para o Power BI.	61
Figura 38 – Campos de manipulação das tabelas importadas no Power BI.	62
Figura 39 – <i>Dashboard</i> no Power BI das cotações de sobressalentes.	66
Figura 40 – <i>Dashboard</i> no Power BI das cotações de serviços e reformas.	66
Figura 41 – <i>Dashboard</i> no Power BI dos dados de entrada.	67
Figura 42 – <i>Dashboard</i> no Power BI dos dados de ROL e Carteira.	68
Figura 43 – <i>Dashboard</i> no Power BI dos dados de ata do PCP.	69
Figura 44 – <i>Dashboard</i> no Power BI dos dados da carteira de vendas de equipamentos.	70
Figura 45 – <i>Dashboard</i> no Power BI dos dados inseridos no executável de aditivos.	71
Figura 46 – <i>Dashboard</i> no Power BI dos custos gerais das obras encerradas.	72
Figura 47 – <i>Dashboards</i> no Power BI dos custos específicos das obras encerradas.	72
Figura 48 – <i>Dashboard</i> no Power BI dos custos com carretas de óleo nas obras encerradas.	73
Figura 49 – <i>Dashboard</i> no Power BI dos custos com aditivos nas obras encerradas.	73

LISTA DE QUADROS

Quadro 1 – Formatação do texto.	29
Quadro 2 – Trecho de código VBA para transferência de dados entre tabelas. . . .	34
Quadro 3 – Uso de dicionário para verificar tipos de equipamentos relevantes. . . .	41
Quadro 4 – Código para criação da base de dados com <i>SQLite3</i>	44
Quadro 5 – Código para inserção de dados na tabela de aditivos.	47
Quadro 6 – Código para edição dos dados da tabela de aditivos.	48
Quadro 7 – Código para exclusão na tabela de aditivos.	49
Quadro 8 – Código para liberação dos aditivos para o histórico.	51
Quadro 9 – Código para filtragem de dados e exibição na tabela.	52

LISTA DE TABELAS

Tabela 1 – Principais Funções DAX e Suas Aplicações	63
---	----

LISTA DE ABREVIATURAS E SIGLAS

DAX	<i>Data Analysis Expressions</i>
DFD	Diagrama de Fluxo de Dados
DML	<i>Data Manipulation Language</i>
GUI	<i>Graphical User Interface</i>
KPI	<i>Key Performance Indicator</i>
MAPI	<i>Messaging Application Programming Interface</i>
OS	Ordem de serviço
OV	Ordem de venda
PCP	Planejamento e Controle da Produção
PEP	Projeto Elétrico de Particulares
ROL	Receita Operacional Líquida
SAP	<i>System Analysis Program Development</i>
SQL	<i>Structured Query Language</i>
VBA	<i>Visual Basic for Applications</i>

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVO GERAL	16
1.2	OBJETIVOS ESPECÍFICOS	16
2	REVISÃO BIBLIOGRÁFICA	17
2.1	MAPEAMENTO DE PROCESSOS ATRAVÉS DE DIAGRAMAS TIPO DFD E FLUXOGRAMAS	17
2.1.1	Mapeamento com o uso de DFD (Diagrama de Fluxo de Dados)	17
2.1.2	Mapeamento com o uso de fluxogramas	18
2.2	DESENVOLVIMENTO DE SISTEMAS AUTOMATIZADOS ATRA- VÉS DO MODELO INCREMENTAL	19
2.3	AQUISIÇÃO DE DADOS AUTOMATIZADA NA PLATAFORMA SAP ATRAVÉS DO USO DE SAP <i>SCRIPTS</i>	21
2.4	OBTENÇÃO E ORGANIZAÇÃO DE DADOS UTILIZANDO MA- CROS NO MICROSOFT EXCEL	22
2.4.1	Integração com pastas de e-mail no Microsoft Outlook	22
2.4.2	Manipulação de dados entre planilhas através de condicionais	23
2.5	INTERFACES GRÁFICAS EM PYTHON INTEGRADAS A BANCOS DE DADOS PARA AUTOMAÇÃO DE PROCESSOS	23
2.5.1	Biblioteca Tkinter	23
2.5.2	Biblioteca CustomTkinter	24
2.5.3	Biblioteca SQLite3	24
2.5.4	Biblioteca Pandas	25
2.6	DASHBOARDS INTERATIVAS NO MICROSOFT POWER BI PARA VISUALIZAÇÃO DE INDICADORES	25
3	DESENVOLVIMENTO DAS AUTOMAÇÕES DE COLETA E ANÁLISE DOS INDICADORES	27
3.1	INDICADORES DA CÉLULA DE VENDAS	27
3.1.1	Cotações	28
<i>3.1.1.1</i>	<i>Coleta de dados do histórico de cotações</i>	<i>29</i>
<i>3.1.1.2</i>	<i>Atualização da base de dados das cotações</i>	<i>30</i>
3.1.2	Entrada, ROL e Carteira	32
3.2	INDICADORES DA CÉLULA DE COMISSIONAMENTO	36
3.2.1	Indicadores de contratação	37
<i>3.2.1.1</i>	<i>Controle da ata do PCP</i>	<i>38</i>
<i>3.2.1.2</i>	<i>Controle da carteira de vendas</i>	<i>39</i>
3.2.2	Indicadores de obras em execução	42
<i>3.2.2.1</i>	<i>Aplicativo executável para cadastro de aditivos</i>	<i>42</i>

3.2.2.1.1	Entrega 1: criação da base de dados, inserção de dados e tela inicial . . .	43
3.2.2.1.2	Entrega 2: edição e exclusão dos dados e inserção de comentários . . .	47
3.2.2.1.3	Entrega 3: histórico, liberação dos dados e filtragem da tabela	50
3.2.2.1.4	Entrega 4: valores orçado e contratado, custos extras e geração de extratos	53
3.2.2.2	<i>Indicadores de custos de ordens de serviço encerradas</i>	<i>57</i>
3.3	ELABORAÇÃO DAS <i>DASHBOARDS</i> INTERATIVAS NO POWER BI A PARTIR DAS BASES DE DADOS	61
3.3.1	Importação de dados no Power BI	61
3.3.2	Criação de colunas calculadas e medidas com DAX	62
4	RESULTADOS	65
4.1	CÉLULA DE VENDAS	65
4.1.1	Cotações	65
4.1.2	Entrada, ROL e Carteira	67
4.2	CÉLULA DE COMISSIONAMENTO	68
4.2.1	Contratações	68
4.2.2	Obras em execução	70
5	CONCLUSÃO	74
	REFERÊNCIAS	75

1 INTRODUÇÃO

No mundo atual, a mudança à qual as empresas estão submetidas devido aos diferentes avanços tecnológicos, desenvolvimentos econômicos, automação de processos, crescimento contínuo, entre outros fatores, motiva que elas estejam preparadas para gerenciar adequadamente os recursos financeiros que permitem tomar decisões alinhadas com os objetivos e metas empresariais (CAJAS, 2019).

Nesse contexto, os indicadores acabam se tornando dados essenciais ao planejamento e controle dos processos organizacionais, pois constituem a base desse planejamento. Dessa forma, eles estabelecem medidas verificadoras do cumprimento de metas e objetivos e sinalizam o rumo que a organização está seguindo (FERNANDES, 2004). Todavia, quando a coleta dos dados exibidos nos indicadores é realizada de maneira manual, surgem algumas limitações significativas.

Além do processo manual demandar um tempo considerável, ele também está suscetível a erros humanos, resultando frequentemente em números incompatíveis com a realidade e que podem distorcer a análise de desempenho. Diante dessas barreiras, muitas empresas têm alterado suas rotinas organizacionais para seguirem um modelo gerenciado por processos e indicadores que possam ser automatizados, aumentando, assim, a agilidade e o controle de atividades e reduzindo custos (MANZUETO, 2016).

Em uma empresa multinacional, onde os processos envolvem um alto nível de complexidade, essa automatização dos indicadores torna-se ainda mais crucial. A abrangência e a escala das operações exigem uma coleta e análise de dados ágeis e confiáveis, essenciais para garantir o controle minucioso sobre a produção, o atendimento aos padrões de qualidade e o cumprimento de prazos rigorosos. Para atender a essa demanda, faz-se necessário o uso de algumas ferramentas tecnológicas para coleta e manipulação de dados e para criação de *dashboards* interativas.

Nesse sentido, a linguagem Python acaba se tornando uma ferramenta extremamente versátil e adequada para uma variedade de aplicações, desde tarefas simples até projetos complexos (CARVALHO; MUSCI, 2024). Da mesma forma, a linguagem VBA permite criar ferramentas padronizadas que executam tarefas mais complexas, como análise de dados, modelagem, comunicação de banco de dados, dentre outros (MICHEL *et al.*, 2021). Já para a criação de *dashboards*, como destacado em (NEVES, 2024), o Microsoft Power BI é uma excelente forma de apresentar informações de maneira objetiva e compreensível, permitindo visualizar rapidamente o desempenho, identificar tendências, analisar padrões e tomar decisões informadas, além da possibilidade de interagir entre os diferentes visuais.

Diante desse cenário, o presente trabalho apresenta uma solução que integra essas tecnologias para a coleta, manipulação e análise dos indicadores referentes às células de vendas de serviços e sobressalentes e de comissionamento em uma empresa de transforma-

dores. A proposta busca aprimorar a precisão e agilidade dos processos utilizando-se de *scripts* em Python e VBA que se comunicam diretamente com a plataforma SAP (*System Analysis Program Development*) para aquisição de dados. Além disso, os dados serão manipulados através de técnicas como SQL (*Structured Query Language*) (em Python) e Macros (em VBA). Por fim, esses dados serão importados e apresentados em diferentes painéis interativos no Power BI.

Para a apresentação dessa solução, esse trabalho está estruturado da seguinte forma: o Capítulo 2 irá apresentar uma revisão bibliográfica, abordando conceitos fundamentais utilizados na confecção do trabalho. O Capítulo 3 descreverá o desenvolvimento da solução proposta, detalhando as ferramentas e metodologias utilizadas. Logo após, no Capítulo 4, serão apresentados os resultados obtidos com a implementação da solução e os seus impactos e melhorias nas operações. Por fim, o Capítulo 5 trará as conclusões e considerações finais do trabalho, com as discussões, contribuições e possibilidades de aprimoramento da solução apresentada.

1.1 OBJETIVO GERAL

O objeto geral deste projeto consiste em desenvolver uma solução automatizada para a obtenção e análise dos indicadores para as células de vendas e de comissionamento de uma empresa de transformadores. Essa solução visa substituir a coleta manual dos dados utilizando ferramentas como Python, VBA e Microsoft Power BI, garantindo maior precisão e eficiência nos processos de aquisição e integração dos dados com *Dashboards* interativas.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desse projeto estão listados a seguir.

- Identificar dados que podem ser adquiridos diretamente pelo SAP ou que necessitem de outra abordagem;
- Implementar um modelo de coleta de dados ideal para cada tipo de indicador por meio das linguagens Python e VBA;
- Desenvolver *dashboards* interativas através do Power BI utilizando-se da linguagem DAX quando necessário;
- Parametrizar e validar eventuais ganhos e melhorias provenientes da automação dos processos já mencionados.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo aborda a análise de trabalhos relacionados e metodologias empregadas em soluções semelhantes ao tema deste estudo. Busca-se identificar abordagens já existentes, discutir suas contribuições e limitações, e explorar práticas que possam ser adaptadas e aplicadas ao problema em questão.

2.1 MAPEAMENTO DE PROCESSOS ATRAVÉS DE DIAGRAMAS TIPO DFD E FLUXOGRAMAS

Para organizar e compreender as atividades de forma detalhada, pode-se mapear os processos por diferentes representações visuais, entre elas estão o DFD e o fluxograma de processos. Essas ferramentas permitem descrever tanto o fluxo de informações e as interações entre atividades quanto a lógica sequencial de cada processo.

2.1.1 Mapeamento com o uso de DFD (Diagrama de Fluxo de Dados)

O DFD é uma ferramenta de modelagem gráfica de análise estruturada que descreve a transformação de entradas e saídas, ou seja, a técnica de análise estruturada (JÓIA; WERNER, 2021). É uma ferramenta que facilita a compreensão de como os dados são processados e movidos entre diferentes componentes. Amplamente utilizado na modelagem de sistemas, cada diagrama é composto por alguns elementos básicos: entidades externas, processos, armazenamento de dados e fluxos de dados. Cada um desses elementos possui uma simbologia própria, como visto na Figura 1 que representa a simbologia chamada Gane-Sarson.

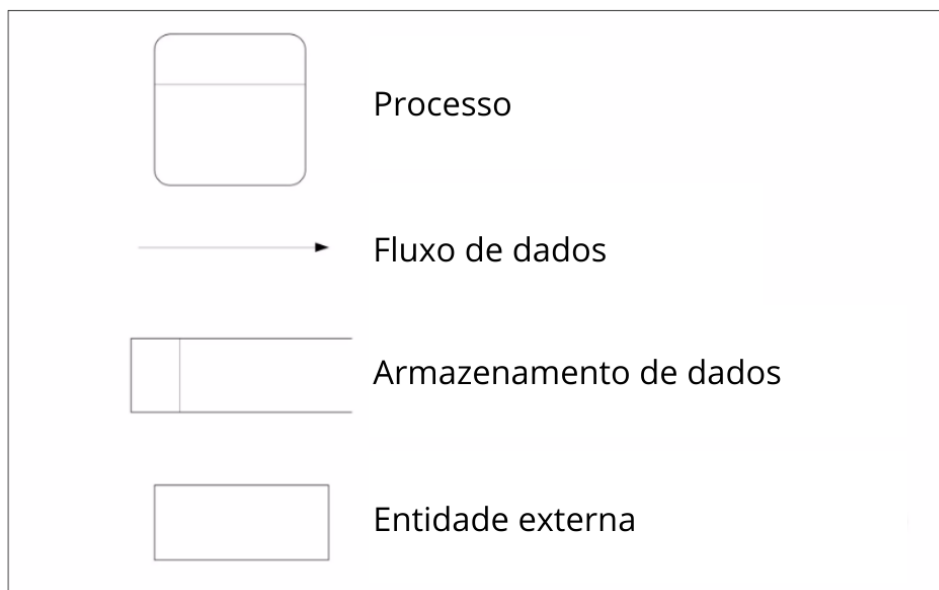
Ao combinar esses elementos, o DFD cria uma representação detalhada de como os dados circulam e são manipulados dentro de um sistema, ajudando na identificação de entradas, saídas e possíveis adversidades.

Para a elaboração de um DFD utiliza-se a abordagem “top-down” em que cada um dos diferentes níveis de detalhe do sistema em estudo é mostrado através de diferentes níveis de DFD. A primeira representação do sistema é elaborada através de um diagrama conhecido como diagrama de contexto. Este diagrama, denominado nível 0, é representado através de um processo e dos fluxos de entrada e saída do sistema, o que permite delimitar a área em estudo. O diagrama de contexto é decomposto num primeiro DFD onde são mostrados os principais processos, fluxos e arquivos de dados, bem como as entidades externas envolvidas. Se o diagrama de contexto se chama nível 0, então este primeiro DFD será de nível 1 (UNIVERSIDADE FERNANDO PESSOA, 1997).

Na Figura 2, temos um esquemático que ilustra a estrutura de um sistema DFD, destacando o fluxo de informações entre os processos e elementos separados por níveis:

Com o DFD, é possível descrever sistemas complexos de forma segmentada, permi-

Figura 1 – Simbologia de um DFD estilo Gane-Sarson.



Fonte: Adaptado de (BILYK, 2019).

tindo que cada nível de detalhe seja explorado conforme necessário. Este tipo de diagrama é particularmente útil na fase de análise de sistemas, pois oferece uma visão organizada e acessível do funcionamento do sistema e de suas interações com o ambiente externo.

2.1.2 Mapeamento com o uso de fluxogramas

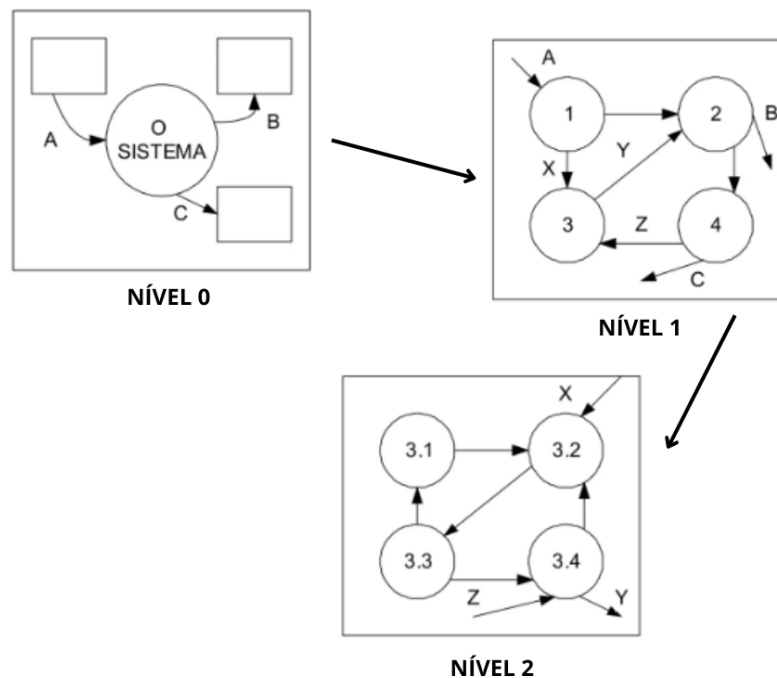
Fluxogramas são ferramentas gráficas amplamente utilizadas para modelar e representar visualmente a sequência lógica de algoritmos ou processos de programação. Cada elemento em um fluxograma representa uma etapa ou decisão específica no fluxo de dados, utilizando uma simbologia padronizada que facilita a visualização das ações e decisões necessárias no processo. Apesar do fluxograma ser uma das formas mais simples de se observar, criar e apresentar um algoritmo em forma gráfica, o uso de ferramentas computacionais tradicionais reduz a naturalidade e a liberdade do usuário na expressão de suas ideias (GONDIM; AMBROSIO, 2008)

Como visto em (PAULISTA; ALVES, R. A., 2015), um fluxograma também tem a finalidade de estudar um processo, identificando a melhor sequência de execução para uma tarefa, com o objetivo de detectar possíveis desvios ou pontos de otimização na lógica. Esses diagramas são elaborados com sequências de decisão e ação e cada um deles possui uma simbologia que ajuda na compreensão do sistema.

A simbologia básica dos fluxogramas inclui elementos que identificam o início e o fim de um processo, bem como as ações, decisões, conexões, entre outros. A Figura 3 apresenta alguns dos símbolos mais comuns utilizados em fluxogramas e seus significados.

Através destes símbolos, é possível descrever os blocos básicos dos algoritmos computacionais, e o seu reduzido número possibilita uma fácil compreensão das suas

Figura 2 – Estrutura de um DFD separado por níveis.



Fonte: Adaptado de (SOUZA, 2016).

funcionalidades para o desenho de algoritmos (MARQUES *et al.*, 2011).

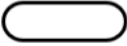
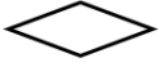
O uso de fluxogramas ajuda a documentar e organizar a lógica de processos de maneira clara, o que facilita o desenvolvimento de algoritmos e melhora a comunicação na equipe. Além disso, fluxogramas são fundamentais para analisar e otimizar processos, permitindo identificar pontos de melhoria e simplificação no fluxo de trabalho, o que leva a uma implementação mais eficiente.

2.2 DESENVOLVIMENTO DE SISTEMAS AUTOMATIZADOS ATRAVÉS DO MODELO INCREMENTAL

O desenvolvimento incremental de sistemas automatizados é uma abordagem em que o sistema é desenvolvido em etapas, permitindo que novas funcionalidades sejam adicionadas de maneira sequencial e que os componentes sejam integrados e testados ao longo do processo. Esse modelo é amplamente utilizado em projetos de software e automação, pois permite a criação gradual de sistemas complexos e facilita o ajuste e a melhoria contínua durante o ciclo de desenvolvimento.

O Modelo Incremental surge como uma melhoria do Modelo em Cascata. Ao invés de especificar e desenvolver tudo de uma só vez, este modelo trabalha com incrementos, ou seja, pequenos pedaços de software entregues de cada vez. Este modelo combina elementos do Modelo em Cascata aplicados de maneira iterativa, ou seja, de forma que o progresso aconteça através de sucessivos refinamentos, melhorados a cada iteração (PRESSMAN,

Figura 3 – Simbologia de um fluxograma.

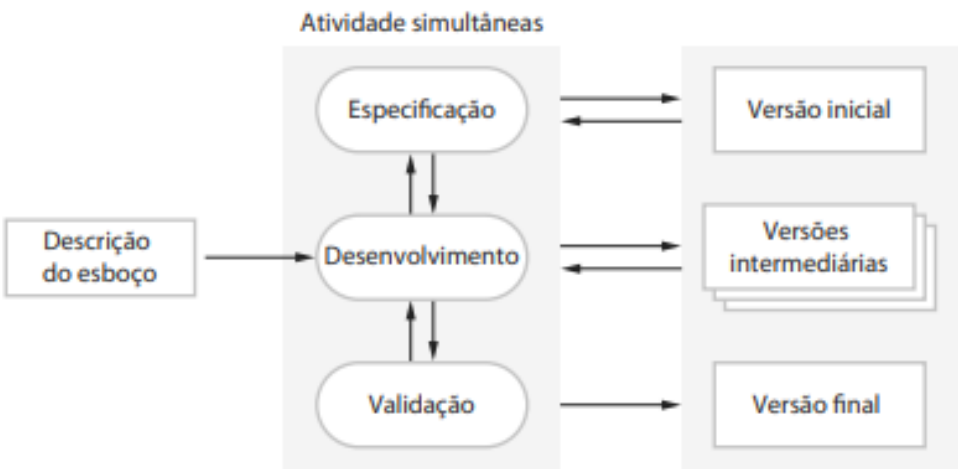
	Indica o início ou fim do processo.
	Indica cada atividade que precisa ser executada.
	Indica um ponto de tomada de decisão (Trata-se de uma afirmação. Se verdadeira, o processo segue por um caminho, se falsa, por outro).
	Indica a direção do fluxo de um ponto ou atividade para outro.
	Indica os documentos utilizados no processo.
	Indica espera. No interior do símbolo é representado o tempo aproximado de espera.
	Indica que o fluxograma continua a partir deste ponto em outro círculo com a mesma letra ou número, que aparece em seu interior.

Fonte: Adaptado de (BASTIANI, 2012).

2006).

O desenvolvimento incremental é baseado na ideia de desenvolver uma implementação inicial, expô-la aos comentários dos usuários e continuar por meio da criação de várias versões até que um sistema adequado seja desenvolvido (SOMMERVILLE, 2011). Essa abordagem é ilustrada na Figura 4 e é essencial para a automação, pois permite que cada módulo ou componente do sistema seja testado isoladamente, garantindo que as integrações sejam feitas de forma gradual.

Figura 4 – Etapas do desenvolvimento incremental de um sistema automatizado.



Fonte: (SOMMERVILLE, 2011).

Essa prática gradual reduz os riscos de erros na integração de diferentes partes

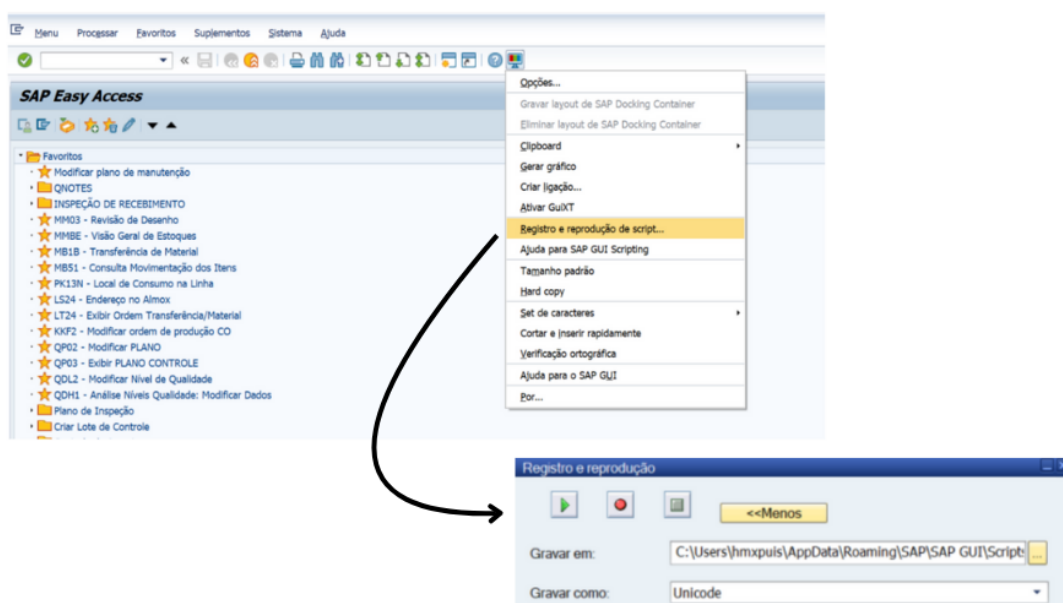
do código e facilita a manutenção e atualização contínua dos sistemas, tornando-os mais adaptáveis a mudanças e/ou atualizações das necessidades dos usuários.

