



UNIVERSIDADE ESTADUAL DE GOIÁS - UEG

WELTON LEITE DA PAIXÃO

**ANÁLISE DE REQUISITOS NO PROCESSO LEGISLATIVO MUNICIPAL:
Uma Abordagem Para Construção de Software**

**TRINDADE
2023**

WELTON LEITE DA PAIXÃO

**ANÁLISE DE REQUISITOS NO PROCESSO LEGISLATIVO MUNICIPAL:
Uma Abordagem Para Construção de Software**

Monografia final de conclusão de curso
apresentada ao Colegiado de Sistema da
Informação – UEG como parte dos
requisitos para obtenção do título de
Bacharel em Sistema da Informação

**Área de concentração: Engenharia de
Software**

Orientador: Prof. Cláudio Garcia

**TRINDADE
2023**

WELTON LEITE DA PAIXÃO

ANÁLISE DE REQUISITOS NO PROCESSO LEGISLATIVO MUNICIPAL
Uma Abordagem Para Construção de Software

Monografia aprovada em ____/____/____ para obtenção do título de Bacharel em
Sistema da Informação.

Comissão Examinadora

Prof. Cláudio Garcia
Universidade Estadual de Goiás

Prof. Olegário Neto
Universidade Estadual de Goiás

Dr. José Carlos Domingues
Procurador da Câmara Municipal de Trindade - GO

TRINDADE
2023

Em memória dos meus pais.

AGRADECIMENTOS

Seria impossível não mencionar minha esposa Sara, meus filhos Khallel e Heloá pelo apoio incondicional em meus estudos. Eu apenas posso mostrar o caminho aos meus filhos, para que possam seguir o meu exemplo de estudar e seguir no melhor caminho possível.

Aos professores Helber Lima Frazão, que foi um grande mentor.

Ao professor Olegário Correa Neto que muito somou em minha jornada acadêmica.

Ao professor e mestre Fábio que é um referencial de carreira, postura e conhecimento.

À professora Nágela que muito me ajudou, principalmente no começo acadêmico.

Ao mestre Paulo Cesar de Almeida, que é uma fonte infinita de conhecimento.

Ao professor Cláudio Garcia, simplesmente o maior gênio que já conheci em minha vida.

A esses professores extraordinários que me incentivaram e me fizeram lutar pelo melhor em minha vida; a vocês, minha eterna gratidão.

Ao amigo Alexandre Souza, que foi a primeira pessoa que me motivou a iniciar uma vida de estudos acadêmicos. Você mudou minha história, meu irmão de coração.

Aos amigos que conheci e muito me ajudaram na caminhada no curso e a tantos outros que já não estão conosco devido ao Covid, como o grande amigo Goiermano Lima.

A Deus, pois ele é a minha força e minha inspiração, nos momentos mais difíceis em minha vida.

LISTAS DE FIGURAS

Figura 1: Diagrama de planejamento.....	16
Figura 2 : Tabela de tempos para desenvolvimento.....	19
Figura 3: Quadro de perguntas de usuário para escopo.....	20
Figura 4: Evolução dos requisitos	21
Figura 5: Passos para a utilização da técnica de questionário.....	26
Figura 6: Diagrama de casos de uso	28
Figura 7: Descrição contínua.....	29
Figura 8: Caso de uso processar venda	29
Figura 9: Caso de uso processar venda (continuação)	30
Figura 10: Técnica consenso no escopo	31
Figura 11: Modelo de análise documental	32
Figura 12: O paradigma da prototipação.....	34
Figura 13: Logo star UML.....	41
Figura 14: Tela inicial star UML.....	41
Figura 15: Diagrama Entidade Relacionamento	42
Figura 16: Diagrama de Classes.....	43
Figura 17: Diagrama de Caso de Uso.....	47
Figura 18: Diagrama de Caso de Uso.....	47
Figura 19: Diagrama de Caso de Uso.....	48
Figura 20: Diagrama de Caso de Uso.....	48
Figura 21: Diagrama de Caso de Uso.....	49
Figura 22: Diagrama de Caso de Uso.....	49
Figura 23: Diagrama de Caso de Uso.....	50

LISTA DE ABREVIATURAS

IEEE	Institute of Electrical and Electronics Engineers
DC	Desenvolvimento Conhecido
DNC	Desenvolvimento Não Conhecido
MAN	Manutenção de Sistemas
UML	Unified Modeling Language

RESUMO

Este trabalho de conclusão do curso tem o objetivo de apresentar as técnicas de levantamento de requisitos para desenvolvimento de software dentro do Poder Legislativo, quer seja na esfera Federal, Estadual ou Municipal, tendo em vista que a maioria das câmaras municipais não possuem tal sistema, pois apenas a Câmara de Jataí possui um sistema similar em *Python*. A criação de um sistema para as pautas de sessão seria totalmente possível com sua implementação e a digitalização do processo legislativo. Isso tornaria mais didáticas as reuniões, bem como a praticidade e rapidez no que tange às votações. As informações foram coletadas por meio de entrevista junto à procuradoria jurídica da Câmara Municipal de Trindade, e a análise documental da Constituição Federal e do Regimento Interno da Câmara Municipal de Trindade. Um fator é que nos envios dos documentos para votação, as redundâncias de informação são corriqueiras e a procuradoria jurídica fica à mercê de erros, tais como: inconstitucionais, redundantes e redigidos incorretamente. O fato do procurador jurídico não dar um parecer por meio de sistema, também limita seu poder de atuação pela sua atribuição desse processo. Por fim, com a criação dos diagramas de classes, caso de uso e entidade e relacionamento podemos dar uma abrangência para futura implementação do software.

Palavras chaves: elicitación de requisitos, análise de requisitos, levantamento de requisitos, técnicas no levantamento de requisitos.

ABSTRACT

This course conclusion work aims to present the techniques for gathering requirements for software development within the Legislative Branch, whether at the Federal, State or Municipal level, considering that the majority of municipal councils do not have such a system, as only the Jataí Chamber has a similar system in Python. The creation of a system for session agendas would be entirely possible with its implementation and the digitalization of the legislative process. This would make meetings more didactic, as well as practical and quick in terms of voting. The information was collected through an interview with the legal attorney of the Trindade City Council, and documentary analysis of the Federal Constitution and the Internal Regulations of the Trindade City Council. One factor is that when sending documents for voting, information redundancies are common and the legal prosecutor's office is at the mercy of errors, such as: unconstitutional, redundant and incorrectly written. The fact that the legal attorney does not provide an opinion through the system also limits his power of action due to his attribution in this process. Finally, with the creation of class, use case and entity and relationship diagrams, we can provide scope for future software implementation.

Keywords: requirements elicitation; requirements analysis; requirements gathering; requirements gathering techniques.

SUMÁRIO

INTRODUÇÃO	12
1. A ENGENHARIA DE REQUISITOS	15
1.1. Requisitos	17
1.2. Gerenciamento De Requisitos	17
1.3. Requisito Funcional	18
1.4. Documentação No Processo De Requisitos	19
1.5. Validação	20
1.6. Controle de qualidade	21
1.7. Gerenciando os requisitos	21
2. DAS TÉCNICAS DE LEVANTAMENTO DE REQUISITOS	24
2.1. Brainstorm	24
2.2. Entrevista	25
2.3. Questionário	25
2.4. Cenário	26
2.5. Caso de uso	27
2.6. Técnica: para obter consenso do escopo	30
2.7. Etnografia	31
2.8. Reunião	32
2.9. Análise Documental	32
2.10. Prototipação	33
2.11. Reutilização de requisitos	34
3. PROCESSO LEGISLATIVO MUNICIPAL	35
3.1. Os três poderes	35
3.1.1. Executivo	35
3.1.2. Judiciário	36
3.1.3. Legislativo	36
3.2. O Poder Legislativo	36
3.2.1. Estrutura da Câmara Municipal	37
3.3. Poder Legislativo Municipal	37
3.3.1. Regimento Interno	37
3.4. Projetos e requerimentos	38
3.4.1. Competência para Proposição	38
3.4.2. Requerimentos	38
3.4.3. Atribuições do Presidente	39

3.4.4.	Aprovação de Projetos e Leis	39
4.	DIAGRAMAS DO PROCESSO DE CONSTRUÇÃO DE SOFTWARE	40
4.1.	Ferramentas usadas	40
4.2.	Abordagem do projeto	41
4.3.	Diagrama de entidade e relacionamento	42
4.4.	Diagrama de classes	43
4.5.	Caso de uso	44
4.5.1.	Diagrama de Caso de Uso	46
5.	TRABALHOS FUTUROS	51
	REFERÊNCIAS	52

INTRODUÇÃO

Com a grande expansão nos últimos anos de desenvolvimento de software, tanto no Brasil quanto no mundo, é possível notar as mais variadas e diversas técnicas adotadas no levantamento de requisitos, pois “o mundo moderno não poderia existir sem o software” (Sommerville, 2011, p. 2). Nesse quesito, a análise de requisitos e o seu levantamento são uma tarefa bastante essencial dentro do processo de criação de um software. Segundo Reinehr (2020, p.13), “é inquestionável a presença da tecnologia em nosso dia a dia. Produtos de software fazem parte de praticamente todas as atividades humanas, sejam pessoais, sejam profissionais”.

Na construção de um software, a especificação de requisitos tem sua grande importância e contribuição no processo de desenvolvimento. Os requisitos podem ser explícitos, normativos e requisitos implícitos. Os implícitos são indesejáveis, pois, não sendo documentados, eles podem ser considerados desenho do produto. Mas, mesmo assim, os requisitos documentados podem apresentar problemas decorrentes de linguagem natural, normalmente usada para expressá-los. Já outros, recorrem a técnicas deficientes nesses levantamentos e especificações (Pádua Filho, 2019).

A fase inicial é uma das etapas mais importantes no desenvolvimento de um sistema. Basicamente, significa entender o que o cliente deseja ou o que ele pensa ser importante, além das regras e processos de negócio. O principal objetivo do levantamento de requisitos é que usuários e desenvolvedores possam ter uma mesma visão do problema a ser resolvido¹

A engenharia de requisitos fornece um mecanismo adequado para entender o que o cliente deseja, analisar as necessidades, avaliar a exequibilidade, negociar uma solução razoável, especificar a solução de maneira não ambígua, validar a especificação e administrar os requisitos à medida que eles são transformados num sistema em operação (Pressman, 2006, p.250).

Neste caso, a engenharia de requisitos vem cumprindo assim sua respectiva atribuição dentro do processo de construção de software. Esse processo de

¹ O que é Levantamento de Requisitos - Tópicos de Engenharia de Software. [S. l.: s. n.], 2018. 1 vídeo (21 min). Bóson Treinamentos. Disponível em: <https://www.youtube.com/watch?v=VcOeM2AD8Yk>. Acesso em 10 mar. 2023.

engenharia de requisitos nos garante que o sistema gerado por esses requisitos possa atender muito bem todas as necessidades, satisfazendo as demais expectativas dos respectivos clientes (Pressman, 2006).

A necessidade que existe em uma engenharia de requisitos é de grande valia para que o processo de construção do software ocorra com toda cautela e sucesso. Isso porque a engenharia de requisitos define o sucesso na construção do software pela elicitación de requisitos. “O levantamento ou elicitación de requisitos de um sistema consiste em uma etapa muito importante no desenvolvimento de projetos” (Amoroso, 2023, S/N).

A tecnologia tornou-se bastante presente em nosso cotidiano; para Reinehr (2020), os softwares hoje em dia passam por grandes problemas devido aos levantamentos de requisitos errôneos, que não foram levantados de acordo com a solução real do problema. Nesta linha de raciocínio, para a autora, a engenharia de requisitos ajuda a minimizar esses problemas, daí sua importância dentro de um projeto de construção de software.

Pressman (1995) coloca que há alguns atributos da análise de sistema, como a identificação das necessidades de determinado usuário, avaliação da qualidade do sistema, a análise econômica, dentre outros. Ele já identificava a análise de requisitos de um software como uma propriedade bastante importante dentro do âmbito do projeto, enfatizando-o na criação do mesmo.

Tal associação tem um grande impacto nos projetos atuais, dentro da análise de requisitos de software e seus processos. “A engenharia de requisitos oferece controles e padrões para que as exigências do projeto sejam claras” (Monitora Team, 2020, S/N). Esta é uma ferramenta essencial na construção de software para estes e outros autores, sendo que a necessidade de aplicação e estudo aprofundado dele pode ser o sucesso do projeto ou não.

