



UNIVERSIDADE FEDERAL DE SANTA CATARINA COMO REQUISITO
CENTRO TECNOLÓGICO, DE CIÊNCIAS EXATAS E EDUCAÇÃO
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Augusto Henry Gasda

Sistema Integrado para Gestão de Atendimentos em Assistência Técnica

Blumenau
2024

Augusto Henry Gasda

Sistema Integrado para Gestão de Atendimentos em Assistência Técnica

Trabalho de Conclusão de Curso do 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.
Orientadora: Prof. Dra. Janaina Goncalves Guimaraes, Dra.

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

Gasda, Augusto Henry
Sistema Integrado para Gestão de Atendimentos em
Assistência Técnica / Augusto Henry Gasda ; orientadora,
Janaina Gonçalves Guimarães, 2024.
45 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. Engenharia de
Controle e Automação. 3. Gestão de Atendimentos. 4.
Automação de Processos. 5. Plataforma Integrada. I.
Guimarães, Janaina Gonçalves. II. Universidade Federal de
Santa Catarina. Graduação em Engenharia de Controle e
Automação. III. Título.

Augusto Henry Gasda

Sistema Integrado para Gestão de Atendimentos em Assistência Técnica

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, 01 de julho de 2024.

Banca Examinadora:

Prof. Dra. Janaina Goncalves Guimaraes,
Dra.
Orientadora

Prof. Beatriz Oliveira Câmara, Ma.
Avaliador
Instituição UnB

Prof. Mauri Ferrandin, Dr.
Avaliador
Instituição UFSC

Este trabalho é dedicado aos meus pais e estimados colegas de trabalho, cuja resiliência e companheirismo foi e tem sido uma grande fonte de apoio, inspiração e força ao longo de minha trajetória.

AGRADECIMENTOS

Agradeço, em primeiro lugar, aos meus pais, irmã e namorada pelo apoio emocional e educacional incondicional. Mesmo distantes, sempre demonstraram preocupação, estiveram prontos para me apoiar e ajudaram a superar os desafios mais difíceis.

À minha família, cujas perguntas constantes sobre o TCC, de certa forma, serviram como motivação e me deram força para seguir em frente e concluir este trabalho.

A minha orientadora, pelo apoio na organização, estrutura e ensinamentos deste trabalho e ao longo da faculdade. Agradeço pela oportunidade de ser seu orientando e pela paciência ao me acompanhar na jornada do TCC.

Ao meu gestor, que não apenas me apoiou, mas também motivou toda a equipe. E foi um pilar para a conclusão deste projeto, e a união entre nós, colegas de equipe não seria tão forte e tão boa.

E aos meus incansáveis e guerreiros colegas de trabalho, que, mesmo sobrecarregados, sempre estiveram dispostos a me auxiliar tecnicamente e a testar o projeto. Eles me mostraram e surpreendem todos os dias com a força e a capacidade de manter a união, mesmo diante dos desafios diários enfrentados em uma assistência técnica.

*“Faça o teu melhor
na condição que você tem,
enquanto você não tem condições melhores
para fazer melhor ainda!”
(Mário Sérgio Cortella)*

RESUMO

Este trabalho de conclusão de curso realizou uma análise detalhada dos desafios diários e dos fluxos operacionais em uma assistência técnica, investigando as plataformas, metodologias e sistemas empregados nos atendimentos. Esses processos frequentemente envolvem a coleta de uma variedade de informações e a interação com múltiplos setores, o que pode tornar as tarefas complexas e burocráticas.

Diante destes desafios, a principal proposta deste trabalho foi desenvolver uma ferramenta integrada que centraliza todas as informações necessárias em uma única plataforma. Este sistema não apenas visa simplificar o gerenciamento dos processos, mas também busca melhorar a qualidade do serviço ao cliente e otimizar a eficiência operacional. A implementação dessa ferramenta incluiu a criação de uma interface que facilita a visualização, análise das informações, a automação da extração de dados das plataformas *System Analysis Program Development (SAP)* e *Microsoft Outlook*, e a introdução de um sistema de checklist que aprimora o controle dos atendimentos. A ferramenta, desenvolvida em *Python*, emprega técnicas avançadas como recursividade, *threading* e integração com bancos de dados e outras aplicações.

Os resultados obtidos com a implementação da ferramenta foram notavelmente positivos. A aplicação provou ser uma solução eficaz para os problemas identificados e contribuiu para uma gestão eficiente dos processos de garantia e atendimento. A padronização dos procedimentos e a centralização das informações reduziram significativamente o tempo de treinamento para novos analistas e minimizaram as chances de erros, resultando em um serviço ao cliente ágil e confiável.

Palavras-chave: Gestão de Atendimentos, Assistência Técnica, Automação de Processos, Plataforma Integrada, *SAP*, *Outlook*, Eficiência Operacional, Qualidade do Serviço, *Python*, Análise de Dados.

ABSTRACT

This thesis conducted a detailed analysis of the daily challenges and operational flows in technical support, exploring the platforms, methodologies, and systems used in service provision. These processes often involve the collection of a variety of information and interaction with multiple sectors, which can make tasks complex and bureaucratic.

Faced with these challenges, the main proposal of this work was to develop an integrated tool that centralizes all necessary information on a single platform. This system not only aims to simplify the management of processes but also seeks to improve the quality of customer service and optimize operational efficiency. The implementation of this tool included the creation of an interface that facilitates the visualization and analysis of information, the automation of data extraction from *System Analysis Program Development* (SAP) and Microsoft Outlook platforms, and the introduction of a checklist system that enhances the control of service operations. Developed in Python, the tool employs advanced techniques such as recursion, threading, and integration with databases and other applications.

The results obtained from the implementation of the tool were remarkably positive. The application proved to be an effective solution for the identified problems and contributed to more efficient management of warranty and service processes. The standardization of procedures and centralization of information significantly reduced the training time for new analysts and minimized the chances of errors, resulting in a more agile and reliable customer service.

Keywords: Service Management, Technical Assistance, Process Automation, Integrated Platform, *SAP*, *Outlook*, Operational Efficiency, Service Quality, *Python*, Data Analysis.

LISTA DE FIGURAS

Figura 1 – Estrutura exemplo de nota de remessa <i>Microsoft Outlook</i> com informações básicas sobre o atendimento	15
Figura 2 – Transações no sistema <i>SAP</i> para coleta de informações	16
Figura 3 – O ciclo essencial de gestão de processos.	19
Figura 4 – Implementação de cálculo fatorial em <i>Python</i>	23
Figura 5 – Estrutura do banco de dados utilizado no sistema	30
Figura 6 – Diagrama salvamento utilizando a estratégia dual-database	31
Figura 7 – Visão geral da interface gráfica da ferramenta de gestão de processos .	32
Figura 8 – Interface dos filtros	33
Figura 9 – Interface de atalhos	34
Figura 10 – Interface do <i>checklist</i>	35
Figura 11 – Diagrama de salvamento automático das informações inseridas no programa	36
Figura 12 – Interface de Filtros de Colunas da Tabela	37
Figura 13 – Diagrama da coleta de dados simplificado	39

SUMÁRIO

1	INTRODUÇÃO	12
1.1	CONTEXTUALIZAÇÃO	12
1.1.1	Fluxo da assistência técnica:	12
1.1.2	Gestão das Informações	14
1.2	OBJETIVOS	15
1.2.1	Objetivo Geral	15
1.2.2	Objetivos Específicos	15
2	FUNDAMENTOS TEÓRICOS E FERRAMENTAS	18
2.1	GESTÃO DE PROCESSOS	18
2.2	SISTEMA <i>SAP</i>	19
2.2.1	Automatizações no <i>SAP</i>	19
2.2.2	LINGUAGEM <i>PYTHON</i>	20
2.2.3	<i>Pandas</i>	20
2.2.4	<i>Pywin32</i>	20
2.2.5	<i>Openpyxl</i>	20
2.2.6	<i>Auto-py-to-exe</i>	21
2.2.7	<i>Customtkinter</i> e <i>Tkinter</i>	21
2.2.8	<i>Exchangelib</i>	21
2.3	BANCO DE DADOS	21
2.4	RECURSIVIDADE	22
3	DESENVOLVIMENTO DO SISTEMA	24
3.1	REQUISITOS DO PROJETO	24
3.1.1	Requisitos Funcionais:	24
3.1.2	Requisitos Não Funcionais:	24
3.2	METODOLOGIA	26
3.2.1	Análise do Problema	27
3.2.2	Definição das Funcionalidades	27
3.2.3	Desenvolvimento da Ferramenta	27
3.2.4	Coleta e Armazenamento de Dados	29
3.2.5	Tratamento de Erros	30
3.2.6	Testes e Validação	31
3.2.7	Controle de Versão	31
4	RESULTADOS	32
4.1	INTERFACE GRÁFICA	32
4.2	FILTROS	32
4.3	ATALHOS	33
4.4	<i>CHECKLIST</i>	34

4.4.1	Personalização das Colunas da Tabela	37
4.5	COLETA DE INFORMAÇÕES NA PLATAFORMA SAP	38
4.6	FUNCIONALIDADES EXTRAS	38
4.6.1	Exportação de Tabela para Microsoft Excel	38
4.6.2	Acesso aos <i>E-mails</i> Relacionados ao Atendimento	40
5	CONCLUSÃO	41
	REFERÊNCIAS	43

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Em um mundo cada vez mais industrializado, a manutenção de equipamentos de pequeno e grande porte é crucial para o sucesso de diversos setores, desde manufatura e energia até mineração e construção. A assistência técnica surge como um pilar fundamental para garantir o funcionamento ideal, a segurança e a longevidade desses equipamentos, minimizando o tempo de inatividade e maximizando a produtividade.

No cotidiano da assistência técnica, os desafios são constantes e multifacetados. As demandas variam desde o esclarecimento de dúvidas simples sobre a operação dos equipamentos até a resolução de problemas complexos e urgentes que exigem intervenção imediata. Para lidar com essa diversidade, a equipe de assistência técnica precisa ser altamente qualificada, com conhecimento técnico aprofundado e habilidades de resolução de problemas. Além disso, a pressão por prazos e a necessidade de minimizar o tempo de inatividade dos equipamentos exigem agilidade e eficiência na resposta às solicitações dos clientes.

A empresa em questão, com seus 18 analistas dedicados ao atendimento direto aos clientes, enfrenta o desafio de gerenciar uma carga de trabalho crescente devido à expansão do mercado e à demanda por equipamentos industriais. O volume de chamados e a variedade de problemas, em que a possível disfunção de um dos equipamentos é capaz de afetar todo o processo fabril, impõem uma pressão considerável sobre a equipe.

Diante desse cenário, a otimização dos processos e das ferramentas de gestão torna-se imperativa. A adoção de soluções que agilizem o atendimento, facilitem a comunicação entre os analistas e os clientes, e permitam o acompanhamento em tempo real do *status* dos chamados é essencial para manter a eficiência e a competitividade da empresa. Além disso, a organização das informações de forma clara e estruturada é fundamental para a análise de dados e a identificação de oportunidades de melhoria.

1.1.1 Fluxo da assistência técnica:

O processo de assistência técnica tem início com a recepção dos pedidos de serviço, que chegam principalmente via *e-mail*. Esses pedidos podem ser originados internamente, pela equipe de campo que trabalha na montagem e instalação dos equipamentos, ou externamente, pelos clientes que utilizam os equipamentos em suas operações.

As solicitações internas geralmente surgem durante as atividades de campo, quando os técnicos identificam problemas ou necessidades de ajustes nos equipamentos vendidos pela empresa. Essas situações podem ocorrer durante a instalação inicial do equipamento. Os técnicos de campo, com seu conhecimento prático sobre o funcionamento dos equipamentos, são responsáveis por relatar os problemas de forma detalhada, incluindo sintomas,

possíveis causas e condições de operação no momento da ocorrência.

As solicitações externas, por sua vez, são provenientes dos clientes que adquiriram os equipamentos da empresa e enfrentam dificuldades operacionais, falhas técnicas ou necessitam de suporte para garantir o desempenho ideal dos equipamentos. Essas dificuldades podem se manifestar de diversas formas, como alarmes, paradas inesperadas, baixo rendimento, problemas de qualidade no produto final ou dúvidas sobre a operação e a manutenção. Os clientes podem entrar em contato com a assistência técnica por telefone ou *e-mail*, fornecendo informações sobre o problema e o histórico de uso do equipamento.

Independentemente da origem, todos os pedidos de serviço passam por uma triagem inicial, na qual as informações básicas são verificadas e organizadas. Essa etapa é crucial para garantir que a equipe de assistência técnica tenha os dados mínimos necessários para iniciar a análise do problema, como nome do cliente, tipo de equipamento, número de série, descrição do problema e dados de contato. A triagem também ajuda a identificar possíveis duplicidades de chamados, evitando o desperdício de recursos.

