



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

ANTHONY TYRONY PEREIRA DA SILVA

FERRAMENTA WEB PARA RASTREABILIDADE DE REQUISITOS

MONOGRAFIA

Recife
2018

ANTHONY TYRONY PEREIRA DA SILVA

FERRAMENTA WEB PARA RASTREABILIDADE DE REQUISITOS

Monografia apresentada ao Centro de Informática da Universidade Federal de Pernambuco, como parte dos requisitos necessários à obtenção do título de Bacharel em Engenharia da Computação.

Orientador: Márcio Lopes Cornélio

Recife
2018

Agradecimentos

Agradeço primeiramente a Deus por ter me abençoado durante toda minha trajetória, fazendo sempre infinitamente mais do que tenho pedido ou pensado.

A minha família, por ter me ajudado e fortalecido no decorrer da graduação, em especial o meu tio, Antônio, o qual sempre me apoio durante a graduação.

A minha mãe, Rosineide, a qual batalhou e se desgastou para que eu e meu irmão pudéssemos ter o que ela não teve, nos proporcionando a oportunidade de estudar e ser quem somos hoje.

A minha amada Josianne por sempre acreditar em mim, mesmo quando eu não acreditava em mim mesmo, e por ter sido companheira, paciente e compreensiva durante essa jornada.

Ao meu orientador, Márcio Lopes Cornélio, pelas contribuições, correções e paciência.

Enfim, a todos que me apoiaram e intercederam por mim, muito obrigado.

Resumo

Durante o ciclo de vida de desenvolvimento de um projeto os requisitos podem mudar devido a fatores internos e externos de uma organização, que podem ser resultados, por exemplo, da mudança do negócio ou adequação a uma nova legislação vigente. Diante desse cenário, é importante manter o mínimo de controle dessas mudanças. A disciplina engenharia de requisitos provê meios para o gerenciamento dos requisitos de um produto de software, entre eles está a rastreabilidade de requisitos, importante atividade na gestão de requisitos, a qual promove maior controle das mudanças dos requisitos e uma melhor análise de seus impactos por manter associações entre requisitos e artefatos gerados no desenvolvimento de um sistema. Neste trabalho será implementada uma ferramenta gratuita e de código aberto, com objetivo de permitir o gerenciamento descentralizado de requisitos de software em apenas uma base e servir de apoio nas atividades de análise de impactos, levantamento de risco e estimativa de custo, mantendo controle das alterações de requisitos e relacionamentos entre eles.

Palavras-chave: Requisitos. Gestão de requisitos. Rastreabilidade de requisitos.

Abstract

During the life cycle of a project's development requirements may change due to internal and external factors of an organization, which may be results, for example, from the change of business or adaptation to a new legislation in force. Given this scenario, it is important to keep minimal control of these changes. The requirements engineering discipline provides a means for managing the requirements of a software product, among them is requirements traceability, an important activity in requirements management, which promotes greater control of requirements changes and a better analysis of their impacts by maintain associations between requirements and artifacts generated in the development of a system. In this work, a free and open-source tool will be implemented, aiming to allow the decentralized management of software requirements on a single data base and to support the activities of impact analysis, risk assessment and cost estimation by keeping control of requirements changes and relationships between them.

Key words: Requirements. Requirements management. Requirements traceability.

Lista de ilustrações

Figura 1 – Principais atividades da gestão de requisitos	17
Figura 2 – Acompanhamento da distribuição do status dos requisitos ao longo do ciclo de desenvolvimento de um projeto	23
Figura 3 – Diagrama de Tipos de Links de Rastreamento	25
Figura 4 – Diagrama da Arquitetura MVC	31
Figura 5 – Diagrama de classe - Parte 1	32
Figura 6 – Diagrama de classe - Parte 2	33
Figura 7 – Modelo de Dados	37
Figura 8 – Exemplo de pré-rastreabilidade	38
Figura 9 – Exemplo de pós-rastreabilidade	39
Figura 10 – Exemplo de relacionamento horizontal	40
Figura 11 – Relacionamentos entre requisitos funcionais e casos de uso	41
Figura 12 – Exemplo de conflito entre demandas	42
Figura 13 – Histórico de alterações de um artefato	42
Figura 14 – Distribuição de status dos artefatos	43
Figura 15 – Diagrama de casos de uso	59
Figura 16 – Diagrama de Sequência - Cadastrar Usuário	96
Figura 17 – Diagrama de Sequência - Criar Projeto	97
Figura 18 – Diagrama de Sequência - Criar Tipo de Artefato	98
Figura 19 – Diagrama de Sequência - Criar Tipo de Relacionamento	99
Figura 20 – Diagrama de Sequência - Criar Artefato	100
Figura 21 – Diagrama de Sequência - Criar Relacionamento	100
Figura 22 – Diagrama de Sequência - Criar Demanda	101
Figura 23 – Diagrama de Sequência - Editar Artefato	102
Figura 24 – Diagrama de Sequência - Reverter Artefato	103
Figura 25 – Diagrama de Sequência - Excluir Projeto	104
Figura 26 – Diagrama de Sequência - Excluir Tipo de Artefato	105
Figura 27 – Diagrama de Sequência - Excluir Tipo de Relacionamento	106
Figura 28 – Diagrama de Sequência - Excluir Artefato	106
Figura 29 – Diagrama de Sequência - Excluir Demanda	107

Lista de tabelas

Tabela 1 – Classes de requisitos voláteis	19
Tabela 2 – Status de requisito sugeridos	22
Tabela 3 – Lista de Requisitos Funcionais	51
Tabela 4 – Lista de Requisitos Não Funcionais	56
Tabela 5 – Permissões dos Atores	57
Tabela 6 – UC - Cadastrar Usuário	60
Tabela 7 – UC - Inativar Usuário	61
Tabela 8 – UC - Confirmar Cadastro	62
Tabela 9 – UC - Reenviar Email de Confirmação	63
Tabela 10 – UC - Resetar Senha	63
Tabela 11 – UC - Efetuar Login	66
Tabela 12 – UC - Editar Perfil	67
Tabela 13 – UC - Criar Projeto	68
Tabela 14 – UC - Editar Projeto	69
Tabela 15 – UC - Excluir Projeto	70
Tabela 16 – UC - Adicionar Usuário ao Projeto	71
Tabela 17 – UC - Criar Demanda	72
Tabela 18 – UC - Editar Demanda	73
Tabela 19 – UC - Excluir Demanda	74
Tabela 20 – UC - Criar Tipo de Artefato	75
Tabela 21 – UC - Editar Tipo de Artefato	76
Tabela 22 – UC - Excluir Tipo de Artefa	77
Tabela 23 – UC - Criar Artefato	77
Tabela 24 – UC - Editar Artefato	79
Tabela 25 – UC - Excluir Arquivo	82
Tabela 26 – UC - Excluir Artefato	83
Tabela 27 – UC - Importar Artefato	84
Tabela 28 – UC - Remover Artefato	85
Tabela 29 – UC - Reverter Artefato	86
Tabela 30 – UC - Criar Tipo de Relacionamento	87
Tabela 31 – UC - Editar Tipo de Artefato	88
Tabela 32 – UC - Excluir Tipo de Relacionamento	90
Tabela 33 – UC - Criar Relacionamento	90
Tabela 34 – UC - Editar Relacionamento	92
Tabela 35 – UC - Excluir Relacionamento	94

Lista de abreviaturas e siglas

CMMI	Capability Maturity Model Integration
CSS	Cascading Style Sheets
FRR	Ferramenta de Rastreabilidade de Requisitos
HTML5	Hyper Text Markup Language version 5
HTTP	Hyper Text Transport Protocol
ID	Identificador
IT	Tecnologia da Informação
MPS.BR	Melhoria do Processo de Software Brasileiro
MVC	Modew-View-Controller
PC	Computador Pessoal
PDF	Portable Document Format
SVN	Subversion
TI	Tecnologia da Informação
UC	Caso de Uso
UML	Linguagem de Modelagem Unificada