Um ponto chave dentro do processo legislativo são as redundâncias de informação que, corriqueiramente, tornam o trabalho repetitivo e abrem margem para a fragilidade dos trabalhos legislativos, tornando a procuradoria jurídica do mesmo refém de correções que tornam lento o processo. Problemas de inconstitucionalidade de projeto podem mostrar a necessidade de filtrar tal problema na fase inicial das pautas.

Com tal premissa, este trabalho visa a corroborar com este tema aprofundando nos levantamentos de requisitos dentro do âmbito de projeto

requerido por modelos definidos e regulamentados, seguindo as normas técnicas de regulamentação. Propõe-se a aplicabilidade das análises de requisitos dentro de um determinado órgão de funcionamento legislativo federal, estadual ou municipal, com a aplicabilidade dessa análise proposta em sua composição.

Ao propor tal tema, temos o enfoque de ressaltar a importância da análise de requisitos no ambiente legislativo, bem como os conceitos práticos e conceituais na elucidação dos requisitos, de maneira que seja uma bússola que aponte para uma direção necessária para levantamento dele.

Propõe-se ao mesmo projeto que seja realizado com as devidas cautelas e cuidados dentro de um projeto real para uso futuros, tendo em vista que o órgão legislativo possui suas necessidades tecnológicas de software que ajudariam no processo de digitalização do mesmo. Tal processo melhoraria a transparência do processo legislativo. Isso ocorreria desde os requerimentos de entradas quanto às ordens de serviços e prestação de contas do legislativo à população, passando pela votação em plenário e requerimentos aprovados, votados e os não votados em sessão, projetos de lei aceitos e demandas do executivo para o legislativo.

Tal processo mostrará a transparência do mesmo, e o acompanhamento por parte da população pelos projetos e análise das pautas e requerimentos em sessão ordinária em execução. Também viabiliza o processo de envio do mesmo ao jurídico da Casa, possibilitando e digitalizando o processo democrático em questão. A usabilidade é garantir o processo e melhorar a transparência do mesmo.

Sendo assim, qualquer pessoa, dentro ou fora do âmbito da tecnologia, independentemente de seu nível de abstração, conseguirá verificar as informações do legislativo e acompanhar as demandas do candidato (e dos demais) em que votou. O mesmo processo contará com as entrevistas visando ao procedimento do mesmo e analisando os passos dentro do processo legislativo atual.

Organiza-se o trabalho de conclusão de curso da seguinte forma: o capítulo um trata do conceito de engenharia de requisitos; o capítulo dois discorre sobre o uso de técnicas para o levantamento de requisitos; o capítulo três aborda a abertura de um parêntese, a explicação do Processo Legislativo Municipal; já o capítulo quatro debate sobre diagramas de caso, de uso e entidade e relacionamento do projeto.

A ENGENHARIA DE REQUISITOS

A análise de requisitos define o escopo num projeto; ela funciona como uma bússola apontando a direção a seguir. Essa definição é necessária antes de se iniciar a construção do software, para entender tal importância e complexidade do mesmo.

Deve haver muita comunicação para que não gere redundâncias de informação, e para que não haja interpretações errôneas ou falsas informações (Pressman, 2006). Para Reinehr (2020), dentre os vários problemas relacionados aos requisitos, tem-se:

- Estar incompleto;
- Níveis insuficientes dos detalhes para etapas de ciclo de desenvolvimento;
- Estar incompatível com outros requisitos;
- Inviável tecnicamente;
- Testes e validação com mais dificuldades;
- Serem conflitantes na visão dos stakeholders.

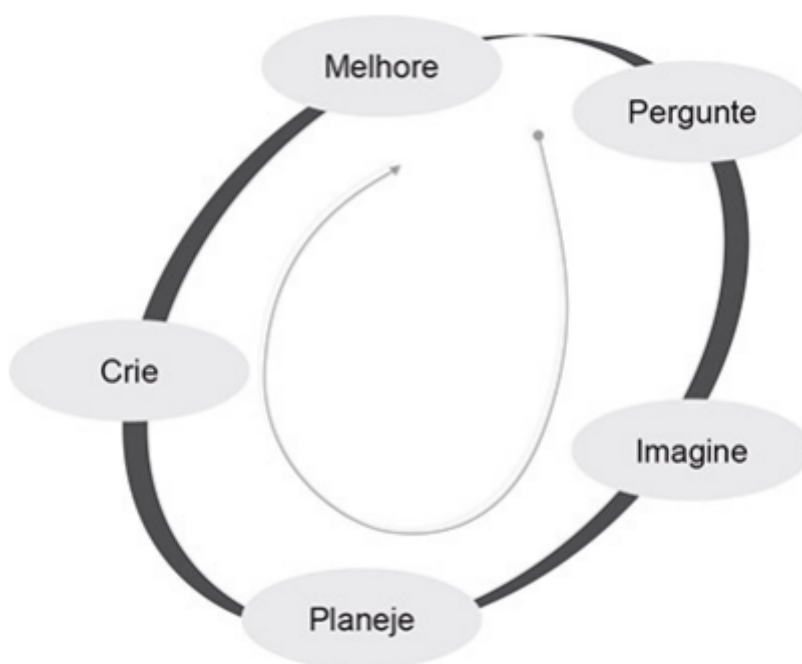
Tais problemas podem acontecer dentro da engenharia de requisitos, variando de acordo com projetos. A engenharia viabiliza o processo de especificação da função do software; assim, “a análise de requisitos proporciona ao projetista uma representação da informação e da função” (Pressman, 2006, p.232).

Os requisitos são obtidos diretamente com o cliente ou usuários, e, caso não haja uma comunicação correta, problemas podem vir à tona nos requisitos e construção do software. Reinehr (2020, p. 17), nos afirma que “é comum ouvir que os usuários não sabem o que querem, mas, na verdade, por mais que eles saibam, sua perspectiva pode mudar pelo simples fato de entrarem em contato com a equipe”.

Isso significa que usuários podem sim ter uma visão de requisitos que pode ser alterada no meio desse levantamento. Geralmente, temos que guiar o usuário para conduzi-lo para onde ele quer chegar. A engenharia de requisitos está, de certa maneira, unida na aquisição de conhecimento para sua criação, também com aperfeiçoamento e implementações dele.

Vazquez e Simões (2016) definem a primeira etapa do processo em buscar e identificar problemas para resolvê-los. Desenhos de diagramas são essenciais nesse processo, como abaixo, em um diagrama onde é retratado um determinado fluxograma no processo de coletas de informações.

Figura 01: Diagrama de planejamento



Fonte: Vazquez e Moura (2016, p.29)

Analisando tal diagrama, vemos o início de um determinado ciclo a partir de perguntas para coleta de informações dentro do processo de construção de software. Logo em seguida, o próximo procedimento nessa entrevista é um fluxo contínuo. Nesse sentido, para Machado (2014), os requisitos são o início na construção de software, tornando-se essenciais dentro deste processo.

Segundo defende Freitas,

Considerando-se os requisitos como o alicerce para a construção do software em questão, é preciso garantir que eles representem efetivamente as necessidades e acordos realizados para que o seu desenvolvimento seja seguro e com qualidade (Freitas, 2006, p. 11 e 12).

Iniciar um projeto tão grande sem um alicerce o torna vulnerável a erros dentro do processo de construção, tornando-se base para ele. Esse é o marco zero na construção em uma jornada que requer atenção, e o mínimo de erros na coleta dos requisitos mediante o usuário.

1.1. Requisitos

Os requisitos, de maneira bem geral, são basicamente divididos em requisitos funcionais, que nos definem o que um software irá fazer; grosso modo, estão ligados à sua função de “o quê”. Há, por outro lado, o requisito não funcional, que define as restrições do mesmo no ambiente de uso para o usuário, ligado sob restrições; por isso, é um requisito não funcional (Valente, 2010).

Tal definição nos demonstra a grande importância relativa que é o tema abordado. Falar de requisitos é de tamanha relevância dentro da análise de requisitos que se torna uma parte capaz de definir o sucesso e a abordagem do software. Um software sem o devido levantamento de requisitos torna -se obsoleto.

Para Soares e Zanin (2010), o software precisa de recursos para que suas funções possam operar pelo dispositivo físico. Neste caso, será necessário definir qual a restrição e o limite de alcance que poderá ter. Tal requisito, não lista apenas a sua funcionalidade, mas pode nos trazer sua especificação sobre como agir; assim, o objetivo será alcançado.

Segundo Gallotti (2016, p. 122), “especificar requisitos é a etapa em que escrevemos os requisitos de usuário e de sistemas no documento de requisitos”. Isso, em tese, com os mesmos claros e objetivos precisos, também devem estar completos, consistentes, porém isso não é tão simples.

1.2. Gerenciamento De Requisitos

Em um projeto de desenvolvimento de software, a produção de informações é ampla:

- Para Pádua Filho (2019), em projetos complexos, necessita-se de um maior investimento na engenharia de requisitos. Ela se torna bastante complexa em novos produtos, e, em casos de versões anteriores, o usuário pode, pela sua experiência, identificar as necessidades mais prioritárias.
- O software deve atender às necessidades do cliente: neste caso, é o resultado fazendo sua relação custo/benefício. Para Hirama (2012), leva-se em consideração o custo e o prazo. É necessário que haja um gerenciamento desses requisitos desde sua especificação até a entrega do software ao

cliente. O gerenciamento de requisitos trata das entregas e das mudanças nesses requisitos.

- Os requisitos devem estar bem identificados para que suas atividades tenham sucesso (Reinehr, 2020).
- A engenharia de requisitos torna a relação cliente x software facilitada pela sua interação; neste caso, irá entender suas necessidades e identificá-las como parte da solução que será entregue (Simões e Vazquez, 2016).

1.3. Requisito Funcional

O requisito dentro de uma construção de software é de extrema importância; é uma necessidade de, pelo requisito, definir todos os detalhes na construção do software, pois isso definirá sua funcionalidade. Ao falarmos de requisito funcional, dizemos à sua funcionalidade no que irá atender ou realizar (Valente, 2016).

Kerr (2015) compreende os requisitos funcionais como definições sobre funcionalidades; tal definição é fornecida pelo sistema, mostrando-se o que se deve fazer. Para Rezende (2005, p.122), “a relação dos requisitos funcionais é definida para determinação do tempo”. Neste caso, sua distribuição se dá por tipos: telas, arquivos, relatórios, processamento e testes. O autor continua na relação desses requisitos:

- Planejar;
- Levantar as necessidades;
- Determinar os requisitos funcionais;
- Elaborar a macro proposta;
- Relatar requisitos funcionais;
- Elaborar os diagramas;
- Descrever a lógica dos requisitos funcionais;
- Desenhar as telas, relatórios e arquivos;
- Análise de integrações;
- Definição da tecnologia e elaboração de plano logístico;
- Análise de custos, benefícios e viabilidade;
- Definir os procedimentos de segurança;
- Documentar o manual para o cliente e o manual técnico;

- Planejar a implementação, treinamento do cliente;
- Finalizar e acompanhar pós-implementação;

Figura 2. Tabela de tempos para desenvolvimento

1. Criação de Arquivos	DC	DNC	MAN
<i>Cadastro</i>			
- arquivo físico pequeno	0,06	0,07	0,06
<i>Movimento</i>			
- arquivo físico grande	0,15	0,20	0,15
2. Definição de Telas			
<i>Lista</i>			
- tela média	0,15	0,20	0,15
<i>Detalhe</i>			
- tela grande	0,25	0,40	0,25

Fonte: Rezende (2006, p. 86)

Como nesta tabela acima, o autor sugere que deve ser atribuído um determinado tempo em minutos na elaboração, neste caso, o DC é o desenvolvimento conhecido, o DNC é o desenvolvimento não conhecido e o MAN é a manutenção de sistemas. Ela mostra alguns dados gerados no tempo pré-definido anteriormente.

1.4. Documentação No Processo De Requisitos

Há uma variedade de possibilidades nesta fase do projeto. Ledur (2017) nos diz que antes que seja iniciado o projeto é necessário estar definido o escopo que será criado. Nesta definição, são levantados alguns diagramas de casos de uso, em que as técnicas abordadas pelo gerente de projeto seguem perguntas cruciais na sua definição. Perguntas como: Quem? O quê? Quando? Como? Onde? Por quê? Essas devem ser palavras chaves para direcionar para um escopo bem elaborado e definido.

