



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

Guilherme Takashiba

Automatização do Controle de Qualidade em Bobinas de Transformadores a Seco

Blumenau

2024

Guilherme Takashiba

Automatização do Controle de Qualidade em Bobinas de Transformadores a Seco

Trabalho de Conclusão de Curso submetido ao curso de Engenharia de Controle e Automação do Centro ou Campus Blumenau da Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Bacharel em Engenharia de Controle e Automação

Orientadora: Profa. Dra. Ana Julia Dal Forno

Blumenau

2024

Takashiba, Guilherme

Automatização do Controle de Qualidade em Bobinas de Transformadores a Seco / Guilherme Takashiba ; orientadora, Ana Julia Dal Forno, 2024.

60 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.

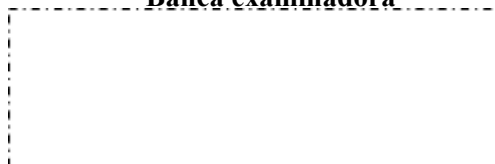
1. Engenharia de Controle e Automação. 2. Transformador a seco. 3. Controle de qualidade. 4. Poka yoke. I. Dal Forno, Ana Julia. II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Controle e Automação. III. Título.

Guilherme Takashiba

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 Engenharia de Controle e Automação.

Blumenau, 16 de dezembro de 2024.

Banca examinadora



Profª. Dra. Ana Julia Dal Forno
Orientadora



Prof. Dr. Mauri Ferrandin
UFSC campus Blumenau



Profª. Dra. Marilise Luiza Martins dos Reis Sayão
UFSC campus Blumenau

Blumenau, 2024

AGRADECIMENTOS

Agradeço a todos que fizeram parte da minha jornada ao longo da graduação, compartilhando comigo tanto os momentos de alegria quanto os de dificuldade. Sou imensamente grato pelos amigos que conquistei durante este percurso, cujo incentivo constante foi essencial para que eu nunca desistisse, mesmo diante dos maiores desafios. Também não posso deixar de agradecer aos meus amigos de longa data, que sempre estiveram ao meu lado, oferecendo apoio e palavras de encorajamento nos momentos mais importantes.

Agradeço profundamente aos meus pais e irmãos, que, de maneira única, ofereceram apoio incondicional e motivação em cada etapa desse processo. Minha gratidão também se estende à professora Marilise, que não apenas compartilhou seus valiosos conhecimentos, mas também foi uma fonte constante de apoio e inspiração, promovendo o crescimento acadêmico e pessoal de todos os seus alunos, inclusive o meu.

Por fim, deixo meu sincero agradecimento à professora Ana Julia, cujo suporte durante o desenvolvimento deste trabalho foi fundamental. Sua colaboração no LabTEI foi decisiva para a ampliação dos meus conhecimentos e para meu aprendizado ao longo da minha jornada acadêmica.

RESUMO

Este trabalho apresentou o desenvolvimento de um sistema automatizado para validação de dados no contexto da fabricação de transformadores a seco. O *software* foi desenvolvido com o objetivo de reduzir as falhas relacionadas às dimensões das bobinas, otimizando o processo de revisão de dados previamente realizado manualmente. A solução é integrada ao GIS, um software interno da empresa utilizado para gerenciamento de projetos de engenharia, e adota padrões de desenvolvimento como *Model-View-ViewModel* (MVVM) e *Composite* para estruturação e validação de dados. Durante os primeiros seis meses de aplicação, foi identificado um potencial de redução de custos de R\$ 10.172,94, eliminando aproximadamente 2,33% do total de não conformidades registradas no período e 2,26% dos custos relacionados a essas falhas. Problemas como espiras incorretas, diâmetros fora dos limites especificados e materiais incompatíveis foram evitados, promovendo maior confiabilidade nos processos. A aplicação de ferramentas da qualidade e da manufatura enxuta, como o *poka yoke* e os princípios do *Six Sigma*, garantiu uma abordagem estruturada para identificar e eliminar falhas no processo, promovendo ganhos em eficiência e qualidade. Os resultados demonstraram que a integração de ferramentas da qualidade em soluções de software impactou positivamente na redução de desperdícios e retrabalhos em ambientes industriais, além de reforçar a confiabilidade e o valor agregado à produção de transformadores a seco.

Palavras-chave: Transformador a seco; Controle de qualidade; *Poka-yoke*.

ABSTRACT

This work presents the development of an automated system for data validation in the context of dry transformer manufacturing. The software was developed to reduce failures related to coil dimensions, optimizing the data review process previously performed manually. The solution is integrated into the GIS, an internal company software used for engineering project management, and adopts development patterns such as Model-View-ViewModel (MVVM) and Composite for data structuring and validation. During the first six months of application, a potential cost reduction of R\$ 10,172.94 was identified, eliminating approximately 2.33% of the total non-conformities recorded during the period and 2.26% of costs related to these failures. Issues such as incorrect turns, diameters outside specified limits, and incompatible materials were avoided, enhancing process reliability. The application of quality and lean manufacturing tools, such as poka yoke and *Six Sigma* principles, ensured a structured approach to identifying and eliminating process failures, promoting gains in efficiency and quality. The results demonstrated that integrating quality tools into software solutions positively impacted waste and rework reduction in industrial environments, as well as reinforcing reliability and the added value of dry transformer production.

Keywords: Dry transformer; Quality control; Poka yoke.

LISTA DE FIGURAS

Figura 1 - Ferramentas utilizadas no desenvolvimento do sistema.....	16
Figura 2 - Tipos de transformadores de energia.....	18
Figura 3 - Exemplo de um núcleo de transformador com destaque para a bobina.....	19
Figura 4 - Diagrama de objetos de exemplo.....	21
Figura 5 - Ferramentas da qualidade e do Lean para gestão da qualidade.....	24
Figura 6 - Diagrama do ciclo DMAIC aplicado no desenvolvimento do sistema.....	26
Figura 7 - Representação do padrão MVVM.....	28
Figura 8 - Padrão de projeto Composite implementado na aplicação.....	29
Figura 9 - Estrutura do padrão de projeto Composite.....	30
Figura 10 - Fluxograma de dados do sistema GIS.....	31
Figura 11 - Demonstração dos módulos e formulários do GIS.....	33
Figura 12 - Modelo de desenvolvimento incremental.....	38
Figura 13 - Fluxograma de validação no sistema.....	42
Figura 14 - Fluxograma de criação das validações na <i>ViewModel</i> de validação.....	44
Figura 15 - Fluxograma do processo de adição de mensagens de aviso.....	45
Figura 16 - Exemplo de exibição das mensagens de validação.....	46
Figura 17 - Exibição das validações no módulo Validate.....	49
Figura 18 - Tela de validação com mensagens de alerta e informativo.....	54
Figura 19 - Tela do formulário LV1 sem a apresentação dos cards de validação.....	55
Figura 20 - Tela do formulário LV1 com a apresentação dos cards de validação.....	56

LISTA DE TABELAS

Tabela 1 - Relação de artigos para referência com foco em ferramentas da qualidade.....	37
Tabela 2 - Tabela de QMs de transformadores a seco entre janeiro e julho de 2024.....	58
Tabela 3 - Não conformidades com relação direta às dimensões de Bobinas.....	58

LISTA DE ABREVIATURAS E SIGLAS

BT	Baixa tensão
GIS	Global Integration System
GUI	Graphical user interface
MVC	Model-View-Controller
MVVM	Model-View-ViewModel
POO	Programação Orientada a Objetos
WPF	Windows Presentation Foundation

SUMÁRIO

1 INTRODUÇÃO	13
1.1 OBJETIVOS	13
1.2 ESTRUTURA DO TRABALHO	14
2 REFERENCIAL TEÓRICO	16
2.1 TRANSFORMADORES	17
2.1.1 Transformadores a seco	18
2.1.2 Enrolamento das bobinas	18
2.1.3 Efeito das dimensões na bobina	20
2.2 PRINCÍPIOS DA PROGRAMAÇÃO ORIENTADA A OBJETOS	20
2.2.1 Classes, objetos e instâncias	21
2.2.2 Encapsulamento	22
2.2.3 Reutilização e organização	23
2.2.4 Interfaces e serviços	23
2.3 FERRAMENTAS DA QUALIDADE	24
2.3.1 Poka-Yoke: Prevenção de erros	25
2.3.2 <i>Six Sigma</i> : Redução de variações	25
2.3.3 Kaizen: Melhoria contínua	26
2.3.4 Feedback	27
2.4 TECNOLOGIAS UTILIZADAS	28
2.4.1 C# e WPF	28
2.4.2 Padrão de projeto Model-View-ViewModel (MVVM)	28
2.4.3 Padrão de projeto Composite	29
2.4.4 GIS	31
2.4.4.1 Estrutura do GIS	31
2.4.4.2 Módulos do GIS	32
2.5 ARTIGOS RELACIONADOS	34
3 METODOLOGIA	38
3.1 METODOLOGIA INCREMENTAL DE DESENVOLVIMENTO DE SOFTWARE	38
3.2 DESENVOLVIMENTO DA FUNÇÃO DE VALIDAÇÃO	40
3.2.1 Pilares do sistema de validação	41
3.2.1.1 GeneralValidation	41
3.2.1.2 DryGeneralValidation	42
3.2.1.3 ValidateVM	44
3.2.2 Funcionamento da validação	45
3.2.3 Tipos de validação	46
3.2.4 Integração com a arquitetura MVVM	47

3.2.5 Veracidade dos dados	48
3.3 APLICAÇÃO DAS FERRAMENTAS DA QUALIDADE NO SISTEMA DE VALIDAÇÃO NA EMPRESA DE TRANSFORMADORES	49
3.3.1 Caracterização da empresa	49
3.3.2 Sistema de validação	49
3.3.3 Poka yoke	52
3.3.4 <i>Six Sigma</i>	52
3.3.5 Kaizen	54
4 RESULTADOS E DISCUSSÃO	55
5 CONCLUSÃO	60

1 INTRODUÇÃO

A indústria de transformadores a seco enfrenta desafios significativos relacionados à precisão das especificações de projeto, particularmente nas dimensões das bobinas, que são componentes críticos para o desempenho e a segurança dos produtos finais. Devido à grande variedade de requisitos demandados por diferentes clientes, como potência, tipo de material e formato dos fios, os parâmetros de projeto estão sujeitos a uma elevada possibilidade de erros. Essas falhas podem resultar em desperdício de matéria-prima, aumento das horas de engenharia para correções e custos elevados na produção.

Nesse contexto, a ausência de mecanismos automatizados para validação de dados técnicos pode comprometer a eficiência do processo produtivo, além de impactar negativamente a qualidade dos produtos finais. Portanto, torna-se imprescindível a implementação de um sistema que automatize a validação das especificações de projeto ainda nas etapas iniciais, assegurando a conformidade técnica e minimizando inconsistências que possam repercutir em toda a cadeia produtiva.

O presente trabalho foi desenvolvido no setor de Pesquisa, Desenvolvimento e Inovação, que tem como foco aprimorar e desenvolver os *softwares* da empresa fabricante de transformadores. O *software* em destaque, denominado GIS (Global Integration System), tem como objetivo unificar dados de um mesmo projeto, anteriormente dispersos por diferentes setores da empresa, promovendo maior integração e consistência no gerenciamento das informações.

Dessa forma, esta pesquisa propõe uma solução para a gestão da qualidade no setor de engenharia, integrando metodologias eficazes, como o *Poka-yoke* (dispositivos à prova de erros) e o *Six Sigma*, ao desenvolvimento de um sistema automatizado de validação de dados. Essas abordagens visam prevenir falhas, reduzir variabilidades e promover a eficiência nos processos de projeto e produção de transformadores a seco.

1.1 OBJETIVOS

O objetivo geral desta pesquisa foi desenvolver e implementar um sistema de validação de dados para projetos de transformadores a seco, com foco nas especificações técnicas das bobinas, garantindo que as dimensões e os parâmetros técnicos estejam em conformidade desde as etapas iniciais do projeto, promovendo a automação das verificações e assegurando maior eficiência e confiabilidade no processo produtivo.

Para isso, os objetivos específicos foram:

- a) Identificar na literatura os conceitos e práticas das ferramentas da qualidade e padrões de projeto tais como *Poka yoke*, *Six Sigma* e MVVM, que fundamentam o desenvolvimento do sistema.
- b) Identificar as principais falhas nas especificações técnicas de transformadores a seco: Analisar dados históricos para mapear não conformidades críticas e seus impactos.
- c) Definir os parâmetros do sistema de validação: Estabelecer critérios técnicos e limites aceitáveis para as especificações das bobinas.
- d) Desenvolver e implementar o sistema de validação integrado ao GIS: Projetar o sistema utilizando padrões de projeto e realizar sua integração com a plataforma interna da empresa.
- e) Analisar os impactos financeiros e operacionais do sistema: Avaliar a potencial redução de custos e os ganhos em eficiência com a eliminação de falhas.

1.2 ESTRUTURA DO TRABALHO

A estrutura deste trabalho foi planejada para apresentar de forma clara as etapas do desenvolvimento e análise do sistema automatizado de validação de dados para transformadores a seco. O primeiro capítulo, a introdução, contextualiza o tema, destacando a importância de sistemas automatizados no controle de qualidade e sua aplicação na indústria, além de apresentar os objetivos do estudo e sua justificativa.

O segundo capítulo aborda a revisão teórica, onde foi realizada uma pesquisa sobre artigos e publicações acadêmicas relacionadas a ferramentas da qualidade, como *Poka yoke* e *Six Sigma*, e padrões de projeto, como *Model-View-ViewModel* (MVVM). Este capítulo fornece a base conceitual necessária para o desenvolvimento do sistema e para os métodos aplicados.

No capítulo de metodologia, são descritas as etapas seguidas para a implementação do sistema, incluindo o uso de padrões de projeto e a integração com o *software* GIS, além da aplicação do ciclo DMAIC do *Six Sigma* para orientar a análise e o controle das soluções. Este capítulo também apresenta os critérios utilizados para a escolha das ferramentas e os procedimentos de validação automatizada.

O capítulo de resultados e discussão apresenta a análise dos dados coletados antes da implementação do sistema e projeta os benefícios que a solução proposta pode gerar. Os

resultados destacam os impactos financeiros positivos alcançados e a significativa redução de falhas já implementada no processo, evidenciando a eficácia das ferramentas da qualidade e da automação no contexto industrial..

Por fim, o capítulo de conclusão reúne os principais resultados do trabalho, apontando as contribuições da pesquisa, as limitações identificadas e as oportunidades para estudos futuros, como a expansão do sistema para outras categorias de não conformidades. Este capítulo reforça a relevância do sistema desenvolvido e sua aplicação prática em ambientes industriais.