Sumário

1	Introdução	11
1.1	Motivação	12
1.2	Objetivos	13
1.3	Organização	13
2	Fundamentação Teórica	15
2.1	Requisitos	15
2.2	Gestão de Requisitos	16
2.2.1	Controle de Versão de Requisitos	18
2.2.2	Controle de Mudanças	18
2.2.2.1	Processo de Controle de Mudanças	19
2.2.3	Acompanhamento do Status dos Requisitos	21
2.2.4	Rastreabilidade de Requisitos	23
2.2.4.1	Tipos de Relacionamentos	25
2.2.4.2	Visualização das Informações de Rastreabilidade	26
2.3	Ferramentas Gratuitas para Gestão de Requisitos	26
2.3.1	SIGERAR	27
2.3.2	OpenReq	27
2.4	Conclusão	27
3	A Ferramenta de Rastreabilidade de Requisitos	29
3.1	Funcionalidades	29
3.2	Arquitetura	30
3.3	Diagrama de Classes	31
3.4	Modelo de Dados	33
3.4.1	Users	33
3.4.2	Projects	33
3.4.3	ProjectsUsers	34
3.4.4	ArtifactTypes	34
3.4.5	ArtifactStatuses	34
3.4.6	Artifacts	34
3.4.7	Versions	34
3.4.8	Attachments	35
3.4.9	Comments	35
3.4.10	ChartData	35
3.4.11	RelationshipTypes	35
3.4.12	Relationships	35

3.4.13	Demands	36
3.4.14	ArtifactDemands	36
3.5	Conclusão	37
4	Prova de Conceito	38
4.1	Pré-rastreabilidade	38
4.2	Pós-rastreabilidade	38
4.3	Rastreamento Bidirecional	39
4.4	Rastreamento Horizontal	39
4.5	Rastreamento Vertical	40
4.6	Múltiplas Matrizes de Rastreabilidade	40
4.7	Conflito Entre Demandas	41
4.8	Controle de Versão	42
4.9	Acompanhamento do Status da Demanda/Projeto	42
5	Comparativo entre Ferramentas	44
6	Conclusão	46
6.1	Trabalhos Futuros	46
	Referências	48
	APÊNDICES	50
	APÊNDICE A – Requisitos Funcionais	51
	APÊNDICE B – Requisitos Não Funcionais	56
	APÊNDICE C – Atores e Permissões	57
	APÊNDICE D – Diagrama de Casos de Uso	59
	APÊNDICE E – Descrição dos Casos de Uso	60
E.1	UC - Cadastrar Usuário	60
E.2	UC - Inativar Usuário	61
E.3	UC - Confirmar Cadastro	62
E.4	UC - Reenviar Email de Confirmação	63
E.5	UC - Resetar Senha	63
E.6	UC - Efetuar Login	66
E.7	UC - Editar Perfil	67
E.8	UC - Criar Projeto	68

E.9	UC - Editar Projeto	69
E.10	UC - Excluir Projeto	70
E.11	UC - Adicionar Usuário ao Projeto	71
E.12	UC - Criar Demanda	72
E.13	Editar Demanda	73
E.14	UC - Excluir Demanda	74
E.15	UC - Criar Tipo de Artefato	75
E.16	UC - Editar Tipo de Artefato	76
E.17	UC - Excluir Tipo de Artefa	77
E.18	UC - Criar Artefato	77
E.19	UC - Editar Artefato	79
E.20	UC - Excluir Arquivo	82
E.21	UC - Excluir Artefato	83
E.22	UC - Importar Artefato	84
E.23	UC - Remover Artefato	85
E.24	UC - Reverter Artefato	86
E.25	UC - Criar Tipo de Relacionamento	87
E.26	UC - Editar Tipo de Relacionamento	87
E.27	UC - Excluir Tipo de Relacionamento	90
E.28	UC - Criar Relacionamento	90
E.29	UC - Editar Relacionamento	92
E.30	Excluir Relacionamento	94
	APÊNDICE F – Diagramas de Sequência	96
F.1	DS1 - Cadastrar Usuário	96
F.2	DS2 - Criar Projeto	97
F.3	DS3 - Criar Tipe de Artefato	98
F.4	DS4 - Criar Tipo de Relacionamento	99
F.5	DS5 - Criar Artefato	100
F.6	DS6 - Criar Relacionamento	100
F.7	DS7 - Criar Demanda	101
F.8	DS8 - Editar Artefato	102
F.9	DS9 - Reverter Artefato	103
F.10	DS10 - Excluir Projeto	104
F.11	DS11 - Excluir Tipo de Artefato	105
F.12	DS12 - Excluir Tipo de Relacionamento	106
F.13	DS13 - Excluir Artefato	106
F.14	DS14 - Excluir Demanda	107

1 Introdução

Um estudo de pesquisa conduzido pelo Standish Group com 365 companhias, englobando 8380 projetos, mostrou que 31,1% dos projetos foram cancelados antes de serem finalizados e 52,7% foram finalizados com atraso, acima do orçamento e com menos funcionalidades. Entre as causas de falha dos projetos citadas estão: requisitos incompletos (13.1%); falta de envolvimento do usuário (12,4%); e mudanças nos requisitos e especificações (8,7%) (STANDISH GROUP, 1995). Um outro estudo conduzido pelo Project Management Institute (PMI) mostrou que 47% dos projetos malsucedidos falharam devido à falta ou mau gerenciamento de requisitos. Além disso, o principal fator identificado foi, assim como no estudo citado anteriormente, a incompletude dos requisitos (PMI, 2014). Esses dados revelam a importância da gestão de requisitos no sucesso dos projetos.

O gerenciamento de requisitos é o processo de coletar, analisar, refinar e priorizar os requisitos do produto e, em seguida, planejar sua entrega (AHA! LABS INC, 2018). Entre as suas principais atividades está a rastreabilidade de requisitos. Ela é o meio de alcançar a integridade e completude da documentação de requisitos e tem um importante papel no gerenciamento de mudanças de requisitos (NUSEIBEH; EASTERBROOK, 2000). Entre os benefícios da aplicação da rastreabilidade de requisitos listados por Wiegers e Beatty (2013) estão:

- Encontrar requisitos isolados, sem uma origem definida (um pedido do usuário) ou um requisito funcional associado;
- Certificar-se que um requisito foi implementado;
- Analisar impactos de mudanças de um requisito, visto que é possível identificar os artefatos de projetos relacionados;
- Reuso: As informações de rastreio facilitam a reutilização dos componentes do produto, identificando pacotes de requisitos, projetos, códigos e testes relacionados;
- Teste: Quando um teste falha, os links entre testes, requisitos e códigos direcionam os desenvolvedores para áreas prováveis que necessitam ser examinadas.

Uma ferramenta de apoio à gestão de requisitos é indispensável para que se possa alcançar os benefícios citados anteriormente. Outro ponto fundamental para se fazer uso de uma ferramenta de apoio à gestão de requisitos é o tamanho do projeto. À medida que o projeto cresce e o tempo passa, fica mais difícil manter o controle do

processo de desenvolvimento e a memória da equipe começa a perder detalhes dos requisitos.

Para sistemas pequenos, pode não ser necessário usar ferramentas especializadas de gerenciamento de requisitos. O processo de gerenciamento de requisitos pode ser realizado usando as facilidades disponíveis em processadores de texto, planilhas e bancos de dados de PC. Contudo, para sistemas maiores, é necessário o uso ferramentas mais especializadas (SOMMERVILLE, 2010).