Após a triagem, os pedidos são submetidos a uma avaliação preliminar para determinar a gravidade e a urgência do problema. Analistas experientes verificam as informações fornecidas e, se necessário, entram em contato com o cliente para obter detalhes adicionais. Essa avaliação leva em consideração o impacto do problema nas operações do cliente, o risco de segurança, a complexidade técnica e a disponibilidade de recursos para o atendimento. Com base nessa análise, os pedidos são classificados em categorias de prioridade, como "urgência crítica", "alta prioridade", "média prioridade" e "baixa prioridade", permitindo que a equipe direcione seus esforços para os casos mais urgentes.

Com a avaliação e a classificação de prioridade concluídas, uma nota de serviço é criada na plataforma *SAP* (SAP SE, 2023). Essa nota funciona como um registro formal do atendimento, contendo todas as informações relevantes, como número do pedido, dados do cliente, descrição do problema, categoria de prioridade, *status* do atendimento e histórico de interações com o cliente. A ordem de serviço é então atribuída a um analista com a experiência e disponibilidade adequadas para lidar com o caso.

A partir daí, o analista responsável realiza uma análise detalhada do problema, utilizando as informações da ordem de serviço, o histórico do equipamento e seu conhecimento técnico. Ele pode consultar manuais, desenhos, diagramas elétricos e outros recursos para entender o problema e suas possíveis causas. Em alguns casos é necessário o retorno do equipamento para testes e simulações para realizar um diagnóstico mais detalhado.

Em casos de solicitação de garantia, o analista verifica se o problema está coberto pelos termos contratuais e se o equipamento está dentro do período de cobertura. Se o problema for originado do equipamento ou do processo de fabricação, o analista coordena o atendimento com os setores responsáveis, como o departamento de qualidade, o setor de peças de reposição e a equipe de campo. Essa coordenação envolve a comunicação entre os diferentes setores, o agendamento de visitas técnicas, a solicitação de peças e o

acompanhamento do processo de reparo ou substituição do equipamento.

1.1.2 Gestão das Informações

A gestão das informações de atendimento é um elemento crítico para o sucesso da assistência técnica. Tradicionalmente, essa gestão era realizada por meio do *software Microsoft Outlook* (MICROSOFT CORPORATION, 2023), no qual os dados eram organizados em pastas específicas para cada atendimento. Dentro de cada pasta, havia uma nota de remessa, um documento que continha informações relevantes sobre o atendimento, como o número do pedido, os dados do cliente, a descrição do problema, o histórico de interações e as ações tomadas pela equipe de assistência técnica.

A nota de remessa servia como um registro centralizado das informações do atendimento, facilitando o acompanhamento do caso e a comunicação entre os membros da equipe. No entanto, a gestão das informações no *Microsoft Outlook* apresentava algumas limitações. A organização das pastas e das notas de remessa podia se tornar complexa e confusa à medida que o volume de atendimentos aumentava, dificultando a localização de informações específicas e o acompanhamento do histórico dos atendimentos. Além disso, a busca por informações específicas em meio a um grande número de *e-mails* e documentos podia ser demorada e trabalhosa, impactando negativamente na eficiência da equipe.

Outra dificuldade enfrentada era a integração das informações entre o *Microsoft Outlook* e a plataforma *SAP*, essencial para o registro e consulta de dados nos atendimentos, como histórico de equipamentos e detalhes de contratos. A coleta e o registro manual desses dados no *Microsoft Outlook* não apenas consumiam tempo significativo dos analistas, que precisavam navegar por diversas transações e telas no *SAP*, mas também aumentavam o risco de erros e inconsistências, prejudicando a tomada de decisões e atrasando os processos de atendimento. Essa falta de integração entre os sistemas dificultava a análise eficaz e a identificação de oportunidades de melhoria nos processos.

Essa complexidade na coleta de dados acarretava em um impacto significativo na eficiência da equipe de assistência técnica. O tempo gasto na busca e organização manual das informações poderia ser melhor utilizado em outras atividades, como a análise dos problemas, o contato com os clientes e a busca por soluções. Além disso, a ocorrência de erros no preenchimento das notas de remessa podia gerar retrabalho, atrasos e até mesmo insatisfação por parte dos clientes, o que poderia comprometer a imagem da empresa e a fidelização dos clientes. Com o intuito de ilustrar a estrutura de uma nota de remessa no *Microsoft Outlook*, a Figura 1 apresenta um exemplo com as informações básicas sobre o atendimento.

A crescente demanda por atendimentos, somada à complexidade e ineficiência dos processos de gestão das informações, evidenciava a necessidade de uma solução mais robusta e automatizada. A busca por uma ferramenta que integrasse os dados do *SAP* e do *Microsoft Outlook*, agilizasse a coleta de informações e minimizasse os erros tornou-

Figura 1 – Estrutura exemplo de nota de remessa *Microsoft Outlook* com informações básicas sobre o atendimento

Postada em
Conversa em Primeiro Atendimento
Assunto Primeiro Atendimento

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34

Equipamentos	Código do produto	Nº de Série	PEP	Unidade Fab.	Localização do equipamento
-	-	-	-	-	-
-	-	-	-	-	-

• ANOTAÇÕES
|
Status Atendimento:
OBS
Contato cliente:

Fonte: Próprio autor

se uma prioridade. O desenvolvimento de um sistema que automatizasse a extração de dados do *SAP* e os organizasse de forma clara e acessível no *Outlook* poderia otimizar significativamente o trabalho da equipe de assistência técnica, liberando os analistas para se concentrarem em atividades mais estratégicas e garantindo um atendimento mais rápido e eficiente aos clientes. A Figura 2 demonstra algumas transações *SAP* utilizadas, nela é possível notar a quantidade de transações necessárias e o volume de informações utilizada nos atendimentos, reiterando a importância de uma ferramenta de gestão.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

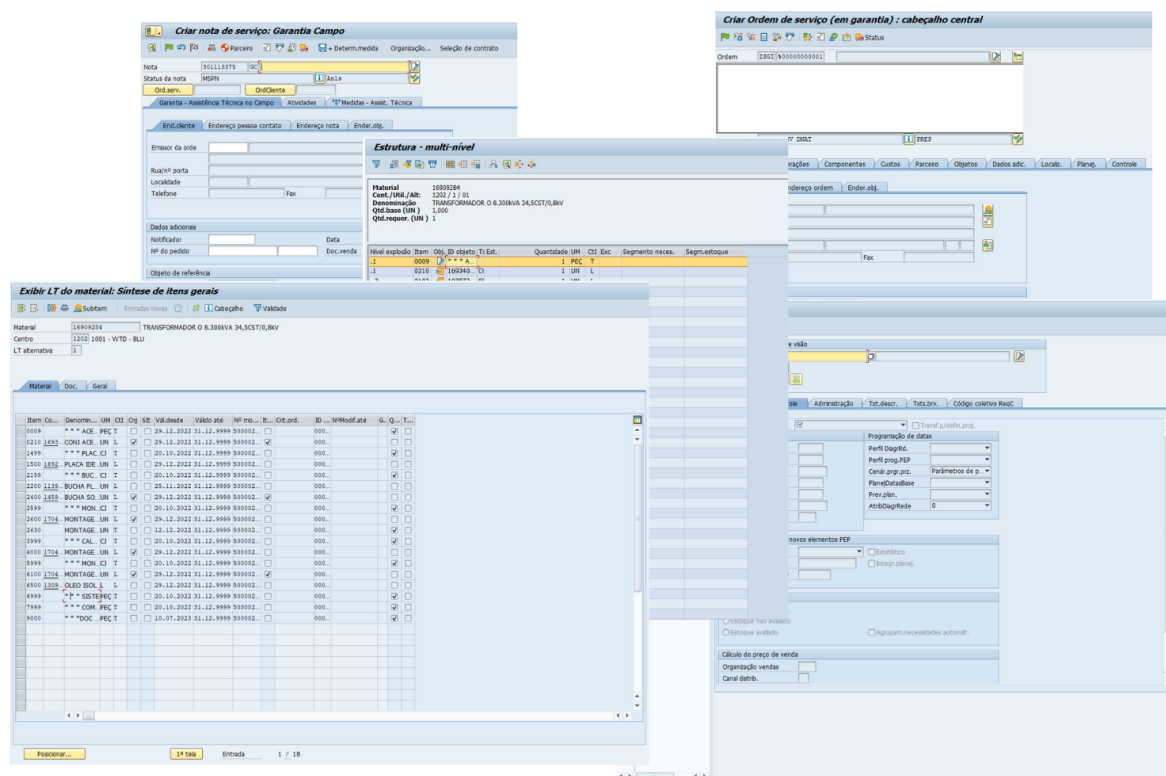
Com base nos desafios identificados, o propósito deste projeto é desenvolver um sistema integrado para a gestão de atendimentos em assistência técnica, centralizando todas as informações pertinentes em uma única plataforma. Este sistema será projetado para integrar-se perfeitamente à rotina operacional da seção, automatizando a coleta de informações e implementando processos operacionais mais eficientes. O sistema não apenas facilitará o gerenciamento eficaz dos processos de atendimento, mas também melhorará significativamente a qualidade do serviço ao cliente e otimizará o desempenho operacional. Além disso, o sistema estará alinhado com os objetivos estratégicos da empresa, contribuindo para a sua competitividade e sustentabilidade no longo prazo.

1.2.2 Objetivos Específicos

Para alcançar o objetivo geral, os seguintes objetivos específicos foram estabelecidos:

- **Desenvolver uma ferramenta que extrai automaticamente informações das plataformas *SAP* e *Microsoft Outlook*:** A extração manual de dados é um processo demorado e propenso a erros, demandando tempo e atenção dos analistas, que poderiam ser utilizados em atividades mais estratégicas, como a análise dos

Figura 2 – Transações no sistema SAP para coleta de informações



Fonte: Próprio autor

problemas e a busca por soluções. A automatização dessa tarefa não apenas libera os analistas para se concentrarem em atividades mais importantes, como também reduz o risco de erros humanos no processo de coleta de dados.

- **Criar um banco de dados que armazene todas as informações obtidas:** Um banco de dados centralizado e estruturado é essencial para facilitar a organização, o acesso e a consulta das informações relevantes para os atendimentos. Isso permite uma gestão mais eficiente dos dados, possibilitando a geração de relatórios e análises que auxiliarão na tomada de decisões estratégicas, como a identificação de problemas recorrentes em equipamentos ou a avaliação do desempenho da equipe.
- **Criar um *layout* claro e funcional para a visualização das informações coletadas, facilitando a análise e a decisão sobre quais processos priorizar:** A interface do sistema deve ser intuitiva e de fácil utilização, permitindo que os analistas visualizem as informações de forma clara e organizada. Recursos como filtros, ordenação e agrupamento de dados são importantes para facilitar a identificação dos atendimentos mais urgentes e a priorização das tarefas, agilizando o processo de tomada de decisão e garantindo que os problemas mais críticos sejam resolvidos em primeiro lugar.

- **Implementar um sistema de *checklist* para melhorar o controle dos atendimentos:** Um *checklist* integrado ao sistema garantirá que todas as etapas do atendimento sejam realizadas de acordo com os padrões estabelecidos pela empresa. Isso contribuirá para a padronização dos processos, a redução de erros e a melhoria da qualidade do serviço prestado.
- **Integrar automações relacionadas aos processos de atendimento, utilizando os dados coletados:** A automatização de tarefas demoradas, como o download automático de desenhos técnicos e busca de documentos dos supervisores de campo, visa otimizar o uso do tempo dos analistas. Ao eliminar a necessidade de processos manuais para acessar e gerenciar esses documentos críticos, os analistas poderão focar em atividades de maior complexidade técnica e valor estratégico. A ferramenta, utilizando dados previamente coletados, facilitará essas automações, aumentando a eficiência operacional e reduzindo a probabilidade de erros associados à gestão manual de documentos importantes.
- **Melhorar significativamente a qualidade e eficiência dos serviços prestados:** O objetivo final do projeto é melhorar a qualidade e a eficiência dos serviços de assistência técnica. Com a implementação da ferramenta, espera-se que os atendimentos sejam mais rápidos e precisos, resultando em maior satisfação dos clientes e em um aumento da competitividade da empresa. A ferramenta deve contribuir para a redução do tempo de resolução dos problemas, a diminuição dos erros e a melhoria da comunicação com os clientes, impactando positivamente a percepção da qualidade do serviço.

2 FUNDAMENTOS TEÓRICOS E FERRAMENTAS

2.1 GESTÃO DE PROCESSOS

A gestão de processos, como um conjunto de metodologias e ferramentas, proporciona uma visão abrangente do ambiente operacional de uma organização (CRUZ, 2008). Ela permite a identificação, modelagem, análise, melhoria e monitoramento dos processos de negócio, com o objetivo de otimizar o desempenho, aumentar a eficiência e garantir a qualidade dos produtos e serviços oferecidos. A gestão de processos engloba diversas áreas da empresa, desde a produção e logística até o atendimento ao cliente e o desenvolvimento de novos produtos. Através da análise sequencial das etapas de um processo, é possível identificar gargalos, redundâncias e atividades que não agregam valor, permitindo a implementação de melhorias e a otimização contínua (DAVENPORT, 1993). A gestão de processos também busca a padronização das atividades, garantindo que sejam executadas de forma consistente e eficiente, independentemente de quem as realize (JURAN, 1992).