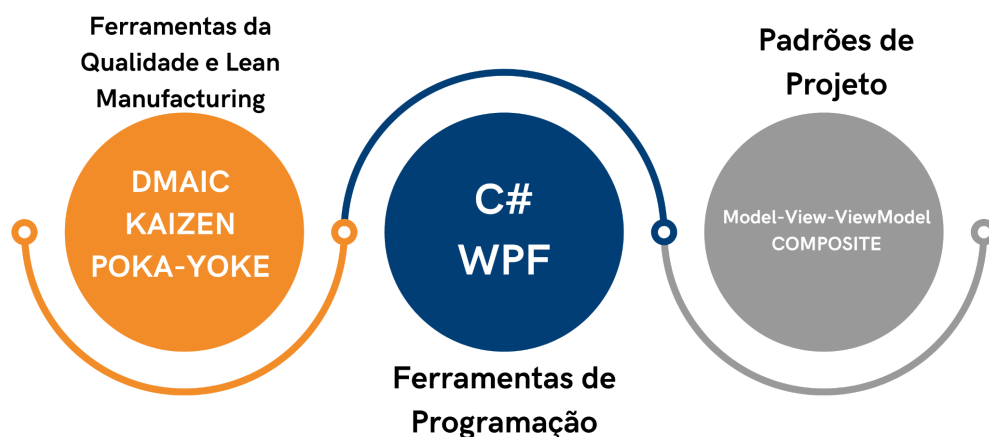
2 REFERENCIAL TEÓRICO

Este capítulo apresenta os fundamentos teóricos que sustentam o desenvolvimento deste trabalho, organizados em três eixos principais. Inicialmente, são explorados os transformadores a seco, destacando suas características estruturais e o impacto das dimensões das bobinas – como diâmetro interno, diâmetro externo e número de espiras – nos parâmetros de desempenho. A importância de um controle rigoroso sobre essas variáveis é discutida, evidenciando como elas influenciam diretamente a eficiência do processo produtivo e a minimização de desperdícios.

Na sequência, são abordados os princípios fundamentais da programação orientada a objetos (POO), que constituem a base do desenvolvimento do sistema de validação de dados. Conceitos como classes, objetos, encapsulamento, reutilização de código e interfaces são apresentados com ênfase na modularidade e escalabilidade do software, essenciais para garantir sua robustez e eficiência.

Por fim, o capítulo discute as ferramentas da qualidade que foram adaptadas para o contexto deste trabalho, como *Poka yoke*, DMAIC e *Kaizen*. Essas metodologias são analisadas em termos de sua aplicação prática, demonstrando como contribuem para a prevenção de falhas, o controle da variabilidade e a promoção da melhoria contínua. A Figura 1 ilustra a integração dos tópicos abordados, demonstrando uma relação entre as tecnologias utilizadas e os fundamentos teóricos.

Figura 1 - Ferramentas utilizadas no desenvolvimento do sistema.



Fonte: O Autor (2024).

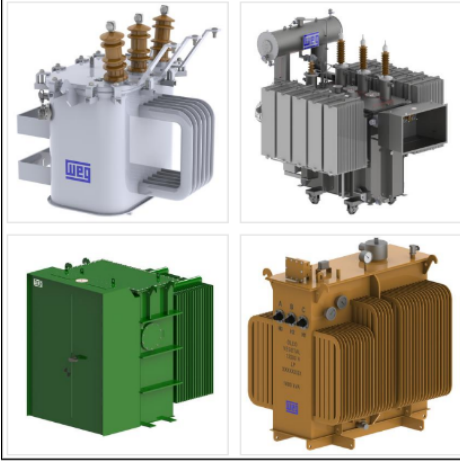
Para contextualizar o uso dessas ferramentas no desenvolvimento do sistema, foi realizada uma pesquisa de trabalhos relacionados à sua implementação em diferentes áreas. Os resultados dessa pesquisa são apresentados em uma tabela ao final do capítulo, reforçando como essas metodologias têm sido eficazes em variados contextos e destaca sua relevância para os desafios abordados neste trabalho.

2.1 TRANSFORMADORES

Os transformadores são dispositivos fundamentais no transporte e na distribuição de energia elétrica, desempenhando papel essencial na adequação dos níveis de tensão entre diferentes etapas do sistema elétrico. Seu funcionamento baseia-se na indução eletromagnética, conforme descrito pela Lei de Faraday. Segundo essa lei, a variação do fluxo magnético em um circuito induz uma força eletromotriz em outro circuito acoplado, fenômeno que possibilita a transferência de energia elétrica por meio de um núcleo magnético e bobinas. Nesse contexto, conforme observado por Patero (2023), "a relação linear entre a variação do fluxo magnético e a força eletromotriz induzida reafirma a aplicação das Leis de Faraday, permitindo a conversão e a transferência eficiente de energia em dispositivos como transformadores".

Esses equipamentos podem ser classificados de diversas formas, sendo uma das principais divisões baseada no meio utilizado para o isolamento e o resfriamento: transformadores a óleo e transformadores a seco. Enquanto os transformadores a óleo utilizam fluidos isolantes, como o óleo mineral, para dissipar calor e proteger os componentes internos, os transformadores a seco utilizam materiais sólidos e a circulação de ar para cumprir a mesma função. A Figura 2 ilustra essa diferença, destacando as características visuais e estruturais de cada tipo.

Figura 2 - Tipos de transformadores de energia.

TRANSFORMADORES	
TRANSFORMADORES A ÓLEO	TRANSFORMADORES A SECO
	

Fonte: WEG (2024).

2.1.1 Transformadores a seco

Os transformadores a seco distinguem-se pela ausência de líquidos refrigerantes em seu isolamento, utilizando materiais sólidos, como resinas epóxi, e a circulação de ar para dissipar calor. Essa característica os torna ideais para aplicações em ambientes onde a segurança contra incêndios é uma prioridade, como edifícios comerciais, hospitais e indústrias químicas.

Além da maior segurança, os transformadores a seco oferecem vantagens como menor necessidade de manutenção e uma durabilidade elevada em ambientes internos. Seu design elimina o risco de vazamentos de óleo, contribuindo para uma operação mais limpa e ambientalmente sustentável. Entretanto, devido à sua menor capacidade de resfriamento em comparação com os transformadores a óleo, seu uso é mais apropriado em situações onde a carga térmica seja moderada.

2.1.2 Enrolamento das bobinas

O enrolamento das bobinas é uma etapa essencial no projeto de transformadores, pois influencia diretamente o desempenho, a durabilidade e a segurança do equipamento. Esse processo consiste na organização dos condutores ao redor do núcleo do transformador, formando as espiras que permitirão a indução eletromagnética necessária para o

funcionamento do dispositivo. As técnicas de enrolamento podem variar conforme a aplicação e as exigências específicas de cada projeto, sendo os métodos de Enrolamento em Camadas (*Layer Winding*) e Enrolamento de Disco (*Disc Winding*) os mais amplamente utilizados.

Figura 3 - Exemplo de um núcleo de transformador com destaque para a bobina.



Fonte: PowerView Sistemas (2024).

Na Figura 3 observa-se um exemplo de núcleo de transformador com as bobinas em destaque, delimitadas pelo cubo vermelho, que evidencia a organização dos condutores.

No Enrolamento em Camadas, as espiras são dispostas em camadas com isolamento entre elas para evitar curtos-circuitos, o que proporciona uma distribuição uniforme do campo magnético ao redor do núcleo, em conformidade com a Lei de Faraday, que descreve a indução eletromagnética. Esse método é comumente empregado em transformadores de baixa e média tensão, onde o isolamento pode ser moderado. Por outro lado, o Enrolamento de Disco é preferido em transformadores de alta tensão, uma vez que os condutores são enrolados em discos separados por espaçadores, o que permite uma dissipação de calor mais eficiente e um isolamento elétrico reforçado. Essa técnica oferece resistência térmica e elétrica superiores, atendendo às demandas de tensão mais elevadas (Halliday; Resnick; Walker, 2014).

2.1.3 Efeito das dimensões na bobina

As dimensões das bobinas, incluindo o diâmetro interno, o diâmetro externo e o número de espiras, desempenham um papel fundamental na capacidade do transformador de suportar e transferir energia elétrica.

A Área da Seção Transversal do núcleo magnético, envolvida pelas bobinas, é diretamente proporcional ao fluxo magnético que pode ser gerado; quanto maior a área, maior a capacidade de potência do transformador, conforme explicado pela Lei de Faraday. Adicionalmente, o Número de Espiras é um fator determinante na relação de transformação de tensão entre os enrolamentos primário e secundário, sendo que um número maior de espiras resulta em uma tensão induzida mais alta. Por fim, o Espaçamento das Espiras e a Espessura do Fio também são parâmetros críticos: espaçamentos inadequados podem levar ao superaquecimento, resultando em falhas térmicas e comprometendo a segurança e eficiência do transformador.

Esses fatores destacam a importância de uma validação precisa das dimensões das bobinas, justificando a implementação de um sistema automatizado que verifique e assegure a conformidade desses parâmetros desde a fase de design.

2.2 PRINCÍPIOS DA PROGRAMAÇÃO ORIENTADA A OBJETOS

A Programação Orientada a Objetos (POO) é uma metodologia amplamente empregada no desenvolvimento de software, sendo especialmente eficaz em sistemas complexos que demandam modularidade, escalabilidade e manutenção eficiente. Essa abordagem oferece ferramentas conceituais como classes, objetos e instâncias, fundamentais para a construção de sistemas organizados e robustos. Segundo Fenves (1990), "a POO permite representar sistemas de engenharia por meio da abstração de dados, promovendo flexibilidade e modularidade no desenvolvimento de software".

Entre os conceitos centrais da POO está o encapsulamento, que protege os dados internos dos objetos e assegura a integridade do sistema ao restringir acessos externos a esses dados. Zhou et al. (1994) destacam que essa característica "não apenas protege os dados de acessos indevidos, mas também aumenta a confiabilidade e facilita a manutenção do software". Outro princípio relevante é a reutilização de código, viabilizada pela herança e pelo polimorfismo, que permite que classes derivadas herdem atributos e métodos de classes superiores, economizando esforços de desenvolvimento e reduzindo redundâncias.

Além disso, a organização hierárquica das classes e a clara definição de interfaces e serviços proporcionam maior flexibilidade e integração entre os componentes do sistema. Essas características tornam a POO uma escolha natural para o desenvolvimento de sistemas como o software de validação de dados, no qual a modularidade e a extensibilidade são essenciais para lidar com regras complexas e em constante evolução.

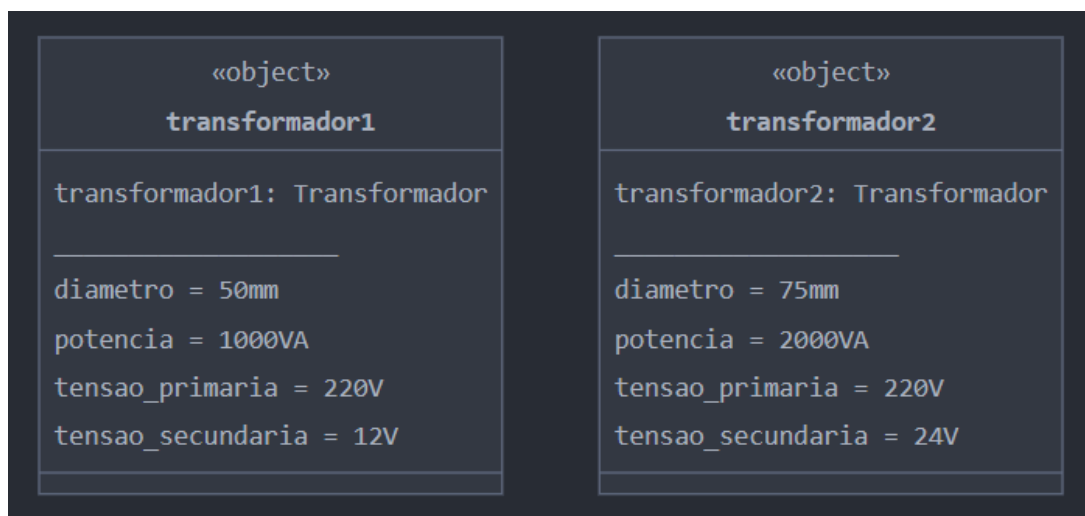
2.2.1 Classes, objetos e instâncias

No desenvolvimento de software orientado a objetos, a classe é um dos conceitos fundamentais e serve como uma estrutura que define as características e comportamentos de um tipo específico de objeto. A classe atua como um modelo, especificando os atributos (ou dados) que os objetos dessa classe podem conter e os métodos (ou comportamentos) que eles podem executar (MICROSOFT, 2024).

No contexto deste trabalho, voltado para o desenvolvimento de sistemas para projetos de transformadores, uma classe chamada Transformador pode ser criada para representar as especificações técnicas de um transformador. Essa classe inclui atributos como diâmetro, espessura e tensão de operação, além de métodos como CalcularEficiencia(), que calcula o desempenho do transformador, e ValidarDimensoes(), que verifica se os atributos estão em conformidade com os requisitos técnicos. Esses métodos e atributos fornecem um conjunto de regras e comportamentos para objetos que são instanciados a partir dessa classe.

A Figura 4 apresenta um exemplo prático do conceito de objetos e suas instâncias. A classe Transformador serve como base para a criação de dois objetos distintos, Transformador1 e Transformador2, que possuem valores próprios para seus atributos. Esses objetos são instâncias concretas da classe, representando transformadores específicos com características únicas. Por exemplo, enquanto o Transformador1 pode ter um diâmetro de 50mm, o Transformador2 pode ter um diâmetro de 75mm, demonstrando como a reutilização da classe permite a criação de múltiplos objetos com dados personalizados.

Figura 4 - Diagrama de objetos de exemplo.



Fonte: O Autor (2024).

Essa abordagem permite o desenvolvimento de sistemas mais flexíveis e escaláveis. A partir de uma única definição de classe, podem ser gerados inúmeros objetos com características próprias, simplificando a manutenção e a expansão do sistema. Além disso, ao encapsular os dados e comportamentos relacionados a transformadores em uma única classe, o sistema mantém uma estrutura coesa e organizada, facilitando sua integração com outras funcionalidades.

2.2.2 Encapsulamento

O encapsulamento é um princípio fundamental da orientação a objetos e desempenha um papel crucial na proteção e integridade dos dados dentro de uma classe. Ao encapsular os dados, eles ficam inacessíveis diretamente de fora da classe, podendo ser modificados ou acessados apenas por meio de métodos específicos. Isso significa que os atributos internos de um objeto estão protegidos e só podem ser manipulados de maneira controlada, reduzindo o risco de alterações acidentais ou incorretas (Fowler, 2004). Por exemplo, ao encapsular os atributos de um transformador – como as dimensões das bobinas – o sistema garante que esses valores sejam alterados apenas por métodos que validem as modificações, assegurando conformidade com as especificações técnicas. Essa prática facilita a criação de sistemas mais seguros e robustos, já que mudanças na implementação interna de uma classe podem ser realizadas sem interferir nas partes do código que dependem dessa classe (Martin, 2009).