1.1 Motivação

Na empresa de tecnologia da informação (TI) - Resource IT Solutions - é prestado serviço de desenvolvimento e manutenção de software a diversas empresas e entidades públicas. Em um dos projetos da empresa, é desenvolvido e mantido o sistema PJe (Processo Judicial Eletrônico) para automação dos processos da Justiça Federal da 5ª Região. Os artefatos de software desse projeto, tais como documento de visão, especificação de requisitos, casos de uso e protótipos de telas, são armazenados e uma ferramenta de controle de versão chamada Subversion (SVN). Já os requisitos, regras de negócio, entre outros, são mantidos em planilhas do Excel nas nuvens.

Este projeto atualmente possui milhares de artefatos e mantê-los através de planilhas não é uma tarefa fácil. Além disso, existem várias pessoas envolvidas e diversos subprojetos, os quais são chamados de demandas, sendo executados em paralelo. Este cenário tem ocasionado diversos problemas, entre eles, o cancelamento de subprojetos, retrabalho de documentação e desenvolvimento, requisitos incompletos ou não implementados.

Um problema recorrente é a interseção de demandas, ou seja, demandas sendo executadas em paralelo com artefatos em comum. Como não há uma ferramenta de apoio à gestão de requisitos, é muito difícil identificar esses conflitos, pois é impraticável analisar cada demanda em busca de artefatos comuns no SVN. Esse problema tem gerado artefatos com versões distintas simultaneamente, gerando inconsistências.

Uma outra questão de fundamental importância é a análise de impacto. Devido à magnitude desse projeto e a falta de ferramenta de apoio à gestão de requisitos, as análises de impacto têm sido feitas a partir do conhecimento do analista sobre o sistema. Essas análises têm se mostrado superficiais e falhas, impactando de forma significativa a gestão de requisitos e a estimativa de custo, levando até o cancelamento de algumas demandas.

Ferramentas de gestão de requisitos com rastreabilidade são difíceis de serem encontradas, e as ferramentas encontradas são antigas e exigiriam um esforço grande

para evoluí-las e adequá-las a novas tecnologias.

Devido estas questões, este trabalho irá implementar uma ferramenta de apoio à gestão de requisitos, mais especificamente, relacionada à rastreabilidade de requisitos. Esta ferramenta tem por objetivo auxiliar na identificação do status do projeto e acompanhar os status dos requisitos e artefatos; auxiliar na atividade de análise de impacto provendo uma matriz de rastreabilidade que exhibe as relações entre os artefatos do projeto; identificar os conflitos de requisitos e artefatos entre demandas; e prover o acesso aos requisitos e artefatos sempre atualizados.

1.2 Objetivos

Objetivo geral: Implementar uma ferramenta de apoio à gestão de requisitos voltada para rastreabilidade.

Objetivos específicos:

- Definir requisitos da ferramenta;
- Identificar os conceitos relacionados à rastreabilidade de requisitos;
- Enumerar as principais técnicas de visualização de relacionamento entre requisitos.

1.3 Organização

Este trabalho está dividido em 5 capítulos, cuja estrutura está descrita a seguir:

- **Capítulo 2 - Fundamentação Teórica:** contém uma definição de requisitos e suas principais características, discorre sobre as principais atividades da gestão de requisitos, define os principais conceitos relacionados a rastreabilidade de requisitos, técnicas de visualização de relacionamentos e apresenta algumas ferramentas para gestão de requisitos;
- **Capítulo 3 - A Ferramenta de Rastreabilidade de Requisitos:** contém os requisitos da ferramenta proposta, sua arquitetura, diagrama de classes e modelo de dados;
- **Capítulo 4 - Prova de Conceito:** lista exemplos de como a ferramenta proposta atende os conceitos relatados no Capítulo 2;
- **Capítulo 5 - Comparativo entre Ferramentas:** Contém um comparativo entre a ferramenta implementada e outras ferramentas gratuitas listadas no Capítulo 3;

- **Capítulo 6 - Conclusão:** contém a conclusão do trabalho e possíveis trabalhos futuros.

2 Fundamentação Teórica

Identificar requisitos e analisá-los é uma das etapas mais importante do processo de desenvolvimento de software. Não desenvolver essas atividades de forma adequada representa grande risco ao sucesso do projeto, visto que 70% a 85% do custo de retrabalho no desenvolvimento de um sistema estão relacionados com erros de requisitos (LEFFINGWELL, 1997).

Os requisitos especificam o sistema e deles surgem os demais artefatos de software que auxiliam no desenvolvimento, validação e teste do sistema. Os requisitos são voláteis, ou seja, podem sofrer mudanças durante todas as fases do processo de desenvolvimento. Diante desse cenário, é importante gerir essas mudanças através de um processo de gestão de requisitos bem definido.

2.1 Requisitos

Segundo o IEEE (1990), um requisito é definido como:

- 1) Uma condição ou capacidade que o usuário precisa para resolver um problema ou alcançar um objetivo;
- 2) Uma condição ou capacidade que o sistema ou um componente do sistema deve atender ou possuir para satisfazer um contrato, padrão, especificação ou outra documentação formal imposta;
- 3) Uma representação documentada de uma condição ou capacidade como em 1 e 2.

Sommerville (2010) define que requisito é a descrição do que um sistema deve fazer – o serviço que ele deve prover e as restrições de suas operações. Os requisitos refletem as necessidades dos clientes de um sistema que atende a um determinado propósito.

Segundo Sommerville (2010), os requisitos devem ser descritos em diferentes níveis de descrição, pois eles comunicam informações sobre o sistema para diferentes tipos de leitores. Um nível, requisitos do usuário, contém informação de alto nível; já outro nível, requisitos do sistema, tem uma descrição mais detalhada do que o sistema deve fazer. Esses dois níveis são definidos como:

- Os requisitos do usuário são declarações, em uma linguagem natural e diagramas, de quais serviços o sistema deve fornecer aos usuários do sistema e as restrições sob as quais ele deve operar.

- Os requisitos do sistema são descrições mais detalhadas das funções, serviços e restrições operacionais do sistema de software. O documento de requisitos do sistema (às vezes chamado de especificação funcional) deve definir exatamente o que deve ser implementado.

Wieggers e Beatty (2013) define um terceiro nível de descrição dos requisitos, a saber, os requisitos de negócio. Os requisitos de negócios descrevem por que a organização está implementando o sistema - os benefícios comerciais que a organização espera alcançar. O foco está nos objetivos de negócios da organização ou no cliente que solicita o sistema.

Os requisitos de sistema são frequentemente classificados como requisitos funcionais ou requisitos não funcionais:

- 1) Requisito funcional são declarações de serviços que o sistema deve prover, como o sistema deve reagir a certas entradas e como deve se comportar em determinadas situações. Em alguns casos, deve, também, declarar o que o sistema não deve fazer.
- 2) Requisito não funcional são restrições sobre os serviços ou funções ofertadas pelo sistema.

O levantamento de requisitos é uma atividade crítica num processo de desenvolvimento de requisitos. O sucesso ou fracasso deste processo baseia-se no reconhecimento das partes interessadas relevantes e na detecção e descoberta de suas necessidades, bem como na qualidade dos requisitos (HANIF; RIZVI, 2010). Para dar suporte a esse processo, é importante que os requisitos sejam documentados e armazenados de forma compartilhada a fim de, caso haja mudança de equipe no projeto, evitar o retrabalho (WIEGERS; BEATTY, 2013).

Segundo Pressman e Maxim (2016), as funções e as características do sistema tornam-se mais evidentes quando os requisitos são coletados. Então são criados cenários para identificar o uso do sistema a ser desenvolvido. Esses cenários podem ser representados por casos de uso, que fornecem uma descrição de como o sistema será usado.

