

**INSTITUTO FEDERAL**  
**SANTA CATARINA**

CÂMPUS FLORIANÓPOLIS  
DEPARTAMENTO ACADÊMICO DE SAÚDE E SERVIÇOS  
CURSO SUPERIOR DE TECNOLOGIA EM GESTÃO  
DE TECNOLOGIA DA INFORMAÇÃO

FERNANDO MASSAO FRAZNIN NISHI

# REQUISITOS DE SOFTWARE: PROPOSTA, ESPECIFICAÇÃO E OTIMIZAÇÃO DE UM METAFRAMEWORK PARA A DOCUMENTAÇÃO

Florianópolis - SC  
2019

Ficha de identificação da obra elaborada pelo autor.

Nishi, Fernando Massao Franzin  
**REQUISITOS DE SOFTWARE: PROPOSTA, ESPECIFICAÇÃO E  
OTIMIZAÇÃO DE UM METAFRAMEWORK PARA A DOCUMENTAÇÃO** / Fernando  
Massao Franzin Nishi ; orientação de Cleverson Tabajara  
Vianna; coorientação de Underléa Cabreira  
Corrêa. - Florianópolis, SC, 2019.

123 p.  
Trabalho de Conclusão de Curso (TCC) - Instituto Federal  
de Santa Catarina, Câmpus Florianópolis. CST  
em Gestão da Tecnologia da Informação. Departamento  
Acadêmico de Saúde e Serviços.  
Inclui Referências.

1. Documentação de requisitos. 2. Template de documento  
de requisitos. 3. Requisitos de software. I. Vianna,  
Cleverson Tabajara. II. Corrêa, Underléa Cabreira.  
III. Instituto Federal de Santa Catarina. Departamento  
Acadêmico de Saúde e Serviços. IV. Título.

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA  
CATARINA  
DEPARTAMENTO ACADÊMICO DE SAÚDE E SERVIÇOS  
CURSO SUPERIOR DE TECNOLOGIA EM GESTÃO DA TECNOLOGIA DA  
INFORMAÇÃO**

**FERNANDO MASSAO FRANZIN NISHI**

**REQUISITOS DE SOFTWARE: PROPOSTA, ESPECIFICAÇÃO E OTIMIZAÇÃO DE UM  
*METAFRAMEWORK* PARA A DOCUMENTAÇÃO**

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Tecnólogo em Gestão da Tecnologia da Informação.

Professor Orientador:

Prof. Cleverson Tabajara Vianna, Me.

Professora Coorientadora:

Prof. Underléa Cabreira Corrêa, Dra;.

**FLORIANÓPOLIS - SC**

**OUTUBRO/2019**

**REQUISITOS DE SOFTWARE: PROPOSTA, ESPECIFICAÇÃO E OTIMIZAÇÃO  
DE UM METAFRAMEWORK PARA A DOCUMENTAÇÃO**

**FERNANDO MASSAO FRANZIN NISHI**

Este trabalho foi julgado adequado para obtenção do Título de Tecnólogo em Gestão da Tecnologia da Informação e aprovado na sua forma final pela banca examinadora do Curso Superior de Tecnologia em Gestão da Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

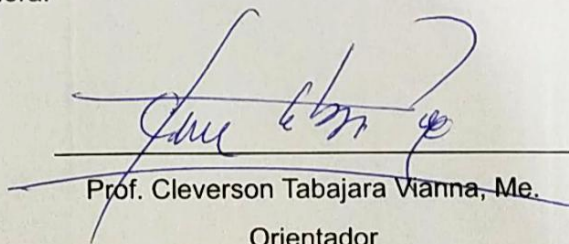
Florianópolis-SC, 05 de novembro de 2019.



Prof. Felipe Cantprio Soares, Msc.

Coordenador do CST em Gestão da Tecnologia da Informação  
Instituto Federal de Santa Catarina

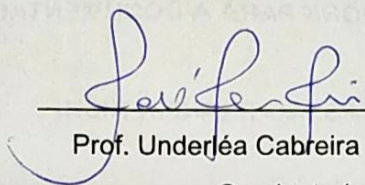
Banca Examinadora:



Prof. Cleverson Tabajara Vianna, Me.

Orientador

Instituto Federal de Santa Catarina

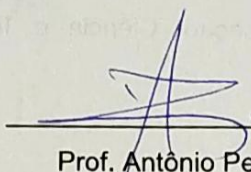


---

Prof. Underléa Cabreira Corrêa, Dra.

Coorientador

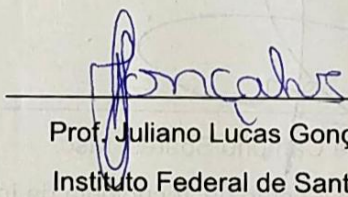
Instituto Federal de Santa Catarina



---

Prof. Antônio Pereira Cândido, Dr.

Instituto Federal de Santa Catarina



---

Prof. Juliano Lucas Gonçalves, Me.

Instituto Federal de Santa Catarina

*Dedico este trabalho a minha família, em especial ao meu pai e a minha irmã, por todo carinho, atenção e apoio que me deram e ainda me dão durante todo o caminho percorrido.*



## **AGRADECIMENTOS**

Agradeço, primeiramente ao meu pai José Shigueo Nishi e minha irmã Mariane Yume Franzin Nishi, que sempre me apoiam e incentivam a buscar o crescimento e a minha felicidade.

Não menos importante, agradeço também a toda minha família, por estarem sempre ao meu lado, tanto nos bons momentos quanto nos ruins.

Ao meu orientador Cleverson Tabajara Vianna e coorientadora Underlea Cabreira Corrêa, que dedicaram valiosas horas de suas vidas para me auxiliar neste trabalho de conclusão de curso.

Agradeço também a todos os professores e ao Instituto Federal de Santa Catarina que me proporcionaram ótimas experiências tanto na vida profissional quanto na acadêmica.

*“Aquilo que se faz por amor está sempre além do bem e do mal.”*

*Friedrich Nietzsche*



## RESUMO

A não utilização de métodos de engenharia de software tem gerado maiores custos e sistemas menos confiáveis. Um dos fatores cruciais é a documentação do sistema que gera dificuldades para clientes, usuários e desenvolvedores. . Em virtude disso, o objetivo principal deste trabalho é elaborar um *metaframework* para automatizar a documentação de requisitos de *software*. Para se obter os devidos resultados, foi elaborado um template de documentação de requisitos e aplicado um questionário com o intuito de validá-lo e coletar informações sobre a percepção dos usuários referente a documentação de requisitos de *software*. Desta forma conclui-se que o presente trabalho visa contribuir para o entendimento sobre algumas dificuldades enfrentadas pelos usuários com uma documentação mal estruturada. Ao mesmo tempo possibilitou a descoberta de parâmetros importantes para o planejamento e construção de uma ferramenta para automatizar a documentação de requisitos. O metaframework proposto apresenta propriedades de BDD como descrição de requisitos funcionais, controle de versionamento, inclusão de protótipos de telas e sua interação.

Palavras-chave: Desenvolvimento de Software; Análise de Requisitos; Documentação de Requisitos.

## ABSTRACT

*Failure to use software engineering methods has led to higher costs and less reliable systems. One of the crucial factors is system documentation that creates difficulties for customers, users, and developers. . Because of this, the main objective of this work is to elaborate a metaframework to automate the documentation of software requirements. To obtain the appropriate results, a requirements documentation template was developed and a questionnaire was applied to validate it and collect information on users' perceptions regarding software requirements documentation. Thus, it is concluded that the present work aims to contribute to the understanding of some difficulties faced by users with poorly structured documentation. At the same time, it made it possible to discover important parameters for the planning and construction of a tool to automate requirements documentation. The proposed metaframework presents BDD properties such as functional requirements description, versioning control, screen prototype inclusion and their interaction.*

Key-words: Software Development; Requirements Analysis; Requirements Documentation.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Modelo em cascata.....	25
Figura 2 - Custo de mudança no projeto no modelo cascata.....	27
Figura 3 - Metodologia Extreme Programming .....	28
Figura 4 - modelo de documento de requisitos (parte 1).....	38
Figura 5 - modelo de documento de requisitos (parte 2).....	40
Figura 6 - modelo de documento de requisitos (parte 3).....	41
Figura 7 - Área de atuação.....	42
Figura 8 - Tempo de atuação na área de tecnologia da informação.....	43
Figura 9 - Contato com documento de requisitos.....	44
Figura 10 - Dificuldades de documentação de sistemas e especificação de requisitos..	46
Figura 11 - Pontos fortes da documentação de sistemas e especificação de requisitos.	47
Figura 12 - Pontos fracos da documentação de sistemas e especificação de requisitos.	48
Figura 13 - Documentos de requisitos (parte 1).....	49
Figura 14 - Documentos de requisitos (parte 2).....	50

## **LISTA DE TABELAS**

Tabela 1 - Quantidade de empresas atuando na área de softwares e serviços.....	16
--	----

## LISTA DE QUADROS

Quadro 1 - Razões para realização da modelagem de sistemas.....	17
Quadro 2 - Vantagens e desvantagens do modelo em cascata.....	26
Quadro 3 - Definição de requisitos.....	29
Quadro 4 - Etapas da engenharia de requisitos.....	30
Quadro 5 - Estrutura de documento de requisitos.....	32
Quadro 6 - Análise de documentos de requisitos.....	34
Quadro 7 - Matriz de questões.....	44
Quadro 8 - Parâmetros para o planejamento de uma ferramenta para documentação de requisitos.....	49

## LISTA DE ABREVIATURAS E SIGLAS

<b>ABES</b>	Associação Brasileira das Empresas de Software
<b>IEC</b>	<i>International Electrotechnical Commission</i>
<b>IEEE</b>	<i>Institute of Electrical and Electronic Engineers</i>
<b>ISO</b>	<i>International Organization for Standardization</i>
<b>IFSC</b>	Instituto Federal de Educação Tecnológica de Santa Catarina
<b>PPC</b>	Projeto Pedagógico do Curso
<b>XP</b>	<i>Extreme Programming</i>

## SUMÁRIO

RESUMO	8
1. INTRODUÇÃO	16
1.1 Justificativa	18
1.2 Definição do Problema	20
1.3 Objetivos	20
1.3.1 Objetivo Geral	21
1.3.2 Objetivos Específicos	21
1.4 Estrutura do Trabalho	21
2. REVISÃO DA LITERATURA	23
2.1 Processos de Software	23
2.2 Engenharia de Requisitos	29
3. PROCEDIMENTOS METODOLÓGICOS	35
4. TEMPLATE DE DOCUMENTO DE REQUISITOS REQUISITOS E VALIDAÇÃO	38
4.1 Template de documento de requisitos	38
4.2 Validação do template de documentação de requisitos	41
4.3 Otimização do template de documentação de requisitos	48
4.4 Proposta de um metaframework	51
5. CONCLUSÕES	54
5.1 Em relação ao objetivo geral	54
5.2 Em relação aos objetivos específicos	55
5.3 Em relação aos trabalhos futuros	56
REFERÊNCIAS	57
APÊNDICE A - DOCUMENTO DE REQUISITOS	62
APÊNDICE B - QUESTIONÁRIO DE VALIDAÇÃO	67
APÊNDICE C - RESPOSTAS DO QUESTIONÁRIO DE VALIDAÇÃO	72



## 1. INTRODUÇÃO

O crescente fluxo de informação presente nos diferentes setores das organizações, bem como a evolução dos processos empresariais, globalização, competitividade e etc, vem aumentando a demanda por ambientes informatizados. Segundo Sommerville (2011, p.2) “o mundo moderno não poderia existir sem o software”, pois infraestruturas e serviços nacionais, produtos elétricos, sistema financeiro e áreas de entretenimento são dependentes de recursos computacionais e softwares para o controle e/ou auxílio de seus processos.

Mediante ao aumento das demandas por recursos computacionais houve um grande aumento na quantidade de empresas atuando no setor de software e serviços, conforme os dados apresentados pela Associação Brasileira das Empresas de Software (ABES).

Tabela 1 - Quantidade de empresas atuando na área de softwares e serviços

<b>Empresas</b>	<b>2015</b>	<b>2016</b>	<b>2017</b>
Desenvolvimento e Produção	4.408	4.872	5.138
Distribuição e comercialização	5.732	6.365	6.513
Prestação de serviços	3.811	4.470	5.342
<b>Total</b>	<b>13.951</b>	<b>15.707</b>	<b>16.993</b>

Fonte: Adaptado a partir de ABES (2018, 2017 e 2016)

O crescimento de 2015 a 2016 foi de aproximadamente 9,53% e de aproximadamente 5,16% entre 2016 e 2017. Observa-se também que o maior crescimento foi na área de prestação de serviços com 1.531 novas empresas, seguido pelas áreas de distribuição e comercialização com 781 e desenvolvimento e produção com 730, somando um total de 3.042 novas empresas na área de softwares e serviços em tecnologia da informação.

Com tal crescimento tanto das necessidades, quanto das empresas de desenvolvimento, ainda é comum encontrar falhas no produto, referentes a problemas no processo de concepção e desenvolvimento de seus projetos de software. De acordo com

Sommerville (2011) a existência de diversos projetos que deram errado, resultando em “falha de software”, as quais são consequências de dois fatores:

- I. Aumento da demanda: “conforme novas técnicas de engenharia de software nos auxiliam a construir sistemas maiores e mais complexos, as demandas mudam” (SOMMERVILLE, 2011, p.2). Desta forma há a necessidade de os sistemas serem construídos e entregues mais rapidamente.
- II. Expectativas baixas: “Muitas empresas foram forçadas a desenvolver softwares à medida que seus produtos e serviços evoluíram” (SOMMERVILLE, 2011, p.3). Com isto elas acabam não utilizando métodos de engenharia de software, e consequentemente, acabam ficando mais caro e menos confiáveis.

O processo de construção de sistemas apresenta um conjunto de atividades a serem executadas. Bezerra (2007) salienta que para a construção de sistemas mais complexos há a necessidade de um planejamento inicial, com isto explica algumas razões para realizar a modelagem de sistemas:

Quadro 1 - Razões para realização da modelagem de sistemas

Gerenciamento da complexidade	“Um dos principais motivos de utilizar modelos é que há limitações no ser humano em lidar com a complexidade”, desta forma uma documentação clara contribui na estruturação do projeto.
Comunicação entre as pessoas envolvidas	“Os modelos servem para promover a difusão das informações relativas ao sistema entre os indivíduos envolvidos em sua construção.”
Redução de custos no desenvolvimento	“A correção de erros é menos custosa quando detectadas e realizadas ainda nos modelos do sistema.”
Previsão do comportamento futuro do sistema	“O comportamento do sistema pode ser discutido mediante uma análise dos seus modelos. Os modelos servem como um ‘laboratório’, em que diferentes soluções para um problema relacionado à construção podem ser experimentadas.”

Fonte: Adaptado a partir de Bezerra (2007, p. 3).

Uma das etapas da modelagem de um software é a **documentação de seus requisitos**. Para Rezende (2005) a documentação é elaborada em todas as etapas do

desenvolvimento e tem como objetivos principais: “divulgar e deixar, transparente e o que o sistema ou software faz, como utilizá-lo, as suas vantagens, operações e requisitos para funcionamento; mostrar quais técnicas ou ferramentas utilizadas para seu desenvolvimento, tempo, custos, equipe e etc” (p.275).

## 1.1 Justificativa

Como as demandas do mercado e dos sistemas estão em constante evolução, seus processos e necessidades também estão, desta forma seus requisitos acabam ficando ultrapassados, antes mesmo que o documento esteja finalizado (SOMMERVILLE, 2011). Como exemplo pode-se citar os sistemas operacionais e frameworks que gradativamente estão sendo atualizados, ora para melhoria de desempenho, ora para melhorias na interface gráfica.

Com isto os programas de computadores e *softwares* necessitam de atualizações constantes, necessitam também gerar novas versões de sua documentação. Apesar da necessidade dos documentos serem finalizados mais rapidamente, é imprescindível que eles sejam coesos e de fácil compreensão. Esta documentação será utilizada pelos demandantes da ferramenta para validação. Servirão como base para a equipe de gestão, desenvolvimento e testes, na elaboração de um plano de ação para executar suas atividades.

De acordo com Gabrieli et al. (2007, p. 4) um software bem avaliado na visão de um projetista deve apresentar um conjunto de características, a saber:

- I. Maior **flexibilidade**: possibilita satisfazer novos requisitos de negócio (funcionalidade) de forma fácil e rápida.
- II. Melhor **adaptabilidade**: possibilita personalizar uma aplicação para vários usuários, usando várias alternativas para oferecer os serviços da aplicação com o mínimo de impacto no seu núcleo.
- III. Melhor **manutenibilidade**: possibilita alterar partes de uma aplicação, de modo que as outras partes sofram um impacto mínimo.
- IV. Melhor **reusabilidade**: possibilita montar aplicações únicas e dinâmicas rapidamente.
- V. Melhor **aproveitamento** do legado: possibilita reutilizar a funcionalidade de sistemas legados em novas aplicações.
- VI. Melhor **interoperabilidade**: possibilita que duas aplicações que executam em plataformas diferentes troquem informações.
- VII. Melhor **escalabilidade**: possibilita distribuir e configurar a execução da aplicação de modo a satisfazer a vários volumes de transação.
- VIII. Melhor **robustez**: possibilita implementar soluções de software com menos defeitos.
- IX. Menor **tempo de desenvolvimento**: possibilita construir novos sistemas de forma mais rápida e com baixo orçamento.

- X. Menor **risco**: possibilita todas as características citadas para um software de qualidade, sem ter o risco de ter projetos fracassados.

Diante a rápida evolução dos requisitos, a elaboração e atualização da documentação para que um projeto seja bem avaliado apresenta dificuldades com a falta de tempo, retrabalho e etc. Desta forma para obter maior efetividade tanto no projeto de software, quanto em sua documentação, a utilização de *frameworks* se torna imprescindível nos dias atuais, e é fator fundamental para a competitividade. Para melhor compreensão do significado de *framework*, Cruz (2019, p. 15) define sendo “grupos de classes, estruturas de dados, implementações prontas para serem utilizadas e abstrações, que operam em conjunto para prover algum tipo de funcionalidade de altíssimo nível”.

Um *framework* possibilita a obtenção do conjunto de características de que um *software* deve conter, pois ele define padrões já consolidados, facilitando a manutenibilidade, reduzindo os riscos de erros e falta de padronização e aumentando sua robustez. Reduz também o tempo de entrega, uma vez que possui um conjunto de códigos e modelos pré estruturados. Pereira (2017, p.579) contribui ao afirmar que “ferramentas como frameworks existem para minimizar a codificação de problemas já resolvidos e maximizar a reutilização de códigos já testados por outros desenvolvedores”.

O presente trabalho justifica-se também no âmbito do curso de gestão de tecnologia de informação, disponibilizado pelo Instituto Federal de Educação Tecnológica de Santa Catarina (IFSC), no qual o Projeto Pedagógico do Curso (PPC) da matriz curricular de 2015 (IFSC, 2019, s/n) o curso tem alguns de seus objetivos específicos:

- I. Capacitar o aluno a conduzir projetos e liderar equipes relacionadas a TI, com o uso de metodologias e processos;
- II. Disseminar conhecimentos tecnológicos e gerenciais que possibilitem ao aluno conduzir projetos, programas e atividades de aplicação das tecnologias da informação com qualidade e segurança;

Estabelecidas a relevância, inicia-se o detalhamento do problema e dos objetivos pertinentes a este Trabalho de Conclusão.

## 1.2 Definição do Problema

Com o crescimento diário das demandas por sistemas e ferramentas tecnológicas os tempos ficam mais escassos e as exigências aumentam. Com isto, um dos desafios importantes para a engenharia de *software* é de encontrar e utilizar ferramentas efetivas, rápidas e práticas de auxílio na coleta, análise e documentação de requisitos.

A escassez de tempo acaba levando as empresas a acelerar ou mesmo eliminar etapas no processo de desenvolvimento dos projetos. Neste sentido a elaboração de ferramentas efetivas que auxiliam neste processo poderá proporcionar maior eficiência no desenvolvimento do produto final.

Atualmente encontra-se em abundância ferramentas que auxiliam na coleta de dados de requisitos e geram artefatos como protótipos de telas, estruturação de banco de dados, relacionamento de dados, modelagem de processos, casos de uso, entre outros. Entretanto, é um desafio unir-los de forma coesa para que a documentação facilite sua compreensão pelas partes envolvidas (técnicos e usuários).