No contexto de validação de dados, o encapsulamento permite que o sistema mantenha a integridade das informações dos projetos de transformadores ao longo de todo o processo de design e produção.

2.2.3 Reutilização e organização

A reutilização de código é outro princípio chave na orientação a objetos e contribui significativamente para a eficiência no desenvolvimento de software. Uma vez que uma classe é definida, ela pode ser reutilizada para criar múltiplos objetos sem a necessidade de duplicação de código. Além disso, a reutilização é facilitada pelo conceito de herança, que permite que uma classe herde atributos e métodos de outra classe, possibilitando a criação de estruturas hierárquicas. Por exemplo, uma classe de Transformador Seco poderia herdar uma classe Transformador, adicionando características específicas para transformadores a seco, como resistência térmica e métodos adicionais para dissipação de calor. Essa organização hierárquica e reutilizável facilita o desenvolvimento de sistemas complexos, uma vez que

permite o uso de componentes básicos para construir estruturas mais elaboradas sem comprometer a legibilidade e manutenção do código.

No sistema de validação, essa abordagem torna o processo mais modular, com diferentes tipos de transformadores podendo compartilhar e estender métodos e atributos comuns, o que promove consistência e facilita a expansão do sistema para novos tipos de projetos de transformadores.

2.2.4 Interfaces e serviços

No desenvolvimento de software orientado a objetos, as interfaces são componentes fundamentais que estabelecem contratos para as classes, especificando os métodos e propriedades que elas devem implementar, mas sem definir como esses métodos serão executados. O uso de interfaces permite flexibilidade e consistência, uma vez que diferentes classes podem implementar a mesma interface e, assim, interagir de maneira previsível dentro do sistema, mesmo quando as implementações desses métodos variam. Em um sistema de validação de dados, interfaces são particularmente úteis para garantir que classes diferentes implementem métodos de validação de maneira padronizada, independentemente das particularidades de cada tipo de transformador.

Além disso, os serviços são utilizados para organizar a lógica de negócios e executar tarefas específicas, como cálculos complexos ou operações de acesso a bancos de dados. Ao separar a lógica principal da aplicação em serviços, o sistema de validação se torna mais modular e eficiente, mantendo a lógica de validação bem organizada e desacoplada de outros componentes. Isso facilita a manutenção e permite que o sistema evolua conforme novas necessidades surgem, como a inclusão de novos tipos de transformadores ou de validações adicionais.

2.3 FERRAMENTAS DA QUALIDADE

As ferramentas de gestão da qualidade desempenham um papel fundamental na padronização de processos, na prevenção de falhas e na promoção da melhoria contínua. Este capítulo introduz algumas das principais metodologias, destacando aquelas aplicadas no desenvolvimento do sistema de validação de transformadores a seco. Entre as ferramentas abordadas estão o *Poka yoke*, o *Six Sigma* e o *Kaizen*, amplamente reconhecidas por sua eficácia na redução de erros e no controle da variabilidade.

Figura 5 - Ferramentas da qualidade e do Lean para gestão da qualidade.



Fonte: O Autor (2024).

A Figura 5 ilustra algumas das ferramentas mais utilizadas no campo da gestão da qualidade, incluindo as que foram adaptadas para o contexto deste trabalho. As metodologias destacadas como DMAIC, *Poka yoke* e *Kaizen*, são exploradas detalhadamente nas seções seguintes, com foco em sua aplicação prática e nos benefícios que oferecem ao sistema de validação.

2.3.1 Poka-Yoke: Prevenção de erros

O conceito de Poka Yoke, originalmente desenvolvido dentro do Sistema Toyota de Produção, é centrado na criação de processos "à prova de erros" e visa eliminar falhas humanas nas etapas de design e produção. Em ambientes industriais, o Poka Yoke é amplamente utilizado para garantir que especificações sejam seguidas rigorosamente, prevenindo que erros avancem para os estágios finais de produção. No trabalho de Moraes e Dal Forno (2022), as autoras analisam a aplicação do Poka Yoke na indústria têxtil, destacando sua eficácia na redução de defeitos em processos como fiação e confecção, onde erros críticos podem comprometer a qualidade do produto final. Um exemplo notável é o uso de sistemas de alarme que reduzem o tempo de operação em até 20%, aumentando a entrega pontual em 25%. Essa abordagem também permite uma inspeção 100% na fonte, interceptando falhas antes que elas avancem no fluxo produtivo. Assim, a aplicação do Poka Yoke no sistema de validação de dados pode garantir que erros críticos, como especificações

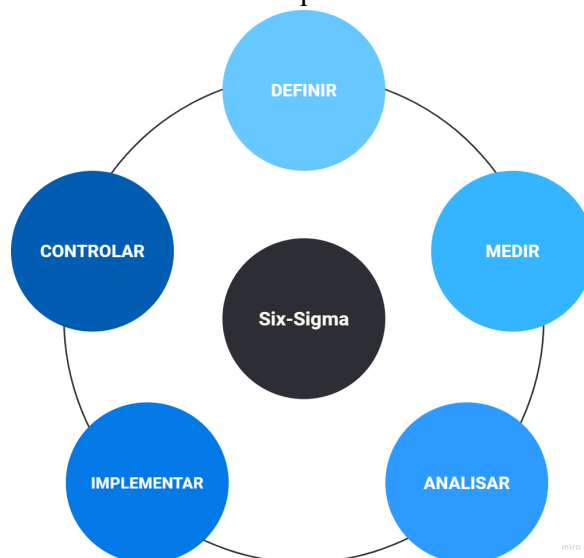
incorretas das dimensões de bobinas, sejam detectados e corrigidos na fase de design, assegurando a integridade do projeto e sua viabilidade na produção.

No contexto do projeto de transformadores, o *Poka yoke* é integrado ao sistema de validação para agir como uma “barreira de segurança”, prevenindo que dados fora dos parâmetros especificados sejam inseridos ou enviados para produção. Essa abordagem minimiza o retrabalho e o desperdício de materiais, alinhando-se à filosofia do *Poka yoke* ao garantir a conformidade das especificações técnicas desde o início.

2.3.2 *Six Sigma*: Redução de variações

O *Six Sigma* é uma metodologia que visa reduzir a variabilidade nos processos e garantir um controle rigoroso de qualidade, utilizando ferramentas estatísticas para identificar e eliminar causas de defeitos. Esse método é amplamente aplicável em ambientes industriais onde a padronização é essencial. Em "A Case Study of Using the Hybrid Model of Scrum and *Six Sigma* in Software Development", a abordagem *Six Sigma* foi aplicada em conjunto com *Scrum* no desenvolvimento de software para aumentar a eficiência e reduzir defeitos. Nesse estudo, a combinação do ciclo DMAIC (Definir, Medir, Analisar, Melhorar e Controlar), sendo ilustrado pela Figura 6, com metodologias ágeis demonstrou que é possível obter resultados significativos na melhoria da qualidade e no controle de processos, mesmo em ambientes de desenvolvimento de software onde os requisitos são dinâmicos e complexos.

Figura 6 - Diagrama do ciclo DMAIC aplicado no desenvolvimento do sistema.



Fonte: Sarpiri (2021).

Para o sistema de validação de transformadores, o *Six Sigma* é adotado para estabelecer limites bem definidos para cada especificação técnica, garantindo que as variações nas dimensões das bobinas estejam dentro dos parâmetros aceitáveis antes da entrada na linha de produção. A aplicação do ciclo DMAIC permite uma análise contínua dos dados coletados durante o processo de validação, possibilitando ajustes conforme necessário para otimizar o desempenho e a precisão do sistema. Esse controle rigoroso é essencial para evitar discrepâncias que possam comprometer a qualidade do produto final, uma vez que o sistema de validação atua como uma ferramenta de controle estatístico, assegurando que as especificações técnicas sejam atendidas com alta precisão.

2.3.3 Kaizen: Melhoria contínua

A filosofia *Kaizen*, que significa "mudança para melhor" em japonês, promove a melhoria contínua e o envolvimento dos colaboradores em todos os níveis da organização. No ambiente de engenharia, essa abordagem incentiva uma cultura de aprimoramento constante, onde cada membro da equipe é motivado a identificar oportunidades de melhoria e a colaborar para aumentar a eficiência dos processos. Em "Providing Early Feedback in the Development Cycle through Automated Application of Model Checking", Barber et al. (2001) exploram uma abordagem de *feedback* contínuo no desenvolvimento de software, utilizando uma ferramenta de verificação de modelos que proporciona uma avaliação antecipada da arquitetura do sistema, permitindo que ajustes sejam realizados logo no início do ciclo de desenvolvimento. Essa prática se alinha com o princípio *Kaizen*, pois promove uma cultura de revisão constante e redução de defeitos antes que eles impactem as fases avançadas do projeto.

No sistema de validação de transformadores, a aplicação do *Kaizen* é evidente na integração de *feedback* em tempo real para os engenheiros, permitindo que ajustes nas especificações sejam feitos rapidamente e com precisão. Além disso, o sistema foi projetado para ser flexível, possibilitando a inserção de novas regras de validação conforme surgem novos requisitos ou melhorias nos processos. Essa filosofia de melhoria contínua é essencial para garantir que o sistema de validação permaneça atualizado e adaptável às necessidades da produção, promovendo uma cultura de qualidade que vai além das exigências de conformidade técnica e inclui o compromisso com a inovação e a eficiência.

2.3.4 *Feedback*

A automação do *feedback* e da validação é uma estratégia amplamente defendida na literatura para garantir a precisão dos dados e a eficiência dos processos. A validação de dados automatizada foi discutida em "On the Experiences of Adopting Automated Data Validation in an Industrial Machine Learning Project", onde Lwakatare et al. (2021) relataram os benefícios e desafios de implementar um sistema de validação automática em um projeto de aprendizado de máquina industrial. Esse sistema visa detectar inconsistências nos dados desde o início, o que é fundamental para evitar que erros se propaguem e comprometam o desempenho dos modelos. A experiência descrita por Lwakatare et al. destaca a importância de um controle de qualidade contínuo, onde a validação automatizada é usada para garantir que as especificações estejam corretas e consistentes antes de qualquer processo subsequente.

No contexto do sistema de validação de transformadores, a automação desempenha um papel crucial ao permitir que os engenheiros detectem falhas e inconsistências nas especificações de forma precoce, evitando que dados incorretos avancem no processo produtivo. A utilização de uma abordagem de *feedback* automatizado proporciona uma visão clara das validações, classificadas por nível de criticidade (Erro, Atenção, Sucesso e Informativo), facilitando a correção rápida e precisa. Esse *feedback* é exibido diretamente na interface do sistema, assegurando que o controle de qualidade seja uma atividade integrada e contínua, alinhada aos objetivos de automação e melhoria contínua.

2.4 TECNOLOGIAS UTILIZADAS

2.4.1 C# e WPF

A linguagem C#, desenvolvida pela *Microsoft*, é amplamente utilizada em aplicações empresariais e sistemas robustos, devido à sua sintaxe clara, segurança e integração facilitada com o *.NET Framework*. No contexto do projeto, o uso de C# possibilitou o desenvolvimento de uma aplicação orientada a objetos, garantindo modularidade e fácil manutenção do código. C# também oferece suporte para integração com ferramentas como o *SolidWorks*, que é fundamental para sincronizar dados de projetos no sistema GIS.

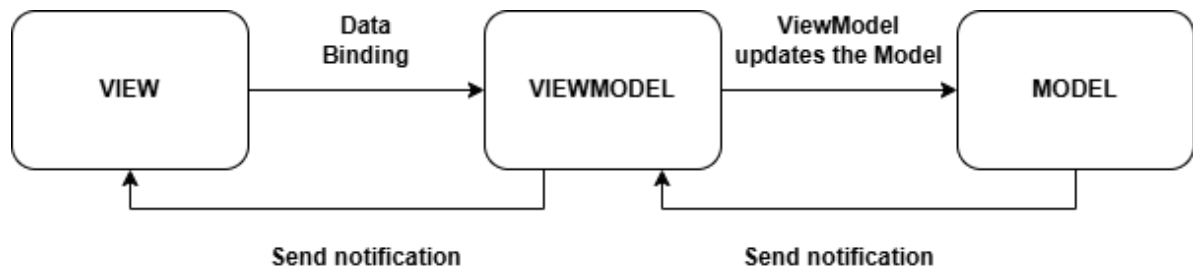
O *Windows Presentation Foundation* (WPF) foi selecionado como *framework* para a construção da interface gráfica (GUI). WPF oferece suporte avançado para *databinding*, que permite conectar a interface diretamente aos dados do sistema, facilitando atualizações em tempo real. Esse recurso foi essencial para exibir as validações de forma clara aos engenheiros, com *feedback* visual imediato sobre erros ou alertas nas especificações. Além

disso, WPF permite a criação de interfaces ricas em elementos gráficos e animações, possibilitando que o sistema seja visualmente intuitivo e fácil de usar.

2.4.2 Padrão de projeto Model-View-ViewModel (MVVM)

Para organizar a estrutura do sistema e promover a separação das responsabilidades, foi implementado o padrão de design *Model-View-ViewModel* (MVVM), a Figura 7 representa o padrão amplamente utilizado em aplicações WPF. Esse padrão é ideal para separar a lógica de negócios (*Model*), a interface gráfica (*View*) e a lógica de interação e gerenciamento de dados (*ViewModel*), tornando o código mais modular e facilitando a manutenção e testes.

Figura 7 - Representação do padrão MVVM.



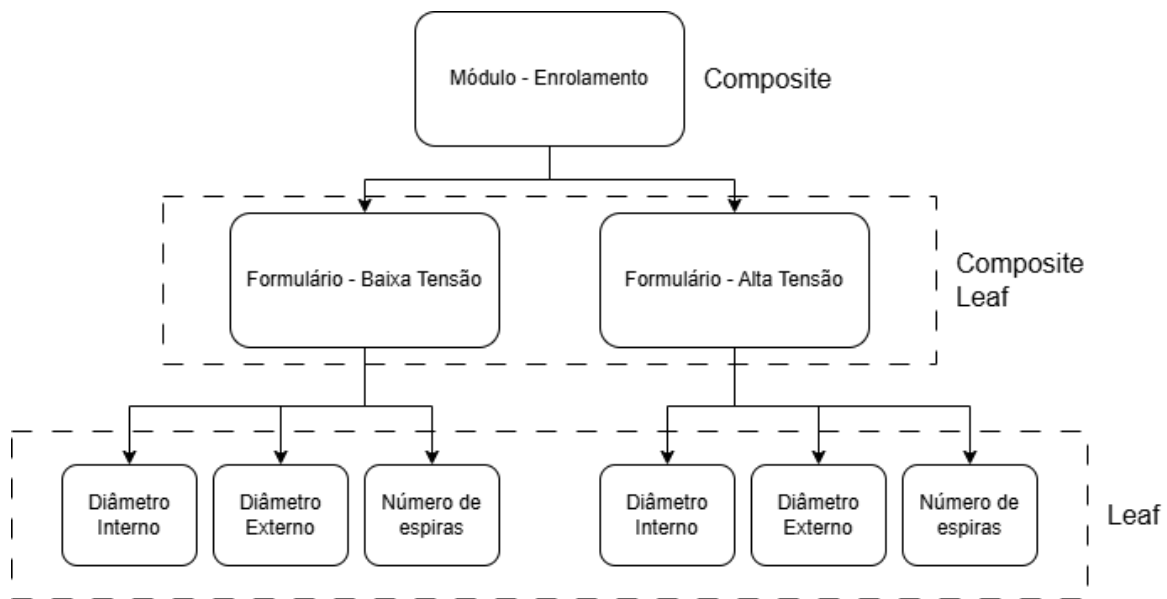
Fonte: O Autor (2024).