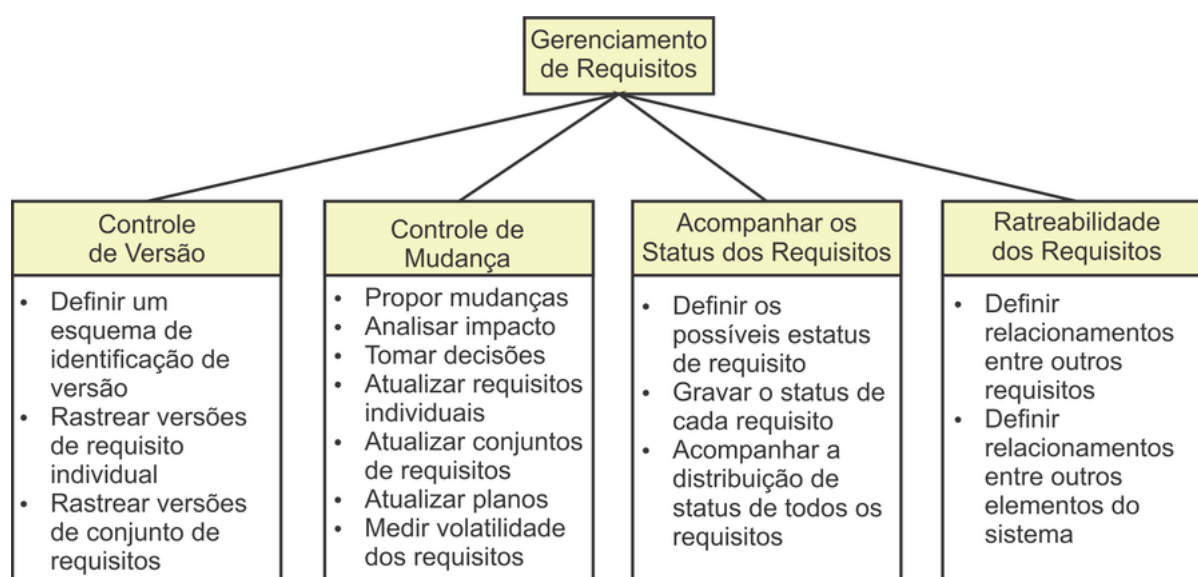
2.2 Gestão de Requisitos

A engenharia de requisitos está presente em todos os ciclos de vida de desenvolvimento – seja ele cascata, iterativo, incremental ou ágil. Ela prevê uma série de atividades que, a depender do ciclo de vida adotado, serão aplicadas em tempos diferentes em um projeto com variação no grau de profundidade e detalhe. Ela é uma

importante etapa em projeto de software e, quando não realizada adequadamente, pode levar o projeto ao fracasso.

Segundo Wiegers e Beatty (2013), a engenharia de requisitos é dividida em duas áreas: desenvolvimento e gerenciamento de requisitos. Ele divide o desenvolvimento de requisitos em quatro subdisciplinas, a saber, elicitação, análise, especificação e validação, todas relacionadas às atividades de exploração, validação, documentação e confirmação dos requisitos de um produto. Já as atividades relacionadas à gestão de requisitos são controle de versão, controle de mudança, acompanhamento do status do requisito e rastreabilidade do requisito (o detalhamento de cada atividade pode ser visualizado na Figura 1). O objetivo do gerenciamento de requisitos é garantir que a organização valide e atenda às necessidades de seus clientes e partes interessadas externas e internas (AHA! LABS INC, 2018).

Figura 1 – Principais atividades da gestão de requisitos



Fonte: Wiegers e Beatty (2013)

O gerenciamento de requisitos é de fundamental importância no processo desenvolvimento de um software, tanto que ela é exigência de dois modelos de maturidade de desenvolvimento de software: CMMI e MPS.BR. No modelo de maturidade CMMI, a exigência do gerenciamento de requisitos é feita no nível 2, primeiro nível de maturidade. Entre os objetivos específicos relacionados à gestão de requisitos está a manutenção da rastreabilidade bidirecional dos requisitos. Esse tipo de rastreabilidade é definido como uma associação entre duas ou mais entidades lógicas que é perceptível em qualquer direção (ou seja, para e de uma entidade) (SEI, 2010). No MPS.BR, o gerenciamento de requisitos é exigido também no primeiro nível, o nível G, e tem como resultado esperado a rastreabilidade bidirecional (SOFTEX, 2013).

Nesta seção, serão abordados cada atividade da gestão de requisitos e técnicas para realizá-las.

2.2.1 Controle de Versão de Requisitos

O controle de versão de requisitos consiste em identificar unicamente suas diferentes versões. Identificar as versões dos requisitos ou conjunto de requisitos permite rastrear as mudanças que eles sofreram ao longo do tempo. Sempre que um requisito sofrer alteração, é imprescindível que ele seja versionado e identificado. As mudanças devem ser claramente documentadas e comunicadas a todas as partes interessadas, pois é fundamental que todos tenham acesso à versão atual dos requisitos.

Quando uma nova versão de um requisito é criada, é necessário identificar a data da mudança, quem realizou a mudança e a razão por trás de cada mudança. Uma forma robusta de manter todas essas informações é através do uso de uma ferramenta de apoio ao gerenciamento de requisitos. Uma ferramenta de apoio ao gerenciamento de requisitos pode automatizar todo o processo de controle de versão, além disso, permitir que um requisito seja revertido a uma versão anterior.

Caso os requisitos sejam armazenados em documentos, pode-se fazer uso da funcionalidade de revisão de processadores de texto. Tal funcionalidade destaca as alterações realizadas no documento identificando visualmente inserções e deleções. Assim, é possível aprovar ou rejeitar uma alteração e, até mesmo, voltar à versão anterior.

2.2.2 Controle de Mudanças

Grande parte da literatura clássica de engenharia de software parte da suposição de que os requisitos são fixos. Um exemplo está no modelo de desenvolvimento de software em cascata (*Waterfall Model*) que congelam os seus requisitos antes da fase de desenvolvimento. Porém, atualmente, é bem aceito que requisitos evoluem e mudanças podem acontecer em qualquer fase do processo de desenvolvimento.

Congelar os requisitos para um novo sistema logo após as atividades iniciais de elicitação é imprudente e irrealista. Em vez disso, estabeleça uma linha de base quando achar que um conjunto de requisitos está suficientemente definido para o início da construção e, em seguida, gerencie as alterações para minimizar seu impacto negativo no projeto (WIEGERS; BEATTY, 2013).

Alguns requisitos são mais suscetíveis a mudança que outros. Harker e Eason (1993) classifica os requisitos em estáveis e voláteis. Os requisitos estáveis derivam da atividade principal da organização e que se relacionam diretamente com o domínio do sistema. Por exemplo, em um hospital sempre haverá requisitos relacionados a

pacientes, médicos, enfermeiros, tratamentos, etc. Já os requisitos voláteis serão provavelmente alterados durante o processo de desenvolvimento do sistema ou após o sistema ter se tornado operacional. Exemplos de requisitos voláteis são requisitos resultantes de políticas públicas de saúde ou mecanismos de cobrança de assistência médica (SOMMERVILLE, 2010). Harker e Eason (1993) classifica os requisitos voláteis em 5 classes de acordo com a origem das mudanças (Tabela 1):

Tabela 1 – Classes de requisitos voláteis

Tipo de Requisito	Descrição
Mutável	Requisitos que mudam devido a alterações no ambiente em que a organização está operando. Por exemplo, nos sistemas hospitalares, o financiamento da assistência ao paciente pode mudar e, assim, exigir que diferentes informações de tratamento sejam coletadas.
Emergente	Requisitos que surgem à medida que o entendimento do cliente sobre o sistema se desenvolve durante o desenvolvimento do sistema. O processo de design pode revelar novos requisitos emergentes.
Consequencial	Requisitos que resultam da introdução do sistema de computador. A introdução do sistema de computador pode alterar os processos da organização e abrir novas formas de trabalho que geram novos requisitos de sistema.
Compatibilidade	Requisitos que dependem dos sistemas ou processos de negócios específicos de uma organização. Como isso muda, os requisitos de compatibilidade no sistema comissionado ou entregue também podem ter que evoluir.

