

UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE ENGENHARIA MECÂNICA  
BACHARELADO EM ENGENHARIA MECATRÔNICA

THALITA ALVES DE SOUSA

**DESENVOLVIMENTO DE UM PROTÓTIPO DE SISTEMA PARA  
GESTÃO DE REQUISITOS DE DOCUMENTAÇÃO EM PROJETOS DE  
ENGENHARIA**

UBERLÂNDIA, MG

2023

THALITA ALVES DE SOUSA

**DESENVOLVIMENTO DE UM PROTÓTIPO DE SISTEMA PARA GESTÃO DE  
REQUISITOS DE DOCUMENTAÇÃO EM PROJETOS DE ENGENHARIA**

Trabalho de Conclusão de Curso apresentado à Faculdade de Engenharia Mecânica da Universidade Federal de Uberlândia como requisito parcial para obtenção do título de bacharel em Engenharia Mecatrônica.

**Orientador:** Prof. Dr. Eustáquio São José de Faria

UBERLÂNDIA, MG

2023

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU  
com dados informados pelo(a) próprio(a) autor(a).

S725 2023	<p>Sousa, Thalita Alves de, 1997- DESENVOLVIMENTO DE UM PROTÓTIPO DE SISTEMA PARA GESTÃO DE REQUISITOS DE DOCUMENTAÇÃO EM PROJETOS DE ENGENHARIA [recurso eletrônico] : Sistema de Gestão de Requisitos / Thalita Alves de Sousa. - 2023.</p> <p>Orientador: Eustáquio São José de Faria. Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Uberlândia, Graduação em Engenharia Mecatrônica. Modo de acesso: Internet. Inclui bibliografia. Inclui ilustrações.</p> <p>1. Mecatrônica. I. Faria, Eustáquio São José de, 1973- , (Orient.). II. Universidade Federal de Uberlândia. Graduação em Engenharia Mecatrônica. III. Título.</p> <p>CDU: 621.03</p>
--------------	--

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:  
Gizele Cristine Nunes do Couto - CRB6/2091  
Nelson Marcos Ferreira - CRB6/3074

Dedico este trabalho à minha família: Alliny,  
João e Lisa.

## AGRADECIMENTOS

Agradeço a mim mesmo por acreditar em mim, por fazer todo o trabalho duro, por não ter dias de folga, por nunca desistir, por sempre tentar dar mais do que receber e por tentar fazer mais o certo do que o errado, e mais do que tudo isso quero agradecer a mim por ter sido eu mesmo em todos os momentos.

Gostaria de agradecer à minha família que são as pessoas mais importantes do mundo! Mãe, obrigada por sempre me incentivar e por ser fonte de inspiração diária. Pai, obrigada por ter me ensinado a ser forte e independente. Lisa, obrigada por ser a melhor irmã e amiga que eu poderia ter. Amo vocês.

Aos meus amigos de Uberlândia os quais me acolheram e foram responsáveis por tornar essa jornada muito divertida, mesmo nos momentos mais difíceis. E agradecimentos especiais ao Luís Gustavo (DIGI) que mesmo à 1500 km de distância sempre esteve presente.

Meus agradecimentos ao corpo docente da Faculdade de Engenharia Mecânica (FEMEC), pelo ensino, dedicação e paciência, especialmente aos professores que me marcaram com suas disciplinas: Pedro Augusto, Louriel, Lucas Nascimento, Priscila, Rogério e Ana Marta.

Ademais, obrigada ao meu professor orientador, Eustáquio São José de Faria, pela orientação, conselhos e palavras de conforto.

Por fim, gostaria de agradecer a equipe de Automação e Instrumentação da Siemens Energy, a qual foi parte indispensável para a realização deste trabalho, sem a ajuda de vocês ele nunca poderia ser concluído. Agradecimentos especiais aos meus gestores: Roman e Jessica, ao colega Guilherme e mentor Gabriel.

*“Cinderella man  
Doing what you can  
They can't understand what it means*

*Cinderella man  
Hang on to your plans  
Try as they might  
They cannot steal your dreams”*

*— Cinderellaman, Rush.*

DE SOUSA, T. A. Desenvolvimento de um protótipo de sistema para gestão de requisitos de documentação em projetos de engenharia. 2023. 48p. Monografia de Conclusão de Curso (Bacharel) – Graduação em Engenharia Mecatrônica, Universidade Federal de Uberlândia, Uberlândia - MG, Brasil, 2023.

## RESUMO

Com a pandemia de COVID-19 e o aumento do trabalho remoto, a gestão de projetos de engenharia tem se adaptado a novas ferramentas e métodos de automatização, para garantir a eficiência e a produtividade. A documentação é um elemento essencial desse processo, fornecendo registros detalhados do projeto e auxiliando na comunicação e conformidade regulamentar. Para criar documentação precisa e eficaz, é necessário ter requisitos de projeto precisos e completos, que servem como base para projeto, teste e verificação e garantem que o produto atenda às necessidades do cliente. A busca por eficiência, redução de custos e aumento de produtividade tem impulsionado o desenvolvimento de sistemas de automatização, como as plataformas "*Low-Code*". Um protótipo de sistema (SG\_Req) foi desenvolvido para automatizar a seleção e visualização de requisitos de documentos de engenharia na área de automação e instrumentação, com base em dados públicos disponíveis no Catálogo de padronização da PETROBRAS. O SG\_Req foi desenvolvido na plataforma *Low-Code* Mendix, com o objetivo de facilitar o acesso e a validação das informações, relacionando cada documento com seus respectivos requisitos, permitindo a validação da arquitetura e funcionalidades e possibilitando a visualização, teste e feedback do produto desenvolvido com foco nessa etapa.

DE SOUSA, T. A. Development of a Prototype System for Management of Documentation Requirements in Engineering Projects. 2023. 48p. Course Conclusion Monograph (Bachelor) – Mechatronics Engineer Graduation, Federal University of Uberlândia, Uberlândia - MG, Brazil, 2023.

## **ABSTRACT**

Due to COVID-19 pandemic and the increase in remote work, engineering project management has been adapting to new automation tools and methods to ensure efficiency and productivity. Documentation is an essential element of this process, providing detailed records of the project and assisting in communication and regulatory compliance. To create accurate and effective documentation, precise and complete project requirements are necessary, serving as the basis for design, testing, and verification, and ensuring that the product meets customer needs. The search for efficiency, cost reduction, and increased productivity has driven the development of automation systems, such as "Low-Code" platforms. A system prototype (SG\_Req) was developed to automate the selection and visualization of engineering document requirements in the automation and instrumentation area, based on publicly available data in the PETROBRAS Standardization Catalog. The SG\_Req was developed on the Mendix Low-Code platform with the aim of facilitating access and validation of information, relating each document to its respective requirements, enabling validation of architecture and functionalities, and allowing visualization, testing, and feedback of the product developed with a focus on this stage."

---

Keywords: Project Management, Technical Documentation, Project Requirements, Low-Code, Engineering.



## LISTA DE ABREVIATURAS

DEV – *Development* (Desenvolvimento)

FEL – Front-End-Loading

SG\_Req – Sistema para Gestão de Requisitos de Documentação em Projetos de Engenharia

VIPs – *Value Improving Practices*

VPL – Valor Presente Líquido.

## LISTA DE FIGURAS

Figura 1 - Etapas da Metodologia FEL para Projetos Offshore .....	17
Figura 2 - Ciclo da Metodologia Ágil.....	20
Figura 3 - Modelo de questionário utilizado para as entrevistas com usuários.....	27
Figura 4 - Página Inicial.....	31
Figura 5 - Página de Overview da Lista de Documentos.....	32
Figura 6 - Página de Visualização de Documento.....	32
Figura 7 - Modelo de dados do SG_Req.....	35
Figura 8 - Modelo de dados para relação entre as entidades.....	36
Figura 9 - Página Home .....	37
Figura 10 - Página Documentos e suas funcionalidades.....	38
Figura 11 - Função Search.....	38
Figura 12 - Função New (Novo).....	39
Figura 13 - Função Edit (Editar).....	39
Figura 14 - Função Delete, seleção de documento.....	40
Figura 15 - Função Delete, confirmação de exclusão de documento selecionado .....	40
Figura 16 - Página Visualização .....	41
Figura 17 - Função filtrar por tipo de projeto.....	41
Figura 18 - Página de exibição de documento específico .....	42
Figura 19 - Página Importar Dados.....	42
Figura 20 - Template utilizado para aquisição de dados via tabela Excel .....	43
Figura 21 - Função de importação de tabelas.....	43
Figura 22 - Página Gestão. ....	44
Figura 23 - Página de Login.....	44

## SUMÁRIO

1. INTRODUÇÃO.....	13
1.1 JUSTIFICATIVA.....	14
1.2 OBJETIVO GERAL.....	14
1.3 OBJETIVOS ESPECÍFICOS .....	15
2. REVISÃO BIBLIOGRÁFICA.....	16
2.1 GERENCIAMENTO DE PROJETOS.....	16
2.2 METODOLOGIA <i>FRONT-END LOADING</i> (FEL) .....	17
2.2.1 FEL 1 - Identificação de oportunidade ( <i>Opportunity identification</i> ).....	18
2.2.2 FEL 2 - Engenharia Conceitual ( <i>Conceptual Engineering</i> ).....	18
2.2.3 FEL 3 Engenharia Básica ( <i>Basic Engineering</i> ).....	18
2.3 ENGENHARIA DE SOFTWARE.....	18
2.4 MÉTODO ÁGIL: SCRUM.....	19
2.5 REQUISITOS .....	21
2.5.1 REQUISITOS DE DOCUMENTAÇÃO EM PROJETOS DE ENGENHARIA.....	22
2.5.2 REQUISITOS DE SOFTWARE.....	23
2.5.2.1 REQUISITOS FUNCIONAIS.....	23
2.5.2.2 <i>USER STORIES</i> (HISTÓRIAS DE USUÁRIOS) .....	23
2.5.2.3 REQUISITOS NÃO-FUNCIONAIS .....	24
2.6 BANCO DE DADOS RELACIONAL .....	24
2.7 DESENVOLVIMENTO DE SOFTWARE EM PLATAFORMA LOW-CODE.....	24
3. SISTEMA PARA GESTÃO DE REQUISITOS DE DOCUMENTAÇÃO EM PROJETOS DE ENGENHARIA (SG_Req).....	26
3.1 MOTIVAÇÃO .....	26
3.2 REQUISITOS DE SOFTWARE .....	27
3.3 SPRINTS .....	28
3.4 WIREFRAME.....	31
3.5 REQUISITOS DE DOCUMENTAÇÃO.....	33
3.6 MODELO DE DADOS.....	34
4. O SG_REQ.....	37
4.1 PÁGINA HOME.....	37
4.2 PÁGINA DOCUMENTOS .....	38
4.2.1 SEARCH.....	38
4.2.2 NEW .....	39
4.2.3 EDIT .....	39

4.2.4	DELETE.....	40
4.3	PÁGINA VISUALIZAÇÃO.....	41
4.4	PÁGINA IMPORTAR DADOS.....	42
4.5	PÁGINA GESTÃO.....	44
4.6	PÁGINA DE LOGIN.....	44
5.	CONCLUSÕES E TRABALHOS FUTUROS.....	45
5.1	TRABALHOS FUTUROS.....	46
	REFERÊNCIAS BIBLIOGRÁFICAS.....	48

## 1. INTRODUÇÃO

Os métodos e ferramentas que auxiliam na gestão de projetos de engenharia vêm evoluindo e se adaptando a um novo modo de trabalho institucionalizado e disseminado, conhecido como home-office, que tem sido amplamente implantado principalmente em consequência da pandemia de COVID-19. Em meados de 2020 a modalidade remota de trabalho, aproximou todos os gestores e executores de projetos dos ambientes virtuais e das mais diversas tecnologias de gestão. Em função disso, houve um crescimento da demanda pela automatização de atividades, culminando na necessidade de sistemas para auxiliar na realização destas.