No sistema GIS, o MVVM permite que as validações de dados e a interface do usuário funcionem de maneira independente. A camada *Model* representa as especificações técnicas dos transformadores, armazenando dados como dimensões e materiais. A camada *ViewModel* serve como intermediária, conectando a lógica de validação ao que é exibido na interface, enquanto a *View* exibe as validações e os *feedbacks* visuais, permitindo que os engenheiros identifiquem rapidamente qualquer inconsistência. Esse padrão facilita o baixo acoplamento entre as camadas, permitindo a expansão e modificação do sistema sem a necessidade de reescrever grandes porções do código. Além disso, o MVVM promove a testabilidade, pois a lógica de validação pode ser testada isoladamente, garantindo a confiabilidade e precisão do sistema de validação.

2.4.3 Padrão de projeto Composite

Para organizar as validações de forma hierárquica, o sistema implementa o padrão de projeto *Composite*. Esse padrão permite que as validações sejam estruturadas em uma árvore, onde cada nó representa uma verificação específica, e cada conjunto de nós forma uma categoria de validações (ex.: dimensões das bobinas, resistência térmica, etc.), como ilustrado na Figura 8.

Figura 8 - Padrão de projeto Composite implementado na aplicação.

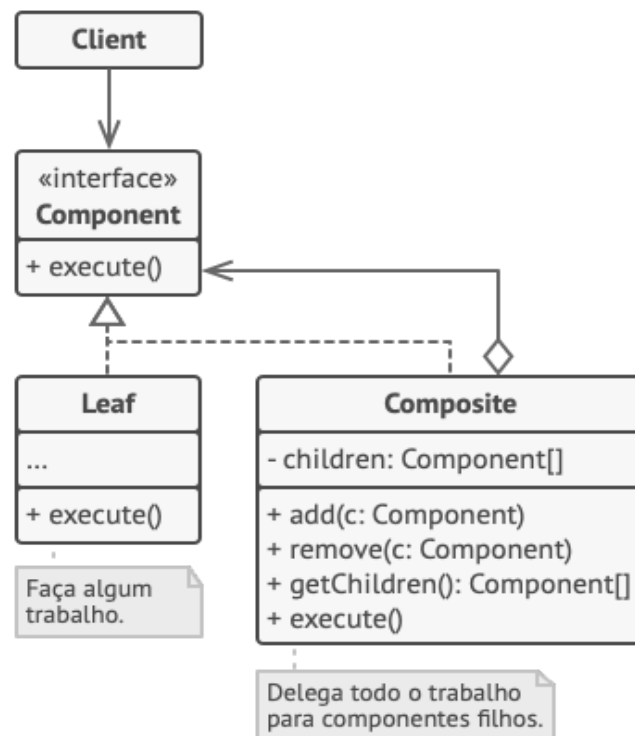


Fonte: O Autor (2024)

O *Composite* é um padrão que organiza elementos em uma estrutura de árvore, permitindo que objetos individuais (chamados *Leaf*) e grupos de objetos (chamados *Composite*) sejam tratados de forma uniforme. A Figura 9 ilustra a estrutura geral do padrão *Composite*, onde:

- O cliente interage com uma interface genérica (*Component*), que pode ser um *Composite* ou um *Leaf*.
- O *Composite* contém um conjunto de filhos (*children*), podendo adicionar, remover ou executar operações nesses filhos.
- O *Leaf* é o elemento final da árvore, que realiza a validação específica sem delegar trabalho a outros objetos.

Figura 9 - Estrutura do padrão de projeto Composite.



Fonte: Guru Pattern (2024)

O uso do Composite no contexto do sistema facilita a categorização das verificações, permitindo que os engenheiros visualizem rapidamente quais áreas precisam de ajustes e oferecendo as seguintes vantagens:

- **Escalabilidade:** Novas regras de validação podem ser adicionadas sem alterar a estrutura básica do sistema, garantindo que ele possa crescer conforme as necessidades do projeto.
- **Organização Visual:** A estrutura em árvore permite uma visualização clara das falhas e inconsistências, com destaque para os problemas prioritários, tornando o processo de revisão mais ágil e eficiente.

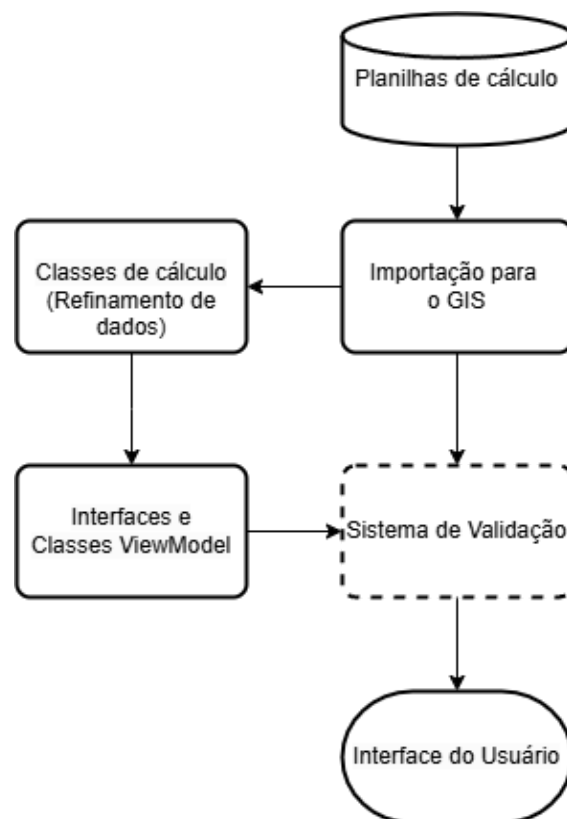
A implementação do padrão Composite e das classes de validação torna o sistema GIS flexível e adaptável a novas demandas, assegurando que as especificações dos transformadores estejam sempre em conformidade com os padrões de qualidade estabelecidos.

2.4.4 GIS

2.4.4.1 Estrutura do GIS

O sistema GIS foi desenvolvido para gerenciar e integrar os dados providos de planilhas e outros sistemas de cálculos no projeto de transformadores a seco, concentrando informações essenciais para a fabricação de transformadores, como especificações das bobinas, lista de materiais e cálculos dimensionais. O fluxo de dados dentro do GIS, ilustrado no fluxograma da Figura 10, começa com a entrada dos dados de cálculo e segue através de várias etapas, incluindo o refinamento e a validação, até alcançar a interface do usuário.

Figura 10 - Fluxograma de dados do sistema GIS.



Fonte: O Autor (2024).

Para facilitar a organização e o acesso a essas informações, o sistema foi dividido em módulos específicos, que refletem as principais áreas de design e produção dos transformadores. Cada módulo é estruturado para operar de forma independente, porém integrada, permitindo que os engenheiros acessem e verifiquem informações relevantes a cada etapa do processo de design.

2.4.4.2 Módulos do GIS

O GIS é composto por diversos módulos e formulários que categorizam as informações e funcionalidades necessárias para o desenvolvimento do projeto. Entre os módulos e formulários principais estão:

- **Módulo de Enrolamento:** Engloba todos os dados do enrolamento das bobinas, tais como altura, diâmetros, peso entre outras características fundamentais para a finalização do projeto.
- **Formulário de Baixa Tensão e Alta Tensão:** Este módulo armazena informações essenciais sobre as especificações das bobinas, como o material, o modo de bobinagem e o número de espiras. Esses dados são fundamentais para garantir que as especificações técnicas estejam de acordo com os padrões e exigências de cada projeto.
- **Formulário da Parte Ativa:** Concentra dados sobre as propriedades físicas do transformador, como o peso total e o formato da bobina, elementos essenciais para cálculos estruturais e térmicos. Esse módulo fornece uma visão geral das características principais do transformador, permitindo uma análise detalhada das especificações técnicas.

Os formulários descritos neste capítulo podem ser visualizados em destaque na Figura 11, que também ilustra o módulo de Validação em funcionamento, mostrando cartões de aviso que indicam mensagens de validação associadas ao projeto. Essa visualização facilita a identificação de quaisquer inconsistências, permitindo uma revisão ágil e precisa por parte da equipe de engenharia.

Cada módulo e formulário possui sua própria camada *View* e *ViewModel*, que trabalham em conjunto para assegurar que as validações sejam exibidas ao usuário de forma clara e organizada. Essa independência facilita a manutenção e a modificação dos componentes, permitindo que novas funcionalidades sejam adicionadas de maneira modular.

Figura 11 - Demonstração dos módulos e formulários do GIS

The screenshot displays the GIS software interface. On the left is a navigation tree with the following categories and sub-items:

- Engineering (highlighted with a red box)
 - Approval
 - Variant Configuration
 - Accessories
 - Checklist
 - Critical Analysis
- Command & Control
 - SAP
- Electrical
 - Bushings
 - Core Assembly
- Windings (highlighted with a red box)
 - LV1
 - HV1
 - Nameplate
 - Datasheet
 - SAP
- Mechanical
 - Enclosure
 - Active Part
 - SAP
- Bill of Materials
 - Validate (highlighted with a green box, showing counts: 1, 16, 1)

The main area shows two forms:

Identification

Quantity [n]	1	Item	-
Client	[Redacted]		
Factory	BLUMENAU		
Standard / Specification	ABNT - NBR 10295/1988		
Specification applied	[Red X icon]		
Oil preservation system	NA		
Oil preservation system	NA		
Saved date	qui 19/09/2024 13:03		
Author	[Redacted]		

Electrical characteristics

Rated powers [kVA]	1000
Cooling	AN
Frequency [Hz]	60
Phases	Trifásico
Connection group	Dyn1
Inrush [pu]	9.1
Impedance [%]	6.211

Reactor: ☐

Fonte: o Autor (2024).

2.5 ARTIGOS RELACIONADOS

A busca por artigos científicos relacionados ao tema deste trabalho foi conduzida de maneira sistemática, utilizando estratégias que garantem a abrangência e a qualidade das fontes selecionadas. Foram consultadas bases de dados reconhecidas, como *IEEE Xplore*, *Science Direct* e *Scopus*, devido à sua relevância e confiabilidade para pesquisas científicas. A formulação das buscas incluiu o uso de palavras-chave específicas, combinadas com operadores booleanos, tais como "validação de dados", "ferramentas da qualidade", "*Poka yoke*", "*Lean Manufacturing*", "*Six Sigma*" e "desenvolvimento de *software*", ajustadas ao foco do trabalho.

Como resultado desse processo, foram selecionados artigos que fornecem uma base sólida e multidisciplinar para o desenvolvimento deste trabalho, visto na Tabela 1, contribuindo com perspectivas relevantes sobre ferramentas da qualidade, automação e

melhoria contínua, todas integradas ao sistema de validação de dados para transformadores a seco.

O estudo de Barber et al. (2001) destaca a aplicação do FMEA no ciclo de desenvolvimento de *software*, enfatizando a identificação precoce de defeitos em requisitos e a correção antes da implementação, utilizando simulações e análises arquitetônicas. Essa abordagem é diretamente aplicável ao sistema proposto, pois enfatiza a detecção de inconsistências nas especificações técnicas de transformadores logo nas etapas iniciais do projeto, evitando retrabalhos e falhas em estágios avançados.

Biessmann et al. (2021) e Lwakatare et al. (2021) exploraram a validação automatizada de dados em sistemas de aprendizado de máquina, abordando desafios de implementação e propondo ferramentas robustas para garantir integridade e consistência dos dados. Essas contribuições reforçam a importância de mecanismos automatizados para validação de dados, um pilar central do sistema de controle de qualidade proposto neste trabalho, evidenciando como práticas bem estruturadas podem reduzir variações e prevenir falhas críticas em ambientes industriais.

Além disso, os princípios do *Six Sigma* e do *Kaizen* foram amplamente discutidos por Zimmermann e Bohomol (2023), que aplicam o ciclo DMAIC em um hospital brasileiro para otimizar fluxos críticos, destacando melhorias significativas na qualidade e na eficiência dos processos. A abordagem estruturada do DMAIC serve como modelo para o desenvolvimento do sistema de validação, permitindo um controle rigoroso das especificações técnicas e promovendo a melhoria contínua no contexto industrial.

O artigo de Moraes e Dal Forno (2022) acrescenta uma perspectiva prática sobre a aplicação de dispositivos *Poka yoke* na indústria têxtil. Os autores destacam como esses dispositivos auxiliam na prevenção de erros humanos e na garantia de um controle mais rígido sobre os processos produtivos, reduzindo falhas e desperdícios. Essa abordagem se conecta diretamente ao sistema de validação de dados proposto, que também utiliza dispositivos preventivos para garantir que inconsistências sejam detectadas antes de impactarem a produção.

A convergência dessas abordagens teóricas e práticas fortalece o escopo deste trabalho, evidenciando como a integração de ferramentas de qualidade com tecnologias de automação pode promover eficiência, redução de falhas e melhoria contínua em processos industriais. O sistema de validação proposto alinha-se diretamente com essas diretrizes, consolidando-se como uma solução robusta e inovadora para a gestão de qualidade na fabricação de transformadores a seco.

Tabela 1 - Relação de artigos para referência com foco em ferramentas da qualidade.