2.3 AQUISIÇÃO DE DADOS AUTOMATIZADA NA PLATAFORMA SAP ATRAVÉS DO USO DE SAP *SCRIPTS*

A aquisição de dados automatizada na plataforma SAP por meio de *scripts* é uma solução eficaz para extrair informações de forma estruturada, reduzindo a necessidade de intervenções manuais e otimizando o fluxo de trabalho em processos de grande volume. Utilizando linguagens como Python e VBA, essa abordagem permite a automação de tarefas repetitivas, como navegação entre telas, preenchimento de campos e extração de dados de transações específicas, o que acelera o processo de coleta e minimiza o risco de erros humanos.

Para acessar a ferramenta de gravação de *scripts*, deve-se clicar no símbolo de monitor colorido e selecionar a opção “Registro e reprodução de script...”. Dessa forma, será aberta a gravadora, que permite iniciar a gravação ao clicar no botão com o círculo vermelho, encerrar uma gravação ou execução com o botão com o quadrado que se torna azul após iniciar algum desses processos e executar um *script* gravado através do triângulo verde. Além disso, os campos adicionais “Gravar em:” e “Gravar como:” permitem definir as características de localização, nome e codificação do arquivo antes de se iniciar sua gravação (SILVA, 2021). Esse processo pode ser visualizado por meio da Figura 5.

Figura 5 – Gravação de *scripts* por meio do SAP.



Fonte: Adaptado de (SILVA, 2021).

Com esse sistema do SAP, cada interação registrada pode ser convertida em um *script* editável, permitindo sua adaptação e aplicação a diferentes cenários de extração e

automação de dados e recriar a qualquer momento do tempo a mesma sequência de passos armazenada dentro do código (ALVES, J. O., 2019). Essas funcionalidades destacam o SAP *Script* como a ferramenta ideal para automatizar o processo de inserção de informações no SAP, proporcionando eficiência e precisão na execução de tarefas repetitivas.

2.4 OBTENÇÃO E ORGANIZAÇÃO DE DADOS UTILIZANDO MACROS NO MICROSOFT EXCEL

O VBA (*Visual Basic for Applications*) é uma linguagem de programação incorporada em muitos softwares da Microsoft, sendo amplamente utilizada em ambientes corporativos, principalmente devido à sua integração direta com aplicativos como o Microsoft Excel, software para gerenciamento de planilhas eletrônicas. Essa característica tem tornado o VBA uma ferramenta poderosa para empresas que buscam soluções de automação e personalização rápidas para tarefas rotineiras (ROSA, 2024).

O uso de macros no Microsoft Excel é amplamente utilizado como uma técnica para automatização de tarefas repetitivas e complexas em diversas áreas. Uma macro é um pequeno programa que contém uma lista de instruções. Por ser um repositório de operações, uma macro pode executar um conjunto de tarefas mediante um único procedimento que pode ser invocado rapidamente (SANTOS, 2015). Essa tecnologia proporciona uma padronização das práticas organizacionais, algo considerado essencial para o gerenciamento dos dados com os objetivos estratégicos de uma empresa.

2.4.1 Integração com pastas de e-mail no Microsoft Outlook

A integração entre Excel e Outlook, aplicativo de gerenciamento de e-mails, usando macros em VBA é uma estratégia amplamente adotada para automatizar a coleta e organização de dados provenientes de e-mails. Com o uso de VBA, é possível configurar macros para acessar pastas de e-mail específicas, buscar mensagens com certos critérios e extrair dados relevantes das pastas para o Excel.

Para iniciar uma automatização nas pastas do Outlook, é comum utilizar uma associação tardia. Essa associação usa a função *GetObject* do VBA ou a função *CreateObject* para inicializar o Outlook. Após estabelecer a conexão, é possível acessar pastas e subpastas específicas utilizando o método *GetNamespace* para obter o espaço de nomes MAPI (*Messaging Application Programming Interface*) e, em seguida, o método *Folders* para navegar pela hierarquia de pastas (MICROSOFT LEARN, 2023). Esse método permite navegar por diferentes níveis de pastas e subpastas no Outlook, facilitando a automação de tarefas específicas em pastas designadas.

Além de melhorar a eficiência, essa integração promove uma centralização das informações, uma vez que os dados do Outlook podem ser automaticamente armazenados e organizados no Excel. Esse recurso é especialmente útil em setores que lidam com grandes

volumes de dados recebidos por e-mail.

2.4.2 Manipulação de dados entre planilhas através de condicionais

O uso de condicionais em macros no Excel para a manipulação de dados entre planilhas é um recurso poderoso para criar processos automáticos de análise e transferência de informações. Utilizando comandos que unem laços de repetição *for/while* e condicionais *if/else* em VBA, os usuários podem estabelecer uma lógica condicional que direciona o fluxo dos dados entre as planilhas. Dentro desse contexto, a utilização de referências como `Range()` e `Cells()` é fundamental para acessar e manipular informações específicas em células.

A função `Range()`, por exemplo, permite acessar uma célula ou intervalo de células por meio de uma referência direta, como "A1". Esse método de referência direta é conveniente para operações pontuais em células específicas, facilitando a construção de instruções VBA que precisam acessar dados localizados em posições fixas da planilha. Além disso, para operações que exigem percorrer uma série de células, a função `Cells()` é amplamente utilizada. Essa função permite realizar *loops* que acessam múltiplas células em linhas ou colunas específicas, controlando a navegação pela planilha com variáveis. Como exemplo, o uso de `Cells(i,1)`, que, dentro de uma estrutura de repetição, permitiria que o incremento da variável de controle *i* percorresse todos os valores associados à primeira coluna da planilha de trabalho (GUTIÉRREZ, 2017).

Esse tipo de método facilita a manipulação de dados entre diferentes planilhas e arquivos, reduzindo o risco de inconsistências. Com isso, garante-se uma maior confiabilidade nos processos de análise e uma execução eficiente das operações, especialmente em ambientes que demandam alto volume de dados e padronização nos resultados.

2.5 INTERFACES GRÁFICAS EM PYTHON INTEGRADAS A BANCOS DE DADOS PARA AUTOMAÇÃO DE PROCESSOS

Para a automação de processos e manipulação de dados, bibliotecas dedicadas são de suma importância para a integração entre interfaces gráficas e bancos de dados. Tkinter e CustomTkinter são opções robustas para o desenvolvimento de interfaces em Python, permitindo que os usuários naveguem de forma intuitiva e interativa. Essas bibliotecas, combinadas com SQLite3 e Pandas, viabilizam o armazenamento e a manipulação de dados de forma simultânea, criando uma base sólida para o desenvolvimento de sistemas de automação que exigem acesso dinâmico a bancos de dados.

2.5.1 Biblioteca Tkinter

Tkinter é uma biblioteca Python usada para criar uma GUI (*Graphical User Interface*). Ela possui uma vasta coleção de *widgets* que podem ser usados para construir

interfaces de usuário complexas de forma rápida. Além disso, permite a programação orientada a eventos, possibilitando que os programadores projetem aplicativos interativos. Essa biblioteca também oferece uma variedade de ferramentas, como botões, rótulos, caixas de texto e menus. Com eles, é possível desenvolver diversos tipos de interfaces de usuário, incluindo formulários, caixas de diálogo e menus (COZAR, 2023).

Outra funcionalidade dessa biblioteca é a disponibilidade de funções para manipulação dos *widgets*, como definir suas propriedades, posicioná-los na tela e responder à entrada do usuário. O módulo Tkinter oferece várias funções que facilitam a criação de uma interface gráfica de usuário. Com ele, é possível iniciar o *loop* principal do aplicativo, permitindo o processamento de eventos e a interação contínua com o usuário. Além disso, há recursos para definir o tamanho, a posição e o título da janela, bem como atualizar sua exibição automaticamente após qualquer modificação, garantindo uma experiência visual dinâmica e responsiva (SEETHA *et al.*, 2023).

2.5.2 Biblioteca CustomTkinter

Já a biblioteca CustomTkinter é uma extensão da biblioteca Tkinter que adiciona novos recursos e ferramentas. Seu objetivo é facilitar a construção de uma GUI mais complexa e poderosa em Python. Ele inclui ferramentas para o desenvolvimento de temas personalizados, como cores, fontes e imagens personalizadas (COZAR, 2023).

Essa biblioteca conta com recursos avançados, como manipulação de eventos, animação e gráficos. A aplicação pode responder à entrada do usuário, como cliques do mouse e pressionamento de teclas, utilizando a manipulação de eventos. A aplicação pode usar animação para criar efeitos visuais dinâmicos, como o desaparecimento gradual de textos ou imagens. Os gráficos permitem que a aplicação gere imagens personalizadas, como gráficos e tabelas (SEETHA *et al.*, 2023).

2.5.3 Biblioteca SQLite3

SQLite3 é uma biblioteca baseada em um mecanismo de banco de dados SQL embutido e não possui um processo de servidor separado como a maioria dos outros bancos de dados SQL. Esse mecanismo lê e grava diretamente em um único arquivo no sistema de arquivos local. O formato de arquivo do banco de dados é multiplataforma. Essas características fazem dessa biblioteca uma escolha popular como formato de arquivo de aplicação (BHOSALE *et al.*, 2015).

Baseados nos comandos DML (*Data Manipulation Language*) em SQL, existem quatro comandos principais usados com essa biblioteca para adicionar, modificar, remover e consultar dados do banco de dados. INSERT adiciona novas linhas, UPDATE modifica linhas existentes, DELETE remove linhas, e SELECT é utilizado para consultar dados específicos. Esses comandos são usados para manter e acessar todos os valores de dados reais dentro do banco de dados. Todos eles operam ao nível de linha, adicionando, alterando,

removendo ou selecionando as linhas especificadas. Embora possam agir sobre múltiplas linhas, cada comando só pode atuar diretamente nas linhas contidas dentro de uma única tabela (KREIBICH, 2010).

A opção de poder utilizar essas funcionalidades sem necessitar de um servidor, torna a biblioteca SQLite3 uma solução prática e eficiente para o armazenamento e manipulação de dados em aplicações de pequeno a médio porte. A ausência de um processo de servidor e a capacidade de ler e gravar diretamente em arquivos locais facilitam a implementação da SQLite3 em sistemas que necessitam de um banco de dados leve, mas robusto.

2.5.4 Biblioteca Pandas

A biblioteca Pandas é uma ferramenta poderosa e amplamente utilizada para manipulação e análise de dados em Python. Ela foi projetada para oferecer uma estrutura flexível e eficiente para trabalhar com grandes conjuntos de dados, fornecendo funcionalidades robustas como estruturas de dados otimizadas, que permitem armazenar e manipular dados tabulares de forma intuitiva e eficiente. Além disso, ela suporta operações avançadas, como filtragem, agregação, fusão e transformação de dados, que são fundamentais em análises exploratórias e processamento de dados (MCKINNEY *et al.*, 2011).

A utilização dessa biblioteca com banco de dados facilita o seu uso em Python ao fornecer ferramentas poderosas para manipulação e análise de dados de forma eficiente e integrada. Com Pandas, é possível conectar-se facilmente a bancos de dados do tipo SQL e extrair grandes volumes de dados diretamente para estruturas de dados do Python, como *DataFrames*, que organizam os dados de maneira tabular e facilitam operações complexas de filtragem, agregação e transformação (BANTILAN, 2020).

2.6 DASHBOARDS INTERATIVAS NO MICROSOFT POWER BI PARA VISUALIZAÇÃO DE INDICADORES

O Microsoft Power BI é composto por uma série de aplicações, conectores e serviços que transformam os dados recolhidos em informações compreensíveis, consistentes e visualmente interativas, que sustentam o processo de tomada de decisão. Apesar da plataforma ser destinada à utilização empresarial, é considerada *user-friendly* pois permite que os utilizadores se conectem a uma determinada fonte de dados e a visualizem sem grandes constrangimentos e exigências (RIBEIRO, 2023). Uma vez que o Power BI facilita a transformação de dados complexos em visualizações interativas e compreensíveis, torna-se possível utilizar uma variedade de "visuais" chamados de *Dashboards* que otimizam a análise e interpretação dos dados, principalmente se tratando de indicadores.

Para a construção destes *dashboards*, podem ser utilizados diversos visuais disponíveis no Power BI, como gráficos, tabelas e filtros personalizáveis que permitem uma visualização aprimorada dos dados selecionados. Cada um desses visuais pode ser desen-

volvido com medidas específicas, criadas utilizando a linguagem DAX (*Data Analysis Expressions*). No Power BI, uma medida é uma fórmula que realiza cálculos sobre os dados de um modelo e se ajusta dinamicamente ao contexto do relatório ou visualização, levando em consideração filtros e outros itens de contexto. O real poder das medidas reside na capacidade de realizar cálculos complexos, como médias ponderadas, taxas de crescimento, acumulados e previsões, indo além das simples agregações (HIPÓLITO, 2023).

Através dessas ferramentas disponíveis no Power BI, as empresas podem compreender e monitorar seus KPIs com precisão e clareza. A flexibilidade dos visuais e a capacidade de personalização das medidas possibilitam a criação de *dashboards* que refletem as necessidades de cada organização, tornando o processo de tomada de decisão mais rápido e transparente.

3 DESENVOLVIMENTO DAS AUTOMAÇÕES DE COLETA E ANÁLISE DOS INDICADORES

Neste capítulo, é apresentado o desenvolvimento dos sistemas previamente propostos, abrangendo as diferentes formas de coleta e manipulação de dados. Primeiramente, serão detalhadas as ferramentas e os processos utilizados para a obtenção dos dados para a célula de vendas de sobressalentes e serviços. Após, será a vez de apresentar os métodos e procedimentos adotados para a aquisição dos dados para a célula de comissionamento. O objetivo de descrever esses processos é oferecer uma visão ampla e detalhada de todas as etapas envolvidas na realização do sistema, desde a concepção inicial até a implementação completa.

Um ponto que é importante destacar é referente ao cenário anterior à aplicação dos processos explicados a seguir. Tanto na célula de vendas quanto na célula de comissionamento, quase não existia análise de indicadores. O que existia eram apenas dados superficiais, coletados manualmente e que demandavam um tempo considerável para serem atualizados. Dessa maneira, o acompanhamento dos resultados de custos e ganhos da seção era impreciso.

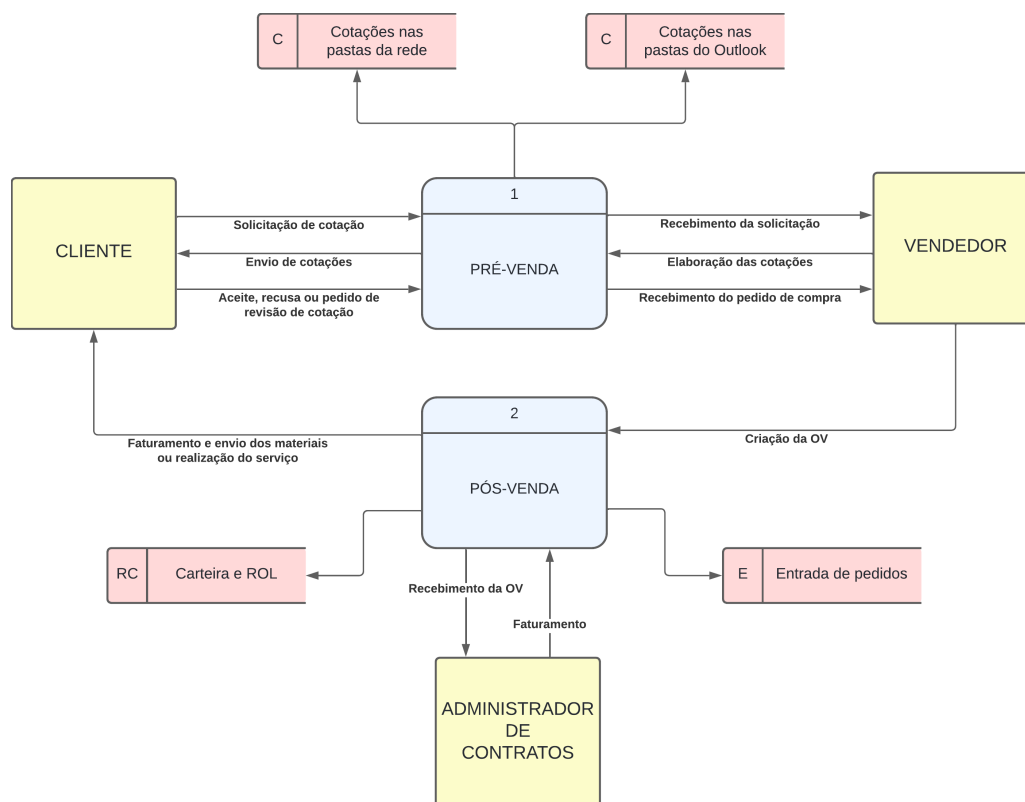
3.1 INDICADORES DA CÉLULA DE VENDAS

Antes de iniciar a implementação, é fundamental entender o funcionamento dos processos na célula de vendas da seção. Essa célula desempenha um papel estratégico na empresa, sendo responsável por atender às demandas dos clientes em diferentes categorias de vendas. A célula é organizada em três principais tipos de vendas:

- **Venda de sobressalentes:** inclui a venda de partes e peças para transformadores (a óleo e a seco) e chaves seccionadoras.
- **Venda de serviços:** abrange atividades executadas em campo, que podem incluir desde o conserto ou troca de peças do equipamento até a realização de ensaios de tensão, temperatura, impacto, entre outros.
- **Venda de produtos digitais (não será abordada):** tipo de venda mais nova da célula, onde são vendidos aparelhos que monitoram a "saúde" dos transformadores, realizando *feedbacks* em tempo real via software.

Pelo fato de produtos digitais serem uma área mais nova e sem tanta estrutura, ela será deixada para futuras implementações, a fim de manter o foco nos outros dois tipos de vendas. Tanto a venda de sobressalentes quanto a venda de serviços compartilham processos similares no que diz respeito à comunicação entre cliente, vendedor e administrador de contratos. Na Figura 6, é possível visualizar o fluxo de dados da interação entre essas entidades.

Figura 6 – Diagrama DFD da célula de vendas.



Fonte: O autor

Basicamente, o processo de venda é feito majoritariamente de forma passiva, onde quem procura pelo produto ou serviço é o próprio cliente. Nesse processo, a primeira etapa de negociação com o cliente são as cotações. Depois que o cliente aceita a proposta e envia o pedido de compras, o vendedor cria uma OV (Ordem de venda) e a repassa ao administrador de contratos. Este, por sua vez, cuida do processo pós-venda até o faturamento e envio dos materiais ou realização do serviço ao cliente. Em vermelho estão os momentos em que as bases de dados são abastecidas.

3.1.1 Cotações

No primeiro contato, geralmente por e-mail, o cliente envia uma solicitação de material ou serviço, juntamente com alguma informação do equipamento (número de série, foto da placa de identificação, desenho, etc.). Em caso de sobressalentes, caso falte alguma informação, o vendedor solicita ao cliente para que lhe dê informações adicionais para ofertar os itens corretos.

Tendo todas as informações em mãos, o vendedor elabora um documento padronizado de cotação dos itens solicitados, inserindo informações padronizadas do cliente, materiais, frete, condição de pagamento, data de emissão, entre outros. O nome desse documento segue sempre o mesmo padrão, e isso é de extrema importância para que mais adiante seja possível coletar os dados desses documentos. No Quadro 1 destacado a seguir,

é possível visualizar o padrão do nome das cotações, onde C,S ou R representam o tipo de cotação (sobressalentes, serviços ou reforma, respectivamente), "XXXXXXXX" representa o código da cotação e "Y" representa qual a revisão daquela cotação, sendo 0 a primeira.

Quadro 1 – Formatação do texto.

C/S/R00XXXXXXXX_rY - CLIENTE - Material/Serviço - TRAFO/SECO/CHAVES

Fonte: O autor

Após o envio da cotação, o cliente avalia se está tudo correto com os materiais e os valores. Caso tenha alguma informação incorreta ou faltante, ou ainda, se o cliente deseja algum desconto ou condição de pagamento a prazo, é feita uma revisão da cotação.

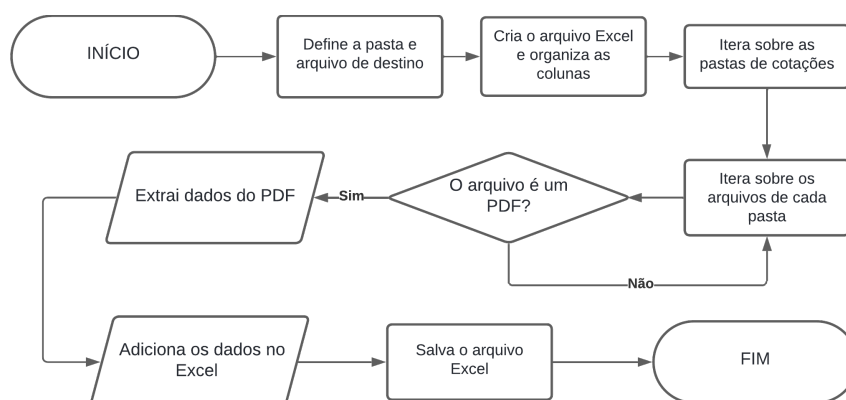
Cada cotação enviada para os clientes é armazenada em uma pasta da rede com a nomenclatura já evidenciada, como um arquivo DOCX, um arquivo XLSX (memorial de cálculo) e um arquivo PDF. Além disso, no Microsoft Outlook, é criada uma nova pasta com a mesma nomenclatura, onde são guardados todos os e-mails provenientes daquela oferta/venda.

3.1.1.1 Coleta de dados do histórico de cotações

Levando em conta que não havia nenhum controle sobre o total de cotações e o total orçado, surgiu a necessidade de elaborar uma maneira de conseguir buscar esses dados de maneira automatizada. Como havia sido dito anteriormente, todas as cotações são padronizadas, tanto o nome dos documentos, quanto os seus conteúdos. Dessa forma, é possível pensar em um código que possa varrer as pastas da rede em busca do histórico dessas cotações e dos seus dados.

Para isso, foi desenvolvido um *script* em Python capaz de ler cada um dos arquivos PDFs presentes nas pastas de cotações da rede, separadas por ano e mês, e salvar os dados requisitados em uma planilha do Excel. A Figura 7 representa, através de um fluxograma, como é feita a aquisição desses dados do interior dos arquivos.

Figura 7 – Fluxograma do processo de aquisição de dados do histórico de cotações.



Fonte: O autor

Para que fosse possível ler os arquivos em segundo plano e extrair os dados do conteúdo deles, utilizou-se a biblioteca em Python chamada PyPDF2. Ao identificar cada um dos dados que, através da padronização, sempre ficam em posições pré-estabelecidas, eram criados vetores para cada arquivo contendo as seguintes informações: data de emissão, código, revisão, cliente, material, tipo de equipamento e mercado (interno ou externo). Depois disso, cada vetor era adicionado como linha na planilha do Excel que, ao final do processo, foi formatada como tabela. Na Figura 8, é possível visualizar parte dessa tabela com os dados obtidos. A coluna de valores foi censurada por serem dados confidenciais.

Figura 8 – Trecho da tabela de cotações com os dados obtidos através do *script* em Python.