Para o presente trabalho nos deparamos com a seguinte pergunta de pesquisa: **quais parâmetros podem contribuir para elaboração de uma ferramenta para automatizar a documentação dos requisitos de software?**

Para responder tal questionamento tem-se o objetivo geral de elaborar um *metaframework* para automatizar a documentação de requisitos de *software*. Conforme apresentado no item 1.3.

Para Nascimento (2018, p.11) um "*metaframework* se refere definido como um conjunto de ações e subsistemas amparados por teorias que, associadas e dentro de uma ordem de execução, possibilitam a geração de insumos", em nosso caso, voltados à documentação.

Naturalmente um *metaframework*, apresenta as dimensões e linhas gerais, não chegando á especificação final, visto que sua completa especificação demandaria um maior tempo de estudo e aplicação.

## 1.3 Objetivos

Para o desenvolvimento deste trabalho foram definidos os seguintes objetivos: geral e específicos.

### 1.3.1 Objetivo Geral

Este trabalho tem por objetivo elaborar um *metaframework* para automatizar a documentação de requisitos de *software*.

### 1.3.2 Objetivos Específicos

- Agrupar modelos teórico-práticos de forma consistente e lógica do ponto de vista do usuário e desenvolvedor.
- Elaborar template de documentação de sistemas.
- Validar o template de documentação de requisitos.
- Aperfeiçoar o template com base nas sugestões oferecidas.

### 1.4 Estrutura do Trabalho

No primeiro capítulo este apresenta os itens que contextualizam o tema. Estão inclusos a apresentação do problema de pesquisa os objetivos geral e específicos.

No capítulo seguinte (segundo), no embasamento teórico abordam conceitos de processos de desenvolvimento de software e alguns modelos de processos como o modelo em Cascata e o Extreme Programming, de modo a identificar como a engenharia de requisitos está envolvida neste processo. Ainda neste tópico apresenta-se a engenharia e documentação de requisitos, abordando sua relação, estrutura e importância para o desenvolvimento de um projeto de *software*.

O capítulo terceiro, referente aos procedimentos metodológicos aborda a forma que está organizado o presente trabalho de conclusão de curso e o tipo de pesquisa que se aplica.

No quarto capítulo apresenta-se o template de documentação de requisitos elaborado no decorrer deste trabalho. Em seguida aborda sua validação, realizado por meio de um questionário.

No quinto e último capítulo, conclusões, são apresentadas as conclusões relativas ao objetivo geral que engloba os aspectos relacionados à proposta de um *metaframework*

para auxiliar na documentação de requisitos de software. E também as considerações finais acerca dos objetivos específicos de agrupar modelos teórico-práticos de forma consistente e lógica do ponto de vista do usuário e desenvolvedor, elaborar template de documentação, validar o template de documentação de requisitos e aperfeiçoar o template com base nas sugestões oferecidas. Ao final do capítulo são apresentados as sugestões para trabalhos futuros referente ao tema desenvolvido.



## 2. REVISÃO DA LITERATURA

### 2.1 Processos de Software

Um conceito utilizado neste trabalho se refere ao processo de software: Bezerra (2007, p.22) compreende processos como sendo “todas as atividades necessárias para definir, desenvolver testar e manter um produto de software”.

Na mesma linha, Pressman (2006,p.16) define processo de software como “um arcabouço de tarefas que são necessárias para construir softwares de qualidade” (p. 16). Soares (2004) corrobora ao afirmar que “um processo de software (ou metodologia de desenvolvimento de software) é um conjunto de atividades e resultados associados que auxiliam na produção de software” (p. 1).

Para o presente trabalho pode-se então compreender que processo de software ou processo de desenvolvimento de software como um conjunto de etapas parcialmente ordenadas, cujo objetivo é guiar na caminhada de construção, implementação de novas funcionalidades e/ou manutenção de um software até a obtenção do produto final.

Atualmente existem diversos modelos de processos, mas independentemente de suas particularidades, todos devem possuir quatro atividades fundamentais para a engenharia de software (SOMMERVILLE, 2011):

Especificação de software: consiste na definição das funcionalidades e restrições do software. Geralmente é nesta etapa que consolida-se o entendimento do projeto e requisitos.

- I. Projeto e implementação de software: ocorre a construção do software de forma a respeitar as funcionalidades definidas na especificação.
- II. Validação de software: nesta etapa ocorre a validação para garantir que está de acordo com a especificação.
- III. Evolução do software: o software precisa evoluir para atender às novas demandas dos stakeholders (cliente, usuário, mercado e etc).

A seguir serão abordadas duas metodologias de desenvolvimento, dentre as diversos existentes, sendo elas: modelo em cascata e o modelo ágil XP (Extreme Programming) .

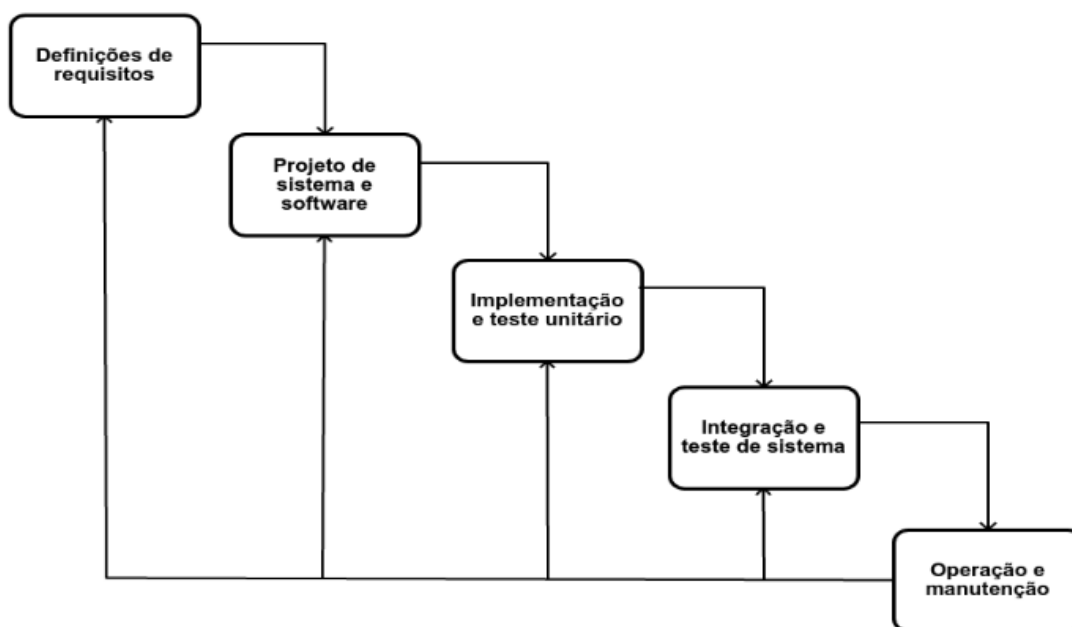
### **2.1.1 Modelo em Cascata**

O modelo em cascata descreve um método sequencial de desenvolvimento, no qual a execução de cada atividade fornecem feedbacks para as fases anteriores. As respostas recebidas de cada etapa do processo são utilizados pelos responsáveis e idealmente estimulam a melhoria e evolução do software (PETERS; PEDRYCZ, 2001).

Sommerville (2011, p.20) afirma que “os principais estágios do modelo em cascata refletem diretamente as atividades fundamentais do desenvolvimento”. O autor ainda destaca os principais estágios sendo:

- I. Análise e definição de requisitos;
- II. Projeto de sistema e software;
- III. Implementação e teste unitário;
- IV. Integração e teste do sistema;
- V. Operação e manutenção

Figura 1 - Modelo em cascata



Fonte: Modelo em cascata (SOMMERVILLE, 2011, p.20).

O modelo em cascata apresenta a sequência de atividades a serem executadas durante o processo de desenvolvimento de software. A cada atividade é gerada uma documentação completa, permitindo assim dar prosseguimento para as etapas posteriores.

Lawrence (2004) apud Stankiewicz (2017) afirma que:

[...] o modelo cascata é considerado como de alto nível, pois depende da compreensão dos envolvidos e a capacidade de prever situações na fase de desenvolvimento que possam colocar risco ao projeto com prazos, tecnologias, entre outros. Além disso, o ciclo de vida clássico gera documentações detalhadas de cada fase do sistema a ser construído, tornando assim, o sistema mais transparente e possibilita um melhor controle por parte dos gestores .

Sommerville (2011) afirma que o modelo em cascata é consistente com outros modelos de processos e por ter sua documentação produzida em cada etapa do processo, permite o monitoramento do progresso de acordo com o plano de desenvolvimento. Entretanto, salienta que este modelo deve ser usado quando os requisitos do sistema já são conhecidos, não possuindo alterações ao longo do projeto.

Quando os requisitos não são completamente conhecidos e/ou o projeto possa sofrer alterações radicais durante o desenvolvimento do projeto, o mesmo poderá apresentar consequências como, má estruturação, não atendimento das necessidades do cliente, bugs entre outros (SOMMERVILLE, 2011).

Melcher (2013) elenca algumas vantagens e desvantagens da adoção deste modelo:

Quadro 2 - Vantagens e desvantagens do modelo em cascata.

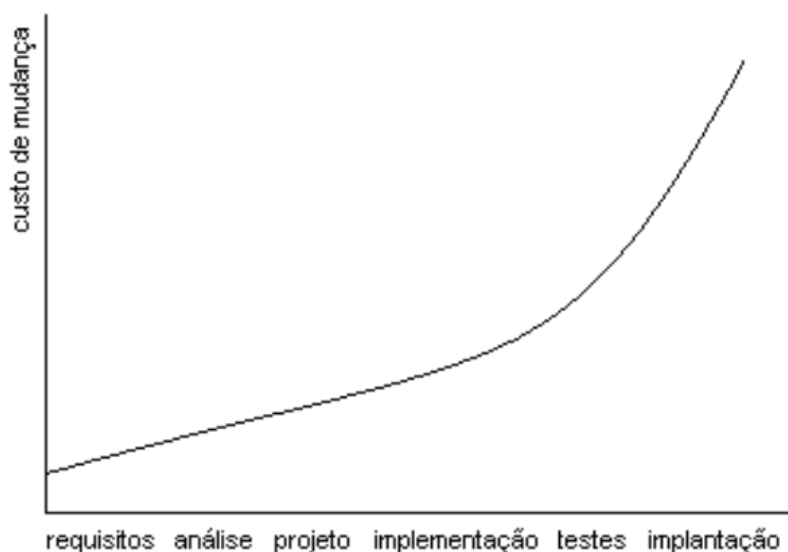
<b>Vantagens</b>	“Abordagem melhor do que um processo de desenvolvimento informal e de estrutura relativamente fácil de aprendizado na transição de um processo para o outro”;
	“Por ter etapas muito parecidas com as genéricas de qualquer modelo, facilita a adaptação de qualquer tipo de equipe e sistema desenvolvido em outro modelo”;
	“Documentação abrangente e mais detalhada do que em outras abordagens. Ideal para sistemas mais complexos”;
	“Indicada para projetos cujos requisitos de sistema não precisam ser alterados durante o desenvolvimento”.
<b>Desvantagens</b>	“Falta de agilidade no processo, devido ao encadeamento das fases”;
	“Documentação excessiva que pode ficar desatualizada e ser utilizada por poucos. Muito tempo envolvido em documentação”;
	“Pouca garantia se todo o escopo do projeto realmente será entregue no final, uma vez que o cenário pode mudar e erros cometidos numa fase podem atrasar o andamento das demais”;
	“Muitos atrasos em prazos e aumento considerável dos custos devido a erros não imaginados e necessidade de mudanças não detectadas previamente”;
	“Os requisitos devem ser fechados com uma antecedência grande, na qual muitas vezes o cliente ainda não tem a certeza necessária, o que pode gerar incertezas futuras”;
	“O cliente só poderá utilizar o produto muito tempo depois de ter contratado o serviço e, por isso, se quiser mudar algo, será tarde para reagir às mudanças do mercado”.

Fonte: Adaptado de Melcher (2013, p.42)

Muitos dos problemas do modelo em cascata, deve-se ao fato de possuir seus requisitos estáticos e podendo repercutir grandes impactos de custos, eficácia e qualidade, caso sofram alterações durante o seu desenvolvimento (MELCHER, 2013).

Soares (2004, p.2) apresenta uma figura com a variação de custo de e alteração no nos requisitos do projeto ao decorrer do processo.

Figura 2 - Custo de mudança no projeto no modelo cascata (custo x fases)



Fonte: Custo de mudança no projeto no modelo cascata (SOARES, 2004, p.2).

O gráfico do custo de mudança apresenta o custo da alteração de requisitos de software ao decorrer das várias etapas do desenvolvimento utilizando o modelo em cascata. Alterações nos requisitos no modelo em cascata são indesejáveis, pois o custo de alteração cresce exponencialmente de acordo com sua etapa, podendo ter um custo de sessenta a cem vezes, caso ocorra nas etapas finais do projeto (PRESSMAN, 2006; SOARES, 2004).

### 2.1.2 XP (Extreme Programming)

O XP (Extreme Programming) é uma abordagem desenvolvida com o objetivo de impulsionar as boas práticas de processos de desenvolvimento de software. Sendo a mais conhecida e talvez o mais utilizado e dos métodos ágeis (SOMMERVILLE, 2011; SOARES, 2004).

Pressman (2006, pg. 64) apud Jeffries (2002) “Extreme Programming é uma disciplina de desenvolvimento de software baseada em valores de simplicidade, comunicação, feedback e coragem”. Afirma ainda que este método inclui um conjunto de regras e práticas que ocorrem no contexto do planejamento, projeto, codificação e teste (PRESSMAN, 2006).

Figura 3 - Metodologia Extreme Programming



Fonte: O Processo *Extreme Programming* (PRESSMAN, 2006, p.64).

Diferente do modelo em cascata o processo XP aborda um fluxo contínuo ao invés de linear, como pode-se observar na figura 3. Soares (2004) afirma que o feedback constante, abordagem incremental e o encorajamento da comunicação entre as pessoas são algumas das principais diferenças em relação às outras metodologias.

Em *Extreming Programming*, os requisitos são expressos como cenários (chamados de estórias do usuário), que são implementados diretamente como uma série de tarefas. Os programadores trabalham em pares e desenvolvem testes para cada tarefa antes de escreverem o código. Quando o novo código é integrado ao sistema, todos os testes devem ser executados com sucesso. Há um curto intervalo entre os *releases* do sistema (SOMMERVILLE, 2004, pg. 44).

Este processo prevê entregas constantes, melhorando a comunicação entre as partes envolvidas e reduzindo os custos da inserção ou alteração de requisitos durante o processo.

Para Soares (2004, pg. 5) o XP é a metodologia ideal para ser utilizada quando as partes envolvidas “não sabem exatamente o que desejam e podem mudar de opinião durante o desenvolvimento do projeto”.

## 2.2 Engenharia de Requisitos

A engenharia de requisitos possui um papel fundamental durante o processo de desenvolvimento de um software, pois não importa quão bem codificado e testado seja um sistema, se o mesmo não atender as necessidades do usuário final o esforço foi ineficaz.

Para o desenvolvimento de um software bem sucedido é imprescindível uma completa compreensão dos seus requisitos, com isto, nota-se que a melhor solução de para satisfazer as necessidades do cliente e usuário final consiste em um processo sólido de engenharia de requisitos (LOPES, MAJDENBAUN, AUDY; 2003).

Sommerville (2011) define engenharia de requisitos sendo o processo de descobrir, analisar, documentar e verificar os serviços e restrições do sistema. Neste sentido, entende-se também que a engenharia de requisitos fornece mecanismos apropriados para facilitar as atividades de análise, documentação e verificação (LOPES, 2004).

Figueira (2012) aborda que “um dos principais objetivos da engenharia de requisitos é encontrar requisitos adequados para o sistema em desenvolvimento” (p. 16), e questiona o real significado de requisito. Para responder tal questionamento o autor traz uma série de definições.

Quadro 3 - Definição de requisitos segundo autores

<b>Autor</b>	<b>Definições</b>
<b>Tayer e Dorfmann (2000)</b>	Requisito é uma característica do software necessária para que o usuário e/ou cliente possa encontrar a solução de um problema de forma a atingir um objetivo.
<b>Sommerville (2007)</b>	Requisitos de um sistema são descrições dos serviços fornecidas pelo sistema e as suas restrições operacionais. Os requisitos demonstram as necessidades de um cliente de um sistema que ajuda a resolver um determinado problema.
<b>Lopes <i>apud</i> Siddiqi (1996)</b>	Requisitos são como uma declaração completa do que o software irá fazer sem referir-se a como fazê-lo.
<b>Lopes <i>apud</i> Kruchten (2000)</b>	Um requisito é como uma condição ou capacidade que um software deve realizar.
<b>Goguem (1997)</b>	Requisitos são propriedades que um software deve ter para funcionar com êxito no ambiente em que for utilizado.

Fonte: Figueira (2012, p.16).



Diante as definições apresentadas por Figueira (2012), entende-se que requisitos de um sistema expressam as descrições das suas funcionalidades, ou seja as características necessárias para para atender as necessidades dos clientes e/ou usuários.

Os passos da engenharia de requisitos são a concepção, levantamento, elaboração, negociação, especificação e validação. tendo como objetivo fornecer aos stakeholders um entendimento por escrito do projeto ou problema (PRESSMAN, 2006).

Quadro 4 - Etapas da engenharia de requisitos

<b>Etapas</b>	<b>Definições</b>
<b>Concepção</b>	Definição do escopo do projeto.
<b>Levantamento</b>	Definição dos requisitos básicos do projeto.
<b>Elaboração</b>	Análise dos requisitos.
<b>Negociação</b>	Alinhamento das reais necessidades para que o objetivo do projeto sejam obtidos e definição de prioridades.
<b>Especificação</b>	Documentação dos requisitos.
<b>Validação</b>	Validação realizada pelo demandante, tendo como finalidade de garantir que a especificação atenderá suas necessidades.

Fonte: Adaptado a partir de Pressman (2006, pg. 116).

Peters e Pedrycz (2001, pg. 102) afirma que os principais resultados da análise de requisitos são:

- **Funcional (ações principais).** Uma descrição funcional identifica as atividades do sistema.
- **Comportamental (atividades de controle).** Uma descrição comportamental descreve uma sequência possível sobreposição das funções do sistema em uma hierarquia de atividades de controle.
- **Não comportamental (atributos).** Uma descrição não-comportamental do software inclui planejamento de engenharia humana e de garantia de qualidade.

Peters e Pedrycz (2001, pg. 102) afirma ainda que os principais produtos são:

- **Especificação de requisitos de software completa.** Uma descrição de um sistema (suas funções, seu comportamento, seu desempenho, suas interfaces interna e externa e seus atributos de qualidade).
- **Plano de garantia de qualidade.** Uma indicação da portabilidade, eficiência, confiabilidade, critérios validação e verificação, custos e critérios de aceitação a serem seguidos pelas equipes de projeto.

O objetivo da engenharia de requisitos é de fazer a ponte entre o demandante do sistema e a equipe de desenvolvimento. Com isto a análise de requisitos apresenta-se como uma das partes mais importantes do processo de desenvolvimento, uma vez que os requisitos mal levantados pode gerar bugs e/ou o não atendimento das necessidades do cliente.

## 2.3 Documentação de Requisitos

Ao se estudar os processos de desenvolvimento, nota-se que a elicitación de requisitos de um sistema é um ponto em comum entre maioria deles, essa necessidade deve-se ao fato de que codificar, estimar e gerenciar um projeto sem saber seus requisitos pode ser comparado como trabalhar com uma venda nos olhos.

Desta forma fica evidente que sua documentação possui igual importância, pois o documento contendo os diversos artefatos gerados pela engenharia de requisitos servirá como apoio para o demandante do software, tanto quanto para a equipe de gerenciamento e desenvolvimento.

Sommerville (2007, p. 91) afirma que “o documento de requisitos possui um conjunto diversificado de usuários, desde a gerência sênior da organização, que para pelo sistema, até os engenheiros responsáveis pelo desenvolvimento do software”. Sommerville (2007) ainda apresenta uma descrição dos usuários envolvidos:

- I. Clientes do sistema: especificam e lêem os requisitos para verificar se eles atendem às suas necessidades. Os clientes especificam as mudanças nos requisitos;
- II. Gerentes: Usam o documento de requisitos para planejar um pedido de proposta para o sistema e planejar o processo de desenvolvimento do sistema;
- III. Engenheiros de sistema: usam os requisitos para compreender qual sistema será desenvolvido;
- IV. Engenheiros de teste do sistema: usam os requisitos para desenvolver testes de validação para o sistema;
- V. Engenheiros de manutenção de sistema: usam os requisitos para compreender o sistema e os relacionamentos entre suas partes.