Referência	País	Ferra-menta	Objetivo ou Resultado
Bayers. 1994	Estados Unidos	Poka Yoke	Eliminação de defeitos no processo de fabricação e desenvolvimento de software, substituindo inspeções tradicionais por dispositivos que previnem erros na origem.
Barber, Graser, Holt. 2001	Estados Unidos	FMEA	Identificação precoce de defeitos nos requisitos e correção antes da implementação, utilizando simulação e análise de propriedades de software,
Tonini, Spínola, Laurindo. 2006	Brasil	DMAIC adaptado para desenvolvimento de <i>software</i> .	Desenvolvimento de um roteiro específico para aplicação do <i>Six Sigma</i> em projetos de software, destacando melhorias organizacionais e de processo.
Redzic, Baik. 2006	Coreia do Sul e Estados Unidos	<i>Six Sigma</i>	Redução de 90% nos defeitos severos em dois anos e melhoria na fidelidade do cliente por meio de mudanças estratégicas e táticas no desenvolvimento de software.
Sarpiri, Gandomani, Mani. 2021	Irã	Modelo híbrido Scrum e <i>Six Sigma</i>	Aumento no desempenho da equipe e satisfação do cliente, porém, com aumento no retrabalho e no tempo de entrega devido à integração das metodologias.
Biessmann, Golebiowski, Rukat, Lange, Schmidt. 2021	Estados Unidos	<i>Six Sigma</i>	Identificação dos desafios de validação de dados em sistemas de aprendizado de máquina, com desenvolvimento de soluções práticas e revisão de lacunas técnicas e culturais para melhor integração em aplicações industriais.
Lwakatare, Ranged, Crnkovic, Bosch. 2021	Suécia	5S e <i>Six Sigma</i>	Desenvolvimento de uma estrutura de validação de dados para projetos de machine learning e identificação de melhores práticas e barreiras para adoção em ambientes industriais.
Moraes, Dal Forno. 2022	Brasil	Poka Yoke	Aplicação do método Poka Yoke na indústria têxtil
Zimmermann, Bohomol. 2023	Brasil	Metodologia Lean <i>Six Sigma</i>	Redução do tempo médio de alta de pacientes em 61%, melhorando a eficiência e reduzindo desperdícios no fluxo de alta hospitalar.

Fonte: O Autor (2024).

3 METODOLOGIA

A metodologia do trabalho seguiu um roteiro estruturado, iniciando com uma revisão bibliográfica sobre metodologias como *Poka yoke*, *Six Sigma* e *Kaizen*, e explorando padrões de desenvolvimento de *software* como o Model-View-ViewModel (MVVM) e *Composite*. A implementação foi realizada utilizando a linguagem C# e o *framework* WPF visto que são ferramentas padrão dentro da empresa e que permitem a criação de uma interface gráfica interativa e modular. O sistema foi integrado ao GIS (Global Integration System), uma plataforma interna da empresa, para realizar validações automatizadas de especificações técnicas das bobinas, garantindo conformidade desde as etapas iniciais do projeto. Para organização e categorização das validações, foram utilizados fluxos hierárquicos e métodos baseados nos padrões de projeto, possibilitando a escalabilidade e a adaptação do sistema a novas demandas. O desenvolvimento foi guiado pelo método DMAIC, assegurando um controle rigoroso das etapas de análise e implementação, e pelo princípio de melhoria contínua, promovendo inovação e eficiência no processo produtivo. Essa estrutura permite que o sistema seja replicado em outros contextos industriais, ajustando-se às necessidades específicas de cada aplicação.

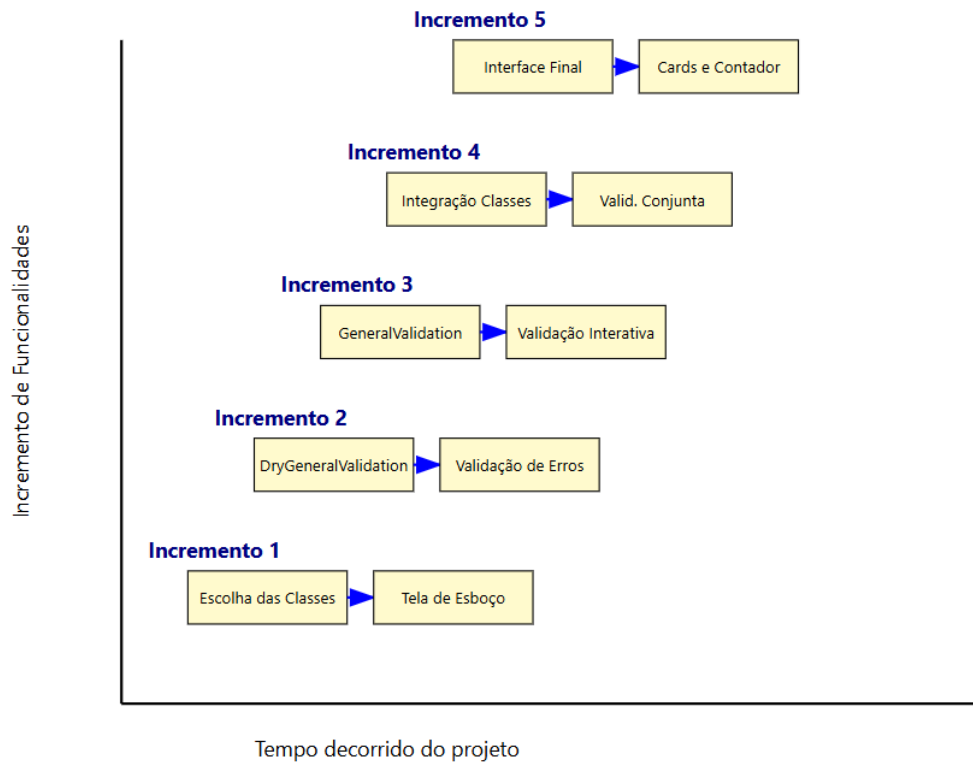
Além disso, detalhamos o fluxo de validação implementado no sistema, destacando as classes centrais, *GeneralValidation* e *DryGeneralValidation*, que centralizam e executam as verificações de conformidade de dados. Por fim, exploramos como ferramentas de qualidade, como *Poka yoke*, *Six Sigma* e *Kaizen*, foram adaptadas para o ambiente de desenvolvimento de *software*, contribuindo para a prevenção de erros, a redução de variabilidade e a promoção de melhorias contínuas. A descrição metodológica apresentada neste capítulo busca não apenas esclarecer as escolhas realizadas, mas também fornecer um guia para a replicação deste sistema em outros contextos industriais.

3.1 METODOLOGIA INCREMENTAL DE DESENVOLVIMENTO DE SOFTWARE

O desenvolvimento do sistema de validação foi conduzido utilizando uma metodologia incremental, uma abordagem amplamente discutida por Sommerville (2011) como eficaz para sistemas que necessitam de entregas graduais e flexíveis. Esse método permite o progresso em etapas bem definidas, com entregas funcionais a cada incremento, promovendo validações frequentes e a possibilidade de incorporar melhorias contínuas ao longo do processo. Essa abordagem foi escolhida devido à sua capacidade de gerenciar a

complexidade do sistema, garantindo que funcionalidades fossem entregues de forma progressiva e confiável, reduzindo riscos e facilitando a adaptação às necessidades do projeto. A seguir na Figura 12 podemos visualizar as etapas da metodologia ao longo do desenvolvimento do projeto.

Figura 12 - Modelo de desenvolvimento incremental.



Fonte: O Autor (2024)

O processo começou com a definição da estrutura inicial, onde foi decidido onde as classes *DryGeneralValidation* e *GeneralValidation* seriam implementadas, estabelecendo a base para o funcionamento do sistema. Simultaneamente, foi desenvolvida uma tela de esboço para validação, permitindo visualizar as primeiras interações entre o sistema e os usuários.

Na segunda etapa, foi implementada a classe *DryGeneralValidation*, que trouxe a funcionalidade inicial de validação automática, capaz de identificar erros no momento em que os dados do projeto são carregados. Esse passo garantiu que problemas fossem detectados ainda nas etapas iniciais do processo. A validação de erros nessa fase foi fundamental para estruturar o sistema de maneira sólida.

Em seguida, na terceira etapa, a classe *GeneralValidation* foi desenvolvida para permitir a validação interativa. Essa funcionalidade adicionou mensagens de erro ou as

removeu conforme o usuário corrigia os problemas em tempo real. Esse incremento foi um marco na usabilidade, garantindo maior agilidade e precisão na interação com o sistema.

A quarta etapa consistiu na integração das classes, onde as funcionalidades das classes *DryGeneralValidation* e *GeneralValidation* foram sincronizadas para trabalharem de forma conjunta. Isso possibilitou uma validação completa e consistente, cobrindo todas as necessidades de verificação do sistema.

Por fim, a última etapa foi dedicada à entrega final da interface completa. Nessa fase, foram implementados os cartões de validação e o contador de mensagens no botão *Validate*, permitindo uma visualização clara e organizada das mensagens de erro. Essa entrega consolidou o sistema como uma solução prática e eficiente para os engenheiros projetistas.

3.2 DESENVOLVIMENTO DA FUNÇÃO DE VALIDAÇÃO

A função de validação de dados é o núcleo do sistema GIS, garantindo que todas as especificações técnicas dos transformadores estejam corretas antes de serem liberadas para a produção. Para alcançar essa confiabilidade, foram adotadas metodologias de controle de qualidade, como *Poka yoke*, *Six Sigma* e *Kaizen*, adaptadas para o ambiente de desenvolvimento de *software*. Essas ferramentas permitiram construir uma estrutura robusta de validação automatizada, minimizando erros nas especificações ainda na fase de projeto, o que se alinha ao conceito de prevenção de falhas em ambientes industriais (Zimmerman, Bohomol, 2023).

Para estruturar a função de validação de forma modular e escalável, foram desenvolvidas classes específicas, como *GeneralValidation* e *DryGeneralValidation*, que centralizam as regras e verificações necessárias para assegurar a conformidade dos projetos. Essas classes foram organizadas de modo a facilitar a implementação e a categorização de validações hierárquicas, permitindo que o sistema se adapte a diferentes exigências técnicas sem comprometer sua integridade. A abordagem modular e flexível se inspira no princípio *Kaizen* de melhoria contínua, que incentiva a evolução incremental e o engajamento da equipe para aprimorar constantemente o sistema de validação, tal como discutido por Barber et al. (2001), que exploram a eficácia do *feedback* contínuo em processos de controle de qualidade.

O uso de validações automatizadas em um sistema de engenharia é especialmente relevante em setores onde a precisão é essencial, como mostrado no estudo de Lwakatare et al. (2021) sobre validação de dados em aprendizado de máquina. Assim como nesses projetos, a validação de dados no GIS assegura que quaisquer inconsistências sejam detectadas e

corrigidas antes que avancem no ciclo produtivo, evitando retrabalho e garantindo conformidade com as especificações estabelecidas.

3.2.1 Pilares do sistema de validação

O sistema de validação implementado no GIS adota uma abordagem modular, baseada no padrão MVVM, para garantir que todas as informações estejam em conformidade desde as etapas iniciais do projeto. Duas classes centrais, *GeneralValidation* e *DryGeneralValidation*, desempenham papéis complementares nesse processo, executando validações em diferentes níveis e garantindo a consistência dos dados ao longo do ciclo de vida do sistema.

3.2.1.1 *GeneralValidation*

A classe *GeneralValidation* funciona como um núcleo centralizador para as validações gerais do sistema. Ela foi projetada para armazenar e gerenciar regras de validação aplicáveis a diferentes módulos e formulários do GIS. Além disso, oferece suporte para o método principal *AddValidationItem*, que é utilizado para registrar, atualizar e remover mensagens de validação dinamicamente. Este método é integrado à interface *IDataErrorInfo*, permitindo que as validações sejam exibidas e atualizadas em tempo real nas *ViewModels* do projeto.

O método *AddValidationItem* opera da seguinte maneira:

- Busca por itens existentes: Verifica se uma validação semelhante já está registrada, com base em critérios como módulo, formulário, tipo de validação e campo.
- Atualização ou exclusão: Caso a validação já exista, ela é atualizada com uma nova mensagem ou removida, caso a mensagem esteja vazia.
- Inserção de novos itens: Se a validação ainda não estiver registrada, ela é criada e adicionada à lista de validações, que é automaticamente ordenada para facilitar a priorização.
- Notificação de alterações: Após qualquer modificação, o método aciona o evento *OnErrorsChanged*, garantindo que as mudanças sejam refletidas na interface do usuário.

Este método é amplamente utilizado em diferentes *ViewModels* para verificar e validar dados inseridos pelos usuários, garantindo que inconsistências sejam identificadas e exibidas no momento da interação.

3.2.1.2 *DryGeneralValidation*

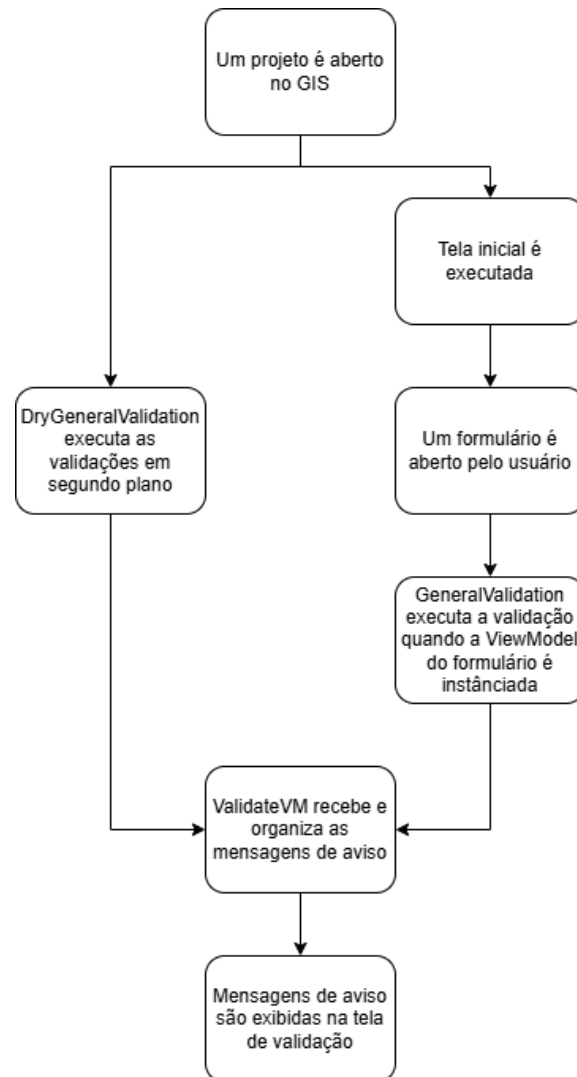
A classe *DryGeneralValidation* estende as funcionalidades de *GeneralValidation*, com o objetivo de realizar as mesmas validações gerais de forma paralela. Sua principal característica é garantir que as validações sejam executadas automaticamente sempre que um projeto é carregado no sistema. Isso assegura que o estado inicial dos dados seja validado antes mesmo de qualquer interação direta do usuário.

Além disso, a *DryGeneralValidation* é acionada novamente toda vez que uma *ViewModel* é instanciada, ou seja, quando o usuário abre uma interface específica no GIS. Este comportamento permite que as validações sejam sincronizadas em tempo real, garantindo que, ao corrigir um problema na interface, as mensagens de validação sejam imediatamente atualizadas. Isso é possível graças à integração entre as camadas *Model* e *ViewModel*, que utilizam o mesmo conjunto de regras para assegurar consistência. A Figura 13 apresenta o fluxo de validação dentro do GIS.

O sistema de validação foi projetado para garantir que todas as especificações técnicas estejam em conformidade ao longo do processo de desenvolvimento do projeto. O fluxo de validação é dividido em três etapas principais, que operam de forma integrada para oferecer precisão e eficiência.