Na área de projetos em engenharia a documentação é parte integrante do processo, fornecendo registros detalhados do projeto, desenvolvimento e teste de um produto ou sistema. Ela é usada para comunicação dentro e fora da equipe de engenharia, bem como para conformidade regulamentar e garantia de qualidade (Watts, 2011). Tendo em vista a importância da elaboração de documentos dentro do escopo dos projetos, é imprescindível que as etapas de execução sejam bem definidas e padronizadas. Como efeito, essa atividade tem sido objeto da aplicação de ferramentas de gestão e do desenvolvimento de sistemas que visam garantir a consistência, qualidade e armazenamento da informação.

Para a concepção de documentação eficaz, precisa e atualizada, faz-se necessário que seja feita previamente a mobilização de informações, documentos e normas que fornecem dados de entrada, denominada como requisitos de projeto. Requisitos precisos e completos são essenciais para o sucesso de qualquer projeto de engenharia. Eles fornecem a base para projeto, teste e verificação e ajudam a garantir que o produto atenda às necessidades e expectativas do cliente (Watts, 2011). Adicionalmente, a gestão dos requisitos de projeto pode ser realizada através de ferramentas automatizadas.

O desenvolvimento de sistemas para automatizar atividades tem sido uma grande força motriz por trás do rápido crescimento tecnológico. Os sistemas de automação podem melhorar a eficiência, reduzir custos e aumentar a produtividade, tornando-os um componente essencial da indústria e do comércio modernos (Love, 2007). Portanto, com o objetivo de facilitar a execução do processo de compilação dos requisitos dos documentos de engenharia no domínio da automação e instrumentação, foi proposta a concepção de um protótipo de sistema que fará a automação da seleção e visualização dos requisitos de documentos baseado nos

dados disponíveis no Catálogo de padronização da PETROBRAS ([canalforneecedor.petrobras.com.br/pt/regras-de-contratacao/catalogo-de-padronizacao/](http://canalforneecedor.petrobras.com.br/pt/regras-de-contratacao/catalogo-de-padronizacao/)).

Atualmente, com a criação de plataformas para a automação de tarefas, tais como: Power Automate, Zapier e Integromat, o ramo de desenvolvimento de softwares tem se aproximando do dia a dia das equipes projetistas. Percebendo essa oportunidade, as empresas começam a fazer o uso das plataformas de desenvolvimento “Low-Code”, que são uma ferramenta útil para organizações que precisam criar aplicativos personalizados de gerenciamento de projetos rapidamente e sem a necessidade de equipes com amplo conhecimento em programação. Por esse motivo, a plataforma Mendix foi selecionada para a elaboração deste trabalho.

### **1.1 JUSTIFICATIVA**

Dado o volume de projetos que são executados em grandes empresas e a multidisciplinaridade de escopos, percebeu-se a necessidade de elaborar ferramentas que auxiliassem na gestão e armazenamento do conhecimento e na padronização dos processos de execução. Essa necessidade é explicitada quando há rotatividade de pessoal (seja por férias, demissões, afastamentos etc.), pois a lacuna aberta por essa ausência faz que o conhecimento gerado e difundido, muitas vezes transmitido no boca-a-boca ou de outras maneiras informais e não catalogadas, seja perdido. Portanto, é de interesse de todos que a informação gerada pelas equipes seja armazenada, organizada e disponibilizada em algum sistema.

Para dar início à classificação, organização e armazenamento do conhecimento, decidiu-se que primeiramente seria desenvolvido um sistema que ajudasse a lidar com a gestão do conhecimento nas primeiras etapas de projeto. Assim, antes de começar a execução própria dos projetos, tem-se a etapa de engenharia de requisitos onde são feitas pesquisas em busca de informações, meios e dados que possibilitarão o início da elaboração. Sendo assim, o desenvolvimento de um sistema de gestão de requisitos a fim de esquematizar, centralizar e facilitar o acesso à informação se faz necessário.

### **1.2 OBJETIVO GERAL**

O objetivo geral deste projeto de graduação é desenvolver um protótipo de sistema web que seja capaz de gerir as informações de entrada (requisitos) dos documentos técnicos de projetos de engenharia. O software deverá relacionar cada documento com seus respectivos requisitos e prover diferentes possibilidades de filtros e relações entre os documentos e requisitos e

também informará sobre o conteúdo e objetivo da documentação. A implementação do sistema focou nas disciplinas de automação de instrumentação, porém, é facilmente adaptável para auxiliar outras disciplinas, tais como: elétrica e mecânica.

### **1.3 OBJETIVOS ESPECÍFICOS**

1. Mapeamento de requisitos técnicos de documentação das disciplinas de Automação e Instrumentação;
2. Mapeamento de documentos do escopo das disciplinas de Automação e Instrumentação;
3. Estudo da plataforma *Low-Code* (Mendix);
4. Estudo de Metodologias ágeis;
5. Desenvolvimento de estrutura de dados para armazenamento de informações em base de dados relacional;
6. Desenvolvimento de interface para cadastro de informações na base de dados supracitada;
7. Construção do protótipo de software para gerenciamento de requisitos.

## **2. REVISÃO BIBLIOGRÁFICA**

Neste capítulo serão apresentados os principais conceitos teóricos que fundamentam este trabalho.

### **2.1 GERENCIAMENTO DE PROJETOS**

Para a execução de projetos de engenharia, é importante que sejam adotados regras e padrões com o propósito de garantir a execução do fluxo de atividades. Por esse motivo, é imperativo o estudo e aplicação de metodologias de gestão de projetos. De acordo com PMBOK (2017), define-se gestão como: a disciplina de planejamento, organização, motivação e controle de recursos para atingir objetivos específicos.

No contexto de projetos, o gerenciamento envolve identificar e definir as metas e objetivos do projeto, desenvolver um plano para atingir essas metas e, então, coordenar e controlar as várias atividades e recursos necessários para concluí-lo. Isso pode incluir tarefas como, por exemplo, identificar e priorizar as entregas e estabelecer cronogramas e orçamentos.

O gerenciamento também envolve motivar e inspirar membros da equipe a trabalharem juntos de maneira eficaz (com o intuito de atingir metas) e comunicação com as partes interessadas para garantir que o projeto esteja alinhado com suas necessidades e expectativas. No geral, o objetivo do gerenciamento é garantir que um projeto seja concluído com sucesso e eficiência e que atenda ou exceda às expectativas de todas as partes interessadas.

A aplicação do gerenciamento se dá por meio de metodologias, as quais possuem etapas bem definidas que se adaptam de acordo com a necessidade e tipo de projeto. Genericamente, as metodologias mais comumente utilizadas segundo PMBOK são: Waterfall, Agile, SCRUM, Lean e Six Sigma.

De acordo com o PMBOK (2017), um projeto é definido como "um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo". Essa definição destaca a natureza temporária dos projetos, bem como o fato de serem realizados para alcançar um resultado ou entrega específica. Sendo assim, ao iniciar um projeto, deve-se ser analisado qual é a melhor metodologia para sua elaboração e bom funcionamento. Vale destacar que podem ser usadas metodologias concomitantes, ou seja, ao dividir o projeto em várias etapas, pode-se aplicar uma metodologia para o escopo geral e outra para fases específicas da sua elaboração.



Como neste trabalho serão abordados projetos de automação e instrumentação baseados nas normas da companhia PETROBRAS, a qual atua na indústria de petróleo e gás, a metodologia de gerenciamento de projetos aplicada ao sistema e associada às etapas de elaboração de documentação é a FEL – *FRONT-END LOADING*. Isso se deve porque FEL (*FRONT-END LOADING*) é a metodologia mais difundida neste ramo, pois maneja projetos complexos focando no planejamento, design e definições de escopo, objetivo e metas nas fases iniciais de projeto.

## 2.2 METODOLOGIA *FRONT-END LOADING* (FEL)

A FEL é uma metodologia comumente utilizada na indústria para planejar e gerenciar projetos grandes e complexos, como os dos campos de petróleo e gás ou projetos de infraestrutura. Tem como objetivo otimizar o desempenho da execução do projeto e minimizar os custos, garantindo que todas as informações e preparações necessárias estejam disponíveis antes do início da execução (Saputelli, 2013).

A FEL é estruturada em uma série de três fases, cada qual focando em aspectos diferentes do planejamento do projeto e da tomada de decisões. Segundo Basilio (2017) a

Figura 1 mostra as fases típicas de um projeto de desenvolvimento de campo offshore<sup>1</sup> completo.

Figura 1 - Etapas da Metodologia FEL para Projetos *Offshore*



Fonte: Adaptado de BARBOSA et al., 2013.

<sup>1</sup> Offshore: ambiente marinho e zona de transição terra-mar ou área localizada no mar.

### **2.2.1 FEL 1 - Identificação de oportunidade (*Opportunity identification*)**

Durante esta fase, uma avaliação de negócios é realizada para determinar se o projeto está alinhado com o plano de negócios geral da empresa e com as oportunidades de mercado em potencial. Além disso, os principais indicadores de viabilidade, como taxa interna de retorno, valor presente líquido (VPL) e *payback* descontado, são calculados nessa etapa.

### **2.2.2 FEL 2 - Engenharia Conceitual (*Conceptual Engineering*)**

Nesta etapa, os indicadores da fase anterior são avaliados e selecionados por meio do desenvolvimento da engenharia conceitual de cada opção. Isso inclui análises de soluções tecnológicas e construtivas, o desenvolvimento de especificações conceituais e estimativas de custo dentro de uma faixa de variação e a aplicação de práticas de melhoria de valor (VIPs – *Value Improving Practices*) como um guia para a próxima fase. É possível que mais de uma alternativa conceitual seja recomendada como *input* da próxima fase, principalmente para projetos de alta complexidade tecnológica.

### **2.2.3 FEL 3 Engenharia Básica (*Basic Engineering*)**

Durante esta fase de engenharia básica, o foco está na preparação do projeto para aprovação e implementação corporativa, realizando o desenvolvimento de documentação técnica detalhada de engenharia das opções de projeto básico selecionadas na fase dois.

Enquanto o planejamento geral do projeto é feito pelo método FEL, a documentação de engenharia pode ser desenvolvida por outros métodos, como a metodologia Scrum, que envolve a quebra do projeto em atividades bem definidas e alocação de recursos da equipe adequadamente. Essa abordagem permite uma melhor gestão do escopo e comunicação com o cliente, além de permitir uma evolução constante do projeto à medida que cada ciclo de trabalho é concluído e pronto para validação do cliente. Isso é particularmente importante para projetos de engenharia que envolvem propostas de novas tecnologias, que podem estar sujeitas a alterações de escopo.

## **2.3 ENGENHARIA DE SOFTWARE**

A Engenharia de Software é uma disciplina dentro da Computação que se concentra no uso de abordagens estruturadas, sistemáticas e mensuráveis para criar, operar, manter e melhorar o software. Envolve a aplicação de princípios de engenharia ao desenvolvimento de software

para construir sistemas confiáveis, eficientes e de fácil manutenção. Algumas atividades importantes na Engenharia de Software incluem coleta de requisitos, design, implementação, teste, implantação e manutenção (Valente, 2020).

O sistema de gestão de requisitos desenvolvido neste projeto busca usar as melhores práticas de engenharia software disseminadas pela comunidade Mendix, por meio dos seus fóruns de discussão, documentação e cursos. Além dos recursos disponibilizados pela plataforma Mendix, utilizou-se as práticas estabelecidas pela metodologia ágil.