É comum pensar que a maior parte da atividade de manutenção de software envolve a correção de defeitos do sistema, mas um estudo mostra que 80% do esforço de manutenção está relacionado a ações não corretivas (PIGOSKI, 1996). Essas ações estão diretamente ligadas a mudanças nos requisitos. Então, uma organização deve estar preparada para lidar com esse cenário.

2.2.2.1 Processo de Controle de Mudanças

Um processo de controle de mudança é essencial para decidir se os benefícios de implementar uma mudança justifica o custo de sua implementação. Uma vez definido um processo, ele deve ser documentado e políticas de como a equipe deve tratar

mudanças nos requisitos e demais artefatos do projeto devem ser comunicadas (WIEGERS; BEATTY, 2013; SOMMERVILLE, 2010).

Um processo de controle de mudanças deve conter os critérios de entrada, que determinam se a execução do processo deve ser iniciada, as atividades envolvidas no processo, os papéis responsáveis por cada atividade e os critérios que determinam que o processo foi encerrado com sucesso (TAYLOR, 2007; WIEGERS; BEATTY, 2013).

Um processo de controle de mudanças pode ser descrito nas seguintes atividades:

- **Análise do problema ou proposta de mudança:** O processo se inicia com a identificação de um problema de requisito ou uma proposta de mudança. Durante essa fase, o problema ou a proposta é analisada e verificada sua validade de acordo com os critérios de entrada. Caso não atenda os critérios, a origem do problema ou proposta da mudança é comunicada, a qual pode fornecer mais informações ou cancelar o pedido (SOMMERVILLE, 2010).
- **Avaliar o pedido de mudança:** Nessa fase devem ser feitas a análise de impacto e avaliação de risco. Uma categoria de risco deve ser atribuída à mudança proposta: alto, moderado ou baixo risco. A mudança de alto risco requer muitas etapas adicionais, como a aprovação da gerência e a notificação das partes interessadas, enquanto a mudança de baixo risco pode exigir apenas aprovação do gerente de projeto e documentação mínima (TAYLOR, 2007). Nessa fase também é levantado o custo de realizar a mudança em termos de documentação e desenvolvimento (SOMMERVILLE, 2010; WIEGERS; BEATTY, 2013).
- **Tomada de decisão:** Os responsáveis por tomar decisões devem decidir se aprovam ou rejeitam a mudança. A cada pedido de mudança aprovado uma data de implementação é atribuída, ou a mudança é incorporada a uma interação do projeto ou *release*. Pode-se simplesmente adicionar um novo requisito ao *backlog* do produto como pendente. Os responsáveis devem atualizar o status do pedido de mudança e notificar todos os membros da equipe afetados (OPERATIONAL EXCELLENCE PROGRAM OFFICE, ; WIEGERS; BEATTY, 2013).
- **Implementação da mudança:** Os documentos de requisitos, quando necessário, os artefatos do projeto e a implementação são modificados. As informações de rastreabilidade são de grande importância para identificar os requisitos e artefatos que foram afetados pela mudança (SOMMERVILLE, 2010; WIEGERS; BEATTY, 2013).

- **Verificar a mudança:** Vários membros da equipe podem verificar as alterações feitas em vários artefatos do projeto por meio de testes ou revisões. Após a conclusão da verificação, os artefatos atualizados são armazenados em locais apropriados e disponibilizados para todos os envolvidos no projeto (WIEGERS; BEATTY, 2013).

2.2.3 Acompanhamento do Status dos Requisitos

Acompanhar o *status* de cada requisito é um meio de identificar o *status* geral de um projeto. Realizar esta atividade facilita a identificação de métricas como quantos requisitos ainda estão pendentes de aprovação pelas partes interessadas; quantos requisitos foram adiados, o que potencialmente mudará o escopo (COVENTRY, 2015). Manter os *status* dos requisitos permite evitar respostas do tipo “90% dos requisitos foram concluídos” vindas dos desenvolvedores, quando perguntando sobre o status do projeto. Esse tipo de resposta não diz muito sobre o status do projeto devido à síndrome dos 90%. Essa síndrome descreve um projeto que atinge estado de cerca de 90% concluído, mas acaba levando o mesmo tempo utilizado para se concluir os 90% do projeto a mais para concluí-lo de fato (FORD; STERMAN, 2003).

Wieggers e Beatty (2013) sugere 10 status de requisito, os quais estão descritos na Tabela 2. Classificar os requisitos em status agrega mais significado do que tentar definir porcentagens para cada um. Com status, é possível gerar relatórios com informações do tipo: dos 100 requisitos do sistema, 75 foram implementados e verificados, 15 foram implementados, mas ainda não foram verificados, e 10 ainda não foram implementados.

Tabela 2 – Status de requisito sugeridos

Proposto	O requisito foi solicitado por uma fonte autorizada.
Em progresso	Um analista de negócios está trabalhando ativamente na elaboração do requisito.
Esboçado	A versão inicial do requisito foi gravada.
Aprovado	O requisito foi analisado, seu impacto no projeto foi estimado e foi alocado à <i>baseline</i> para uma liberação específica. Os principais interessados concordaram em incorporar o requisito e o grupo de desenvolvimento de software se comprometeu a implementá-lo.
Implementado	O código que implementa o requisito foi projetado, escrito e testado. O requisito foi rastreado para os elementos de <i>design</i> e código. O software que implementou o requisito agora está pronto para teste, revisão ou outra verificação.
Verificado	O requisito satisfaz os seus critérios de aceitação, o que significa que o funcionamento correto do requisito implementado foi confirmado. O requisito foi rastreado para testes pertinentes. Agora é considerado completo.
Adiado	Um requisito aprovado está agora planejado para implementação em uma versão posterior.
Deletado	Um requisito aprovado foi removido da <i>baseline</i> . Inclua uma explicação de por que e por quem a decisão foi tomada para excluí-lo.
Rejeitado	O requisito foi proposto, mas nunca foi aprovado e não está planejado para implementação em qualquer <i>release</i> futura. Inclua uma explicação de por que e por quem foi tomada a decisão de rejeitá-lo.

Wieggers e Beatty (2013)

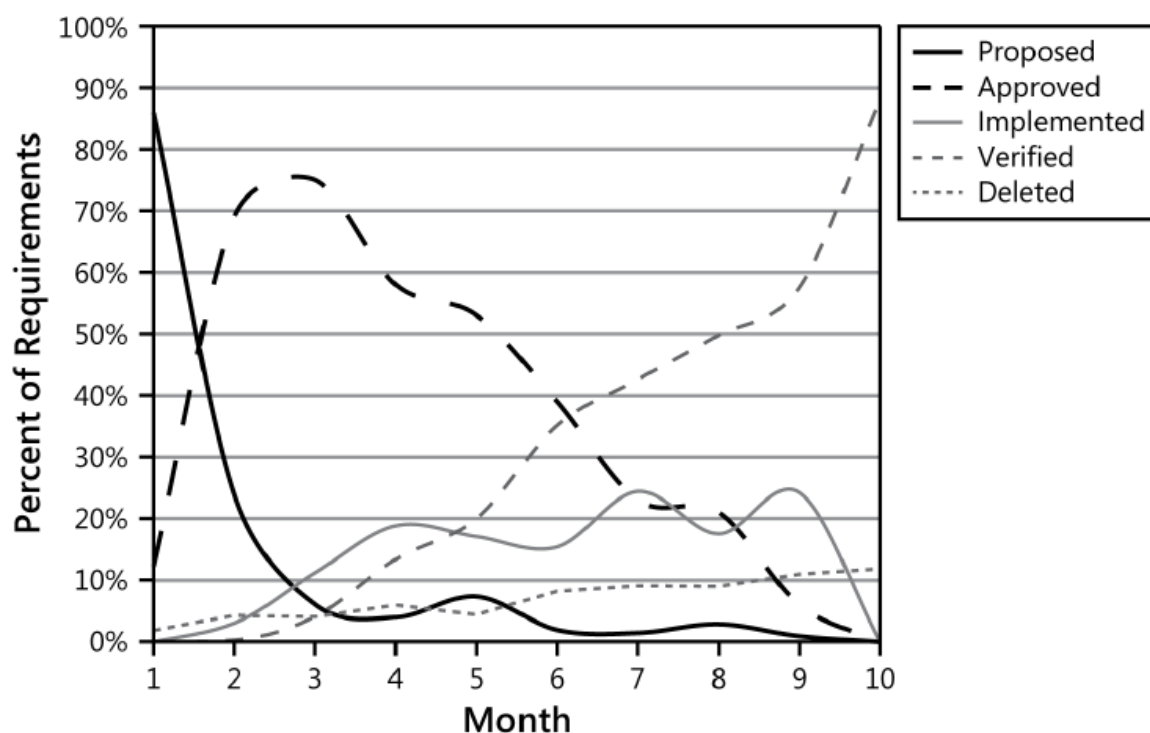
Para Wieggers e Beatty (2013) é importante manter um registro dos requisitos rejeitados e os motivos pelos quais eles foram rejeitados. Os requisitos rejeitados tendem a ressurgir mais tarde durante o desenvolvimento ou em um projeto futuro. O status Rejeitado permite manter um requisito proposto disponível para possível referência futura sem sobrecarregar o conjunto de requisitos comprometidos de uma