Com estas perguntas realizadas a respeito do sistema, dificilmente ficarão dúvidas quanto ao escopo do projeto, já que elas tentam induzir o gerente de projetos a pensar em todo o processo de uso e interação que o software terá com o mundo real. Por exemplo, veja no Quadro 1 como são feitos os questionamentos para fazer o levantamento de escopo de uma aplicação real para celulares (Ledur, 2017, p. 40).

No quadro abaixo, o autor demonstra bem sua ideia:

Figura 3: Quadro de perguntas do usuário para escopo

Quem?	O quê?	Quando?	Como?	Onde?	Por quê?
Usuário	Escolhe uma aplicação para rodar	A qualquer momento	Através de uma interface de seleção de aplicações	No celular (ou outro dispositivo móvel)	Porque deseja executar a aplicação
Usuário	Executa uma aplicação	Quando a aplicação estiver rodando	Preenchendo formulários, navegando entre as páginas da aplicação, enviando informação...	No celular (ou outro dispositivo móvel)	Para utilizar a aplicação disponibilizada

Fonte: Ledur (2017, p.40)

Segundo consta na Norma IEEE (1996), que é a prática recomendada para especificações de exigências de software, há uma descrição o conteúdo e a qualidade de uma boa especificação de exigências de software. Nele, são descritas as abordagens para uma especificação de requisitos de software, que são exigências deste mesmo software.

Em relação à natureza do documento, quem o descreve deve levar em consideração pontos bases como funcionalidades, interfaces externas, performance, atributos e restrições. Seu propósito, harmoniza a definição de conteúdo para resultados do processo de ciclo de vida do software entre padrões de engenharia de software IEEE.

1.5. Validação

Dentro da validação de requisitos, a consistência tem sido de fundamental preocupação no decorrer da elucidação de requisitos. Tais artefatos, que são produzidos nesse processo da engenharia de requisitos, são avaliados ao longo do processo, que é comumente chamado de validação de requisitos (Pressman e Maxim, 2021).

Para Vetorazzo (2018), produtos defeituosos são evitados pelo controle de qualidade, e esse detalhe evita que o software seja entregue de maneira defeituosa ao cliente. A garantia visa ao gerenciamento e melhoria dos processos existentes, onde as inspeções são realizadas na etapa de validação, evitando, nos testes, que o software seja entregue com erros.

[...]as inspeções verificam a conformidade com uma especificação do software, mas não verificam a não conformidade com os requisitos reais do cliente e nem características não funcionais, como por exemplo, desempenho, usabilidade e adaptabilidade (Vetorazzo, 2018, p. 77).

O teste de software e a depuração são fatores de atividades relacionados intrinsecamente à validação do software. Neste caso, podemos distingui-las com algumas perguntas: a primeira é se o produto a ser construído é o correto; neste caso, estamos tratando com validação. E a segunda pergunta é se o produto está correto quanto a sua construção - esta é ligada à verificação (Inthurn, 2001).

1.6. Controle de qualidade

Para Staa (1983), o controle de qualidade faz o papel de operar o detalhe em sentido geral. Primeiramente, as métricas serão obtidas para apenas depois quantificar os diversos fatores pelas funções normalistas. Neste caso, poderá haver rejeições nas ocasiões dentro do controle de qualidade - para este caso, somente se os controles caírem abaixo dos limites tolerados.

De acordo com Inthurn (2001, p.27), “a tarefa de revisar a especificação é algo muito importante para que falhas sejam detectadas o mais breve possível”. Para a autora, tal trabalho requererá uma dedicação consistente à equipe desenvolvedora do projeto, para que o resultado seja satisfatório e, logo no início, sejam verificadas as vantagens no desenvolvimento do sistema.

1.7. Gerenciando os requisitos

Dentro de um projeto, podemos destacar a importância de identificar e gerenciar os requisitos, definir seu gerenciamento e suas mudanças, definindo neste processo as ferramentas necessárias para a gestão dele. Gerenciar requisitos

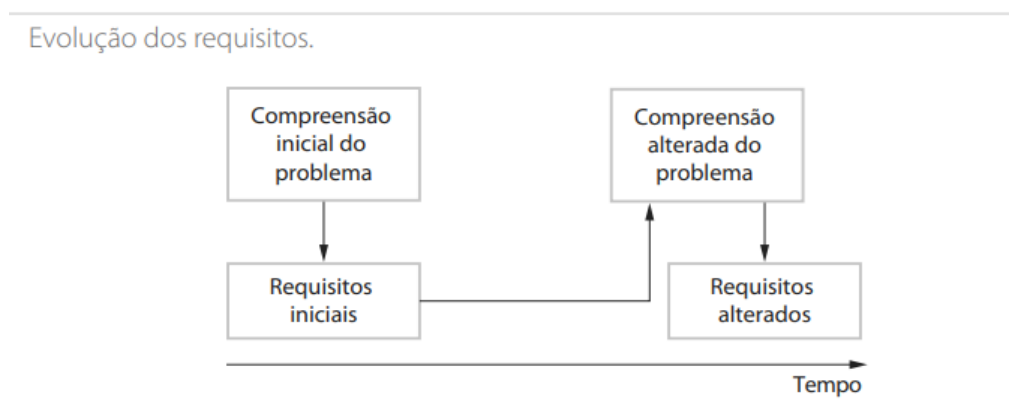
torna-se atrativo ao requerer alguns desafios e competências técnicas da equipe, tornando-o um recurso muito importante para o processo de construção do software.

Para Sommerville (2011, p.77), “uma vez que um sistema tenha sido instalado e seja usado regularmente, inevitavelmente surgirão novos requisitos [...] quando usuários finais tiverem a experiência de um sistema descobrirão novas prioridades”. Para o autor, fica evidente a tese de que existem várias razões pelas quais as mudanças são inevitáveis.

Segundo Leffingwell e Widrig, (2003, apud Ledur, 2017, p.174), a pergunta que fica é: “porque desenvolvedores falham em construir a coisa certa?”. Tal pergunta ecoa para os desenvolvedores e equipes de projeto. Segundo os autores, a concepção de novos sistemas gera grande parte dessas falhas ou um gerenciamento ruim desses sistemas de produtos.

Tais requisitos possuem um ciclo de vida definido. É a atividade de elucidação que faz esses ciclos nascerem. Os mesmos serão validados e analisados e somente depois estarão prontos a serem codificados. A partir daí, o requisito é parte fixa dentro do produto, exceto que não seja exigida mais dentro do software (Reinehr, 2020). Logo, abaixo veremos um gráfico da evolução dos requisitos para melhor elucidar o tema.

Figura 4: Evolução dos requisitos



Fonte: Sommerville (2011, p. 78)

Para Sommerville (2011), há três estágios principais em um processo de gerenciamento de mudanças, que são:

- Análise de problema e especificação de mudanças. Nesta fase é inicializado com a identificação dos problemas de requisitos. Neste estágio os problemas ou as propostas serão analisados para verificação de validade.
- Análise de mudanças e custos. Todas as mudanças serão avaliadas por informações de rastreabilidade do sistema, bem como conhecimentos gerais dos requisitos.
- Implementação de mudanças. Organiza - se os documentos de requisitos quando necessário com alterações que não haja reformulações.

2. DAS TÉCNICAS DE LEVANTAMENTO DE REQUISITOS

No último capítulo, vimos sobre a engenharia de requisitos; nele, vimos a sua importância e necessidade dentro do escopo do projeto. Observamos seus fundamentos, os ciclos de planejamento, o que são os requisitos, seu gerenciamento e uma diferença entre funcionais e não funcionais. Vimos também o processo de documentação dessa etapa no projeto, sua validação e controle de qualidade.

Tivemos uma abordagem introdutória sobre o gerenciamento de requisitos, onde pudemos ver os três estágios de mudança de gerenciamento, seguindo a análise de Sommerville (2011). Nele, dispomos a evolução dos requisitos para melhor elucidar o tópico abordado.

Outro tema importante referente ao capítulo anterior é a validação dos requisitos, tema abordado como um aspecto da elucidação de requisitos, dando ênfase nos testes de software como instrumento dessa validação, que está ligada à verificação.

Nas próximas seções, mostrar-se-ão algumas das técnicas de levantamentos de requisitos.

2.1. Brainstorm

Brainstorm é uma das técnicas de levantamentos de requisitos que possui um conceito em determinado conjunto de ideias. Tal técnica se vê desprovida de julgamentos ou até mesmo análises. Ela é ideal para a busca de novas ideias e produtos (Zanchett, 2015).

Para Reinehr (2020), esta técnica é um conglomerado de ideias chamado de “tempestade de idéias” pela autora, em que não há condições analíticas e julgamentais, ocorrendo em um ambiente totalmente informatizado.

Nessa técnica, há uma reunião entre membros da equipe; neste caso, desenvolvedores serão requisitados a fazerem um brainstorm dos grupos de tarefas detalhadamente, visando a sua meta. Alguns desses exemplos são: tarefas de banco de dados, tarefas de integração de sistemas externos, tarefas de desenvolvimento orientado a objetos da camada de domínio etc. (Larman, 2007).

2.2. Entrevista

Sommerville (2011) descreve entrevistas e questionários como as técnicas mais usadas na análise de processos. Esta técnica de entrevista pode ser realizada de maneira formal ou informal com *stakeholders*². Estes são questionados sobre o sistema usado e sobre o próximo sistema a ser desenvolvido. Os requisitos podem surgir a partir de respostas às perguntas; neste caso, podemos ter dois tipos de entrevistas: fechada e aberta (Sommerville, 2011).

A diferença para as duas é que na entrevista fechada haverá um conjunto de perguntas pré-definidas pela equipe de análise de requisitos. Para Sommerville (2011, p. 72), “entrevistas abertas, em que não existe uma agenda pré-definida. A equipe de engenharia de requisitos explora uma série de questões com os stakeholders do sistema e, assim, desenvolvem uma melhor compreensão de suas necessidades”.

2.3. Questionário

O questionário é descrito com uma das duas principais técnicas, assim como a entrevista (Sommerville, 2011). Para Zanchett (2015, p.70), “questionários visam extrair respostas através de questões subjetivas e objetivas “. O questionário é na realidade um grande instrumento que possibilita a coleta de dados quando o grupo é muito grande, ou quando cada integrante mora geograficamente distante (Reinehr, 2020).

Para a autora, há alguns passos que necessitam a serem seguidos viabilizando a aplicação da técnica.

Figura 5: Passos para a utilização da técnica de questionário

²Acionistas, investidores, gestores e todos envolvidos no desenvolvimento do projeto (Pressman e Maxim, 2021, p. 492).



Fonte: Reinehr (2020, p. 76).

Em seguida, é necessário identificar o público-alvo da pesquisa. Aplicar o questionário para o público-alvo pode ser constituído de membros internos ou externos à organização. As questões serão elaboradas em função do perfil desse público e será definida a forma mais adequada de aplicação (Reinehr, 2020, p. 76).

2.4. Cenário

Descrições abstratas tornam os exemplos mais difíceis na descrição. Pessoas preferem se relacionar com exemplos reais do cotidiano. Os cenários são bastante eficazes para agregar detalhes em uma determinada descrição dos requisitos (Sommerville, 2007).

A elicitación baseada em cenários envolve o trabalho com os stakeholders para identificar cenários e capturar detalhes que serão incluídos nesses cenários. Os cenários podem ser escritos como texto, suplementados por diagramas, telas etc. Outra possibilidade é uma abordagem mais estruturada, em que cenários de eventos ou casos de uso podem ser usados (Sommerville, 2007, p. 73).

Para Kouri (2007, apud Leite, 2000, p. 44), há uma estrutura que é definida na descrição de cenário, que contém os seguintes atributos.

- Título: define ou identifica o cenário.
- Objetivo: estabelece finalidades e o modo como este objeto deve ser alcançado.

- Contexto: é a descrição de um estado inicial do cenário. Nele, há três ênfases: tempo, local e pré-condições.
- Recursos: mostram os objetos passivos que passam os atores.
- Atores: são pessoas ou organizações que possuem vínculo de ação com o cenário.
- Episódios: são ações realizadas pelo ator, que podem participar de mais atores tendo utilizado recursos ao seu dispor.
- Exceções: tratamento de erro ou situação fora do habitual comum.