Na primeira etapa, as validações ocorrem automaticamente durante o carregamento do projeto. Nesse momento, a classe *DryGeneralValidation* é responsável por executar verificações em segundo plano, assegurando que os dados sejam analisados antes de qualquer interação do usuário. Os resultados dessas validações são armazenados em uma lista central e exibidos no painel de validação, permitindo que os engenheiros tenham uma visão clara dos potenciais problemas logo no início do processo.

Figura 13 - Fluxograma de validação no sistema.



Fonte: O Autor (2024).

Na segunda etapa, as validações são realizadas durante as interações do usuário com a interface gráfica. As *ViewModels* utilizam a interface *IDataErrorInfo*, que chama o método *AddValidationItem* da classe *GeneralValidation* para atualizar as mensagens de validação em tempo real. Esse mecanismo permite que inconsistências sejam detectadas imediatamente à medida que os dados são inseridos ou alterados, garantindo uma resposta rápida e precisa ao usuário.

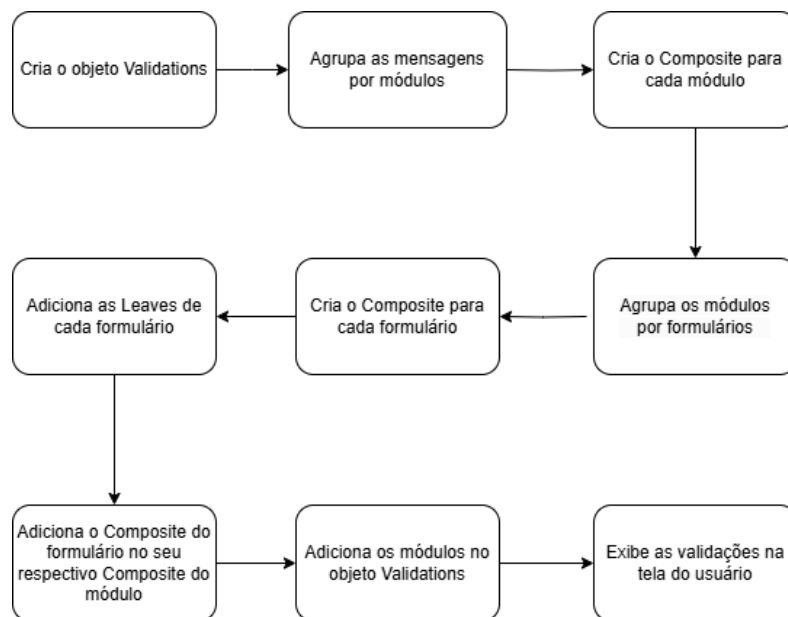
Por fim, o sistema opera em um estado de sincronização contínua. Quando o usuário corrige um erro diretamente na interface, a validação é novamente executada na *ViewModel*, atualizando as mensagens e refletindo as alterações no painel de validação. Essa funcionalidade assegura que o sistema esteja sempre sincronizado, proporcionando uma experiência de validação dinâmica e eficiente.

Esse fluxo de validação integrado não apenas aprimora a confiabilidade dos dados inseridos, mas também melhora a eficiência operacional, reduzindo o tempo necessário para identificar e corrigir erros. Além disso, a combinação de validações automáticas e interativas promove maior segurança no processo, assegurando que as especificações técnicas estejam em conformidade desde as etapas iniciais até a conclusão do projeto.

3.2.1.3 *ValidateVM*

A classe *ValidateVM* é uma peça essencial do sistema de validação, encarregada de organizar e exibir as mensagens de aviso na tela de validação. Seu funcionamento se baseia no padrão de projeto *Composite*, já discutido nos capítulos anteriores, o que possibilita a criação de uma estrutura hierárquica para os avisos. Aqui, o padrão é aplicado para agrupar as mensagens de maneira lógica e eficiente, garantindo uma experiência clara e organizada ao usuário. Este processo ocorre em três etapas principais dentro do método principal *ToValidationComponent*, como apresentado na Figura 14.

Figura 14 - Fluxograma de criação das validações na *ViewModel* de validação.



Fonte: O Autor (2024).

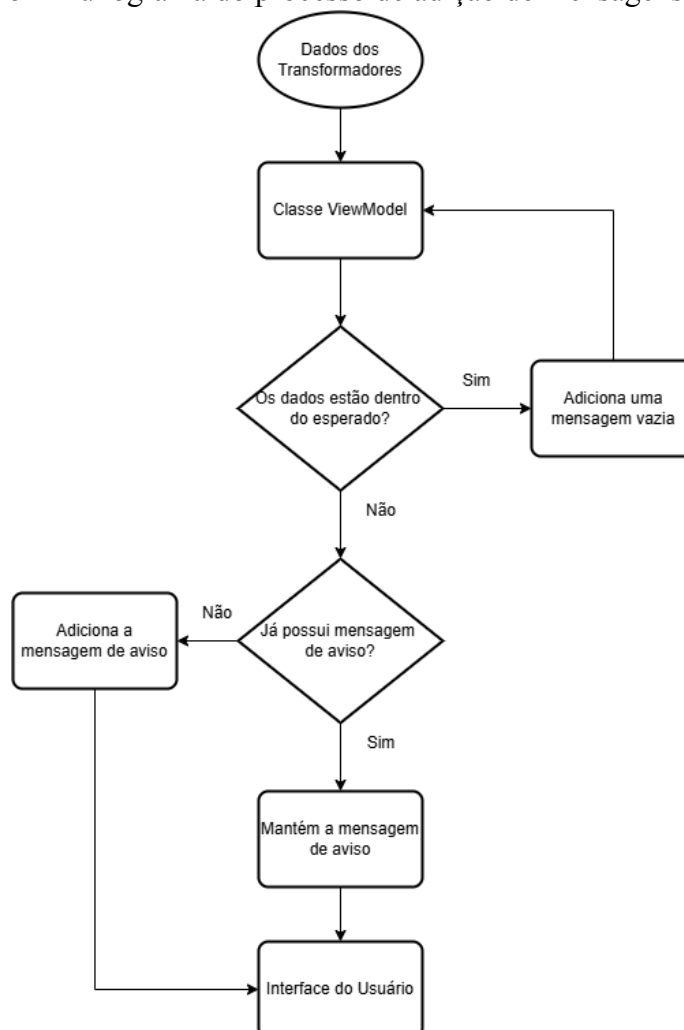
O uso do padrão *Composite*, conforme detalhado anteriormente no trabalho, destaca-se por sua capacidade de lidar com estruturas complexas de forma uniforme. Ao aplicar esse padrão, a *ValidateVM* organiza os elementos em uma estrutura flexível, onde módulos,

formulários e mensagens são tratados de maneira homogênea. Essa abordagem facilita tanto a manutenção do código quanto a interação do usuário com o sistema.

3.2.2 Funcionamento da validação

O processo de validação segue um fluxo bem definido, conforme ilustrado no fluxograma da Figura 15, que detalha a lógica de verificação do sistema. O processo inicia com a coleta de dados dos transformadores, passando pela classe *DryGeneralValidation*, onde as regras de validação são aplicadas. A interface utiliza o padrão *IDataErrorInfo* para exibir mensagens classificadas como “Erro” diretamente ao usuário, destacando visualmente quaisquer inconsistências nas especificações, o que permite uma rápida identificação e correção de problemas. Enquanto as demais validações são exibidas somente na tela de validação.

Figura 15 - Fluxograma do processo de adição de mensagens de aviso.



Fonte: O Autor (2024).

3.2.3 Tipos de validação

O sistema de validação de dados foi projetado para executar uma série de verificações específicas, classificadas de acordo com o nível de criticidade, para orientar os engenheiros no processo de design dos transformadores. Conforme ilustrado na Figura 12, as validações são categorizadas em quatro tipos principais: Erro, Atenção, Sucesso e Informativo. Cada categoria possui uma função específica no processo, facilitando a detecção e correção de inconsistências antes que o projeto avance para a produção.

Validações marcadas como Erro representam problemas críticos, como discrepâncias graves nas dimensões das bobinas que poderiam comprometer a funcionalidade e a segurança do produto final. Já as validações de Atenção indicam inconsistências menos severas que, embora não impeçam o andamento do projeto, exigem verificação adicional devido às especificidades de cada projeto. As validações de Sucesso confirmam que os parâmetros críticos atendem aos requisitos estabelecidos, enquanto as de Informativo apresentam dados adicionais importantes para o processo produtivo, como o uso de moldes específicos ou requisitos personalizados de clientes. Essa classificação, exemplificada na Figura 16, facilita o trabalho dos engenheiros, permitindo uma visão clara das validações necessárias e das ações a serem tomadas em cada etapa do projeto.

Figura 16 - Exemplo de exibição das mensagens de validação.

✖	Campo de Validação com Erro	Exemplo de mensagem de Erro
⚠	Campo de Validação com Alerta	Exemplo de mensagem de Alerta
✔	Campo de Validação com Sucesso	Exemplo de mensagem de Sucesso
i	Campo de Validação com Informação	Exemplo de mensagem de Informação

Fonte: O Autor (2024).

3.2.4 Integração com a arquitetura MVVM

A arquitetura MVVM (*Model-View-ViewModel*) desempenhou um papel crucial no design e na operação das classes de validação. Como uma evolução do padrão MVC (*Model-View-Controller*), o MVVM separa claramente as responsabilidades do sistema em três camadas principais:

- *Model*: Representa os dados e a lógica de negócios. No caso do sistema GIS, o *Model* armazena as especificações técnicas dos transformadores, garantindo a integridade e a precisão dos dados inseridos.
- *View*: Refere-se à interface gráfica do usuário, desenvolvida com o WPF, permitindo que os engenheiros visualizem as validações e recebam notificações de erros de forma interativa e prática.
- *ViewModel*: Atua como uma camada intermediária entre *Model* e *View*, processando os dados necessários para exibição e recebendo comandos do usuário. No sistema de validação, o *ViewModel* capta as validações das classes *GeneralValidation* e *DryGeneralValidation* e as organiza para uma exibição clara e eficiente na interface.

Essa separação de responsabilidades proporcionada pelo MVVM oferece diversas vantagens. O baixo acoplamento entre as camadas permite que as validações sejam executadas de forma independente da interface, possibilitando que o sistema seja modificado ou expandido sem impactar a exibição ou outras funcionalidades. Esse aspecto é especialmente útil em sistemas que exigem alta modularidade e manutenção frequente, conforme descrito no estudo de um aplicativo de diagnóstico para perda auditiva, onde o MVVM proporcionou flexibilidade para alterações na GUI sem comprometer a lógica de negócios e facilitou o desenvolvimento independente da interface e da lógica subjacente.

Além disso, a estrutura do MVVM aumenta a testabilidade, pois a divisão de responsabilidades permite testar cada componente de forma isolada. As classes *GeneralValidation* e *DryGeneralValidation* podem ser testadas individualmente, garantindo a eficácia das validações. Outro benefício é a facilidade de manutenção e expansão: ajustes e melhorias podem ser implementados diretamente no *Model* ou *ViewModel* sem exigir grandes modificações no restante do código, promovendo uma arquitetura flexível e escalável.

3.2.5 Veracidade dos dados

A integração das classes de validação com a arquitetura MVVM é essencial para a garantia de veracidade e integridade dos dados. As classes *GeneralValidation* e *DryGeneralValidation* asseguram que os parâmetros técnicos dos transformadores estejam dentro dos limites especificados antes que o projeto seja finalizado. Esse processo de validação prévia evita que informações incorretas sejam enviadas para a produção, o que poderia comprometer a qualidade e a segurança do produto.

O sistema é projetado para ser flexível e escalável, permitindo a fácil adição de novas regras de validação conforme as normas e exigências técnicas evoluem. O uso do padrão *Composite* na organização das validações facilita a visualização e categorização dos problemas, proporcionando aos engenheiros uma compreensão clara de onde estão as falhas e quais correções são necessárias. A separação das camadas no MVVM garante que, mesmo que a interface ou outras funcionalidades precisem ser alteradas, a lógica de validação se mantém sólida e confiável. Essa estrutura preserva a integridade dos dados ao longo de todo o processo de projeto, prevenindo erros dispendiosos e mantendo a conformidade do projeto com os padrões fabris.

3.3 APLICAÇÃO DAS FERRAMENTAS DA QUALIDADE NO SISTEMA DE VALIDAÇÃO NA EMPRESA DE TRANSFORMADORES

3.3.1 Caracterização da empresa

Este trabalho foi desenvolvido na unidade da WEG em Blumenau, Santa Catarina, que atua na fabricação de transformadores de potência, transformadores de distribuição, transformadores a seco, reatores e chaves seccionadoras. A unidade, que possui um ambiente altamente técnico e exigente, busca continuamente a máxima qualidade em seus produtos, visando a redução de falhas em todas as etapas do processo produtivo.

Nesse contexto, o sistema de validação de dados implementado alinha-se ao compromisso da empresa com a excelência, integrando ferramentas da qualidade ao *software* utilizado nos projetos de engenharia. Essa iniciativa reforça a importância da gestão da qualidade como um elemento central para assegurar a confiabilidade dos produtos entregues pela empresa, promovendo eficiência e inovação alinhadas às necessidades do mercado.

3.3.2 Sistema de validação

A validação de dados desempenha um papel fundamental no desenvolvimento de projetos de transformadores a seco, garantindo que todas as especificações técnicas estejam em conformidade antes que o projeto avance para a produção. Para assegurar essa conformidade, o sistema de validação automatizado foi integrado ao GIS (Global Integration System) da empresa, permitindo a verificação precisa de parâmetros críticos, como o diâmetro interno e externo das bobinas e o número de espiras. Baseando-se em metodologias de

controle de qualidade, como o *Poka yoke* e o *Six Sigma*, o sistema detecta inconsistências e classifica as mensagens de validação em quatro categorias:

"Erro", "Atenção", "Sucesso" e "Informativo", cada uma indicada por uma cor específica que facilita a identificação do problema por parte dos engenheiros.

A categorização e a sinalização antecipada de falhas permitem que os dados sejam corrigidos ainda na fase de projeto, impedindo que erros cheguem ao chão de fábrica. Essa abordagem evita desperdícios de matéria-prima e horas de trabalho, alinhando-se aos princípios do *Poka-Yoke* ao prevenir falhas e garantir que os dados sejam mantidos dentro dos limites de especificação desde o início do processo (Lwakatare et al, 2021). Além disso, a filosofia *Kaizen* de melhoria contínua é aplicada por meio de um sistema de feedback visual constante, onde cada tipo de validação é representado por uma cor. Esse recurso é exibido em cartões que podem ser visualizados em qualquer tela do sistema, proporcionando aos engenheiros uma visão em tempo real do status atual do projeto e das validações associadas.

A Figura 15 ilustra exemplos de validação exibidos no módulo *Validate*, demonstrando a funcionalidade dos cartões e a exibição de cada tipo de validação com sua respectiva cor e quantidade. Essa visualização facilita a tomada de decisões rápidas e assertivas, contribuindo para a eficiência e a qualidade do processo de design dos transformadores.