DATA EMISSÃO	COTAÇÃO	REVISÃO	CLIENTE	MATERIAL	MODALIDADE	VALOR	MERCADO
28/10/2024	C0021608645	r0		Sobressalentes	TRAFO		MERCADO INTERNO
28/10/2024	C0021606809	r0		Pino	CHAVES		MERCADO INTERNO
28/10/2024	C0021608555	r0		Sensor de temperatura	TRAFO		MERCADO INTERNO
28/10/2024	C0021604429	r0		RELE SEL	TRAFO		MERCADO INTERNO
25/10/2024	C0021599491	r0		Sobressalentes	SECO		MERCADO INTERNO
25/10/2024	C0021607494	r0		Conectores	TRAFO		MERCADO INTERNO
24/10/2024	C0021609413	r0		Radiadores	TRAFO		MERCADO INTERNO
24/10/2024	C0021601661	r0		Kit Vedacao	TRAFO		MERCADO INTERNO
23/10/2024	C0021607474	r0		Kit ventilação forçada	SECO		MERCADO INTERNO
22/10/2024	C0021607845	r0		DAP	TRAFO		MERCADO INTERNO
21/10/2024	C0021589092	r0		Buchas	TRAFO		MERCADO INTERNO
19/10/2024	C0021599431	r0		ENROLAMENTO AT	SECO		MERCADO INTERNO
18/10/2024	C0021603604	r0		Bucha	TRAFO		MERCADO INTERNO
18/10/2024	C0021607906	r0		Sobressalentes	TRAFO		MERCADO INTERNO
18/10/2024	C0021601967	r0		Bucha	TRAFO		MERCADO INTERNO
17/10/2024	C0021608064	r0		Buchas	TRAFO		MERCADO INTERNO
17/10/2024	C0021605324	r0		Buchas	TRAFO		MERCADO INTERNO
17/10/2024	C0021605765	r0		Painel corrugado	TRAFO		MERCADO INTERNO
17/10/2024	C0021607325	r0		Valvulas	TRAFO		MERCADO INTERNO
15/10/2024	C0021601400	r0		Buchas	TRAFO		MERCADO INTERNO
14/10/2024	C0021606394	r0		Indicador de nível	TRAFO		MERCADO INTERNO
14/10/2024	C0021602108	r0		Relé VRT	TRAFO		MERCADO INTERNO
11/10/2024	C0021604926	r0		Caixa IP	SECO		MERCADO INTERNO
11/10/2024	C0021607233	r0		Sensor de temperatura	TRAFO		MERCADO INTERNO
11/10/2024	C0021604007	r0		Sensor de temperatura	TRAFO		MERCADO INTERNO
11/10/2024	C0021607133	r0		Polos	CHAVES		MERCADO INTERNO
11/10/2024	C0021591690	r0		TC	TRAFO		MERCADO INTERNO
10/10/2024	C0021604847	r0		Indicador de nível	TRAFO		MERCADO INTERNO
10/10/2024	C0021598810	r0		Termômetros	TRAFO		MERCADO INTERNO
09/10/2024	C0021604377	r0		Monitor de temperatura	TRAFO		MERCADO INTERNO
09/10/2024	C0021604846	r0		Sobressalentes	TRAFO		MERCADO INTERNO
09/10/2024	C0021600109	r0		INOC	TRAFO		MERCADO INTERNO
08/10/2024	C0021605714	r0		Kit vedação	TRAFO		MERCADO INTERNO

Fonte: O autor

Na tabela estão os dados de cotações de sobressalentes, organizados com informações detalhadas como data de emissão, código da cotação, revisão, cliente, material, preço e mercado. Para as cotações de serviços e reformas, a tabela foi adaptada, incluindo uma coluna extra para diferenciá-las, enquanto a coluna "material" foi substituída pelo "tipo de serviço" correspondente. Com essas adaptações, o histórico de cotações está completo e organizado,

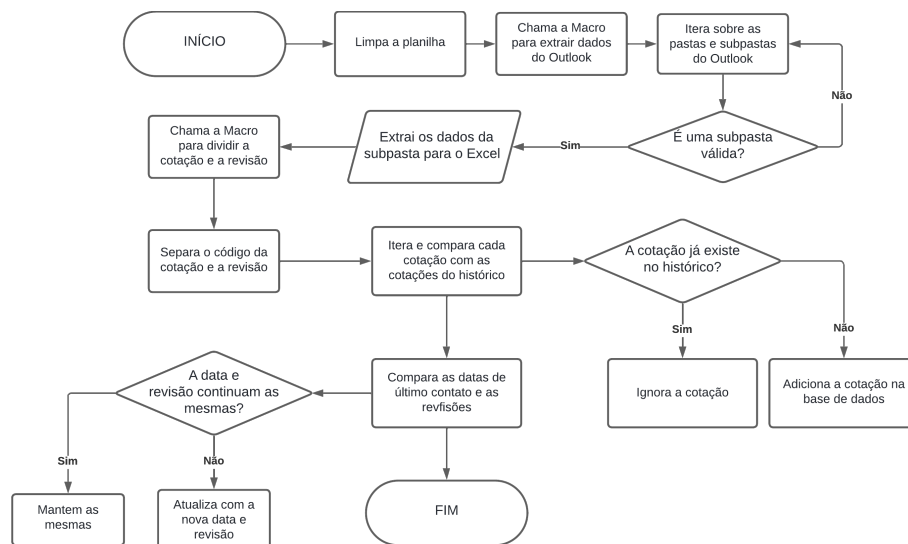
3.1.1.2 Atualização da base de dados das cotações

Após ter adquirido o histórico das cotações, era necessário encontrar uma maneira mais rápida e eficaz de atualizar a base de dados à medida que novas cotações eram criadas. Como já dito anteriormente, para cada caso são criadas pastas no Outlook com o mesmo padrão das pastas da rede. Essa compatibilidade de informações acabou por ser de grande utilidade para o caso. As pastas de cotações do Outlook, assim como as pastas da rede, também são separadas primeiro por mercado, depois por ano e por mês.

Através desse compartilhamento de padrões, foi elaborado um *script* feito através de macros em VBA que é disparado semanalmente pelo vendedor, comparando as cotações

presentes nas pastas de e-mail com as cotações já presentes na base de dados do histórico. Caso alguma cotação esteja presente nas pastas de e-mail, porém não esteja incluída na base de dados, uma nova linha é inserida com os dados dessa cotação. No fluxograma presente na Figura 9, é possível acompanhar como funciona o código que atualiza a base de dados das cotações.

Figura 9 – Fluxograma do processo de atualização da base de dados de cotações.



Fonte: O autor

Para estabelecer a conexão com a aplicação em execução do Microsoft Outlook, utiliza-se a função `GetObject("Outlook.Application")`. Caso o programa não esteja em execução, uma nova instância é criada por meio da função `CreateObject("Outlook.Application")`. Após essa etapa, o acesso às pastas desejadas é realizado utilizando o ambiente MAPI, uma interface robusta empregada pelo Outlook para o gerenciamento de pastas e itens de e-mail.

Dentro desse ambiente, o objeto `Folders()` permite a navegação hierárquica entre todas as pastas de e-mail do usuário. Após localizar a pasta correspondente ao mês em questão, um laço de repetição é executado para iterar sobre cada subpasta de cotação contida na pasta mensal. Durante esse processo, o nome de cada subpasta é extraído, permitindo a identificação e manipulação dos dados necessários.

Além disso, para aprimorar o acompanhamento das atividades dos vendedores, o programa utiliza a propriedade `ReceivedTime` para calcular o intervalo de tempo desde o último e-mail registrado em cada pasta. Esse recurso permite identificar há quanto tempo ocorreu a última interação com o cliente em cada caso específico. Com essa análise, os vendedores podem priorizar os casos mais oportunos para solicitar atualizações aos clientes, aumentando as chances de venda.

Após coletar as cotações e suas respectivas informações, o código realiza uma comparação entre cada cotação extraída e a base de dados já existente. Caso uma cotação

não esteja presente na base, uma nova linha é automaticamente adicionada à planilha. Além disso, o código verifica a data do último contato e a revisão associada a cada cotação. Se houver qualquer alteração nessas informações desde a última atualização, os dados na planilha são atualizados.

Com todas essas informações das novas cotações disponíveis, o vendedor apenas precisa atribuir a "temperatura" da cotação (frio, esquentando ou quente), indicando o quão próxima a venda está de ser concretizada. Além disso, também é necessário atualizar o status da cotação, pois caso a venda seja concluída, o vendedor deve marcar esse campo como "GANHO". Essa classificação permite calcular de forma eficiente a taxa de conversão das cotações diretamente no Power BI.

3.1.2 Entrada, ROL e Carteira

Para organizar a base de dados de entrada, ROL e carteira, é fundamental compreender o significado e a relevância de cada um desses elementos. Diferentemente das cotações, a entrada e o ROL estão diretamente vinculados às metas da seção, o que torna sua análise ainda mais essencial. A seguir, é apresentado o significado de cada um deles.

- **Entrada:** refere-se aos pedidos que foram confirmados e convertidos em vendas em um período específico, ou seja, a data utilizada como base é a data de criação da OV.
- **ROL (Receita Operacional Líquida):** são todas as vendas que foram faturadas em um determinado período, ou seja, a data utilizada como base é a data de faturamento.
- **Carteira:** são todas as vendas que estão na entrada, mas que ainda não foram convertidas em ROL, tendo a data utilizada a data de previsão de faturamento.

Quando uma OV é criada, é atribuído um valor total da venda nela. Nesse valor total, estão presentes o valor líquido somado ao valor dos impostos daquela venda. Portanto, o valor que entra na base de dados interna como entrada e carteira é o valor líquido daquela venda. Essa base de dados é encontrada no sistema da empresa e pode ser exportada como arquivo XLSX.

O formato dos dados dessa base pode ser visto na Figura 10. A tabela apresenta diversas colunas que auxiliam na segmentação dos dados, permitindo a separação por unidade da empresa, data, centro de vendas e, principalmente, pelo código da OV. Esse código é fundamental para identificar a origem da venda: se inicia com "7", trata-se de uma venda de serviços, já se inicia com "5", refere-se a uma venda de sobressalentes, sucata ou Tessin, este último refere-se a uma venda específica de silício.

Outro aspecto relevante da tabela são as colunas que detalham os valores. Esses estão organizados em valor de entrada, valor de ROL e valor de carteira. Além disso, destaca-se a coluna intermediária denominada "Valor Pendente Faturamento". Essa coluna

Figura 10 – Formatação dos dados de entrada, ROL e carteira.

Local de Negócio	Equipe de Vendas	Data	Doc. Vendas	Cliente						
					Vlr. Entrada	Vlr. Carteira	Vlr. Pendente Faturamento	Vlr. Pendente Rec. Recota	Vlr. ROL	
					BRL	BRL			BRL	BRL
BETM	BC3	WTD VD NAC COM-PP	09.02.2022	51123079	#/802082					
BETM	BC3	WTD VD NAC COM-PP	21.02.2022	51123079	#/802082					
BETM	BC3	WTD VD NAC COM-PP	23.02.2022	51241091	#/911696					
BETM	BC3	WTD VD NAC COM-PP	25.02.2022	51241091	#/911696					
BETM	BC3	WTD VD NAC COM-PP	01.03.2022	51241091	#/911696					
BETM	BC3	WTD VD NAC COM-PP	01.03.2022	51458708	#/596876					
BETM	BC3	WTD VD NAC COM-PP	08.03.2022	51241091	#/911696					
BETM	BC3	WTD VD NAC COM-PP	19.03.2022	51503808	#/935574					
BETM	BC3	WTD VD NAC COM-PP	22.03.2022	51503808	#/935574					
BETM	BC3	WTD VD NAC COM-PP	28.03.2022	51458708	#/596876					
BETM	BC3	WTD VD NAC COM-PP	29.03.2022	51458708	#/596876					
BETM	BC3	WTD VD NAC COM-PP	01.04.2022	51503808	#/935574					
BETM	BC3	WTD VD NAC COM-PP	19.04.2022	51556421	#/446050					
BETM	BC3	WTD VD NAC COM-PP	29.04.2022	51556421	#/446050					
BETM	BC3	WTD VD NAC COM-PP	06.05.2022	51582415	#/446050					
BETM	BC3	WTD VD NAC COM-PP	23.05.2022	51582415	#/446050					
BETM	BC3	WTD VD NAC COM-PP	23.06.2022	51646768	#/446050					
BETM	BC3	WTD VD NAC COM-PP	23.06.2022	51136047	#/446050					
BETM	BC3	WTD VD NAC COM-PP	23.06.2022	51137638	#/446050					
BETM	BC3	WTD VD NAC COM-PP	24.06.2022	51646768	#/446050					
BETM	BC3	WTD VD NAC COM-PP	24.06.2022	51136047	#/446050					
BETM	BC3	WTD VD NAC COM-PP	24.06.2022	51137638	#/446050					
BETM	BC3	WTD VD NAC COM-PP	22.07.2022	51675818	#/446050					
BETM	BC3	WTD VD NAC COM-PP	26.07.2022	51152283	#/446050					
BETM	BC3	WTD VD NAC COM-PP	12.08.2022	51746030	#/446050					
BETM	BC3	WTD VD NAC COM-PP	18.08.2022	51157183	#/446050					
BETM	BC3	WTD VD NAC COM-PP	19.08.2022	51675818	#/446050					
BETM	BC3	WTD VD NAC COM-PP	19.08.2022	51746030	#/446050					
BETM	BC3	WTD VD NAC COM-PP	19.08.2022	51157183	#/446050					
BETM	BC3	WTD VD NAC COM-PP	20.08.2022	51756893	#/446050					
BETM	BC3	WTD VD NAC COM-PP	26.08.2022	51756893	#/446050					

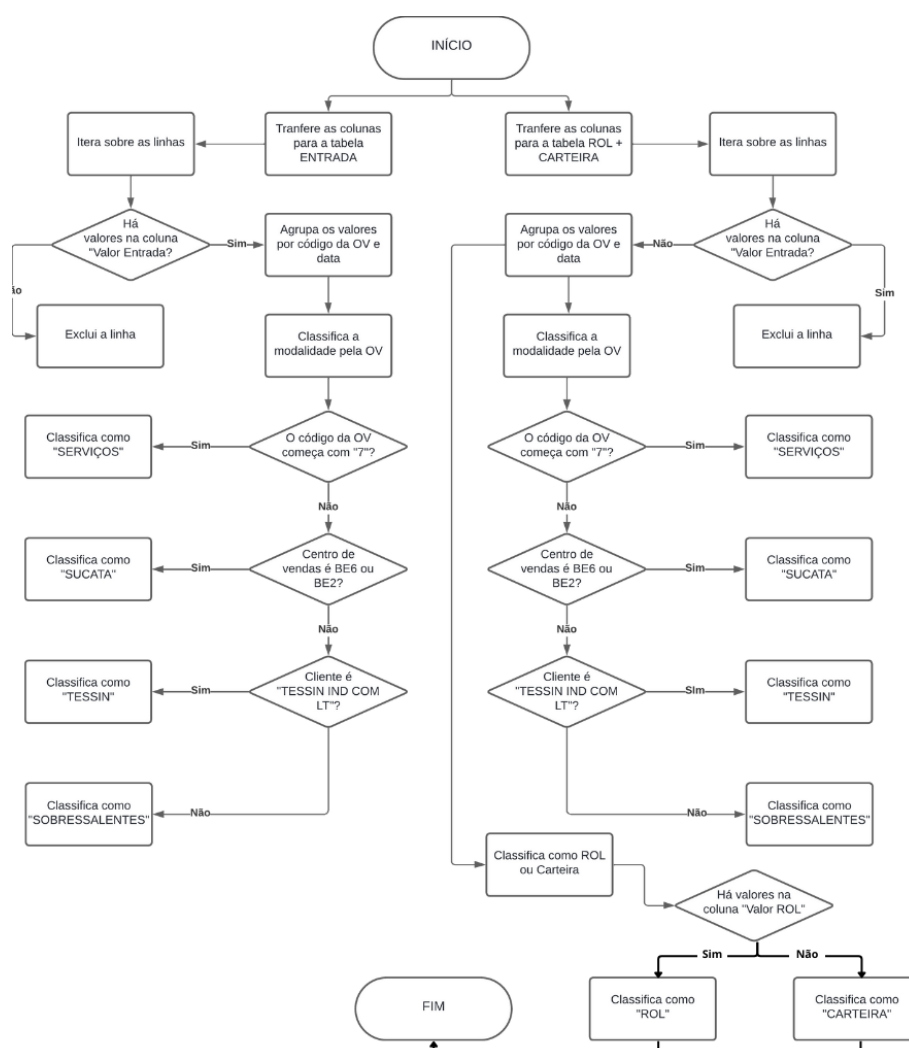
Fonte: O autor

apresenta os valores da carteira que estão em trânsito, isto é, materiais que já saíram da empresa, mas que só serão faturados após o recebimento pelo cliente. Todos esses dados são essenciais para o acompanhamento das metas da célula.

Com a base de dados consolidada, o próximo passo é separar as informações de entrada das relacionadas a ROL e Carteira. Para isso, foi desenvolvido um *script* em VBA que, utilizando macros específicas, manipula os dados da tabela principal e os organiza em duas novas tabelas distintas. Esse processo é detalhado na Figura 11, que apresenta um fluxograma ilustrando o passo a passo para a criação dessas tabelas de forma estruturada.

No fluxograma, é possível observar que a tabela de dados inicial é transferida para duas tabelas distintas: a de entrada e a de ROL/Carteira. Esse processo de transferência é realizado utilizando o comando `Cells(linha, coluna)`, que permite manipular valores diretamente em células específicas, com base na posição de linhas e colunas. Para facilitar a organização e a manipulação, cada tabela é definida como um objeto específico no código, como, por exemplo, a tabela inicial referenciada como "MAIN" e a tabela de entrada como "ENT". O código que implementa essa lógica está apresentado no Quadro 2.

Figura 11 – Fluxograma do processo de manipulação dos dados de entrada, ROL e carteira.



Fonte: O autor

Quadro 2 – Trecho de código VBA para transferência de dados entre tabelas.

```

R = 5
i = 2

Do While MAIN.Cells(R, 1).Value <>

    ENT.Cells(i, 1).Value = MAIN.Cells(R, 4).Value
    ENT.Cells(i, 2).Value = MAIN.Cells(R, 5).Value
    ENT.Cells(i, 3).Value = MAIN.Cells(R, 2).Value
    ENT.Cells(i, 4).Value = MAIN.Cells(R, 7).Value
    ENT.Cells(i, 5).Value = MAIN.Cells(R, 1).Value
    ENT.Cells(i, 6).Value = MAIN.Cells(R, 8).Value

    R = R + 1
    i = i + 1

Loop

```

Fonte: O autor

O código transfere dados de uma tabela inicial "MAIN" para outra tabela "ENT", utilizando um laço de repetição que percorre todas as linhas da tabela de origem enquanto a célula da coluna 1 não estiver vazia. Ele inicializa as variáveis R e i para definir as linhas iniciais de tabela, respectivamente, e em cada iteração copia os valores das colunas indicadas de uma tabela para outra. Após cada transferência, as variáveis de linha são incrementadas, garantindo o progresso no processamento. O laço termina automaticamente ao atingir uma linha vazia na tabela de origem, completando a transferência de dados de forma automatizada. Nas Figuras 12 e 13 estão as duas tabelas formuladas de entrada e de ROL/Carteira.

Figura 12 – Trecho da tabela de dados de entrada.

DATA DE ENTRADA	OV	EQUIPE VENDAS	CLIENTE	CENTRO DE FATURAMENTO	VALOR	TIPO DE VENDA
07/12/2023	70113174	BE3		ITA		3 SERVIÇOS
15/12/2023	70123215	BE3		ITA		3 SERVIÇOS
24/01/2024	52519704	BE3		ITA		2 SOBRESSALENTES
30/01/2024	70096694	BE3		ITA		3 SERVIÇOS
30/01/2024	70103780	BE3		ITA		3 SERVIÇOS
31/01/2024	70103871	BE3		ITA		3 SERVIÇOS
31/01/2024	70114157	BE3		ITA		3 SERVIÇOS
31/01/2024	70120738	BE3		ITA		3 SERVIÇOS
07/02/2024	70096694	BE3		ITA		3 SERVIÇOS
19/02/2024	70101714	BE3		ITA		3 SERVIÇOS
19/02/2024	70103780	BE3		ITA		3 SERVIÇOS
19/02/2024	70103871	BE3		ITA		3 SERVIÇOS
19/02/2024	70109762	BE3		ITA		3 SERVIÇOS
04/03/2024	70096694	BE3		ITA		3 SERVIÇOS
11/03/2024	70096694	BE3		ITA		3 SERVIÇOS
13/03/2024	70096694	BE3		ITA		3 SERVIÇOS
14/03/2024	70096694	BE3		ITA		3 SERVIÇOS
19/03/2024	70114157	BE3		ITA		3 SERVIÇOS
19/03/2024	70120738	BE3		ITA		3 SERVIÇOS
10/06/2024	52519704	BE3		ITA		3 SOBRESSALENTES
11/07/2024	52659812	BE3		ITA		3 SOBRESSALENTES
19/07/2024	70101413	BE3		ITA		7 SERVIÇOS
19/07/2024	70120930	BE3		ITA		3 SERVIÇOS
29/07/2024	52659812	BE3		ITA		3 SOBRESSALENTES
12/08/2024	52659812	BE3		ITA		7 SOBRESSALENTES
03/09/2024	52519704	BE3		ITA		3 SOBRESSALENTES
09/09/2024	70106586	BE3		ITA		3 SERVIÇOS
10/09/2024	52519704	BE3		ITA		3 SOBRESSALENTES
17/09/2024	70120736	BE3		ITA		3 SERVIÇOS
23/10/2024	52816306	BE3		ITA		3 SOBRESSALENTES
24/10/2024	52816306	BE3		ITA		3 SOBRESSALENTES
22/10/2024	52810394	BE6		ITA		3 SUCATA
30/10/2024	52810394	BE6		ITA		3 SUCATA
30/10/2024	52816786	BE6		ITA		3 SUCATA
17/07/2024	70133517	BE7		ITA		3 SERVIÇOS
14/08/2024	70130782	BE1		WTD-DI		4 SERVIÇOS

Fonte: O autor

Figura 13 – Trecho da tabela de dados de ROL/Carteira.

DATA DE ENTRADA	OV	EQUIPE VENDAS	CLIENTE	CENTRO DE FATURAMENTO	VALOR ROL	VALOR CARTEIRA	VALOR EM TRÂNSITO	TIPO DE VENDA	FATURADO
14/03/2023	7011360	BE3		ITA				SERVIÇOS	SIM
05/04/2023	70106586	BE3		ITA				SERVIÇOS	SIM
10/04/2023	70106329	BE3		ITA				SERVIÇOS	SIM
11/04/2023	70113842	BE3		ITA				SERVIÇOS	SIM
15/05/2023	70101575	BE3		ITA				SERVIÇOS	SIM
23/05/2023	70094974	BE3		ITA				SERVIÇOS	SIM
05/07/2023	70106586	BE3		ITA				SERVIÇOS	SIM
13/07/2023	70106586	BE3		ITA				SERVIÇOS	SIM
04/09/2023	70106586	BE3		ITA				SERVIÇOS	SIM
12/10/2023	70106586	BE3		ITA				SERVIÇOS	SIM
13/11/2023	70122256	BE3		ITA				SERVIÇOS	SIM
17/11/2023	70123215	BE3		ITA				SERVIÇOS	SIM
20/11/2023	70121105	BE3		ITA				SERVIÇOS	SIM
21/11/2023	70121105	BE3		ITA				SERVIÇOS	SIM
05/12/2023	70122256	BE3		ITA				SERVIÇOS	SIM
07/12/2023	70113174	BE3		ITA				SERVIÇOS	SIM
15/12/2023	70123215	BE3		ITA				SERVIÇOS	SIM
19/01/2024	70107180	BE3		ITA				SERVIÇOS	SIM
07/02/2024	70096694	BE3		ITA				SERVIÇOS	SIM
04/03/2024	70096694	BE3		ITA				SERVIÇOS	SIM
13/03/2024	70096694	BE3		ITA				SERVIÇOS	SIM
14/03/2024	70096694	BE3		ITA				SERVIÇOS	SIM
01/04/2024	70096694	BE3		ITA				SERVIÇOS	SIM
05/06/2024	70123117	BE3		ITA				SERVIÇOS	SIM
16/08/2024	52659812	BE3		ITA				SOBRESSALENTES	SIM
09/09/2024	70106586	BE3		ITA				SERVIÇOS	SIM
17/09/2024	70120736	BE3		ITA				SERVIÇOS	SIM
25/09/2024	52519704	BE3		ITA				SOBRESSALENTES	SIM
10/11/2024	70120271	BE3		ITA				SERVIÇOS	NÃO
01/02/2025	70115422	BE3		ITA				SERVIÇOS	NÃO
12/05/2025	52816306	BE3		ITA				SOBRESSALENTES	NÃO
01/03/2026	70120118	BE3		ITA				SERVIÇOS	NÃO
30/01/2025	52799963	BE5		ITA				SOBRESSALENTES	NÃO
30/10/2024	52816786	BE6		ITA				SUCATA	NÃO
28/02/2025	70133517	BE7		ITA				SERVIÇOS	NÃO
04/10/2024	70130899	BE1		WTD-DI				SERVIÇOS	SIM

Fonte: O autor

Com as tabelas de entrada e ROL/Carteira devidamente formatadas e finalizadas, conclui-se a coleta dos dados necessários para os indicadores da célula de vendas. A próxima etapa consiste em transformar esses dados em *dashboards* interativas no Power BI, permitindo análises visuais mais eficientes e estratégicas. Contudo, antes de abordar a criação das *dashboards*, será detalhado o processo de coleta de dados dos indicadores da célula de comissionamento.