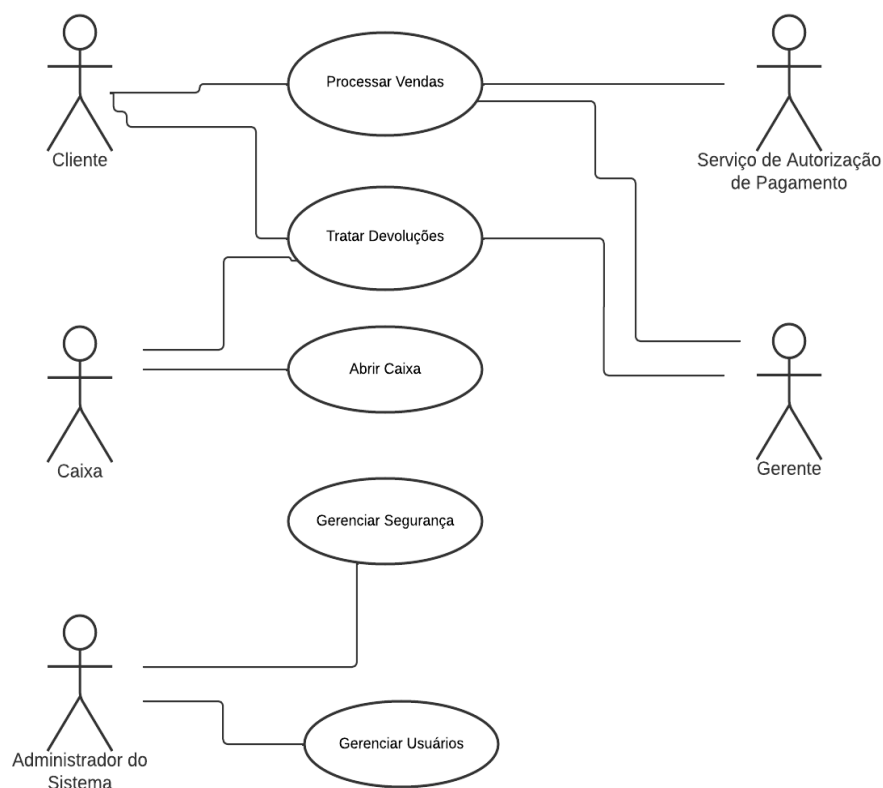
2.5. Caso de uso

Dentre muitas técnicas no processo de desenvolvimento de software, destacamos o caso de uso. Se pudéssemos descrevê-lo, seria, como um conto, uma jornada de interação entre o usuário do sistema e um conjunto de circunstâncias possíveis (Pressman e Maxim, 2021). Pressman e Maxim descrevem algumas características de sua sintaxe, como vemos a seguir:

1. Quem são atores primários e secundários.
2. As metas do ator.
3. Condições para que aconteça.
4. As tarefas e funções realizadas pelo ator.
5. Exceções ao longo do caminho de percurso.
6. Possíveis variações na interação do ator.
7. Informações adquiridas, produzidas e modificadas pelo ator.
8. Mudanças externas ao ambiente e à informação obtida e informada pelo ator.
9. Informações desejadas pelo ator no sistema.
10. Mudanças inesperadas e se o ator quer ser informado sobre.

Para Fowler (2005, p.104), “os casos de uso são uma técnica para capturar os requisitos funcionais de um sistema. Eles servem para descrever as interações típicas entre os usuários de um sistema[...]”. Para ele, é mais fácil descrever os cenários ao invés de iniciar com o caso de uso. Os cenários, segundo Fowler (2005), são sequências de passos descritos na interação entre usuários e sistemas.

Figura 6: Diagrama de casos de uso



Fonte: Larman (2007, p.116)

A figura 6 nos mostra um exemplo de diagrama de caso de uso, onde existem alguns atores envolvidos que são: cliente, caixa, administrador do sistema, gerente, serviço de autorização de pagamento. Todos se relacionam com o sistema; alguns com permissões distintas de outros.

Um cliente interage com o sistema da loja ao processar a venda, e o serviço de autorização de pagamento interage autorizando ou não esse processamento. Ao mesmo tempo, o cliente pode tratar devoluções e o caixa interage no mesmo intuito do tratamento dessas devoluções.

O gerente pode interagir tanto no tratamento das devoluções quanto em processar as vendas, porém, somente o caixa pode interagir abrindo o caixa. Já o administrador interage tanto no gerenciamento da segurança quanto de usuários do sistema.

Os casos de uso, na realidade, são textos narrativos de um determinado sistema, que visa atingir os objetivos (Larman, 2007). Agora, vejamos um exemplo de caso de uso em formato textual na figura abaixo.

Figura 7: Descrição contínua

Processar Venda: um cliente chega em um ponto de pagamento com itens que deseja adquirir. O caixa usa o sistema PDV para registrar cada item comprado. O sistema apresenta um total parcial e detalhes de linha de item. O cliente entra os dados sobre o pagamento, que são validados e, em seguida, registrados pelo sistema. O sistema atualiza o estoque. O cliente recebe um recibo do sistema e sai com os itens comprados.

Fonte: Larman (2007, p. 89)

Vejamos abaixo um modelo de um caso de uso definido por Larman (2007, p. 86):

Figura 8: Caso de uso - processar venda

Caso de uso CDU1: processar venda

Escopo: Aplicação PDV ProxGer

Nível: objetivo do usuário

Ator Principal: Caixa

Interessados e Interesses:

- Caixa: deseja entrada de pagamento rápida, precisa e sem erros, pois a falta de dinheiro na gaveta do caixa será deduzida do seu salário.
- Vendedor: deseja comissões sobre vendas atualizadas.
- Cliente: deseja comprar, receber um serviço rápido e com o mínimo esforço. Deseja a exibição, facilmente visível, dos itens e preços inseridos. Deseja um comprovante da compra, necessário no caso de devoluções de mercadorias.
- Empresa: deseja registrar precisamente as transações e satisfazer aos interesses do cliente. Quer garantir que os pagamentos a receber do Serviço de Autorização de Pagamentos sejam registrados. Deseja algum tipo de proteção contra falhas para permitir que as vendas sejam capturadas mesmo se os componentes do servidor (por exemplo, validação remota de crédito) se encontrarem indisponíveis. Deseja uma atualização automática e rápida da contabilidade e do estoque.
- Gerente: deseja poder realizar rapidamente operações de correção e facilmente corrigir os problemas do Caixa.
- Órgãos fiscais governamentais: desejam cobrar os impostos de cada venda. Podem estar envolvidos vários órgãos, como, por exemplo, federais, estaduais e municipais.
- Serviço de autorização de pagamentos: deseja receber solicitações de autorização digital no formato e protocolo corretos. Deseja contabilizar com precisão seus débitos a pagar para a loja.

Pré-Condições: Caixa está identificado e autenticado.

Garantia de Sucesso (ou Pós-Condições): Venda foi salva. Impostos foram corretamente calculados. Contabilidade e Estoque foram atualizados. Comissões foram registradas. Recibo foi gerado. Autorizações de pagamento foram registradas.

Cenário de Sucesso Principal (ou Fluxo Básico):

1. Cliente chega à saída do PDV com bens ou serviços para adquirir.
2. Caixa começa uma nova venda.
3. Caixa insere o identificador do item.
4. Sistema registra a linha de item da venda e apresenta uma descrição do item, seu preço e total parcial da venda. Preço calculado segundo um conjunto de regras de preços. Caixa repete os passos 3 e 4 até que indique ter terminado.
5. Sistema apresenta o total com impostos calculados.

Fonte: Larman (2007, p.94)

Figura 9: Caso de uso processar venda (continuação)

<p>6. Caixa informa total ao Cliente e solicita pagamento.</p> <p>7. Cliente paga e Sistema trata pagamento.</p> <p>8. Sistema registra venda completada e envia informações de venda e pagamento para Sistema externo de contabilidade (para contabilidade e comissões) e para Sistema de Estoque (para atualizar o estoque).</p> <p>9. Sistema apresenta recibo.</p> <p>10. Cliente vai embora com recibo e mercadorias (se houver).</p> <p>Extensões (ou Fluxos Alternativos):</p> <p>*a. A qualquer momento, Gerente solicita uma operação de correção:</p> <ol style="list-style-type: none"> 1. Sistema entra no modo Autorizado pelo Gerente. 2. Gerente ou Caixa realiza uma das operações do modo Gerente, por exemplo, modificação do saldo em dinheiro, retoma uma venda suspensa em outro registrador, anula uma venda, etc. 3. Sistema reverte para o modo Autorizado pelo Caixa. <p>*b. A qualquer momento, Sistema falha:</p> <p>Para fornecer suporte à recuperação e à correta contabilidade, garanta que todos os estados e os eventos sensíveis das transações possam ser recuperados a partir de qualquer passo do cenário.</p> <ol style="list-style-type: none"> 1. Caixa reinicia Sistema, registra-se e solicita a recuperação do estado anterior. 2. Sistema restaura estado anterior. <p>2a. Sistema detecta anomalias que impedem a restauração:</p> <ol style="list-style-type: none"> 1. Sistema avisa Caixa sobre erro, registra o erro e, então, entra em um novo estado consistente. 2. Caixa começa uma nova venda. <p>3a. Cliente ou Gerente indica a retomada de uma venda suspensa.</p> <ol style="list-style-type: none"> 1. Caixa realiza a operação retomada e insere a Identidade para recuperar a venda. 2. Sistema mostra o estado da venda retomada, com subtotal. <p>2a. Venda não encontrada</p> <ol style="list-style-type: none"> 1. Sistema avisa Caixa sobre o erro. 2. Caixa, provavelmente, começa nova venda e re-insere todos os itens. <p>3. Caixa continua a venda (provavelmente inserindo mais itens ou tratando o pagamento).</p> <p>2-4a. Cliente diz ao Caixa que tem uma condição de isenção de imposto (por exemplo, idoso, cidadão local)</p> <ol style="list-style-type: none"> 1. Caixa verifica e depois insere o código de condição de isenção de imposto.

Fonte: Larman (2007, p.95)

2.6. Técnica: para obter consenso do escopo

Apresentaremos a seguir técnicas que visam atingir o escopo pela consolidação, dentre elas vemos:

- Técnica: Declaração de problema.
- Modelagem de escopo (Modelagem Ambiental).

Com o Canvas (*Business Model Canvas*), descreve-se um determinado modelo de negócio. Apesar de ser uma ferramenta para o empreendedorismo, ela permite descrever, inventar e alavancar tal modelo de negócio. Neste caso, ela ajuda a descrever a fala do cliente (Vasquez e Simões, 2016).

Para os Vasquez e Simões (2016, p. 197), “o objetivo da modelagem de escopo é identificar limites apropriados para o sistema de desenvolvimento [...] indicando basicamente o que é interno e externo ao sistema”. É o que os autores chamam de modelagem ambiental.

Figura 10: Técnica consenso no escopo

Problema	O planejamento e o controle no atendimento ambulatorial em consultórios e clínicas médicas são feitos manualmente, o que os torna ineficientes e muito sujeitos a erros.
Afeta	Clientes, médicos e atendentes.
Impacto	Os clientes aguardam mais tempo que o necessário e têm com frequência suas consultas canceladas porque um mesmo horário com um determinado médico foi agendado para duas ou mais pessoas. O médico tem seu tempo subutilizado e o fluxo de caixa é prejudicado por problemas operacionais no faturamento junto aos planos de saúde. Os atendentes têm duplicidade na execução de tarefas em função de erros e há um maior investimento de tempo em seu treinamento, já que não há um sistema que os oriente nos processos sob a sua responsabilidade.
Uma solução para ser bem-sucedida deve	Impedir a sobreposição de agendas; apontar a identificação de horários disponíveis em tempo hábil para procurar eliminá-los e conforme um plano geral de atendimento definido para cada profissional; e propor agendas alternativas para utilização da disponibilidade de maneira proativa por parte do atendente. Reduzir a glosa na documentação enviada para faturamento contra as operadoras de planos de saúde ao patamar de 10% do volume total de guias.

Fonte: (Vazquez e Simões, 2016, p. 196)

2.7. Etnografia

Dentro da observação, temos a etnografia; segundo Moraes e Zanin (2017), trata-se de uma técnica de observação, neste caso, usada com o intuito de compreender os processos operacionais que podem ajudar na extração de dados e/ou requisitos nos processos, os requisitos de apoio. Segundo os autores, há diversas observações no trabalho diário, onde possuem anotações das tarefas nas quais os participantes estão diretamente envolvidos.