A criação de um processo formal é o ponto de partida para o ciclo de gestão de processos, conforme ilustrado na Figura 3. Esse ciclo envolve o planejamento, a execução, o monitoramento e o controle do processo, com o objetivo de garantir que ele atinja os resultados esperados. O desempenho do processo é constantemente avaliado e comparado com as metas estabelecidas, que podem ser baseadas em indicadores de desempenho, expectativas dos clientes, *benchmarks* da concorrência ou outros critérios relevantes (HAMMER, 2013).

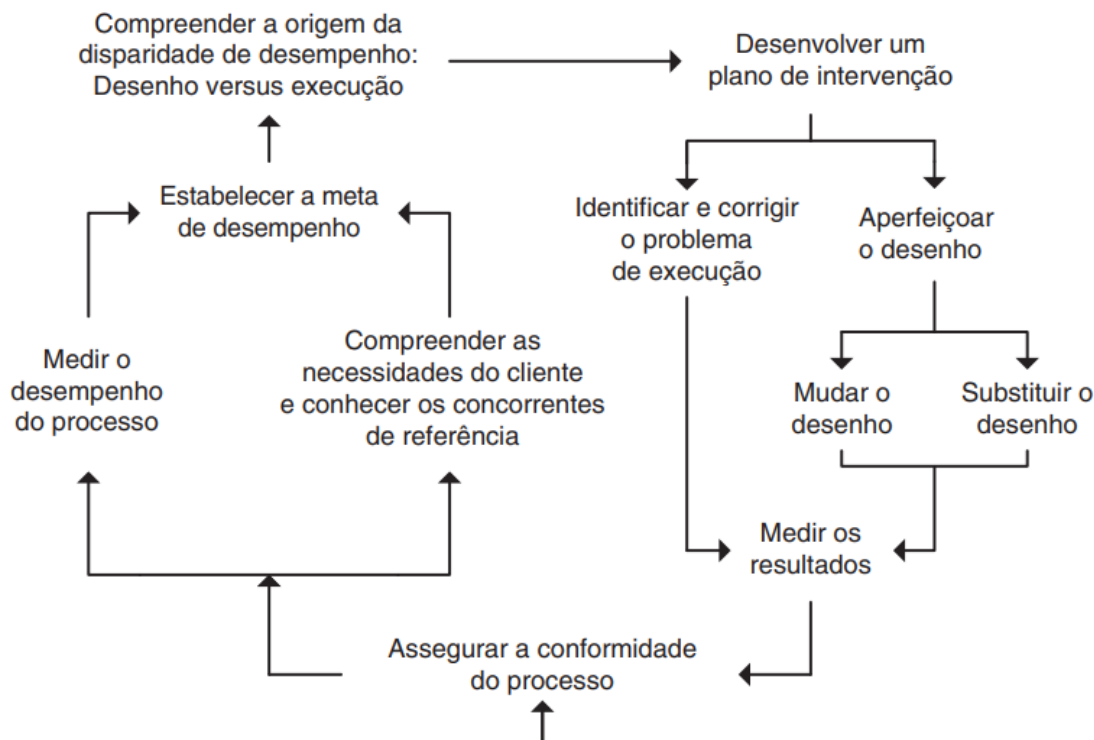
Se o processo não atingir as metas, as causas da falha devem ser investigadas e ações corretivas devem ser tomadas. As falhas podem ser de execução, devido a treinamento inadequado, recursos insuficientes ou de desenho, indicando falhas na estrutura do processo em si. No caso de falhas de execução, a causa raiz deve ser identificada e corrigida, enquanto falhas de desenho exigem uma reestruturação mais ampla do processo.

Após a intervenção corretiva, os resultados são avaliados e o ciclo se reinicia, buscando continuamente a otimização do processo e o alcance dos objetivos. A gestão de processos é um ciclo contínuo de melhoria, que busca a excelência operacional e a adaptação às mudanças do mercado e do ambiente de negócios.

No ambiente empresarial, a implementação de automações é uma estratégia fundamental para o aperfeiçoamento dos processos, visando aumentar a agilidade na produção, reduzir custos, simplificar tarefas e garantir a conformidade regulatória (HARMON, 2019). A automação de processos permite que as empresas executem tarefas de forma rápida, precisa e eficiente, liberando os funcionários para se concentrarem em atividades estratégicas e que agregam valor ao negócio.

A automação pode ser aplicada em diversas áreas da empresa, como finanças, contabilidade, produção, logística, atendimento ao cliente e recursos humanos. Por exemplo, é possível automatizar o processamento de faturas, a geração de relatórios, o controle

Figura 3 – O ciclo essencial de gestão de processos.



Fonte: (HAMMER, 2013)

de estoque, o agendamento de tarefas e o envio de *e-mails*. A automação de processos repetitivos e manuais não apenas reduz o tempo e o custo de execução dessas tarefas, mas também minimiza o risco de erros humanos e garante a consistência e a qualidade dos resultados.

2.2 SISTEMA SAP

O *System Analysis Program Development (SAP)* é uma plataforma fundamental para o planejamento e controle empresarial, sendo o cerne de uma infraestrutura abrangente (CRUZ, 2021). Ela possibilita a integração e gestão em diversas camadas organizacionais, incluindo áreas como gestão financeira, recursos humanos, gerenciamento de projetos, interações com fornecedores, logística de suprimentos, relacionamento com clientes e soluções customizadas adaptadas às necessidades específicas da organização. Além de facilitar o gerenciamento de informações em tempo real, o *SAP* oferece notáveis vantagens competitivas e benefícios substanciais, exigindo, no entanto, formação adequada dos colaboradores para seu uso eficaz.

2.2.1 Automatizações no SAP

O *SAP* também oferece a execução autônoma de tarefas, além da robusta capacidade de armazenamento de informações. Conforme destacado por Cirolini (CIROLINI,

2016), as automações no sistema *SAP* são viabilizadas pela linguagem de programação *Advanced Business Application Programming (ABAP)* (ABAP, 2021), que suporta interfaces tanto internas quanto externas. Essa flexibilidade facilita a implementação de soluções automatizadas essenciais para a eficiência operacional. Além disso, o sistema permite o desenvolvimento de indicadores para monitorar processos empresariais, a criação de interfaces com aplicações externas para melhor usabilidade pelos usuários e a automação de tarefas repetitivas e burocráticas. O resultado é a redução de custos operacionais e a minimização de erros nos processos.

2.2.2 LINGUAGEM PYTHON

Python (VAN ROSSUM, 1995) é amplamente reconhecida como uma linguagem ideal para a realização de tarefas em ambientes de trabalho burocráticos e controlados. Sua popularidade advém não apenas da facilidade de uso e da sintaxe clara, mas também de uma vasta comunidade de desenvolvedores e de um rico ecossistema de bibliotecas que ampliam suas aplicações (JARA, s.d). A seguir, detalham-se algumas das principais bibliotecas que exemplificam como o *Python* integra-se e automatiza diversas plataformas de maneira flexível e eficiente.

2.2.3 Pandas

Essencial para a manipulação e análise de dados, o *Pandas* (MCKINNEY, 2010) oferece estruturas de dados flexíveis e poderosas, como *DataFrames* e *Series*. Esta biblioteca facilita a importação, limpeza, transformação e análise de dados de diversas fontes, como arquivos *CSV*, *Microsoft Excel* (MICROSOFT CORPORATION, 2023) e bancos de dados *SQL*, demonstrando a capacidade de integração do *Python* com outras ferramentas.

2.2.4 Pywin32

Este módulo permite a interação direta com componentes do sistema operacional *Windows*, como a *Application Programming Interface (API)* do *Windows*, manipulação de processos e comunicação entre aplicativos. *Pywin32* (HAMMOND, 2023) é particularmente útil para automatizar tarefas no ambiente *Windows*, ilustrando como o *Python* pode conectar-se com sistemas operacionais específicos para melhorar a eficiência operacional (HAMMOND, 2000).

2.2.5 Openpyxl

Projetada para ler e escrever arquivos do *Microsoft Excel*, esta biblioteca permite a criação, edição e formatação de planilhas, além da manipulação de dados em células, linhas e colunas. *Openpyxl* (CLARK, 2023) é uma ferramenta valiosa para automatizar tarefas relacionadas ao *Microsoft Excel* e integrar dados com outras aplicações *Python*, mostrando

a versatilidade do *Python* em trabalhar com aplicativos populares (BARRANQUERO, 2014).

2.2.6 *Auto-py-to-exe*

Esta ferramenta simplifica o processo de converter *scripts Python* em arquivos executáveis autônomos para *Windows*, *macOS* ou *Linux*. Facilita a distribuição de aplicativos *Python*, eliminando a necessidade de os usuários finais terem *Python* instalado, o que exemplifica a escalabilidade e a adaptabilidade do *Python* em diferentes plataformas (KLOSS, 2019).

2.2.7 *Customtkinter* e *Tkinter*

Tkinter (FOUNDATION, 2023), a biblioteca padrão do *Python* para criação de interfaces gráficas, e *Customtkinter* (SCHIMANSKY, 2023), que oferece *widgets* mais modernos e personalizáveis, demonstram a flexibilidade do *Python* em criar interfaces interativas e visualmente atraentes. Estas ferramentas são fundamentais para projetar GUIs que podem melhorar a interação do usuário com automações *Python* (LUTZ, 2013).

2.2.8 *Exchangelib*

Permite a interação com servidores *Microsoft Exchange*, facilitando o acesso a *e-mails*, calendários, contatos, entre outros. Esta biblioteca é um exemplo de como o *Python* pode ser usado para automatizar tarefas de comunicação e gerenciamento de informações, integrando-se perfeitamente com sistemas de comunicação corporativa (MICROSOFT CORPORATION, 2023).

A versatilidade do *Python*, destacada pela capacidade de suportar múltiplos paradigmas de programação, incluindo programação orientada a objetos e paralelismo, permite que a linguagem seja adaptada para realizar tarefas complexas de maneira organizada, ajustando-se perfeitamente aos processos internos de uma empresa. Com a possibilidade de evoluir para atender a processos mais amplos, *Python* mostra-se uma escolha robusta e escalável para automações e integrações em ambientes corporativos.

2.3 BANCO DE DADOS

Um banco de dados é uma estrutura projetada para ser compartilhada e integrada, oferecendo uma solução altamente eficiente para armazenar conjuntos de dados do usuário, conforme definido por Rob (ROB, 2011). Para gerenciar e controlar o acesso a esses dados, utilizam-se os Sistemas de Gerenciamento de Banco de Dados (SGBD). Esses sistemas facilitam a interação entre os usuários e o banco de dados, processando solicitações e permitindo o compartilhamento dos dados por múltiplas aplicações e usuários de forma

eficiente e segura. Date (DATE, 2004) enfatiza que o SGBD é essencial para garantir a integridade e a segurança dos dados armazenados.

A escolha do tipo de banco de dados é importante para o desenvolvimento de sistemas modernos. Eles podem se apresentar em diferentes abordagens, como descreve Rob (ROB, 2011). Os Bancos de Dados Relacionais, como *MySQL* (ORACLE CORPORATION, s.d.) ou *PostgreSQL* (THE POSTGRESQL GLOBAL DEVELOPMENT GROUP, s.d.), organizam os dados em tabelas e permitem criar relações entre elas, garantindo integridade referencial e possibilitando consultas usando a linguagem *Structured Query Language (SQL)*. O *SQL* é a linguagem padrão para gerenciar bancos de dados relacionais e inclui componentes para manipulação e definição de dados, permitindo operações como consulta, inserção, atualização e exclusão de dados.

O *SQLite* (SQLITE DEVELOPMENT TEAM, 2024) é um exemplo de um sistema de banco de dados relacional leve que, diferentemente de sistemas mais robustos, não requer um processo de servidor separado, operando diretamente com arquivos de disco comuns. Esta característica simplifica sua implementação e oferece vantagens em termos de velocidade para operações de consulta e atualização de dados. *SQLite* é amplamente usado em aplicações para dispositivos móveis, como *Android* e *iOS*, e em dispositivos embarcados em diversas indústrias.