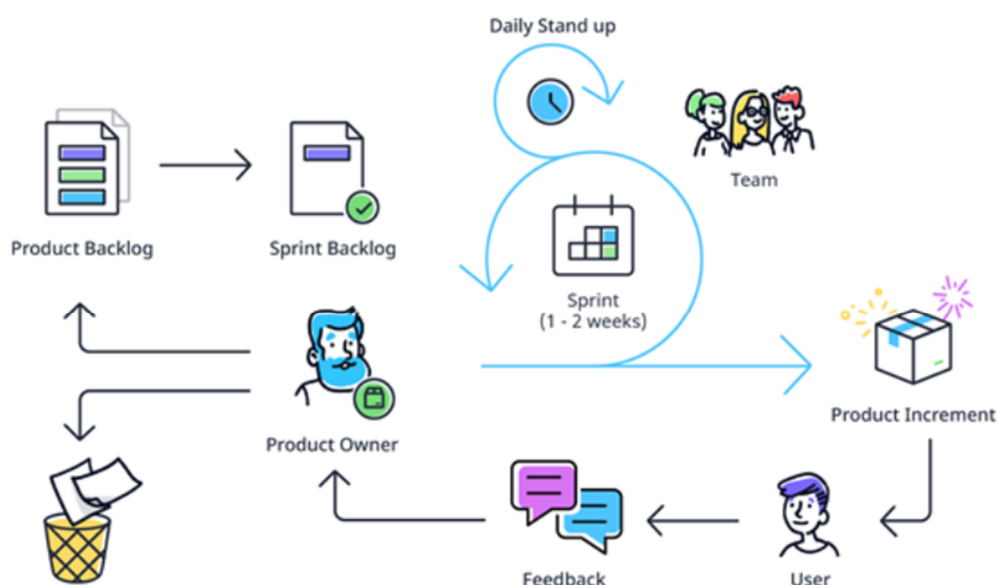
## **2.4 MÉTODO ÁGIL: SCRUM**

Os métodos ágeis são um conjunto de princípios para o desenvolvimento de software em que os requisitos e soluções evoluem por meio do esforço colaborativo de equipes auto-organizadas e multifuncionais e seus clientes. Os métodos ágeis são uma alternativa às abordagens mais tradicionais e orientadas a planos, pois, em vez de seguir um plano rígido e inicial, as equipes ágeis são flexíveis e receptivas às mudanças, e entregam software funcional de forma incremental ao longo do projeto. Isso permite que eles respondam aos novos requisitos e às mudanças nas necessidades dos clientes de maneira mais rápida e eficaz (Valente, 2020).

Um processo de desenvolvimento de software define o conjunto de passos, tarefas, eventos e práticas a serem seguidos por desenvolvedores de software na produção de sistemas computacionais. Proposto por Jeffrey Sutherland e Ken Schwaber (1995), Scrum é um método ágil de gerenciamento de projetos iterativo e incremental. Entre os métodos ágeis, este é o mais conhecido e amplamente utilizado. Esse fato se deve à indústria que cresceu em torno de sua adoção, incluindo a produção de livros, cursos diversos, consultorias e certificações. (Valente, 2020).

O Scrum baseia-se na ideia de desenvolvimento iterativo e incremental, no qual um projeto é desenvolvido em uma série de ciclos curtos, chamados "sprints", durante os quais um conjunto específico de trabalho é concluído e demonstrado ao cliente, como ilustrado na Figura 2.

Figura 2 - Ciclo da Metodologia Ágil



Fonte: [academy.mendix.com/link/modules/311/lectures/2342/2.3-Agile-Project-Methodology](https://academy.mendix.com/link/modules/311/lectures/2342/2.3-Agile-Project-Methodology)

Para o desenvolvimento do sistema de gestão de requisitos, objeto do presente trabalho, utilizou-se as etapas do processo de desenvolvimento desta metodologia, que são as seguintes:

1. Definição de objetivos do projeto e elaboração do roteiro de produto para orientar o trabalho;
2. Formação de equipe multifuncional com habilidades necessárias para execução do projeto.
3. Divisão do projeto em pequenas partes gerenciáveis chamadas "Histórias de Usuários / User Stories";
4. Definição de prioridades das User Stories, chamado de "backlog do produto";
5. Planejamento de sprints selecionando um conjunto de histórias de usuários no topo do backlog do produto;
6. Execução de sprints:
  - a. Durante o sprint, a equipe trabalha para concluir as histórias de usuário selecionadas;

- b. No final do sprint, a equipe demonstra o trabalho concluído para o cliente e obtém feedback;
  - c. A equipe revisa e reflete sobre o sprint, identificando as lições aprendidas e as áreas de melhoria.
7. O processo é repetido, com a equipe se adaptando e melhorando continuamente com base no feedback do cliente.
8. Para execução das etapas listadas anteriormente, segundo a metodologia, alguns papéis foram definidos, dentre eles:

**Product Owner** - responsável por representar os interesses das partes interessadas e definir os recursos e requisitos do produto. Eles são responsáveis por manter o backlog do produto e priorizar o trabalho a ser feito.

**Scrum Master**- responsável por facilitar o processo Scrum e garantir que a equipe esteja seguindo o framework Scrum. Eles ajudam a equipe a manter o foco e remover quaisquer obstáculos que possam estar bloqueando o progresso.

**Dev Team**- equipe multifuncional de indivíduos responsáveis por concluir o trabalho definido no sprint. O Dev team (Equipe de Desenvolvimento) é auto-organizado e tem autonomia para determinar a melhor forma de concluir o trabalho.

Além dessas funções, também pode haver partes interessadas, como clientes ou usuários, que estão envolvidos no processo Scrum. Eles fornecem informações e feedback ao *Product Owner* e participam de revisões e demonstrações do trabalho concluído.

## 2.5 REQUISITOS

O papel dos requisitos no gerenciamento de projetos há muito é reconhecido como crucial e, à medida que o mundo dos negócios se torna cada vez mais complexo, as organizações estão começando a entender o valor de usar a análise de negócios para definir, gerenciar e controlar os requisitos para obtenção de vantagem competitiva. O processo de gerenciamento de requisitos pode começar antes mesmo de um projeto ser iniciado e pode envolver uma avaliação de necessidades como parte do portfólio ou planejamento do programa. De acordo com o guia "Gerenciamento de requisitos: um guia prático", o processo de gerenciamento de requisitos começa com uma avaliação de necessidades que pode ocorrer em vários estágios do planejamento do projeto (PMBOK, 2017).

Para este projeto, é necessário fazer a distinção entre dois tipos de requisitos: os requisitos que serão usados na elaboração da documentação de engenharia e os requisitos que foram definidos para a elaboração do sistema computacional.

### **2.5.1 REQUISITOS DE DOCUMENTAÇÃO EM PROJETOS DE ENGENHARIA**

Os requisitos de projeto consistem em premissas ou conhecimentos que devem estar presentes em um produto, com a finalidade de cumprir especificações técnicas impostas formalmente. Os requisitos incluem as necessidades quantificadas e documentadas e as expectativas do cliente, que precisam ser obtidos, analisados e registrados com detalhes suficientes para serem incluídos na linha de base do escopo e medidos uma vez que a execução do projeto inicie (PMBOK, 2017).

A obtenção de Requisitos envolve as atividades de determinação, documentação e supervisão das necessidades das partes interessadas para atingir os objetivos. A principal vantagem desse processo é que ele estabelece as bases para definir e controlar o escopo do produto e do projeto.

Sendo assim, neste projeto, o produto é um documento de engenharia e seus requisitos são as de informações que devem ser coletadas para sua elaboração. Estas informações são encontradas nos mais diversos tipos de formatos, podendo ser tabelas, outros documentos, gráficos etc.

## 2.5.2 REQUISITOS DE SOFTWARE

Requisitos de software são a base do processo de desenvolvimento de sistemas computacionais. Eles definem o que o aplicativo deverá fazer, como se comportará e como será usado. Sem requisitos claros e concisos, o desenvolvimento de software pode se tornar caótico e imprevisível. Um bom software começa com bons requisitos, e é importante defini-los e documentá-los cuidadosamente para criar uma solução que atenda às necessidades do usuário (Sommerville, 2011).

No desenvolvimento de software é necessário ainda fazer a diferenciação entre dois tipos de requisitos: (1) Requisitos funcionais; e (2) Requisitos não funcionais.

### 2.5.2.1 REQUISITOS FUNCIONAIS

Os requisitos funcionais geralmente são o foco principal do processo de desenvolvimento de software, pois descrevem o que um sistema de software deve fazer e como deve se comportar, e especificam as funções que o sistema deve executar e os resultados que deve produzir.

Sommerville (2011) sugere que os requisitos funcionais devem ser especificados de maneira clara e concisa, e devem ser fáceis de entender tanto pelos interessados técnicos quanto pelos não técnicos. Ele recomenda o uso de técnicas como *user stories* e modelagem baseada em cenários para capturar requisitos funcionais com foco no usuário.

É importante definir e documentar cuidadosamente os requisitos funcionais para garantir que o sistema de software atenda às necessidades dos usuários finais e atinja os objetivos de negócios desejados.

### 2.5.2.2 USER STORIES (HISTÓRIAS DE USUÁRIOS)

As histórias de usuários são usadas em metodologias de desenvolvimento ágil, como Scrum e Kanban, para ajudar as equipes a entender o resultado que é esperado de um software e fornecer, de maneira clara e concisa, os requisitos aos desenvolvedores. Segundo Valente (2020), Histórias de Usuário são documentos simplificados de especificação de requisitos que, no desenvolvimento de software, consistem na descrição curta e simples de um recurso ou função que o usuário gostaria de ver implementado no sistema.

Os *User Stories* são obtidos por meio de entrevistas em que são coletadas informações sobre as necessidades e desejos de usuários potenciais ou existentes. Isso pode ser feito pessoalmente, por telefone ou por meio de pesquisas online. Após a realização das entrevistas, o conteúdo gerado é analisado para que a informação seja processada em atividades de implementação do software que serão designadas para o time de desenvolvimento.

### **2.5.2.3 REQUISITOS NÃO-FUNCIONAIS**

Requisitos não-funcionais descrevem o desempenho, segurança, usabilidade e outras características do sistema, mas não especificam funções ou comportamentos específicos. Estes requisitos são necessários porque tem impacto significativo na qualidade e no valor do software.

## **2.6 BANCO DE DADOS RELACIONAL**

De acordo Garcia-Molina (2009), um banco de dados relacional é uma coleção de dados organizados em tabelas e acessados usando linguagem de consulta estruturada (SQL). Em um banco de dados relacional, cada tabela representa uma relação ou um conjunto de dados relacionados. As tabelas são conectadas por relacionamentos que definem como os dados em uma tabela se relacionam com os dados em outra tabela. As relações são um conjunto de tuplas, ou linhas, que possuem os mesmos atributos, ou colunas. Cada atributo representa um determinado dado, como o nome de um cliente ou o custo total de um pedido. Os relacionamentos entre as tabelas são estabelecidos por meio do uso de chaves, que são atributos que identificam exclusivamente cada tupla dentro de uma tabela.

Neste projeto os dados estão estruturados de acordo com o princípio relacional, porém a linguagem SQL está embutida no modelo Low-Code definido pela plataforma Mendix. Com isso, tem-se as tabelas de dados e duas relações organizadas em gráficos e funções definidas visualmente na aba de funcionalidades denominada *Domain Model* (Modelo de Domínio).

## **2.7 DESENVOLVIMENTO DE SOFTWARE EM PLATAFORMA LOW-CODE**

O desenvolvimento de aplicativos Low-Code é uma tendência crescente na indústria de software, com muitos especialistas prevendo que se tornará a forma dominante de criar aplicativos em breve. As plataformas Low-Code oferecem vários benefícios, incluindo a capacidade de criar aplicativos mais rapidamente e com menos habilidades necessárias, graças ao seu alto nível de abstração e recursos de programação visual (Sahay et al., 2020).