A compreensão do processo torna-se importante para que seja bem-sucedida; devido isso, as diversas observações dos trabalhos diários são eficazes para lidar com os imprevistos no processo de construção com suas tratativas e soluções abordadas no decorrer desse processo. Para Sommerville (2018), a etnografia é muito eficaz, principalmente para descobrir os requisitos:

- Requisitos que derivam de como realmente trabalham as pessoas. Pois as definições de processos fazem com que elas trabalhem de maneira diferente do habitual.

- Requisitos que derivam de conhecimento e cooperação de atividades dos envolvidos.

2.8. Reunião

Uma forma bastante usual para a elucidação de requisitos é a reunião. Para Reinehr (2020), nada mais é do que troca de informações e cultura pelos meios de comunicação; essas informações podem exprimir mudanças profundas no espaço e na construção durante esta etapa.

A reunião exige uma certa preparação por parte de quem a conduzirá, devendo constar na agenda os tópicos abordados. O tempo de cada assunto com um ambiente amplo, ventilado e iluminado - nunca se esquecendo dos slides, quadro branco, projetor ou computador como apoio complementar. Além disso, há três momentos que são: abertura, abordagem de tópicos e conclusão (Reinehr, 2020).

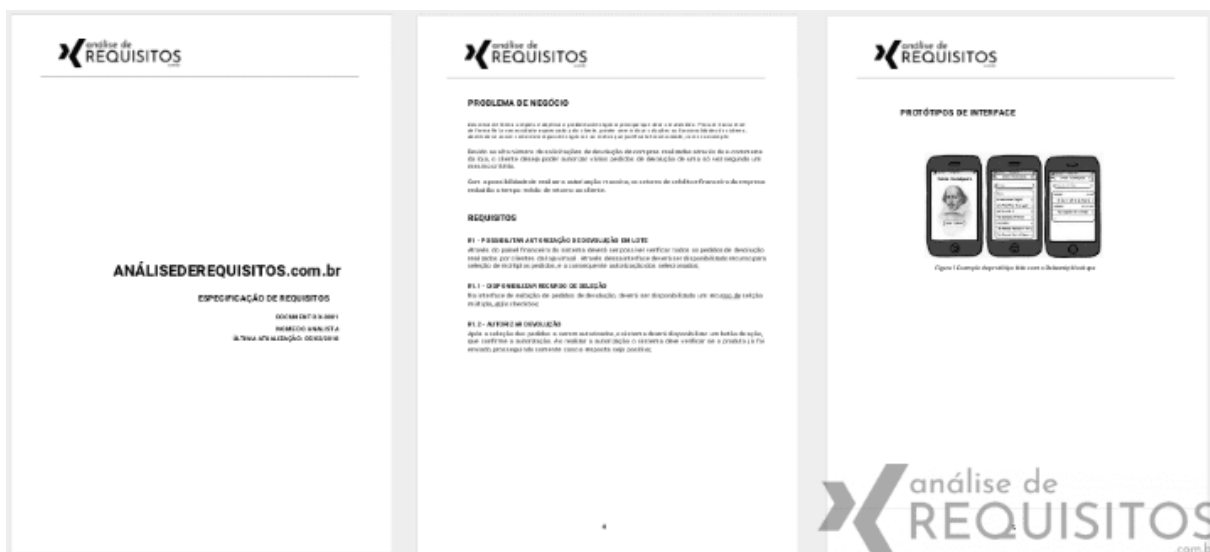
2.9. Análise Documental

Analisar documentos é também uma maneira de fazer a elucidação dos requisitos através dos estudos nos quais a documentação de determinada solução que exista na identificação dessa informação. Este estudo deve estar disponível para que a informação seja relevante no desenvolvimento da solução (Vasquez e Simões, 2016).

Uma das funções dessa técnica é documentar os requisitos funcionais e não funcionais de um projeto de software, tornando a documentação necessária e importante no processo e fase da elicitação dos requisitos. Alff (2018), em seu artigo de análise de requisitos, nos mostra o quão importante é nessa etapa de projeto e construção.

Para Alff (2018), o documento de requisitos é utilizado no processo, e documentar esses requisitos junto aos stakeholders é uma diretiva; além do mais, teria ampla rastreabilidade como matriz. Esta documentação é essencial e importante no ciclo de vida do projeto. Abaixo há um exemplo desse modelo.

Figura 11: Modelo de análise documental



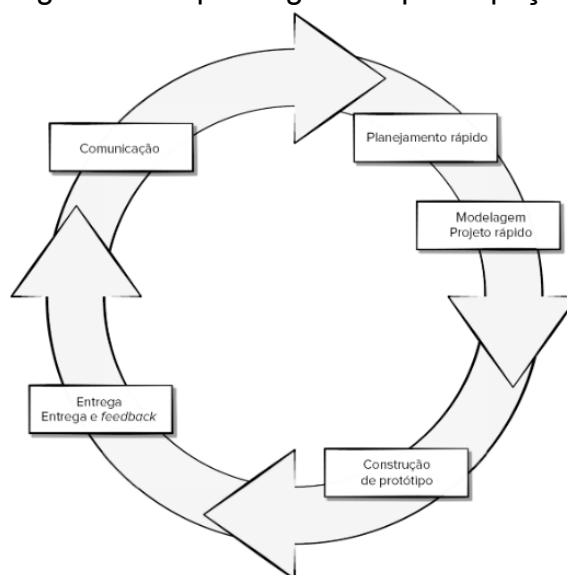
Fonte: Alff (2018)³

2.10. Prototipação

Por mais que seja utilizada isoladamente como modelo de processo, é comumente utilizada como uma técnica de levantamento de requisitos. O paradigma da prototipação pode auxiliar a equipe desenvolvedora na melhor compreensão da construção de software. Isso se aplica principalmente quando os requisitos estão obscuros (Pressman e Maxim, 2021). Abaixo, vemos uma figura que retrata melhor esta abordagem:

³ Imagem retirada do site Análise de Requisitos, de um texto publicado por Francilvio Roberto Alff, em fevereiro de 2018. Disponível em: [Documentode-requisitos-de-software \(analisederequisitos.com.br\)](http://documentode-requisitos-de-software.analisederequisitos.com.br). Acesso em: 11 de nov. de 2023.

Figura 12: O paradigma da prototipação



Fonte: (Pressman e Maxim, 2021, p. 27)

Um protótipo de software tem por objetivo a permissão das avaliações propostas de usuários para desenvolvedores sobre o projeto final; nesse caso, o software com testes ao invés das interpretações, bem como sua avaliação baseada na descrição (Ledur, 2017).

A prototipação pode iniciar a proposta de interface gráfica do software em relação ao usuário. Tal técnica permite que o usuário perceba o funcionamento dos requisitos pela simulação antes que o software seja lançado em sua versão final, fazendo que seja analisado se sua necessidade está sendo atendida ou não nos requisitos do mesmo (Vasquez e Simões, 2016).

2.11. Reutilização de requisitos

É importante entendermos que a reutilização de requisitos pode ser comumente confundida com reuso de software. Porém, não são a mesma coisa. A reutilização de requisitos está diretamente ligada ao fato de requisitos previamente documentados em outros projetos de construção de software, que, com algumas mudanças ou adaptações, possam ser reutilizados em síntese no novo projeto.

3. PROCESSO LEGISLATIVO MUNICIPAL

Até aqui temos visto assuntos relacionados à análise de requisitos para construção de software, visto que tal tema é a base do trabalho proposto para o objetivo de estudo. As devidas abordagens das técnicas de levantamentos de requisitos que foram tema central do último capítulo, com alguns tópicos abrangentes ao referido.

Dos capítulos iniciais ao último, tudo foi relacionado à importância das análises de requisitos na construção de software, com abordagens e explanação teórico-científica para clarear o entendimento do leitor, independentemente de seu grau acadêmico sobre o mesmo, para uma melhor compreensão.

Abrimos agora um parêntese necessário antes de recorrermos ao próximo capítulo, que é sobre o Processo Legislativo Municipal, acerca do qual iniciamos com a introdução e, logo após, o processo legislativo, pelo qual se compreende a importância e necessidade deste trabalho acadêmico com sua proposta de solução ao mesmo.

3.1. Os três poderes

A União, que nada mais é que a República Federativa do Brasil, compõe-se de três poderes, a saber: Poder Legislativo, Poder Executivo e Poder Judiciário, que são harmônicos e independentes entre si (Brasil, 1988). Ou seja, é prevista, em constituição, a independência e harmonização desses três poderes, cada um em sua atribuição prevista em constituição.

3.1.1. Executivo

É o poder que executa, assim como está descrito no nome. Sua responsabilidade e atribuição é a de observar as necessidades coletivas, atendendo como ordena a Constituição por meios tais como oferta de serviços públicos: saúde, educação, segurança e infraestrutura. O executivo aplica as leis de maneira prática e funcional. Nele estão embutidas as prefeituras, os governos de estados e a presidência da república (Prado *et al.*, 2021).

3.1.2. Judiciário

Sua função primária é a de interpretar e aplicar as leis, bem como julgá-las de acordo com os fatos sempre em consonância à Constituição Federal. Este é o nosso guardião das leis, funcionando como aquele que tem o dever de guardá-las e deferi-las. É composta de juízes, desembargadores, procuradores e os ministros (Prado *et al.*, 2021).

3.1.3. Legislativo

Este tem função representativa ideológica e/ou valores existentes na sociedade que representa. Este poder tem como atribuição revisar e elaborar leis, mas também a de fiscalizar, tornando-se essencial ao funcionamento do Estado (município), podendo também julgar em casos mais extremos (Prado *et al.*, 2021).

3.2. O Poder Legislativo

Os municípios são compostos de poder Legislativo e Executivo com base na Constituição Federal, “a fiscalização do Município será exercida pelo Poder Legislativo Municipal, mediante controle externo, e pelos sistemas de controle interno do Poder Legislativo Executivo Municipal, na forma da lei” (Machado, 2017, p. 14).

A Câmara é o Poder Legislativo; no município, é composta pelos vereadores e possui poder próprio, não sendo subordinada à prefeitura. A Câmara edita a lei orgânica, onde é definida as regras para o poder público municipal, como número de vereadores de acordo com limites constitucionais (Senado Federal, 2016).

Sua competência primária é a de representar a população, atuar na produção de criação de leis e a fiscalização dos atos do Executivo, ou seja, a prefeitura. Tais leis produzidas devem estar de acordo com o processo legislativo, o qual é específico, podendo apenas e somente tratar de temas referentes ao limite municipal (Senado Federal, 2016).

3.2.1. Estrutura da Câmara Municipal

Dentro da esfera legislativa pela visão constitucional, vemos, na Constituição Federal, no artigo 29 e inciso IV, a composição das Câmaras Municipais e sua regra quantitativa de acordo com o número de habitantes. Num limite de quinze mil habitantes até trinta mil, serão nove vereadores; de trinta mil a cinquenta mil, serão treze, e assim por diante (Brasil, 1988, p. 28).

“Nos termos do art. 29 da Constituição, a lei Orgânica do município é aprovada por 2/3 dos vereadores da Câmara Municipal, em dois turnos de votação, com intervalos de dez dias entre eles” (Senado Federal, 2016, p. 24). O STF e TSE recomendam seguir os critérios estipulados no art. 29, inciso IV, da Constituição.

3.3. Poder Legislativo Municipal

“A fiscalização do Município será exercida pelo Poder Legislativo Municipal, mediante controle externo, e pelos sistemas de controle interno do Poder Executivo Municipal, na forma da lei” (Brasil, 1988, Art. 31, p. 31). Este artigo da Constituição Federal nos demonstra que a câmara municipal é um Poder e não uma repartição pública comum (Machado, 2017, p. 14).

Este poder, em si, é composto por vereadores que são os representantes da população. Há basicamente duas atribuições da câmara que são legislativa e fiscalizatória. Legislativa, pois ela incide sobre temas de interesse local e criação de leis; e fiscalizatória pois ela controla a ação do Poder Executivo, acompanhando os serviços municipais (Senado Federal, 2016, p. 19).

3.3.1. Regimento Interno

Regimento interno é um normativo central, que possui a função de regulamentar os trabalhos e o seu desempenho. Nele, há previsão da quantidade de comissões, critérios para a concessão ao vereador para que possa discursar em plenário, diretrizes, regras propositivas entre outros (Senado Federal, 2016, p.21).