A ISO/IEC/IEEE 29148:2011(E) define os termos e apresenta os processos de engenharia de requisitos necessários para o desenvolvimento de sistema. Recomenda também uma série de elementos que devem compor a especificação de requisitos.

Sommerville (2007, p. 92) aborda que “embora o padrão do IEEE não seja o ideal, ele contém uma grande quantidade de boas recomendações de como redigir requisitos e evitar problemas. Ele é muito geral para funcionar como padrão de uma organização”. A partir de tal afirmação o autor apresenta um quadro ilustrando a estrutura de um documento de requisitos baseado no padrão IEEE 29148:2011(E).

Quadro 5 - Estrutura de documento de requisitos

Capítulo	Descrição
<b>Prefácio</b>	Deve definir o público-alvo do documento e descrever seu histórico de versões, incluindo uma justificativa lógica para a criação de nova versão e um resumo das mudanças feitas em cada versão.
<b>Introdução</b>	Deve descrever a necessidade do sistema. Deve descrever basicamente as funções e explicar como o sistema irá funcionar como outros sistemas. Deve descrever como sistema atende aos objetivos gerais de negócios e estratégicos da organização que encomendou o software.
<b>Glossário</b>	Deve definir os termos técnicos usados no documento. Você não deve fazer suposições sobre a experiência ou habilidade do leitor.
<b>Definições de requisitos de usuário</b>	Os serviços fornecidos ao usuário e os requisitos não funcionais do sistema devem ser descritos nesta seção. Essa descrição pode usar linguagem natural, diagramas e outras notações compreensíveis pelos clientes. Padrões de produto e de processo a serem seguidos devem ser especificados.
<b>Arquitetura de sistema</b>	Este capítulo deve apresentar uma visão geral de alto nível da arquitetura prevista do sistema, mostrando a distribuição das funções nos módulos do sistema. Os componentes de arquitetura reusados devem ser destacados.
<b>Especificação de requisitos do sistema</b>	Deve descrever os requisitos funcionais e não funcionais mais detalhadamente. Caso necessário, mais detalhes também podem ser adicionados aos requisitos não funcionais; por exemplo, interfaces com outros sistemas devem ser definidos.
<b>Modelos de sistema</b>	Deve estabelecer ou um ou mais modelos de sistema, mostrando os relacionamentos entre os componentes e o sistema e seu ambiente. Podem ser modelos de objetivos, modelos de fluxos de dados e modelos semânticos de dados.
<b>Evolução de sistema</b>	Deve descrever hipóteses fundamentais sobre as quais o sistema está baseado, além de mudanças previstas devido à evolução do hardware, mudança das necessidades do usuário etc.
<b>Apêndices</b>	Deve descrever informações detalhadas e específicas relacionadas a aplicação que está sendo desenvolvida. Exemplos de apêndices que podem ser incluídos são descrições de hardware e de banco de dados. Os requisitos de hardware definem as configurações mínimas e ideal para o sistema. Os requisitos de banco de dados definem a organização lógica dos dados usados pelo sistema e os relacionamentos entre os dados.
<b>Índice</b>	Podem ser incluídos diversos índices para o documento. Assim como um índice alfabético normal, pode haver índice dos diagramas, índice das funções etc.

Fonte: Sommerville (2007, p. 93).

Baseado na estrutura proposta pelo IEEE 29148:2011(E), Sommerville et al, e nos artefatos gerados no processo de engenharia de requisitos diversos autores elaboraram templates para padronizar a documentação de requisitos.

Com base no quadro 5 apresentado por Sommerville (2007) foi realizado uma verificação em quatro modelos de documentos de requisitos.

Quadro 6 - Análise de documentos de requisitos

Tópicos	Modelo 1 (Anexo I)	Modelo 2 (Anexo II)	Modelo 3 (Anexo III)	Modelo 4 (Anexo IV)
Prefácio	x	x		x
Introdução	x	x	x	x
Glossário	x	x	x	x
Definição de requisitos do usuário	x	x	x	x
Arquitetura do sistema	x	x	x	x
Especificação de requisitos do sistema	x	x	x	x
Modelos de sistemas	x	x	x	x
Evolução do sistema	x	x	x	x
Apêndices	x	x	x	x
Índices	x	x	x	x

Fonte: Elaborado pelo autor a partir dos modelos pesquisados

Apesar de a maioria dos modelos analisados contemplarem todos os tópicos apresentados por Sommerville, eles estão estruturados e organizados de formas diferentes. Com base nos modelos, observa-se que a relação entre os elementos de interface com o usuário, *backend* e banco de dados fica dependente de marcações e referenciamento realizado de forma manual pelo analista.

### 3. PROCEDIMENTOS METODOLÓGICOS

Este trabalho em termos de classificação de pesquisas é de **natureza** aplicada, pois tem como **objetivo** geral construir um modelo simplificado de documentação de requisitos de software que contemple de metodologias tradicionais e ágeis. Para que isto seja possível, tem-se os objetivos específicos de realizar o agrupamento de modelos teórico-práticos de forma consistente e lógica do ponto de vista do usuário e desenvolvedor, elaborar um template de documentação, validar o template de documentação de requisitos e aperfeiçoar o template com base nas sugestões oferecidas.

A pesquisa aplicada consiste na utilização de conhecimentos já sistematizados, com o intuito de uma aplicação em situações específicas (ALMEIDA, 2014; GIL, 2010). Neste sentido Vianna (2013, p. 1) contribui ao afirmar que uma pesquisa de natureza aplicada possui “finalidades imediatas, gera produtos e/ou processos”.

Em um trabalho científico devem-se ser estabelecidos os procedimentos metodológicos, os quais estão relacionados a escolha criteriosa e sistêmica para elaborar a descrição, explicação, e análise de fatos e fenômenos. Tais procedimentos podem ser realizado por meio de uma abordagem qualitativa, quantitativa ou ambas (OLIVEIRA, 2012).

Uma abordagem qualitativa de acordo com Van Maanen (1979) apud Miguel et al (2012, p. 53) possui como foco os processos do objeto de estudo, podendo ser vista como “um guarda-chuva que abriga uma série de técnicas de interpretação que procuram descrever, decodificar, traduzir, e qualquer outro termo relacionado com o entendimento das variáveis de determinado fenômeno”.

Já uma abordagem quantitativa realiza o tratamento de dados utilizando métodos e ferramentas estatísticas, buscando quantificar os dados obtidos, permitindo assim maior precisão na explicação quantitativa dos fenômenos e correlação entre variáveis, que são previamente estabelecidas (ALMEIDA, 2014; OLIVEIRA, 2012).

A partir destas definições, este trabalho apresenta uma abordagem qualitativa, uma vez que busca analisar, entender e descrever os fenômenos que estão em torno do objeto de estudo.

Oliveira (2012, p. 58) esclarece ainda que uma abordagem qualitativa “se preocupa com uma visão sistêmica do problema ou objeto de estudo”. Neste sentido pode-se caracterizar como sendo um estudo detalhado do objeto, visando buscar “informações

fidedignas para se explicar em profundidade o significado e as características de cada contexto em que encontra o objeto de pesquisa” (p. 60).

O avanço tecnológico possibilitou o acesso a uma grande massa de informações, ou seja, a obtenção de pesquisas científicas, anais e revistas entre muitas outras fontes de conhecimento podem ser encontrados em formato digital. Outra contribuição proporcionado por este avanço foi a catalogação dos acervos das bibliotecas, facilitando a localização de autores ou títulos de temas específicos.

Com isto será realizado uma pesquisa bibliográfica e documental para coletar, analisar e expor os temas do objeto de estudo. Desta forma será possível elaborar o referencial teórico sobre processos de desenvolvimento, análise de requisitos e documentação de requisitos, possibilitando ainda a análise de diversos modelos de documentos utilizados na elicitação de requisitos e elaborar de um modelo padrão, para utilização no decorrer do projeto.

Uma pesquisa bibliográfica é a base para as demais pesquisas, uma vez que tem como objetivo proporcionar o saber, direcionando o pesquisador à pesquisas de determinado assunto (FACHIN, 2006).

Koche (2013, p. 122) aborda que uma pesquisa bibliográfica consiste na utilização dos conhecimentos disponíveis a partir das publicações existentes na tentativa de explicar um problema e/ ou para o entendimento de determinado tema. Desta forma o autor destaca que uma pesquisa bibliográfica pode ser utilizada com diferentes fins:

- I. para ampliar o grau de conhecimento em determinada área, capacitando o investigador a compreender ou delimitar melhor um problema de pesquisa;
- II. para dominar o conhecimento disponível e utilizá-lo como base ou fundamentação na construção de um modelo teórico explicativo de um problema, isto é, como instrumento auxiliar para a construção e fundamentação das hipóteses;
- III. para descrever ou sistematizar o estado da arte, daquele momento, pertinente a determinado tema ou problema.

Já a pesquisa descritiva tem como finalidade observar, registrar e analisar o objeto de estudo e suas características com o máximo de exatidão possível (ALMEIDA, 2014; CERVO e BERVIAN, 2002). Vianna (2013, p. 1) salienta que “Registra e descreve os fatos observados sem interferir neles. Descreve as características de determinada população ou fenômeno; estabelece relações entre variáveis”.

Para validação do template proposto foi realizado a aplicação de um questionário utilizando a ferramenta de formulários da google. O questionário pode ser compreendido

como um conjunto de questões organizadas, tendo como objetivo coletar informações de um conjunto de pessoas (FACHIN, 2006).

O questionário é considerado um instrumento de coleta de dados construído por uma série ordenada de perguntas descritivas (perfis socioeconômicos, como renda, idade, escolaridade, profissão e outros), comportamentais (padrões de consumo, de comportamento social, econômico e pessoal, dentre outros) e preferenciais (opinião e avaliação de alguma condição ou circunstâncias) (ZANELLA, 2007 *apud* ALMEIDA, 2014, pg. 62).

Dencker e Viá (2012, pg. 162) elenca uma série de vantagens do questionário, sendo elas:

- I. é menos dispendioso que a entrevista, pois requer menos envolvimento do pesquisador.
- II. exige menor habilidade do pesquisador, podendo até mesmo dispensar sua presença nos casos em que pode ser enviado pelo correio, por e-mail ou por outros meios de contato.
- III. pode ser aplicado a um número maior de pessoas, aumentando a representatividade das amostras.
- IV. a impessoalidade, característica da forma de aplicação, atua no sentido aumentar a uniformidade da situação de mensuração.
- V. a confiança no anonimato é maior, uma vez que o entrevistado possui o controle do registro das respostas, podendo levar ao fornecimento de informações mais francas.

O com o objetivo de entender o perfil do respondente, coletar sua opinião documentação de requisitos e validar o template proposto, foi elaborado um questionário não identificado, constituindo de três questões de múltipla escolha, um conjunto com quatro questões agrupadas em matriz conhecida como escala de avaliação e utilizado o conceito de escala likert e três questões abertas, somando um total de dez questionamentos.

O envio foi realizado via e-mail para programadores de três empresas de desenvolvimento de software na data de 13 de setembro de 2019 e encerrando o período para respostas no dia 25 de setembro de 2019. Com isto foram obtidos um total de nove respostas.

## 4. TEMPLATE DE DOCUMENTO DE REQUISITOS REQUISITOS E VALIDAÇÃO

Este capítulo tem como objetivo de apresentar e validar o template proposto na utilização e especificação do framework de documentação de requisitos, para isto são abordados três tópicos, sendo eles: template de documentação de requisitos, validação do template de documentação de requisitos e proposta de um framework para automatização de documentação de requisitos.

### 4.1 Template de documento de requisitos

O modelo proposto é uma adaptação dos modelos existentes de documentação e baseando-se nas considerações abordadas por Sommerville, conforme apresentado no quadro 5. O template de documentação de requisitos apresenta os tópicos de prefácio, introdução, glossário, definições de requisitos de usuário, arquitetura de sistema, especificação de requisitos do sistema, modelos do sistema, evolução de sistema, apêndices e índice.

Figura 4 - modelo de documento de requisitos (parte 1)

#### 1. Introdução

<Descrição geral do que se propõe com o sistema>

#### Definições e abreviações

Termo	Descrição

#### 2. Atores:

- a. [ATR01] <ator>  
<Breve descrição...>
- b. [ATR02] <ator>  
<Breve descrição...>

Figura 4. Elaborado pelo autor



A figura 4 apresenta o tópico inicial do documento de requisitos, contemplando as etapas de prefácio, introdução e glossário. Este tópico inicia-se com a introdução na qual realiza-se a descrição dos objetivos, esclarecimento do escopo e versionamento do projeto. Para tal a introdução deve ser elaborada de forma objetiva e concisa, permitindo o entendimento prévio do projeto pelas partes envolvidas.

O tópico de definições e abreviações contempla a parte de glossário. De acordo com Sommerville (2007, p.93) o glossário “deve definir os termos técnicos usados no documento. Não deve fazer suposições sobre a experiência ou habilidade do autor”.

Para finalizar a parte de introdutória faz-se o levantamento dos atores, ou seja, todas as partes envolvidas como patrocinadores, gestores, usuários entre outros. No levantamento dos atores contempla também um descritivo sobre o papel de cada um dentro do projeto.

Figura 5 - modelo de documento de requisitos (parte 2)

### 3. Caso de Uso:

#### 3.1 [UC01] Caso de Uso: <Nome do caso de uso com verbo no infinitivo>

<Descrição do que se propõe com o caso de uso. Deve-se comentar de forma bem resumida qual é o objetivo deste documento>

##### 3.1.1 [UC01-PRT01] Título

<Inserir imagem quando houver>

##### **Descritivo**

<Descritivo da imagem. No descritivo deverá conter informações sobre todos os campos apresentados, juntamente com máscaras, validações e etc.

Observação: pode-se preencher o descritivo quando não houver imagem. Neste caso, entende-se que deverá existir uma tela contemplando as informações descritas.>

##### a. [UC01-PRT01-C1] Título

**Cenário:** *Os cenários dos protótipos deverão conter os fluxos da tela: gravar, editar, excluir e regras.*

**Dado:**

**Quando:**

**Então:**

A figura 5 apresenta o tópico de requisitos funcionais, interfaces com o usuário, artefatos de tela e estrutura de dados. O template propõe que os requisitos funcionais sejam abordados em forma de caso de uso. Cada um dos casos podem conter artefatos de tela, no qual seu descritivo contempla a interface de comunicação com o usuário e estrutura de dados.

Logo após a apresentação da interface com o usuário o modelo proposto sugere que seja descrito os casos de uso e suas regras de negócio. Para transcrição pode ser utilizado qualquer forma de modelagem como o UML (unified modeling language), BDD (test driven development), por cenários e/ou outros. Por fim deve ser exposto a estrutura de dados, no qual assim como a descrição dos requisitos funcionais podem ser representados por qualquer um dos modelos existentes.

Nesta especificação deve ser descrito também o relacionamento entre os componentes, sistema e seu ambiente. Deve contemplar também a arquitetura de sistema na qual “deve apresentar uma visão geral de alto nível da arquitetura prevista do sistema, mostrando a distribuição das funções nos módulos do sistema. Os componentes de arquitetura reusada devem ser destacados” (SOMMERVILLE, 2007, pg.93).

Figura 6 - modelo de documento de requisitos (parte 3)

## 4. Estrutura de dados:

(Quando houver)

- a. [ED01] Título
- b. [ED01] Título

## 5. Requisitos Não Funcionais

*<Relatar todos os requisitos não funcionais como infraestrutura e condições para execução>*

- a. [RN01] Título

## 6. Plano de Testes

*<Definir>*



Fonte: Elaborado pelo autor.

A figura 6 foca na apresentação da estrutura de dados, requisitos não funcionais e plano de testes.

#### 4.2 Validação do template de documentação de requisitos

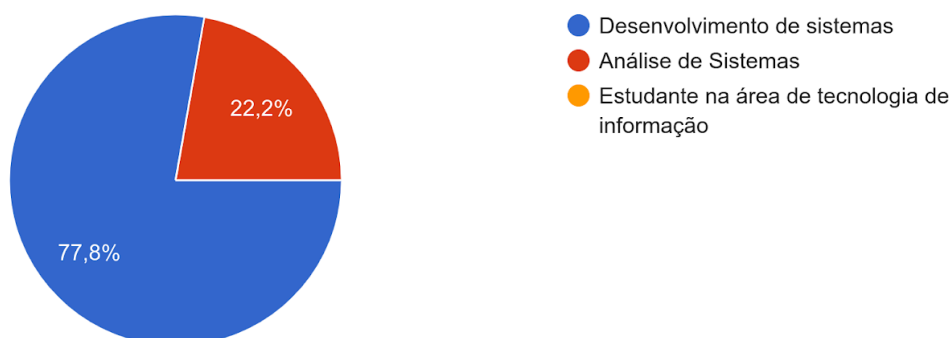
Para validação do template de documentação de requisitos proposto no presente trabalho foi realizado um questionário ordenando composto por dez questões e enviado por email para três empresas de desenvolvimento de software na data de 13 de setembro de 2019 e encerrando o período para respostas no dia 25 de setembro de 2019.

Com isto foram obtidos nove respostas com o objetivo de entender o perfil do respondente, coletar sua opinião documentação de requisitos e validar o template proposto.

Figura 7 - Área de atuação

##### Área de atuação

9 respostas



Fonte: Elaborado pelo autor

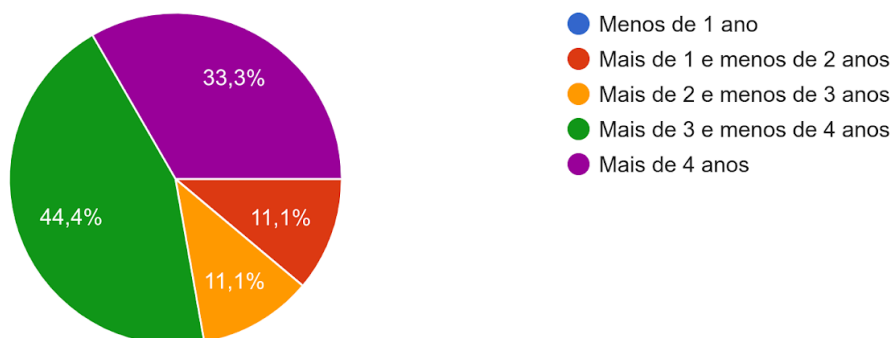
Na figura 7 realiza o questionamento sobre a área de atuação dos respondentes, sendo sete atuante na área de desenvolvimento e dois em análise de sistemas. O questionamento permite o entendimento sobre o papel atuante de cada respondente no ciclo de vida do projeto.

Com isto nota-se que 22,2 % dos respondentes atuam na coleta, análise e documentação de requisitos e 77,8% atuam na área de compreensão e desenvolvimento do projeto.

Figura 8 - Tempo de atuação na área de tecnologia da informação

### Tempo de atuação na área de Tecnologia da Informação

9 respostas



Fonte: Elaborado pelo autor

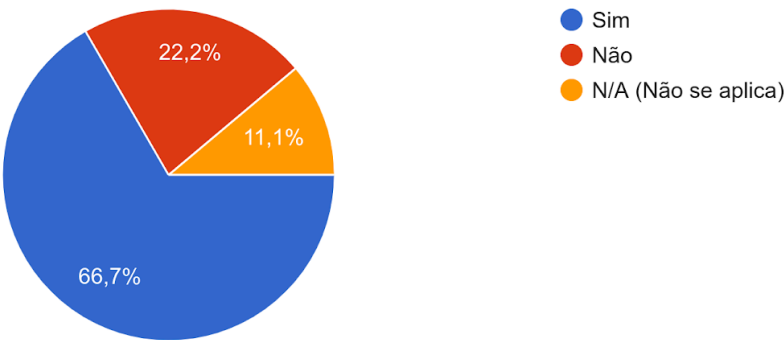
Na figura 8 busca verificar o tempo de atuação de cada participante do questionário na área de atuação da tecnologia da informação, das respostas obtidas mostram que três dos respondentes atuam a mais de quatro anos, quatro atuam entre três e quatro, um atua entre dois e três e um entre um e dois anos.

Com as respostas obtidas permite observar que 77,7 % dos colaboradores que responderam o questionário possuem mais de três anos de experiência na área de tecnologia da informação e 22,2 % possuem menos de três anos.

Figura 9 - Contato com documento de requisitos

Já trabalhou, utilizou e/ou teve contato com documento de requisitos

9 respostas



Fonte: Elaborado pelo autor

O questionamento apresentado na figura 9 aborda sobre o contato que os participantes possuem com documento de requisitos. Com isto torna-se possível realizar uma análise individual das respostas, verificando as respostas de quem já teve e de quem está realizando o primeiro contato com a documentação de especificação de requisitos.