Essas plataformas também costumam oferecer suporte à implantação automatizada em infraestruturas de nuvem, que podem ser confiáveis e escaláveis. Fornecedores especializados (como OutSystems, Mendix e Appian) e provedores de nuvem estabelecidos (como Microsoft) oferecem plataformas Low-Code, enquanto a pesquisa acadêmica sobre essas plataformas está aumentando. Alguns estudos se concentraram em plataformas específicas em detalhes, enquanto outros tentam fornecer uma visão mais ampla das opções disponíveis e seus recursos (Lichtenthäler et al., 2022).

A plataforma de desenvolvimento Low-Code utilizada neste trabalho é a plataforma Mendix, e foi escolhida pelo motivo da autora ter total acesso a todos os seus recursos.

A documentação oficial da plataforma Mendix fornece um guia abrangente para os seus recursos e funcionalidades, incluindo informações sobre como criar e implantar aplicativos personalizados, integrar-se a outros sistemas e ampliar os recursos da plataforma. O compilado de documentação, localizado em [docs.mendix.com](https://docs.mendix.com), é a principal fonte de referências utilizada durante todo o desenvolvimento do sistema de gestão de requisitos (apelidado de SG\_Req).

### **3. SISTEMA PARA GESTÃO DE REQUISITOS DE DOCUMENTAÇÃO EM PROJETOS DE ENGENHARIA (SG\_Req)**

O Sistema para Gestão de Requisitos de Documentação em Projetos de Engenharia é o protótipo de um sistema que visa garantir a gestão, centralização e padronização dos conhecimentos sobre desenvolvimento de documentação das áreas técnicas de instrumentação e automação. Como o processo de desenvolvimento de documentação é constituído de várias etapas, este protótipo é focado na etapa inicial: a coleta de requisitos.

#### **3.1 MOTIVAÇÃO**

A ideia de desenvolver um sistema para gestão de requisitos em projetos de engenharia nasceu de conversas com os gestores do programa de estágio, do qual a autora deste trabalho faz parte, sobre dificuldades encontradas para iniciar o processo de desenvolvimento de documentação técnica. Em razão da informação se encontrar descentralizada e espalhada por diferentes fontes de armazenamento, chegou-se à conclusão de que um sistema que unificasse a informação e fornecesse dicas iniciais de como encontrá-la seria o primeiro passo para digitalizar e automatizar a atividade de mapeamento de requisitos.

Atualmente, a etapa de mapeamento de requisitos é feita por meio de reuniões entre os membros que possuem as informações e a pessoa que necessita trabalhar em um determinado documento. Por exemplo, a pessoa A é a responsável por desenvolver o documento Lista de Entradas e Saídas, para obter os documentos que são requisitos da lista ela necessita contatar a pessoa B que possui a informação (seja esta outros documentos, planilhas com esquemas da informação ou links de sites de armazenamento).


Por esse motivo, o SG\_Req foi desenvolvido como um protótipo, possibilitando a visualização e validação das funcionalidades desenvolvidas até o momento, e permitindo uma análise preliminar do sistema no que diz respeito à relação entre documentos e requisitos. Esse estágio inicial do desenvolvimento servirá como base para a versão final do SG\_Req, onde serão realizados melhorias e refinamentos com base nas necessidades e requisitos identificados durante a etapa de prototipagem. Com base no feedback recebido durante essa etapa, serão incorporados melhorias e aprimoramentos nas próximas versões do sistema.

### 3.2 REQUISITOS DE SOFTWARE

Para o desenvolvimento do SG\_Req, primeiramente foram definidos grupos de usuários para obter uma visão melhor das necessidades e expectativas em relação ao escopo da aplicação. Os grupos definidos incorporaram os cargos que estão mais ligados à elaboração dos documentos, sendo eles: Coordenador Técnico, Líder Técnico e Executor de Documentação.

Foram selecionados 3 colaboradores de cada cargo para participarem de entrevistas individuais, onde foi apresentado o objetivo do sistema e os usuários eram incentivados a compartilhar seus pensamentos e ideias, seguindo o modelo de perguntas representado na Figura 3.

Figura 3 - Modelo de questionário utilizado para as entrevistas com usuários

USER STORIES - COORDENADOR TÉCNICO / LÍDER TÉCNICO / EXECUTOR 

Como <COORDENADOR TÉCNICO / LÍDER TÉCNICO / EXECUTOR >

Eu quero <O que eu gostaria que a ferramenta oferecesse>

De modo que <Como a ferramenta pode melhorar a eficiência da elaboração de documentação>

Após entrevistas, as respostas obtidas para a pergunta modelo foram esquematizadas, analisadas e posteriormente compuseram a lista de requisitos de software (requisitos funcionais) demonstrada abaixo.

1. Visualização de documento: O sistema deve conter páginas para cada documento listando seus requisitos e informações definidas pelo banco de dados.
2. Edição de informação: O sistema deve incluir um campo em que seja possível fazer a edição de informações cadastradas em cada documento.

3. Cadastro de novo documento: O sistema deve incluir um campo em que seja possível fazer o cadastro de novo documento e suas informações.
4. Listar por tipo: O sistema deve permitir que uma lista de documentos seja gerada de acordo com o tipo de projeto selecionado.

Para os requisitos não-funcionais, alguns requisitos genéricos, que costumam ser usados na maioria dos softwares, foram selecionados de acordo com as necessidades da aplicação. Segue a lista:

1. Usabilidade: O sistema deve oferecer interface intuitiva e clareza nas informações apresentadas;
2. Segurança: O sistema deve oferecer proteção dos dados e informações;
3. Escalabilidade: O sistema deve ter capacidade de expansão, para lidar com o crescimento futuro da demanda;
4. Portabilidade: O sistema deve ter capacidade de ser executado em diferentes plataformas, sistemas operacionais, navegadores ou dispositivos, permitindo uma fácil adaptação em diferentes ambientes.
5. Legalidade: O sistema deve cumprir leis, regulamentações ou padrões específicos, como proteção de dados pessoais, direitos autorais e conformidade com normas da indústria.

### 3.3 SPRINTS

No contexto do SG\_REQ, as *sprints* são atividades definidas para o desenvolvimento e entrega de funcionalidades específicas do sistema. Para garantir a funcionalidade das *sprints* foi definido um *time Scrum* com três papéis: Product Owner, em que o responsável foi o gestor da área de automação e instrumentação, Scrum Master, em que o responsável foi um dos engenheiros da área e Dev Team, em que o responsável foi o autor do projeto.

Durante cada *sprint*, a equipe de desenvolvimento se concentrou em implementar as funcionalidades planejadas, realizar testes e garantir a qualidade do sistema. O escopo de cada *sprint* foi determinado com base nas prioridades e nas necessidades do projeto, levando em consideração os requisitos levantados e as demandas dos usuários. As *sprints* de execução foram definidas da seguinte forma:

***Sprint 1:*** Elaboração e Cadastro de Informações no Banco de Dados

- Definir a estrutura do banco de dados para armazenar as informações relacionadas aos documentos.
- Criar os modelos de dados e as tabelas necessárias para o armazenamento dos documentos e suas informações.
- Implementar os formulários e campos necessários para o cadastro das informações no sistema.
- Desenvolver a lógica de validação dos campos e inserção dos dados no banco de dados.
- Realizar testes para garantir o correto funcionamento do cadastro de informações.

***Sprint 2:*** Visualização de documento

- Implementar as páginas no sistema para exibir os documentos existentes.
- Realizar a integração com o banco de dados para recuperar as informações relacionadas a cada documento.
- Desenvolver a lógica de apresentação dos requisitos e informações do documento na página correspondente.

***Sprint 3:*** Edição de informação

- Adicionar a funcionalidade de edição nas páginas dos documentos.
- Implementar campos editáveis que permitam a alteração das informações cadastradas.
- Garantir que as alterações sejam salvas corretamente no banco de dados.

***Sprint 4:*** Cadastro de novo documento

- Criar um formulário de cadastro de novo documento.
- Implementar a validação dos campos e a inserção dos dados no banco de dados.
- Adicionar a lógica necessária para exibir o novo documento na lista de documentos.

***Sprint 5:*** Listar por tipo

- Implementar a opção de seleção de tipo de projeto no sistema.
- Desenvolver a funcionalidade de filtragem da lista de documentos com base no tipo de projeto selecionado.

- Exibir corretamente a lista de documentos filtrados de acordo com o tipo de projeto escolhido.

#### ***Sprint 6:*** Implementação de Login com Senha

- Desenvolver a funcionalidade de autenticação de usuário no sistema.
- Criar a página de login, onde os usuários poderão inserir suas credenciais.
- Implementar a validação das informações de login e senha.
- Integrar com um mecanismo de segurança, como autenticação baseada em tokens ou criptografia de senha.
- Garantir que apenas usuários autenticados tenham acesso às funcionalidades do sistema.

#### ***Sprint 7:*** Importação de Dados a partir de Arquivo Excel

- Desenvolver uma página no sistema para receber um arquivo Excel contendo as informações a serem populadas no banco de dados.
- Implementar a funcionalidade de upload de arquivo, permitindo que os usuários selecionem o arquivo Excel a ser importado.
- Realizar a leitura e análise do arquivo Excel, extraindo as informações relevantes para popular o banco de dados.
- Validar os dados extraídos do arquivo, garantindo que estejam corretos e coerentes com as regras de negócio do sistema.
- Realizar a inserção dos dados no banco de dados, seguindo as relações e estrutura definidas.
- Realizar testes e verificações para assegurar que a importação de dados, a partir do arquivo Excel, esteja funcionando corretamente.

No início de cada *sprint*, a equipe realizou uma reunião de planejamento para definir quais funcionalidades iriam ser desenvolvidas e estabelecer metas claras e alcançáveis para o período. Durante a *sprint*, foram realizadas reuniões semanais de acompanhamento, para manter a equipe alinhada, identificar possíveis obstáculos e ajustar o planejamento.

Ao final de cada *sprint* ocorreu uma revisão para demonstrar as funcionalidades desenvolvidas e colher feedback do Project Manager e Scrum Master. Essa revisão permitiu a avaliação do progresso, pontuação de ajustes e definição de prioridades para a próxima *sprint*.

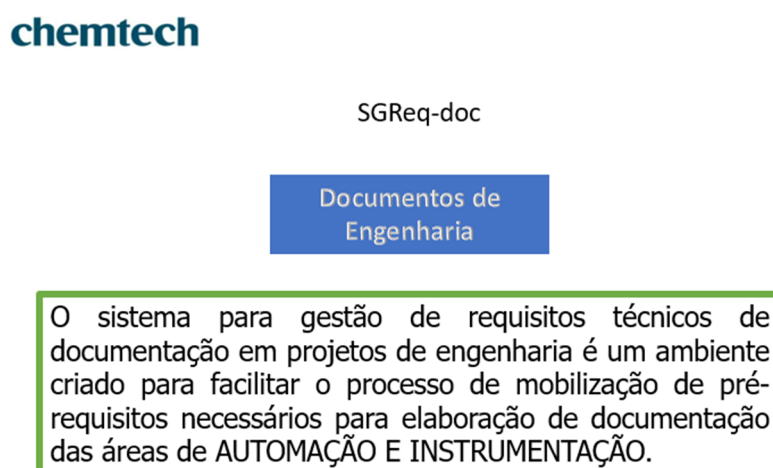
### 3.4 WIREFRAME

A fim de definir como seria sua interface básica com o usuário, foi elaborado um *Wireframe*<sup>2</sup>, baseado nos requisitos funcionais e não-funcionais obtidos como resultado das entrevistas com usuários, o qual mostra a disposição de elementos como botões, campos de texto e outros componentes de interface, mas não inclui cor, estilo ou outros elementos de design.