3.2 INDICADORES DA CÉLULA DE COMISSIONAMENTO

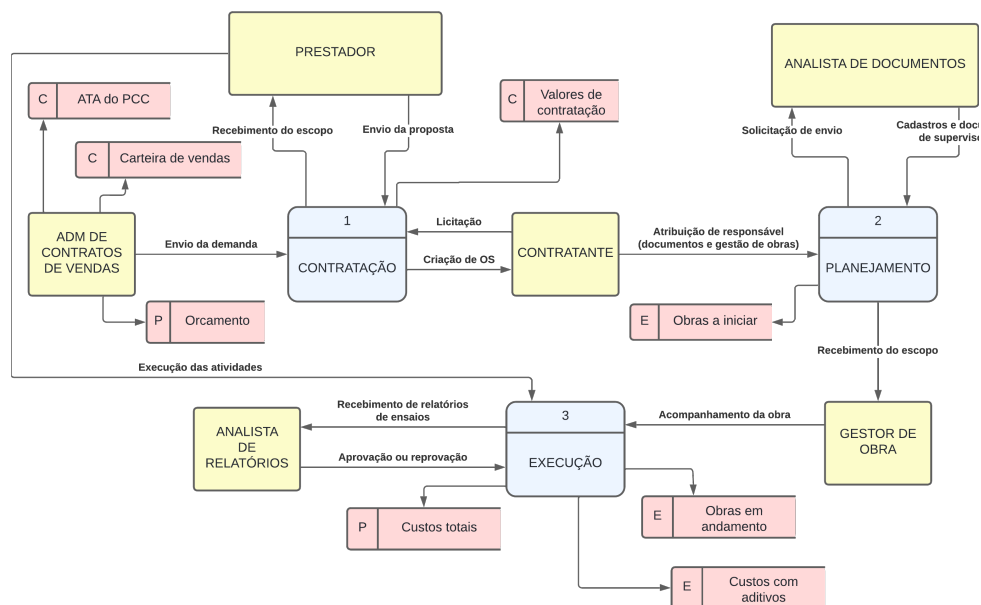
A outra célula da seção, chamada Comissionamento e Start-Up, é responsável por gerenciar todo o processo de montagem, supervisão, comissionamento e outras atividades relacionadas à entrega dos transformadores ou chaves seccionadoras aos clientes. Essa célula desempenha um papel crucial para assegurar que os equipamentos estejam em pleno funcionamento e conforme as especificações exigidas. Para organizar suas funções, a célula é dividida em quatro áreas principais:

- **Contratação:** envolve o recebimento das demandas da seção de contratos e a contratação de equipes e prestadores terceirizados para realização das atividades conforme a demanda recebida.
- **Documentação:** responsável pelo cadastro de supervisores e demais membros das equipes que realizaram as atividades em campo e demais documentações.
- **Gestão de Obras:** equipe que gerencia todas as obras desde a formulação do escopo da atividade, acompanhamento do andamento da obra até que seja finalizada.
- **Análise de relatórios:** responsável pela análise dos relatórios de ensaios feitos nos equipamentos após a realização da atividade em campo. Tem a missão de aprovar ou reprovar os relatórios com base nos dados tabelados de cada ensaio

Das quatro áreas acima, as que são pertinentes para a coleta de dados para os indicadores são a contratação e a gestão de obras. Nelas estão contidas as informações essenciais e cruciais para a análise profissional dos indicadores da célula. Na Figura 14, é apresentado o diagrama de fluxo de dados referente à célula de Comissionamento. Esse diagrama destaca como os dados fluem durante os processos da célula, permitindo identificar os pontos-chave para extração de informações que servirão para a confecção dos indicadores.

É importante observar que os dados extraídos abrangem desde etapas anteriores à contratação, como a ata da seção de PCP (Planejamento e Controle da Produção), até informações pós-encerramento das obras, que são essenciais para a análise das metas e do desempenho da célula. Seguindo a ordem do fluxo do diagrama, o primeiro processo a ser abordado são os indicadores de contratação.

Figura 14 – Diagrama de fluxo de dados da célula de comissionamento.



Fonte: O autor

3.2.1 Indicadores de contratação

A área responsável pela contratação é uma área extremamente importante para a célula de comissionamento. É nessa área que o profissional responsável recebe as demandas vindas da seção de administração de contratos de vendas dos equipamentos. Nessas demandas, estão descritas as principais informações acerca da atividade, como quantidade de equipamentos, potência, tensão, localidade, prazo, entre outras. Dessa maneira, o contratante tenta, da melhor forma, distribuir essas atividades para a equipe de obras. A equipe de obras, por sua vez, monta o escopo da atividade e retorna para o contratante iniciar as tratativas com os prestadores em processo licitatório.

É nesse momento que o contratante abre uma OS (Ordem de serviço) no sistema e insere nela todas as informações pertinentes para cada atividade. Após a abertura da OS, o contratante espera o recebimento das cotações e, ao final do processo, decreta a vencedora pelo menor custo e, assim, atribui o custo na OS com uma requisição de compra.

Entretanto, uma das principais preocupações da área de contratação sempre foi a falta de planejamento adequado por parte da equipe de contratos. Frequentemente, as demandas eram recebidas em prazos muito curtos para a realização das atividades necessárias, o que forçava o contratante a investir esforços para atender aos requisitos dentro do tempo estipulado. Essa situação resultava em custos adicionais significativos, como despesas com equipes extras, diárias, maquinário e outros recursos, que poderiam ter sido evitados com um planejamento mais antecipado.

Pensando nisso, foi feita uma análise de duas bases de dados existentes que poderiam auxiliar a encontrar possíveis demandas a curto, médio e até longo prazo, sem precisar esperar o administrador de contratos enviar a demanda. Essas bases tratam da ata do

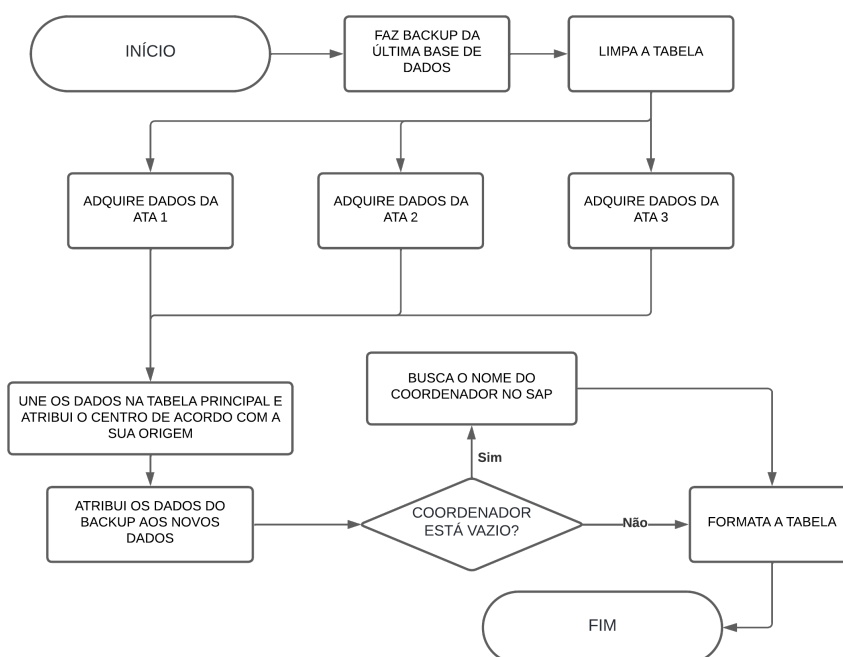
PCP e da carteira de vendas.

3.2.1.1 Controle da ata do PCP

A ata do PCP nada mais é do que uma planilha atualizada semanalmente pelas equipes do PCP das diferentes unidades da empresa. Nela estão contidas as informações dos equipamentos que estão para serem embarcados para o cliente final e um curto e médio prazo (cerca de 4 meses para frente). No entanto, essas planilhas são extremamente mal formatadas, algo que dificulta e muito a obtenção eficiente dos dados.

Para resolver isso, foi analisada a estrutura de cada uma das planilhas com o intuito de encontrar padrões que tornassem possível a manipulação dos dados para uma tabela mais organizada e estruturada. Nesse sentido, foi elaborado um *script* em VBA que manipula cada uma das planilhas de acordo com a sua estrutura. Com esse código, foi possível adquirir os dados mais relevantes para a análise, como cliente, tensão, potência, código PEP (Projeto Elétrico de Particulares) e, principalmente, a data de previsão de embarque do equipamento. A Figura 15 apresenta o fluxograma do processo de aquisição dos dados das diferentes planilhas.

Figura 15 – Fluxograma do processo de aquisição de dados das atas do PCP.



Fonte: O autor

Basicamente, o programa passa pelas três planilhas das atas e adquire os dados pertinentes, lavando-os a uma nova tabela onde esses dados são classificados conforme a sua origem. Caso apareça um caso novo que não constava na ata anterior, o programa vai até a janela do SAP, entra na transação *CJ20N*, insere o código PEP e obtém o nome do coordenador daquela venda. Na Figura 16 é visto um trecho da tabela final.

Figura 16 – Trecho da tabela com os dados das atas do PCP.

CLIENTE	POTÊNCIA (MVA)	CLASSE DE TENSÃO (kV)	PEP	PREVISÃO EMBARQUE	RESPONSÁVEL	CENTRO
150	500.00	150-2300177-7C		24/09		BLUMENAU
200	230.00	150-2300216-44N		19/09		BLUMENAU
150	900.00	150-2300177-5C		19/09		BLUMENAU
150	500.00	150-2300177-2C		23/09		BLUMENAU
150	500.00	150-2300177-3C		26/09		BLUMENAU
75	345.00	150-2300345-2C		19/09		BLUMENAU
75	345.00	150-2300345-3C		24/09		BLUMENAU
62.5	220.00	150-2300207-1		27/09		BLUMENAU
100	550.00	150-2300348-1C		23/09		BLUMENAU
100	550.00	150-2300348-2C		25/09		BLUMENAU
100	550.00	150-2300348-3C		26/09		BLUMENAU
50	220.00	150-2300054-5		27/09		BLUMENAU
500	400.00	150-2300373-12		30/09		BLUMENAU
225	550.00					BLUMENAU
80	550.00	150-2300365-1C	05/11			BLUMENAU
80	550.00	150-2300365-2C	05/11			BLUMENAU
80	550.00	150-2300365-3C	05/11			BLUMENAU
80	550.00	150-2300365-4C	05/11			BLUMENAU
45.33	500.00	150-2300177-18C	15/10			BLUMENAU
45.33	500.00	150-2300177-19C	15/10			BLUMENAU
45.33	500.00	150-2300177-21C	15/10			BLUMENAU
185	230.00	150-2300479-11111N	22/10			BLUMENAU
180	230.00	150-2400814-1C	18/10			BLUMENAU
45.33	500.00	150-2300177-22C	15/10			BLUMENAU
500	400.00	150-2300375-12				BLUMENAU
75	345.00	150-2300345-4C	13/11			BLUMENAU
10	230.00	150-2300177-28C	28/10			BLUMENAU
10	230.00	150-2300177-29C	28/10			BLUMENAU
33.33			30/10			BLUMENAU
200	230.00	150-2300216-42N				BLUMENAU
10	230.00	150-2300360-1	05/11			BLUMENAU
50	500.00	150-2300432-1C	29/10			BLUMENAU
500	400.00	150-2300375-22				BLUMENAU
45.33	500.00	150-2300177-23C				BLUMENAU
50	500.00	150-2300432-2C				BLUMENAU
100	525.00	150-2300378-3C	12/11			BLUMENAU

Fonte: O autor

Na tabela estão descritos os dados obtidos das atas, como cliente, potência, tensão, PEP, data de embarque, coordenador e cidade onde o equipamento está sendo produzido.

3.2.1.2 Controle da carteira de vendas

Diferentemente da ata do PCP, que fornece uma visão imediata, a carteira de vendas oferece uma perspectiva de longo prazo sobre os equipamentos com previsão de saída de fábrica. Essa previsão pode abranger períodos extensos, chegando a até 6 anos no futuro. Assim como a carteira de vendas de sobressalentes, esses dados são armazenados em uma base de dados interna e podem ser facilmente exportados como planilha no formato XLSX. A Figura 17 evidencia como é o formato dessa planilha exportada do banco de dados interno.

Figura 17 – Formatação dos dados da carteira de vendas.

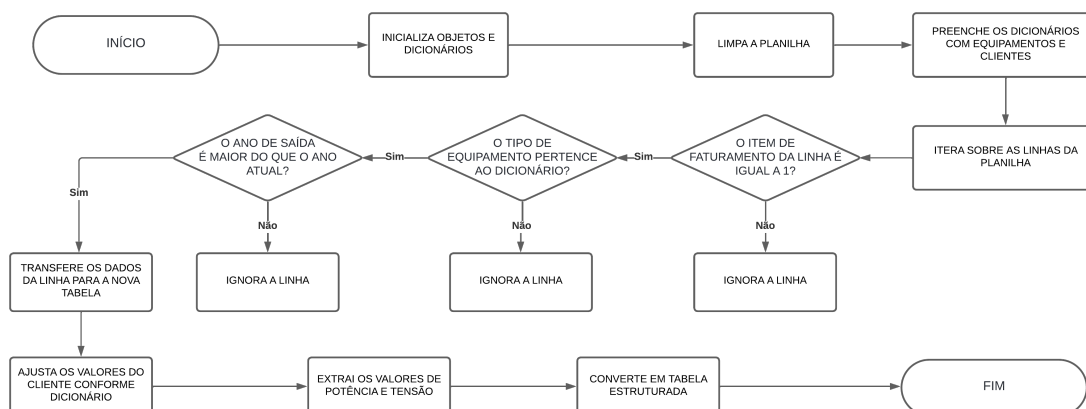
Detalhes - Carteira de Pedidos								
Local de Reparo	Coordenador Vendas	Cliente	MATERIAL	Nº do Produto 2	Data Saída Merc.	Cidade do Cliente	Estado do Cliente	Fala do Cliente
BETM			10289624 TRANSFORMADOR O 150 000VA 50CST3BVCU	TRANSFORMADORES A ÓLEO FORÇA	10.04.2023	PAULO AFONSO	Bahia	BR Brasil
BETM			10289640 AUTOTRANSFORMADOR O 150 0667kVA 52CST5T4	TRANSFORMADORES A ÓLEO FORÇA	14.03.2023	BELO HORIZONTE	Minas Gerais	BR Brasil
BETM			10289640 AUTOTRANSFORMADOR O 150 0667kVA 52CST5T4	TRANSFORMADORES A ÓLEO FORÇA	14.03.2023	BELO HORIZONTE	Minas Gerais	BR Brasil
BETM			10289640 AUTOTRANSFORMADOR O 150 0667kVA 52CST5T4	TRANSFORMADORES A ÓLEO FORÇA	14.03.2023	BELO HORIZONTE	Minas Gerais	BR Brasil
BETM			10289640 AUTOTRANSFORMADOR O 150 0667kVA 52CST5T4	TRANSFORMADORES A ÓLEO FORÇA	14.03.2023	BELO HORIZONTE	Minas Gerais	BR Brasil
BETM			10289640 AUTOTRANSFORMADOR O 150 0667kVA 52CST5T4	TRANSFORMADORES A ÓLEO FORÇA	27.04.2023	BELO HORIZONTE	Minas Gerais	BR Brasil
BETM			10289640 AUTOTRANSFORMADOR O 150 0667kVA 52CST5T4	TRANSFORMADORES A ÓLEO FORÇA	27.04.2023	BELO HORIZONTE	Minas Gerais	BR Brasil
BETM			10289640 AUTOTRANSFORMADOR O 150 0667kVA 52CST5T4	TRANSFORMADORES A ÓLEO FORÇA	27.04.2023	BELO HORIZONTE	Minas Gerais	BR Brasil
BETM			10289640 AUTOTRANSFORMADOR O 150 0667kVA 52CST5T4	TRANSFORMADORES A ÓLEO FORÇA	27.04.2023	BELO HORIZONTE	Minas Gerais	BR Brasil
BETM			10289640 AUTOTRANSFORMADOR O 150 0667kVA 52CST5T4	TRANSFORMADORES A ÓLEO FORÇA	27.04.2023	BELO HORIZONTE	Minas Gerais	BR Brasil
BETM			10289640 AUTOTRANSFORMADOR O 150 0667kVA 52CST5T4	TRANSFORMADORES A ÓLEO FORÇA	27.04.2023	BELO HORIZONTE	Minas Gerais	BR Brasil
BETM			10289640 AUTOTRANSFORMADOR O 150 0667kVA 52CST5T4	TRANSFORMADORES A ÓLEO FORÇA	27.04.2023	BELO HORIZONTE	Minas Gerais	BR Brasil
BETM			10289640 AUTOTRANSFORMADOR O 150 0667kVA 52CST5T4	TRANSFORMADORES A ÓLEO FORÇA	27.04.2023	BELO HORIZONTE	Minas Gerais	BR Brasil
BETM			10289640 AUTOTRANSFORMADOR O 150 0667kVA 52CST5T4	TRANSFORMADORES A ÓLEO FORÇA	27.04.2023	BELO HORIZONTE	Minas Gerais	BR Brasil
BETM			10289640 AUTOTRANSFORMADOR O 150 0667kVA 52CST5T4	TRANSFORMADORES A ÓLEO FORÇA	27.04.2023	BELO HORIZONTE	Minas Gerais	BR Brasil
BETM			10289640 AUTOTRANSFORMADOR O 150 0667kVA 52CST5T4	TRANSFORMADORES A ÓLEO FORÇA	21.09.2023	BELO HORIZONTE	Minas Gerais	BR Brasil
BETM			10289640 AUTOTRANSFORMADOR O 150 0667kVA 52CST5T4	TRANSFORMADORES A ÓLEO FORÇA	21.09.2023	BELO HORIZONTE	Minas Gerais	BR Brasil
BETM			10289640 AUTOTRANSFORMADOR O 150 0667kVA 52CST5T4	TRANSFORMADORES A ÓLEO FORÇA	21.09.2023	BELO HORIZONTE	Minas Gerais	BR Brasil
BETM			10289640 AUTOTRANSFORMADOR O 150 0667kVA 52CST5T4	TRANSFORMADORES A ÓLEO FORÇA	17.11.2023	BELO HORIZONTE	Minas Gerais	BR Brasil

Fonte: O autor

A planilha apresenta diversas colunas relevantes para a análise de contratações, incluindo informações como cliente, tipo de equipamento, classe de tensão, localidade, coordenador responsável e a previsão de saída do equipamento. A análise integrada desses dados proporciona uma visão estratégica de longo prazo sobre as demandas futuras, permitindo um planejamento mais eficiente. Com isso, é possível antecipar necessidades, como o cadastro de novos prestadores, a capacitação de profissionais e a alocação de recursos, assegurando que a célula esteja preparada para atender às demandas com maior eficiência e controle.

Para filtrar apenas os equipamentos relevantes e com a data de saída apenas a partir do próximo ano, foi elaborado um código VBA que manipula os dados dessa planilha para uma tabela separada apenas com as linhas e colunas pertinentes. A seguir, na Figura 18 está evidenciado o fluxograma desse processo de manipulação de dados através das macros em VBA.

Figura 18 – Fluxograma do processo de aquisição de dados da carteira de vendas de equipamentos.



Fonte: O autor

Nesse código, representado pelo fluxograma, utilizou-se de dicionários que, em VBA, são objetos obtidos a partir da biblioteca **Scripting.Dictionary**. Eles são utilizados para armazenar e gerenciar pares de chave-valor. Isso é extremamente útil quando é preciso acessar dados de forma rápida com base em uma chave específica, sem necessidade de ficar repetindo várias vezes as condicionais **if/elseif/else**. No Quadro 3 a seguir está um trecho do código que mostra a utilização dessa ferramenta. Nota-se que somente se o tipo de equipamento existir no dicionário, as informações referentes àquela linha são transferidas da planilha de origem (CART) para a planilha de destino (aba). Essa transferência garante que apenas os registros válidos sejam incluídos, aplicando filtros baseados em critérios como o ano de saída.

Quadro 3 – Uso de dicionário para verificar tipos de equipamentos relevantes.

```

Set tipos = CreateObject("Scripting.Dictionary")

tipos.Add "REATORES A ÓLEO FORÇA", True
tipos.Add "REATORES A ÓLEO MÉDIA FORÇA", True
tipos.Add "SECCIONADORES", True
tipos.Add "SUBESTAÇÕES MÓVEIS", True
tipos.Add "SUBESTAÇÕES MÓVEIS(SOMENTE TRANSFORMADOR)", True
tipos.Add "TRANSFORMADORES A ÓLEO FORÇA", True
tipos.Add "TRANSFORMADORES A ÓLEO MEDIA FORÇA I", True
tipos.Add "TRANSFORMADORES A ÓLEO MEDIA FORÇA II", True
tipos.Add "TRANSFORMADORES A ÓLEO MEIA FORÇA", True
tipos.Add "TRANSFORMADORES SECO", True

If tipos.Exists(CART.Cells(i, 8).Value) Then

    aba.Cells(j, 1).Value = Replace(CART.Cells(i, 10).Value, ".", "/")
    aba.Cells(j, 2).Value = CART.Cells(i, 6).Value
    aba.Cells(j, 3).Value = CART.Cells(i, 8).Value
    aba.Cells(j, 4).Value = CART.Cells(i, 7).Value
    aba.Cells(j, 8).Value = CART.Cells(i, 4).Value
    aba.Cells(j, 10).Value = CART.Cells(i, 15).Value

End If

```

Fonte: O autor.

Além disso, o código organiza os dados em colunas específicas, ajustando formatos e preenchendo informações como cliente, centro de vendas, potência e tensão. Esse processo resulta na criação de uma tabela estruturada, formatada automaticamente pelo código. A tabela final, apresentada na Figura 19, exibe as informações coletadas e formatadas, permitindo que sejam utilizadas diretamente para o planejamento futuro da célula.

Figura 19 – Trecho da tabela com os dados da carteira de vendas.

DATA SAÍDA	MATERIAL	TIPO	POTÊNCIA	TENSÃO	CLIENTE	COORDENADOR	CENTRO	PAÍS
31/01/2025	16370213	SUBESTAÇÕES MÓVEIS(SOMENTE TRANSFORMADOR					BETIM	Brasil
09/04/2025	16370213	SUBESTAÇÕES MÓVEIS(SOMENTE TRANSFORMADOR					BETIM	Brasil
11/05/2025	16370213	SUBESTAÇÕES MÓVEIS(SOMENTE TRANSFORMADOR					BETIM	Brasil
12/03/2025	16370213	SUBESTAÇÕES MÓVEIS(SOMENTE TRANSFORMADOR					BETIM	Brasil
30/12/2025	16370213	SUBESTAÇÕES MÓVEIS(SOMENTE TRANSFORMADOR					BETIM	Brasil
05/09/2025	17366424	TRANSFORMADORES A ÓLEO FORÇA	51	230			BETIM	Brasil
13/05/2025	17366424	TRANSFORMADORES A ÓLEO FORÇA	51	230			BETIM	Brasil
01/07/2025	17375335	TRANSFORMADORES A ÓLEO FORÇA	250	500			BETIM	Brasil
01/07/2025	17375335	TRANSFORMADORES A ÓLEO FORÇA	250	500			BETIM	Brasil
01/08/2025	17375335	TRANSFORMADORES A ÓLEO FORÇA	250	500			BETIM	Brasil
17/03/2025	17404638	REATORES A ÓLEO FORÇA	66,556	550			BETIM	Brasil
20/06/2025	17375474	TRANSFORMADORES A ÓLEO FORÇA	200	500			BETIM	Brasil
23/06/2025	17375474	TRANSFORMADORES A ÓLEO FORÇA	200	500			BETIM	Brasil
23/06/2025	17375474	TRANSFORMADORES A ÓLEO FORÇA	200	500			BETIM	Brasil
27/06/2025	17375474	TRANSFORMADORES A ÓLEO FORÇA	200	500			BETIM	Brasil
14/01/2025	17404638	REATORES A ÓLEO FORÇA	55,1	550			BETIM	Brasil
14/01/2025	17404638	REATORES A ÓLEO FORÇA	55,1	550			BETIM	Brasil
14/01/2025	17404638	REATORES A ÓLEO FORÇA	55,1	550			BETIM	Brasil
14/01/2025	17404638	REATORES A ÓLEO FORÇA	55,1	550			BETIM	Brasil
14/02/2025	17404638	REATORES A ÓLEO FORÇA	55,1	550			BETIM	Brasil
14/03/2025	17404638	REATORES A ÓLEO FORÇA	55,1	550			BETIM	Brasil
14/03/2025	17404638	REATORES A ÓLEO FORÇA	55,1	550			BETIM	Brasil
14/03/2025	17404643	TRANSFORMADORES A ÓLEO FORÇA	133,333	500			BETIM	Brasil
14/03/2025	17404644	REATORES A ÓLEO FORÇA	55,1	550			BETIM	Brasil
30/06/2026	17707661	SUBESTAÇÕES MÓVEIS(SOMENTE TRANSFORMADOR					BETIM	Brasil
30/06/2026	17707661	SUBESTAÇÕES MÓVEIS(SOMENTE TRANSFORMADOR					BETIM	Brasil
30/06/2026	17707661	SUBESTAÇÕES MÓVEIS(SOMENTE TRANSFORMADOR					BETIM	Brasil
30/06/2026	17707661	SUBESTAÇÕES MÓVEIS(SOMENTE TRANSFORMADOR					BETIM	Brasil
30/06/2026	17707661	SUBESTAÇÕES MÓVEIS(SOMENTE TRANSFORMADOR					BETIM	Brasil
30/06/2026	17707661	SUBESTAÇÕES MÓVEIS(SOMENTE TRANSFORMADOR					BETIM	Brasil
13/02/2025	17691285	TRANSFORMADORES A ÓLEO FORÇA	380	500			BETIM	Brasil
13/02/2025	17691285	TRANSFORMADORES A ÓLEO FORÇA	380	500			BETIM	Brasil
04/10/2025	17691285	TRANSFORMADORES A ÓLEO FORÇA	380	500			BETIM	Brasil

Fonte: O autor

3.2.2 Indicadores de obras em execução

Ao finalizar o processo de contratação da equipe responsável por realizar as atividades determinadas no escopo, o andamento do processo fica a cargo dos coordenadores de obras. Eles têm como função principal acompanhar por completo o planejamento e execução de cada obra. Ao receberem o processo do contratante, eles arrastam a pasta de e-mail relativa àquela obra para uma pasta superior chamada "A INICIAR". Durante esse período, o coordenador acompanha o processo de chegada do equipamento em campo até o momento em que esteja tudo mobilizado para que a obra seja iniciada.