versão específica.

O status de um requisito deve ser atualizado apenas quando a condição de transição for atendida. Certas atualizações requerem que os relacionamentos do requisito sejam atualizados, como no caso de um requisito com status de implementado. Nesse caso deve-se relacionar o requisito com o código que o implementa.

A Figura 2 apresenta como os status de um conjunto de requisitos podem ser monitorados visualmente através do tempo de desenvolvimento de um projeto. Ela mostra o percentual dos requisitos do sistema com cada status ao final de cada mês. As curvas ilustram como o projeto está se aproximando de sua meta de verificação completa de todos os requisitos aprovados.

Figura 2 – Acompanhamento da distribuição do status dos requisitos ao longo do ciclo de desenvolvimento de um projeto



Fonte: Wiegers e Beatty (2013)

2.2.4 Rastreabilidade de Requisitos

A rastreabilidade de requisitos é de grande importância em um processo de desenvolvimento de software, visto que os requisitos podem mudar de acordo com diversos fatores internos ou externos de uma organização. Esses fatores podem ocasionar mudanças nas regras de negócio de uma organização e por fim impactarem nos requisitos do produto de software. Isso é inevitável, então é preciso gerir essas mudanças.

A principal fonte de mudanças são os stakeholders. Eles, normalmente, não possuem uma boa clareza do que querem no início de um projeto. Quando se segue para fase de desenvolvimento, novas funcionalidades podem ser descobertas e incorporadas ao produto. Com um processo de gerenciamento de requisitos, tem-se um maior controle dessas mudanças e a rastreabilidade de requisitos é um ponto chave na identificação e documentação dos impactos e na tomada de decisão acerca do gerenciamento do projeto (mudança de cronograma, aumento de custo).

Wieggers e Beatty (2013) entendem que requisitos excelentes devem ser rastreáveis. Para que eles sejam rastreáveis, cada um deve ser rotulado de forma única e persistente, para que se possa consultá-lo de maneira inequívoca durante o projeto. Além disso, um requisito rastreável deve estar vinculado à sua origem (quem solicitou), a requisitos derivados, elementos de design, código que o implementa e testes que verificam sua implementação.

Para Sommerville (2010), uma ferramenta para rastreabilidade é de grande importância na gestão de requisitos em grandes projetos. Ela pode ser útil na realização de algumas tarefas relacionadas à gestão de requisitos (WIEGERS; BEATTY, 2013):

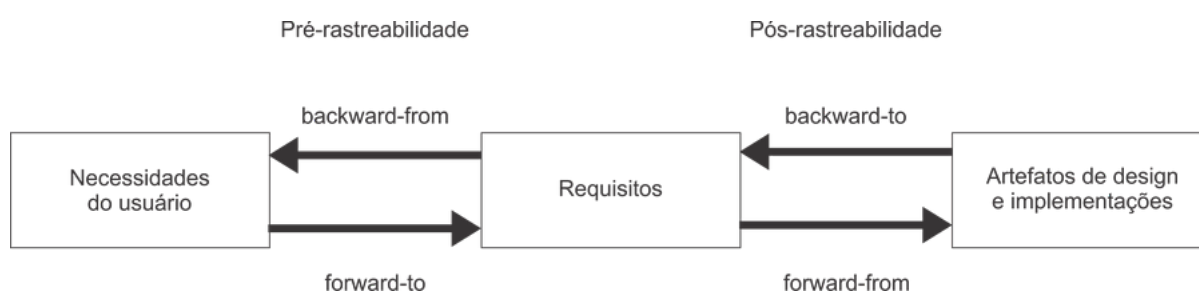
- Gerenciar versões e alterações, ou seja, manter um histórico das alterações feitas em cada requisito, registrar a lógica por trás de cada decisão de alteração e reverter para uma versão anterior de um requisito, se necessário.
- Armazenar atributos de requisitos, como a data em que um requisito foi criado, seu número de versão atual e número de release e, ainda, definir atributos adicionais de vários tipos de dados.
- Analisar o impacto que uma alteração proposta em um requisito específico terá em outros elementos do sistema. Por exemplo, pode-se consultar uma lista de todos os requisitos originados de uma regra de negócio específica, para que se possa julgar as consequências de uma mudança nessa regra.
- Identificar a falta de requisitos, como requisitos do usuário que não possuem requisitos funcionais mapeados, assim como, identificar requisitos que não podem ser rastreados até uma origem razoável, levantando a questão de saber se esses requisitos são necessários.
- Acompanhar o status dos requisitos e, assim, acompanhar o status do projeto.
- Definir permissões de acesso para indivíduos ou grupos de usuários e compartilhar informações com uma equipe geograficamente dispersa por meio de uma interface da Web.

- Reutilizar requisitos em vários projetos ou subprojetos, armazenando-os uma única vez e referenciando-os sempre que necessário para evitar duplicidade.

2.2.4.1 Tipos de Relacionamentos

A rastreabilidade pode ser dividida em dois grupos: pré-rastreabilidade e pós-rastreabilidade. A pré-rastreabilidade conceitua os relacionamentos dos requisitos com sua origem; a pós-rastreabilidade, por exemplo, relaciona os requisitos com os elementos do projeto de design, casos de teste e código (POHL, 1996).

Figura 3 – Diagrama de Tipos de Links de Rastreamento



Fonte: Wiegers e Beatty (2013)

A Figura 3 ilustra quatro tipos de links de rastreamento. No grupo pré-rastreabilidade temos o primeiro, *forward-to* (para frente), que liga as necessidades do usuário aos requisitos, assim é possível identificar quais requisitos serão alterados caso a necessidade do usuário mude. O link *backward-from* (para trás de) relaciona os requisitos a sua origem. No grupo pós-rastreabilidade temos o link *forward-from* (para frente de). Este tipo de link permite determinar que um requisito foi atendido, pois é possível identificar os elementos de projeto de design e elementos de codificados. Já o link *backward-to* relaciona elementos específicos do produto aos requisitos, o que permite identificar a razão desses elementos terem sido criados (WIEGERS; BEATTY, 2013).

A literatura relata ainda um outro tipo de link, o link bidirecional. Esse tipo de link pode ser navegados em ambas direções, da origem ao destino e de forma reversa (GOTEL et al., 2012). Neste projeto é utilizado este último conceito.

Uma outra classificação de tipos de rastreabilidade são: rastreabilidade vertical e horizontal. O termo rastreamento vertical, no contexto de engenharia de software, é usado ao analisar artefatos em diferentes níveis de abstração, de modo a acomodar todo o ciclo de vida ou a rastreabilidade de ponta a ponta, como do requisito ao código. Já o termo rastreamento horizontal é comumente usado ao rastrear artefatos no mesmo nível de abstração, como relacionar os requisitos que tratam da performance do sistema (GOTEL et al., 2012).