O protótipo é composto por 3 páginas principais, sendo elas: Página Inicial, Página de Overview da Lista de Documentos e Página de Visualização de Documento. No esquemático do Wireframe, os campos em azul são botões e os campos verdes são elementos textuais.

Na Figura 4, tem-se a tela inicial da aplicação que consiste em: breve descrição do sistema e botão que abre a Página de Overview da Lista de Documentos.

Figura 4 - Página Inicial

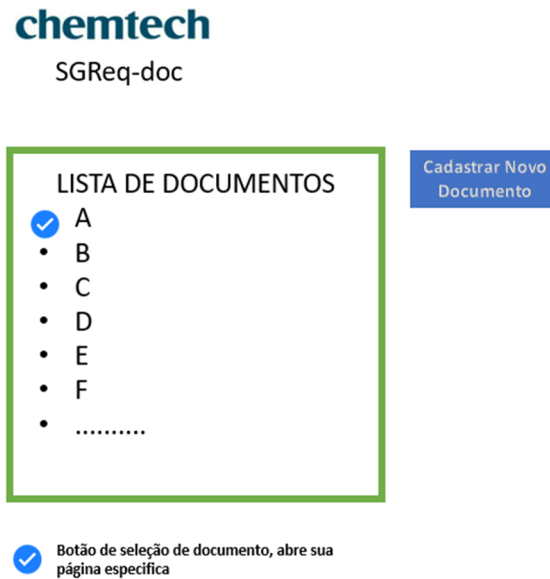


A Figura 5 esquematiza a página que conterá todos os documentos cadastrados no sistema e disponíveis para consulta, sendo possível selecionar um dos documentos da lista pelo botão azul em frente ao seu nome. Ao clicar nesse botão, a Página de Visualização de Documento é aberta, além disso também será possível cadastrar um novo documento.

---

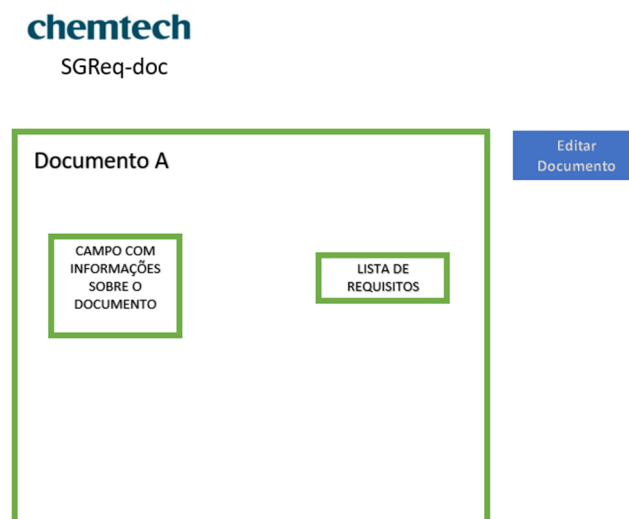
<sup>2</sup> Wireframe: Representação visual do layout e da estrutura de uma interface de usuário, normalmente usada nos estágios iniciais do desenvolvimento de aplicativo.

Figura 5 - Página de Overview da Lista de Documentos



Na Figura 6, é demonstrada a Página de Visualização de Documento, onde estarão as informações descritivas do documento e sua lista de requisitos, assim como um botão para editar e inserir informações.

Figura 6 - Página de Visualização de Documento



Essas serão as páginas desenvolvidas no escopo deste trabalho. Vale ressaltar que o Wireframe é a ferramenta inicial de design da aplicação, sendo assim, não consiste em um modelo fiel do produto finalizado.



### 3.5 REQUISITOS DE DOCUMENTAÇÃO

Os documentos analisados para a construção do sistema são de domínio da Petrobras e se encontram disponíveis no site <https://canalforneceador.petrobras.com.br/pt/regras-de-contratacao/catalogo-de-padronizacao/>. Neste catálogo, foram selecionadas a norma N-1883 “Apresentação de Projeto de Instrumentação/Automação”, a qual descreve os documentos que são elaborados para as áreas de instrumentação e automação.

Com base na norma N-1883, os projetos foram categorizados em dois grupos: Projeto Básico e Projeto Executivo. A partir dessas categorias, foram relacionados os documentos correspondentes a cada tipo de projeto, o quais estão relacionados nas Tabela 1 e Tabela 2. Para cada documento selecionado, foram realizadas análises nos arquivos da base de dados da Chemtech, com o objetivo de identificar e catalogar os requisitos específicos de cada um.

Tabela 1 – Projeto Básico

<b>TIPO DE PROJETO</b>	<b>DOCUMENTO</b>
<b>PROJETO BÁSICO</b>	Arquitetura de Sistema de Automação e Controle
	Descritivo de Malhas de Controle
	Folha de Dados de SIF
	Matriz de Causa e Efeito
	Memória de Cálculo de Verificação do SIL e do MTTFS da SIF
	Memorial Descritivo de Automação de Equipamentos
	Memorial Descritivo do Sistema de Instrumentação

Fonte:N-1883, Petrobras (2016)

Tabela 2 – Projeto Executivo

<b>TIPO DE PROJETO</b>	<b>DOCUMENTO</b>
<b>PROJETO EXECUTIVO</b>	Diagrama de Controle de Processo
	Diagrama de Interligação Elétrica
	Diagrama de Malha
	Diagrama de Segmento de Rede
	Diagrama Funcional
	Diagrama Lógico
	Folha de Dados de Instrumentos
	Lista de Cabos
	Lista de Cargas Elétricas de Instrumentação
	Lista de Comunicação
	Lista de Entradas e Saídas
	Lista de Instrumentos
	Listas de Materiais
	Mapa de Carregamento
	Memória de Cálculo de Dimensionamento de Elementos Primários de Vazão
	Memória de Cálculo de Dimensionamento de Poço de Temperatura
	Memória de Cálculo de Dimensionamento de Válvula de Alívio e Segurança
	Memória de Cálculo de Dimensionamento de Válvula de Controle
	Memória de Cálculo para Elevação de Temperatura em Painéis
	Memória de Cálculo para Medição de Nível através de Pressão Diferencial
	Memorial de Cálculo de Enlace Óptico
	Memorial Descritivo de Automação
	Parecer Técnico
	Planta de Instrumentação Elétrica
	Planta de Instrumentação Pneumática
Requisição de Material	

Fonte: N-1883, Petrobras (2016)

O sistema utilizará essa base de dados para relacionar os documentos aos seus requisitos correspondentes. Isso garantirá uma visualização clara das exigências e critérios contidos em cada documento, facilitando a busca por informações específicas e a rastreabilidade dos requisitos ao longo do ciclo de vida dos projetos que implementarem seu uso.

### 3.6 MODELO DE DADOS

O banco de dados do SG\_REQ foi projetado e implementado utilizando a plataforma Mendix. Ele armazena e gerencia as informações relacionadas aos documentos, projetos e requisitos do sistema. Através do Mendix, foi possível modelar uma estrutura simples e escalável que atende às necessidades de armazenamento e consulta dos dados.

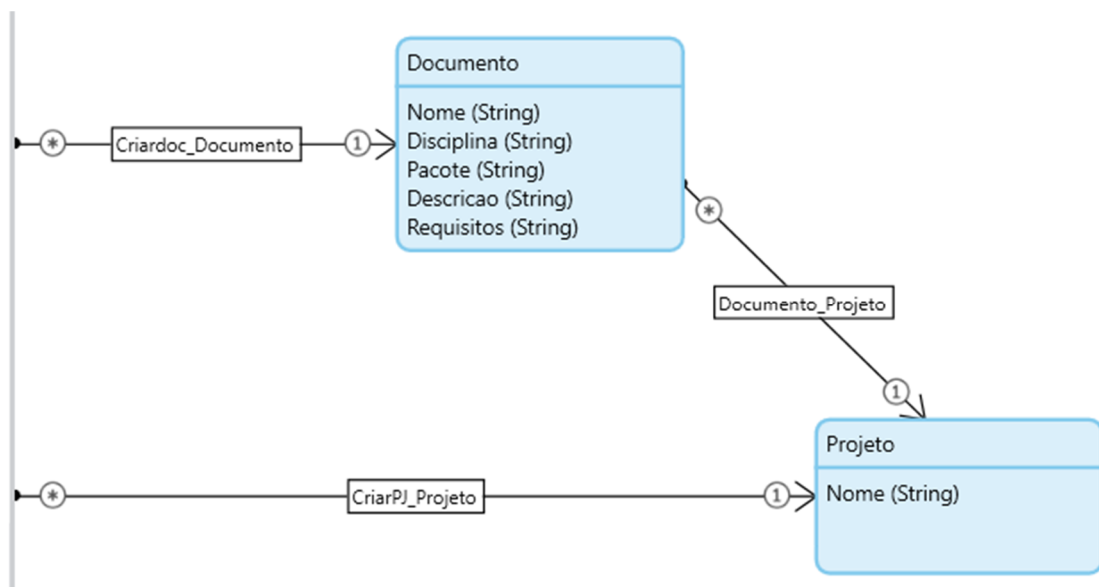
Como o sistema visa ser uma plataforma cooperativa, a povoação do banco de dados se dará por meio de planilhas estruturadas de acordo com as entidades e relacionamentos que foram definidos de acordo com requisitos de software.

Para que seja feito o cadastro, visualização, edição e relacionamento com seu tipo de projeto, foram utilizadas as tabelas Documento e Projeto, apresentadas na Figura 7. A tabela Documento contém entidades que são as principais informações de interesse desde projeto:

- Nome (String): Armazena o título do documento.
- Disciplina (String): Representa a disciplina relacionada ao documento.
- Pacote (String): Indica o pacote ao qual o documento pertence.
- Descrição (String): Contém uma descrição breve e informações adicionais sobre as normas de elaboração do documento.
- Requisitos (String): Contém os requisitos específicos do documento.

Já a tabela Projeto conta somente com o Nome para que seja feita a classificação adequada dos documentos.

Figura 7 - Modelo de dados do SG\_Req



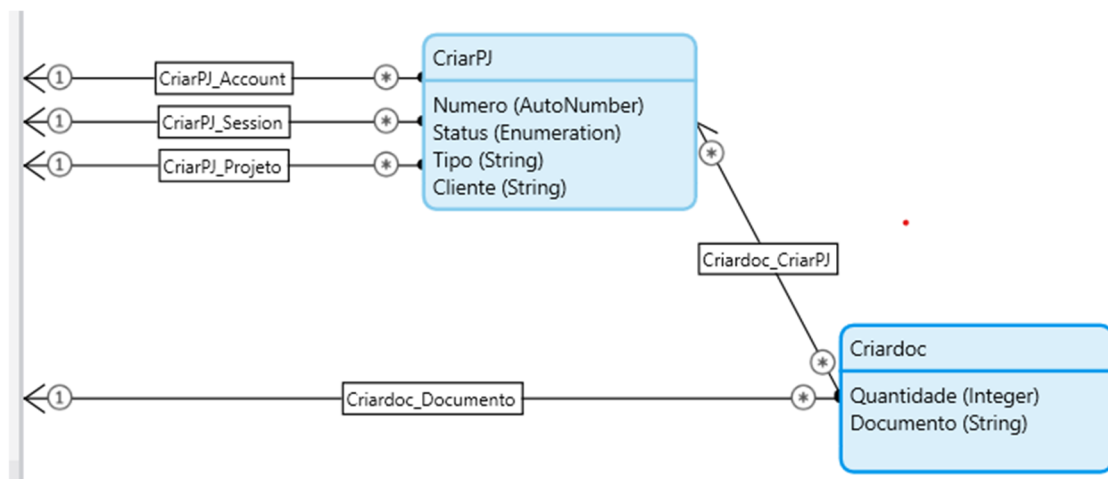
O modelo de dados do SG\_REQ utiliza setas para representar as relações entre as tabelas, indicando as conexões entre as entidades. Além disso, existem linhas que partem da Figura, indicando relações com tabelas de outros domínios. Os domínios foram divididos em duas

partes principais: o domínio relacionado ao desenvolvimento das entidades e o domínio relacionado às relações entre as entidades.