Além disso, existem os Bancos de Dados Não-Relacionais (NoSQL), nos quais os dados são organizados em estruturas como árvores (hierarquias), adequados para aplicações que exigem escalabilidade, flexibilidade de esquema e manipulação eficiente de grandes volumes de dados não estruturados. A Tabela 1 apresenta uma comparação entre os bancos de dados relacionais e não relacionais:

2.4 RECURSIVIDADE

A recursividade é uma técnica de programação caracterizada por uma função que pode chamar a si mesma para resolver problemas complexos, dividindo-os em subproblemas menores (RAMOS, 2018). Fundamentada no mecanismo básico de repetições nas linguagens funcionais, ela resulta em programas mais elegantes, simples e concisos. Na prática, a recursividade é aplicada por algoritmos, cuja função consiste em solucionar um problema através de instâncias menores do próprio problema (BALREIRA, 2022). Esses algoritmos têm códigos mais legíveis e compactos, dando agilidade para a manipulação de estrutura de dados. Um exemplo clássico de recursividade é o cálculo fatorial de um número. O fatorial de um número inteiro não negativo é o produto de todos os números inteiros positivos menores ou iguais a ele. Por exemplo, o fatorial de 5 é 120, pois $120 = 5 \times 4 \times 3 \times 2 \times 1$. Podemos implementar uma função recursiva para calcular o fatorial em *Python* da forma mostrada na figura 4.

Tabela 1 – Comparação entre Bancos de Dados Relacionais e Não-Relacionais

Aspecto	Banco de Dados Relacional	Banco de Dados Não Relacional (NoSQL)
Integridade de Dados	Alta, com garantias fortes devido ao controle de transações e relações	Menor, pode depender de técnicas específicas para manter a integridade
Escalonamento	Complexo e custoso, com dificuldade na inserção dinâmica de novos nós	Facilitado pela flexibilidade e ausência de esquema pré-definido
Consistência	Forte, com regras rigorosas garantindo a consistência das informações	Eventual, com consistência garantida apenas sob certas condições
Disponibilidade	Pode ser comprometida em sistemas distribuídos	Alta, especialmente em sistemas distribuídos devido à replicação nativa
Modelo de Dados	Estruturado em tabelas com esquema definido e rigidez	Modelos variados (documento, grafo, chave-valor) com esquemas flexíveis
APIs	Baseado em <i>SQL</i> para todas as operações	APIs variadas, frequentemente não baseadas em <i>SQL</i>
Tipo de Dados Armazenados	Melhor para dados estruturados e relacionamentos complexos	Ideal para dados semi-estruturados ou não estruturados
Custo de Implementação	Alto, especialmente para escalabilidade	Menor em comparação, especialmente para grandes volumes de dados
Flexibilidade	Limitada devido ao esquema rígido	Alta, permite mudanças rápidas e dinâmicas na estrutura de dados
Uso Ideal	Aplicações que requerem integridade e consistência de dados	Aplicações com grandes volumes de dados ou que necessitam de escalabilidade e disponibilidade alta

Fonte: (GARCIA, 2019)

Figura 4 – Implementação de cálculo fatorial em *Python*

```
>>>def fatorial(n):
...     if n <= 1:
...         return 1
...     return n * fatorial(n - 1)
```

Fonte: (UNIVERSIDADE FEDERAL FLUMINENSE, 2011)

A recursividade pode ser facilmente aplicada a uma ampla gama de problemas, incluindo aqueles que envolvem definições recursivas ou autossimilaridade. Entretanto, deve-se levar em consideração que chamadas recursivas podem consumir muita memória, especialmente para problemas que envolvem recursão profunda. Além disso, erros em funções recursivas podem ser difíceis de rastrear e depurar.

3 DESENVOLVIMENTO DO SISTEMA

3.1 REQUISITOS DO PROJETO

O desenvolvimento da ferramenta de gestão de atendimentos para a assistência técnica foi norteador por uma série de requisitos cruciais, que visavam garantir a eficácia, usabilidade e segurança da solução. Estes requisitos foram definidos com base nas necessidades da empresa, nas expectativas dos usuários e nas particularidades do ambiente operacional. A definição clara e precisa dos requisitos é fundamental para o sucesso de qualquer projeto de desenvolvimento de *software*, pois eles servem como guia para as decisões de *design*, implementação e testes, além de garantir que o produto final atenda às necessidades dos usuários e aos objetivos do negócio.

3.1.1 Requisitos Funcionais:

- **Padronização dos Dados:** A ferramenta deveria garantir a padronização dos dados coletados e processados, evitando inconsistências e erros que pudessem comprometer a qualidade das informações. A padronização dos dados era essencial para facilitar a análise e a comparação dos atendimentos, além de garantir a confiabilidade dos relatórios e das análises geradas pela ferramenta.
- **Integração com SAP:** A ferramenta deveria se integrar perfeitamente à plataforma *SAP*, mantendo os dados armazenados no sistema e garantindo a sincronização das informações. A integração com o *SAP* era essencial para evitar a duplicação de dados e garantir a consistência das informações em ambos os sistemas.
- **Filtros Avançados:** A capacidade de filtrar os atendimentos por diversos critérios era essencial para facilitar a organização e a priorização das tarefas. Os filtros deveriam permitir a busca por indicadores da seção, dados do cliente, tipo de equipamento, local de fabricação e outras informações relevantes. Essa funcionalidade permitiria aos analistas encontrar rapidamente os atendimentos que demandam sua atenção, agilizando o processo de tomada de decisão e otimizando o tempo gasto na busca por informações. A implementação de filtros avançados, como filtros por data, *status*, palavras-chave e combinações de critérios, aumentaria a flexibilidade e a eficiência da ferramenta na organização e análise dos dados.

3.1.2 Requisitos Não Funcionais:

- **Implementação Rápida:** A necessidade de uma solução ágil e eficiente impôs um prazo de desenvolvimento de no máximo 6 meses. Essa restrição de tempo exigiu uma abordagem focada e eficiente, com a utilização de metodologias ágeis, que permitem a entrega de funcionalidades em ciclos curtos e a adaptação rápida às mudanças

de requisitos. Além disso, a utilização de ferramentas de desenvolvimento que acelerassem o processo de criação e implementação da ferramenta, como *frameworks* e bibliotecas de código aberto, foi fundamental para cumprir o prazo estabelecido.

- **Intuitividade:** A ferramenta deveria ser de fácil utilização, mesmo para usuários com pouca experiência em tecnologia. Isso implicava em um *design* de interface intuitivo, com menus claros, ícones autoexplicativos e fluxos de trabalho simples e lógicos. A facilidade de uso era fundamental para garantir a rápida adoção da ferramenta pelos analistas e maximizar seus benefícios, evitando a necessidade de treinamentos extensos e reduzindo a curva de aprendizado. Para alcançar esse objetivo, foram realizadas pesquisas com usuários, testes de usabilidade e análise de *feedback*¹ para garantir que a interface fosse amigável e eficiente.
- **Simplicidade:** A ferramenta deveria ser simples e objetiva, evitando a criação de novas burocracias ou processos complexos que pudessem dificultar o trabalho dos analistas. A simplicidade era um requisito fundamental para garantir a rápida adoção e a utilização eficiente da ferramenta no dia a dia da assistência técnica. Para isso, foram adotadas práticas de *design* minimalista, com foco na funcionalidade e na facilidade de uso. Além disso, foram realizados testes de usabilidade com os analistas para identificar e eliminar possíveis dificuldades ou complexidades na utilização da ferramenta.
- **Acesso Rápido às Transações SAP:** A ferramenta deveria facilitar o acesso às transações *SAP* mais utilizadas pelos analistas, agilizando o processo de consulta e atualização de dados. Essa funcionalidade permitiria aos analistas economizar tempo e esforço, concentrando-se na análise dos problemas e na busca por soluções. Para isso, foram criados atalhos e *links* diretos para as transações mais frequentes, além da possibilidade de personalizar a interface da ferramenta para exibir as transações mais relevantes para cada usuário.
- **Conformidade com Segurança:** A ferramenta deveria estar em conformidade com as políticas de segurança da empresa, garantindo a proteção dos dados confidenciais e o acesso restrito às informações sensíveis. A segurança era um requisito fundamental para evitar o vazamento de dados e garantir a privacidade dos clientes e da empresa.
- **Desempenho e Estabilidade:** A ferramenta foi projetada para oferecer uma experiência ágil e confiável, funcionando de forma fluida e estável em todos os centros operacionais da empresa, independentemente da localização geográfica ou das condições da rede. Essa estabilidade operacional é fundamental para garantir a continuidade do trabalho dos analistas, evitando interrupções que possam comprometer

¹ Feedback: Opiniões e sugestões dos usuários sobre o *software*.

a produtividade e a qualidade do atendimento aos clientes. Para alcançar esse nível de desempenho, a ferramenta passou por rigorosos testes em diversos ambientes, simulando diferentes cenários de uso e carga de trabalho. Além disso, o código e o banco de dados foram otimizados para garantir tempos de resposta rápidos e um baixo consumo de recursos, proporcionando uma experiência de uso eficiente e sem frustrações para os usuários.

- **Uso Simultâneo:** A ferramenta deveria permitir o acesso e a utilização por múltiplos usuários simultaneamente, sem comprometer o desempenho ou a integridade dos dados. Essa característica era fundamental para atender às necessidades de uma equipe de assistência técnica distribuída em diferentes localidades. A implementação de um sistema de controle de concorrência, que gerencia o acesso simultâneo aos dados e evita conflitos, foi essencial para garantir a integridade e a consistência das informações.

Para o desenvolvimento do programa, foi necessário realizar um estudo aprofundado da linguagem *Python*, uma vez que os projetos anteriores no departamento eram feitos em *Microsoft Excel*. Além da ferramenta, o aluno teve que se aprofundar no estudo sobre banco de dados, sendo esta uma novidade, e aprender a criar uma interface gráfica apenas via código.

3.2 METODOLOGIA

Considerando a urgência do projeto e a necessidade de uma solução rápida e eficiente para os problemas enfrentados pela assistência técnica, optou-se por uma metodologia experimental, com foco na agilidade e na adaptabilidade. Adicionalmente, para o desenvolvimento da ferramenta, optou-se por uma abordagem funcional, utilizando funções e bibliotecas como *Pandas* para manipulação e análise de dados. Essa escolha se deu pela natureza específica do projeto, que prioriza a integração de dados e a automação de tarefas, e pela simplicidade e foco na funcionalidade. Embora a Programação Orientada a Objetos (POO) ofereça vantagens em termos de organização e reutilização de código em projetos maiores, para o escopo deste trabalho, o modelo funcional se mostrou suficiente e eficiente. Essa metodologia, comumente conhecida como desenvolvimento ágil, permitiu que a ferramenta fosse construída de forma interativa e incremental, com ciclos curtos de desenvolvimento, testes e *feedback* dos usuários. A cada ciclo, novas funcionalidades foram adicionadas e refinadas, garantindo que o produto final fosse moldado de acordo com as necessidades reais da equipe de assistência técnica.

3.2.1 Análise do Problema

A primeira etapa do desenvolvimento consistiu em uma análise detalhada dos problemas existentes nos processos da assistência técnica. Essa análise envolveu a identificação dos gargalos, das ineficiências e das principais dificuldades enfrentadas pelos analistas no cotidiano. Para isso, foram realizadas entrevistas com os analistas, observação do fluxo de trabalho e análise dos dados disponíveis nos sistemas da empresa. As entrevistas permitiram coletar informações sobre as dificuldades e necessidades dos usuários, enquanto a observação do fluxo de trabalho possibilitou a identificação de etapas redundantes, demoradas ou propensas a erros. A análise dos dados, por sua vez, forneceu *insights* sobre o desempenho dos processos e as principais causas de atrasos e falhas.

3.2.2 Definição das Funcionalidades

Com base na análise do problema, foram identificadas as principais funcionalidades que a ferramenta deveria ter para solucionar os problemas e otimizar os processos. Com a colaboração dos analistas da assistência técnica, as prioridades foram estabelecidas, assegurando que a ferramenta fosse desenvolvida de forma alinhada com as necessidades dos usuários e os objetivos do projeto.

3.2.3 Desenvolvimento da Ferramenta

Durante o desenvolvimento, foram realizadas diversas interações, nas quais as funcionalidades implementadas eram testadas e validadas pelos analistas da assistência técnica. O *feedback* dos usuários foi fundamental para identificar possíveis problemas, ajustar as funcionalidades e garantir que a ferramenta atendesse às necessidades reais da equipe. Essa abordagem interativa e incremental permitiu que a ferramenta fosse aprimorada continuamente ao longo do desenvolvimento, garantindo que o produto final fosse uma solução eficiente para os desafios enfrentados pela assistência técnica.

Para ilustrar de forma concreta a implementação das funcionalidades do sistema, a tabela 3 apresenta uma visão detalhada das principais funções desenvolvidas. Cada entrada na tabela especifica a função, as entradas necessárias, o processamento realizado e as saídas geradas. Este detalhamento permite uma compreensão clara da lógica operacional e do fluxo de dados dentro do sistema, destacando como cada componente contribui para a eficiência geral e eficácia do atendimento técnico. A inclusão destas informações serve para evidenciar a aplicação prática dos conceitos teóricos discutidos anteriormente e demonstrar a capacidade técnica do sistema desenvolvido em responder às necessidades identificadas durante a fase de análise de requisitos.