2.2.4.2 Visualização das Informações de Rastreabilidade

Um objetivo da rastreabilidade é visualizar a relação entre os artefatos. À medida que o número e a complexidade dos *links* de rastreamento aumentam, são necessárias técnicas de visualização de rastreabilidade. Uma visualização pode incluir informações sobre os artefatos (por exemplo, tipo de artefato, metadados, atributos) e *links* (por exemplo, tipo do *link*, metadados, força do *link*). Visualizações comuns para informações de rastreabilidade são matrizes, gráficos, listas e *hiperlink*.

A maneira mais comum de representar os *links* entre requisitos e outros elementos do sistema está em uma matriz de rastreabilidade de requisitos, também chamada de matriz de rastreio de requisitos ou uma tabela de rastreabilidade (WIEGERS; BEATTY, 2013). Ela mapeia requisitos e outros artefatos em uma representação bidimensional. Tipicamente, as colunas representam os requisitos e as linhas os artefatos. Se um elemento $a_{i,j}$ estiver marcado, isso significa que o requisito da coluna j está relacionado com o artefato da linha i (LI; MAALEJ, 2012).

Gráficos permitem a visualização de relacionamentos multidimensional entre requisitos e outros artefatos. Em um gráfico, os artefatos são representados como nós e os relacionamentos são arestas entre eles. Os gráficos são especialmente adequados para tarefas de desenvolvimento. Eles permitem obter uma visão geral dos links de forma explorativa e são caracterizados por uma alta taxa de compreensão de informações (LI; MAALEJ, 2012).

A lista representa cada relacionamento, junto com suas informações de origem e destino, assim como seus atributos, em uma única entrada. Ela é frequentemente utilizada em processo de recuperação links, onde os links são exibidos de forma sequencial obedecendo uma ordem de similaridade com a consulta (LI; MAALEJ, 2012).

Hyperlinks permite que usuários possam, enquanto navegando em um artefato, acessar outros artefatos facilmente através de links. Hyperlinks conecta conceitos relacionados, palavras-chave ou frase de forma natural (LI; MAALEJ, 2012).

2.3 Ferramentas Gratuitas para Gestão de Requisitos

Nesta seção serão apresentadas algumas ferramentas de código aberto de apoio a gestão de requisitos relacionadas ao controle de mudanças e rastreabilidade de requisitos. Ferramentas como estas são difíceis de encontrar, são antigas ou foram descontinuadas.

2.3.1 SIGERAR

O SIGERAR (GRANDE; MARTINS, 2006) é operado via interface Web e foi desenvolvido em linguagem Java com JSP (Java Server Pages) e com o SGBD (Sistema Gerenciador de Banco de Dados) Firebird 1.5.

Ela tem como principais características o tratamento da rastreabilidade dos requisitos no processo de alteração. A ferramenta permite análise e atribuição de valores de risco, importância, impacto, prioridade e custo a todos requisitos envolvidos (origem e dependentes), que auxilia ao Gerente do Projeto a analisar o contexto do impacto e custos da alteração.

2.3.2 OpenReq

O OpenReq (GRINGS; SAYÃO, 2009) foi construído para a plataforma Web. Arquitetado em três camadas, ele implementado em linguagem Java, fornece uma estrutura básica de funcionalidades onde novas extensões (plug-ins) podem ser adicionadas, assim ampliando suas funções. Os requisitos são criados através de um editor próprio, possui um controle de versões e apoia o processo de controle de mudanças.

As funcionalidades implementadas no OpenReq estão agrupadas em:

- **Segurança:** componente responsável por validar o acesso dos usuários ao sistema através de login e senha;
- **Controle de Acesso:** responsável por garantir o acesso apenas aos usuários cadastrados, restringindo suas funcionalidades conforme as políticas de permissão do usuário;
- **Gerência de Módulos:** permite a adição, configuração e gerência de módulos da ferramenta;
- **Gerência de Usuários:** permite criar e gerenciar os usuários que terão acesso a ferramenta;
- **Controle de Mudanças:** gerencia as alterações ocorridas nos artefatos gerenciados pelo sistema;

2.4 Conclusão

A gestão de requisitos prevê diversas técnicas para manter controle sobre os requisitos, definindo um processo de controle de mudança, que é essencial garantir o sucesso de um projeto. Uma das técnicas prevista é a rastreabilidade de requisitos.

A rastreabilidade de requisitos pode auxiliar na atividade de análise de impactos e riscos, o que permite tomar decisões mais acertadas no momento de incorporar alguma mudança nos requisitos ao projeto. Existem diversas técnicas de visualização de artefatos, porém a mais utilizada é a matriz de rastreabilidade.

3 A Ferramenta de Rastreabilidade de Requisitos

Para construção da ferramenta de rastreabilidade de requisitos (FRR), os requisitos foram identificados junto à empresa de tecnologia Resource IT Solution, a qual será designada como Resource. Foram entrevistados alguns funcionários da empresa que trabalham com o projeto de desenvolvimento do PJe. Entre os funcionários entrevistados havia 1 gerente de projeto, 7 analistas de sistemas, 2 desenvolvedores e 2 testadores. Os requisitos foram levantados através de entrevistas, nas quais os funcionários expuseram suas necessidades para desempenhar suas atividades e dificuldades no uso das soluções implantadas no projeto para gestão de requisitos. As soluções implantadas, no momento desse trabalho, correspondiam a uma planilha do Excel online, na qual eram armazenados requisitos funcionais e não funcionais, regras de negócio, domínio, interface e mensagens, além de referências aos artefatos gerados, tais como, documento de visão, especificação de requisitos, entre outros. Os artefatos propriamente dito eram armazenados no SVN.

3.1 Funcionalidades

Como a empresa possui diversos projetos, um dos requisitos da FRR é permitir a gestão de projetos individualmente. Como cada projeto tem suas particularidades, como conjunto de artefatos gerados e metodologia de desenvolvimento, a FRR permite o cadastro dos tipos de artefatos a serem rastreados por projeto. Os tipos de relacionamentos também são customizados conforme necessidade de cada projeto.

Devido ao tamanho do projeto de desenvolvimento do PJe, o seu desenvolvimento é dividido em demandas menores. Essa visão foi incorporada à FRR. Desse modo, os requisitos e artefatos podem ser agrupados em demandas, permitindo uma melhor visualização dos relacionamentos entre os artefatos e determinando uma *baseline* para cada fase de evolução do projeto. Como um artefato pode estar relacionado a diversas demandas, a FRR permite a fácil identificação dessas demandas. Quando isso acontece, os artefatos são destacados, chamando a atenção do analista para possíveis conflitos.

Para assistir na atividade de análise de impactos, a FRR permite relacionar dois artefatos através de tipos de relacionamentos. Esses relacionamentos são exibidos em uma matriz de rastreabilidade, onde, nessa matriz, as linhas representam a origem do relacionamento e as colunas, o destino. A matriz de rastreabilidade dispõe de filtros para que o usuário da FRR possa visualizar os artefatos de origem e destino desejados, os tipos de relacionamentos, além de permitir a visualização dos relacionamentos por demandas ou pelo projeto como um todo. Os tipos de relacionamentos são representados por cores no preenchimento dos elementos da matriz.

Para assistir o gerente de projeto e suas atividades, tais como acompanhamento do status do projeto ou demanda e o acompanhamento do status de um artefato, a FRR permite a informação de atributos nos artefatos, tais como, status, data de criação e atualização. O status do projeto ou demanda é apresentado em um gráfico de distribuição, no qual é possível identificar o percentual de artefatos em determinado status ao longo do tempo.