O domínio de desenvolvimento das entidades abrange a criação e o gerenciamento das tabelas que armazenam as informações dos documentos, projetos e requisitos do sistema. Por outro lado, o domínio de relação entre as entidades foi criado para tratar os fluxos de consulta, edição das tabelas e abertura de páginas relacionadas.

Na Figura 8, apresenta-se o modelo de dados específico para o domínio de relação entre as entidades. Esse modelo foi criado para estabelecer as conexões necessárias para permitir consultas eficientes, atualizações e navegação entre as tabelas, facilitando assim a interação do usuário com o sistema.

Figura 8 - Modelo de dados para relação entre as entidades.



Após a estruturação dos requisitos de software, definição das *Sprints*, criação do Wireframe, mapeamento dos requisitos de documentação e estruturação do modelo de dados, deu-se início ao desenvolvimento das páginas no ambiente Mendix. Cada *sprint* foi abordada individualmente, com foco nas atividades específicas definidas, visando atender aos requisitos estabelecidos.

## 4. O SG\_REQ

O desenvolvimento da ferramenta foi feito integralmente na plataforma Mendix, a qual conta com uma interface intuitiva, oferecendo uma abordagem visual e permitindo que os desenvolvedores arrastem e soltem componentes para montar a interface do usuário. Foram arquitetadas páginas para visualização, edição e cadastro de documentos de forma rápida e eficiente, sem a necessidade de escrever código extenso, possibilitando um processo de desenvolvimento ágil e produtivo.

De acordo com o planejamento das *Sprints* foram criadas as seguintes páginas:

### 4.1 PÁGINA HOME

Esquematizada na Figura 9, a página Home possui uma breve descrição do sistema e um menu lateral que dá acesso as outras páginas do sistema.

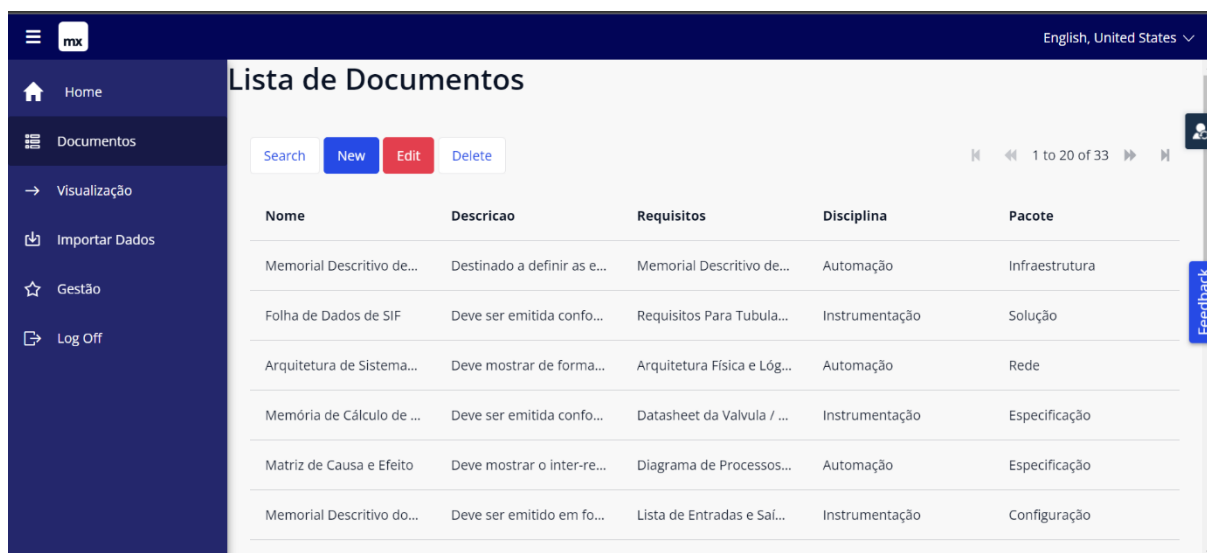
Figura 9 - Página Home



## 4.2 PÁGINA DOCUMENTOS

A página Documentos, ilustrada na Figura 10, é responsável pela exibição dos documentos cadastrados no banco de dados.

Figura 10 - Página Documentos e suas funcionalidades



Além disso, nesta página foram implementadas as principais funcionalidades da ferramenta, que incluem:

### 4.2.1 SEARCH

Função *Search* (Busca), ilustrada na Figura 11, permite que o documento seja procurado digitando seu título, pacote ou descrição.

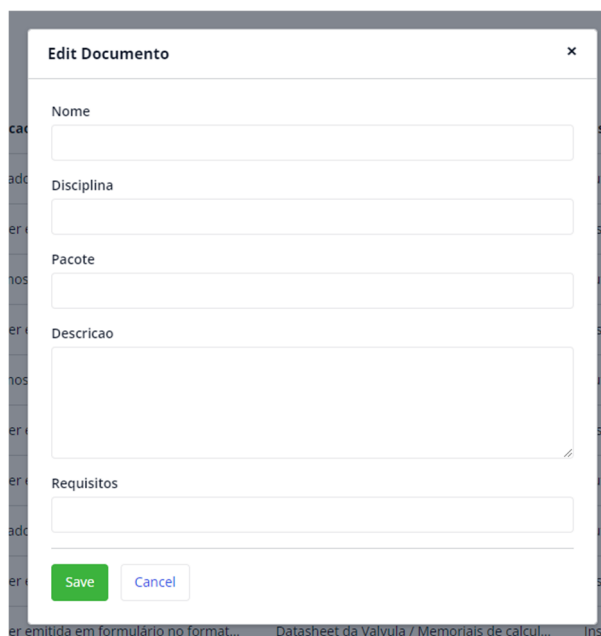
Figura 11 - Função Search



#### 4.2.2 NEW

Função *New* (Novo), ilustrada na Figura 12, permite que seja feito o cadastro de um novo documento no banco de dados.

Figura 12 - Função *New* (Novo)

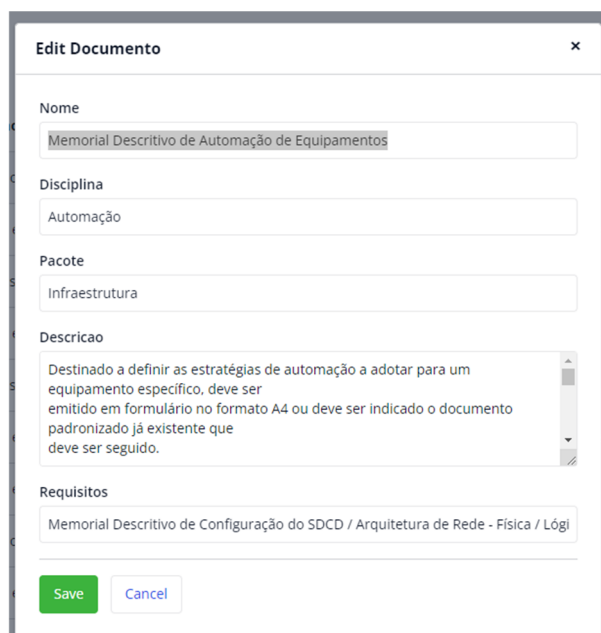


The screenshot shows a web form titled "Edit Documento" with a close button (x) in the top right corner. The form contains several input fields: "Nome" (empty), "Disciplina" (empty), "Pacote" (empty), "Descricao" (a large text area), and "Requisitos" (empty). At the bottom of the form, there are two buttons: a green "Save" button and a white "Cancel" button. The form is overlaid on a background that appears to be a spreadsheet or data table.

#### 4.2.3 EDIT

Função *Edit* (Editar), ilustrada na Figura 13, permite que seja feita a edição de um documento existente.

Figura 13 - Função *Edit* (Editar)



The screenshot shows the same "Edit Documento" form as in Figure 12, but with pre-filled data. The "Nome" field contains "Memorial Descritivo de Automação de Equipamentos", "Disciplina" contains "Automação", and "Pacote" contains "Infraestrutura". The "Descricao" field contains the text: "Destinado a definir as estratégias de automação a adotar para um equipamento específico, deve ser emitido em formulário no formato A4 ou deve ser indicado o documento padronizado já existente que deve ser seguido." The "Requisitos" field contains "Memorial Descritivo de Configuração do SDCD / Arquitetura de Rede - Física / Lógi". The "Save" and "Cancel" buttons are at the bottom.

#### 4.2.4 DELETE

Função *Delete* (Deletar), ilustrada nas Figuras Figura 14 e Figura 15, permite que um documento existente seja deletado do banco de dados.

Figura 14 - Função Delete, seleção de documento.

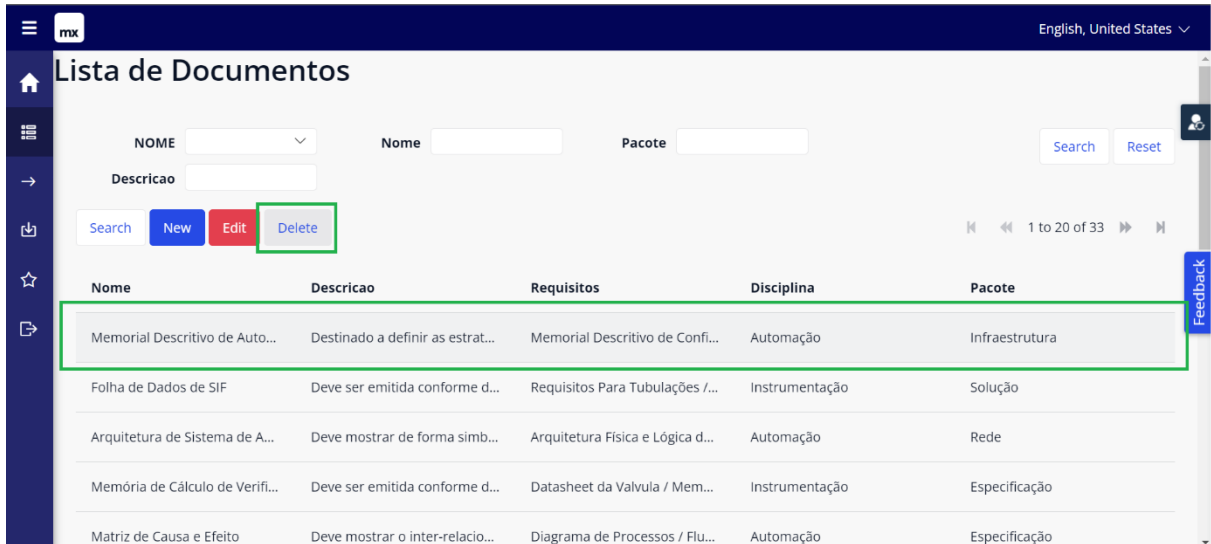
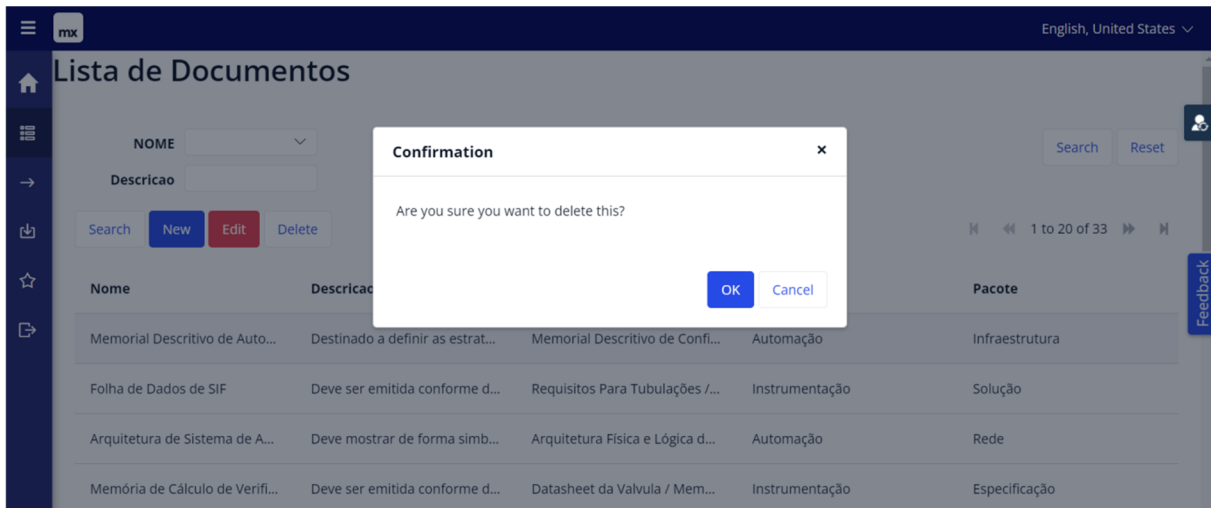