Dos respondentes 66,7 % já trabalhou, utilizou e/ou teve contato com documento de requisitos e 22,2 % não teve contato com a documentação.

Quadro 7 - Matriz de questões

Questão	Discordo totalmente		Discordo em parte		Nem concordo e nem discordo		Concordo parcialmente		Concordo totalmente	
	Qt	%	Qt	%	Qt	%	Qt	%	Qt	%
Com base no exemplo especificado, é possível realizar o desenvolvimento do FRONTEND?			1	11,11%			3	33,33%	5	55,56%
Com base no exemplo especificado, é possível realizar a estruturação do BANCO DE DADOS?			2	25%	2	25%	1	12,5%	3	37,50%
Com base no exemplo especificado, é possível realizar o desenvolvimento do projeto como um todo,			1	11,11%	1	11,11%	4	44,44%	3	33,33%

realizando a conexão entre frontend, backend e banco de dados?										
Você se considera um usuário experiente em documentação de sistemas?	1	11,11%			3	33,33%	3	33,33%	2	22,22%

Fonte: Elaborado pelo autor

O quadro 7 apresenta questionamentos sobre o entendimento do projeto exemplo apresentado no questionário de validação do documento, o qual tem como objetivo verificar se o template cumpre o objetivo de levar o entendimento do sobre o projeto as partes envolvidas.

Com isto foi abordado quatro questionamento e na obtenção das respostas utilizou-se a escala likert com cinco pontos, sendo 1 para discordo totalmente, 2 para discordo em parte, 3 para nem concordo e nem discordo, 4 para concordo parcialmente e 5 para concordo.

A primeira questão buscou compreender com base no exemplo especificado se é possível realizar o desenvolvimento do frontend ou interface com o usuário. Em relação ao primeiro questionamento pode-se analisar que a moda foi na resposta “concordo” atingindo um total de 55,56% dos respondentes, 33,33% concordam parcialmente e 11,11% discorda em parte.

A segunda questão buscou compreender com base no exemplo especificado se é possível realizar a estruturação do banco de dados. De acordo com as respostas 37,50% concordam, 12,5% concordam parcialmente, 25% nem concorda e nem discorda e 25% discordam em parte.

O terceiro questionamento buscou entender sobre o planejamento e desenvolvimento do projeto como um todo, relacionando o frontend, backend e banco de dados. em relação a este questionamento 33,33% concordam, 44,44% concordam parcialmente, 11,11% nem concorda e nem discorda e 11,11% discorda em parte.

A quarta e última questão do quadro busca compreender se os respondentes se consideram um usuário experiente de documentação de sistemas. Neste questionamento 22,22% concordam, 33,33% concordam parcialmente, 33,33% nem concordam e nem discordam e 11,11% discorda totalmente.

Por fim foi aplicado três questões abertas sendo elas:

- I. Quando se trata de documentação de sistemas e especificação de requisitos, o que você considera como sendo as maiores dificuldades?
- II. Considere quais dois pontos fortes que o documento apresenta.
- III. Considere dois pontos fracos que podem ser melhorados no documento.

Figura 10 - Dificuldades de documentação de sistemas e especificação de requisitos.

**Quando se trata de documentação de sistemas e especificação de requisitos, o que você considera como sendo as maiores dificuldades?**

8 respostas

Entender de maneira detalhada a interface de comunicação entre os componentes do sistema(backend, banco de dados, frontend)
Estruturação de banco de dados e prever todas as possíveis alterações futuras do sistema.
Representar a necessidade real do cliente e ainda ordenar por ordem de prioridade os requisitos.
Entender o cliente
Falta de detalhamento dos fluxos; Falta de diagramas; Falta de colaboração/comunicação até o fechamento das especificações
Manter atualizada a documentação
Detalhes e Exceções
Casos de uso, diagrama de classe e modelagem do DB

Fonte: Elaborado pelo autor

No questionamento apresentado na figura 10 obteve oito respostas, e qual teve como objetivo entender qual as maiores dificuldades de uma documentação de sistemas e especificação de requisitos na ótica dos respondentes. Dentre as respostas obtidas destaca-se o real entendimento das necessidades do cliente, comunicação e relacionamento entre as diversas partes do sistema e por fim o versionamento e atualização do documento.

Diante aos tópicos ressaltados podem ser trabalhados com um modelo de documento conciso. Já as demais respostas ressaltadas abordam sobre os artefatos gerados durante o processo de análise e especificação de requisitos.

Figura 11 - Pontos fortes da documentação de sistemas e especificação de requisitos.



## Considere quais dois pontos fortes que o documento apresenta

7 respostas

Protótipos de tela e especificação de endpoints de uma api
Endpoints e BDD
Visão da tela e descrição da API REST
Codificação das chamadas REST, as ações tomadas de decisão também estão bem descritas
Cenários e fluxos descritos
Bem detalhado, clareza na descrição dos pontos.
Casos de usos com endpoints e o retorno das status HTTP. O plano de testes também é um ponto forte

Fonte: Elaborado pelo autor.

A segunda questão aberta tem como objetivo compreender os pontos fortes do modelo de documento apresentado. Neste questionamento houve sete respostas, dentre elas destaca-se a interface com usuário e sua interação com as demais partes do sistema, a estrutura de casos de uso em BDD e a clareza na descrição dos pontos.

Figura 12 - Pontos fracos da documentação de sistemas e especificação de requisitos.

## Considere dois pontos fracos que podem ser melhorados no documento.

7 respostas

Falta de relação de entidades de banco de dados, casos de uso necessitam de um pouco mais de explicação sobre o problema que o sistema resolve.
Design das telas e relacionamento do banco de dados
Design de tela, regras de validação de campos
As ações do sistema podem ser melhor vistas se forem feitas em formato de diagrama de caso de uso. Especificar os testes, em testes de usabilidade, unitários e integração
Falta o modelo ER (não está claro quais serão todos os campos do usuário); Falta definir (ou deixar claro que está em aberto) o uso de tecnologias front-end (jQuery, SPA), validações; E de tecnologias back-end (frameworks JPA, etc).
Falta definir casos de teste válidos e inválidos.
mais detalhes no testes.
Talvez a falta de um diagrama de classes e a modelagem do DB

Fonte: Elaborado pelo autor

Já no terceiro questionamento aberto tem como objetivo compreender os pontos fracos do modelo de documento apresentado. Na última questão conforme apresentado na figura 12 também houve sete respostas, diante as respostas destaca-se o melhor detalhamento de interface como regras de validações e suas interações, melhor detalhamento dos cenários dos casos de uso, requisitos não funcionais e testes. Os tópicos referentes a banco de dados e design de tela englobam o escopo de artefatos gerados durante análise e especificação de requisitos.

### 4.3 Otimização do template de documentação de requisitos

Após análise do questionário realizado foi realiza alguns ajustes no modelo de documentação de requisitos proposto, tendo como finalidade realizar melhorias quanto ao seu entendimento.

Figura 13 - Documentos de requisitos (parte 1)

# 1. Introdução

<Deve descrever a necessidade do sistema. Deve descrever basicamente as funções e explicar como o sistema irá funcionar como outros sistemas. Deve descrever como sistema atende aos objetivos gerais de negócios e estratégicos da organização que encomendou o software>

## 1.1 Versões

Versões	Descrição

<Deve definir o público-alvo do documento e descrever seu histórico de versões, incluindo uma justificativa lógica para a criação de nova versão e um resumo das mudanças feitas em cada versão>

## 1.2 Definições e abreviações

Termo	Descrição

<Deve definir os termos técnicos usados no documento. Você não deve fazer suposições sobre a experiência ou habilidade do leitor.>

## 1.3 Atores:

- a. [ATR01] <ator>  
<Breve descrição...>
- b. [ATR02] <ator>  
<Breve descrição...>

Fonte: Elaborado pelo autor

Conforme as respostas obtidas pelo questionário, uma das grandes dificuldades é de entender as diversas versões do documento e qual objetivo de cada uma. Neste sentido foi incluído uma tabela para inserir o histórico de versões juntamente com um descritivo dos principais objetivos a serem alcançados em cada versão.

Figura 14 - Documentos de requisitos (parte 2)

## 2. Caso de Uso:

### 2.1 [UC01] Caso de Uso: <Nome do caso de uso com verbo no infinitivo>

<Descrição do que se propõe com o caso de uso. Deve-se comentar de forma bem resumida qual é o objetivo deste documento>

#### 2.1.1 [UC01-PRT01] Título

<Inserir imagem quando houver>

##### Descritivo

<Descritivo da imagem. No descritivo deverá conter informações sobre todos os campos apresentados, juntamente com máscaras, validações e etc.

Observação: pode-se preencher o descritivo quando não houver imagem. Neste caso, entende-se que deverá existir uma tela contemplando as informações descritas.>

#### a. [UC01-PRT01-C1] Título

**Cenário:** *Os cenários dos protótipos deverão conter os fluxos da tela: gravar, editar, excluir e regras.*

**Dado:**

**Quando:**

**Então:**

#### 2.1.1.1 [UC01-ED01]

*Estrutura de dados*

#### 2.1.2 [UC01-PRT02] Título

<Inserir imagem quando houver>

##### Descritivo

<Descritivo da imagem. No descritivo deverá conter informações sobre todos os campos apresentados, juntamente com máscaras, validações e etc.

Observação: pode-se preencher o descritivo quando não houver imagem. Neste caso, entende-se que deverá existir uma tela contemplando as informações descritas.>

Fonte: Elaborado pelo autor

A figura 14 apresenta a descrição do caso uso, no qual é sugerido a utilização da metodologia BDD (*Behavior Driven Development*) para descrição dos requisitos funcionais, alternativos e de exceção. Apresenta ainda a inclusão da estrutura de dados como um subtópico dentro do caso de uso, facilitando o entendimento de quais cenários estão sendo representados.

#### 4.4 Proposta de um *metaframework*

Este capítulo tem como objetivo apresentar parâmetros para a realização de um planejamento e construção de uma ferramenta para documentação de requisitos. Tais parâmetros busca facilitar tanto no processo de análise e documentação quanto em sua legibilidade e compreensão.

Os tópicos do *metaframework* proposto utilizam como base as recomendações apresentadas por Sommerville, conforme exposto como no quadro 5, e também o modelo de documentação abordado no item 4.1, levando em consideração as observações apresentadas no questionário de validação.

Quadro 8 - Parâmetros para o planejamento de uma ferramenta para documentação de requisitos

Capítulo	Descrição
<b>Prefácio</b>	Para o prefácio recomenda-se a construção de uma interface que permita a inclusão de usuários (partes envolvidas) e descrição de seu papel atuante dentro do projeto, permitindo referenciar-los em suas etapas.  Quanto ao versionamento, sugere-se que qualquer edição após a finalização do projeto seja gerado um novo projeto herdando a especificação anterior, indicando uma referencia as alterações, seja de inclusão, edição e/ou exclusão. Toda nova versão deve conter um breve resumo sobre sua finalidade.
<b>Introdução</b>	Referente a introdução deve-se ser elencados por tópicos as necessidades do sistema, contendo em cada um deles um descritivo do seu resultado esperado.
<b>Glossário</b>	Deve definir os termos técnicos usados no documento.
<b>Definições de requisitos de usuário</b>	Referente aos serviços oferecidos ao usuário, deve-se permitir a inclusão de anexos para protótipos de telas e/ou qualquer outro artefato gerado no processo de análise. Deve ser possível também fazer um descritivo e indicar componentes elaborados na etapa de arquitetura de sistema.
<b>Arquitetura de sistema</b>	Referente a arquitetura dos sistema, sugere-se a criação de uma interface que permita a criação de componentes, com regras, validações, anexos e etc. Possibilitando ainda reutilizar estes componentes durante a documentação das etapas do projeto.
<b>Especificação de</b>	Neste tópico deve-se incluir os requisitos funcionais em casos de uso, neste caso

<b>requisitos do sistema</b>	aconselha-se a utilização da metodologia BDD ( <i>test driven development</i> ). Deve conter também as regras de negócio.
<b>Modelos de sistema</b>	Deve estabelecer um ou mais modelos de sistema, mostrando os relacionamentos entre os componentes e o sistema e seu ambiente. Podem ser modelos de objetivos, modelos de fluxos de dados e modelos semânticos de dados.
<b>Evolução de sistema</b>	Este tópico destina-se a descrição de requisitos não funcionais, neste caso é aconselhado permitir inserir em os itens de forma ordenada, contendo uma identificação e podendo referência ou ser qualquer item especificado.  Deve também, possibilitar a descrição de hipóteses fundamentais sobre as quais o sistema está baseado, além de mudanças previstas devido à evolução do hardware, mudança das necessidades do usuário etc.
<b>Apêndices</b>	Deve ser possível inserir os pré e pós requisitos do sistema e de cada caso de uso.  Deve ser possível também, inserir os requisitos de banco de dados que definem a organização lógica dos dados usados pelo sistema, os relacionamentos entre os dados e os artefatos gerados no processo de análise.
<b>Índice</b>	Quanto ao índice, sugere-se a construção de um sumário que facilite a navegação e localização dos itens e tópicos no documento.

Fonte: Elaborado pelo autor

Com base nos parâmetros e sugestões apresentadas no quadro 8 e levando em consideração as respostas obtidas no questionário de validação do modelo de documentação de requisitos, foi elencado uma série de tópicos que carecem de atenção especial, tanto no planejamento da ferramenta quando no decorrer do processo de análise. Sendo eles:

- I. Interface com o usuário: elaboração de representações visuais de interface com o usuário, como os wireframes, protótipos, mock ups e descrição do comportamento de tela.
- II. Cenários: descritivo dos cenários dos casos de uso, permitindo assim que o leitor compreenda a necessidade do cliente em relação a funcionalidade descrita.
- III. Estruturação de dados: apresentação dos relacionamentos entre os dados, endpoints e outros.
- IV. Integração: descritivo da interação entre as partes do sistema

- V. Versionamento: atualização do documento de requisitos a cada alteração, apresentando de forma conciso o que está sendo alterado e seu objetivo.
- VI. Regras de negócio: apresentação das regras.
- VII. Fluxos: representação dos fluxos alternativos e de exceções nos casos de uso.
- VIII. Pré e pós condições: descrição dos requisitos que precedem cada fluxo principal e seu resultado esperado.
- IX. Planejamento de teste: detalhamento do planejamento de testes e homologação do projeto.

## 5. CONCLUSÕES E CONSIDERAÇÕES FINAIS

Este capítulo tem como objetivo trazer as conclusões alcançadas a partir do presente trabalho, tanto referente ao objetivo geral quanto aos específicos, assim como também, as considerações acerca dos temas propostos para trabalhos futuros.

### 5.1 Em relação ao objetivo geral

Motivado pelos conhecimentos técnicos obtidos no CSTGI, pela atuação na área de engenharia de requisitos, surgiu o interesse em desenvolver uma solução tecnológica que pudesse facilitar e automatizar a documentação dos requisitos de software. Assim, procurou-se estabelecer objetivos com potencial para este problema.

Em relação ao objetivo geral de elaborar um *metaframework* para automatizar a documentação de requisitos de software, conclui-se que este foi alcançado; A antecipação do prazo de entrega, por necessidades administrativas e acadêmicas impediu o seu aprofundamento.

Ao decorrer do presente trabalho foi possível identificar parâmetros que serviram para elaboração de uma análise e planejamento completo para construção da ferramenta. Esta ferramenta propõe que se automatize este processo de documentação de sistemas, trazendo assim mais agilidade, redução de prazos e custos. Os que mais se destacaram foram os parâmetros de interface com usuário, pré e pós condições, estruturação de dados, fluxos e integração.

Trabalhos com o intuito de melhorar e agilizar o processo de análise e documentação de requisitos são aspectos relevantes para o desenvolvimento de software. Conforme exposto neste trabalho, Sommerville (2011) afirma que a existência de diversos projetos que deram errado, resultando em “falhas de software”, as quais são consequências do aumento de demanda no projeto. Esta demanda, gera a necessidade de os sistemas serem construídos mais rapidamente e estas premências e expectativas fazem como que as empresas não utilizem métodos de engenharia de software, e consequentemente, acabam seu desenvolvimento e implantação ficam mais caros e menos confiáveis.



O capítulo 2 apresentou por meio da revisão de literatura os aspectos pertinentes ao tema, sendo processo de desenvolvimento de software, engenharia de requisitos e documentação de requisitos. Tais temas abordam o papel da documentação dentro do processo de desenvolvimento, bem como sua importância e características. Com isto a elaboração de um modelo de documento conciso e uma ferramenta que auxilia e traz agilidade nesta etapa do processo mostra-se viável.

No capítulo 3 foram estabelecidos os procedimentos metodológicos utilizado neste trabalho, a pesquisa utilizada e o template desenvolvido. Como classificação de pesquisa tivemos a natureza aplicada, com uma abordagem qualitativa, uma vez que buscou analisar, entender e descrever os fenômenos que estão em torno do objeto de estudo. Para coleta, análise e exposição dos objetos de estudo foi realizada uma pesquisa bibliográfica e documental. Para validação do template proposto foi realizado a aplicação de um questionário utilizando a ferramenta de formulário da google.

No capítulo 4 foi apresentado um template de documentação de requisitos adaptado a partir dos de quatro modelos disponíveis na internet e utilizando como base o material estudado e apresentado no capítulo 2. Ainda no mesmo capítulo apresenta a validação do modelo de documentação e sua viabilidade, logo após é apresentado algumas diretrizes para elaboração de uma ferramenta para automatizar requisitos de software.

Verifica-se que o processo de análise de requisitos geram diversos artefatos e especificações, os quais por sua vez precisam ser transparentes de forma clara e coerente para que assim sejam compreendidos pelos leitores, sejam eles desenvolvedores, analistas, gestores de projetos ou clientes. Com isto fica evidente a importância de um modelo de documentação padrão a ser seguido, pois assim facilita a sua transcrição e seu entendimento.

Um dos focos do trabalho era de validar um modelo padrão de documentação de requisitos, para que possa ser utilizado como base da ferramenta sugerida para automatização da documentação. Concomitantemente buscou-se compreender também quais as maiores dificuldades e impeditivos encontrados na documentação de requisitos para o projeto proposto seja elaborado. Neste sentido a aplicação do questionário oportunizou tal compreensão.

## **5.2 Em relação aos objetivos e construção do *metaframework* e *template***

O primeiro objetivo específico, foi de agrupar modelos teórico-práticos de forma consistente e lógica do ponto de vista do usuário e desenvolvedor, e produzir o embasamento teórico para seu atingimento. Com base nos estudos realizados para construção do mesmo, foi possível identificar e filtrar modelos de documentação de requisitos que contemplaram a maior parte dos tópicos citados no capítulo 2.

Em relação ao segundo objetivo específico, elaborar template de documentação, também obteve-se sucesso, porque baseado no referencial teórico, juntamente com os modelos teórico-práticos inseridos pelo metaframework, foi possível elaborar o template que contemplou os tópicos sugeridos.

No que tange o terceiro objetivo, validar o template de documentação de requisitos, foi alcançado a partir da análise do questionário aplicado aos envolvidos, cujos dados podem ser observados na seção 4.2. Neste capítulo e seção, foi possível identificar o grau de entendimento do modelo de template proposto e também entender as maiores dificuldades encontradas pelos usuários em relação a elaboração e fácil compreensão de um documento de requisitos.

Por último, aperfeiçoar o template com base nas sugestões oferecidas, foi possível após o feedback do questionário aplicado e discussão com o orientador do trabalho. Neste caso foi incluído uma tabela de versões e descritivos, permitindo melhor entendimento da evolução do software.

## **5.3 Em relação aos trabalhos futuros**

Vislumbra-se como trabalhos futuros a especificação completa e a elaboração do planejamento para desenvolvimento da ferramenta de autorização da documentação de requisitos, como forma de tornar-se tangível o resultado desta pesquisa. Tem-se também como sugestão para trabalhos futuros a realização de estudos sobre automatização dos demais tópicos de análise e do processo de desenvolvimento de software, permitindo integrar todos eles, obtendo assim um único produto.

### 5.3 Aspectos finais

A aplicação a dos conhecimentos transmitidos pelas disciplinas do curso de Gestão da Tecnologia da Informação, produziram bons resultados quando levados à prática neste trabalho. Foi possível confrontar o desenho e a implementação do template levando em conta os diversos aspectos ressaltados pelos professores.

Conceitos levantados na bibliografia foram verificados e se procurou estabelecer sua inter-relação bem como sua aplicação na engenharia de requisitos. Foram definidas as dimensões que propiciaram a construção do template, observando-se todo o processo;

O template inicialmente proposto foi submetido aos técnicos na área e observou-se que nenhuma teoria “sozinha” foi plenamente satisfatória. Ocorreu a união de BDD que vem a facilitar as atividades de teste do sistemas, bem como diagramas, desenhos de tela e descrição do banco de dados, são fatores que os técnicos consideraram essenciais e devem constar da documentação.