Figura 17 - Exibição das validações no módulo Validate.



Fonte: O Autor (2024).

As notas de rodapé da Figura 15 são:

- Aviso de Erro (ícone vermelho): Pode impactar negativamente o projeto, incluindo documentos técnicos, desenhos, lista de materiais, etc.
- Aviso de Alerta (ícone amarelo): Esteja ciente deste problema para evitar resultados inesperados. Pode exigir uma verificação dupla para alinhar com os padrões do projeto.
- Aviso de Sucesso (ícone verde): Validação bem-sucedida. Os dados estão em conformidade com os requisitos e padrões do projeto, garantindo confiabilidade e precisão.
- Aviso de Informação (ícone azul): Fornece detalhes suplementares relevantes ao projeto. Útil para obter melhores insights e compreender o contexto.

A classificação para cada tipo de validação foi desenvolvida em colaboração com o time de engenharia, cuja experiência permitiu estabelecer parâmetros técnicos claros para diferentes tipos de projetos de transformadores. Essa equipe determinou os limites aceitáveis de variação para cada atributo relevante, identificando quando uma discrepância se qualificaria como um "Erro" crítico ou apenas um "Alerta". Um "Erro" é gerado sempre que

um parâmetro se desvia significativamente dos valores especificados, comprometendo a integridade ou a segurança do projeto, enquanto um "Alerta" indica uma variação menor, que ainda atende aos padrões mínimos de operação, mas merece atenção adicional para evitar potenciais problemas futuros.

A automação das verificações e o controle rigoroso de dados garantem que falhas potenciais sejam interceptadas o mais cedo possível no ciclo de desenvolvimento, minimizando retrabalho e contribuindo diretamente para a qualidade do produto final. Essa abordagem está alinhada com várias ferramentas de gestão de qualidade amplamente aplicadas em ambientes industriais, incluindo o *Poka yoke*, o *Six Sigma*, e o *kaizen*.

A implementação desse sistema de validação demonstra como ferramentas de qualidade, originalmente aplicadas em processos físicos de fabricação, podem ser adaptadas ao desenvolvimento de *software* para assegurar a confiabilidade dos dados e otimizar a eficiência dos processos produtivos. No GIS, cada ferramenta desempenha um papel específico e essencial: o *Poka yoke* atua como um mecanismo de prevenção de erros, identificando inconsistências nas especificações e impedindo que falhas avancem para a produção, similar ao seu uso no controle de qualidade hospitalar em ambientes críticos (Zimmermann e Bohomol, 2023). O *Six Sigma* contribui para a redução de variações e aprimoramento da eficiência por meio de controles rigorosos e limites definidos, seguindo uma abordagem comparável à utilizada na saúde e no desenvolvimento ágil de *software*, onde a precisão e a adaptação são cruciais (Lwakatare et al., 2021; Barber et al., 2001). Já o *Kaizen* incentiva a melhoria contínua e a participação ativa dos colaboradores, promovendo um ambiente de inovação constante, o que também foi observado como fundamental para *feedbacks* contínuos e preventivos em ciclos de desenvolvimento (Barber et al., 2001).

3.3.3 Poka yoke

O *Poka yoke* é uma técnica essencial para prevenir erros em processos de produção, originalmente desenvolvida no Sistema Toyota de Produção. O termo significa "à prova de erros", visa eliminar ou minimizar falhas humanas, criando mecanismos que previnem a ocorrência de erros ou identifiquem inconsistências antes que se tornem problemas críticos. Essa abordagem é amplamente utilizada em ambientes industriais para assegurar a qualidade e a confiabilidade dos processos, como descrito por Bayers (1994), que destaca o papel dos dispositivos à prova de erros na redução de defeitos em etapas essenciais da produção.

No contexto do sistema de validação de dados desenvolvido para o GIS, o conceito de *Poka yoke* foi implementado de forma digital para interceptar erros críticos nas dimensões das bobinas e outras especificações técnicas ainda na fase de projeto. O sistema atua como um "dispositivo" de *Poka yoke*, emitindo mensagens de erro ou avisos sempre que dados fora das especificações são inseridos. Assim como os dispositivos físicos descritos por Bayers (1994), que impedem falhas antes que elas se manifestem no produto final, o sistema de validação digital identifica inconsistências em tempo real, possibilitando uma rápida correção por parte da equipe de engenharia.

Para os transformadores a seco, especificações incorretas, como diâmetro inadequado ou número incorreto de espiras, poderiam comprometer todo o processo de fabricação, resultando em desperdício de materiais e falhas no produto final. A integração do sistema de validação com o GIS permitiu que esses erros fossem visualizados de forma rápida e acessível por toda a equipe de engenharia, reforçando a filosofia do *Poka yoke* ao eliminar os erros na origem. Como observado por Bayers (1994), a prevenção de erros na etapa inicial não só assegura a conformidade das especificações, mas também reduz significativamente os custos com retrabalho e perda de material, benefícios igualmente obtidos com a implementação dessa filosofia no sistema de validação do GIS.

3.3.4 Six Sigma

A metodologia *Six Sigma* é amplamente reconhecida por sua abordagem estatística rigorosa para a redução de variações e defeitos em processos produtivos, com o objetivo de alcançar altos padrões de qualidade e eficiência. Originalmente aplicada em ambientes industriais complexos, onde o controle de qualidade é crítico, o *Six Sigma* fundamenta-se na análise detalhada das causas dos defeitos e na implementação de soluções eficazes para eliminá-los. Esse processo é organizado por meio do ciclo DMAIC (Definir, Medir, Analisar, Implementar e Controlar), que estrutura a metodologia em etapas focadas na identificação e mitigação de variações que possam comprometer a consistência e a qualidade do processo (Tonini, Spínola, Laurindo, 2006).

No contexto do sistema de validação de dados para transformadores, a implementação DMAIC se mostrou essencial para garantir que as especificações técnicas, como as dimensões das bobinas, estivessem dentro dos limites rigorosamente definidos. Os conceitos centrais de redução de variação e controle estatístico foram totalmente incorporados para aumentar a

precisão do sistema de validação. A aplicação do ciclo DMAIC permitiu que as verificações contínuas detectassem e corrigissem variações nas dimensões das bobinas, assegurando que os dados inseridos no sistema atendessem aos padrões estabelecidos antes de serem encaminhados para a produção.

Para ilustrar essa adaptação do ciclo DMAIC no sistema de validação, as etapas foram ajustadas da seguinte forma:

- Definir (D): No sistema atual, foi identificado que não existiam limites fabris claros, o que permitia o envio para a produção de peças com dimensões incompatíveis com os padrões de fabricação.
- Medir (M): Observou-se que a revisão manual dos dados era um processo demorado, exigindo muitas horas de análise para garantir a conformidade das especificações, o que impactava a eficiência da equipe de engenharia.
- Analisar (A): O histórico de produção mostrava que peças já haviam sido descartadas ou enviadas para retrabalho devido a falhas nas dimensões, indicando a necessidade de um sistema mais robusto de controle preventivo.
- Implementar (I): Foi então implantado o sistema de validação automatizado, que verifica se os dados do projeto estão de acordo com os limites fabris, reduzindo drasticamente a ocorrência de erros nas especificações antes de entrar na linha de produção.
- Controlar (C): A validação e o controle são realizados automaticamente toda vez que o engenheiro acessa o projeto, assegurando que os parâmetros críticos sejam checados e que o projeto esteja em conformidade antes de ser liberado.

A aplicação do DMAIC no sistema de validação segue as diretrizes da metodologia *Six Sigma* para promover uma padronização e um controle eficaz das especificações, conforme descrito por Tonini, Spínola e Laurindo (2006). Com a detecção automática de variações indesejadas, o sistema não só reduz a possibilidade de erros no produto final, mas também otimiza o tempo e os recursos empregados na fase de design, minimizando o retrabalho e o desperdício de materiais. A integração dos princípios do *Six Sigma* no desenvolvimento dos transformadores permite um processo mais eficiente e preciso, garantindo que as especificações sejam atendidas com consistência.

3.3.5 Kaizen

O *Kaizen*, que significa "mudança para melhor", é uma filosofia de melhoria contínua amplamente aplicada em processos industriais, incentivando todos os colaboradores a participar ativamente na busca por aprimoramentos incrementais. Esse conceito, originado no Japão, promove a criação de um ambiente colaborativo onde as melhorias são sustentáveis e progressivas, acumulando-se ao longo do tempo.

No sistema de validação de dados desenvolvido para o GIS, o *Kaizen* foi integrado como uma resposta direta ao programa de melhoria contínua da empresa. Onde os colaboradores são incentivados a propor novos desenvolvimentos e melhorias para toda e qualquer etapa em sua cadeia produtiva, desde processos administrativos até processos industriais. Assim, o *Kaizen* reforça a participação ativa dos colaboradores, assegurando que as melhorias não sejam isoladas, mas contínuas e acumulativas.

Por meio dessa filosofia, o sistema de validação não apenas captura problemas iniciais, mas também evolui para atender às novas demandas dos projetos e aos requisitos de qualidade. Com a prática do *Kaizen*, a validação de dados é aprimorada de maneira contínua, promovendo um sistema flexível e em constante aperfeiçoamento, garantindo que as especificações dos transformadores estejam sempre atualizadas e em conformidade com os padrões de eficiência da empresa.

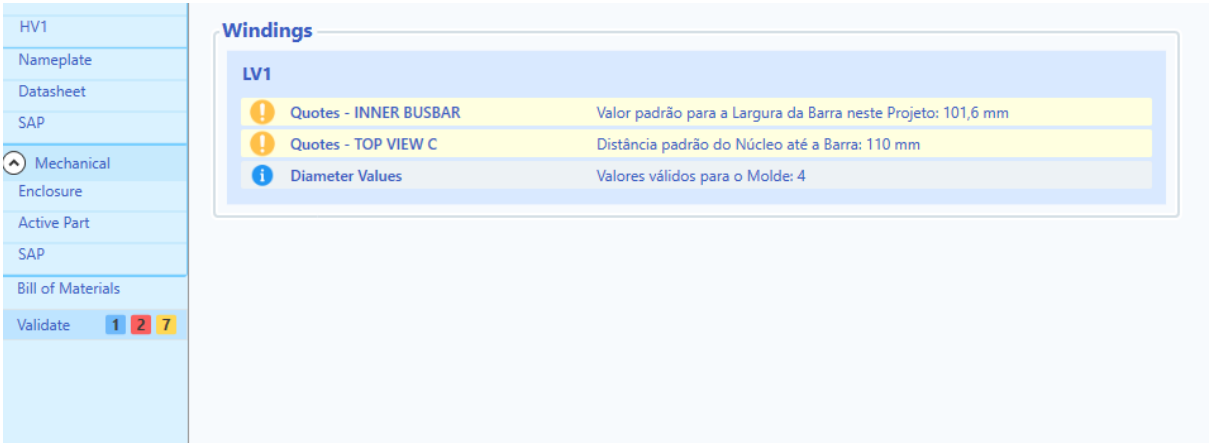
4 RESULTADOS E DISCUSSÃO

O sistema de validação desenvolvido foi projetado para garantir a conformidade técnica dos dados inseridos no *software* de projetos de transformadores. Por meio da integração de ferramentas da qualidade, como *Poka yoke* e DMAIC, o sistema contribui para a detecção precoce de inconsistências, promovendo a prevenção de erros e a melhoria contínua no processo produtivo. Esta seção discute a eficácia das ferramentas utilizadas e apresenta exemplos práticos do uso diário do sistema na empresa.

A implementação das ferramentas da qualidade no sistema demonstrou grande eficácia na mitigação de erros frequentes nos projetos de transformadores. O uso do *Poka yoke*, por exemplo, garantiu que valores incorretos fossem detectados automaticamente, prevenindo que avançassem para as etapas subsequentes do processo. Além disso, o ciclo DMAIC foi essencial para estruturar as validações, junto ao time experiente de engenharia. Essas ferramentas, adaptadas ao contexto do desenvolvimento de *software*, não apenas proporcionam maior precisão nos projetos, mas também reforçam a confiabilidade dos dados técnicos apresentados aos engenheiros.

O sistema é utilizado diariamente pelos engenheiros durante o desenvolvimento de projetos. Para ilustrar seu funcionamento, apresentamos três cenários que demonstram a eficiência do sistema no uso prático:

Figura 18 - Tela de validação com mensagens de alerta e informativo.



Fonte: O Autor (2024).

A Figura 18 apresenta os detalhes das mensagens geradas durante o processo de validação. Cada mensagem indica o campo específico que precisa ser corrigido, oferecendo orientações claras para os engenheiros resolverem as inconsistências.

Figura 19 - Tela do formulário LV1 sem a apresentação dos cards de validação.

The screenshot displays the LV1 form interface with the following sections:

- Left Sidebar:** A navigation menu with options: Engineering, Approval, Variant Configuration, Accessories, Checklist, Critical Analysis, Command & Control, SAP, Electrical, Bushings, Core Assembly, Windings, LV1, HV1, Nameplate, Datasheet, SAP, Mechanical, Enclosure, Active Part, SAP, and Bill of Materials.
- Sequence Table:**

Direction	Name	Qty.	Thickness	Total	Ø Ext.	Adj. Tol.
	Mandril	1	0	252		
	Barra	1	12,7	265		
	Bloco 1	1	10,97	287		
	Canal 1-T	1	8	309		
	Bloco 2	1	10,97	331		
	Bloco 3	1	10,97	367		
	Barra	1	12,7	379		
	Papel externo	3	0,23	381		
	Canal 1-T BT-AT	1	23,4	427		
	Cilindro 1 - BT-AT	1	1,2	430		
	Canal 2-T BT-AT	1	23,4	477		
- General Section:**
 - Voltage [V]: 460
 - Voltage class [kV]: 1,1
 - Connection: YN
 - Lead position: Superior
 - Neutral connection: Por dentro
 - Winding direction: Direita
 - Winding type: Laminated
 - Conductor type: Folio
 - Insulation material (Nom.): THERNOMID 2008 RVD
 - Has Half paper: ☐
- Conductors Section:**
 - Description: Alumínio (1350-Q)
 - Insulation [mm]: Desnudo
 - Yield strength [N/mm²]: 35
 - Width: 1200
 - Special: 1,9
 - Thickness: 0
 - Special: 0
 - Weight for all Phases [kg]: 293,1
 - Conductor 1 [mm]: 0
 - Conductor 2 [mm]: 0
 - Axial distance [mm]: 0
 - Parallel: Radial 1, Axial 1, Total 1
 - Material code: [dropdown]
- Dimensions Section:**
 - Coil head [mm]: 10
 - Mechanical height [mm]: 1200
 - Coil head [mm]: 10
 - Total height [mm]: 1220
 - Winding Factor: Axial 1, Radial 1,03
 - GIS: Phases 276, Bars 252
 - DTDS: Phases 276, Bars 252
- Quotes Section:**
 - Coil view:**

A	276
B	399
C	114
D	68
 - Side view:**

A	1200
B	10
Inner busbar	127x12,7
Outer busbar	127x12,7

Fonte: O Autor (2024).