Figura 15 - Função Delete, confirmação de exclusão de documento selecionado

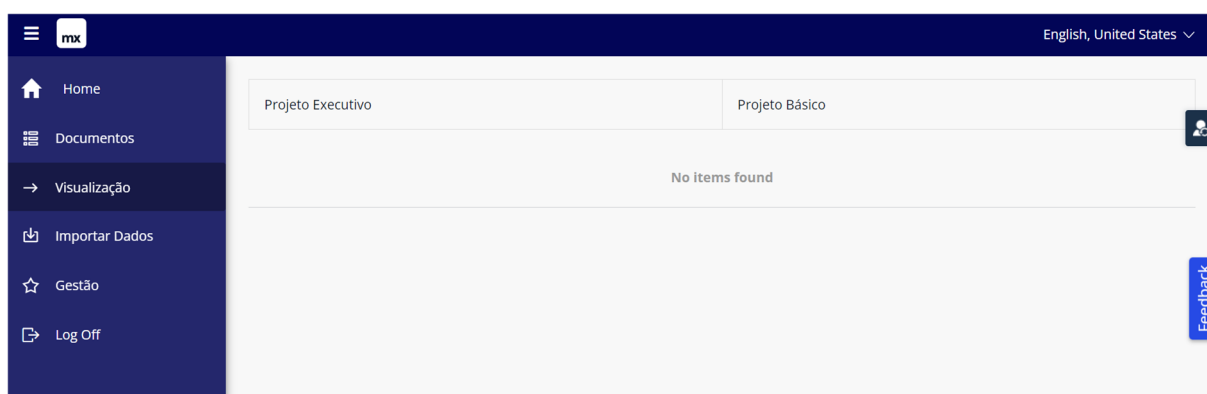




### 4.3 PÁGINA VISUALIZAÇÃO

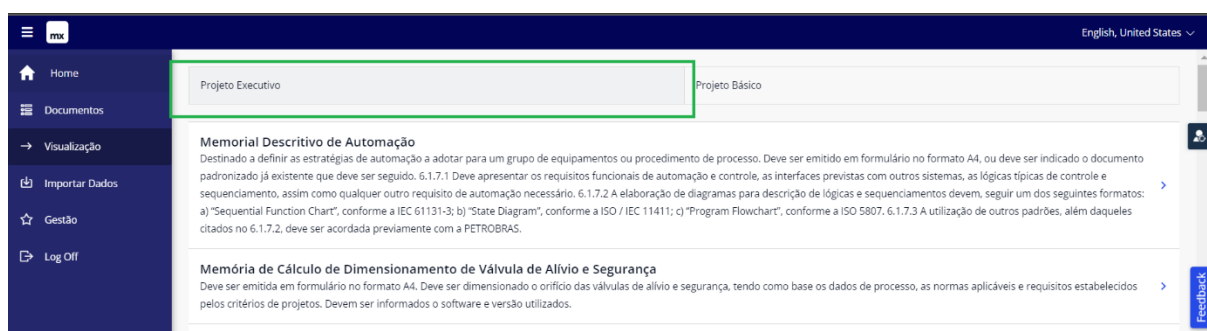
A página Visualização, representada na Figura 16, permite aos usuários visualizarem os documentos cadastrados, diferentemente da página "Documentos", que se concentra nas funções de edição, inserção e remoção. Nessa página, os documentos são organizados por tipo de projeto, facilitando sua classificação e tornando mais fácil encontrar informações específicas.

Figura 16 - Página Visualização



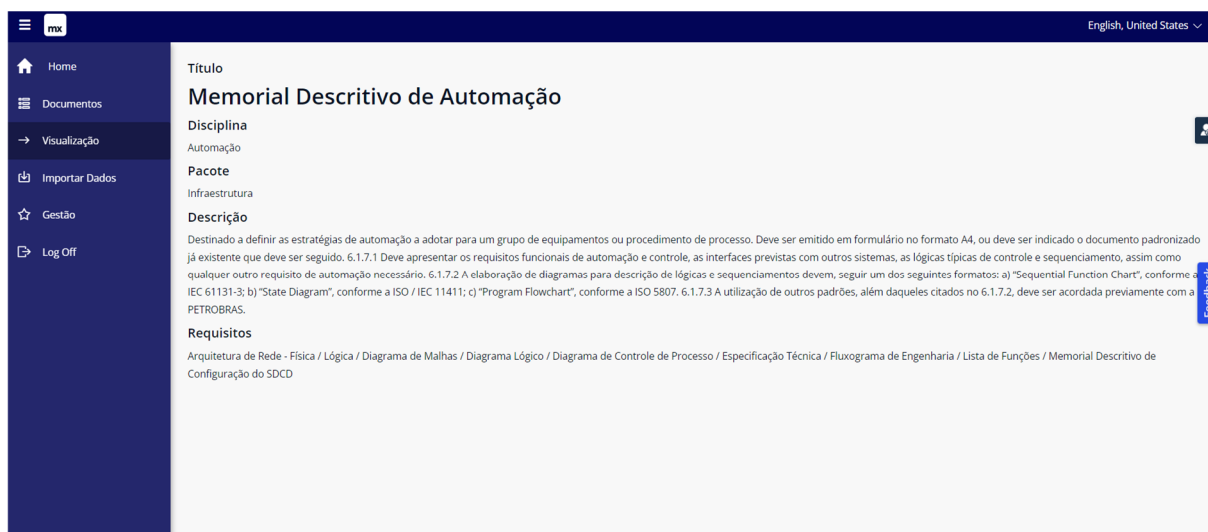
Ao clicar no título do tipo de projeto, uma lista de documentos relacionados é exibida, como representado na Figura 17.

Figura 17 - Função filtrar por tipo de projeto.



Ao clicar na seção de cada documento é aberta sua página específica, a qual exibe todas suas informações como título, disciplina, pacote e o principal: seus requisitos, como demonstrado na Figura 18.

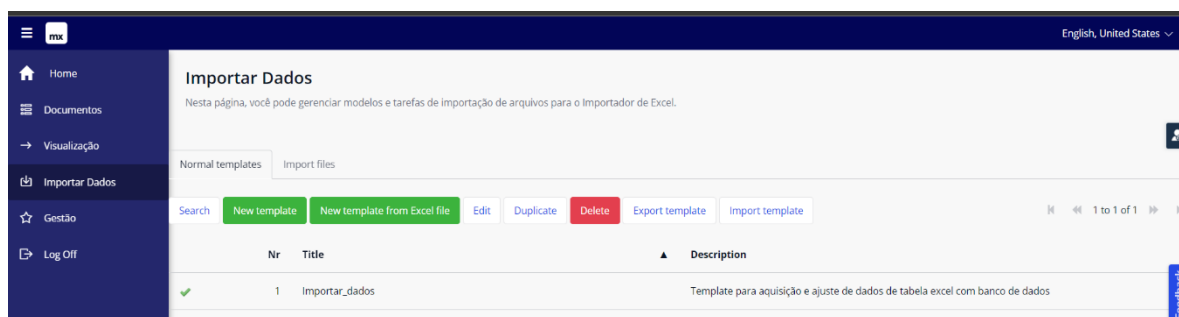
Figura 18 - Página de exibição de documento específico



#### 4.4 PÁGINA IMPORTAR DADOS

A página Importar Dados, representada na Figura 19, foi implementada utilizando o modelo da plataforma Mendix chamado "Excel Importer". Essa página oferece aos usuários uma interface intuitiva para realizar a importação de dados por meio de arquivos Excel.

Figura 19 - Página Importar Dados



Dentro da aba "Normal templates", é possível definir mapeamentos entre as colunas do arquivo Excel e os atributos correspondentes no banco de dados do SG\_Req. ao configurar o mapeamento de colunas, os usuários têm o controle total sobre como os dados serão inseridos e atualizados no sistema. Eles podem definir regras de correspondência, ignorar ou mapear colunas específicas, e até mesmo criar lógicas avançadas para tratar situações especiais durante o processo de importação. A Figura 20 demonstra as regras criadas para importação dos dados de acordo com as entidades do SG\_Req.

Figura 20 - Template utilizado para aquisição de dados via tabela Excel

Connect columns to attributes

Col Nr 0 = A, ... 10 = K, ... etc

Search New Edit Delete Connect matching attributes

Col nr ▲	Caption	Is key	Case S.	Type	Details	Parser
✓ 0	Nome	Yes, only for the ma...	No	Attribute	Attribute: Nome, type: String	
✓ 1	Disciplina	NoKey	No	Attribute	Attribute: Disciplina, type: String	
✓ 2	Pacote	NoKey	No	Attribute	Attribute: Pacote, type: String	
✓ 3	Descricao	NoKey	No	Attribute	Attribute: Descricao, type: String	
✓ 4	Requisitos	NoKey	No	Attribute	Attribute: Requisitos, type: String	
✓ 5	Projeto	Yes, only for the ass...	No	Reference	Reference: MasterData.Documento_Projeto/MasterData.Projeto/No...	

Save Cancel

Já na aba "Import files", destacada em verde e representada na Figura 21, os usuários podem selecionar os arquivos Excel que desejam importar para o SG\_Req. Esses dados serão organizados de acordo com as regras criadas na aba "Normal Templates". Para testar essa funcionalidade, foi criado o modelo de regras "Importar\_dados" e a tabela Documentos, a qual foi o resultado do mapeamento dos requisitos de engenharia. Obteve-se êxito na importação de dados através do template "Excel Importer".

Figura 21 - Função de importação de tabelas

mx English, United States

Importar Dados

Nesta página, você pode gerenciar modelos e tarefas de importação de arquivos para o Importador de Excel.