A partir do template proposto, pretende-se efetuar a especificação completa do sistema. elaborado o planejamento para sua execução e implementado o sistema que fará a automatização dos processos de documentação, envolvendo sua criação e atualização.

Resta a certeza de que o trabalho irá contribuir para um maior sucesso no desenvolvimento de software e que o embasamento teórico-prático apresentado pelo Curso Superior em Gestão da Tecnologia da Informação (IFSC campus Florianópolis) contribuiu decisivamente para o sucesso deste Trabalho de Conclusão de Curso.

## REFERÊNCIAS

\_\_\_\_\_. ICMC. **Especificação de requisitos de software**. Disponível em: < <https://edisciplinas.usp.br/mod/resource/view.php?id=1104110> >. Acesso em: 16 de setembro de 2019.

\_\_\_\_\_. UFU-FACOM. **Documento de requisitos**. < Disponível em: <http://www.facom.ufu.br/~bacala/MDS/Documento%20de%20Requisitos.doc> >. Acesso em: 16 de setembro de 2019.

ALMEIDA, Mário de Souza. **Elaboração de projeto, TCC, dissertação e tese: uma abordagem simples, prática e objetiva**. 2 ed. São Paulo: Atlas, 2014.

ASSOCIAÇÃO BRASILEIRA DAS EMPRESAS DE SOFTWARE. **Mercado Brasileiro de Software: panorama e tendências, 2016**. Disponível em: < <http://central.abessoftware.com.br/Content/UploadedFiles/Arquivos/Dados%202011/ABES-Publicacao-Mercado-2016.pdf> >. Acesso em: 12 de maio de 2019.

ASSOCIAÇÃO BRASILEIRA DAS EMPRESAS DE SOFTWARE. **Mercado Brasileiro de Software: panorama e tendências, 2017**. Disponível em: < <http://www.assespropr.org.br/wp-content/uploads/2018/03/ABES-Publicacao-Mercado-2017.pdf> >. Acesso em: 12 de maio de 2019.

ASSOCIAÇÃO BRASILEIRA DAS EMPRESAS DE SOFTWARE. **Mercado Brasileiro de Software: panorama e tendências, 2018**. Disponível em: < [http://central.abessoftware.com.br/Content/UploadedFiles/Arquivos/Dados%202011/a\\_f\\_abes\\_publicacao-mercado\\_2018\\_small.pdf](http://central.abessoftware.com.br/Content/UploadedFiles/Arquivos/Dados%202011/a_f_abes_publicacao-mercado_2018_small.pdf) >. Acesso em: 12 de maio de 2019.

BARBOSA, Glívia; WERNECK, Marcelo; ASSIS, Helen; FERNANDES, Ulisses; SILVA, Ismael. **Um processo de elicitação de requisitos com foco na seleção da técnica de elicitação**. Disponível em < <http://www.lbd.dcc.ufmg.br/colecoes/sbqs/2009/014.pdf> >. Acesso em: 19 de maio de 2019.

BELGAMO, Anderson. **Estudo Comparativo sobre as Técnicas de Elicitação de Requisitos do Software**. Disponível em: < [https://s3.amazonaws.com/academia.edu.documents/43384152/Estudo\\_Comparativo](https://s3.amazonaws.com/academia.edu.documents/43384152/Estudo_Comparativo) >.

\_sobre\_as\_Tcnicas\_de\_E20160305-17298-1q2z41d.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1558323347&Signature=zki0tFEevook3Sml7M1pq1P%2Fg%2B8%3D&response-content-disposition=inline%3B%20filename%3DEstudo\_Comparativo\_sobre\_as\_tecnicas\_de.pdf >. Acesso em: 19 de maio de 2019.

BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistemas com UML**. Rio de Janeiro: Elsevier, 2007.

CERVO, Amado Luiz; BERVIAN, Pedro Alcino. **Metodologia científica**. 5 ed. São Paulo: Prentice Hall, 2002.

CRUZ, Arthur Rodrigues. **Implementação de um Framework para Desenvolvimento de Sistemas de Recuperação da Informação Distribuídos**. Disponível em: <  
<http://clyde.dr.ufu.br/bitstream/123456789/23743/1/ImplementacaoFrameworkDesenvolvimento.pdf> >. Acesso em: 28 de maio de 2019.

DENCKER, Ada de Freitas Maneti; VIÁ, Sarah Chucid da. **Metodologia científica: pesquisa empírica em ciências humanas**. 2 ed. São Paulo: Saraiva, 2012.

FACHIN, Odília. **Fundamentos de Metodologia**. 5ed. São Paulo: Saraiva, 2006.

FIGUEIRA, Anderson Marques da Silva. **Análise das técnicas de levantamento de requisitos para desenvolvimento de software nas empresas de Vitória da Conquista - BA**. Disponível em: <  
<http://www2.uesb.br/computacao/wp-content/uploads/2014/09/AN%C3%81LISE-DAS-T%C3%89CNICAS-DE-LEVANTAMENTO-DE-REQUISITOS-PARA-DESENVOLVIMENTO-DE-SOFTWARE-NAS-EMPRESAS-DE-VIT%C3%93RIA-DA-CONQUISTA-%E2%80%93-BA.pdf> >. Acesso em: 19 de maio de 2019.

GABRIELI, Leandro Vettorazzi; NETO, Eduardo Kazmierczak; KAPPEL, Alexandre Moreira; LUDWIG, Icaro Paulo. **Uso de um framework no desenvolvimento de sistemas de informação web**. Disponível em: <  
[http://www.abepro.org.br/biblioteca/enegep2007\\_tr640475\\_0336.pdf](http://www.abepro.org.br/biblioteca/enegep2007_tr640475_0336.pdf) >. Acesso em: 26 de maio de 2019.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 5 ed. São Paulo: Atlas, 2010.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA - IFSC - Campus de Florianópolis - SC. **Projeto Pedagógico do Curso Superior de Tecnologia em Gestão da Tecnologia da Informação.** Disponível em:

<

<http://florianopolis.ifsc.edu.br/images/stories/ppc/graduacao/ppc%20cst%20gestao%20em%20tecnologia%20da%20informacao%20%202015.pdf> >. Acesso em: 29 de maio de 2019.

ISO/IEC/IEEE. **ISO/IEC/IEEE 29148:2011(E): Systems and software engineering - Life cycle processes - Requirements engineering.** Disponível em: < [https://edisciplinas.usp.br/pluginfile.php/1077344/mod\\_folder/content/0/iso-iec-ieee-29148-2011.pdf?forcedownload=1](https://edisciplinas.usp.br/pluginfile.php/1077344/mod_folder/content/0/iso-iec-ieee-29148-2011.pdf?forcedownload=1) >. Acesso em: 20 de maio de 2019.

KOCHE, José Carlos. **Fundamentos de metodologia científica: teoria da ciência e iniciação à pesquisa.** 33 ed. Petrópolis, RJ: Vozes, 2013.

LOPES, Leonardo Teixeira; AUDY, Jorge Luiz Nicolas. **Um modelo de processo de engenharia de requisitos para ambientes de desenvolvimento distribuído de software.** Disponível em: < <http://tede2.pucrs.br/tede2/handle/tede/5300> >. Acesso em: 19 de maio de 2019.

LOPES, Leonardo T; MAJDENBAUM, Azriel; AUDY, Jorge Luiz N. **Uma proposta para processo de requisitos em ambientes de desenvolvimento distribuído de software.** Disponível em: < <https://pdfs.semanticscholar.org/718f/a647614818baf3bd67baff85e1854e8c0f4c.pdf> >. Acesso em: 19 de maio de 2019.

MELCHER, Christiane. **Proposta metodológica para avaliações otimizadas de usabilidade em websites desenvolvido com método ágil: um estudo de caso.** Disponível em: < <https://www.maxwell.vrac.puc-rio.br/colecao.php?strSecao=resultado&nrSeq=21839@1> >. Acesso 02 de setembro de 2019.

MIGUEL, Paulo Augusto Cauchick; FLEURY, Afonso; MELLO, Carlos Henrique Pereira; NAKANO, Davi Noboru; LIMA, Edson Pinheiro; TURRIONI, João Batista; HO, Linda Lee; MORABITO, Reinaldo; MARTINS, Roberto Antonio; SOUSA, Rui;

COSTA, Sérgio E Gouvêa da; PUREZA, Vitória. **Metodologia de pesquisa em engenharia de produção e gestão de operações**. 2 ed. Rio de Janeiro: Elsevier: ABEPRO, 2012.

NASCIMENTO. Everton Ricardo do. **Metaframework de coprodução em ambientes complexos para a geração de insumos estratégicos**. Disponível em: < <http://btd.egc.ufsc.br/wp-content/uploads/2018/09/Everton-Ricardo-do-Nascimento.pdf> >. Acesso em: 17 de novembro de 2019.

OLIVEIRA, Maria Marly de. **Como fazer uma pesquisa qualitativa**. 4 ed. Petrópolis, RJ: Vozes, 2012.

PÁDUA, Clarindo Isaías P. S. **Especificação dos requisitos de software**. Disponível em: < <https://homepages.dcc.ufmg.br/~clarindo/arquivos/disciplinas/eu/material/gabaritos/er-sw.dot> >. Acesso em: 02 de setembro de 2019.

PEREIRA, André Luiz Kuczner; RODRIGUES, Andrei Ricardo; AMARAL, Eliane Cristina; SABINO, Eliney; MUNIZ, Mário Sérgio de Almeida; ABE, Narumi. **Frameworks para desenvolvimento de jogos: uma abordagem vantajosa no desenvolvimento de jogos eletrônicos**. Disponível em: < [http://portal.unisepe.com.br/unifia/wp-content/uploads/sites/10001/2018/06/058\\_frameworks.pdf](http://portal.unisepe.com.br/unifia/wp-content/uploads/sites/10001/2018/06/058_frameworks.pdf) >. Acesso em: 26 de maio de 2019.

PETERS, James F.; PEDRYCZ, Witold. **Engenharia de software: teoria e prática**. Rio de Janeiro: Elsevier, 2001.

PRESSMAN, Roger S. **Engenharia de software**. 6 ed. São Paulo: McGraw-Hill, 2006.

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

RUBIRA, Cecília M F. **Modelo de documento de especificação de requisitos**. Disponível em: < [https://moodle.unesp.br/ava/pluginfile.php/25063/mod\\_resource/content/1/ModeloinstrucoesDocReq.pdf](https://moodle.unesp.br/ava/pluginfile.php/25063/mod_resource/content/1/ModeloinstrucoesDocReq.pdf) >. Acesso em: 02 de setembro de 2019.

SANTOS, Antonio Henrique dos; CARVALHO, Nelson Ribeiro. **Frameworks e seus**

**benefícios no desenvolvimento de software.** Disponível em: < [http://revistapensar.com.br/tecnologia/pasta\\_upload/artigos/a95.pdf](http://revistapensar.com.br/tecnologia/pasta_upload/artigos/a95.pdf) >. Acesso em: 26 de maio de 2019.

SOARES, Michel dos Santos. **Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software.** Disponível em: < <http://www.dcc.ufla.br/infocomp/index.php/INFOCOMP/article/view/68> >. Acesso em: 02 de maio de 2019.

SOARES, Michel dos Santos. **Metodologias ágeis extreme programming e scrum para o desenvolvimento de software.** Disponível em: < <http://periodicosibepes.org.br/index.php/reinfo/article/view/146> >. Acesso em: 02 de setembro de 2019.

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

SOMMERVILLE, Ian. **Engenharia de Software.** 8 ed. São Paulo: Pearson Addison-Wesley, 2007.

STANKIEWICZ, Alessandro. **Modelo de integração ágil: uma adaptação do modelo cascata à organização de pequenas e médias empresas.** Disponível em: < [http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/8336/1/PG\\_COADS\\_2017\\_2\\_02.pdf](http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/8336/1/PG_COADS_2017_2_02.pdf) >. Acesso em: 02 de setembro de 2019.

VIANNA, Cleverson Tabajara. **Classificação das Pesquisas Científicas - Notas para os alunos.** Florianópolis, 2013, 2p. Disponível em: < <https://pt.slideshare.net/cleversontabajara1/metodologia-cientifica-tipos-de-pesquisa-ultimate> >. Acesso em: 21 de maio de 2019.



**APÊNDICE A - [DOCUMENTO DE REQUISITOS](#)**

**dp20**  
**Descrição de Caso de Uso**

**<PROJETO>**  
**<Cliente>**  
**<Versão: 1.0>**

## Sumário

<b>1. Introdução</b>	<b>3</b>
<b>1.1 Definições e abreviações</b>	<b>3</b>
<b>1.2 Atores:</b>	<b>3</b>
<b>2. Caso de Uso:</b>	<b>3</b>
<b>2.1 [UC01] Caso de Uso: &lt;Nome do caso de uso com verbo no infinitivo&gt;</b>	<b>3</b>
2.1.1 [UC01-PRT01] Título	3
a. [UC01-PRT01-C1] Título	4
b. [UC01-PRT01-C2] Título	4
2.1.1.1 [UC01-ED01]	4
2.1.2 [UC01-PRT02] Título	4
a. [UC01-PRT02-C1] Título	4
2.1.2.1 [UC01-ED02]	4
<b>3. Requisitos Não Funcionais</b>	<b>4</b>
a. [RN01] Título	5
<b>4. Plano de Testes</b>	<b>5</b>

## 1. Introdução

<Deve definir o público-alvo do documento e descrever seu histórico de versões, incluindo uma justificativa lógica para a criação de nova versão e um resumo das mudanças feitas em cada versão>

<Deve descrever a necessidade do sistema. Deve descrever basicamente as funções e explicar como o sistema irá funcionar como outros sistemas. Deve descrever como sistema atende aos objetivos gerais de negócios e estratégicos da organização que encomendou o software>

### 1.1 Versões

Versões	Descrição

### 1.2 Definições e abreviações

Termo	Descrição

<Deve definir os termos técnicos usados no documento. Você não deve fazer suposições sobre a experiência ou habilidade do leitor.>

### 1.3 Atores:

- a. [ATR01] <ator>  
<Breve descrição...>
- b. [ATR02] <ator>  
<Breve descrição...>

## 2. Caso de Uso:

### 2.1 [UC01] Caso de Uso: <Nome do caso de uso com verbo no infinitivo>

<Descrição do que se propõe com o caso de uso. Deve-se comentar de forma bem resumida qual é o objetivo deste documento>

#### 2.1.1 [UC01-PRT01] Título

<Inserir imagem quando houver>

##### Descritivo

<Descritivo da imagem. No descritivo deverá conter informações sobre todos os campos apresentados, juntamente com máscaras, validações e etc.

Observação: pode-se preencher o descritivo quando não houver imagem. Neste caso, entende-se que deverá existir uma tela contemplando as informações descritas.>

#### a. [UC01-PRT01-C1] Título

**Cenário:** *Os cenários dos protótipos deverão conter os fluxos da tela: gravar, editar, excluir e regras.*

**Dado:**

**Quando:**

**Então:**

#### b. [UC01-PRT01-C2] Título

**Cenário:** *Os cenários dos protótipos deverão conter os fluxos da tela: gravar, editar, excluir e regras.*

**Dado:**

**Quando:**

**Então:**

#### 2.1.1.1 [UC01-ED01]

#### 2.1.2 [UC01-PRT02] Título

<Inserir imagem quando houver>

##### Descritivo

<Descritivo da imagem. No descritivo deverá conter informações sobre todos os campos apresentados, juntamente com máscaras, validações e etc.

Observação: pode-se preencher o descritivo quando não houver imagem. Neste caso, entende-se que deverá existir uma tela contemplando as informações descritas.>

#### a. [UC01-PRT02-C1] Título

**Cenário:** *Os cenários dos protótipos deverão conter os fluxos da tela: gravar, editar, excluir e regras.*

**Dado:**

**Quando:**

Então:

#### 2.1.2.1 [UC01-ED02]

### 3. Requisitos Não Funcionais

*<Relatar todos os requisitos não funcionais como infraestrutura e condições para execução>*

**a. [RN01] Título**

*<Descritivo dos requisitos não funcionais>*

**b. [RN02] Título**

*<Descritivo dos requisitos não funcionais>*

### 4. Plano de Testes

*<Relatar todos os planos de testes e resultados esperados>*

**a. [PT01] Título**

*<Descritivo do plano de testes>*

**b. [PT02] Título**

*<Descritivo do plano de testes>*

## **APÊNDICE B - QUESTIONÁRIO DE VALIDAÇÃO**

Este questionário tem como objetivo, entender e validar um template de DOCUMENTAÇÃO DE REQUISITOS. Os resultados deste questionário serão utilizados no TCC do acadêmico Fernando M F Nishi.

Questão 1 - Área de atuação

- ☐ Desenvolvimento de sistemas
- ☐ Análise de Sistemas
- ☐ Estudante na área de tecnologia da informação
- ☐ Outro: \_\_\_\_\_

Questão 2 - Tempo de atuação na área de Tecnologia da Informação

- ☐ Menos de 1 ano
- ☐ Mais de 1 e menos de 2 anos
- ☐ Mais de 2 e menos de 3 anos
- ☐ Mais de 3 e menos de 4 anos
- ☐ Mais de 4 anos

Questão 3 - Já trabalhou, utilizou e/ou teve contato com documento de requisitos

- ☐ Sim
- ☐ Não
- ☐ N/A (Não se aplica)

### **Quanto a Compreensão do exemplo**

Esta seção tem como objetivo identificar o grau de compreensão do modelo proposto

1. Introdução	3
2. Atores:	3
3. Caso de Uso:	3
3.1 [UC01] Caso de Uso: Acesso ao sistema	3
3.1.1 [UC01-PRT01] Cadastro de usuário	3
a. [UC01-PRT01-C1] Realizar cadastro de usuário	4
3.1.1.1 [UC01-ED01] Cadastro de usuário	4
3.1.2 [UC01-PRT02] Realizar login	5
a. [UC01-PRT02-C1] Acessar sistema	5
3.1.2.1 [UC01-ED01] Cadastro de usuário	5
4. Requisitos Não Funcionais	6
a. [RN01] Estrutura do sistema	6
5. Plano de Testes	6

1. Introdução

O presente projeto é um exemplo de documentação de requisitos. Este exemplo tem como objetivo possibilitar o acesso de um usuário em um sistema, neste caso "Sistema Exemplo".

2. Atores:

- a. [ATR01] gestor  
Permissão de acesso total ao sistema.
- b. [ATR02] usuário  
Pessoas cadastradas no sistema por um gestor

3. Caso de Uso:

3.1 [UC01] Caso de Uso: Acesso ao sistema  
Este caso de uso aborda o cadastramento de um usuário e a realização de login no sistema.