Tabela 3 – Diagrama Funcional: Sistema de Gestão de Atendimentos e Ordens de Serviço

Função	Entradas	Processamento	Saídas
inicializar_programa()	(Nenhuma)	Inicializa o sistema, carrega configurações de usuário e inicializa interfaces gráficas.	(Nenhuma)
ler_DB()	(Nenhuma)	Carrega e converte os dados do banco de dados em um DataFrame do pandas para uso interno.	Tabela atualizada
verificar_SAP_aberto()	(Nenhuma)	Verifica se a aplicação SAP está rodando; se não, inicia a aplicação.	(Nenhuma)
atualizar_atendimentos()	(Nenhuma)	Sincroniza os dados de atendimentos do SAP e e-mails para atualizar o banco de dados local.	Tabela atualizada
filtrar_atendimentos()	Tabela de atendimentos atualizada	Aplica filtros definidos pelos widgets à tabela de atendimentos para refinar a visualização.	Tabela com os filtros aplicados
backup_DB()	(Nenhuma)	Verifica a data do ultimo backup, se for a mais de uma horas realiza um novo backup do banco de dados.	(Nenhuma)
abrir_CS()	Número da nota de atendimento	Abre a nota de atendimento no SAP para visualização.	(Nenhuma)
modificar_CS()	Número da nota de atendimento	Permite a modificação da nota de atendimento no SAP.	(Nenhuma)
abrir_OS()	Número da ordem de serviço	Abre ordem de serviço no SAP para visualização.	(Nenhuma)
modificar_OS()	Número da ordem de serviço	Permite a modificação da ordem de serviço no SAP.	(Nenhuma)
abrir_pasta_qualidade()	Número de série do equipamento	Acessa a pasta de qualidade de um equipamento específico.	(Nenhuma)
inserir_modificar_item_DB()	Código SQL, dados, número da nota de atendimento	Inserir ou atualiza itens no banco de dados baseado nos critérios especificados.	(Nenhuma)
ler_item_DB()	Código SQL, número da nota de atendimento	Lê um item específico do banco de dados com base na consulta SQL.	Dados do item consultado
abrir_CS03()	Centro de produção, código de produto	Abre a lista de materiais categorizada no SAP.	(Nenhuma)
abrir_CS12()	Centro de produção, código de produto	Abre a lista completa de materiais no SAP.	(Nenhuma)

3.2.4 Coleta e Armazenamento de Dados

Para a coleta de dados do *SAP*, utilizou-se a ferramenta de gravação de *macros*, que permite registrar as ações realizadas no sistema e gerar um código em *Advanced Business Application Programming (ABAP)*, a linguagem de programação nativa do *SAP*. Essa ferramenta se mostrou extremamente útil, pois permitiu automatizar a extração de dados do *SAP* de forma rápida e eficiente, sem a necessidade de desenvolver *scripts* complexos ou realizar configurações avançadas no sistema.

O código *ABAP* gerado pela ferramenta foi então adaptado para *Python*, a linguagem escolhida para o desenvolvimento da ferramenta, e integrado ao fluxo de trabalho da aplicação. A linguagem de desenvolvimento *Python* foi escolhida devido à sua versatilidade, facilidade de aprendizado e grande comunidade de usuários e desenvolvedores. O *Python* oferece uma ampla gama de bibliotecas e ferramentas que facilitam o desenvolvimento de aplicações, como a biblioteca *pandas* para manipulação de dados, *pywin32* para automação de tarefas no Windows e a biblioteca *tkinter* para a criação de interfaces gráficas. Além disso, o *Python* é uma linguagem de código aberto, o que significa que não há custos de licenciamento e que a comunidade de desenvolvedores contribui ativamente para o seu aprimoramento.

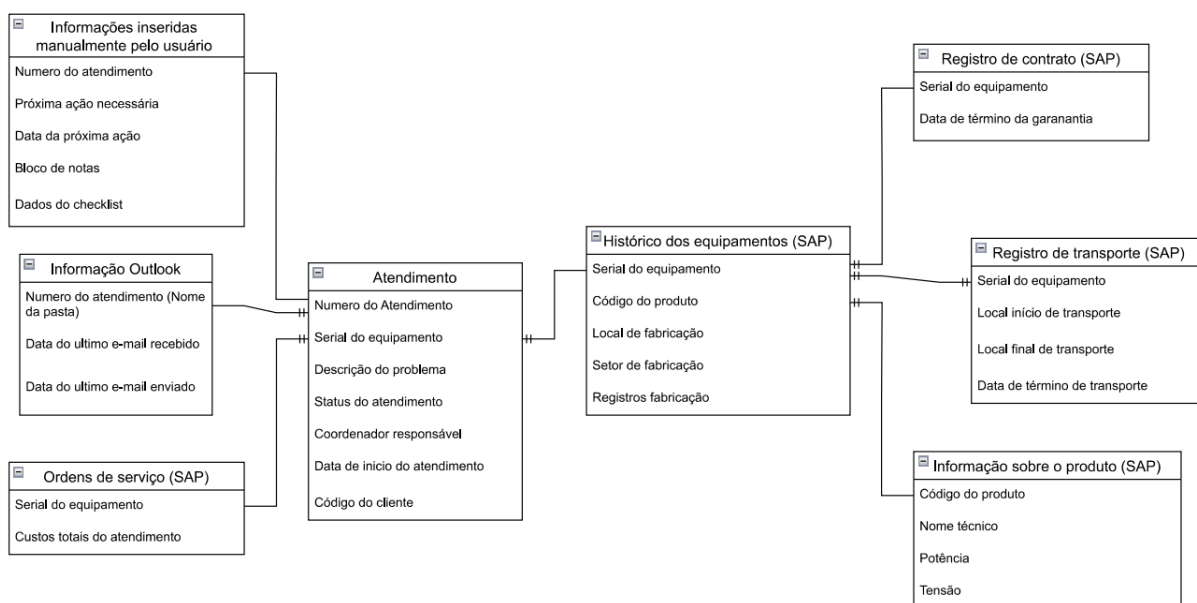
Inicialmente, o armazenamento dos dados coletados do *SAP* seria realizado em planilhas do *Microsoft Excel*, acessíveis por meio das *macros*. No entanto, essa solução mostrou-se lenta e pouco escalável, especialmente para grandes volumes de dados. Diante disso, foi realizada uma pesquisa para identificar um banco de dados mais adequado para o projeto.

As opções de bancos de dados online, como *PostgreSQL* e *MySQL*, foram consideradas, mas a equipe de Tecnologia da Informação (TI) da empresa expressou preocupações com a segurança e a necessidade de liberar licenças para esses bancos. Para contornar essa situação, optou-se por utilizar o *SQLite*, um banco de dados relacional de código aberto que se destaca por sua simplicidade, portabilidade e eficiência. O *SQLite* não requer um servidor dedicado e opera diretamente com arquivos de disco comuns, o que facilita sua implementação e uso. Além disso, o *SQLite* é uma opção segura, pois não expõe a rede da empresa a possíveis ataques e vulnerabilidades, como pode acontecer com bancos de dados online.

No entanto, o uso do *SQLite* em outros centros da empresa, situados em diferentes regiões, revelou problemas de desempenho devido à distância ao servidor que hospedava o banco de dados. Para resolver essa questão, foi adotada a técnica de *dual-database*. Esta abordagem envolve o uso de um banco de dados local no computador do usuário para atualizações imediatas da interface gráfica sem travamentos, enquanto as atualizações no banco de dados principal, hospedado no servidor, ocorrem em segundo plano. Essa técnica assegura eficiência na operação da ferramenta em todas as unidades operacionais, independentemente da localização geográfica.

A estrutura do banco de dados utilizado no sistema é ilustrada na figura (5). A tabela 'Atendimento' é a única tabela do banco de dados e armazena os dados centrais de cada atendimento, incluindo informações extraídas do sistema *SAP*. As tabelas marcadas com *SAP* são utilizadas exclusivamente para representar dados externos, facilitando a integração e análise das informações. Devido à natureza desta implementação, focada na operacionalidade e na integração direta com sistemas externos, o diagrama apresentado não adota as formalidades de um Diagrama Entidade-Relacionamento (DER) tradicional, que normalmente envolveria múltiplas tabelas para representar entidades e relações complexas.

Figura 5 – Estrutura do banco de dados utilizado no sistema

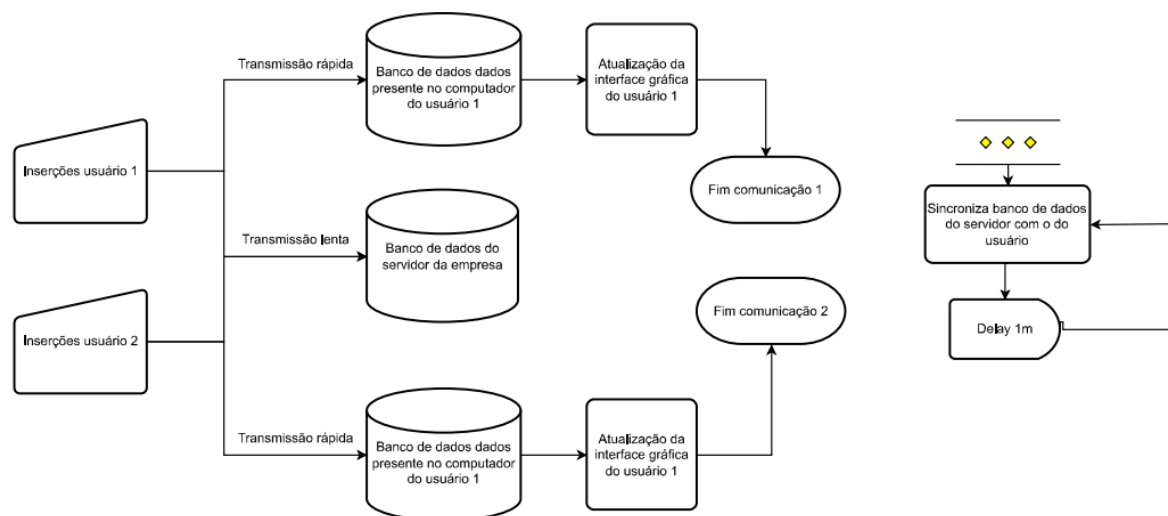


A sincronização entre o banco de dados local e o principal é realizada periodicamente, assegurando que as informações estejam sempre atualizadas e consistentes em todas as localidades. A Figura 6 apresenta um fluxograma que ilustra como as informações são armazenadas pelo programa usando a estratégia de *dual-database*.

3.2.5 Tratamento de Erros

Durante o desenvolvimento da ferramenta, a implementação de um tratamento de erros robusto foi fundamental para garantir a confiabilidade e a estabilidade do sistema. Utilizando a estrutura *try-except* do *Python*, foi possível capturar e gerenciar exceções que poderiam ocorrer durante a execução do programa, como erros na coleta de dados do *SAP*, problemas de conexão com o banco de dados ou falhas na execução de *scripts*. Caso ocorresse algum erro durante a coleta de dados do *SAP*, o sistema tentava diferentes caminhos para encontrar a informação desejada, garantindo que todos os dados necessários fossem coletados de forma precisa e completa. Essa abordagem minimizou a interrupção do fluxo de trabalho dos analistas e aumentou a confiabilidade da ferramenta.

Figura 6 – Diagrama salvamento utilizando a estratégia dual-database



Fonte: Próprio autor

3.2.6 Testes e Validação

Para garantir a qualidade e a eficácia da ferramenta, foi adotado um processo de testes e validação contínuo ao longo do desenvolvimento. A cada nova funcionalidade implementada, uma versão *beta* da ferramenta era disponibilizada para um grupo de analistas, que a utilizavam em seu cotidiano e forneciam *feedback* sobre sua usabilidade, desempenho e eficácia. Esse *feedback* era utilizado para identificar e corrigir erros, ajustar funcionalidades e aprimorar a ferramenta de acordo com as necessidades dos usuários. Ao longo do desenvolvimento, foram realizadas 15 versões de teste e 4 versões oficiais, cada uma com melhorias e correções de erros identificados nas versões anteriores. Esse processo iterativo de testes e validação permitiu que a ferramenta fosse refinada continuamente, garantindo que o produto final fosse uma solução robusta, confiável e eficaz para os problemas enfrentados pela assistência técnica.

3.2.7 Controle de Versão

O controle de versão do código fonte da ferramenta foi realizado de forma manual, com cada nova versão sendo enviada por *e-mail* para todos os usuários, acompanhada de um *changelog*² detalhando as alterações e correções implementadas. As versões de teste eram identificadas por números decimais (por exemplo, 3.1, 3.2), enquanto as versões oficiais eram identificadas por números inteiros (por exemplo, 3). Essa abordagem, embora manual, permitiu que os usuários tivessem acesso às últimas versões da ferramenta e que o histórico de desenvolvimento fosse documentado de forma organizada.

² Changelog: Registro detalhado das alterações em cada nova versão do *software*.