Na situação da Figura 19, o formulário LV1 é preenchido com valores incorretos. Sem a validação ativa, esses erros poderiam passar despercebidos, avançando para as próximas etapas do projeto. Já na Figura 20 com a função *Validate* em funcionamento, o sistema realiza verificações automáticas, apontando os erros e alertas encontrados. Os cartões de aviso no botão indicam a quantidade de inconsistências, proporcionando uma visão clara do status de validação do formulário.

A adoção do sistema de validação trouxe benefícios significativos para a empresa, incluindo:

1. Redução de Erros: Detecção automática de inconsistências, evitando retrabalhos.
2. Agilidade no Processo: *Feedback* imediato sobre os dados inseridos, permitindo correções rápidas.
3. Confiança nos Dados: Garantia de que os dados enviados para produção estão em conformidade com os padrões técnicos.

Os exemplos apresentados anteriormente ilustram como o sistema de validação identifica e comunica inconsistências de maneira clara e eficiente, promovendo correções antes que os projetos avancem para as etapas de produção. Esses benefícios, que inicialmente

podem parecer focados apenas na organização e confiabilidade dos dados, refletem-se diretamente nos resultados financeiros e operacionais.

Figura 20 - Tela do formulário LV1 com a apresentação dos cards de validação.

The screenshot displays the LV1 form interface, which includes a sidebar menu on the left with categories like Engineering, Mechanical, and Electrical. The main area is divided into several sections:

- Sequence:** A table listing components and their quantities.

Direction	Name	Qty.	Thickness	Total	Ø Ext.	Adj. Tol.
Mandril		1	0	252		
Barra		1	12.7	265		
Bloco 1		1	10.97	287		
Canal 1-T		1	8	309		
Bloco 2		1	10.97	331		
Bloco 3		1	10.97	367		
Barra		1	12.7	379		
Papel externo		3	0.23	381		
Canal 1-T BT-AT		1	23.4	427		
Cilindro 1 - BT-AT		1	1.2	430		
Canal 2-T BT-AT		1	23.4	477		
- General:** Fields for Voltage [V] (460), Voltage class [kV] (1.1), Connection (YN), Winding type (Laminated), and Insulation material (THERNOMID 2008 RVD).
- Conductors:** Fields for Description (Aluminio (1350-O)), Insulation (Desnudo), Yield strength (35), and conductor dimensions (Width, Special, Thickness, Special, Weight for all Phases [kg]).
- Dimensions:** Fields for Coil head [mm] (10), Mechanical height [mm] (1200), Coil head [mm] (10), and Total height [mm] (1220).
- Coil:** A diagram showing the coil layout with dimensions A, B, C, and D.

Dimension	Value
A	276
B	399
C	114
D	68
- Quotes:** A diagram showing the coil layout with dimensions A, B, C, and D.

Dimension	Value
A	1200
B	10
Inner busbar	127x12.7
Outer busbar	127x12.7

Fonte: O Autor (2024).

Entre janeiro e julho de 2024, foram registradas 429 ocorrências de não conformidades na produção de transformadores a seco, totalizando um custo de R\$ 448.991,76 para a empresa, conforme apresentado na Tabela 2. Entre essas ocorrências, destacam-se 29 não conformidades relacionadas às bobinas, das quais 10 casos necessitaram de retrabalho, gerando um impacto financeiro de R\$ 10.172,94, o que corresponde a 2,33% do total de não conformidades registradas no período. Os outros 19 casos foram remanejados para projetos semelhantes ou sucateados sem custos adicionais, totalizando um impacto financeiro zero. Dessa forma, considerando apenas os custos das não conformidades tratadas pelo sistema, temos 2,26% do custo total das falhas durante o período.

Analisando especificamente as não conformidades envolvendo as dimensões de bobinas e a parte elétrica, observamos que estas representam 8% dos casos que geraram custos de retrabalho para a empresa. Essa análise reforça a relevância de abordar essas falhas de maneira sistemática, especialmente aquelas relacionadas às especificações técnicas, como número de espiras, diâmetros e materiais.

Tabela 2 - Tabela de QMs de transformadores a seco entre janeiro e julho de 2024.

Quantidade	Não conformidade	Custo (R\$)
29	Relação direta com as dimensões das Bobinas	10.172,94
83	Envolvendo ligamento ou isolamento elétrico falho	39.007,39
96	Outras falhas envolvendo o projeto elétrico	301.708,36
128	Referente a acabamento ou aspectos físicos estéticos	58.953,82
93	Referente a componentes danificados	39.069,25
429	Valor total	448.991,76

Fonte: O Autor (2024).

O sistema de validação automatizado proposto possui a capacidade de realizar verificações automatizadas dessas especificações, o que poderia ter eliminado as 10 ocorrências relacionadas às bobinas com custo de retrabalho. A implementação do sistema teria gerado uma economia direta de R\$ 10.172,94 no período de seis meses e uma projeção anual de R\$ 20.345,88, conforme demonstrado na Tabela 3. Esses resultados reforçam o impacto positivo do uso de ferramentas de qualidade e automação no contexto industrial.

Tabela 3 - Não conformidades com relação direta às dimensões de Bobinas.

Custo (R\$)	Causa
391,64	Bobina com espiras a menos
1.958,20	Bobina com espiras a menos
587,46	Bobina com espiras a menos
160,30	Bobina com uma espira a menos
508,52	Bobina furo radial externo maior
587,46	Relação de espiras BT
587,46	BT Material errado
238,69	Bobina BT diâmetro maior
160,30	Espiras a menos
4.992,91	Bobina com diâmetro externo fora do limite
10.172,94	Valor total de janeiro até julho

Fonte: O Autor (2024).

Além disso, existe um potencial significativo para a expansão da validação automatizada a outras categorias de falhas. Entre as 208 não conformidades relacionadas ao projeto elétrico, 96 falhas adicionais envolvendo ligamentos ou isolamento elétrico falhos correspondem a 22,38% do total de não conformidades registradas. Com a ampliação das funcionalidades do sistema, seria possível abordar essas falhas críticas no futuro, promovendo uma redução ainda mais expressiva dos custos associados a retrabalhos e desperdícios.

5 CONCLUSÃO

Este trabalho apresentou o desenvolvimento e a implementação de um sistema automatizado para validação de dados no contexto da fabricação de transformadores a seco, com foco na redução de falhas nas dimensões das bobinas e na otimização do processo de revisão de dados. A solução proposta foi integrada ao GIS, um *software* interno utilizado na gestão de projetos de engenharia, e utilizou padrões de desenvolvimento consagrados, como o MVVM e o *Composite*, para estruturar e validar dados de forma eficiente e escalável.

Ao longo do estudo, foi possível demonstrar que a aplicação de ferramentas da qualidade, como o *Poka yoke* e os princípios do *Six Sigma*, desempenhou um papel crucial na identificação e eliminação de falhas no processo. A adoção de práticas orientadas à qualidade resultou em melhorias significativas, tanto em termos de precisão nos dados validados quanto na redução de custos associados a erros de projeto e horas de trabalho.

Os resultados alcançados reforçam a importância de integrar ferramentas de qualidade ao desenvolvimento de soluções de *software* em ambientes industriais. Essa integração não apenas aprimora a eficiência operacional, mas também contribui para a redução de desperdícios e para a consolidação de processos mais robustos e confiáveis. Além disso, a metodologia desenvolvida neste trabalho pode servir como base para projetos futuros em outros contextos industriais, demonstrando a aplicabilidade e a flexibilidade da abordagem adotada.

Por fim, este trabalho contribui para o campo da Engenharia de Controle e Automação ao combinar conceitos de desenvolvimento de *software*, ferramentas de qualidade e práticas de gestão operacional. Acredita-se que os conhecimentos adquiridos e as soluções propostas não apenas beneficiam o contexto específico da fabricação de transformadores, mas também fornecem um modelo replicável para a melhoria de processos em diversos setores industriais.

REFERÊNCIAS

- BARBER, K. Suzanne; GRASER, Thomas; HOLT, Jim. **Providing early feedback in the development cycle through automated application of model checking to software architectures**. Automated Software Engineering, 2001 (ASE 2001). Proceedings. 16th Annual International Conference. IEEE, 2001. p. 341-350. Disponível em: <https://ieeexplore.ieee.org/document/989825>. Acesso em: 02 dez. 2024.
- BASILI, Victor R.; WEISS, David M. **A methodology for collecting valid software engineering data**. IEEE Transactions on Software Engineering, v. 10, n. 6, p. 728-738, nov. 1984. Disponível em: <https://ieeexplore.ieee.org/document/5010301>. Acesso em: 02 dez. 2024.
- BAYERS, P. C. **Using Poka Yoke (mistake proofing devices) to ensure quality**. APPLIED POWER ELECTRONICS CONFERENCE AND EXPOSITION, 1994. Conference Proceedings, APEC '94. Ninth. IEEE, 1994. p. 201-204. Disponível em: <https://ieeexplore.ieee.org/document/316399>. Acesso em: 02 dez. 2024.
- BIESSMANN, Felix; GOLEBIEWSKI, Jacek; RUKAT, Tammo; LANGE, Dustin; SCHMIDT, Philipp. **Automated Data Validation in Machine Learning Systems**. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2015. Disponível em: <https://www.amazon.science/publications/automated-data-validation-in-machine-learning-systems>. Acesso em: 02 dez. 2024.
- BRIAND, Lionel C.; BASILI, Victor R.; THOMAS, William M. **A pattern recognition approach for software engineering data analysis**. IEEE Transactions on Software Engineering, v. 18, n. 11, p. 931-942, nov. 1992. Disponível em: <https://ieeexplore.ieee.org/document/177363>. Acesso em: 02 dez. 2024.
- CARDELL-OLIVIER, Rachel; Zhang, Lu; Barady, Ricky; Lim, You Hai; Naveed, Asad; Woodings, Terry. **Automated feedback for quality assurance in software engineering education**. 21st Australian Software Engineering Conference (ASWEC), 2010. Proceedings. IEEE, 2010, p. 157-164. Disponível em: <https://ieeexplore.ieee.org/document/5475060>. Acesso em: 02 dez. 2024.
- FENVES, Gregory L. **Object-oriented programming for engineering software development**. *Engineering with Computers*, v. 6, p. 1-15, 1990.
- FOWLER, Martin; BECK, Kent; BRANT, John; OPDYKE, William; ROBERTS, Don. **Refactoring: Improving the Design of Existing Code**. Boston: Addison-Wesley, 1999.
- HALLIDAY, David; RESNICK, Robert; WALKER, Jearl. **Fundamentals of Physics**. 9. ed. Extended. Hoboken: John Wiley & Sons, 2011.
- KHAN, Muhammad Naeem Ahmed; KHALID, Muhammad; HAQ, Sami ul. Review of Requirements Management Issues in Software Development. **I.J. Modern Education and Computer Science**, v. 1, p. 21-27, 2013. DOI: 10.5815/ijmecs.2013.01.03. Acesso em: 02 dez. 2024.

MARTIN, Robert C. **Clean Code: A Handbook of Agile Software Craftsmanship**. Boston: Pearson Education, 2008.

MORAES, Jacqueline Rodrigues; DAL FORNO, Ana Julia. **Aplicações do método Poka Yoke na indústria têxtil: uma revisão da literatura**. *X Simpósio de Engenharia de Produção (SIMPEP)*, Rio de Janeiro, Brasil, 2022.

PATERO, Judelyn L. **Electromagnetic induction: unraveling Faraday's laws and their applications in generators and transformers**. *International Journal of Advanced Research in Science, Communication and Technology*, v. 3, n. 1, p. 814-817, 2023. DOI: 10.48175/IJARSCT-12376.

SHAKIROV, M. **The electrotonic state in a power transformer**. *2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, IEEE, 2020.

SHEIKH, Waseem; SHEIKH, Nadeem. A Model-View-ViewModel (MVVM) Application Framework for Hearing Impairment Diagnosis - Design and Features. **2020 Intermountain Engineering, Technology and Computing (IETC). Proceedings**. IEEE, 2020, p. 1-10. Disponível em: <https://ieeexplore.ieee.org/document/9249124>. Acesso em: 02 dez. 2024.

SARPIRI, Mona Najafi; GANDOMANI, Taghi Javdani. A case study of using the hybrid model of Scrum and Six Sigma in software development. **International Journal of Electrical and Computer Engineering (IJECE)**, v. 11, n. 6, p. 5342-5350, dez. 2021. Disponível em: <https://www.researchgate.net/publication/352559529>. Acesso em: 02 dez. 2024.

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

TANTITHAMTHAVORN, Chakkrit; MCINTOSH, Shane; HASSAN, Ahmed E.; MATSUMOTO, Kenichi. An Empirical Comparison of Model Validation Techniques for Defect Prediction Models. **IEEE Transactions on Software Engineering**, 2016. Disponível em: <https://ieeexplore.ieee.org/document/7497471>. Acesso em: 02 dez. 2024.

TONINI, Antonio Carlos; SPÍNOLA, Mauro de Mesquita; LAURINDO, Fernando José Barbin. Six Sigma and software development process: DMAIC improvements. **PICMET 2006 Proceedings, 9-13 July, Istanbul, Turkey**. Portland International Conference on Management of Engineering and Technology, 2006, p. 2815-2823. Disponível em: <https://ieeexplore.ieee.org/document/4077697>. Acesso em: 02 dez. 2024.

LWAKATARE, Lucy Ellen; RANGED, Ellinor; CRNKOVIC, Ivica; BOSCH, Jan. On the experiences of adopting automated data validation in an industrial machine learning project. **2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)**. IEEE, 2021, p. 248-257. Disponível em: <https://ieeexplore.ieee.org/document/9402048>. Acesso em: 02 dez. 2024.

REDZIC, Cvetan; BAIK, Jongmoon. Six Sigma Approach in Software Quality Improvement. **Fourth International Conference on Software Engineering Research, Management and Applications (SERA'06)**. IEEE, 2006. p. 373-379. Disponível em:

https://www.researchgate.net/publication/221541895_Six_Sigma_Approach_in_Software_Quality_Improvement. Acesso em: 02 dez. 2024.

ZHOU, Minlin; GREENWELL, Richard; TANNOCK, James. **Object-oriented methods for manufacturing information systems**. *Computer Integrated Manufacturing Systems*, v. 7, n. 2, p. 113-121, 1994.

ZIMMERMANN, Guilherme dos Santos; BOHOMOL, Elena. Lean Six Sigma methodology to improve the discharge process in a Brazilian intensive care unit. **Revista Brasileira de Enfermagem**, v. 76, n. 3, p. e20220538, 2023. Disponível em: <https://www.researchgate.net/publication/372259080>. Acesso em: 02 dez. 2024.