3.1.1 [UC01-PR01] Cadastro de usuário

www.sistemaexemplo.com.br

Cadastro de Usuário

Nome Completo

Insira seu nome completo

CPF

123.456.789 - 00

Login

exemplo@sistemaexp.com.br

Senha

\*\*\*\*\*

Confirmação de Senha

\*\*\*\*\*

✓

Senha confirmada com sucesso

Gravar

Cancelar

**Descritivo**

**Nome completo:** type text (obrigatório, placeholder "Insira nome completo");

**CPF:** type number (obrigatório, validador de CPF válido, máscara "três dígitos '.' três dígitos '-' dois dígitos", placeholder "123.456.789-00");

**Login:** type text (obrigatório, validador de email, placeholder "exemplo@sistemaexp.com.br");

**Senha:** type password (obrigatório);

**Confirmação de senha:** type password (obrigatório, validador de confirmação de senha);

**Confirmação de senha:** notificação

Ação: quando senha e confirmação de senha forem iguais, então mostrar notificação "Senha confirmada com sucesso";

Ação: quando senha e confirmação de senha forem diferentes, então mostrar notificação "A confirmação de senha é diferente da senha";

**Gravar:** type button

Ação: quando pressionar gravar, então gravar informações inseridas;

**Cancelar:** type button

Ação: quando pressionar cancelar, então limpar informações inseridas;

a. [UC01-PR01-C1] Realizar cadastro de usuário

Cenário: gestor [ATR01] realiza cadastro de um usuário [ATR02] no sistema

Dado que gestor deseja cadastrar um usuário

Quando preencher todas as informações <lista 1>

Então gravar

Porque todos os dados são válidos

Cenário: gestor [ATR01] cancela cadastro de um usuário [ATR02] no sistema

Dado que um gestor esteja cadastrando um usuário

Quando cancelar

Então limpar informações

Lista 1

Nome completo	João das Neves
CPF	519.301.410-09
Login	exemplo@teste.com.br
Senha	123456
Confirmação de senha	123456

3.1.1.1 [UC01-ED01] Cadastro de usuário

endereço: [host:porta/contexto]/cadastroUsuario/  
método: POST

body: { }

Input: ex:{

"name": string ex:"João das Neves",

"cpf": string ex:"12312312333",



Sistema Exemplo

Identificação documental: dp01  
Versão 1.0 20/08/2019

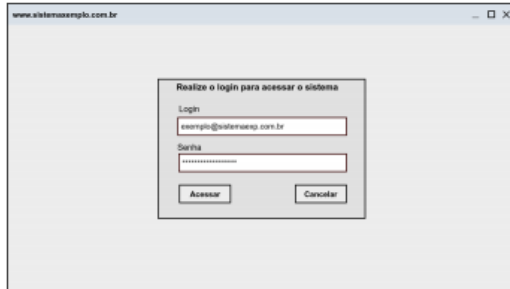
```

"login": string ex: 'exemplo@sistema.exp.com.br',
"senha": string ex: '123#exemplo',
}

code: 200 - successful operation
485 - invalid input

```

### 3.1.2 [UC01-PR02] Realizar login



#### Descritivo

**Login:** type text;  
**Senha:** type password;  
**Acessar:** type button  
 Ação: quando pressionar acessar, então permitir acesso ao sistema;  
**Cancelar:** type button  
 Ação: quando pressionar cancelar, então limpar informações inseridas;

#### a. [UC01-PR02-C1] Acessar sistema

**Cenário:** o usuário [ATR01] realizará login para acessar o sistema

**Dado** que usuário deseja acessar o sistema  
**Quando** usuário preencher o login <exemplo@teste.com.br>  
**E** senha <123456>  
**Então** acessar  
**Porque** login cadastrado como usuário no sistema  
**E** senha válida para o login

### 3.1.2.1 [UC01-ED01] Cadastro de usuário

Realiza login de usuário

Página 5 de 7

Sistema Exemplo

Identificação documental: dp01  
Versão 1.0 20/08/2019

```

endereço: [host:porta/contexto]/logUsuario/
método: POST

body: { }

Input: ex:{
  "login": string ex: 'exemplo@sistema.exp.com.br',
  "senha": string ex: '123#exemplo',
}

code: 200 - successful operation
481 - not authorized

```

## 4. Requisitos Não Funcionais

### a. [RN01] Estrutura do sistema

O sistema deverá ser construído em PHP  
 O design do deverá ser construído com Bootstrap  
 O sistema deverá ser responsivo  
 O Banco de dado deverá ser em SQL

### b. [RN02] Disponibilidade

O sistema deverá estar disponível 24 horas por dia  
 O sistema deverá estar disponível 7 dias por semana

## 5. Plano de Testes

### a. [PT01] Teste de acesso ao sistema [UC01]

**Cenário:** teste de acesso com usuário devidamente cadastrado

**Dado** usuário possua login e senha cadastrado no sistema <lista 1>  
**Quando** login <exemplo@teste.com.br>  
**E** senha <123456>  
**Então** permitir acesso

**Cenário:** teste de usuário devidamente cadastrado, com senha errada

**Dado** que usuário possua login e senha cadastrado no sistema <lista 1>  
**Quando** login <exemplo@teste.com.br>  
**E** senha <0000000>  
**Então** negar acesso  
**E** notificar "Senha incorreta"  
**Porque** senha não corresponde ao login

Página 6 de 7

Sistema Exemplo

Identificação documental: dp01  
Versão 1.0 20/08/2019

**Cenário:** teste de usuário não cadastrado

**Dado** que usuário não cadastro no sistema  
**Quando** login <testando@teste.com.br>  
**E** senha <0000000>  
**Então** negar acesso  
**E** notificar "Usuário não cadastrado no sistema"  
**Porque** login não corresponde aos logins registrado como usuário no sistema

Questão 4 - Escolha a resposta mais adequada na escala de 1 a 5 na direita.

Com base no exemplo especificado, é possível realizar o desenvolvimento do FRONTEND?	( ) 1 Discordo Totalmente ( ) 2 Discordo em Parte ( ) 3 Nem concordo nem discordo ( ) 4 Concordo parcialmente ( ) 5 Concordo Totalmente
Com base no exemplo especificado, é	( ) 1 Discordo Totalmente ( ) 2 Discordo em Parte

possível realizar a estruturação do BANCO DE DADOS?	<input type="radio"/> 3 Nem concordo nem discordo <input type="radio"/> 4 Concordo parcialmente <input type="radio"/> 5 Concordo Totalmente
Com base no exemplo especificado, é possível realizar o desenvolvimento do projeto como um todo, realizando a conexão entre frontend, backend e banco de dados?	<input type="radio"/> 1 Discordo Totalmente <input type="radio"/> 2 Discordo em Parte <input type="radio"/> 3 Nem concordo nem discordo <input type="radio"/> 4 Concordo parcialmente <input type="radio"/> 5 Concordo Totalmente
Você se considera um usuário experiente em documentação de sistemas?	<input type="radio"/> 1 Discordo Totalmente <input type="radio"/> 2 Discordo em Parte <input type="radio"/> 3 Nem concordo nem discordo <input type="radio"/> 4 Concordo parcialmente <input type="radio"/> 5 Concordo Totalmente

Questão 5 - Quando se trata de documentação de sistemas e especificação de requisitos, o que você considera como sendo as maiores dificuldades?

R:

### **Sobre a estrutura do documento apresentado**

Esta seção tem como objetivo identificar aspectos sobre a estruturação do template de documentação de requisitos utilizado.

Questão 6 - Considere quais dois pontos fortes que o documento apresenta.

R:

Questão 7 - Considere dois pontos fracos que podem ser melhorados no documento.

R:

## APÊNDICE C - RESPOSTAS DO QUESTIONÁRIO DE VALIDAÇÃO

A seguir estão as respostas das perguntas realizadas no questionário. Cada respondente será identificado pela letra R adicionado de um número, desta forma o primeiro respondente será identificado com R1 e assim sucessivamente.

### Questão 1 - Área de atuação

Área de atuação	
R1	Desenvolvimento de sistemas
R2	Desenvolvimento de sistemas
R3	Desenvolvimento de sistemas
R4	Análise de Sistemas
R5	Desenvolvimento de sistemas
R6	Análise de Sistemas
R7	Desenvolvimento de sistemas
R8	Desenvolvimento de sistemas
R9	Desenvolvimento de sistemas

### Questão 2 - Tempo de atuação na área de Tecnologia da Informação

Tempo de atuação na área de Tecnologia da Informação	
R1	Mais de 3 e menos de 4 anos
R2	Mais de 2 e menos de 3 anos
R3	Mais de 3 e menos de 4 anos
R4	Mais de 4 anos
R5	Mais de 3 e menos de 4 anos
R6	Mais de 4 anos
R7	Mais de 4 anos
R8	Mais de 3 e menos de 4 anos
R9	Mais de 1 e menos de 2 anos

### Questão 3 - Já trabalhou, utilizou e/ou teve contato com documento de requisitos

Já trabalhou, utilizou e/ou teve contato com documento de requisitos
--

R1	N/A (Não se aplica)
R2	Sim
R3	Sim
R4	Não
R5	Não
R6	Sim
R7	Sim
R8	Sim
R9	Sim

Questão 4 - Escolha a resposta mais adequada na escala de 1 a 5 na direita.

Com base no exemplo especificado, é possível realizar o desenvolvimento do FRONTEND?	R1	2 Discordo em Parte
	R2	5 Concordo Totalmente
	R3	4 Concordo parcialmente
	R4	5 Concordo Totalmente
	R5	5 Concordo Totalmente
	R6	4 Concordo parcialmente
	R7	4 Concordo parcialmente
	R8	5 Concordo Totalmente
	R9	5 Concordo Totalmente
Com base no exemplo especificado, é possível realizar a estruturação do BANCO DE DADOS?	R1	Sem Resposta
	R2	5 Concordo Totalmente
	R3	3 Nem concordo nem discordo
	R4	5 Concordo Totalmente
	R5	5 Concordo Totalmente
	R6	2 Discordo em Parte
	R7	2 Discordo em Parte
	R8	4 Concordo parcialmente
	R9	3 Nem concordo nem discordo
	R1	3 Nem concordo nem discordo

Com base no exemplo especificado, é possível realizar o desenvolvimento do projeto como um todo, realizando a conexão entre frontend, backend e banco de dados?	R2	5 Concordo Totalmente
	R3	4 Concordo parcialmente
	R4	5 Concordo Totalmente
	R5	5 Concordo Totalmente
	R6	4 Concordo parcialmente
	R7	2 Discordo em Parte
	R8	4 Concordo parcialmente
	R9	4 Concordo parcialmente
Você se considera um usuário experiente em documentação de sistemas?	R1	1 Discordo Totalmente
	R2	4 Concordo parcialmente
	R3	3 Nem concordo nem discordo
	R4	4 Concordo parcialmente
	R5	3 Nem concordo nem discordo
	R6	5 Concordo Totalmente
	R7	4 Concordo parcialmente
	R8	5 Concordo Totalmente
	R9	3 Nem concordo nem discordo

Questão 5 - Quando se trata de documentação de sistemas e especificação de requisitos, o que você considera como sendo as maiores dificuldades?

Quando se trata de documentação de sistemas e especificação de requisitos, o que você considera como sendo as maiores dificuldades?	
R1	
R2	Entender de maneira detalhada a interface de comunicação entre os componentes do sistema(backend, banco de dados, frontend)
R3	Estruturação de banco de dados e prever todas as possíveis alterações futuras do sistema.

R4	Representar a necessidade real do cliente e ainda ordenar por ordem de prioridade os requisitos.
R5	Entender o cliente
R6	Falta de detalhamento dos fluxos
R7	Manter atualizada a documentação
R8	Detalhes e Exceções
R9	Casos de uso, diagrama de classe e modelagem do DB

Questão 6 - Considere quais dois pontos fortes que o documento apresenta

Considere quais dois pontos fortes que o documento apresenta	
R1	
R2	Protótipos de tela e especificação de endpoints de uma api
R3	Endpoints e BDD
R4	Visão da tela e descrição da API REST
R5	Codificação das chamadas REST, as ações tomadas de decisão também estão bem descritas
R6	Falta de diagramas
R7	
R8	Bem detalhado, clareza na descrição dos pontos.
R9	Casos de usos com endpoints e o retorno das status HTTP. O plano de testes também é um ponto forte

Questão 7 - Considere dois pontos fracos que podem ser melhorados no documento.

Considere dois pontos fracos que podem ser melhorados no documento.	
R1	

R2	Falta de relação de entidades de banco de dados, casos de uso necessitam de um pouco mais de explicação sobre o problema que o sistema resolve.
R3	Design das telas e relacionamento do banco de dados
R4	Design de tela, regras de validação de campos
R5	As ações do sistema podem ser melhor vistas se forem feitas em formato de diagrama de caso de uso. Especificar os testes, em testes de usabilidade, unitários e integração
R6	Falta de colaboração/comunicação até o fechamento das especificações
R7	
R8	mais detalhes no testes.
R9	Talvez a falta de um diagrama de classes e a modelagem do DB

## **ANEXO 1**



# **Documento de Requisitos** **<Nome do Sistema>**

# Ficha Técnica

---

**Equipe Responsável pela Elaboração**

<nome> <divisão-região>  
<nome> <divisão-região>  
<nome> <divisão-região>  
<nome> <divisão-região>

---

**Público Alvo**

Este manual destina-se a <especifique o público alvo deste documento>

---

Versão <x.y> - <local>, <mês> de <ano>

---

Dúvidas, críticas e sugestões devem ser encaminhadas por escrito para o seguinte endereço postal:

<especifique o endereço para correspondência>

Ou para o seguinte endereço eletrônico:

<especifique o e-mail para contato>

Recomendamos que o assunto seja identificado com o título desta obra. Alertamos ainda para a importância de se identificar o endereço e o nome completos do remetente para que seja possível o envio de respostas.

---

Windows e Microsof Word são marcas registradas da Microsoft Corporation

---

---

# Sumário

<b>INTRODUÇÃO.....</b>	<b>P2</b>
<b>Visão geral deste documento.....</b>	<b>P2</b>
<b>Convenções, termos e abreviações.....</b>	<b>P2</b>
1. Identificação dos Requisitos.....	P2
2. Prioridades dos Requisitos.....	P2
<b>Referências.....</b>	<b>P2</b>
<b>CAPÍTULO 1 - DESCRIÇÃO GERAL DO SISTEMA.....</b>	<b>C1 . P2</b>
<b>Abrangência e sistemas relacionados.....</b>	<b>C1 . P2</b>
<b>Descrição dos usuários.....</b>	<b>C1 . P2</b>
1. <Opcional> <Nome de um tipo específico de usuário> .....	C1 . P2
2. <Opcional> <Nome de outro tipo específico de usuário > .....	C1 . P2
3.....	C1 . P2
<b>CAPÍTULO 2 - REQUISITOS FUNCIONAIS (CASOS DE USO).....</b>	<b>C2 . P2</b>
<b>&lt;Nome de subseção para agrupar casos de uso correlacionados&gt;.....</b>	<b>C2 . P2</b>
[RF001] <Nome do caso de uso> .....	C2 . P2
Fluxo de eventos principal.....	C2 . P2
<Opcional> Fluxos secundários (alternativos e de exceção).....	C2 . P2
[RF...] <Nome de outro caso de uso> .....	C2 . P2
<b>&lt;Nome de outra subseção para agrupar outros casos de uso correlacionados&gt;.....</b>	<b>C2 . P2</b>
.....	C2 . P2
<b>CAPÍTULO 3 - REQUISITOS NÃO FUNCIONAIS.....</b>	<b>C3 . P2</b>
<b>Usabilidade.....</b>	<b>C3 . P2</b>
[NF001] <Nome do requisito> .....	C3 . P2
[NF...] <Nome do requisito> .....	C3 . P2
<b>Confiabilidade.....</b>	<b>C3 . P2</b>
[NF...] <Nome do requisito> .....	C3 . P2
<b>Desempenho.....</b>	<b>C3 . P2</b>
[NF...] <Nome do requisito> .....	C3 . P2

<b>Segurança.....</b>	<b>C3 . P2</b>
[NF...] <Nome do requisito>.....	C3 . P2
<b>Distribuição.....</b>	<b>C3 . P2</b>
[NF...] <Nome do requisito>.....	C3 . P2
<b>Padrões.....</b>	<b>C3 . P2</b>
[NF...] <Nome do requisito>.....	C3 . P2
<b>Hardware e software.....</b>	<b>C3 . P2</b>
[NF...] <Nome do requisito>.....	C3 . P2

<b>CAPÍTULO 4 - &lt;OPCIONAL&gt; DESCRIÇÃO DA INTERFACE COM O USUÁRIO.....</b>	<b>C4 . P2</b>
<Identificador de uma interface>.....	C4 . P2
1.<Opcional> Críticas da interface.....	C4 . P2
<Identificador de outra interface>.....	C4 . P2
.....	C4 . P2

# Introdução

<Este espaço deve ser usado para descrever os objetivos deste documento e o público ao qual ele se destina. Complete e/ou adapte o texto abaixo para fornecer essas informações.>

Este documento especifica o sistema <Nome do sistema>, fornecendo aos desenvolvedores as informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação do sistema.

## Visão geral deste documento

<Esta seção fornece uma breve descrição de como o resto deste documento está organizado. Complete e/ou adapte o texto abaixo para fornecer essa informação.>

Esta introdução fornece as informações necessárias para fazer um bom uso deste documento, explicitando seus objetivos e as convenções que foram adotadas no texto, além de conter uma lista de referências para outros documentos relacionados. As demais seções apresentam a especificação do sistema <Nome do sistema> e estão organizadas como descrito abaixo.

- **Seção 2** – Descrição geral do sistema: apresenta uma visão geral do sistema, caracterizando qual é o seu escopo e descrevendo seus usuários.
- **Seção 3** – Requisitos funcionais (casos de uso): especifica todos os requisitos funcionais do sistema, descrevendo os fluxos de eventos, prioridades, atores, entradas e saídas de cada caso de uso a ser implementado.
- **Seção 4** – Requisitos não funcionais: especifica todos os requisitos não funcionais do sistema, divididos em requisitos de usabilidade, confiabilidade, desempenho, segurança, distribuição, adequação a padrões e requisitos de hardware e software.
- **Seção 5** – Descrição da interface com o usuário: apresenta desenhos, figuras ou rascunhos de telas do sistema.

## Convenções, termos e abreviações

<Esta subseção deve descrever as convenções, termos e abreviações necessários para interpretar apropriadamente este documento. As explicações necessárias podem ser fornecidas diretamente nesta seção ou através de referências para outros documentos ou para apêndices deste documento. Complete e/ou adapte o texto abaixo para fornecer essas informações.>

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

### .1 Identificação dos Requisitos

Por convenção, a referência a requisitos é feita através do nome da subseção onde eles estão descritos, seguido do identificador do requisito, de acordo com o esquema abaixo:

[nome da subseção.identificador do requisito]

Por exemplo, o requisito [Recuperação de dados.RF016] está descrito em uma subseção chamada “Recuperação de dados”, em um bloco identificado pelo número [RF016]. Já o requisito não funcional [Confiabilidade.NF008] está descrito na seção de requisitos não funcionais de Confiabilidade, em um bloco identificado por [NF008].

## .2 Prioridades dos Requisitos

Para estabelecer a prioridade dos requisitos foram adotadas as denominações “essencial”, “importante” e “desejável”.

- **Essencial** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
- **Importante** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.
- **Desejável** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis são requisitos que podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

## Referências

<Esta seção deve prover uma lista de todos os documentos relacionados a este documento. Complete e/ou adapte o texto abaixo para fornecer essas informações. Não esqueça de inserir uma referência para o glossário.>

Documentos relacionados ao <Nome do sistema> e/ou mencionados nas seções a seguir:

1. Título; Número (se aplicável); Data; Instituição, divisão ou equipe responsável pelo documento; Link para o documento (se aplicável);
2. Título; Número (se aplicável); Data; Instituição, divisão ou equipe responsável pelo documento; Link para o documento (se aplicável);
3. ...

# Descrição geral do sistema

<Descreva aqui, em linhas gerais, os objetivos do sistema, comunicando o propósito da aplicação e a importância do projeto para todas as pessoas envolvidas.

Se for necessário apresentar detalhes mais técnicos sobre o sistema, você também pode usar esta seção para descrever em linhas gerais a arquitetura do sistema, indicando seus módulos principais, o uso (se existir) da Internet ou outra rede de comunicação, componentes *on-line* e *off-line*, e a interação (se existir) com outros sistemas. Use um diagrama se achar conveniente.>

## Abrangência e sistemas relacionados

<Nesta seção, descreva em linhas gerais o que o sistema irá fazer (suas principais funcionalidades) e o que ele **não** irá fazer (escopo negativo), deixando claro se o sistema irá interagir com outros sistemas relacionados ou se ele é independente e totalmente auto-contido.

As funcionalidades principais do sistema devem ser apenas citadas, para dar uma idéia geral ao leitor dos serviços que serão fornecidos pelo sistema. Os detalhes serão fornecidos posteriormente, na seção 3 deste documento. Funcionalidades que a princípio seriam da alçada do sistema e que não serão implementadas também devem ser listadas, registrando-se o motivo pela qual elas não serão contempladas (porque serão fornecidas por outros sistemas relacionados, por exemplo, ou porque serão implementadas apenas em projetos futuros).

Se o sistema for independente e totalmente auto-contido diga isso explicitamente, caso contrário, liste e descreva brevemente os outros sistemas com os quais este sistema deve interagir, explicando, de maneira geral, quais os papéis de cada um e o meio de comunicação entre eles.>

## Descrição dos usuários

<Para efetivamente prover produtos e serviços que atendam às necessidades dos usuários, é necessário entender os desafios que eles enfrentam para executar suas funções. Esta seção deve descrever os futuros usuários do sistema e os principais problemas que limitam sua produtividade.

O grau de detalhamento necessário nas descrições depende do sistema que você está especificando. Em projetos onde a equipe já acumulou um bom conhecimento dos usuários, uma breve descrição de suas características gerais é suficiente. Em projetos onde não se tem muita informação sobre os usuários do sistema, deve-se descrever as características específicas de cada usuário nas subseções a seguir.