De acordo com o Senado Federal (2016), há competências específicas relacionadas à câmara como o Plenário, Mesa e Comissões.

- Plenário: é o órgão que reúne vereadores, constituindo as instâncias máxima e decisória da casa legislativa.
- Mesa: também composta por vereadores, e pelos seus pares eleitos para condução dos trabalhos.
- Comissões: são órgãos colegiados de caráter permanente ou temporário. São constituídas por vereadores, que, nas proposições legislativas, as examinam. Possui caráter investigativo e acompanha os atos do Poder Executivo.

3.4. Projetos e requerimentos

Expor a visão jurídica do Poder Legislativo nos direciona a um tema mais específico; projetos e requerimentos são de maneira sistemática explanadas, direcionadas e organizadas para o desenrolar do meio legislativo. Cada órgão legislativo possui sua peculiaridade e necessidade regimental de trabalho, porém, tal proposto se direciona como objeto de estudo apenas o órgão Legislativo da cidade de Trindade (GO), por se tratar primeiramente do município de origem e sua facilidade de acesso físico.

3.4.1. Competência para Proposição

Consta no regime interno que proposição é um projeto de lei que regula as matérias as quais sua competência cabe ao município pela sanção do prefeito. “Projeto de lei é a proposição que tem por fim regular toda matéria de competência do Município e sujeita à sanção do Prefeito” (Regimento Interno. Art. 84, p. 30).

§ 1º - A iniciativa dos projetos de lei será;
I. Do Vereador
II. Da Mesa;
III. De Comissão da Câmara;
IV. Do Prefeito;
V. De 5%(cinco por cento) do eleitorado do Município (Regimento Interno. 2000, cap. II, art. 84).

3.4.2. Requerimentos

O vereador poderá sugerir a proposição que é o requerimento, através do qual irá sugerir medidas populares que interessem ao público. Tal medida de

assuntos da vida comunitária poderá ser de qualquer tema no que abrange aos aspectos econômicos, sociais, culturais e políticos. A competência dos requerimentos são:

- a) Sujeito à deliberação do Presidente;
- b) Sujeito à deliberação do Plenário (Regimento Interno, 2000, cap. III, art. 92).

3.4.3. Atribuições do Presidente

As atribuições do presidente segundo o regimento interno:

São atribuições do presidente, além das que estão expressas neste Regimento ou decorram da natureza de suas funções prerrogativas: Quanto às proposições: a receber as proposições apresentadas e distribuir proposições, pelos processos e documentos às comissões, solicitar informações e colaborações técnicas para estudo de matérias sujeitas à apreciação da Câmara, quando requerido pelas comissões (Regimento Interno, 2000, cap. I, art. 15, inc. II).

3.4.4. Aprovação de Projetos e Leis

Há alguns trâmites para o procedimento do projeto de lei que seguem seu rito normativo descrito no regimento. Após aprovada a lei, tem-se dez dias para encaminhá-la ao Executivo com as devidas assinaturas; o prazo para sancioná-la ou vetá-la é de quinze dias. Após esse prazo, caso não haja nenhuma manifestação do Prefeito, o Presidente da Câmara poderá promulgá-la (Regimento Interno, 2000, art. 121).

4. DIAGRAMAS DO PROCESSO DE CONSTRUÇÃO DE SOFTWARE

Um diagrama UML é um conjunto em que usa-se notação gráfica, ajudando na descrição do projeto. UML basicamente é *Unified Modeling Language* ou linguagem de modelo unificada. Tais linguagem existem há algum tempo na indústria de software. Unificada por muitas linguagens, nasceu com base nessa unificação de maneira gráfica, que em 1997 fizera sua primeira aparição (Fowler, 2005).

Todo projeto de software possui um início, neste caso a UML será o escopo, o pontapé inicial do projeto. Desenhar os diagramas é o primeiro ato de todo desenvolvedor; somente após isso é escrita toda codificação necessária para o software. A UML é o projeto desenhado de maneira que o desenvolvedor, em qualquer linguagem, seja necessário desenvolvê-lo (Fowler, 2005).

4.1. Ferramentas usadas

Para a diagramação e inserção da visualização do projeto, torna-se necessário o uso de determinada ferramenta de apoio para o mesmo. Neste caso, usa-se nesse projeto a ferramenta StarUML para a criação dos diagramas.

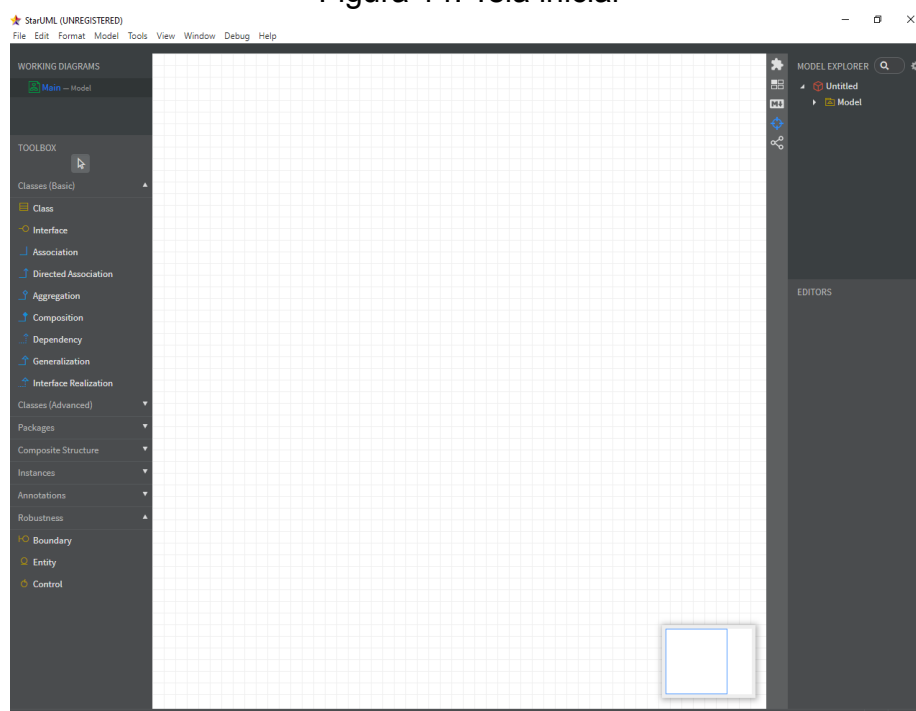
Figura 13: Logo StarUML



Fonte: StarUML ([2023])

A seguir, mostra-se a tela de início para a construção do diagrama no projeto:

Figura 14: Tela inicial



Fonte: StarUML ([2023])

4.2. Abordagem do projeto

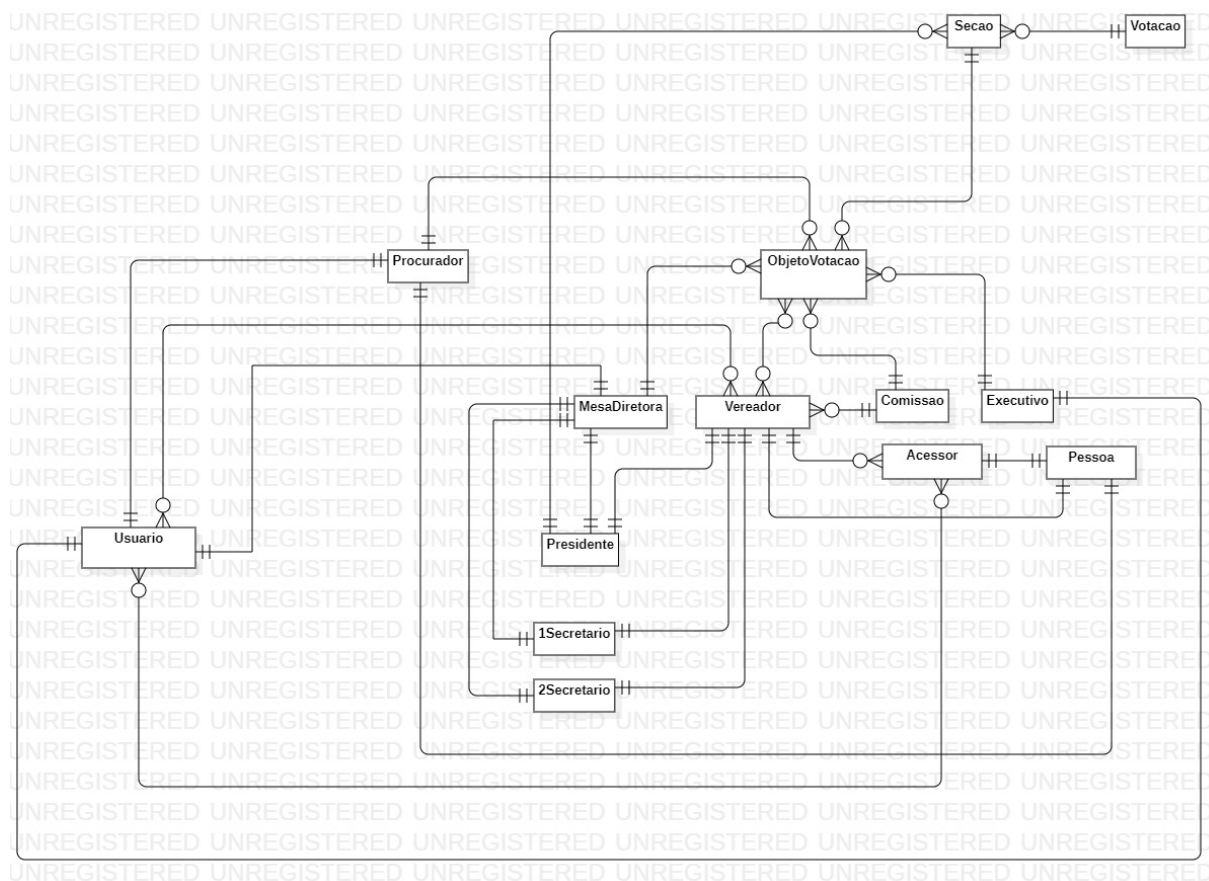
No que se refere à abordagem do projeto para construções futuras, foi de grande receptividade por parte dos envolvidos na abordagem analítica para que o andamento dele fosse plausível. O impacto proporcionado gerou uma grande ênfase de solução para o que lhe foi proposto na introdução.

Houve, de antemão, além de materiais nos termos jurídicos e específicos; o diálogo ou entrevista sobre a necessidade de ter um software que atenda as necessidades, trazendo um impacto significativo para solução das demandas dele. Sem esse parecer da procuradoria, este projeto tornaria inviável sua elaboração e construção.

Além disso, mostrou-se necessária a participação de outros profissionais do legislativo para que a visão do projeto abrangesse padrões e visões de diferentes usuários no sistema. Por se tratar de um software jamais visto pelo legislativo municipal local de Trindade (GO), será de extrema importância que, ao dar prosseguimento para construções futuras, a equipe desenvolvedora esteja ciente do tamanho e da proporção do mesmo por ser inédito no órgão.