Assim que a obra é iniciada, o coordenador move o processo para uma pasta denominada "EM ANDAMENTO". Esse estágio é considerado o mais crítico, pois qualquer imprevisto durante a execução exige uma resposta rápida e eficiente do coordenador para amenizar os impactos e encontrar soluções. Tais imprevistos, frequentemente, resultam em custos adicionais não previstos no planejamento inicial. Esses custos, em sua maioria, estão relacionados a diárias de carretas de óleo em campo, mobilização de equipes extras e diárias adicionais de supervisores.

Um grande desafio para a célula como um todo era conseguir trazer esse controle de quanto era gasto com aditivos em obras em andamento, principalmente para os superiores, que só recebiam a notícia da presença de aditivos nessas obras quando chegava a solicitação de aprovação da compra ao final do processo. Para solucionar esse ponto, foi desenvolvido um programa executável em que os coordenadores, assim que notavam a necessidade de incluir custos em aditivos, cadastrassem esse custo que apareceria automaticamente em uma *dashboard* para o acompanhamento dos superiores.

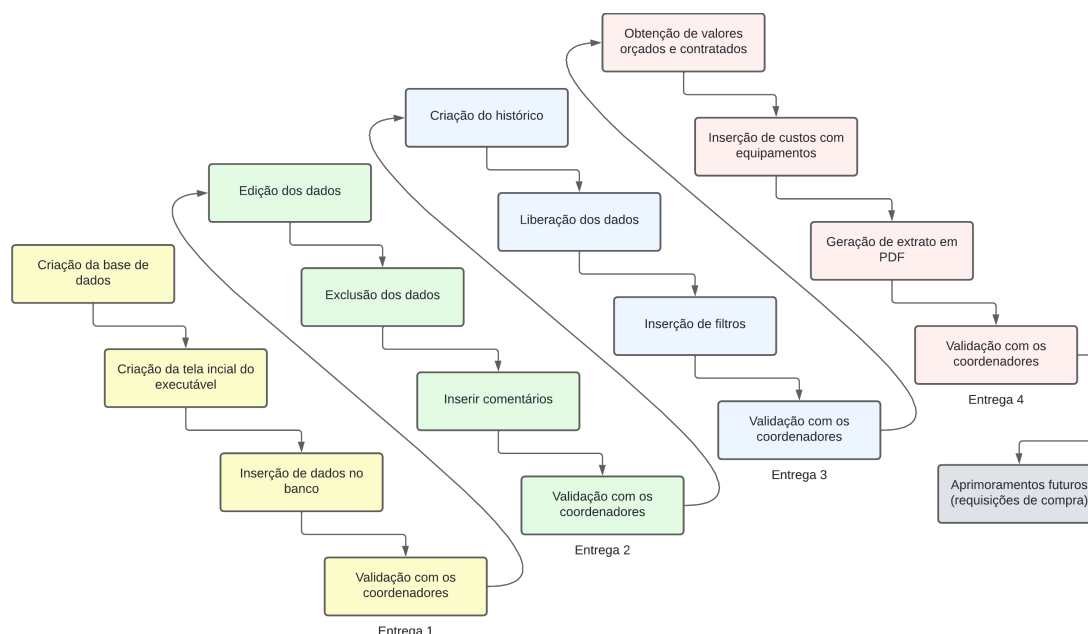
3.2.2.1 Aplicativo executável para cadastro de aditivos

Para a construção do executável, foi utilizada a linguagem Python, combinada com bibliotecas especializadas para atender tanto às necessidades do *front-end* quanto à manipulação de bases de dados. No *front-end*, foram empregadas as bibliotecas Tkinter e CustomTkinter, enquanto, no processamento de dados, SQLite3 e Pandas. O desenvolvimento e os testes do código foram realizados no Visual Studio Code, software da Microsoft.

O desenvolvimento do executável seguiu a metodologia incremental, permitindo que cada etapa concluída fosse testada diretamente pelos coordenadores de obras para validar o funcionamento da aplicação. Esse processo assegurava que o executável permanecesse funcional até a etapa implementada, enquanto novas atualizações eram desenvolvidas. Ao final de cada ciclo de aprimoramento, uma nova versão do aplicativo era disponibilizada, permitindo que os coordenadores utilizassem a aplicação com as funcionalidades mais recentes. Essa abordagem foi possível graças ao armazenamento contínuo da base de dados, garantindo que nenhuma informação fosse perdida durante as atualizações. Na

Figura 20 é evidenciado como se deu essa implementação de maneira incremental.

Figura 20 – Desenvolvimento incremental do executável de aditivos.



Fonte: O autor

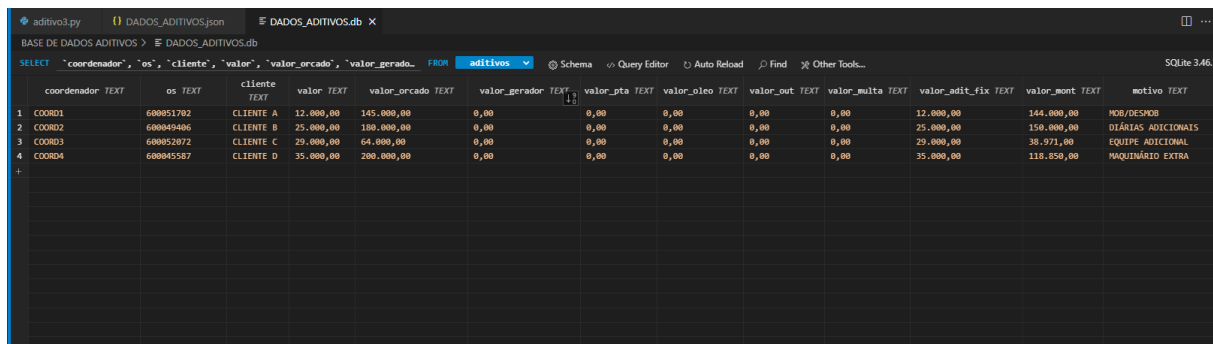
Observa-se que a implementação se deu em quatro diferentes entregas. Ao final de cada etapa, os coordenadores já estavam aptos para inserir os aditivos no executável e, a cada nova atualização, novas funcionalidades eram disponibilizadas, facilitando o controle e usabilidade do aplicativo. A partir disso, será apresentado agora como ocorreu a implementação de cada uma dessas etapas.

3.2.2.1.1 Entrega 1: criação da base de dados, inserção de dados e tela inicial

A primeira etapa a ser implementada consiste em três processos que, em conjunto, viabilizam um controle inicial dos aditivos. O primeiro processo foi a criação da base de dados, utilizando duas bibliotecas amplamente eficazes para esse tipo de aplicação. A biblioteca *SQLite3* foi empregada para gerenciar um arquivo do tipo DB (Database), armazenado diretamente na rede do sistema, possibilitando a execução de operações na linguagem DML, amplamente utilizada em bancos de dados do tipo SQL. Além disso, a biblioteca *Pandas* foi integrada ao processo, fornecendo estruturas de dados como *DataFrames*, que permitem a leitura, transformação e visualização de grandes volumes de informações de forma eficiente. Na Figura 21 está um trecho retirado de uma extensão do VSCode que possibilita a visualização e manipulação de arquivos DB. Na imagem, é possível visualizar as colunas criadas com a inserção de dados genéricos. Nota-se que nessa imagem já estão todas as colunas inseridas ao longo de todas as entregas. Para a criação dessas colunas, utilizaram-se alguns comandos que fazem parte das bibliotecas

citadas anteriormente. No Quadro 4 é visto como se dá a implementação para a criação do arquivo DB.

Figura 21 – Visualização dos dados do arquivo DB.



	coordenador TEXT	os TEXT	cliente TEXT	valor TEXT	valor_orcado TEXT	valor_gerador TEXT	valor_pta TEXT	valor_oleo TEXT	valor_out TEXT	valor_multa TEXT	valor_adit_fix TEXT	valor_mont TEXT	motivo TEXT
1	COORD1	600051702	CLIENTE A	12.000,00	145.000,00	0,00	0,00	0,00	0,00	0,00	12.000,00	144.000,00	NOS/DESPOR
2	COORD2	600049406	CLIENTE B	25.000,00	180.000,00	0,00	0,00	0,00	0,00	0,00	25.000,00	150.000,00	DIÁRIAS ADICIONAIS
3	COORD3	600052072	CLIENTE C	25.000,00	64.000,00	0,00	0,00	0,00	0,00	0,00	25.000,00	38.971,00	EQUIPE ADICIONAL
4	COORD4	600045587	CLIENTE D	35.000,00	200.000,00	0,00	0,00	0,00	0,00	0,00	35.000,00	118.850,00	MAQUINÁRIO EXTRA

Fonte: O autor

Quadro 4 – Código para criação da base de dados com *SQLite3*.

```
conn = sqlite3.connect(caminho_arquivoDB)
c = conn.cursor()

c.execute("""CREATE TABLE IF NOT EXISTS aditivos
(id INTEGER PRIMARY KEY AUTOINCREMENT,
coordenador TEXT,
os TEXT,
valor REAL,
motivo TEXT,
data_alteracao TEXT,
"')

conn.commit()
conn.close()
```

Fonte: O autor.

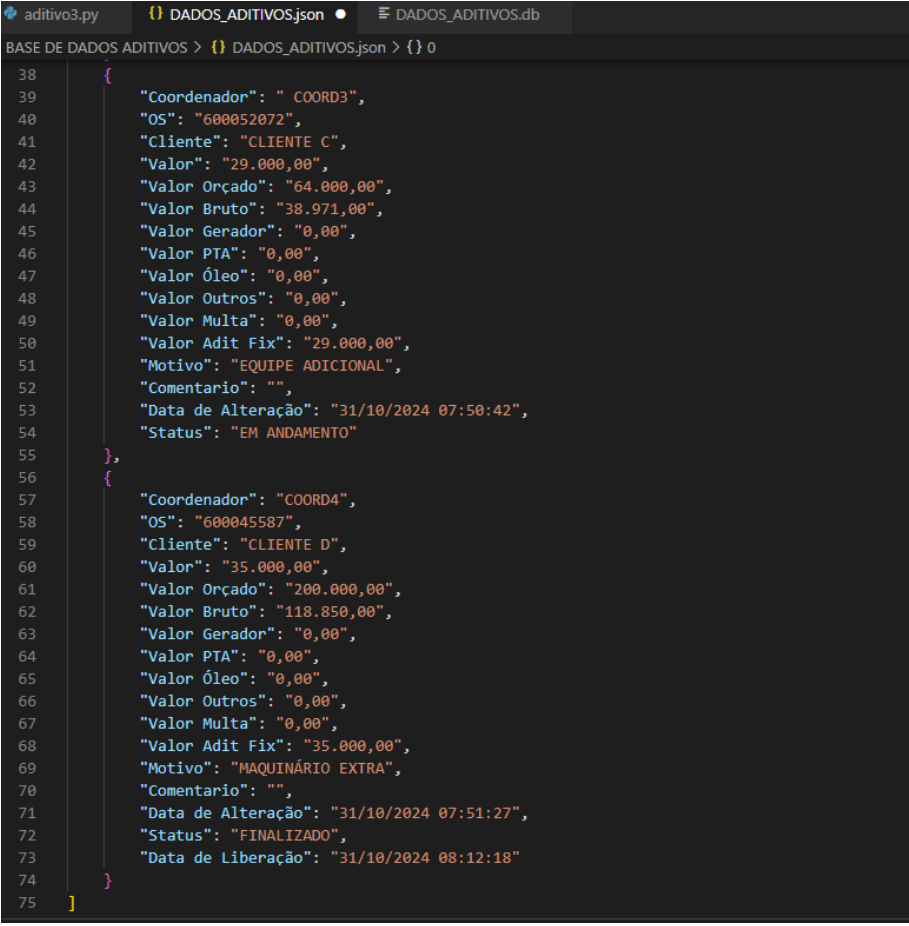
Primeiramente, o código inicia estabelecendo uma conexão com o banco de dados utilizando o comando `sqlite3.connect()`. Caso o arquivo não exista, o SQLite criará automaticamente um novo banco de dados. A conexão criada é armazenada na variável `conn`, que será utilizada para realizar todas as operações subsequentes no banco de dados. Em seguida, um cursor é criado com o comando `conn.cursor()`. O cursor é uma interface que permite a execução de comandos DML no banco de dados. Ele serve como uma conexão entre o Python e o banco de dados.

O próximo passo é a criação da tabela de dados. Para isso, utiliza-se o comando `CREATE TABLE IF NOT EXISTS`. Esse comando garante que a tabela seja criada apenas se ela ainda não existir no banco de dados. A estrutura da tabela é definida com as colunas `id` (chave primária), `coordenador`, `OS`, `valor`, `motivo` e `data`. Depois que a estrutura da

tabela foi definida, o comando `conn.commit()` é executado. Esse comando é utilizado para confirmar e salvar as alterações feitas no banco de dados. Por fim, a conexão com o banco de dados é encerrada com o comando `conn.close()`.

Com a criação do arquivo DB, o próximo passo foi gerar um novo arquivo no formato JSON. Isso foi necessário porque, embora o Power BI não ofereça suporte direto a arquivos no formato DB, ele permite a importação de arquivos JSON. Assim, foi desenvolvida uma função para extrair os dados da base de dados SQLite e convertê-los para o formato JSON. Na Figura 22, é apresentado um trecho dos dados armazenados na base de dados, já convertidos para o formato JSON. Esses dados também estão compostos por todas as colunas criadas ao longo de toda a implementação.

Figura 22 – Visualização dos dados do arquivo JSON.



```
38 {
39   "Coordenador": "COORD3",
40   "OS": "600052072",
41   "Cliente": "CLIENTE C",
42   "Valor": "29.000,00",
43   "Valor Orçado": "64.000,00",
44   "Valor Bruto": "38.971,00",
45   "Valor Gerador": "0,00",
46   "Valor PTA": "0,00",
47   "Valor Óleo": "0,00",
48   "Valor Outros": "0,00",
49   "Valor Multa": "0,00",
50   "Valor Adit Fix": "29.000,00",
51   "Motivo": "EQUIPE ADICIONAL",
52   "Comentario": "",
53   "Data de Alteração": "31/10/2024 07:50:42",
54   "Status": "EM ANDAMENTO"
55 },
56 {
57   "Coordenador": "COORD4",
58   "OS": "600045587",
59   "Cliente": "CLIENTE D",
60   "Valor": "35.000,00",
61   "Valor Orçado": "200.000,00",
62   "Valor Bruto": "118.850,00",
63   "Valor Gerador": "0,00",
64   "Valor PTA": "0,00",
65   "Valor Óleo": "0,00",
66   "Valor Outros": "0,00",
67   "Valor Multa": "0,00",
68   "Valor Adit Fix": "35.000,00",
69   "Motivo": "MAQUINÁRIO EXTRA",
70   "Comentario": "",
71   "Data de Alteração": "31/10/2024 07:51:27",
72   "Status": "FINALIZADO",
73   "Data de Liberação": "31/10/2024 08:12:18"
74 }
75 ]
```

Fonte: O autor

Com a programação para criação dos arquivos de dados concluída, agora é necessário criar a interface inicial do programa executável. Para isso, utilizou-se da vasta gama de artifícios disponíveis nas bibliotecas *Tkinter* e, principalmente, *CustomTkinter*. Entre os principais elementos utilizados na criação da interface visual estão:

- **Labels (CTkLabel):** exibem textos estáticos.
- **Entry (CTkEntry):** campos de entrada para digitação de texto.

- **Combobox (CTkComboBox)**: menu suspenso que permite selecionar uma opção pré-definida.
- **Botões (CTkButton)**: botões interativos para executar comandos.
- **Frames (CTkFrame)**: caixas para organizar os elementos e criar seções na interface.

A partir desses elementos, foi desenvolvida a primeira versão da interface gráfica do executável, apresentada na Figura 23. À esquerda, encontra-se o *Frame* destinado à inserção de dados, composto por elementos como *Entry* e *ComboBox*.

Figura 23 – Janela inicial do executável.

COORDENADOR	OS	VALOR ORÇADO (R\$)	VALOR MONT. (R\$)	VALOR ADITIVO (R\$)	MOTIVO
COORD1	600051702	145.000,00	144.000,00	12.000,00	MGB/DESMOB
COORD2	600049406	180.000,00	150.000,00	25.000,00	DIÁRIAS ADICIONAIS
COORD3	600052072	64.000,00	38.971,00	29.000,00	EQUIPE ADICIONAL

Fonte: O autor

Esses componentes permitem ao usuário fornecer os dados necessários que serão registrados na base de dados e posteriormente exibidos na tabela, localizada à direita. Para realizar a inserção dos dados, o usuário deve preencher os campos de coordenador, OS, valor e motivo e, em seguida, clicar no botão "INSERIR". O ato de clicar no botão faz com que seja chamada a função de inserção de dados.

Nessa função, os dados presentes nos campos são extraídos com o comando `get()` e salvos em variáveis locais. Essas variáveis são utilizadas no comando "INSERT" que faz com que seja criada uma nova linha de dados no arquivo DB. O código que representa essa ação é visto no Quadro 5.

O código utiliza a biblioteca `datetime` para capturar a data no momento da inserção. Esse valor será armazenado no campo "data" da tabela, indicando quando a entrada foi criada. O próximo passo é a execução do comando SQL para inserir dados na tabela aditivos. A instrução `INSERT INTO` é utilizada para preencher os campos da tabela, referente ao coordenador, OS, valor, motivo e data. Ao final, a função `salvar_json()` é chamada para exportar os dados atualizados do banco para o arquivo JSON. Ao final dessa implementação, a primeira etapa de criação do executável foi finalizada.

Quadro 5 – Código para inserção de dados na tabela de aditivos.

```
conn = sqlite3.connect(caminho_arquivoDB)
c = conn.cursor()

data_atual = datetime.now().strftime("%d/%m/%Y")

c.execute("INSERT INTO aditivos (coordenador, os, valor, motivo, data)
VALUES (?, ?, ?, ?, ?)",
(coordenador, os, valor, motivo, data_atual))

conn.commit()
conn.close()

salvar_json()
```

Fonte: O autor.

3.2.2.1.2 Entrega 2: edição e exclusão dos dados e inserção de comentários

Na segunda etapa, o objetivo principal foi a implementação de funções que pudessem manipular os dados já existentes no arquivo de dados. Para isso, foram elaboradas duas funções distintas, uma para a edição de dados e outra para a exclusão de dados.

A função de edição de dados permite ao usuário alterar qualquer informação de uma linha específica na tabela. Para habilitar a janela de edição, o usuário dispõe de duas opções. A primeira consiste em clicar duas vezes com o botão esquerdo do mouse diretamente sobre a linha que deseja editar. A segunda opção envolve selecionar a linha desejada clicando uma vez sobre ela, tornando-a a linha ativa, e, em seguida, clicar no botão de edição, representado por um ícone de lápis e papel. Ao abrir a janela de edição, os dados daquela linha são exibidos em diferentes elementos do tipo *Entry*, conforme Figura 24. Na janela, é possível observar que há alguns elementos extras. Eles fazem parte de colunas que foram implementadas em etapas seguintes, portanto, devem ser desconsiderados no momento.

Ao realizar a edição dos dados, o usuário deve clicar no botão "SALVAR". Esse botão possui um comando que chama a função responsável pela edição dos dados. Essa função utiliza-se do comando SQL "UPDATE", responsável por atualizar os dados requeridos. No Quadro 6 a seguir é mostrado como é o código responsável por realizar essa operação.

Primeiro, o código identifica a linha que será editada. Para isso, ele utiliza a função `tabela.item(item_id) ['values'] [1]`, que acessa a linha selecionada na interface gráfica e extrai o valor correspondente à OS antiga. Esse valor é armazenado na variável `os_antigo` e será utilizado na condição "WHERE" do comando, garantindo que apenas a linha selecionada seja atualizada no banco de dados. A atualização dos dados é realizada com o comando UPDATE. Esse comando altera os valores das colunas da tabela, substituindo-os pelos novos valores fornecidos pelo usuário, que estão armazenados nas novas variáveis.

Figura 24 – Janela de edição de dados.

Fonte: O autor

Quadro 6 – Código para edição dos dados da tabela de aditivos.

```

conn = sqlite3.connect(caminho_arquivoDB)
c = conn.cursor()
os_antigo = tabela.item(item_id)['values'][1]
c.execute("UPDATE aditivos SET coordenador = ?, os = ?, valor = ?,
motivo = ? WHERE os = ?",
        (novo_coordenador,novo_os,novo_valor,novo_motivo,os_antigo))
conn.commit()
conn.close()

tabela.item(item_id,values=(novo_coordenador,novo_os,novo_valor,novo_motivo))
salvar_json()

```

Fonte: O autor.

Além de atualizar o banco de dados, o código também atualiza as mudanças na interface gráfica. A função `tabela.item(item_id, values=(...))` substitui os valores antigos exibidos na interface pelos novos valores fornecidos em tempo real.

Outra funcionalidade desenvolvida nesta etapa foi a exclusão de dados. Essa função segue o mesmo princípio da função de edição, mas agora utilizando o comando SQL DELETE. Para excluir um registro, o usuário deve selecionar uma linha específica na

tabela e pode optar por pressionar a tecla "Delete" do teclado ou clicar no botão de exclusão, representado por um ícone de lixeira. Esse botão, assim como o botão de edição, também está localizado no canto superior direito da tela inicial. No Quadro 7 está disposto o código da função de excluir linhas. A etapa que difere do código de edição, além do comando SQL, é a função `tabela.delete()` que prontamente retira a linha da interface.

Quadro 7 – Código para exclusão na tabela de aditivos.

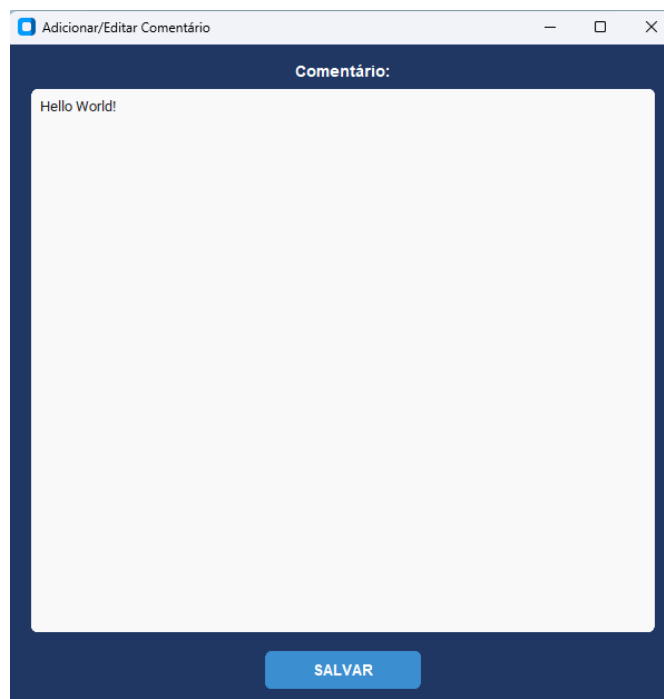
```
obra = tabela.item(selected_item[0])['values'][1]
conn = sqlite3.connect(caminho_arquivoDB)
c = conn.cursor()
c.execute("DELETE FROM aditivos WHERE os = ?", (obra,))
conn.commit()
conn.close()

tabela.delete(selected_item[0])
salvar_json()
```

Fonte: O autor.

Para finalizar a segunda etapa de implementação do executável, foi criada uma nova coluna na base de dados referente à inclusão de comentários. Ao adicionar uma nova linha na tabela de dados, a coluna comentário é iniciada em branco. Caso o usuário queira adicionar anotações acerca daquele aditivo, basta selecionar a linha e clicar no botão "COMENTÁRIO" localizado à esquerda dos botões de edição e exclusão. Ao clicar no botão, uma janela secundária será aberta, como visto na Figura 25.

Figura 25 – Janela para inserção de comentários.



Fonte: O autor

Após o usuário ter escrito o texto na caixa, basta clicar no botão salvar, que, através do comando `UPDATE`, irá atualizar a base de dados. Para identificar as linhas que já possuem comentários, já que eles não aparecem na interface principal, foi inserido um comando que avalia se a coluna comentário não está vazia. Caso isso seja verdade, a cor de fundo daquela linha muda para um tom de verde. Com esses dados implementados, a segunda entrega está concluída.

3.2.2.1.3 Entrega 3: histórico, liberação dos dados e filtragem da tabela

Para a terceira etapa do desenvolvimento do executável, foi adicionada uma nova janela principal que, ao clique de um botão, permite alternar com a janela inicial. Essa nova janela se trata da janela do histórico. Quando um aditivo é inserido na tabela da janela inicial, significa que a obra ligada àquela OS está em andamento. Ao finalizar a obra, o aditivo não necessita mais aparecer na tabela como custo em elaboração. Para isso, foi criada uma nova coluna chamada de status.