4 RESULTADOS

4.1 INTERFACE GRÁFICA

A interface gráfica da ferramenta de gestão de processos, como demonstra a Figura 7, foi projetada para ser intuitiva e funcional, buscando otimizar a experiência do usuário e a eficiência do processo de atendimento. A tela principal é dividida em quatro seções principais: Filtros, Atalhos, *Checklist* e Tabela, sendo cada seção com um propósito específico e recursos para simplificar a navegação e execução das tarefas diárias dos analistas.

Figura 7 – Visão geral da interface gráfica da ferramenta de gestão de processos

Nº	CS	OS	Status CS	Ação	Data Prev. Ação	Status (OS)	Status 2 (OS)
N2	301091	180380	GART	Esperar custos e encerrar	10/06/2024	LIB CAPC MOME SCDM	PREP BLOK
N2	301091	180380	GART	Esperar chegada	21/06/2024	LIB CAPC NDLQ SCDM	EMAV BLOK MERI
N1	301101	0	ANLS	Esperar	12/06/2024	None	None
N3	301085	0	GART	Esperar cliente - Na regão da enchente	17/06/2024	None	None
N2	301032	180357	GART	esperar	19/06/2024	LIB CAPC MOME	EMAV BLOK MERI
N3	301059	600051	GART	Esperar chegada	21/06/2024	LIB CAPC NDLQ SCDM	EMAV BLOK MERI
N2	301091	600051	GART	Esperar custos	31/12/2024	LIB CAPC MOME NDLQ SCDM	EMAV BLOK MERI
N2	301057	0	GART	None	nan	None	None
N2	301032	180375	ANLS	None	nan	LIB CAPC MATT MOME	EMAV MERI
N2	301074	0	ANLS	None	nan	None	None
N2	301057	180372	GART	None	nan	LIB CAPC NDLQ SCDM	PREP BLOK
N2	301062	0	ANLS	None	nan	None	None
N2	301039	180366	GART	Aguardar separação	nan	LIB CAPC MATT	EMAV BLOK MERI
N2	301039	180366	GART	Aguardar separação	nan	LIB CAPC MATT	EMAV BLOK MERI
N3	301096	600051	GART	None	nan	LIB CAPC MATT	EMAV BLOK MERI
N2	301029	180366	GART	Aguar separação	nan	LIB CAPC MATT	EMAV BLOK MERI
N2	301087	0	ANLS	None	nan	None	None
N2	301078	180376	ANLS	None	nan	LIB CAPC SCDM	EMAV BLOK MERI
N2	301078	0	ANLS	None	nan	None	None
N2	301052	0	ANLS	None	nan	None	None
N2	301038	0	ANLS	None	nan	None	None
N2	301039	180366	GART	Esperar chegada	nan	LIB CAPC EXP MATT MOME I	EMAV BLOK MERI
N1	301021	0	GART	None	nan	None	None
N1	301021	0	GART	None	nan	None	None
N1	301087	0	ANLS	None	nan	None	None
N2	301064	0	ANLS	None	nan	None	None
N2	301052	0	ANLS	None	nan	None	None
N2	301075	0	ANLS	None	nan	None	None
N2	301103	0	ANLS	None	nan	None	None
N3	301078	0	GART	None	nan	None	None
N3	301067	0	GART	None	nan	None	None
N3	301078	0	GART	None	nan	None	None
N3	301092	0	ANLS	None	nan	None	None
N3	301067	0	ANLS	None	nan	None	None
N2	301029	180361	GART	None	nan	LIB CAPC EXP SCDM	PREP BLOK
N2	301109	0	FEVE	None	nan	None	None
N2	301109	0	FEVE	None	nan	None	None
N2	301032	0	GART	None	nan	None	None
N2	301109	0	FEVE	None	nan	None	None

Fonte: Próprio autor

4.2 FILTROS

A seção de Filtros, conforme ilustra a Figura 8, permite aos usuários refinar a visualização dos dados na tabela de atendimentos, aplicando filtros com base em diversos critérios. Essa funcionalidade é essencial para que os analistas possam encontrar rapidamente os atendimentos que requerem sua atenção, permitindo a busca por informações específicas do cliente, indicadores da seção, *status* do atendimento e características do equipamento.

Os filtros disponíveis incluem campos para o nome do cliente, número do pedido, data de abertura do chamado, tipo de equipamento, modelo, número de série, local de instalação, *status* do atendimento (aberto, em andamento, concluído, etc.) e outros critérios relevantes. Além dos filtros básicos, a ferramenta oferece filtros avançados, como

Figura 8 – Interface dos filtros

A imagem mostra uma interface de usuário para filtrar atendimentos. O título é "Filtrar Atendimentos" em um cabeçalho azul. Abaixo, há cinco campos de filtro, cada um com um menu suspenso: "Apenas casos férias?" com o valor "NAO", "Coordenador:" com "AUGUSTO GASDA", "Nome do cliente:" com "TODOS", "Status Atendimento:" com "TODOS", e "Tipo de atendimento:" com um campo vazio. Cada menu suspenso tem uma seta para baixo azul.

Fonte: Próprio autor

a possibilidade de combinar múltiplos critérios de busca, utilizar operadores lógicos (E, OU) e realizar buscas por intervalos de datas.

A flexibilidade e a abrangência dos filtros permitem que os analistas personalizem a visualização dos dados de acordo com as próprias necessidades e prioridades, facilitando a identificação dos atendimentos mais urgentes, o acompanhamento do progresso das atividades e a tomada de decisões estratégicas. Por exemplo, um analista pode utilizar os filtros para identificar todos os atendimentos em aberto de um determinado cliente, ou para visualizar os chamados relacionados a um modelo específico de equipamento que apresentou um problema recorrente.

A interface dos filtros foi projetada para ser intuitiva e de fácil utilização. Os campos de busca são claramente identificados e organizados de forma lógica, e os operadores lógicos são apresentados de forma visual, facilitando a compreensão e a utilização dos filtros. Além disso, a ferramenta oferece dicas e sugestões de busca, auxiliando os usuários a encontrar as informações desejadas de forma rápida.

4.3 ATALHOS

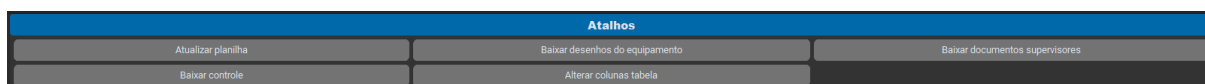
A seção de Atalhos oferece acesso rápido a ferramentas e funcionalidades importantes para o gerenciamento de atendimentos. Essa seção centraliza os atalhos para as ferramentas mais utilizadas pelos analistas, agilizando o acesso e otimizando o fluxo de trabalho. A Figura 9 ilustra a interface de atalhos da ferramenta, mostrando os atalhos disponíveis e sua organização visual. Com apenas alguns cliques, os usuários podem realizar diversas tarefas, como:

- **Download de Documentos do Equipamento:** Permite o *download* de documen-

tos, como manuais técnicos, desenhos e diagramas elétricos, que são importantes para entender o histórico e as especificações técnicas do equipamento em questão. Essa funcionalidade agiliza o processo de diagnóstico e resolução de problemas, pois os analistas têm acesso imediato às informações relevantes sobre o equipamento.

- **Documentação de Técnicos da empresa:** Agiliza a coleta e organização da documentação dos técnicos, como certificados, contratos e qualificações, para que o analista possa enviar ao cliente e dar início ao processo de integração de forma eficiente.
- **Atualização da Tabela de Atendimentos:** Permite a atualização imediata da tabela de atendimentos, garantindo que os dados estejam sempre atualizados e acessíveis. Essa funcionalidade é essencial para que os analistas tenham uma visão em tempo real do *status* dos atendimentos, possibilitando uma visão da priorização das tarefas e o acompanhamento do progresso das atividades.

Figura 9 – Interface de atalhos



Fonte: Próprio autor

4.4 CHECKLIST

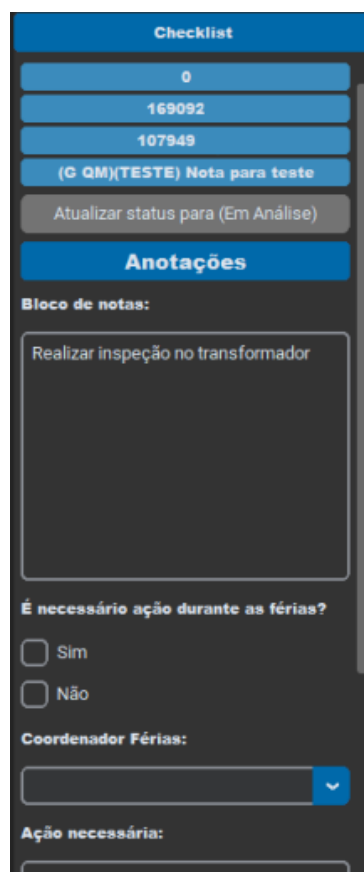
A seção de *Checklist*, mostrada na Figura 10, foi desenvolvida para gerenciar informações detalhadas de um atendimento selecionado na tabela principal. Esta aba permite:

- Fazer anotações detalhadas sobre o atendimento, como o histórico de interações com o cliente, as ações tomadas pela equipe de assistência técnica e as próximas etapas a serem realizadas. Essas anotações são importantes para manter um registro completo do atendimento e beneficiar a comunicação entre os membros da equipe, garantindo a todos, o acesso às informações relevantes sobre o caso.
- Definição de datas importantes, como a data de abertura do chamado, a data de agendamento da visita técnica, a data prevista para a conclusão do atendimento e a data da próxima ação. Essa funcionalidade permite aos analistas acompanhar o progresso do atendimento e o planejamento para as próximas etapas, garantindo o cumprimento de prazos e mantém o cliente informado a respeito do caso.
- Acesso direto a atalhos relacionados às transações na plataforma *SAP*, como a visualização do histórico do equipamento, a consulta de informações fiscais e a

atualização do *status* do atendimento. Essa funcionalidade, por meio da eliminação da navegação por diferentes menus e telas, agiliza o acesso às informações relevantes no *SAP* e otimiza o tempo gasto pelos analistas na busca por dados.

- Verificação de critérios necessários para classificar um atendimento em garantia, como a data de compra do equipamento, o tipo de problema e os termos da garantia. Essa funcionalidade auxilia os analistas na tomada de decisão sobre a cobertura da garantia, assegurando que os clientes sejam atendidos de acordo com as políticas da empresa e que os custos de reparo ou substituição sejam cobertos pela garantia quando aplicável.

Figura 10 – Interface do *checklist*



Fonte: Próprio autor

Os atalhos *SAP* são integrados de forma que um clique executa uma função específica relacionada ao objeto, e dois cliques iniciam outra função distinta. A implementação desses atalhos resultou em uma redução de aproximadamente 933,33 horas de tempo ocioso anualmente nos atendimentos. Para quantificar essa economia de tempo, foi utilizada a equação 1, onde ($T/Médio$) é o tempo que cada analista gasta para buscar informações necessárias para acessar cada transação, valor este calculado a partir de uma cronoanálise, ($Freq. Acessos$) reflete a média de vezes que cada analista acessa as transações, calculado empiricamente e ($Qnt At.$) é a quantidade total de atendimentos realizados no ano anterior.

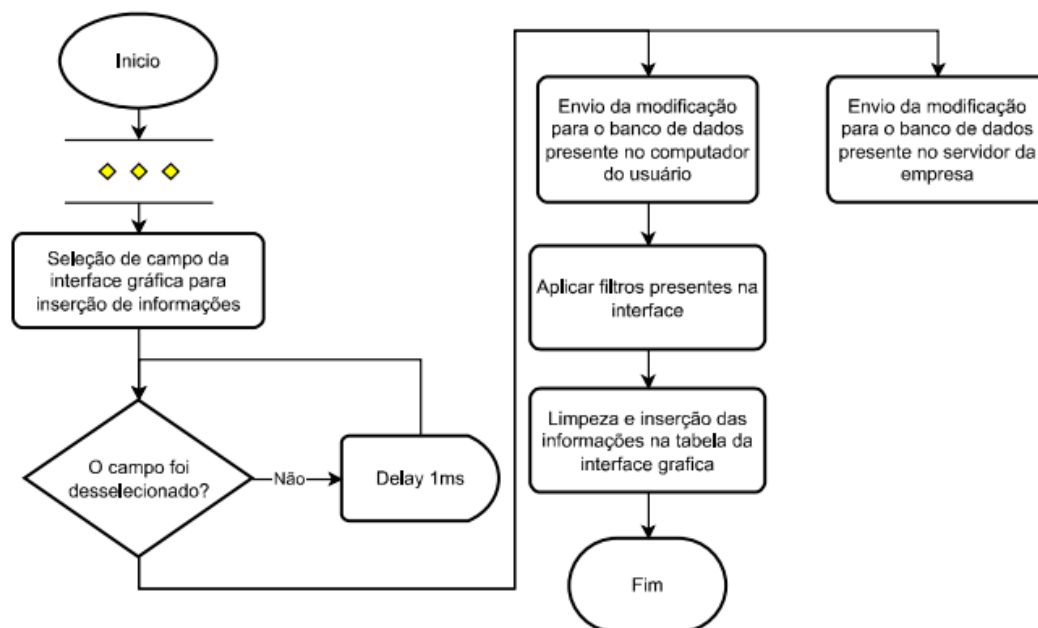
$$\text{Economia de Tempo} = (\text{T}/\text{Médio} \times \text{Freq. Acessos}) \times \text{Qnt At.} \quad (1)$$

O método empírico utilizado para determinar a frequência de acesso consistiu na coleta de dados diretamente dos analistas por meio de um questionário. Este foi projetado para determinar com que frequência cada analista utiliza as transações do sistema SAP durante um atendimento típico. A partir desses dados, calculou-se a média de acessos, proporcionando uma estimativa do uso dessas transações.