Normal templates Import files

Search New Edit Import file Delete

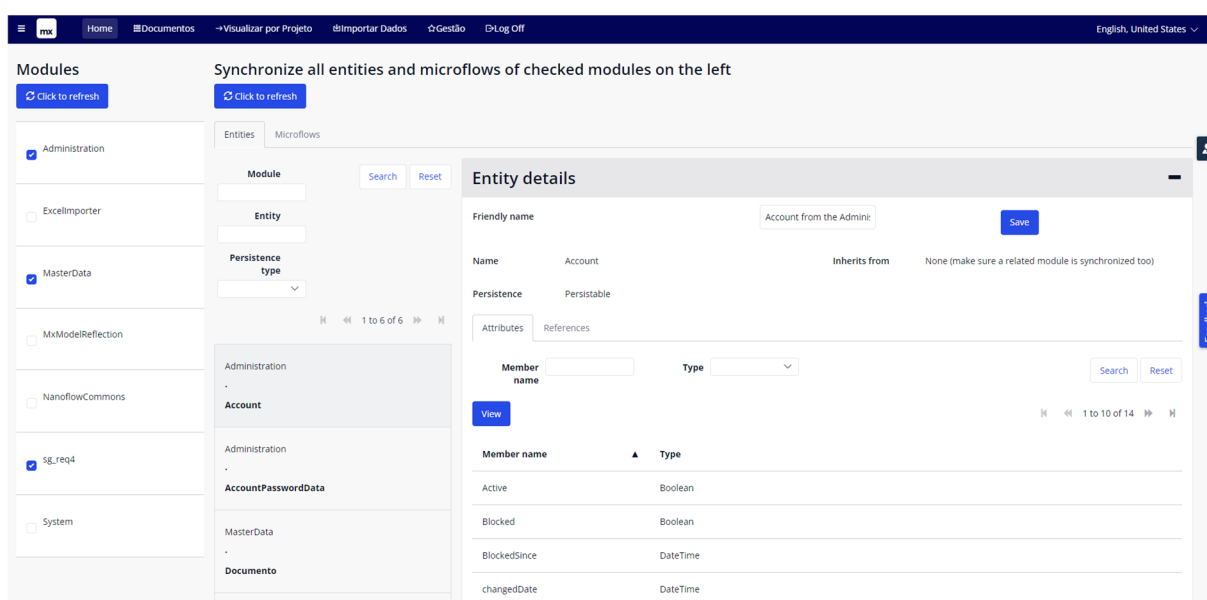
Filename	Template nr	Template name
Documento.xlsx	1	Importar_dados

Feedback

## 4.5 PÁGINA GESTÃO

A página Gestão, ilustrada na Figura 22, utiliza o módulo "Mx Model Reflection" do Mendix para possibilitar a atualização e gerenciamento dinâmico das entidades e suas regras no banco de dados. Esse módulo permite transferir os metadados das entidades, como nomes, descrições e atributos, e oferece a verificação das propriedades dos atributos, incluindo tipos de dados e restrições. Além disso, possibilita o acesso às associações entre as entidades, incluindo multiplicidades e direções, e permite modificar o modelo de dados em tempo de execução.

Figura 22 - Página Gestão.



## 4.6 PÁGINA DE LOGIN

A página de login, representada na Figura 23, garante a segurança e restrição de acesso ao sistema. Nela, os usuários podem autenticar suas identidades por meio de credenciais, nome de usuário e senha, antes de obter acesso às funcionalidades do sistema.

Figura 23 - Página de Login



## 5. CONCLUSÕES E TRABALHOS FUTUROS

O projeto do SG\_REQ foi desenvolvido com o objetivo de automatizar e facilitar o processo de elaboração, organização e armazenamento de informações relacionadas aos documentos técnicos de projetos. Ao longo da implementação deste projeto, explorou-se diversos conceitos, metodologias e tecnologias que contribuíram para o desenvolvimento deste sistema.

No início do projeto, foi identificada a importância da documentação dentro do escopo dos projetos de engenharia, fornecendo registros detalhados e fundamentais para a comunicação, conformidade regulamentar e garantia de qualidade. Com base nessa necessidade, estabeleceu-se o propósito de criar um sistema que pudesse centralizar, padronizar e simplificar o gerenciamento dessas informações, visando a consistência e qualidade dos documentos produzidos.

Para alcançar esse objetivo, optou-se por utilizar a plataforma Mendix, uma ferramenta de desenvolvimento de aplicativos *Low-Code* que proporcionou agilidade e facilidade na criação do SG\_REQ. Através do Mendix, desenvolve-se a arquitetura, o banco de dados do sistema, e suas páginas utilizando entidades, relacionamentos e domínios para estruturar e organizar as informações de forma coerente.

A modelagem do banco de dados do SG\_REQ garantiu a eficiência e escalabilidade do sistema e, a partir dele, foram criadas as páginas do sistema, as quais possuem todas as funcionalidades mapeadas durante as entrevistas com usuários. Dividiu-se as entidades em domínios, em que o domínio de desenvolvimento se relacionou às informações específicas de cada documento, e o domínio de relação entre entidades focou-se nos fluxos de consulta, edição e abertura de páginas.

Ao longo do desenvolvimento, fez-se uso da metodologia ágil Scrum para gerenciar o projeto de forma iterativa e incremental. Definiu-se *sprints*, que representavam ciclos de trabalho com objetivos bem definidos. Cada *sprint* abordou requisitos específicos do sistema, como a visualização de documentos, edição de informações, cadastro de novos documentos, listagem por tipo e a implementação de login com senha para segurança do software. Essa abordagem permitiu controle efetivo do progresso e uma entrega contínua de funcionalidades.

O resultado do projeto consiste em um protótipo do SG\_REQ, que tem como principal foco a fase de mapeamentos de requisitos na elaboração de documentos técnicos de engenharia. Embora seja uma versão básica, o sistema cumpre o objetivo de centralizar, organizar e simplificar o acesso às informações de como encontrar e mapear os requisitos técnicos de documentos dos projetos de engenharia. Além do mais, o sistema oferece recursos como visualização, edição e cadastro de informações, também como funcionalidades de listagem por tipo e segurança com login e senha.

Entre os diferenciais do SG\_Req vale ressaltar sua arquitetura de código. Projetado de forma intuitiva e escalável, ele transcende as aplicações iniciais para as quais foi desenvolvido. Embora originalmente concebido para gerir os requisitos da Petrobrás na área de automação e instrumentação, sua flexibilidade permite adaptá-lo para atender às necessidades de outras empresas e disciplinas. No entanto, o SG\_Req apresenta áreas que requerem melhorias, como a funcionalidade de busca de documentos, apesar de possuir uma página dedicada a essa função, ela não está integrada à página principal de visualização, o que pode ser desvantajoso para sistemas que possuam grande volume de documentos.

É importante destacar que o projeto também teve como propósito explorar a plataforma Mendix como uma alternativa viável para o time de automação e instrumentação da ChemTech, evidenciando as capacidades e vantagens dessa plataforma no desenvolvimento de aplicativos corporativos.

Em suma, o projeto do SG\_REQ representa um passo importante no aprimoramento dos processos de gestão de requisitos de documentação em projetos de engenharia. Através da aplicação da metodologia ágil Scrum, utilização da plataforma Mendix e adoção de boas práticas de desenvolvimento, foi possível desenvolver um protótipo funcional e alinhado com as necessidades dos usuários. O SG\_REQ demonstrou ser uma ferramenta promissora para otimizar o trabalho das equipes de engenharia, centralizar as informações e garantir a qualidade e consistência dos documentos produzidos.

## **5.1 TRABALHOS FUTUROS**

Embora o projeto do SG\_REQ tenha sido concluído com sucesso, por ele ter sido implementado com foco na etapa de coleta de requisitos, algumas funcionalidades mapeadas nas entrevistas com usuário não foram implementadas, fomentando escopo para futuras

melhorias e expansões. Abaixo são listadas as melhorias mapeadas para versões futuras do sistema:

1. Melhorias na coordenação técnica: Desenvolver recursos que facilitem a coordenação entre as disciplinas e a comunicação com os clientes. Isso pode incluir a lista de precedências para cobrança do cliente e entre as disciplinas, bem como a definição de requisitos para reuniões iniciais com o cliente.
2. Criação de página para gestão de projetos: essa página permitirá o gerenciamento dos documentos associados a projetos personalizados, proporcionando uma coordenação mais eficiente e a validação do planejamento.
3. Reforçar a base de conhecimento técnico: Desenvolver um campo de ajuda que forneça informações detalhadas sobre as especificações técnicas e orientações para adquirir o conhecimento necessário para elaborar cada documento.
4. Análise de Dados e Geração de Relatórios: A partir dos dados armazenados no SG\_REQ, é possível realizar análises e gerar relatórios para obter *insights* valiosos sobre o desempenho dos projetos, tendências de requisitos, qualidade da documentação, entre outros aspectos relevantes. Essa análise de dados pode fornecer informações úteis para a tomada de decisões e melhoria contínua dos processos.

Esses são apenas alguns exemplos de possíveis trabalhos futuros para o SG\_REQ. É importante ressaltar que o desenvolvimento do sistema é um processo contínuo, e as necessidades e exigências da empresa e dos usuários podem evoluir ao longo do tempo. Portanto, é fundamental manter uma abordagem ágil e adaptativa, buscando constantemente melhorias e inovações que atendam às demandas em constante mudança.

## REFERÊNCIAS BIBLIOGRÁFICAS

WATTS, Frank B. *Engineering Documentation Control Handbook*. 4. ed. William Andrew, 2011.

LOVE, Jonathan. *Process Automation Handbook: A Guide to Theory and Practice*. 1. ed. Softcover. 2007.

PROJECT MANAGEMENT INSTITUTE. *Um Guia do Conhecimento em Gerenciamento de Projetos*. Guia PMBOK®. 6. ed. EUA: Project Management Institute, 2017.

LICHTENTHAELER, Robin; BÖHM, Sebastian; MANNER, Johannes; WINZINGER, Stefan. *A Use Case-based Investigation of Low-Code Development Platforms*. Congresso Virtual: ZEUS2022. Distributed Systems Group. University of Bamberg, Germany, Fevereiro, 2022.

SAHAY, Apurvanand; INDAMUTSA, Arsene; DI RUSCIO, Davide; PIERANTONIO, Alfonso. *Supporting the Understanding and Comparison of Low-Code Development Platforms*. 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica. Università degli Studi dell'Aquila, Agosto, 2020.

SAPUTELLI, Luigi; BLACK, Alan; PASSALACQUA, Herminio; BARRY, Kevin. *Front-End-Loading (FEL) Process Supporting Optimum Field Development Decision Making*. SPE Kuwait Oil and Gas Show and Conference, Mishref, Kuwait. Society of Petroleum Engineers, Outubro, 2013.

BASILIO, Leandro. *Managing your offshore project: Front-end Loading and cost accuracy*. OilandGas IQ, 23 junho 2017. Disponível em: <https://www.oilandgasiq.com/strategy-management-and-information/articles/managing-your-offshore-project-front-end-loading>.

Acesso em: 21 de janeiro de 2023.

BRASIL. Presidência da República. Secretaria-Geral. Subchefia para Assuntos Jurídicos. *Decreto N° 8.437*, de 22 de abril de 2015. Disponível em: [http://www.planalto.gov.br/ccivil\\_03/\\_Ato2015-2018/2015/Decreto/D8437.htm](http://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2015/Decreto/D8437.htm). Acesso em: 21 de janeiro de 2023.



BARBOSA, P. T. et al. *Metodologia FEL: Sua Importância na Avaliação de Riscos e Redução de Impactos em Escopo, Tempo e Custo de Projetos Complexos de Engenharia*. XXXIII Encontro Nacional de Engenharia de Produção – ENEGEP, Salvador, 2013.

VALENTE, Marco Tulio. *Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade*. Editora: Independente, 2020.

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

GARCIA-MOLINA, Hector; ULLMAN, Jeffrey D.; WIDOM, Jennifer. *Database Systems: The Complete Book*. 2. ed. Upper Saddle River, New Jersey: Pearson Education, 2008.