Ao adicionar um novo aditivo na tabela, é atribuído o status "EM ANDAMENTO" para aquele caso. A partir disso, só aparecerão na tabela da tela inicial os casos que tiverem esse status. Ao finalizar a obra, o usuário pode transferir esse aditivo para a tabela do histórico selecionando a linha e clicando em um novo botão de liberação (representado por uma bandeira) inserido no frame superior ao lado dos botões de edição e exclusão.

Ao clicar nesse botão, o status da linha selecionada é alterado, através do comando `UPDATE`, para "FINALIZADO" e é atribuída uma nova coluna de data de liberação, que salva o momento em que esse aditivo foi transferido para o histórico. No Quadro 8, é possível ver a função que realiza essas operações.

Depois de atualizar o banco de dados, o código atualiza também as alterações na interface gráfica. O comando `tabela.delete(item_id)` remove a linha correspondente da tabela exibida na interface. Isso garante que o usuário visualize a remoção da linha quase que instantaneamente. Por fim, é chamada a função `filtrar_dados()` para atualizar os registros exibidos nas duas interfaces com base no status. Se a tabela for `tree` (tabela inicial), apenas os registros com o status "EM ANDAMENTO" serão exibidos. Já se for `hist_table` (tabela do histórico), serão mostrados apenas os registros com o status "FINALIZADO".

Para acessar a janela que exibe a tabela com os dados do histórico, foi adicionado um botão posicionado acima do *frame* de inserção de dados. Ao clicar nesse botão, a janela inicial é substituída pela janela do histórico. Essa nova janela mantém praticamente as mesmas funções e botões presentes na janela inicial. Além disso, para retornar à janela inicial, o usuário pode clicar no botão "Voltar", representado por um ícone de flecha apontando para a esquerda, localizado à esquerda da tabela de dados do histórico. Na Figura 26, é apresentada a disposição dos elementos na interface da janela do histórico.

Quadro 8 – Código para liberação dos aditivos para o histórico.

```

selected_item = tabela.selection()
if selected_item:
    item_id = selected_item[0]
    valores = tabela.item(item_id)['values']
    coordenador = valores[0]
    os = valores[1]
    valor = valores[2]
    motivo = valores[3]
    data_liberacao = datetime.now().strftime("%d/%m/%Y %H:%M:%S")

    conn = sqlite3.connect(caminho_arquivoDB)
    c = conn.cursor()
    c.execute("UPDATE aditivos SET status = ?, data_liberacao = ? WHERE os = ?",
              ('FINALIZADO', data_liberacao, os))
    conn.commit()
    conn.close()

    tabela.delete(item_id)
    salvar_json()

    if tabela == tree:
        filtrar_dados(tree, status_filtro="EM ANDAMENTO")
    elif tabela == hist_table:
        filtrar_dados(hist_table, status_filtro="FINALIZADO")

```

Fonte: O autor.

Figura 26 – Janela do histórico dos aditivos.

COORDENADOR	OS	VALOR ORÇADO (R\$)	VALOR MONT. (R\$)	VALOR ADITIVO (R\$)	MOTIVO
COORD4	600043587	200.000,00	118.850,00	35.000,00	MAQUINÁRIO EXTRA

Fonte: O autor

Na interface, é possível visualizar que os botões de edição, exclusão e de comentário foram mantidos nas mesmas posições. Aliás, na imagem é notável a presença de dois campos de inserção de valores localizados no campo superior da tela. Esses campos são utilizados para realizar a filtragem de dados nas tabelas, tanto pelo número da OS, quanto pelo nome do coordenador responsável. Na Figura 27, há uma representação mais aproximada desses campos.

Figura 27 – Campos para filtragem de dados.

A interface de filtragem de dados é exibida em um fundo azul escuro. Ela contém dois campos de entrada de texto brancos. O primeiro campo é precedido pelo rótulo "OS:" e o segundo pelo rótulo "Coordenador:". À direita de cada campo há um ícone de funil azul, usado para aplicar o filtro. O campo "Coordenador:" também possui uma seta de rolagem para alternar entre o texto e o número.

Fonte: O autor

Basta o usuário escrever um trecho do código da OS ou do nome do coordenador que as linhas da tabela que não fazem parte do filtro inserido serão ocultadas. Os ícones de funis ao lado dos campos são os botões que, ao serem clicados, realizam a filtragem com base nos valores que estão nos campos. No entanto, caso o usuário pressione "Enter", os filtros também serão aplicados. Para a filtragem dos dados, foi elaborada uma função que pode ser vista no Quadro 9.

Quadro 9 – Código para filtragem de dados e exibição na tabela.

```
for item in tabela.get_children():
    tabela.delete(item)
dados = carregar_dados()
for row in dados:
    coordenador, os, valor, motivo, comentario, data_alteracao, status = row
    if status_filtro and status != status_filtro:
        continue
    if filtro_coordenador and filtro_coordenador.lower()
not in coordenador.lower():
        continue
    if filtro_os and filtro_os.lower() not in os.lower():
        continue
    tabela.insert("end", values=(coordenador, os, valor,
        motivo, comentario, data_alteracao, status))
```

Fonte: O autor.

Primeiramente, o código limpa todos os registros atualmente exibidos na tabela da interface gráfica. Isso é realizado por meio do laço `for item in tabela.get_children():` `tabela.delete(item)`, que percorre todos os itens existentes na tabela e os remove. Em seguida, a função `carregar_dados()` é chamada para obter todos os registros armazenados. O retorno dessa função é armazenado na variável "dados", que contém uma lista de tuplas, onde cada tupla representa um registro com várias colunas, como coordenador, os, valor, motivo, entre outras. O código então percorre cada registro em "dados" utilizando `for row in dados`. Em cada iteração, as colunas do registro são separadas em variáveis individuais.

Dentro do laço, o código aplica filtros com base nos critérios fornecidos pelo usuário. Se o filtro `status_filtro` estiver ativo e o campo status do registro não corresponder ao filtro, o comando `continue` é executado, pulando para o próximo registro. Da mesma forma, se o filtro `filtro_coordenador` estiver ativo e o valor do campo coordenador não contiver o texto especificado no filtro, o registro também é ignorado. Por último, o

filtro `filtro_os` verifica se o valor do campo "os" contém o texto fornecido no filtro, caso contrário, o registro é descartado. Após passar pelos filtros, os registros que atendem aos critérios são adicionados novamente à tabela. Ao finalizar a implementação dos filtros, a etapa 3 está encerrada, passando agora para a última etapa de implementação do executável de aditivos.

3.2.2.1.4 Entrega 4: valores orçado e contratado, custos extras e geração de extratos

Na última entrega, foram implementadas funções adicionais que puderam suprir uma demanda extra dos coordenadores de obras ao ter que adicionar aditivos dentro de cada OS. Sempre que era criada uma requisição de compra de aditivos, os coordenadores tinham que imprimir um arquivo em PDF contendo um extrato daquela OS e de quanto a presença do aditivo impactou no valor salvo em custos daquela obra. Para entender melhor, o valor salvo se trata de diferença entre o valor orçado daquela obra e o valor realizado. Caso o valor realizado ultrapasse o valor orçado, o coordenador tem que escrever uma justificativa citando o porquê, naquela obra, precisou-se gastar um valor maior do que foi orçado.

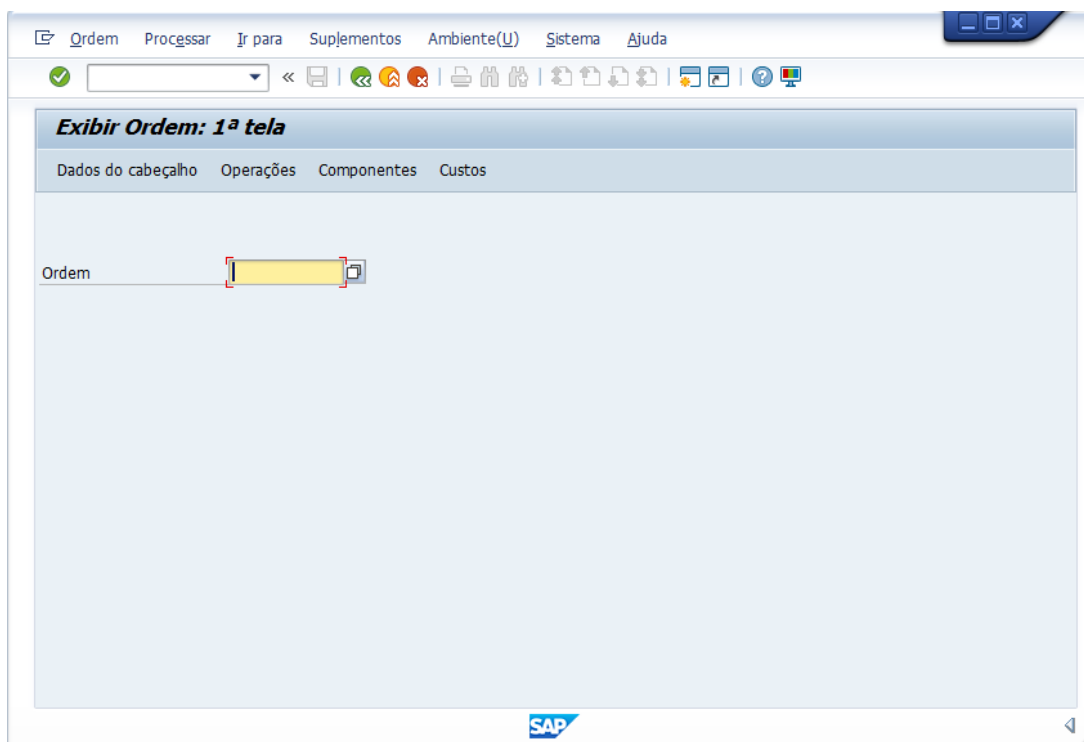
Para implementar essa função, primeiro precisou-se adicionar duas novas colunas que são preenchidas no momento em que uma nova linha é inserida. Essas novas colunas são referentes ao valor orçado e ao valor contratado. Para coletar essas informações no momento da inserção de dados, foi preciso utilizar o SAP.

Ao clicar no botão "INSERIR", antes de adicionar os dados no arquivo DB e JSON, uma função chamada `dados_SAP` é chamada. Nela, o número da OS inserida no campo é adquirido e é realizada uma tentativa de conexão com a janela do SAP GUI. Caso o SAP não esteja aberto, uma outra função é chamada para abrir a interface do programa. Após verificar se o programa estava aberto, é inserida automaticamente no campo de transações a transação IW33, responsável por exibir os dados de uma respectiva OS. A janela do SAP apresentada na Figura 28 representa a interface do SAP após entrar nessa transação.

No campo disponível à direita de "Ordem" é inserido o número da OS e, após isso, é executada a ação de pressionar o botão "Enter" do teclado. Dessa maneira, a janela daquela OS é aberta, podendo assim adquirir os dados requisitados. Para a aquisição do valor orçado, existe um campo dentro da OS que comporta, de forma padronizada, inúmeros dados acerca daquela OS como cliente, localização da obra, PEP, tensão e potência, escopo e o valor orçado. Esse campo pode ser visto na Figura 29.

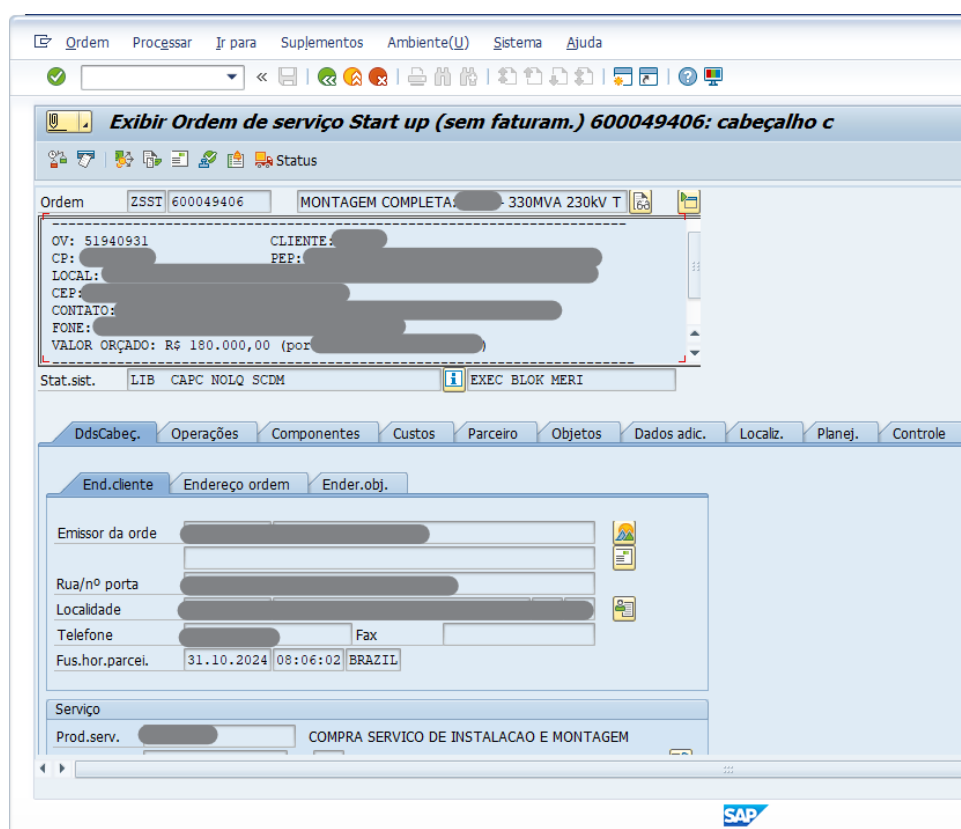
Ao adquirir o texto completo do campo, são realizadas várias manipulações com a *string* até chegar somente ao valor. Essas manipulações podem variar um pouco de um caso para outro, porém, o valor orçado sempre estará na mesma posição, o que facilita a extração do mesmo.

Figura 28 – Janela do SAP GUI para exibição de ordens de serviço.



Fonte: O autor

Figura 29 – Campo onde está localizado o valor orçado da OS.



Fonte: O autor

Com o valor orçado extraído do SAP, agora é preciso extrair também o valor de contratação. Esse valor está localizado em uma tabela preenchida pelo contratante. Nessa tabela estão contidos todos os prestadores que participaram do processo licitatório daquela obra, juntamente com o valor das propostas de cada um. Ao final da tabela dos prestadores, é atribuído o valor da proposta vencedora. Na Figura 30 está uma representação dessa tabela.

Figura 30 – Tabela de dados no SAP onde está localizado o valor de contratação.

DdsCabeç. Operações Componentes Custos Parceiro Objetos Dados adic. Localiz. Planej. Controle										
Oper	SOp	CenTrab	Ce...	Ch...	ChvMo...	C..	Txt.breve operação			TD
0010		05040007	1202	SM01			MONTAGEM COMPLETA 330MVA			Trabalho real
0020		05040007	1202	SM01			- COTAÇÕES:			0,000
0030		05040007	1202	SM01			-> DECLINOU (Indisponível - via			0,000
0040		05040007	1202	SM01			-> R\$ 150.000,00			0,000
0050		05040007	1202	SM01			-> R\$ 154.993,34			0,000
0060		05040007	1202	SM01			-> DECLINOU (Sem resposta)			0,000
0070		05040007	1202	SM01			-> DECLINOU (Sem resposta)			0,000
0080		05040007	1202	SM01			-> DECLINOU (Sem resposta)			0,000
0090		05040007	1202	SM01			-> DECLINOU (Sem resposta)			0,000
0100		05040007	1202	SM01			.			0,000
0110		05040007	1202	SM01			APROVADO:			0,000
0120		05040007	1202	SM01			-> R\$ 150.000,00			0,000
0130		05040007	1202	SM01			.			0,000
0140		05040007	1202	SM01			RESULTADO: 16,7%			0,000
0150		05040007	1202	SM01			.			0,000
0160		05040007	1202	SM03			- MONTAGEM COMPLETA			0,000

Fonte: O autor

Nota-se que o valor requisitado localiza-se na linha abaixo do termo "APROVADO:", com esse tipo de padronização, foi possível construir uma função que itera sobre as linhas dessa tabela e, ao encontrar o termo citado, extrai o valor da linha seguinte. A partir daí, é preciso apenas manipular a *string* adquirida para ter o valor correto de contratação.

Com os valores adquiridos, agora é possível passar para a implementação do gerador de extratos. Na parte superior da interface do executável, foi adicionado um botão chamado "EXTRATO". Ao selecionar uma linha e clicar nele, uma janela é aberta contendo as informações que estarão presentes no arquivo do extrato. Essa janela pode ser vista na Figura 31.

Nessa janela, o usuário tem a possibilidade de conferir se as informações estão corretas antes de gerar o PDF. As informações de cliente e escopo são extraídas daquele mesmo campo no SAP onde estava localizado o valor orçado. Todos os outros campos já estavam presentes na base de dados, inclusive o comentário, que agora pode ser usado como campo para descrever a justificativa de aditivo que estará presente no arquivo do extrato.

No extrato, também consta uma linha que apresenta o valor do resultado, isto é, a porcentagem que o valor realizado ficou em relação ao valor orçado. Para calcular esse número, foi utilizada a fórmula apresentada na Equação (1) a seguir.

Figura 31 – Janela com as informações do extrato.

Fonte: O autor

$$resultado = 100 - \left(\frac{valor\ realizado}{valor\ orçado} \times 100 \right) \quad (1)$$

Na janela do extrato também existem dois botões, o primeiro representado por uma figura de um olho, abre uma janela que possibilita a visualização prévia de como ficará o extrato. Já o segundo botão, representado por um ícone de arquivo PDF, abre uma janela do Windows de seleção de pasta. Ao selecionar a pasta, o arquivo PDF do extrato é salvo nela com a nomenclatura "EXTRATO - OS 00000000.pdf".

Para gerar esse extrato, foi utilizada a biblioteca **reportlab**. Essa biblioteca disponibiliza inúmeros elementos e funções que facilitam a construção de estruturas complexas, como tabelas, gráficos e estilos personalizados. No caso deste projeto, ela foi utilizada para construir a estrutura da tabela onde serão colocadas cada uma das informações necessárias no extrato. Além disso, o **reportlab** permite configurar o layout do documento, ajustando margens, fontes, cores e espaçamentos. Na Figura 32, está uma representação genérica do modelo do extrato, que reflete como esses elementos foram organizados visualmente.

Após o gerador de extratos ser finalizado, uma última adição foi feita no executável durante essa etapa. Trata-se da inclusão de seis novas colunas na base de dados. Essas colunas foram criadas para armazenar informações detalhadas que complementam os dados já existentes. Elas representam despesas adicionais que são planejadas antes da execução da obra, eventuais multas revertidas devido à aplicação do aditivo, e o valor inicial do aditivo. Os valores dessas colunas podem ser inseridos diretamente pelo usuário por meio

Figura 32 – Modelo de genérico do extrato.

EXTRATO: CLIENTE / OS 00000000		
ATIVIDADE	VALOR ORÇADO	VALOR REALIZADO
ESCOPO	100.000,00	95.000,00
MOTIVO DO ADITIVO		25.000,00
TOTAL	100.000,00	120.000,00
RESULTADO		-20,00%
OBS	Justificativa	

Fonte: O autor

da janela de edição, que foi adaptada para incluir os campos adicionais (Figura 24). A seguir, estão descritas, em detalhes, cada uma dessas novas informações incluídas na base de dados.

- **Custo Gerador:** Custo planejado referente à necessidade de alugar um gerador para a obra.
- **Custo PTA:** Custo planejado referente à necessidade de alugar uma plataforma de trabalho aéreo para a obra.
- **Custo Óleo:** Custo planejado referente à necessidade de ter carretas de óleo extras na obra.
- **Custo Outros:** Demais custos planejados para a obra.
- **Multa revertida:** Valores de multas que foram revertidos devido à contratação de diferentes aditivos que auxiliaram na entrega da obra dentro do prazo.
- **Valor inicial de aditivo** Valor inicial do aditivo, que fica fixo caso o valor principal seja alterado.

Todos esses valores serão de extrema importância para a análise dos indicadores de aditivos no Power BI, pois complementam e deixam mais claras as motivações para a existência de cada um dos aditivos. Com essa última adição, o executável está bem estruturado e preparado para receber futuras atualizações.

3.2.2.2 Indicadores de custos de ordens de serviço encerradas

Ao finalizar todos os processos relacionados à obra, o coordenador precisa encerrar a sua respectiva OS. Ao final de cada mês, todos os custos presentes em cada OS que foi encerrada durante o mês precisam ser extraídos e inseridos em uma base de dados para que seja obtido o resultado daquele respectivo mês. Esse resultado faz parte de uma das metas da seção, por conta disso, é de extrema importância possuir uma análise precisa desses dados.

Antes da implementação dos indicadores, essa análise era realizada de forma manual, em um processo que demandava tempo e atenção redobrada para não deixar nenhum custo

passar despercebido. Diante desse cenário, o risco do valor total dos custos extraídos estar correto era alto devido à alta suscetibilidade a erros humanos. Para isso, foi implementado um sistema robusto capaz de adquirir, de forma extremamente eficiente e rápida, todos os custos discriminados em um determinado período delimitado pelo usuário que for utilizá-lo. Esse projeto foi desenvolvido através de inúmeras macros em VBA, onde cada uma delas é responsável por realizar uma ação para a aquisição discriminada dos custos.

Para gerenciar a execução dessas macros, foi confeccionada, em uma planilha no Excel, uma interface com vários botões que, ao serem pressionados, rodam uma macro específica atribuída a eles. Na Figura 33 está um recorte dessa interface contendo a tabela dos dados discriminados, juntamente com os botões de ação.

Figura 33 – Planilha em Excel contendo a interface de comando do sistema.

	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	OS	ESCOPO	CLIENTE	PRESTADOR	VALOR ORÇADO	CUSTO PRESTADOR	CUSTO WEG	CUSTO TOTAL (IW73)	CUSTO TOTAL (KOB1)	DIFERENÇA	SAP			
2									RS	-	RS	-		
3									RS	-	RS	-		
4									RS	-	RS	-		
5									RS	-	RS	-		
6									RS	-	RS	-		
7									RS	-	RS	-		
8									RS	-	RS	-		
9									RS	-	RS	-		
10									RS	-	RS	-		
11									RS	-	RS	-		
12									RS	-	RS	-		
13									RS	-	RS	-		
14									RS	-	RS	-		
15									RS	-	RS	-		
16									RS	-	RS	-		
17									RS	-	RS	-		
18									RS	-	RS	-		
19									RS	-	RS	-		
20									RS	-	RS	-		
21									RS	-	RS	-		
22									RS	-	RS	-		
23									RS	-	RS	-		
24									RS	-	RS	-		
25									RS	-	RS	-		
26									RS	-	RS	-		
27									RS	-	RS	-		
28									RS	-	RS	-		
29									RS	-	RS	-		
30									RS	-	RS	-		
31									RS	-	RS	-		
32									RS	-	RS	-		
33									RS	-	RS	-		
34									RS	-	RS	-		

Fonte: O autor

Na interface, estão visualmente dispostos à direita os botões de execução das macros. O primeiro botão (em verde), chamado "BUSCAR DADOS", executa, de maneira ordenada, as macros responsáveis por adquirir os dados em diferentes transações do SAP, manipula esses dados com base em várias condicionais e os insere em suas colunas na tabela da interface. A seguir será mostrado como se dá o processo de execução dessas macros.

No momento em que o usuário clica no botão de busca, são abertas duas janelas sequenciais de inserção de valores das datas de início e data final. Essas janelas podem ser vistas na Figura 34. Após o preenchimento das datas de início e término na interface, o sistema inicia a execução automatizada das macros, acessando diferentes transações do SAP para capturar os custos discriminados.

Na transação IW73, conforme ilustrado na Figura 35 que mostra sua configuração, é responsável por listar todas as ordens de serviço encerradas dentro de um período determinado pelo usuário. Durante essa etapa, são coletadas informações como número da ordem, cliente, data de encerramento, escopo do serviço e valores gerais relacionados. Já a transação KOB1, cuja janela inicial é apresentada na Figura 36, fornece o detalhamento

Figura 34 – Caixas para inserção das datas de início e data final.

The image shows two separate dialog boxes for date selection. The top dialog is titled 'Data Inicial' and the bottom one is titled 'Data Final'. Both dialogs have a close button (X) in the top right corner. Inside each dialog, the text 'Formato DD/MM/AAAA' is displayed. Below this text is a single-line text input field. To the right of the input field are two buttons: 'OK' and 'Cancelar'.

Fonte: O autor

dos custos discriminados, dividindo-os entre valores associados aos prestadores de serviço e os custos internos da empresa.

Figura 35 – Janela inicial da transação IW73 no SAP.

The image shows the initial screen of the SAP transaction IW73, titled 'Exibir ordem de serviço: seleção de ordens'. The screen has a standard SAP header with a menu bar and a toolbar. Below the header, there is a sub-header with the title 'Exibir ordem de serviço: seleção de ordens' and a button 'Receptor de apropriação de custos MAP'. The main area is divided into two sections. The top section, 'Status da ordem', contains several checkboxes: 'pendente', 'em procmt', 'encerrado' (which is checked), and 'histórico'. There is also a field for 'Esq.seleção' and an 'End.' button with a red X icon. The bottom section, 'Seleção de ordens', contains a table with various fields for selection. The fields are: 'Ordem', 'Tipo de ordem' (with value 'ZSST'), 'Local de instalação', 'Equipamento', 'Material', 'Nº de série', 'Dds.adics.dispos.', 'Nota', 'Centro trab.respons.' (with value '05040007'), 'Centro p/cen.trab.', 'Período', 'Parceiro', and 'Moeda'. Each field has a corresponding 'até' field and a selection icon (a yellow box with a magnifying glass).