Descreva os aspectos gerais, relacionados a todos os usuários, aqui. Depois, se for necessário, descreva nas subseções abaixo as características específicas de cada usuário.>

### .1 <Opcional> <Nome de um tipo específico de usuário>

<Se for conveniente fornecer mais detalhes sobre um tipo específico de usuário, use esta subseção para descrevê-lo.>

### .2 <Opcional> <Nome de outro tipo específico de usuário >

<Prossiga no detalhamento das características dos usuários, descrevendo todos os tipos de usuário que for necessário, cada um em uma subseção.>

### **.3 ...**



## Requisitos funcionais (casos de uso)

<Nesta seção, apresente todos os requisitos funcionais, ou casos de uso, do sistema. Em sistemas grandes é comum haver muitos casos de uso e, para facilitar a visualização deste documento, você pode agrupá-los em subseções de casos de uso correlacionados. Os nomes das subseções devem ser únicos e pequenos (3 palavras no máximo) e podem ser formados por palavras, números e/ou abreviações.

Cada um dos casos de uso deve ser descrito em um bloco específico, seguindo o modelo descrito abaixo. O identificador do bloco deve conter o número do caso de uso (por exemplo, [RF001]) e o seu nome. Se os casos de uso forem agrupados em subseções específicas, a numeração deles deve ser reiniciada a cada subseção (dentro de uma mesma subseção, todo caso de uso deve ter um número de identificação único).

Quando a primeira versão deste documento for disponibilizada para a equipe de desenvolvimento, os nomes das subseções e os números dos casos de uso não devem ser modificados ou reaproveitados, para não invalidar referências externas feitas a eles.>

### <Nome de subseção para agrupar casos de uso correlacionados>

<Utilize este espaço para descrever características comuns dos casos de uso desta seção, explicitando o motivo do seu agrupamento em uma seção única.

Se todos os casos de uso desta seção estiverem relacionados com o mesmo ator você pode informar isso aqui, especificando qual é o ator em questão, e eliminar o campo “Ator:” das descrições dos casos de uso feitas nos blocos a seguir.>

### [RF001] <Nome do caso de uso>

<Opcional – forneça uma pequena explicação do propósito do caso de uso (útil quando o nome do caso de uso não deixa suficientemente claro qual é o seu objetivo) e o(s) seu(s) respectivo(s) ator(es). Em seguida, substitua um dos símbolos abaixo por ☒, para indicar a prioridade do caso de uso.>

**Ator:** <informe o(s) ator(es) do caso de uso >

**Prioridade:**      ☐ Essencial                      ☐ Importante                      ☐ Desejável

<**Opcional**> **Interface(s) associada(s):** <inclua aqui o(s) identificador(es) da(s) respectiva(s) interface(s) do caso de uso (descrita(s) na Seção 5).>

**Entradas e pré condições:** <Liste aqui todas as entradas e/ou pré condições do caso de uso. Pré condição de um caso de uso é o estado em que o sistema deve estar para realizar o caso de uso.>

**Saídas e pós condições:** <Liste aqui todas as saídas e/ou pós condições do caso de uso. Pós condição de um caso de uso é a lista de possíveis estados em que o sistema pode estar imediatamente após o término da realização do caso de uso.>

### Fluxo de eventos principal

<Descreva aqui o fluxo de eventos principal que ocorre durante a execução do caso de uso.>

**<Opcional> Fluxos secundários (alternativos e de exceção)****<Fluxo secundário XXX>**

<Use este espaço para descrever o fluxo secundário XXX do caso de uso.>

**<Fluxo secundário YYY>**

<Prossiga na descrição dos fluxos secundários do caso de uso, descrevendo cada um deles separadamente.>

**[RF...] <Nome de outro caso de uso>**

<Utilize os mesmos campos mostrados no bloco anterior para descrever este e os demais requisitos funcionais (casos de uso) desta subseção.>

**<Nome de outra subseção para agrupar outros casos de uso correlacionados>**

<Prossiga de maneira similar à subseção anterior para descrever quaisquer outras subseções que forem usadas para agrupar requisitos funcionais.>

...

# Requisitos não funcionais

<Esta seção deve conter os requisitos não funcionais do sistema. Para uma melhor organização deste documento, utilize as subseções abaixo para agrupar os requisitos não funcionais relacionados. Naturalmente, o número e tipo de subseções utilizadas depende do sistema que está sendo especificado e não é preciso utilizar todas elas. Simplesmente elimine as subseções para as quais não for encontrado nenhum requisito.>

Os requisitos não funcionais devem ser identificados com um identificador único, da mesma maneira que os requisitos funcionais (casos de uso). Inicie a numeração com o identificador NF001 e prossiga incrementando os números a medida que forem surgindo novos requisitos não funcionais. Reinicie a numeração em cada subseção. Forneça também um nome para o requisito, como foi feito para os requisitos funcionais.

Descreva o requisito, assinale a sua prioridade e, em seguida, caso o requisito esteja relacionado a um caso de uso ou a um grupo de casos de uso específicos, utilize o campo “**Caso(s) de uso associado(s):**” para identificar o(s) caso(s) de uso correspondente(s). Se for um requisito não funcional do sistema como um todo, esse campo não precisa ser utilizado.>

## Usabilidade

Esta seção descreve os requisitos não funcionais associados à facilidade de uso da interface com o usuário, material de treinamento e documentação do sistema.

### [NF001] <Nome do requisito>

<Descreva o requisito não funcional e substitua um dos símbolos abaixo por ☒, para indicar a sua prioridade.>

**Prioridade:**      ☐ Essencial                      ☐ Importante                      ☐ Desejável

<**Opcional**> **Caso(s) de uso associado(s):** <use este campo para identificar a que caso(s) de uso o requisito de usabilidade está relacionado.>

### [NF...] <Nome do requisito>

<Utilize os mesmos campos mostrados no bloco anterior para descrever este e os demais requisitos não funcionais de usabilidade.>

## Confiabilidade

Esta seção descreve os requisitos não funcionais associados à frequência, severidade de falhas do sistema e habilidade de recuperação das mesmas, bem como à corretude do sistema.

### [NF...] <Nome do requisito>

<Utilize os mesmos campos mostrados na seção 4.1 para descrever este e os demais requisitos não funcionais de confiabilidade.>

## Desempenho

Esta seção descreve os requisitos não funcionais associados à eficiência, uso de recursos e tempo de resposta do sistema.

### [NF...] <Nome do requisito>

<Utilize os mesmos campos mostrados na seção 4.1 para descrever este e os demais requisitos não funcionais de desempenho.>

## Segurança

Esta seção descreve os requisitos não funcionais associados à integridade, privacidade e autenticidade dos dados do sistema.

### [NF...] <Nome do requisito>

<Utilize os mesmos campos mostrados na seção 4.1 para descrever este e os demais requisitos não funcionais de segurança.>

## Distribuição

Esta seção descreve os requisitos não funcionais associados à distribuição da versão executável do sistema.

### [NF...] <Nome do requisito>

<Utilize os mesmos campos mostrados na seção 4.1 para descrever este e os demais requisitos não funcionais de distribuição.>

## Padrões

Esta seção descreve os requisitos não funcionais associados a padrões ou normas que devem ser seguidos pelo sistema ou pelo seu processo de desenvolvimento.

<Se você mencionar documentos relacionados, não esqueça de listá-los na seção 1.3.>

### [NF...] <Nome do requisito>

<Utilize os mesmos campos mostrados na seção 4.1 para descrever este e os demais requisitos não funcionais de adequação a padrões.>

## Hardware e software

Esta seção descreve os requisitos não funcionais associados ao hardware e software usados para desenvolver ou para executar o sistema.

### [NF...] <Nome do requisito>

<Utilize os mesmos campos mostrados na seção 4.1 para descrever este e os demais requisitos não funcionais de hardware e software.>

## <Opcional> Descrição da interface com o usuário

<Esta seção deve conter desenhos ou rascunhos das telas do sistema que forem necessários ou convenientes para esclarecer algum dos requisitos do sistema. Para sistemas que possuem protótipos ou versões já desenvolvidas é possível capturar as telas e apresentar figuras das mesmas.

Use nomes e/ou números para identificar cada interface e descreva-as em seções independentes.>

### <Identificador de uma interface>

<Descreva a interface em questão, através de figuras, diagramas e/ou texto.

#### .1 <Opcional> Críticas da interface

<Você pode fazer aqui a descrição de críticas simples de interface, como o tamanho e máscara de campos, simplificando assim a descrição dos fluxos de exceção.>

### <Identificador de outra interface>

<Prossiga no detalhamento das interfaces do sistema, descrevendo todas que for necessário, cada uma em uma subseção.>

...

## **ANEXO 2**

**< nome da organização fornecedora >**

# **Especificação dos Requisitos do Software**

**< nome do produto >**

**{ Versão revisada < n > }**

**Autores: < nomes dos autores >**

**< local >**

**< data >**

Projeto < nome do projeto >

## ***Aprovação***

Aprovamos a Especificação dos Requisitos { Versão revisada < n > } do projeto < nome do projeto >.

<Nome>	<Data>	_____
<Nome>	<Data>	_____
<Nome>	<Data>	_____
<Nome>	<Data>	_____
<Nome>	<Data>	_____
<Nome>	<Data>	_____



### ***Versões revisadas anteriores***

<b>Versão revisada</b>	<b>Comentário</b>	<b>Data</b>

# Especificação dos Requisitos do Software

## Sumário

Aprovação.....	2
Versões revisadas anteriores.....	3
Especificação dos Requisitos do Software.....	4
1 Introdução.....	5
1.1 Objetivos deste documento.....	5
1.2 Escopo do produto.....	5
1.2.1 Nome do produto e de seus componentes principais.....	5
1.2.2 Missão do produto.....	5
1.2.3 Limites do produto.....	5
1.2.4 Benefícios do produto.....	6
1.3 Materiais de referência.....	6
1.4 Definições e siglas.....	6
1.5 Visão geral deste documento.....	7
2 Descrição geral do produto.....	7
2.1 Perspectiva do produto.....	7
2.1.1 Diagrama de contexto.....	7
2.1.2 Interfaces de usuário.....	8
2.1.3 Interfaces de hardware.....	8
2.1.4 Interfaces de software.....	8
2.1.5 Interfaces de comunicação.....	8
2.1.6 Restrições de memória.....	8
2.1.7 Modos de operação.....	9
2.1.8 Requisitos de adaptação ao ambiente.....	9
2.2 Funções do produto.....	9
2.3 Usuários e sistemas externos.....	9
2.3.1 Descrição.....	9
2.3.2 Características dos usuários.....	10
2.4 Restrições.....	10
2.5 Hipóteses de trabalho.....	10
2.6 Requisitos adiados.....	10
3 Requisitos específicos.....	11
3.1 Requisitos de interface externa.....	11
3.1.1 Interfaces de usuário.....	11
3.1.1.1 Interface de usuário << nome da interface >>.....	11
3.1.2 Interfaces de hardware.....	12
3.1.2.1 Interface de hardware << nome da interface >>.....	12
3.1.3 Interfaces de software.....	13
3.1.3.1 Interface de software << nome da interface >>.....	13
3.1.4 Interfaces de comunicação.....	13
3.1.4.1 Interface de comunicação << nome da interface >>.....	13
3.2 Requisitos funcionais.....	14
3.2.1 Diagramas de casos de uso.....	14
3.2.1.1 Diagrama de casos de uso << nome do diagrama de casos de uso >>.....	14
3.2.2 Casos de uso.....	15
3.2.2.1 Caso de uso << nome do caso de uso >>.....	15
3.3 Requisitos não funcionais.....	16
3.3.1 Requisitos de desempenho.....	16
3.3.1.1 Requisito de desempenho << nome do requisito >>.....	16
3.3.2 Requisitos de dados persistentes.....	16
3.3.2.1 Diagrama de classes persistentes.....	16
3.3.2.2 Classes persistentes.....	17
3.3.2.3 Propriedades das classes persistentes.....	17

3.3.3	Restrições ao desenho.....	17
3.3.3.1	Restrição ao desenho << nome da restrição >> .....	17
3.3.4	Atributos da qualidade.....	17
3.3.4.1	Atributo da qualidade << nome do atributo >> .....	17
3.3.5	Outros requisitos.....	18
3.3.5.1	Requisito << nome do requisito >> .....	18
4	Informação de suporte.....	19

## **1    *Introdução***

### **1.1    Objetivos deste documento**

--

### **1.2    Escopo do produto**

#### **1.2.1    *Nome do produto e de seus componentes principais***

--

#### **1.2.2    *Missão do produto***

--

#### **1.2.3    *Limites do produto***

--

#### **1.2.4    *Benefícios do produto***

Número de ordem	Benefício	Valor para o cliente
1		
2		
3		
4		
5		
6		
7		

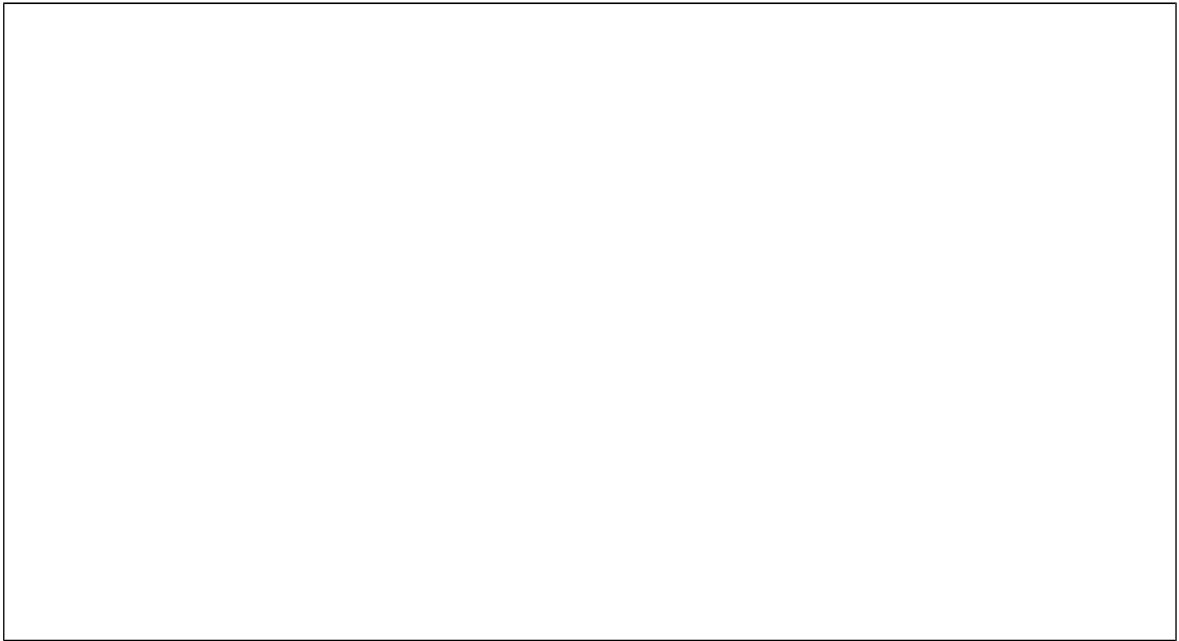
### 1.3 Materiais de referência

Número de ordem	Tipo do material	Referência bibliográfica
1		
2		
3		
4		
5		
6		

### 1.4 Definições e siglas

Número de ordem	Sigla	Definição
1		
2		
3		
4		
5		
6		
7		
8		
9		

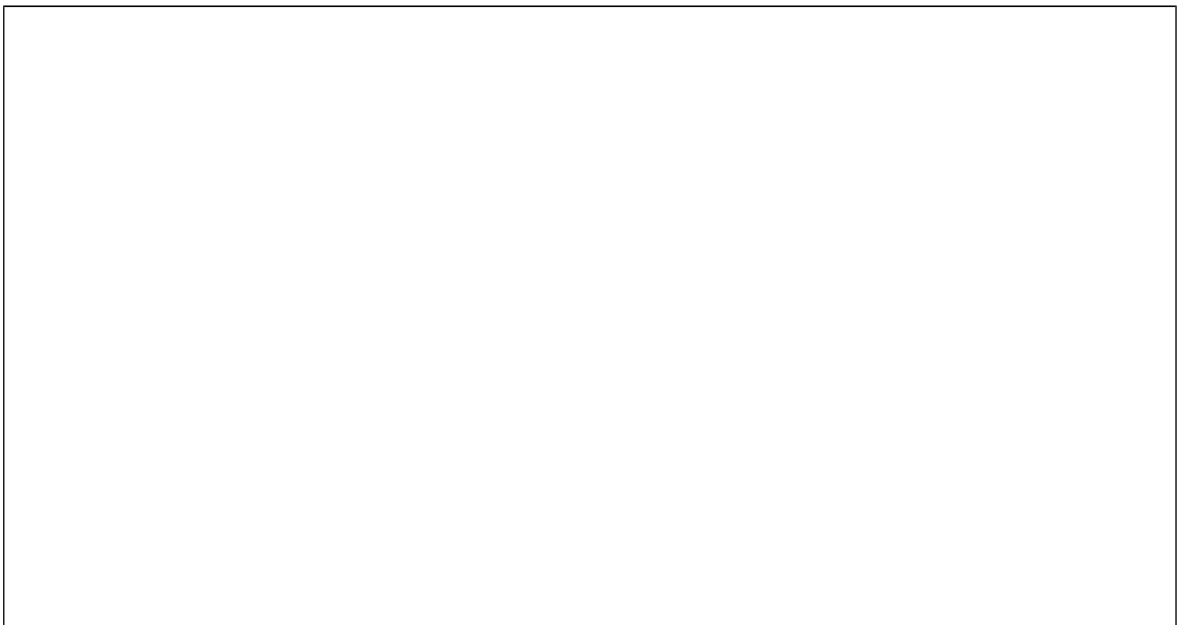
## **1.5 Visão geral deste documento**



## **2 Descrição geral do produto**

### **2.1 Perspectiva do produto**

#### **2.1.1 Diagrama de contexto**



**2.1.2 Interfaces de usuário**

<b>Número de ordem</b>	<b>Nome</b>	<b>Ator</b>	<b>Caso de uso</b>	<b>Descrição</b>
1				
2				
3				
4				
5				
6				

**2.1.3 Interfaces de hardware**

<b>Número de ordem</b>	<b>Nome</b>	<b>Ator</b>	<b>Caso de uso</b>	<b>Descrição</b>
1				
2				

**2.1.4 Interfaces de software**

<b>Número de ordem</b>	<b>Nome</b>	<b>Ator</b>	<b>Caso de uso</b>	<b>Descrição</b>
1				
2				
3				

**2.1.5 Interfaces de comunicação**

<b>Número de ordem</b>	<b>Nome</b>	<b>Ator</b>	<b>Caso de uso</b>	<b>Descrição</b>
1				
2				
3				
4				

**2.1.6 Restrições de memória**

<b>Número de ordem</b>	<b>Tipo de memória</b>	<b>Limites aplicáveis</b>
1		
2		

### 2.1.7 *Modos de operação*

Número de ordem	Tipo de operação	Descrição da operação	Detalhes de operação
1			
2			
3			

### 2.1.8 *Requisitos de adaptação ao ambiente*

Número de ordem	Requisito	Detalhes
1		
2		
3		

## 2.2 *Funções do produto*

Número de ordem	Caso de uso	Descrição
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

## 2.3 *Usuários e sistemas externos*

### 2.3.1 *Descrição*

Número de ordem	Ator	Definição
1		
2		
3		
4		



**2.3.2 Características dos usuários**

Número de ordem	Ator	Frequência de uso	Nível de instrução	Proficiência na aplicação	Proficiência em informática
1					
2					
3					
4					

**2.4 Restrições**

Número de ordem	Restrição	Descrição
1		
2		
3		
4		
5		

**2.5 Hipóteses de trabalho**

Número de ordem	Hipótese	De quem depende
1		
2		
3		
4		
5		

**2.6 Requisitos adiados**

Número de ordem	Referência ao requisito	Detalhes
1		
2		
3		
4		
5		

### 3 Requisitos específicos

### 3.1 Requisitos de interface externa

### 3.1.1 Interfaces de usuário

### 3.1.1.1 Interface de usuário << nome da interface >>

#### 3.1.1.1.1 *Leiaute sugerido*

--

#### 3.1.1.1.2 Diagrama de estados

--

#### 3.1.1.1.3 *Relacionamentos com outras interfaces*

--

#### 3.1.1.1.4 Campos

Número	Nome	Descrição	Valores válidos	Formato	Tipo	Restrições
1						
2						
3						
4						
5						
6						
7						
8						
9						