Essa redução significativa no tempo gasto em tarefas manuais e repetitivas demonstra a eficiência da ferramenta na otimização do fluxo de trabalho e na liberação dos analistas para se concentrarem em atividades mais estratégicas.

Para o registro das informações inseridas pelos usuários, adotou-se um método de salvamento automático após o desfoco do campo selecionado, conforme ilustrado no diagrama da Figura 11. Essa abordagem garante que as informações sejam salvas em tempo real, evitando a perda de dados em caso de fechamento inesperado da ferramenta ou de problemas técnicos.

Figura 11 – Diagrama de salvamento automático das informações inseridas no programa



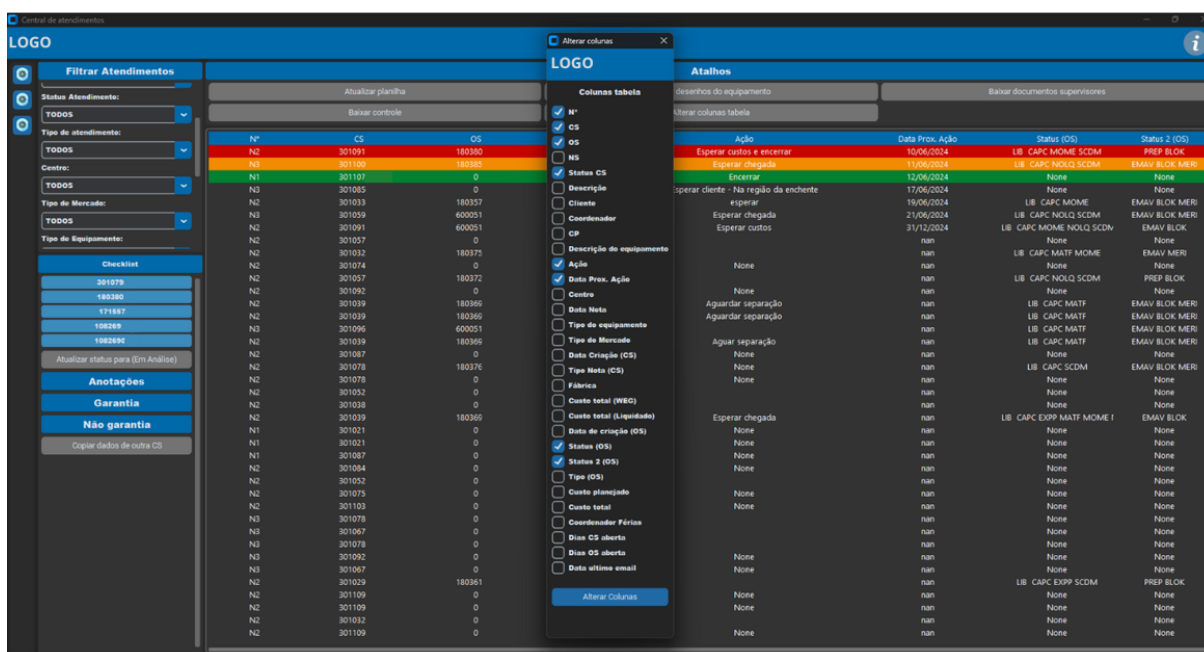
Fonte: Próprio autor

Essa abordagem de salvamento, aplicada em conjunto com a metodologia *dual database*, permitiu que a atualização das informações na interface gráfica fosse realizada em menos de 50 milissegundos, conforme medições realizadas no software de desenvolvimento. Essa velocidade de atualização garante uma experiência de usuário fluida e responsiva, evitando travamentos que poderiam prejudicar a produtividade dos analistas.

4.4.1 Personalização das Colunas da Tabela

Devido ao grande volume de informações, introduziu-se a funcionalidade que permite aos usuários personalizar a visualização a partir da seleção das colunas esperadas para exibir na tabela. Esta opção, ilustrada na Figura 12, aumenta a flexibilidade da ferramenta, adaptando-se às preferências e necessidades de cada usuário. A possibilidade de escolher quais informações são exibidas na tabela permite aos analistas o foco nos dados mais relevantes para cada tarefa, otimizando o tempo e a eficiência do trabalho.

Figura 12 – Interface de Filtros de Colunas da Tabela



Fonte: Próprio autor

A personalização das colunas da tabela é realizada por meio de uma interface simples, na qual os usuários podem marcar ou desmarcar as caixas de seleção correspondentes às colunas que desejam exibir ou ocultar. A ferramenta também permite salvar as configurações de personalização, para que o usuário não precise refazer a seleção das colunas em cada momento de utilização da ferramenta.

A implementação da personalização das colunas da tabela foi realizada utilizando a biblioteca *pandas*, a qual oferece recursos para a manipulação e exibição de *DataFrames*, a estrutura de dados utilizada para armazenar os dados da tabela. A biblioteca *pandas* permite selecionar colunas específicas de um *DataFrame*, reordenar as colunas e exibir apenas as colunas desejadas, o que possibilitou a implementação da funcionalidade de personalização de forma eficiente e flexível.

4.5 COLETA DE INFORMAÇÕES NA PLATAFORMA SAP

A coleta de dados sobre os equipamentos é realizada por meio de um botão integrado à seção de atalhos da Figura 9, utilizando programação *ABAP* e *scripts* em *Python* para acessar e extrair dados diretamente do *SAP*, sem intervenção manual do usuário.

O processo de atualização, executado em segundo plano em uma *thread* separada, leva em torno de um minuto para ser concluído. Essa duração, validada no ambiente de desenvolvimento, não impacta a utilização contínua da ferramenta, permitindo que os usuários prossigam com suas atividades sem interrupções. A atualização compartilhada garante que todos os usuários tenham acesso às informações mais recentes simultaneamente.

O diagrama simplificado da coleta de dados na Figura 13, ilustra as etapas do processo, incluindo a execução de *scripts*, tratamento de erros, processamento e consolidação das informações armazenadas em uma tabela do banco de dados. A coleta em segundo plano e a atualização compartilhada proporcionam uma experiência de usuário fluida e responsiva, permitindo que os analistas trabalhem sem interrupções enquanto as informações são atualizadas.

O processo otimizado de coleta de dados proporcionou uma economia estimada de 433,33 horas. Essa estimativa foi calculada com base em uma cronoanálise que mensurou o tempo gasto na coleta de dados e na elaboração da nota de remessa. A soma desses tempos foi multiplicada pelo número total de atendimentos realizados no ano anterior, conforme demonstrado na equação 2, onde $(T/M\ Coleta)$ representa o tempo médio de coleta de dados, $(T/M\ Escrita)$ indica o tempo médio para a escrita da nota de remessa, e $(Qnt\ At.)$ refere-se à quantidade de atendimentos realizados no ano anterior.

$$\text{Economia de tempo} = (T/M\ Coleta + T/M\ Escrita) \times Qnt\ At. \quad (2)$$

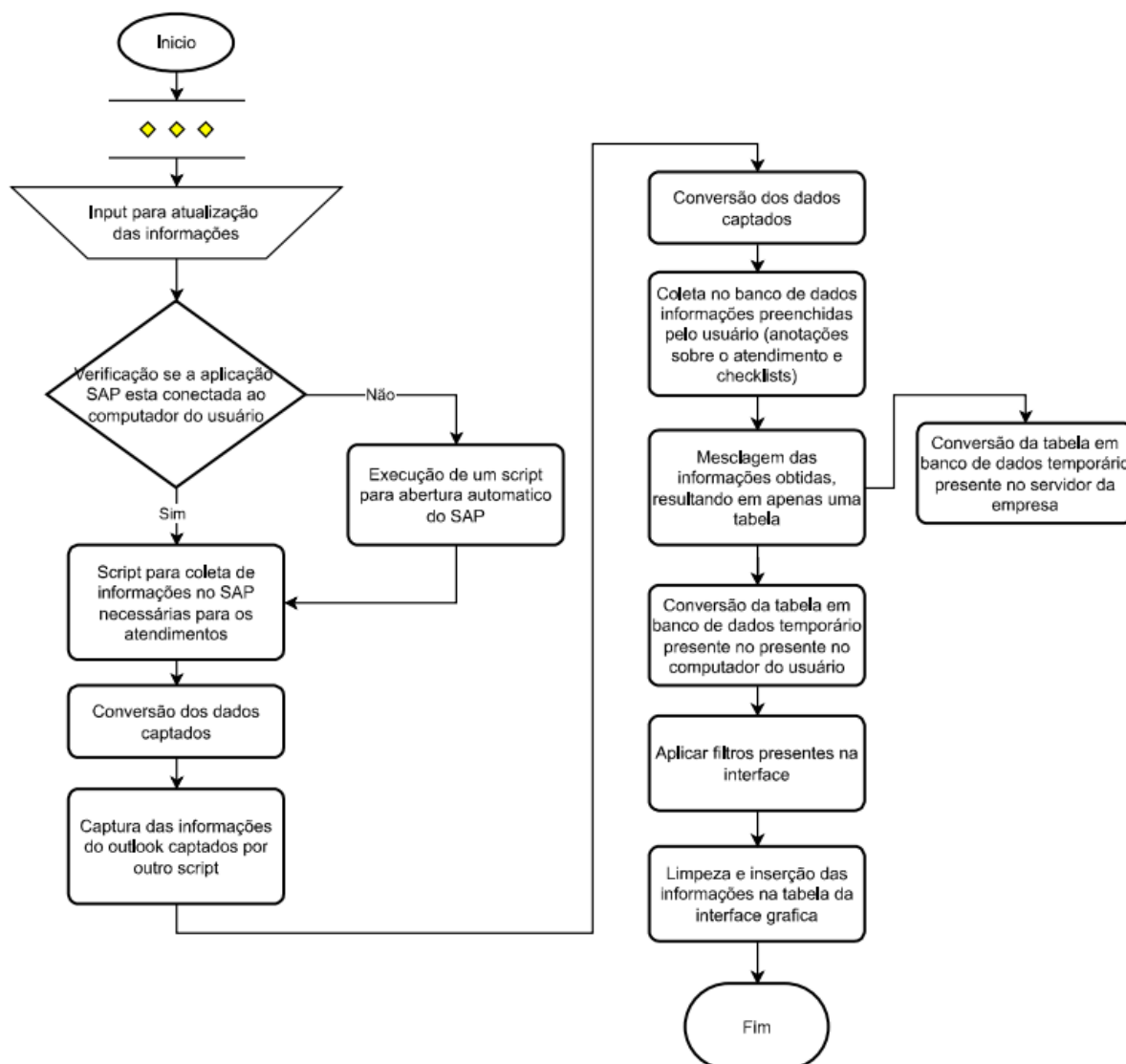
4.6 FUNCIONALIDADES EXTRAS

A ferramenta de gestão de processos desenvolvida para a assistência técnica não se limita às funcionalidades básicas de coleta e organização de dados. Ela também incorpora uma série de recursos extras capazes de aperfeiçoar a produtividade da equipe, facilitar a análise e o compartilhamento de informações. Esses recursos extras foram desenvolvidos com base no *feedback* dos usuários e nas necessidades identificadas durante o processo de desenvolvimento, demonstrando a capacidade da ferramenta de se adaptar e evoluir para atender às demandas da equipe de assistência técnica.

4.6.1 Exportação de Tabela para Microsoft Excel

Uma das funcionalidades extras implementadas na ferramenta é a exportação da tabela visualizada para um arquivo *Microsoft Excel*, sendo preservado a formatação da aplicação. Essa funcionalidade permite aos usuários a manipulação dos dados *offline*, utili-

Figura 13 – Diagrama da coleta de dados simplificado



Fonte: Próprio autor

zação das ferramentas e recursos do *Microsoft Excel* para realizar análises complexas, criar gráficos, tabelas dinâmicas e compartilhar informações filtradas de maneira conveniente.