Fonte: O autor

Figura 36 – Janela inicial da transação KOB1 no SAP.

Exibir part.indiv.de custos reais p/ordens : 1ª tela

Outros critérios de seleção...

Área contab.custos: BR00

Ordem: [] até: []

ou

Grupo de ordens: []

Classe de custo: [] até: []

ou

Grupo de classes de custo: []

Dados do lançamento

Data de lançamento: 01.01.2018 até: 31.12.2024

Configurações

Variante de exibição: /INDIC C&S INDICADORES C&S

Outras opções...

Fonte: O autor

Após a coleta dos dados, eles são transferidos automaticamente para a planilha auxiliar da interface, onde passam por um processo de organização e validação. Durante essa etapa, uma macro específica verifica a consistência dos dados extraídos, removendo valores negativos ou inválidos e padronizando formatos para facilitar a análise subsequente.

Além disso, a automação também abrange a distribuição de custos. Com o uso de dicionários configurados para identificar prestadores e escopos de serviço, os dados são classificados e atribuídos de forma precisa. Isso permite que as despesas sejam distribuídas entre custos internos da empresa e custos dos prestadores, viabilizando um controle mais rigoroso sobre cada etapa do processo. Esse detalhamento é essencial não apenas para fins operacionais, mas também para análises financeiras que embasam decisões estratégicas.

Outro ponto fundamental do processo é a validação e correção dos dados. Uma macro revisa ordens que possuem valores orçados zerados, utilizando filtros adicionais para buscar informações faltantes diretamente no SAP. Esse procedimento garante que nenhuma OS com valor orçado seja ignorada na análise. Além disso, macros específicas ajustam diferenças entre valores orçados e realizados, melhorando ainda mais a qualidade dos dados.

As macros restantes são responsáveis por inserir cada OS da tabela auxiliar da interface nas diferentes tabelas de custos gerais, custos com carretas de óleo e custos com aditivos. Para cada uma delas, existe uma lógica baseada em diversos condicionais particulares de cada tipo de custo. Por exemplo, para saber se o custo é de carreta de óleo, os valores de uma das colunas na tabela da transação KOB1 devem ser iguais a "RPA-00000".

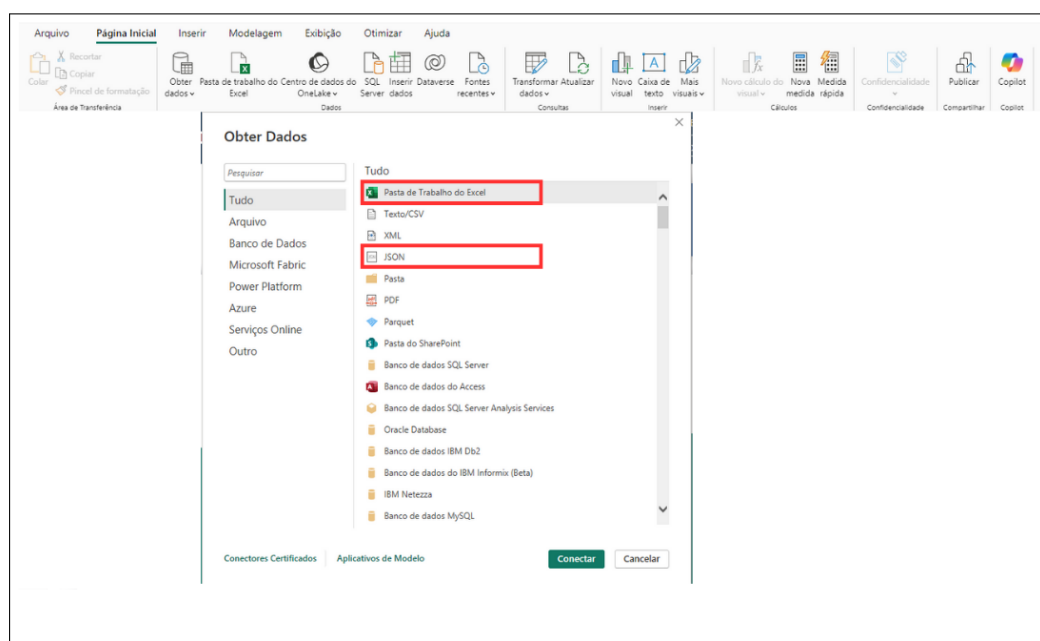
3.3 ELABORAÇÃO DAS *DASHBOARDS* INTERATIVAS NO POWER BI A PARTIR DAS BASES DE DADOS

Após reunir e organizar todos os dados relevantes sobre as cotações e a entrada, ROL e carteira em suas respectivas bases de dados estruturadas, o próximo passo foi apresentá-los de uma forma intuitiva e interativa. Para isso, foram desenvolvidas telas interativas no Power BI, projetadas para exibir essas informações de maneira visual e acessível. A integração das tabelas de dados do Excel ao Power BI foi realizada por meio do recurso de importação direta, garantindo que todas as informações fossem transferidas de forma organizada e prontas para análise.

3.3.1 Importação de dados no Power BI

O processo de importação consiste em selecionar o arquivo Excel no Power BI, identificar a tabela onde estão os dados e ajustar os campos necessários. Esse procedimento cria conexões dinâmicas e possibilita atualizações automáticas: qualquer alteração na base de dados original será refletida no Power BI ao atualizar os dados. A Figura 37 ilustra o exemplo da importação de uma tabela de cotações de partes e peças.

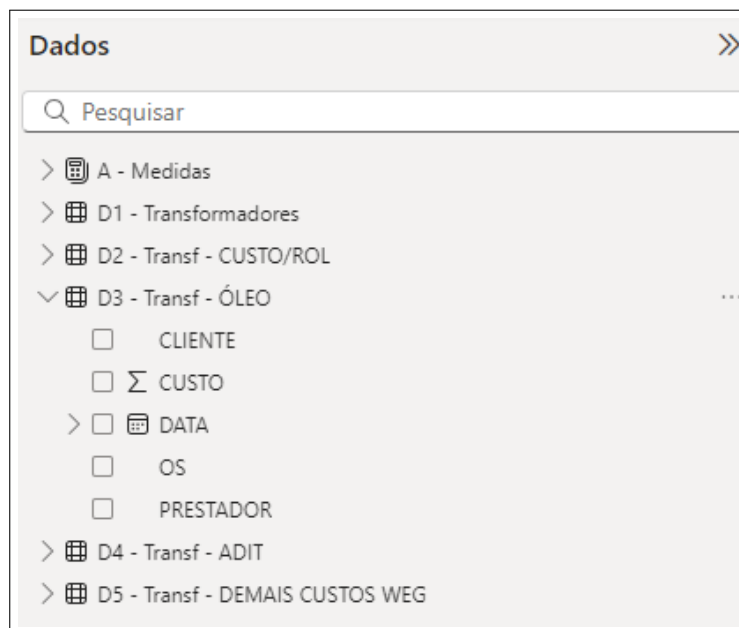
Figura 37 – Importação direta de dados para o Power BI.



Fonte: O autor

Na imagem são destacados os tipos de arquivos de dados que foram utilizados para esse projeto. Com os dados importados, as tabelas aparecem na janela de campos do Power BI, como mostrado na Figura 38. Além das colunas originais, é possível adicionar novas colunas e criar medidas diretamente no Power BI utilizando a linguagem DAX.

Figura 38 – Campos de manipulação das tabelas importadas no Power BI.



Fonte: O autor

3.3.2 Criação de colunas calculadas e medidas com DAX

A linguagem DAX permite a realização de cálculos personalizados e dinâmicos. As colunas calculadas aprimoram as colunas originais ou criam novos filtros derivados de informações existentes, enquanto as medidas possibilitam cálculos agregados mais complexos.

Por exemplo, medidas como **HASONEVALUE** são utilizadas para validar se apenas uma única seleção está ativa em um relatório, enquanto funções como **SELECTEDVALUE** recuperam o valor selecionado em um campo específico. Essas funcionalidades promovem maior precisão e clareza nas análises, ajustando automaticamente os resultados aos filtros aplicados pelo usuário. Além disso, funções como **CALCULATE** permitem modificar o contexto dos filtros aplicados a uma medida, e **IF** viabiliza a aplicação de lógica condicional. Essas capacidades tornam o Power BI uma ferramenta poderosa para a criação de relatórios dinâmicos e interativos.

A utilização de medidas no Power BI aprimora significativamente a interação com os dados. Medidas são fórmulas que permitem cálculos dinâmicos baseados em filtros, dimensões e seleções feitas nas visualizações. Isso garante que os relatórios sejam altamente responsivos, fornecendo análises personalizadas e orientadas ao contexto. Por exemplo, com medidas complexas, é possível calcular médias ponderadas, taxas de crescimento e valores acumulados, indo além das simples agregações. Essas funcionalidades ajudam a criar gráficos, tabelas e outros visuais que refletem diretamente os critérios selecionados pelos usuários, como intervalos de datas ou categorias específicas.

A seguir, apresenta-se uma tabela com as principais funções DAX utilizadas no

Power BI, destacando suas aplicações:

Tabela 1 – Principais Funções DAX e Suas Aplicações

Função DAX	Descrição
HASONEVALUE	Verifica se há exatamente um valor selecionado em uma coluna, útil para validar filtros ou contextos específicos.
SELECTEDVALUE	Retorna o valor atualmente selecionado em um campo; se houver múltiplos valores, retorna o padrão definido.
CALCULATE	Modifica o contexto do filtro aplicado em uma medida, permitindo cálculos sob diferentes condições de filtragem.
IF/ELSE	Aplica lógica condicional para retornar resultados diferentes com base em uma condição avaliada como verdadeira ou falsa.
UNION	Combina duas ou mais tabelas em uma única tabela, útil para trabalhar com dados de múltiplas fontes.
SUMX	Realiza a soma de expressões avaliadas linha a linha em uma tabela especificada.
FILTER	Retorna uma tabela filtrada com base em uma condição definida, permitindo manipulação precisa dos dados.
ALL	Remove todos os filtros do contexto atual em uma medida ou coluna, útil para cálculos de totais ou proporções.
DIVIDE	Realiza divisões com tratamento para erros em divisões por zero, retornando um valor padrão nesses casos.
RELATED	Recupera um valor de uma tabela relacionada com base em uma relação definida no modelo de dados.

Nas telas criadas para os indicadores no Power BI, várias funções DAX foram usadas para atender às necessidades de análise e visualização de dados de forma prática e eficiente. A função CALCULATE foi uma das mais importantes, pois permitiu ajustar os filtros aplicados nos cálculos, sendo essencial para criar métricas como taxas de conversão e acompanhar o desempenho com base em diferentes cenários e filtros específicos, como status de cotações ou tipos de pedidos. A função FILTER foi outra ferramenta fundamental, usada para criar tabelas temporárias que isolavam dados específicos, como cotações de maior valor ou ordens de serviço em situações críticas. Isso ajudou a destacar informações relevantes nos dashboards, deixando a análise mais focada e objetiva.

Já a função DIVIDE foi bastante usada em cálculos que envolvem proporções, como o percentual de metas atingidas ou a relação entre custos orçados e realizados. Uma característica dessa função é que ela trata automaticamente divisões por zero, garantindo resultados consistentes e sem erros. Para exibir valores dinâmicos dependendo das seleções dos usuários, a função SELECTEDVALUE foi muito utilizada. Com ela, foi possível mostrar detalhes específicos, como tipos de venda ou categorias de serviços, diretamente ligados ao que o usuário escolheu no momento, deixando a experiência mais interativa e personalizada.

Nas análises mais detalhadas, como a dos custos de obras encerradas, a função RELATED desempenhou um papel importante, permitindo buscar informações em tabelas relacionadas. Isso facilitou, por exemplo, a comparação de custos reais com os valores orçados, evidenciando desvios de forma precisa. Outras funções como a HASONEVALUE foram usadas para validar seleções e garantir que os cálculos estivessem corretos em contextos específicos. Já a SUMX, que calcula expressões linha a linha, foi essencial para somar valores ponderados ou realizar análises mais complexas nos dados.

Essas funções foram combinadas de forma estratégica nas *dashboards*, resultando em relatórios visuais e interativos que atendem a diferentes demandas. Gráficos, tabelas e indicadores dinâmicos criados com DAX não só aceleraram a análise dos dados por meio de cálculos automatizados e filtrações rápidas, como também facilitaram decisões ao fornecer informações detalhadas e organizadas em tempo real. Essa integração permitiu identificar tendências e padrões que antes não eram perceptíveis. Além disso, foi possível detectar gargalos operacionais e monitorar metas com maior precisão, tornando o processo decisório mais ágil e embasado em dados consistentes e atualizados.

4 RESULTADOS

Para analisar os resultados obtidos através do processo de obtenção dos dados obtidos ao longo dos diferentes métodos de coleta e de apresentação dos mesmos em *dashboards* interativas no Power BI, esta seção detalhará os principais pontos positivos alcançados ao longo do tempo. Serão exibidas comparações entre o cenário anterior e o posterior à implementação, destacando os ganhos financeiros, a eficiência operacional e a melhoria nos processos de tomada de decisão.

4.1 CÉLULA DE VENDAS

Os resultados alcançados na célula de vendas foram estruturados com base nas principais etapas implementadas, sendo elas a coleta dos dados de cotações e o acompanhamento das entradas, do ROL e da carteira. A efetividade desses resultados está refletida nas diversas telas interativas do Power BI.

4.1.1 Cotações

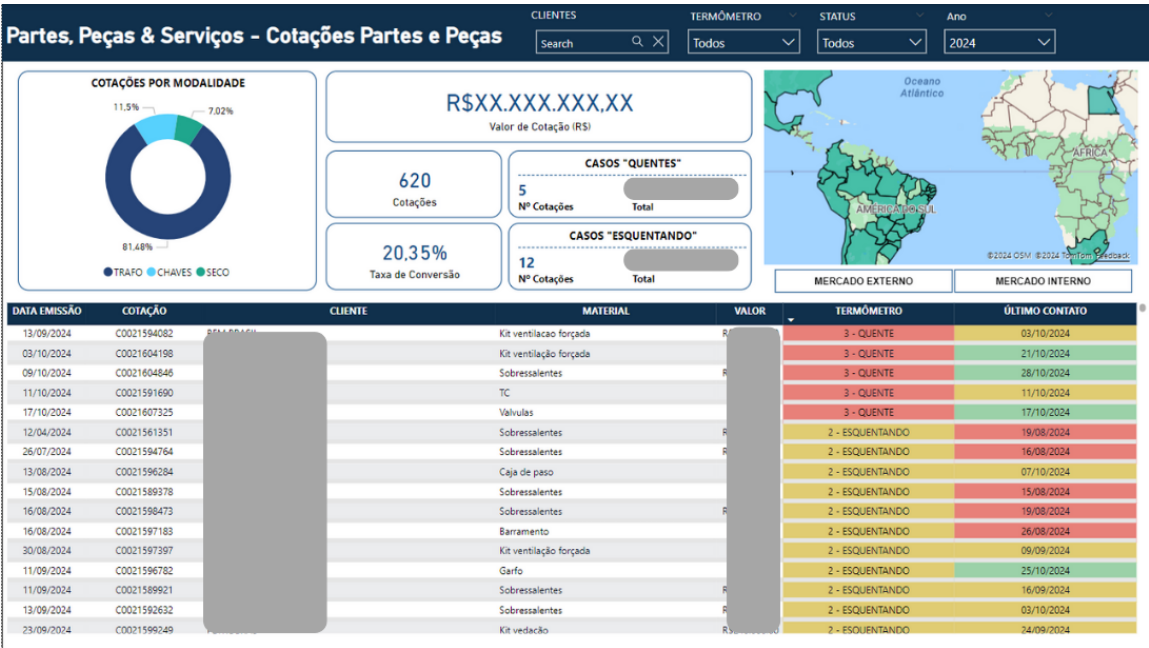
Começando pelas cotações, antes da implementação do sistema para capturar, organizar e analisar as mesmas, o cenário era definido pela falta de um controle e pela alta dependência de processos manuais e dispersos. As cotações eram gerenciadas de maneira quase que desestruturada, sem uma base de dados consolidada que permitisse análises ou consultas rápidas. Esse modelo acabava por dificultar o acompanhamento do desempenho da equipe e a avaliação das taxas de conversão.

Além disso, a ausência de um sistema unificado impedia que os superiores tivessem uma visão clara do status das cotações. Não era possível saber rapidamente quais cotações estavam em aberto, ou perto de serem fechadas, nem identificar clientes com maior potencial de conversão. Essa falta de controle também dificultava a formulação de estratégias baseadas em dados confiáveis, impactando diretamente o planejamento da célula de vendas.

A partir da implementação do sistema, mudou-se completamente esse panorama. Com a automação e a centralização dos dados, foi possível eliminar esses obstáculos, trazendo mais organização e uma maior capacidade de análise e tomada de decisão. Na Figura 39 e 40 é possível visualizar como ficaram as *dashboards* relativas às cotações de sobressalentes e de serviços.

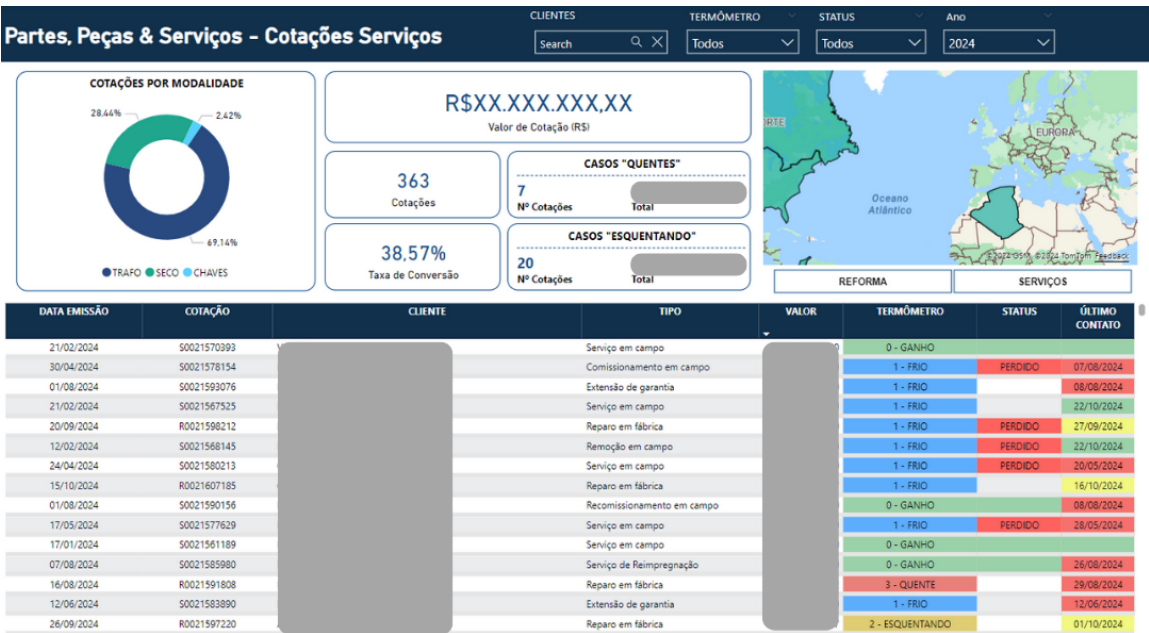
A centralização dos dados em uma interface visualmente clara e funcional permite o monitoramento em tempo real das cotações, destacando o acompanhamento por status, como "quente" e "esquentando". A inclusão de gráficos e mapas deixa a análise por categoria de produtos e por regiões, enquanto a tabela detalhada organiza informações essenciais, como cliente, material e último contato. Por sua vez, com filtros dinâmicos, localizados

Figura 39 – Dashboard no Power BI das cotações de sobressalentes.



Fonte: O autor

Figura 40 – Dashboard no Power BI das cotações de serviços e reformas.



Fonte: O autor

no topo, o sistema otimiza o acompanhamento por cliente, termômetro, status e datas, podendo comparar, por exemplo, como a taxa de conversão evoluiu com o passar dos anos.

Figura 42 – Dashboard no Power BI dos dados de ROL e Carteira.



Fonte: O autor

Já a tela de ROL e Carteira, além de trazer campos semelhantes aos da tela de entrada, também apresenta a evolução dos valores passando de carteira (azul) para ROL (cinza) ao longo dos meses, com o gráfico de colunas.

Dessa maneira, os aprimoramentos implementados na análise dos dados de entrada e ROL/Carteira trouxeram ganhos em termos de eficiência e precisão na análise de dados. Agora, com gráficos interativos e dados organizados por tipo de venda, metas e status financeiro, é possível realizar planos de ação mais assertivos e embasados. Esses aprimoramentos possibilitaram a criação de uma rotina de acompanhamento das metas de vendas estratégica e eficaz.

4.2 CÉLULA DE COMISSIONAMENTO

Para os resultados alcançados na célula de comissionamento, estruturou-se uma análise com base nas principais etapas implementadas, sendo elas a coleta dos dados dos indicadores de contratações (ata do PCP e carteira de vendas) e dos indicadores de execução de obras (aditivos e resultados das obras encerradas). A eficácia desses resultados, assim como em vendas, também está refletida nas diversas telas interativas do Power BI. Mas também em resultados financeiros baseados em economia de tempo de trabalho.

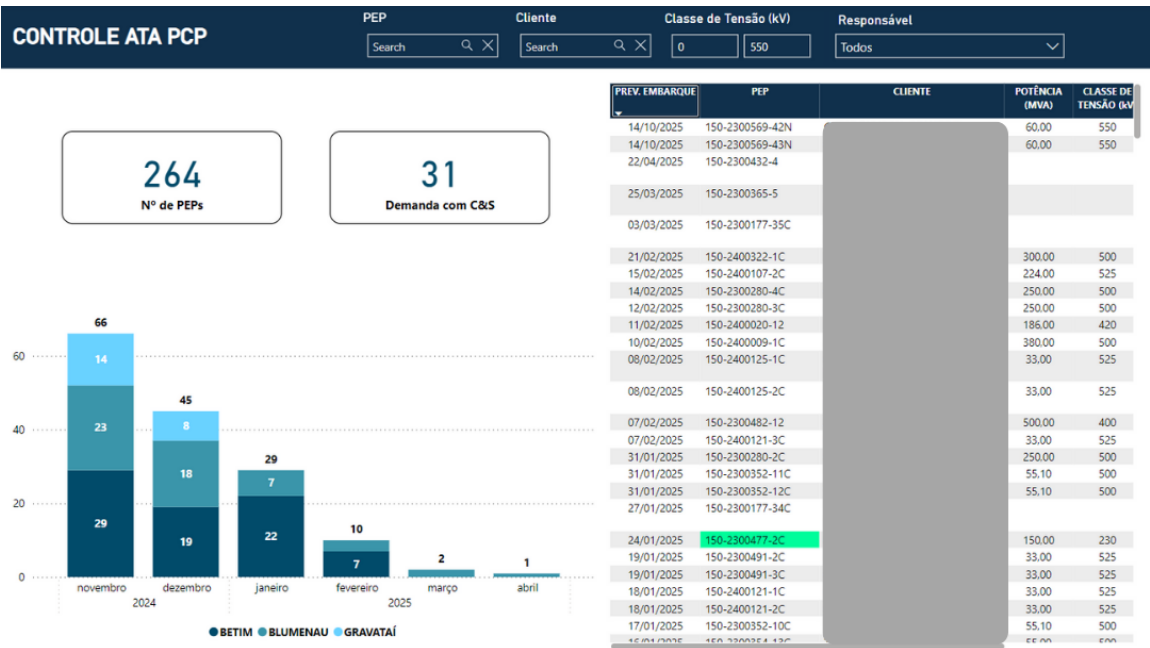
4.2.1 Contratações

O cenário anterior à implementação dos indicadores para a área de contratações da célula era marcado pela ausência de qualquer tipo de controle ou acompanhamento

estratégico sobre as demandas futuras. Não existiam ferramentas que permitissem planejar essas possíveis demandas com base em dados concretos. Isso gerava uma série de problemas operacionais, como atrasos na mobilização de equipes, alocação ineficiente de recursos e gastos extras devido à necessidade de atender às demandas de última hora. As decisões eram tomadas de maneira passiva, com base em informações vindas nas demandas, o que impossibilitava uma visão estruturada sobre as necessidades futuras da célula.