**3.1.1.1.5 Comandos**

Número	Nome	Ação	Restrições
1			
2			
3			
4			
5			
6			
7			
8			

**3.1.1.1.6 Observações**

--

**3.1.2 Interfaces de hardware****3.1.2.1 Interface de hardware << nome da interface >>****3.1.2.1.1 Fonte da entrada**

--

**3.1.2.1.2 Destino da saída**

--

**3.1.2.1.3 Relacionamentos com outras interfaces**

--

**3.1.2.1.4 Formato**

--

**3.1.2.1.5            Observações**

--

**3.1.3            Interfaces de software**

**3.1.3.1            Interface de software << nome da interface >>**

**3.1.3.1.1            Fonte da entrada**

--

**3.1.3.1.2            Destino da saída**

--

**3.1.3.1.3            Relacionamentos com outras interfaces**

--

**3.1.3.1.4            Formato**

--

**3.1.3.1.5            Observações**

--

**3.1.4            Interfaces de comunicação**

**3.1.4.1            Interface de comunicação << nome da interface >>**

**3.1.4.1.1            Fonte da entrada**

--

**3.1.4.1.2            Destino da saída**

--

**3.1.4.1.3                    *Relacionamentos com outras interfaces***

--

**3.1.4.1.4                    *Formato***

--

**3.1.4.1.5                    *Observações***

--

**3.2        Requisitos funcionais**

**3.2.1        *Diagramas de casos de uso***

**3.2.1.1                    Diagrama de casos de uso << nome do diagrama de casos de uso >>**

--

### **3.2.2      *Casos de uso***

#### **3.2.2.1          Caso de uso << nome do caso de uso >>**

##### **3.2.2.1.1              *Precondições***

--

##### **3.2.2.1.2              *Fluxo principal***

--

##### **3.2.2.1.3              *Subfluxos***

###### **3.2.2.1.3.1              Subfluxo << nome do subfluxo >>**

--

##### **3.2.2.1.4              *Fluxos alternativos***

###### **3.2.2.1.4.1              Fluxo alternativo << nome do fluxo alternativo >>**

<b>Precondições</b>	
<b>Passos</b>	

**3.2.2.1.5**                    ***Diagrama de estado / Diagrama de atividade***



**3.2.2.1.6**                    ***Observações***



**3.3**                    **Requisitos não funcionais**

**3.3.1**                    ***Requisitos de desempenho***

**3.3.1.1**                    **Requisito de desempenho << nome do requisito >>**



**3.3.2**                    ***Requisitos de dados persistentes***

**3.3.2.1**                    **Diagrama de classes persistentes**



### 3.3.2.2 Classes persistentes

Número de ordem	Nome	Descrição
1		
2		
3		

### 3.3.2.3 Propriedades das classes persistentes

Número de ordem	Nome	Frequência de consulta (por dia)	Frequência de atualização (por dia)	Cardinalidade máxima	Restrições aplicáveis
1					
2					
3					
4					
5					
6					
7					

### 3.3.3 Restrições ao desenho

#### 3.3.3.1 Restrição ao desenho << nome da restrição >>

--

### 3.3.4 Atributos da qualidade

#### 3.3.4.1 Atributo da qualidade << nome do atributo >>

--



### **3.3.5**      *Outros requisitos*

#### **3.3.5.1**      Requisito << nome do requisito >>

--

Projeto < nome do projeto >

## **4    *Informação de suporte***

## **ANEXO 3**

## Modelo de documento de especificação de requisitos

O “template” de documento de especificação de requisitos apresentado a seguir, foi criado utilizando como referências os modelos utilizados por [1][2][3]. O capítulo 5 do livro “Engenharia de Software” [4] mostra um exemplo de um documento de requisitos preenchido que pode auxiliá-lo, no entanto, o “template” usado é um pouco diferente do proposto nesse documento. O modelo que você deve adotar será está descrito a seguir.

### **1. Introdução**

#### **1.1. Objetivo**

Delinear o objetivo da especificação de requisitos e especificar os leitores deste documento.

#### **1.2. Escopo**

- Identificar pelo nome o produto de software a ser produzido (e.g. Gerenciador Eletrônico de Documentos – GED).
- Explicar o que o produto vai e, se necessário, não vai fazer.
- Descrever a aplicação de software que está sendo especificada, incluindo seus benefícios, objetivos e metas.
- Ser consistente com outras especificações de alto nível do sistema, se elas existirem.

#### **1.3. Definições, acrônimos e abreviações**

Esta sub-seção deve conter todas as definições de termos, acrônimos e abreviações necessárias para corretamente entender esta especificação. Esta informação pode ser apresentada em apêndice ou em referências à outros documentos.

#### **1.4. Referências**

Esta sub-seção deve:

- Conter uma lista completa de **todos** os documentos referenciados na especificação
- Identificar cada documento adequadamente com título, autores, data, editor etc.
- Especificar as fontes de onde as referências foram obtidas

#### **1.5. Visão geral**

Esta sub-seção deve

- Descrever resumidamente o conteúdo do restante da especificação
- Explicar como a especificação está organizada

## **2. Descrição Geral**

### **2.1. Requisitos funcionais**

Descrever as funcionalidades do software – produzir uma lista de todos os requisitos funcionais e classificá-los como obrigatórios, desejáveis ou opcionais.

### **2.2. Requisitos de interface**

Definir como o software interage com as pessoas, com o hardware do sistema, com outros sistemas e com outros produtos. Detalhar os aspectos das interfaces do produto (normalmente é feito um esboço das interfaces, levantado através de um protótipo ou de estudos em papel; são também detalhadas as interfaces com outros sistemas e componentes de sistemas). É obrigatório o desenho das telas referentes às principais funcionalidades do produto.

### **2.3. Atributos de qualidade**

Descrever os requisitos de desempenho (velocidade de processamento, tempo de resposta, etc.) e outros aspectos considerados necessários para que o produto atinja a qualidade desejada (por exemplo portabilidade, manutenibilidade, confiabilidade, etc.). Finalmente, classificar e rever os requisitos, estabelecendo prioridades (obrigatório, desejável ou opcional).

### **2.4. Características dos usuários**

Descrever as características gerais dos usuários do produto, incluindo o nível educacional, a experiência e os conhecimentos técnicos.

### **2.5. Restrições**

Enumerar as restrições impostas pela aplicação, tais como padrões, linguagem de implementação, ambientes operacionais e limites de recursos.

### **2.5. Suposições e dependências**

Listar todos os fatores que afetam os requisitos da especificação. Esses fatores não são restrições ao projeto do sistema, mas sim mudanças que podem afetar os requisitos. Por exemplo, uma suposição pode ser que a aplicação será instalada em um sistema operacional específico. Se, este sistema operacional não for disponível, isso poderia afetar os requisitos.

## **3. Anexo**

Citar todos os recursos e técnicas utilizados para a extração de requisitos, assim como as questões feitas, o nome das pessoas, empresas, telefones e datas de contato.

## **Referências**

[1] G.Kotonya e I.Sommerville. *Requirements engineering – processes and techniques*. Editora John Wiley and Sons – 2002.

[2] IEEE 830-1998 *Recommended Practice for Software Requirements Specification*  
[<http://ieeexplore.ieee.org/iel4/5841/15571/00720574.pdf?tp=&isnumber=15571&arnumber=720574>]

[3] A. Carvalho. *Documento de especificação de requisitos*

[<http://www.ic.unicamp.br/~ariadne/inf301/modulo2-v.pdf>]

[4] W.P.P. Filho. *Engenharia de software – fundamentos, métodos e padrões*. 2ªed. Editora LTC - 2003

## **ANEXO 4**

**<Nome do Projeto>  
Especificação de Requisitos de Software  
Para <Nome do Sistema>**

**Versão <1.0>**

*[Observação: O template a seguir é fornecido para uso com o Rational Unified Process (RUP). O texto em azul exibido entre colchetes e em itálico (style=InfoBlue) foi incluído para orientar o autor e deve ser excluído antes da publicação do documento. Qualquer parágrafo inserido após esse estilo será definido automaticamente como normal (estilo=BodyText).]*



<Nome do Projeto>	Versão: <1.0>
Especificação de Requisitos de Software	Data: <dd/mmm/yy>
<document identifier>	

## Histórico da Revisão

Data	Versão	Descrição	Autor
<dd/mmm/yy>	<x.x>	<details>	<name>

<Nome do Projeto>	Versão: <1.0>
Especificação de Requisitos de Software	Data: <dd/mmm/yy>
<document identifier>	

## Tabela de Conteúdo

1.	Introdução	4
1.1	Finalidade	4
1.2	Escopo	4
1.3	Definições, Acrônimos, e Abreviações	4
1.4	Referências (Opcional)	4
1.5	Visão Geral (Opcional)	4
2.	Descrição Geral	4
3.	Requisitos Específicos	5
3.1	Requisitos Funcionais	5
3.1.1	<Requisito funcional um>	5
3.2	Requisitos de Performance	5
3.2.1	<Requisito de Desempenho Um>	5
3.3	Atributos do Sistema de Software (Opcional)	5
3.3.1	Restrições do Design	5
3.3.2	Interfaces	6
3.3.3	Componentes Comprados	6
3.3.4	Observações Legais, de Direitos Autorais etc	6
3.3.5	Padrões Aplicáveis	6
3.4	Outros Requisitos (Opcional)	6
3.4.1	Requisitos de Usabilidade	6
3.4.2	Requisitos de Confiabilidade	6
3.4.3	Suportabilidade	7
3.4.4	Requisitos de Sistema de Ajuda e de Documentação de Usuário On-line	7
3.4.5	Requisitos de Licenciamento	7

<Nome do Projeto>	Versão: <1.0>
Especificação de Requisitos de Software	Data: <dd/mm/yy>
<document identifier>	

# Especificação de Requisitos de Software

## 1. Introdução

*[A introdução da Especificação de Requisitos de Software (SRS) deve fornecer uma visão geral de toda a SRS. Ela deve incluir a finalidade, o escopo, as definições, os acrônimos, as abreviações, as referências e a visão geral da SRS.]*

*[Observação: A Especificação de Requisitos de Software (SRS) captura todos os requisitos de software do sistema ou de uma parte do sistema.. A seguir, há um esquema de uma SRS típica para um projeto que utiliza somente requisitos em estilo de linguagem natural tradicional - sem modelagem de casos de uso. Essa SRS captura todos os requisitos em um único documento, com seções aplicáveis inseridas a partir das Especificações Suplementares (que não serão mais necessárias). Para ter acesso a um template de uma SRS que utilize a modelagem de casos de uso, que consiste em um pacote contendo Casos de Uso do modelo de casos de uso e Especificações Suplementares aplicáveis, assim como outras informações de suporte, consulte o arquivo rup\_SRS-uc.dot.]*

*[É possível organizar uma SRS de várias maneiras diferentes. Consulte o padrão [IEEE830-1998] para obter explicações mais detalhadas, assim como outras opções de organização da SRS.]*

### 1.1 Finalidade

*[Especifique a finalidade desta SRS. A SRS deverá descrever totalmente o comportamento externo do aplicativo ou do subsistema identificado. Ela também deverá descrever requisitos não funcionais, restrições de design e outros fatores necessários para fornecer uma visão completa e abrangente dos requisitos do software.]*

### 1.2 Escopo

*[Uma breve descrição do aplicativo de software a que se aplica a SRS; o recurso ou outro agrupamento de subsistemas; a que modelo(s) de Caso de Uso a SRS está associada; e tudo o mais que seja afetado ou influenciado por este documento.]*

### 1.3 Definições, Acrônimos, e Abreviações

*[Esta subseção deve fornecer as definições de todos os termos, acrônimos e abreviações necessárias à adequada interpretação da SRS. Essas informações podem ser fornecidas mediante referência ao Glossário do projeto.]*

### 1.4 Referências (Opcional)

*[Esta subseção deve fornecer uma lista completa de todos os documentos mencionados em qualquer outra parte da SRS. Cada documento deverá ser identificado por título, número do relatório (se aplicável), data e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento.]*

### 1.5 Visão Geral (Opcional)

*[Esta subseção descreve o que o restante da SRS contém e explica como o documento está organizado.]*

## 2. Descrição Geral

*Esta seção da SRS deve descrever os fatores gerais que afetam o produto e seus requisitos. Ela não deve especificar requisitos específicos. Em vez disso, deve fornecer uma base para esses requisitos, que serão definidos detalhadamente na Seção 3, e facilitar sua compreensão. Inclua itens como:*

- perspectiva do produto
- funções do produto

<Nome do Projeto>	Versão: <1.0>
Especificação de Requisitos de Software	Data: <dd/mmm/yy>
<document identifier>	

- *características do usuário*
- *restrições*
- *suposições e dependências*
- *subconjuntos de requisitos*

### 3. Requisitos Específicos

*[Esta seção da SRS deve conter todos os requisitos de software em um nível de detalhamento suficiente para possibilitar que os designers projetem um sistema que satisfaça esses requisitos e que os testadores verifiquem se o sistema satisfaz esses requisitos. Quando for utilizada a modelagem de casos de uso, esses requisitos serão capturados nos Casos de Uso e nas especificações suplementares aplicáveis. Se a modelagem de casos de uso não for utilizada, o esquema das especificações suplementares poderá ser inserido diretamente nesta seção, conforme mostrado a seguir.]*

#### 3.1 Requisitos Funcionais

*[Esta seção descreve os requisitos funcionais do sistema que são expressos no estilo de linguagem natural. Para muitos aplicativos, este poderá ser o volume do Pacote da SRS. É necessário refletir muito para organizar esta seção. Normalmente, esta seção é organizada por recurso, mas outros métodos de organização também poderão ser adequados como, por exemplo, a organização por usuário ou a organização por subsistema. Entre os requisitos funcionais poderão estar incluídos conjuntos de recursos, capacidades e segurança.*

*[Quando as ferramentas de desenvolvimento de aplicativos, como ferramentas de requisitos, ferramentas de modelagem, entre outras, forem utilizadas para capturar a funcionalidade, esta seção fará referência à disponibilidade desses dados, indicando o local e o nome da ferramenta usada para capturar os dados.]*

##### 3.1.1 <Requisito funcional um>

*[A descrição do requisito deve ser feita aqui.]*

#### 3.2 Requisitos de Performance

*[As características de desempenho do sistema devem ser descritas nesta seção. Inclua tempos de resposta específicos. Quando aplicável, faça referência, por nome, aos Casos de Uso relacionados.*

- *tempo de resposta de uma transação (médio, máximo)*
- *taxa de transferência como, por exemplo, transações por segundo*
- *capacidade como, por exemplo, o número de clientes ou de transações que o sistema pode acomodar*
- *modos de degradação (o modo aceitável de operação quando o sistema tiver sido degradado de alguma maneira)*
- *a utilização de recursos como, por exemplo, memória, disco, comunicações, etc.]*

##### 3.2.1 <Requisito de Desempenho Um>

*[A descrição do requisito deve ser feita aqui]*

#### 3.3 Atributos do Sistema de Software (Opcional)

##### 3.3.1 Restrições do Design

*[Esta seção deve indicar todas as restrições de design referentes ao sistema que está sendo criado. As restrições de design representam decisões de design que foram impostas e devem ser respeitadas. Entre os exemplos desse tipo de restrição estão linguagens de software, requisitos de processo de software, uso*

<Nome do Projeto>	Versão: <1.0>
Especificação de Requisitos de Software	Data: <dd/mm/yy>
<document identifier>	

*prescrito de ferramentas de desenvolvimento, restrições de design e de arquitetura, componentes comprados, bibliotecas de classes, etc]*

### 3.3.1.1 <Restrição de Design Um>

*[A descrição do requisito deve ser feita aqui]*

### 3.3.2 Interfaces

*[Esta seção define as interfaces que devem ser suportadas pelo aplicativo. Ele deve conter especificidades, protocolos, porta e endereços lógicos adequados, entre outros, para que o software possa ser desenvolvido e verificado em relação aos requisitos de interface.]*

#### 3.3.2.1 Interface de Usuário

*[Descreva as interfaces de usuário que deverão ser implementadas pelo software.]*

#### 3.3.2.2 Interfaces de Hardware

*[Esta seção define todas as interfaces de hardware que devem ser suportadas pelo software, incluindo a estrutura lógica, os endereços físicos, o comportamento esperado, etc. ]*

#### 3.3.2.3 Interfaces de Software

*[Esta seção descreve as interfaces de software com outros componentes do sistema de software. Poderão ser componentes comprados, componentes reutilizados de outro aplicativo ou componentes que estejam sendo desenvolvidos para subsistemas fora do escopo desta SRS, mas com os quais esse aplicativo de software deve interagir.]*

#### 3.3.2.4 Interfaces de Comunicação

*[Descreva todas as interfaces de comunicação com outros sistemas ou dispositivos como, por exemplo, redes locais, dispositivos seriais remotos, etc.]*

### 3.3.3 Componentes Comprados

*[Esta seção descreve todos os documentos comprados para serem usados com o sistema, quaisquer restrições de utilização ou de licenciamento aplicáveis, e quaisquer padrões associados de compatibilidade e de interoperabilidade ou de interface.]*

### 3.3.4 Observações Legais, de Direitos Autorais etc

*[Esta seção descreve todos os avisos legais necessários, garantias, observações sobre direitos autorais, observações sobre patentes, logomarcas, marcas comerciais ou problemas de conformidade com logotipos referentes ao software.]*

### 3.3.5 Padrões Aplicáveis

*[Esta seção descreve, por meio de referências, todos os padrões aplicáveis e as seções específicas desses padrões que se aplicam ao sistema que está sendo descrito. Entre esses padrões estão incluídos, por exemplo, padrões reguladores, de qualidade e legais, padrões de indústria referentes à usabilidade, interoperabilidade, internacionalização, compatibilidade com o sistema operacional, etc.]*

## 3.4 Outros Requisitos (Opcional)

### 3.4.1 Requisitos de Usabilidade

*[Esta seção deve incluir todos os requisitos que afetam a usabilidade. Por exemplo,*

- *especifique o tempo de treinamento necessário para que usuários normais e usuários com conhecimentos avançados se tornem produtivos em operações específicas*
- *especifique períodos de tempo mensuráveis para tarefas típicas ou baseie os requisitos de*

<Nome do Projeto>	Versão: <1.0>
Especificação de Requisitos de Software	Data: <dd/mmm/yy>
<document identifier>	

*usabilidade do novo sistema em outros sistemas que os usuários conheçam e gostem*

- especifique requisitos de forma que estejam em conformidade com padrões comuns de usabilidade como, por exemplo, os padrões CUA da IBM ou os padrões GUI da Microsoft]*

#### 3.4.1.1 <Requisito de Usabilidade Um>

*[A descrição do requisito deve ser feita aqui]*

#### 3.4.2 Requisitos de Confiabilidade

*[Os requisitos de confiabilidade do sistema devem ser especificados aqui. A seguir, há algumas sugestões:*

- Disponibilidade - especifique a porcentagem de tempo disponível (xx.xx%), as horas de uso, o acesso à manutenção, as operações de modo degradado, etc.*
- Tempo Médio entre Falhas (MTBF) - normalmente especificado em horas, mas também poderá ser especificado em termos de dias, meses ou anos.*
- Tempo Médio para Reparo (MTTR) - quanto tempo o sistema poderá ficar sem funcionar após uma falha?*
- Exatidão - especifique a precisão (resolução) e a exatidão (através de algum padrão conhecido) necessárias na saída do sistema.*
- Taxa Máxima de Erros ou Defeitos - geralmente expressa em termos de erros por milhares de linhas de código (erros/KLOC) ou de erros por ponto de função (erros/ponto de função).*
- Taxa de Erros ou Defeitos - categorizados em termos de erros pouco importantes, importantes e críticos: o(s) requisito(s) devem definir o que se entende por um erro "crítico"; por exemplo, a perda total de dados ou uma total incapacidade de usar determinadas partes da funcionalidade do sistema.]*

#### 3.4.2.1 <Requisito de Confiabilidade Um>

*[A descrição do requisito deve ser feita aqui.]*

#### 3.4.3 Suportabilidade

*[Esta seção indica todos os requisitos que aprimorarão a suportabilidade ou manutenibilidade do sistema que está sendo criado, incluindo padrões de codificação, convenções de nomeação, bibliotecas de classes, acesso à manutenção e utilitários de manutenção.]*

#### 3.4.3.1 <Requisito de Suportabilidade Um>

*[A descrição do requisito deve ser feita aqui.]*

#### 3.4.4 Requisitos de Sistema de Ajuda e de Documentação de Usuário On-line

*[Descreva os requisitos, se houver, de documentação de usuário on-line, sistemas de ajuda, observações sobre ajuda, etc.]*

#### 3.4.5 Requisitos de Licenciamento

*[Esta seção define todos os requisitos de imposição de licenciamento ou outros requisitos de restrição de utilização que devem ser exibidos pelo software.]*