A exportação para *Microsoft Excel* é especialmente útil para os analistas que precisam compartilhar informações com outros departamentos da empresa, como o departamento de qualidade, o setor de peças de reposição ou a equipe de vendas. Com essa funcionalidade, os analistas estão aptos para elaborar relatórios personalizados, com os dados filtrados e formatados de acordo com as necessidades de cada departamento, contribuindo para a comunicação e a colaboração entre as equipes.

Além disso, a exportação para *Microsoft Excel* também permite a realização de análises aprofundadas dos dados, utilizando as ferramentas de análise de dados do *Microsoft Excel*, como gráficos, tabelas e funções estatísticas. Essa funcionalidade pode ser utilizada para identificar tendências, padrões e oportunidades de melhoria nos processos

de assistência técnica, contribuindo para a tomada de decisões mais assertivas e para o aprimoramento contínuo dos serviços prestados.

4.6.2 Acesso aos *E-mails* Relacionados ao Atendimento

Outra funcionalidade extra implementada na ferramenta é o acesso rápido aos *e-mails* relacionados a um atendimento específico. Essa funcionalidade utiliza um mecanismo de busca recursiva de *e-mails* no *Microsoft Outlook* para identificar a pasta correspondente a um atendimento, utilizando o número de atendimento ao cliente como referência.

Com essa funcionalidade, os analistas acessam o histórico de comunicação com o cliente, incluindo *e-mails* trocados, informações sobre o problema, solicitações de peças de reposição e agendamentos de visitas técnicas. Essa aplicação é essencial para que os analistas tenham uma visão completa do atendimento, possibilitando o acompanhamento do caso e a tomada de decisões.

A busca recursiva de *e-mails* é realizada em uma *thread* separada, o que significa que o usuário pode continuar utilizando a ferramenta enquanto a busca é realizada em segundo plano. Essa abordagem garante uma experiência de usuário contínua e responsiva, evitando que a ferramenta fique travada enquanto a busca é realizada.

A implementação dessa funcionalidade resultou em uma economia significativa de tempo para os analistas, que antes precisavam procurar manualmente os *e-mails* relacionados a um atendimento em diversas pastas do *Microsoft Outlook*. Com a busca recursiva, os *e-mails* são encontrados em questão de segundos, liberando os analistas para se concentrarem em outras atividades importantes.

Em suma, a implementação da ferramenta de gestão de processos resultou em uma economia de tempo superior a 1.300 horas anuais para a seção de assistência técnica, conforme equações 1 e 2. Essa economia foi alcançada através da automatização da coleta de dados do *SAP*, da otimização do acesso às transações do *SAP* e da implementação da busca recursiva de *e-mails* no *Microsoft Outlook*. Além disso, a ferramenta promoveu melhorias significativas na organização, padronização e eficiência dos processos de atendimento, contribuindo para a redução de erros, o aumento da produtividade e a melhoria da qualidade do serviço prestado aos clientes.

5 CONCLUSÃO

A ideia, estudo, análise e desenvolvimento do projeto foram realizados por minha autoria, com o apoio e orientação da equipe de assistência técnica. Este estudo examinou os desafios e o fluxograma de uma assistência técnica, destacando os principais problemas enfrentados e as complexas e burocráticas tarefas necessárias para o controle e gestão de atendimentos, as quais se mostravam ineficientes e propensas a erros. Em resposta a esses desafios, foi desenvolvida uma ferramenta integrada que centraliza todas as informações necessárias em uma única plataforma, buscando otimizar o fluxo de trabalho, aumentar a eficiência e reduzir os erros.

Os resultados da implementação dessa ferramenta foram notavelmente positivos, demonstrando uma melhoria significativa na eficiência operacional. A padronização dos processos e a centralização das informações conseguiram não apenas reduzir o tempo necessário para treinar novos analistas, mas também diminuíram as chances de erros, proporcionando um serviço ao cliente mais ágil e confiável. Adicionalmente, a ferramenta promoveu uma maior colaboração entre os membros da equipe, permitindo o compartilhamento de informações e a resolução de problemas de forma conjunta.

Com a implementação da ferramenta, a coleta e organização das informações sobre os atendimentos, que antes eram realizadas manualmente, passaram a ser automatizadas com apenas alguns cliques. Essa mudança aumentou significativamente a eficiência operacional, pois centralizou todos os dados críticos em um único local. Assim, eliminou-se a necessidade de procurar informações em diversas pastas e *e-mails*, tornando o acesso muito mais rápido e eficaz. A automação também liberou os analistas de tarefas repetitivas e burocráticas, permitindo que eles se concentrassem em atividades mais estratégicas e que agregam valor ao negócio.

A centralização das informações também facilitou a automação de processos futuros, já que os principais dados estão organizados e prontos para uso. O controle burocrático dos processos tornou-se mais preciso e confiável, reduzindo a possibilidade de retrabalho, já que a automação minimiza a ocorrência de erros na extração de informações. A ferramenta também possibilitou a criação de *dashboards* e relatórios personalizados, fornecendo aos gestores uma visão clara e em tempo real do desempenho da equipe e dos resultados dos atendimentos.

Anteriormente, a gestão dos processos era dispersa entre planilhas *Microsoft Excel*, *e-mails* do *Microsoft Outlook* e *Microsoft OneNote* (MICROSOFT CORPORATION, 2024). Com a nova ferramenta, essa gestão tornou-se muito mais ágil, precisa e eficiente. O sistema agora integra anotações, dados do último *e-mail* trocado, informações do *SAP* e acesso às transações com poucos cliques. Essa padronização permite que o gestor acompanhe os processos de forma mais fácil e que o histórico dos atendimentos seja armazenado de maneira organizada e acessível. A integração entre os diferentes sistemas utilizados pela

equipe também eliminou a necessidade de alternar entre diferentes aplicativos, agilizando o trabalho e reduzindo o risco de erros.

Em resumo, a ferramenta desenvolvida não apenas resolveu os problemas iniciais, mas também abriu caminho para contínuas melhorias e inovações nos processos internos da assistência técnica. Os resultados demonstram um avanço significativo na gestão de atendimentos, beneficiando tanto os analistas quanto os clientes finais, e proporcionando um serviço mais eficiente e de alta qualidade. A ferramenta também se mostrou flexível e escalável, podendo ser adaptada às mudanças nas necessidades da empresa e aos avanços tecnológicos.

A experiência adquirida durante o desenvolvimento da ferramenta reforça a importância da colaboração entre as áreas de negócio e tecnologia. A participação ativa dos analistas da assistência técnica no processo de desenvolvimento foi fundamental para garantir que a ferramenta atendesse às suas necessidades e expectativas. Além disso, a utilização de metodologias ágeis e ferramentas de desenvolvimento modernas permitiu que a ferramenta fosse entregue em um curto espaço de tempo, sem comprometer a qualidade e a funcionalidade.

Este trabalho de conclusão de curso demonstra o potencial da tecnologia para otimizar processos, aumentar a eficiência e melhorar a qualidade dos serviços em empresas de diversos setores. A ferramenta desenvolvida para a assistência técnica serve como um exemplo de como a tecnologia pode ser utilizada para solucionar problemas reais e gerar valor para um negócio. Espera-se que este projeto inspire outras iniciativas de transformação digital e que a ferramenta continue a evoluir e a contribuir para o sucesso da empresa no futuro.

REFERÊNCIAS

BALREIRA, D.G. **Programação Didática com Linguagem C**. [S.l.]: Paco e Littera, 2022. Acesso em: 14-jul-2024. Disponível em:
https://books.google.com.br/books?id=g_1gEAAAQBAJ.

BARRANQUERO, Joaquín *et al.* Development of a tool for the automatic generation of spreadsheets based on Python and Openpyxl. **Journal of Chemical Education**, American Chemical Society, v. 92, n. 2, p. 347–351, 2014.

CIROLINI, Eduardo Giacomini. Aplicação ABAP para gerenciar acessos a ambientes SAP. **AMF**, 2016.

CLARK, Charlie *et al.* **Openpyxl: A Python library to read/write Excel 2010 xlsx/xlsm files**. [S.l.: s.n.], 2023. Disponível em:
<https://openpyxl.readthedocs.io/en/stable/>.

CRUZ, António Manuel Monteiro da. **Sistemas integrados de gestão empresarial: evolução e contextos de aplicação do SAP ERP**. 2021. Diss. (Mestrado).

CRUZ, Tadeu. **BPM & BPMS-Business Process Management & Business Process Management Systems**. [S.l.]: Brasport, 2008.

DATE, Christopher J. **Introdução a sistemas de bancos de dados**. [S.l.]: Elsevier Brasil, 2004.

DAVENPORT, Thomas H. **Process innovation: reengineering work through information technology**. [S.l.]: Harvard business press, 1993.

ERIK CEDERSTRAND. **Exchangelib**. [S.l.: s.n.], 2019.
<https://github.com/ecederstrand/exchangelib>.

FOUNDATION, Python Software. **Tkinter: GUI toolkit for the Python programming language**. [S.l.], 2023. Disponível em:
<https://docs.python.org/3/library/tkinter.html>.

GARCIA, Vinícius Salles; SOTTO, Eder Carlos Salazar. Comparativo Entre os Modelos de Banco de Dados Relacional e Não-Relacional. **Revista Interface Tecnológica**, v. 16, n. 2, p. 12–24, 2019.

GRAINGER, Joshua. **Beginning PyQt: A Hands-on Approach to GUI Programming**. [S.l.]: Packt Publishing Ltd, 2021.

HAMMER, Michael. O que é Gestão de Processos de Negócio. **Manual de BPM: Gestão de processos de negócio**, Bookman, Porto Alegre, p. 164–180, 2013.

HAMMOND, Mark; ROBINSON, Andy. **Python programming on Win32**. [S.l.]: O'Reilly Media, Inc., 2000.

HARMON, P. **Business Process Change: A Business Process Management Guide for Managers and Process Professionals**. [S.l.]: Elsevier Science, 2019. ISBN 9780128158487. Disponível em:
<https://books.google.com.br/books?id=S-WKDwAAQBAJ>.

JARA, Gabriel Flauzino; DA SILVA, Lincoln Mateus; SANCHES, Vanderlei Nunes. Interfaces de Usuário em Python com PyQt5: Uma Abordagem Front-End, s.d.

JURAN, Joseph M; GRYNA, Frank M. **Juran on quality by design: The new steps for planning quality into goods and services**. [S.l.]: Simon e Schuster, 1992.

KLOSS, Brent. **Auto PY to EXE**. [S.l.: s.n.], 2019.

LUTZ, Mark. **Programming Python**. [S.l.]: O'Reilly Media, Inc., 2013.

MCKINNEY, Wes *et al.* Data structures for statistical computing in python. *In*: AUSTIN, TX. PROCEEDINGS of the 9th Python in Science Conference. [S.l.: s.n.], 2010. P. 51–56.

MICROSOFT CORPORATION. **Microsoft Excel**. [S.l.], 2023. Disponível em:
<https://www.microsoft.com/excel>.

MICROSOFT CORPORATION. **Microsoft OneNote**. [S.l.: s.n.], 2024.
<https://www.onenote.com>.

MICROSOFT CORPORATION. **Microsoft Outlook**. [S.l.], 2023. Disponível em:
<https://outlook.live.com>.

MOREIRA, Eliana Alves; CARRIEL, Gerson Nunho. **Paralelismo em Nível de Thread**. [S.l.: s.n.], 2015. entre.

ORACLE CORPORATION. **MySQL**. [S.l.: s.n.]. <https://www.mysql.com>.

PRESS, Galileo. **Design patterns in object-oriented ABAP**. 1. ed., 2. reprint. Boston, Mass. u.a.: [s.n.], 2008.

RAMOS, José Marcio Benite; LACERDA, Liluyoud Cury de; DUARTE, Sara Luize Oliveira. **Técnicas de Programação**. [S.l.: s.n.], 2018.

ROB, Peter; CORONE, Carlos. Sistemas de banco de dados. **Projeto, implementação e**, 2011.

SAP SE. **SAP ERP**. [S.l.], 2023. Disponível em:
<https://www.sap.com/products/erp.html>.

SCHIMANSKY, Tom. **customtkinter: A modern and customizable GUI-library for tkinter**. [S.l.: s.n.], 2023. Disponível em:
<https://github.com/TomSchimansky/CustomTkinter>.

SQLITE DEVELOPMENT TEAM. **SQLite**. [S.l.], 2024. Versão 3.43.0. Disponível em:
<https://www.sqlite.org/>.

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL**. [S.l.: s.n.].
<https://www.postgresql.org>.

UNIVERSIDADE FEDERAL FLUMINENSE. Tutorial de Introdução ao Python, 2011.

VAN ROSSUM, Guido; DRAKE JR, Fred L. **Python reference manual**. [S.l.]:
Centrum voor Wiskunde en Informatica Amsterdam, 1995.

VOLLEBREGT, Brent. **auto-py-to-exe: Converts Python scripts into executable Windows programs**. [S.l.: s.n.], 2023. Disponível em:
<https://pypi.org/project/auto-py-to-exe/>.