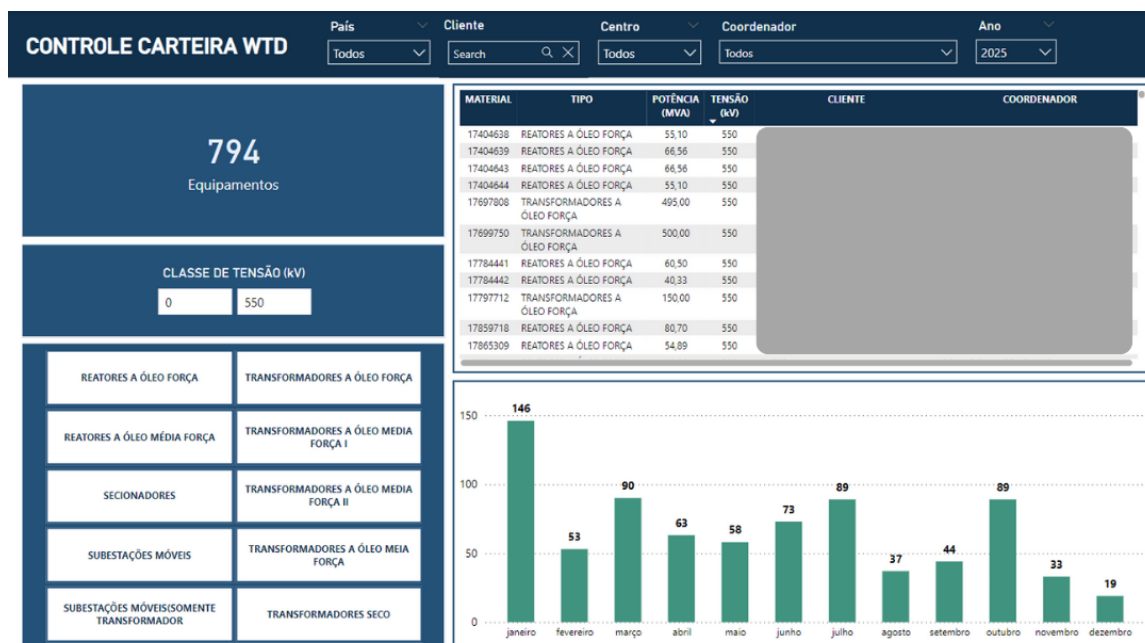
A falta de um sistema estruturado para prever demandas e organizar as contratações resultava em desperdício de tempo e dinheiro. Por exemplo, a ata do PCP, que contém informações essenciais sobre a previsão de saída de equipamentos, não era utilizada. A carteira de vendas, que já apresenta os equipamentos programados para os próximos anos, também não era explorada para alinhar a quantidade de possíveis contratações futuras com as demandas previstas. Com isso, a célula de Comissionamento tinha muita dificuldade em planejar futuras demandas, ficando sempre dependente de improvisações e soluções de última hora. Com a implementação dos novos dados coletados, a equipe passou a contar com telas que transformaram a maneira de planejar e acompanhar os processos. Nas Figuras 43 e 44, estão as *dashboards* contendo as demandas da ata do PCP e da carteira de vendas, respectivamente.

Figura 43 – *Dashboard* no Power BI dos dados de ata do PCP.



Fonte: O autor

A tela da ata do PCP permite visualizar os transformadores com previsão de saída, ajudando a organizar as equipes de montagem de forma eficiente. É possível também filtrar as demandas por unidade, cliente, tensão, coordenador da venda e pelo código PEP. As demandas que já têm equipes contratadas aparecem em verde na coluna do PEP na tabela.

Figura 44 – *Dashboard* no Power BI dos dados da carteira de vendas de equipamentos.

Fonte: O autor

Já a tela da carteira permite visualizar os equipamentos que têm a previsão de saída das fábricas em um prazo maior e mais longo. Além dos filtros indicados para a ata do PCP, na carteira também pode-se filtrar por país de destino do equipamento e por tipo de equipamento (reator, transformador a óleo, transformador a seco, seccionadora, entre outros).

Com a utilização dessas telas pela equipe de contratação, houve uma redução na necessidade de improvisos e uma maior previsibilidade no gerenciamento de recursos, deixando o processo de contratação muito mais ativo do que passivo devido à rastreabilidade das demandas. Além disso, os custos adicionais causados pela falta de planejamento foram minimizados, trazendo mais eficiência aos processos da célula.

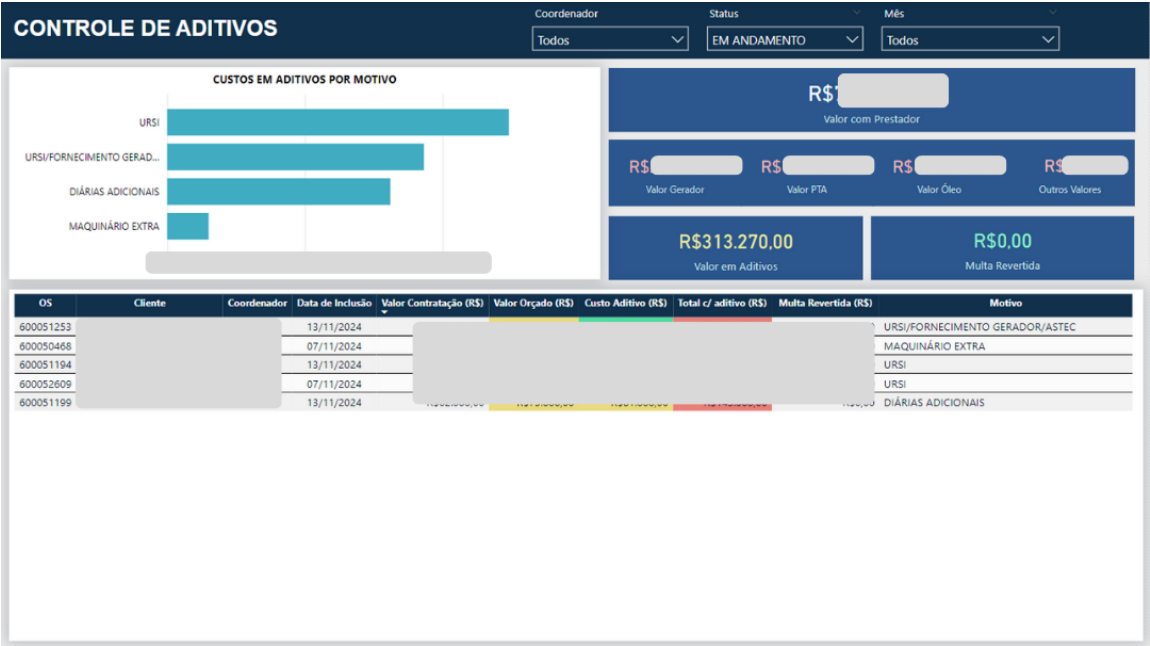
4.2.2 Obras em execução

O cenário anterior ao desenvolvimento dos indicadores para as obras em execução era separado em dois aspectos. Para os custos com aditivos durante a execução das obras, não havia nenhum controle desses dados. Já para o controle dos custos das obras encerradas, havia um controle completamente manual, demorado e suscetível a erros.

Em relação aos custos com aditivos durante o andamento das obras, com a presença do executável para o cadastro dos custos, houve um início muito promissor para o controle desses custos a um médio e longo prazo. À medida que vão sendo cadastrados novos aditivos, mais interessantes serão os dados a serem analisados junto a *dashboard* no Power BI. Essa tela está representada na Figura 45.

Na tela é possível acompanhar as obras que tiveram o seu valor orçado superado

Figura 45 – *Dashboard* no Power BI dos dados inseridos no executável de aditivos.



Fonte: O autor

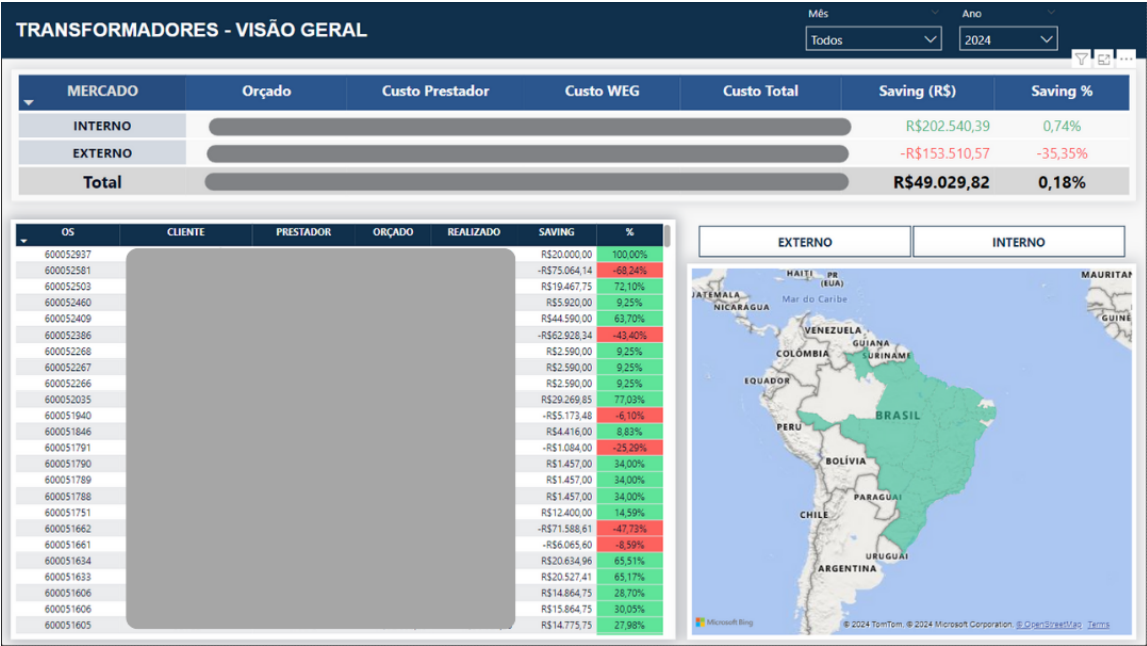
devido à presença de aditivos (coluna do valor realizado em vermelho). Também pode-se acompanhar os custos por motivo, coordenador e intervalo de tempo. Outro aspecto são os custos totais planejados, com gerador, PTA, óleo, que também não tinham controle algum antigamente. Além disso, também é possível ver o valor em multas que foi evitado devido aos custos extras com os aditivos para cumprir com prazos contratuais.

Passando agora para a análise dos custos mensais com obras encerradas, anteriormente, o processo manual para adquirir os custos, separar em prestador e empresa, completar tabela demorava um turno inteiro todos os meses, ou seja, cerca de 4 horas. Além disso, a chance de passar algum valor incorreto era enorme caso esse processo não fosse feito com atenção.

Com a implementação da interface que captura automaticamente os custos detalhados, o processo que antes levava 4 horas agora é concluído em menos de 20 minutos. Esse ganho em tempo se traduz em uma economia financeira baseada nas horas de trabalho que foram economizadas. Além disso, tornou-se possível filtrar esses custos de inúmeras novas maneiras, proporcionando uma análise muito mais rica e eficiente em comparação ao cenário anterior.

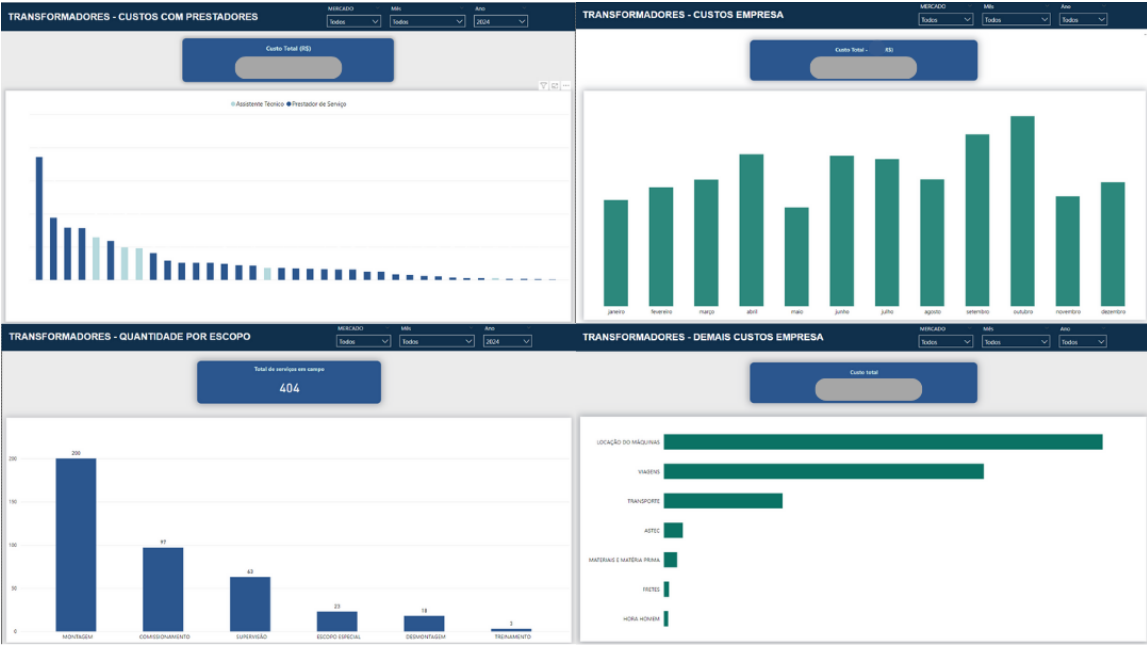
Ao adquirir esses custos no mês e atribuir nas bases de dados geral, óleo e aditivos, basta atualizar o Power BI e visualizar os novos dados aparecendo nas diferentes telas. As Figuras 46, 47, 48 e 49 mostram, respectivamente, as telas de custos gerais, custos específicos (por prestador, por escopo, pela empresa e por demais custos), custos com carretas de óleo e custos com aditivos.

Figura 46 – *Dashboard* no Power BI dos custos gerais das obras encerradas.



Fonte: O autor

Figura 47 – *Dashboards* no Power BI dos custos específicos das obras encerradas.



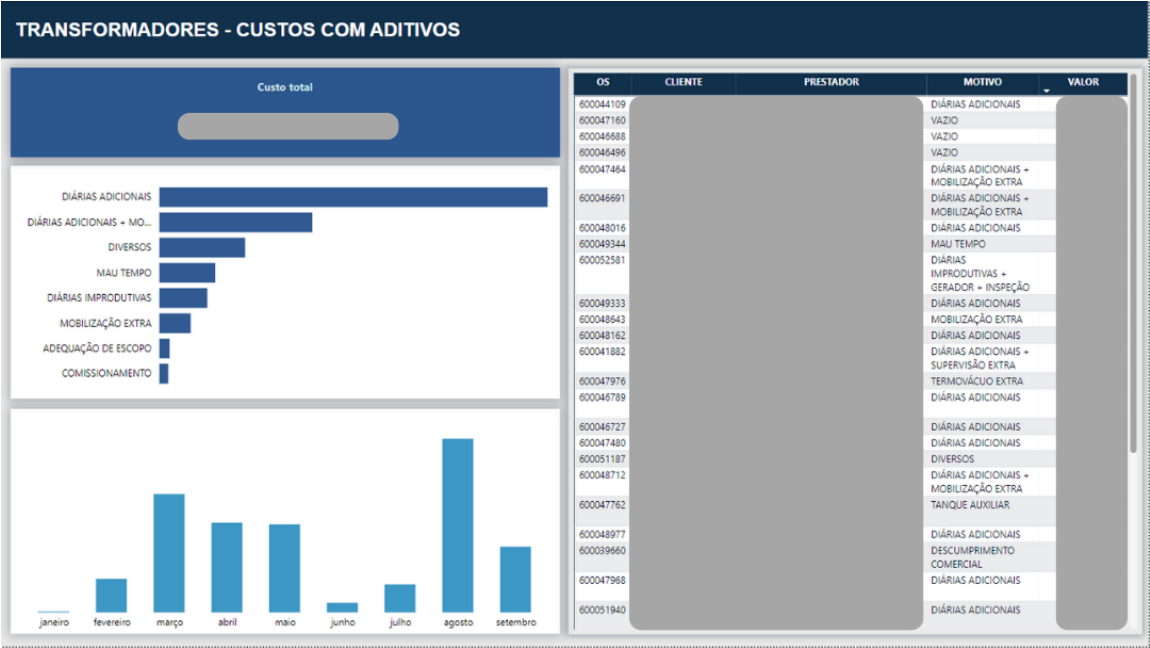
Fonte: O autor

Figura 48 – *Dashboard* no Power BI dos custos com carretas de óleo nas obras encerradas.



Fonte: O autor

Figura 49 – *Dashboard* no Power BI dos custos com aditivos nas obras encerradas.



Fonte: O autor

5 CONCLUSÃO

A integração de dados em empresas que utilizam sistemas de diferentes plataformas é um obstáculo significativo, especialmente quando esses sistemas não foram projetados para comunicação direta. Essa dificuldade é ampliada pela existência de diversas fontes de informações, como planilhas locais, bancos de dados internos e mensagens de e-mail, que frequentemente resultam em dados dispersos e inconsistentes. Essa fragmentação não apenas dificulta a análise integrada, mas também compromete a agilidade e a confiabilidade dos processos, tornando essencial o desenvolvimento de soluções capazes de centralizar e organizar esses dados de forma eficiente.

O objetivo deste trabalho foi desenvolver e implementar uma solução integrada para a automação da coleta, organização e análise de indicadores, com foco nas áreas de vendas e comissionamento. Por meio de ferramentas como VBA, Python, SAP e Power BI, buscou-se otimizar processos manuais, reduzir erros humanos e aumentar a eficiência operacional.

Os resultados obtidos demonstraram que a automação proposta é eficaz na centralização e análise de dados provenientes de diferentes fontes. A integração de sistemas possibilitou a criação de *dashboards* interativas e intuitivas, facilitando a visualização e interpretação de indicadores para a tomada de decisões. Além disso, o desenvolvimento de *scripts* para extração automatizada de dados do SAP e integração com pastas de e-mail no Outlook mostrou-se eficiente para aperfeiçoar processos que eram morosos e ou sequer existiam.

Entretanto, embora a solução tenha se mostrado robusta, alguns desafios ainda precisam ser considerados, especialmente relacionados à compatibilidade entre sistemas e à necessidade de acompanhamento dos usuários para melhor aproveitamento das ferramentas desenvolvidas. Outro ponto importante é a dependência de estruturas computacionais locais, o que pode limitar a escalabilidade da solução.

Como sugestões para trabalhos futuros, recomenda-se explorar a integração com tecnologias de bancos de dados em nuvem, permitindo uma maior escalabilidade e acessibilidade dos dados em tempo real. Além disso, seria interessante ampliar o escopo de automação para incluir o uso de aprendizado de máquina na análise preditiva dos indicadores, proporcionando percepções mais aprofundadas e decisões ainda mais apropriadas. Por fim, o aprimoramento das funcionalidades do sistema, aliado à utilização contínua dos usuários, é essencial para garantir a manutenção e o impacto positivo do sistema no ambiente empresarial.

REFERÊNCIAS

- ALVES, Jeancarlo Oliveira. Melhoria contínua da gestão de processos do sistema SAP ECC® por meio de programação VBA® e SAP script. **Repositório UFU**, Universidade Federal de Uberlândia, 2019.
- BANTILAN, Niels. pandera: Statistical Data Validation of Pandas Dataframes. *In*: SCIPY. [S.l.: s.n.], 2020. P. 116–124.
- BASTIANI, Jeison Arenhart De. **Fluxograma de Processo: Entenda o que é, saiba os benefícios e aprenda como fazer!** [S.l.: s.n.], 2012. Acesso em: 25 nov. 2024. Disponível em: <https://blogdaqualidade.com.br/fluxograma-de-processo/>.
- BHOSALE, ST *et al.* Sqlite: Light database system. **Int. J. Comput. Sci. Mob. Comput**, v. 44, n. 4, p. 882–885, 2015.
- BILYK, V. **Data Flow Diagrams (DFD) Explained**. [S.l.: s.n.], 2019. Acesso em: 30 out. 2024. Disponível em: <https://www.artofba.com/post/data-flow-diagrams-dfd-explained>.
- CAJAS, Carlos Germán Garcés. **Indicadores financieros para la toma de decisiones en la empresa Disgarta**: Proyecto de investigación previo a la obtención del título de Ingeniería en Contabilidad y Auditoría. 2019. B.S. thesis – Pontificia Universidad Católica del Ecuador.
- CARVALHO, Carlos Vitor de Alencar; MUSCI, Marcelo. Utilizando Python no ensino da construção naval: experiências de um curso de extensão. **Revista Naval e Oceânica**, n. 1, 2024.
- COZAR, David Sánchez. **Manejo de un dron mediante poses (I)**. 2023. B.S. thesis – Universitat Politècnica de Catalunya.
- FERNANDES, Djair Roberto. Uma contribuição sobre a construção de indicadores e sua importância para a gestão empresarial. **Revista da FAE**, v. 7, n. 1, p. 5, 2004.
- GONDIM, HWAS; AMBROSIO, APL. Esboço de fluxogramas no ensino de algoritmos. *In*: SN. WEI-WORKSHOP sobre Educação em Computação. [S.l.: s.n.], 2008. P. 109–117.

GUTIÉRREZ, Óscar del Hoyo. Optimización y automatización en la gestión de procesos con VBA Y SAP Script. **Biblioteca ETSI Industriales**, Industriales, 2017.

HIPÓLITO, Rafael Antônio Lima. Tutorial de uso do Power Bi para criação de um Dashboard exemplificado em um caso real. **Repositório UFU**, Universidade Federal de Uberlândia, 2023.

JÓIA, Andréia Chudrik; WERNER, Ilvili Andrea. Diagrama de fluxos de dados. **Revista Terra & Cultura: Cadernos de Ensino e Pesquisa**, v. 15, n. 30, p. 46–69, 2021.

KREIBICH, Jay. **Using SQLite**. [S.l.]: "O'Reilly Media, Inc.", 2010.

MANZUETO, Mauricio Santos. **Automation of process**: The influence of software on organizational routines. [S.l.], 2016.

MARQUES, Célio Gonçalo *et al.* Ensino da Programação através da Linguagem Algorítmica e Fluxográfica. *In*: IEEE, SOCIEDADE DE EDUCAÇÃO: CAPÍTULOS ESPANHOL E PORTUGUÊS. TICAI 2009. TIC's para a Aprendizagem da Engenharia. [S.l.: s.n.], 2011.

MCKINNEY, Wes *et al.* pandas: a foundational Python library for data analysis and statistics. **Python for high performance and scientific computing**, Seattle, v. 14, n. 9, p. 1–9, 2011.

MICHEL, Mozart Fayán *et al.* Roteiro de cálculo implementado em Excel/VBA para dimensionamento de lodos ativados. **Universidade Federal de Santa Catarina**, Florianópolis, SC., 2021.

MICROSOFT LEARN. **Automatizando o Outlook de um aplicativo do Visual Basic**. [S.l.: s.n.], 2023. Acesso em: 02 nov. 2024. Disponível em: <https://learn.microsoft.com/pt-br/office/vba/outlook/concepts/getting-started/automating-outlook-from-a-visual-basic-application>.

NEVES, David. **Power BI: Diferença entre dashboards e relatórios**. [S.l.: s.n.], 2024. Acesso em: 21 out. 2024. Disponível em: <https://www.alura.com.br/artigos/dashboards-vs-relatorios?srsId=AfmBOop6CkcmpoiitiU-nIVIselCr2OZ35g1VQfm6xgqNXhjiZq-1Zhc>.

PAULISTA, Paulo Henrique; ALVES, Raphaelly Antunes. Ferramentas da qualidade: uma revisão bibliográfica e análise de publicações no ENEGEP. **III Simpósio de Engenharia de Produção**, Universidade Federal de Campina Grande, 2015.

PRESSMAN, Roger S. **Engenharia de Software**. 6. ed. Nova York, EUA: McGraw-Hill, 2006.

RIBEIRO, Maria Francisca Borges da Gama de Freitas. **Os Sistemas de Business Intelligence nas Organizações: Construção de um Dashboard em Power BI**. 2023. Diss. (Mestrado) – Centro Tecnológico da Cerâmica e do Vidro (CTCV).

ROSA, Gabriel Chicoli Nunes. Sistema de análise de dados e faturamento. **Repositório UNESP**, Universidade Estadual Paulista (Unesp), 2024.

SANTOS, Joana Filipa de Nóbrega. **Automatização de Tarefas em VBA: Projeto no Santander Totta**. 2015. Tese (Doutorado) – Instituto Politécnico de Setúbal. Escola Superior de Ciências Empresariais.

SEETHA, Harshavardhan; TIWARI, Vimal; ANUGU, Kartik Reddy; MAKKA, DS; KARNATI, DR. A gui based application for pdf processing tools using python & customtkinter. **Int. J. Res. Appl. Sci. Eng. Technol**, 2023.

SILVA, João Vítor Gabriel. Otimização de processos e indicadores através de VBA e SAP Script, orientados a Agile e Lean. **Repositório UFU**, Universidade Federal de Uberlândia, 2021.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo, Brasil: Pearson, 2011.

SOUZA, Rafael Rodrigues de. **Diagrama de fluxo de dados DFD**. [S.l.: s.n.], 2016. Acesso em: 22 out. 2024. Disponível em:
<https://pt.slideshare.net/slideshow/diagrama-de-fluxo-de-dados-dfd/65503105>.

UNIVERSIDADE FERNANDO PESSOA. **Análise e Projeto em Sistemas de Informação**. [S.l.: s.n.], 1997. Apêndice A. Acesso em: 25 out. 2024. Disponível em:
http://homepage.ufp.pt/lmbg/reserva/tm_06.pdf.