4.3. Diagrama de entidade e relacionamento

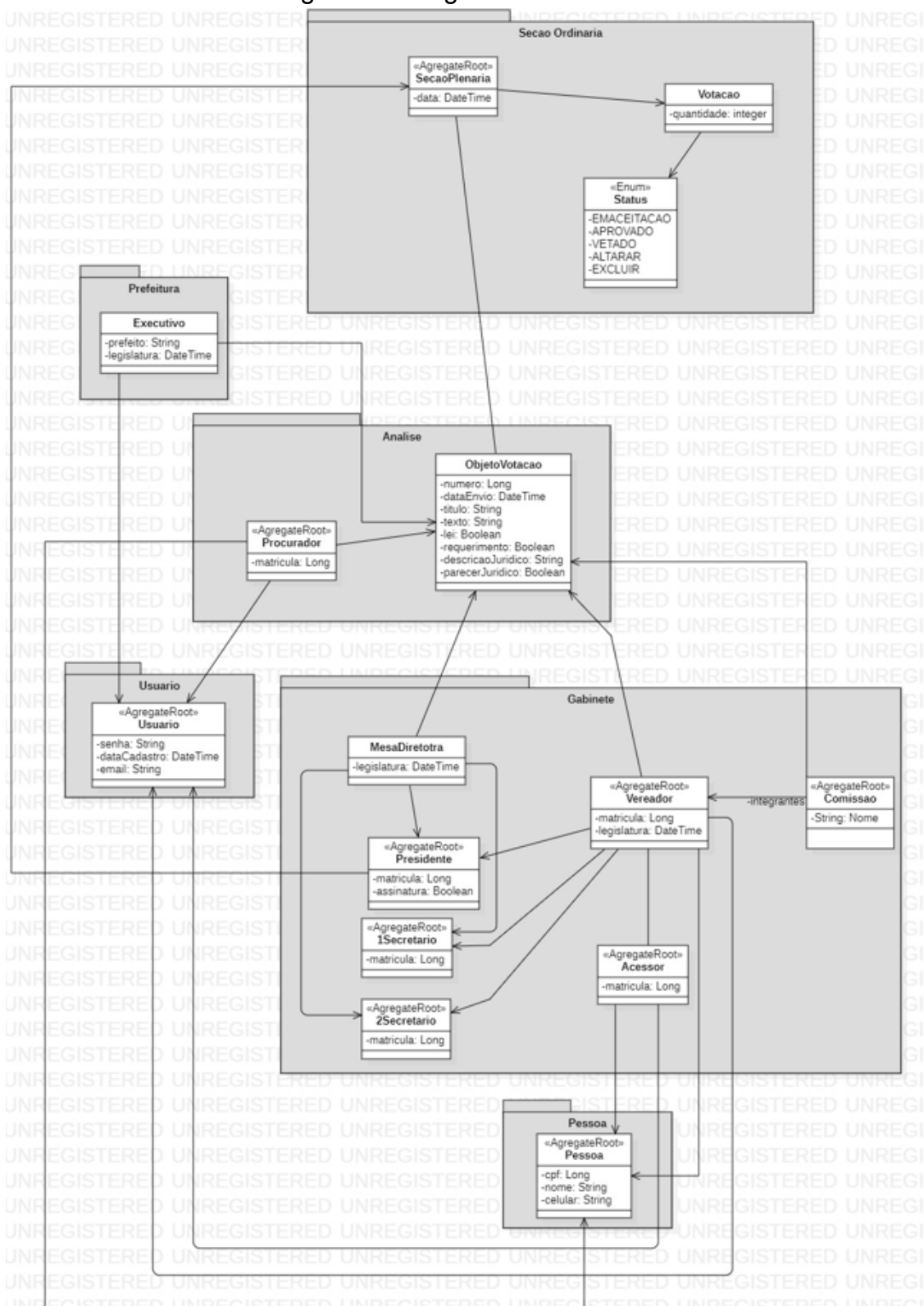
Figura 15: Entidade Relacionamento



Fonte: org. do autor (2023)

4.4. Diagrama de classes

Figura 16: Diagrama de Classes



Fonte: org. do autor (2023)

4.5. Caso de uso

Caso de Uso: Sistema de Projetos Legislativo (Prolegis)

-Atores envolvidos: procurador jurídico, vereadores, presidente, mesa diretora, comissão, prefeito e assessores.

- Pré-Condições:

- Os envolvidos serão previamente cadastrados para login pelo administrador.
- Todos os envolvidos devem fazer Login no sistema.
- O presidente deverá ter autonomia sobre a seção.
- O procurador deverá ter autonomia sobre o objeto de votação.

- Pós-Condições:

- Poderá ser realizada a impressão dos arquivos enviados.
- Deverão ser encaminhadas cópias do arquivo para o e-mail do Jurídico como ordem do dia.

-Fluxo Principal de Sucesso:

1. Serão cadastrados no sistema os usuários com e-mail, senha e dados pessoais:
 - a. Vereadores
 - b. Presidente
 - c. Mesa Diretora
 - d. Comissão
 - e. Procurador Jurídico
 - f. Prefeito
 - g. Assessores
2. Pautas de Sessão
 - a. As pautas serão enviadas para a procuradoria e para análise.
 - b. O prefeito poderá enviar projetos por um login específico do executivo, através de um servidor designado pela mesa diretora
3. Análise dos Projetos
 - a. A procuradoria irá analisar:
 - i. Inconstitucionalidade
 - ii. Legalidade
 - iii. Erros ortográficos
 - iv. Erros de sintaxe ou semântica

4. Projetos Aprovados

- a. Após aprovado irá como objeto de votação (pauta) para sessão plenária.
- b. O presidente, após cada votação, irá informar no sistema através de notebook ou tablet.

-Fluxo Secundário Erro:

1. Erro de Login:

- a. O administrador poderá recadastrar novamente.

2. Alteração Usuário ou Senha

- a. O usuário poderá alterar a senha na caixa de alterar senha.
- b. A senha será enviada para o e-mail cadastrado no sistema.

3. Alteração de Projetos já enviados:

- a. Somente poderão alterar projetos para aqueles sem parecer positivo pela procuradoria.
- b. Para alterar projetos que foram dados parecer positivo.
 - i. O solicitante (Vereador, Assessor, Executivo ou Mesa Diretora), poderá solicitar ao procurador que retorne o projeto para alteração.
 - ii. O projeto retornado será marcado com o parecer negativo, pela Procuradoria

4. Projetos com Parecer Negativo do Jurídico:

- a. Neste caso, ficará disponível para alteração ou exclusão do projeto.
- b. Caso o Solicitante (Vereador, Assessor, Executivo ou Mesa Diretora), não queira alterar, mas excluir, deverá clicar em excluir o projeto que será deletado automaticamente.

5. Projetos Retirados da Pauta:

- a. Somente o procurador poderá retirar o projeto da pauta de votação; neste caso, apenas será aceita a retirada se a sessão de votação não tiver começado.
- b. Somente será retirado de votação para alteração ou exclusão pelo presidente.
- c. Caso o presidente retire da votação:
 - i. Para exclusão,
 1. O procurador terá permissão para excluir o projeto.

ii. Para Alteração

1. O procurador alterará o projeto e enviará para pauta de votação.

6. Projetos retirados para alteração na pauta:

- a. O Presidente irá clicar (Tablet, Notebook) para retirar de votação no projeto, que deverá ser retirado de votação e ficará excluído para o procurador alterar ou excluir segundo o que lhe for proposto.
- b. Projetos que saírem da sessão para alteração poderão ser retornados para pauta de votação plenária pelo procurador segundo o tópico 5.

7. Exclusão do Projeto

- a. Somente procurador poderá excluir projeto de pauta solicitado pelo Presidente.
- b. Para alterar o projeto de sessão.
- c. Os projetos excluídos deverão ficar salvos para consultas posteriores.
- d. Deverá ser colocado o motivo da exclusão do projeto.

8. Projetos Aprovados:

- a. Sai da pauta de sessão e fica salvo em banco de dados.

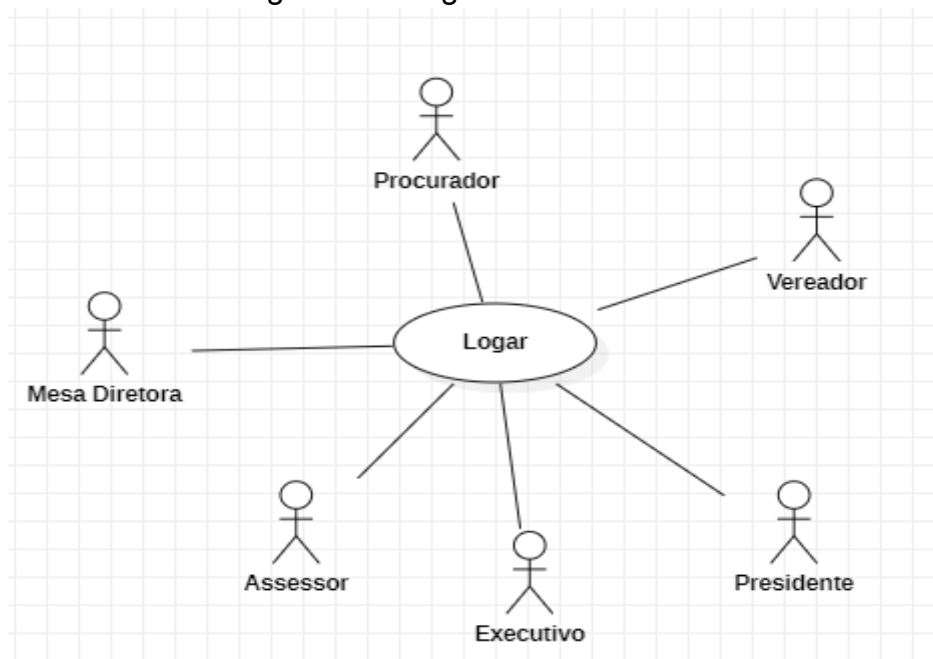
9. Projetos Vetados:

- a. Retornam para ser editados ou excluídos.

4.5.1. Diagrama de Caso de Uso

- Para Logar no sistema:

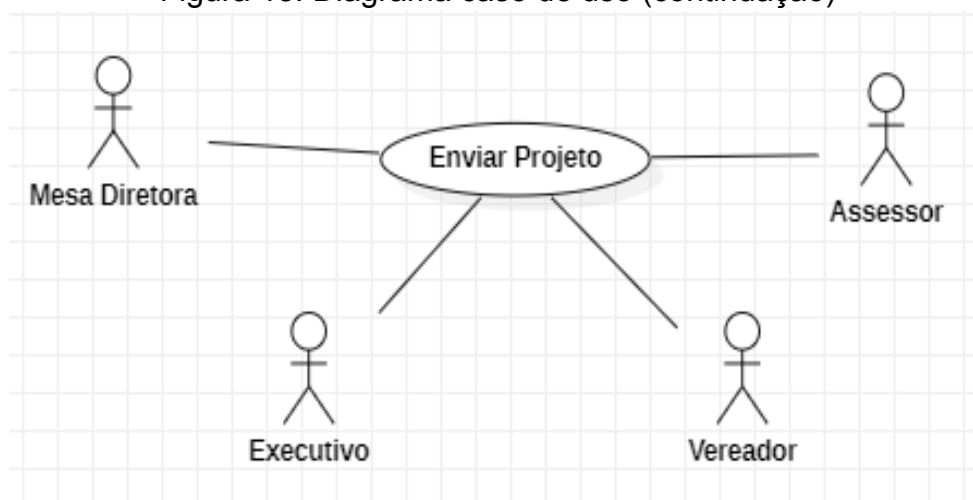
Figura 17: Diagrama caso de uso



Fonte: org. do autor (2023)

- Envio de projeto:

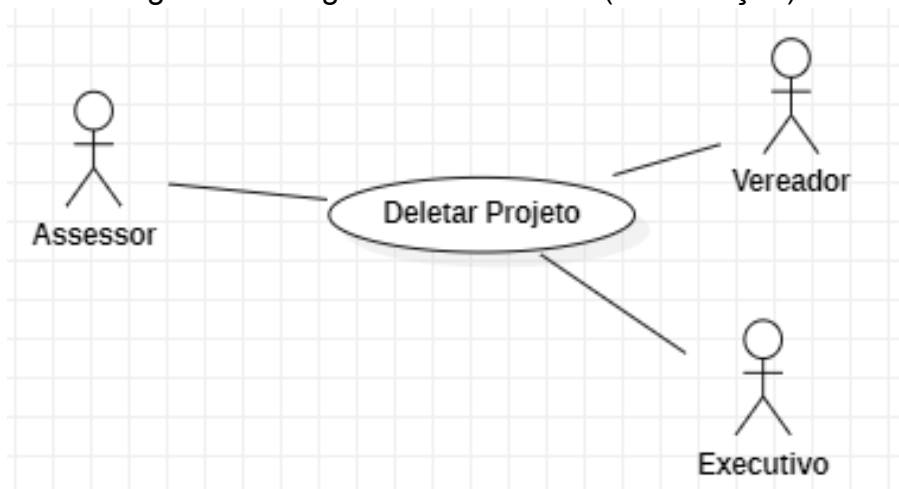
Figura 18: Diagrama caso de uso (continuação)



Fonte: org. do autor (2023)

- Deletar projeto:

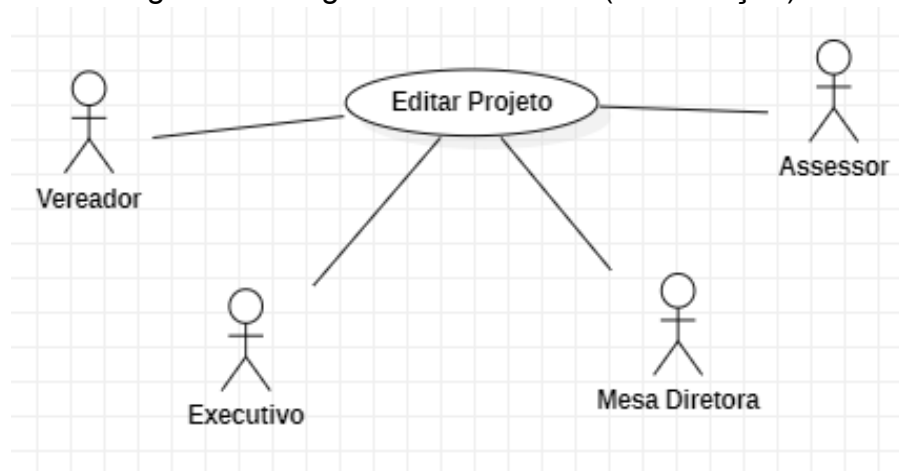
Figura 19: Diagrama caso de uso (continuação)



Fonte: org. do autor (2023)

- Editar projeto:

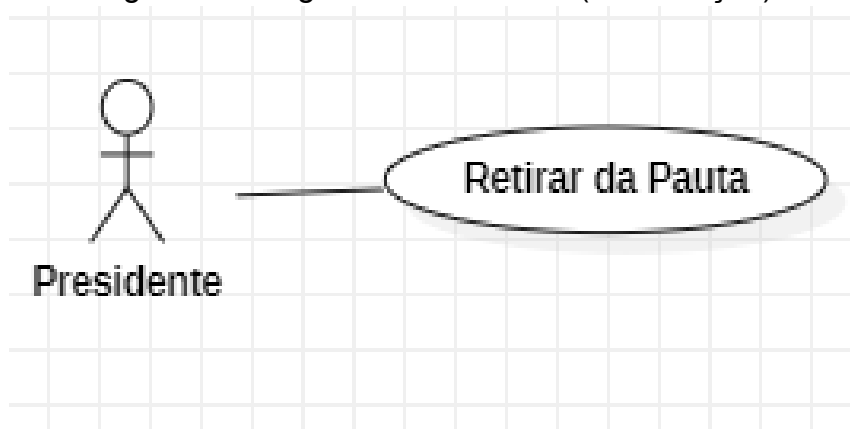
Figura 20: Diagrama caso de uso (continuação)



Fonte: org. do autor (2023)

- Retirar da pauta de votação:

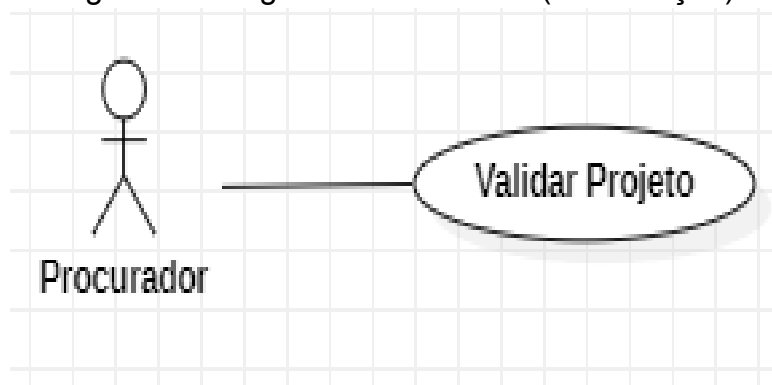
Figura 21: Diagrama caso de uso (continuação)



Fonte: org. do autor (2023)

- Validar projeto:

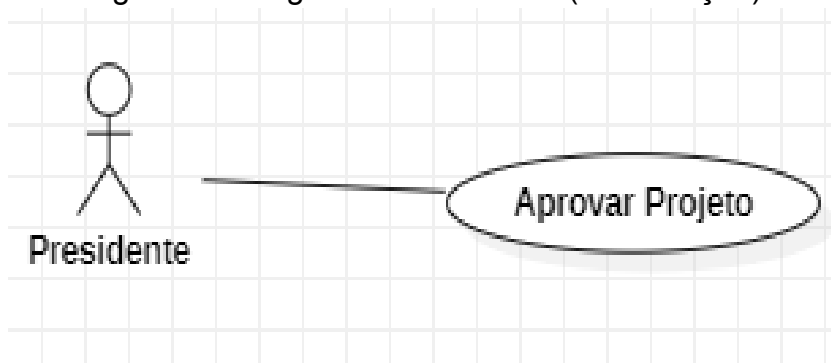
Figura 22: Diagrama caso de uso (continuação)



Fonte: org. do autor (2023)

- Aprovar projeto:

Figura 23: Diagrama caso de uso (continuação)



Fonte: org. do autor (2023)

5. TRABALHOS FUTUROS

A sugestão para os trabalhos posteriores é, primeiramente, a implementação do software em sua fase de aplicação web, para ser usado de maneira interna, no ambiente da Câmara Municipal, podendo atender às necessidades dos usuários que enviam as demandas para pauta plenária de votação na sessão ordinária.

Uma outra sugestão seria a extensão da aplicação mobile, estendendo até a comunidade suas solicitações e demandas, bem como os projetos de lei para que seja averiguada a possibilidade e viabilidade do mesmo para envio de pautas diretamente pela população sem a necessidade de que o vereador faça esta intermediação.

REFERÊNCIAS

ALFF, Francisco Roberto. Documentos de requisitos de software. **Portal Análise de Requisitos**. 15 Jan. 2018. Disponível em: <https://analisederequisitos.com.br/documento-de-requisitos-de-software/>. Acesso em: 17 Jul. 2023.

AMOROSO, Edgar. Engenharia de Requisitos de Software. **Católica.edu**. Disponível em:

https://conteudo.catolica.edu.br/conteudos/nbt_cursos/engenharia_requisitos/tema_02/index.html?access_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwczpcL1wvY29udGV1ZG8uY2F0b2xpY2EuZWZWR1LmJyIiwiaXVkljoiaHR0cHM6XC9cL2NvbnRldWRvLmNhdG9saWNhLmVkdS5icilslmlhdCI6MTU5Mzk4NjE0NiwiImJmljoxNTkzOTg2MTQ1LCJkYXRhIjpbIjNkZWE2OWRlIiwiaXVhYTEyNjEiXX0.iZ3ceZBLZEMmEdbijnCqxFAQH7GuNtAahhDVLQlgVN6g. Acesso em: 09 Mar. 2023.

BRASIL. [Constituição (1988)]. **A Constituição da República Federativa do Brasil**. Organizado por Aloysio de Brito Vieira. Brasília: Edição Administrativa do Senado, 2021.

FOWLER, Martin. **UML Essencial: Um Breve Guia para a Linguagem-Padrão de Modelagem de Objetos**. 3º Ed. Porto Alegre: Brookman, 2005.

FREITAS, Danilo Pestana de. **Ampliando a Colaboração no Levantamento de Sistemas**. Dissertação de Mestrado em Informática. Instituto de Matemática e Núcleo de Computação Eletrônica. UFRJ, 2006.

GALLOTTI, Giocondo Marino Antonio. **Arquitetura de Software**. São Paulo: Pearson Education do Brasil, 2016.

HIRAMA, Kechi. **Engenharia de Software: Qualidade e Produtividade com Tecnologia**. Rio de Janeiro: Elsevier, 2011.

INTHURN, Cândida. **Qualidade & Teste de Software**. Florianópolis: Visual Books, 2001.

IEEE Std. 830, IEEE Guide to Software Requirements Specification, The Institute of Electrical and Electronics Engineers, New York, EUA, (1984).

IEEE Std. 12333-1996 IEEE Guide for Developing System Requirements Specification. The Institute of Electrical and Electronics Engineers. Piscataway, NJ. Guide, EUA, 1996.

KERR, Eduardo Santos. **Gerenciamento de Requisitos**. São Paulo: Pearson, 2015.

KOURI, Márcia G. Definição de requisitos para um sistema de monitoramento de veículos no transporte rodoviário de cargas. São Paulo, 2007. 165f.. Dissertação (Mestrado em Engenharia Elétrica) – Escola Politécnica, Universidade de São Paulo.

LARMAN, Craig. **Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientados a Objetos e ao Desenvolvimento Iterativo**. 3 ed. Porto Alegre: Bookman, 2007.

LEDUR, Cleverson Lopes. **Análise de Projeto de Sistemas**. Porto Alegre: Sagah, 2017.

LEITE, Jair C. Notas de Aula de Engenharia de Software. 2000. Disponível em: <www.dimap.ufrn.br/~jair/ES/c4.html>. Acesso em: 12 mai. 2023.

MACHADO, Armando Marcondes. **Poder Legislativo Municipal**. Vereador: alicerce e exemplo para a valorização da cidadania. 2 ed. São Paulo: Conan, 2017.

MACHADO, Felipe Nery Rodrigues. **Análise e Gestão de Requisitos de Software: Onde Nasce os Sistemas**. 2 ed. São Paulo: Érica, 2014.

MONITORA, Team. Análise de Requisitos de Software: quais desafios e como fazer?. **Monitora**. 21 Jan. 2021. Disponível em: <https://www.monitoretec.com.br/blog/analise-de-requisitos-de-software/#:~:text=A%20an%C3%A1lise%20de%20requisitos%20de%20software%20ajuda%20o%20cliente%20e,as%20fun%C3%A7%C3%B5es%20do%20novo%20sistema>. Acesso em: 10 Mar. 2023.

MONITORA, Team. SERVIÇO de Engenharia de Requisitos: Entenda como Funciona. **Monitora**. 28 Ago. 2020. Disponível em: <https://www.monitoretec.com.br/blog/analise-de-requisitos-de-software/#:~:text=A%20an%C3%A1lise%20de%20requisitos%20de%20software%20ajuda%20o%20cliente%20e,as%20fun%C3%A7%C3%B5es%20do%20novo%20sistema>. Acesso em 10 mar. 2023.

MORAIS, Izabelly Soares de. **Engenharia de Software**. Porto Alegre: Sagah, 2017.

PAULA FILHO, Wilson de Pádua. **Engenharia de Software: Produtos**. 4 ed. Rio de Janeiro: LTC, 2019.

PRESSMAN, Roger. **Engenharia de Software**. 3. Ed. São Paulo: Makron Books, 1995.

PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de Software: uma abordagem profissional**. 9. Ed. Porto Alegre: AMGH, 2021.

REINERH, Sheila. **Engenharia de Requisitos**. Porto Alegre: Sagah, 2020.

REZENDE, Denis Alcides. **Engenharia de Software e Sistemas de Informação**. 3 ed. rev. e ampl. Rio de Janeiro: Brasport, 2005.

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

SOMMERVILLE, Ian. **Engenharia de Software**. 10 ed. São Paulo: Pearson Education do Brasil, 2018.

SENADO FEDERAL. **O Poder Legislativo Municipal no Brasil**: Papel instituições, desafios e perspectivas. Brasília: Conselho de Estudos Políticos, 2016.

TRINDADE. [Regimento Interno (2000)]. **Regimento Interno: Câmara Municipal de Trindade**. Organizado pela Câmara Municipal. Trindade, 2000. 48 p.

VALENTE, Marco Tulio. **Engenharia de Software Moderna**: Princípios e Práticas para Desenvolvimento de Software com Produtividade. Belo Horizonte: Independente, 2020. E-book.

VAZQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira. **Engenharia de Requisitos**: Software Orientado ao Negócio. Rio de Janeiro: Brasport Livros, 2016.

VETORAZZO, Adriana de Souza. **Engenharia de Software**. Porto Alegre: SAGAH, 2018.

VALENTE, Marco Túlio. A Engenharia de Software Moderna. EBook, Belo Horizonte: Engosoft Moderna, 2020.

ZANCHETT, Pedro Simei. **Engenharia e Projeto de Software**. Indaial: Uniasselvi, 2015.

Vídeos

O que é Levantamento de Requisitos - Tópicos de Engenharia de Software. [S. l.: s. n.], 2018. 1 vídeo (21 min). **Bóson Treinamentos**. Disponível em: <https://www.youtube.com/watch?v=VcOeM2AD8Yk>. Acesso em 10 mar. 2023.