Para atividade de controle de versão, a FRR versiona cada artefato quando atualizado, permitindo reverter um artefato para uma versão anterior. A FRR mantém informações de quando o artefato foi modificado e por quem foi feita a mudança. A FRR permite também o cadastro de comentários nos artefatos, assim, o usuário poderá justificar uma mudança ou adicionar informações relevantes para o desenvolvimento de um requisito.

Como alguns artefatos mantidos pela empresa são arquivos de texto binário, como documentos do Word e PDF, a FRR permite que arquivos sejam anexados a um artefato e, sempre que um artefato tiver o arquivo substituído ou removido, também é gerada uma versão do artefato juntamente com o anexo.

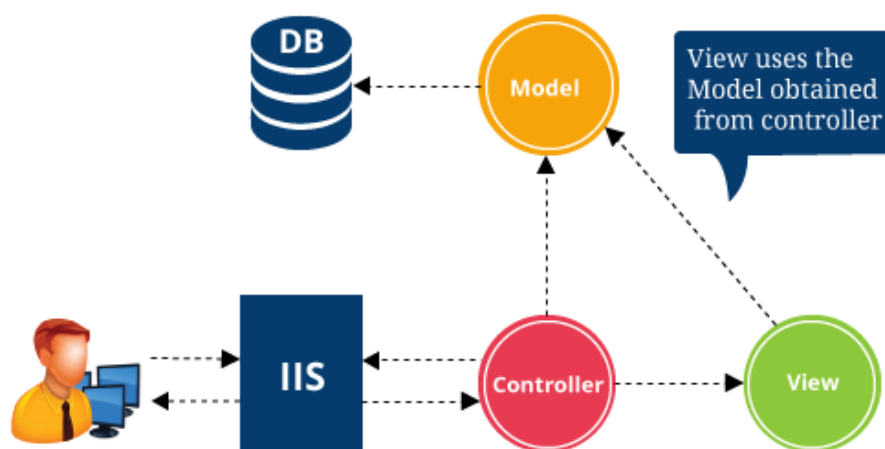
Como cada funcionário possui funções distintas na empresa, a FRR permite o cadastro de usuários com múltiplos papéis e, para manter a integridade dos artefatos, cada papel possui permissões diferenciadas. A lista completa de papéis e permissões pode ser visualizada no Apêndice E.30. Um usuário só pode visualizar os artefatos e demandas do projeto que ele faz parte.

As listas completas dos requisitos funcionais (Apêndice A) e não funcionais (Apêndice B), casos de uso e alguns diagramas de sequências (Apêndice F) relevantes são encontradas nos apêndices.

3.2 Arquitetura

O projeto da FRR foi desenvolvido utilizando o *framework* Rails, o qual utiliza a linguagem orientada a objeto Ruby. O *framework* faz uso do padrão MVC (do inglês *Model-View-Controller*), o que divide a aplicação em três camadas (Figura 4) com responsabilidade específica. A camada *Model* encapsula a lógica de negócio; a camada *Controller* lida com as requisições HTTP (do inglês *Hypertext Transfer Protocol*) provendo respostas adequadas; já a camada *View* provê representações adequadas dos recursos da aplicação como, por exemplo, a resposta de um *controller* (RUBY ON RAILS ORGANIZATION, 2018).

Figura 4 – Diagrama da Arquitetura MVC



Retirada do site <https://www.credosystemz.com>

3.3 Diagrama de Classes

Nos diagramas de classes apresentados nas Figuras 5 e 6, é possível visualizar todas as classes do projeto agrupadas por camadas da arquitetura MVC. Neste projeto, as *views*, representadas pelas classes de fronteiras, são as telas do sistema codificadas em HTML5 (do inglês HyperText Markup Language), javascript e CSS. As classes de controle, agrupadas na camada de controle, são responsáveis por receber uma requisição HTTP e responder com a representação adequada (*views*). Já na camada de modelos, estão as entidades e classes auxiliares à lógica da ferramenta. Neste diagrama é possível visualizar todas as dependências, relacionamentos e cardinalidade entre cada elemento do projeto.

Figura 5 – Diagrama de classe - Parte 1

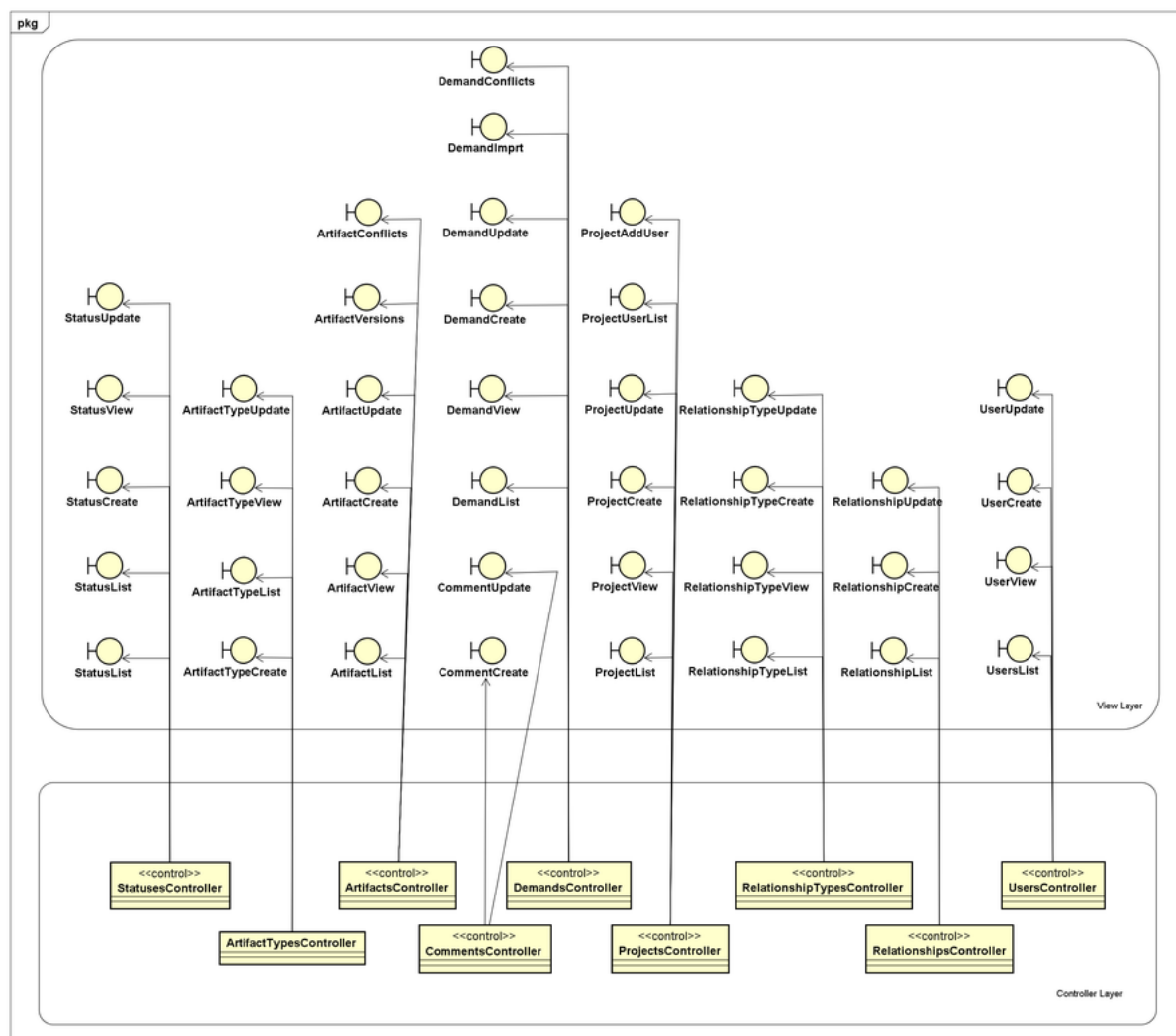
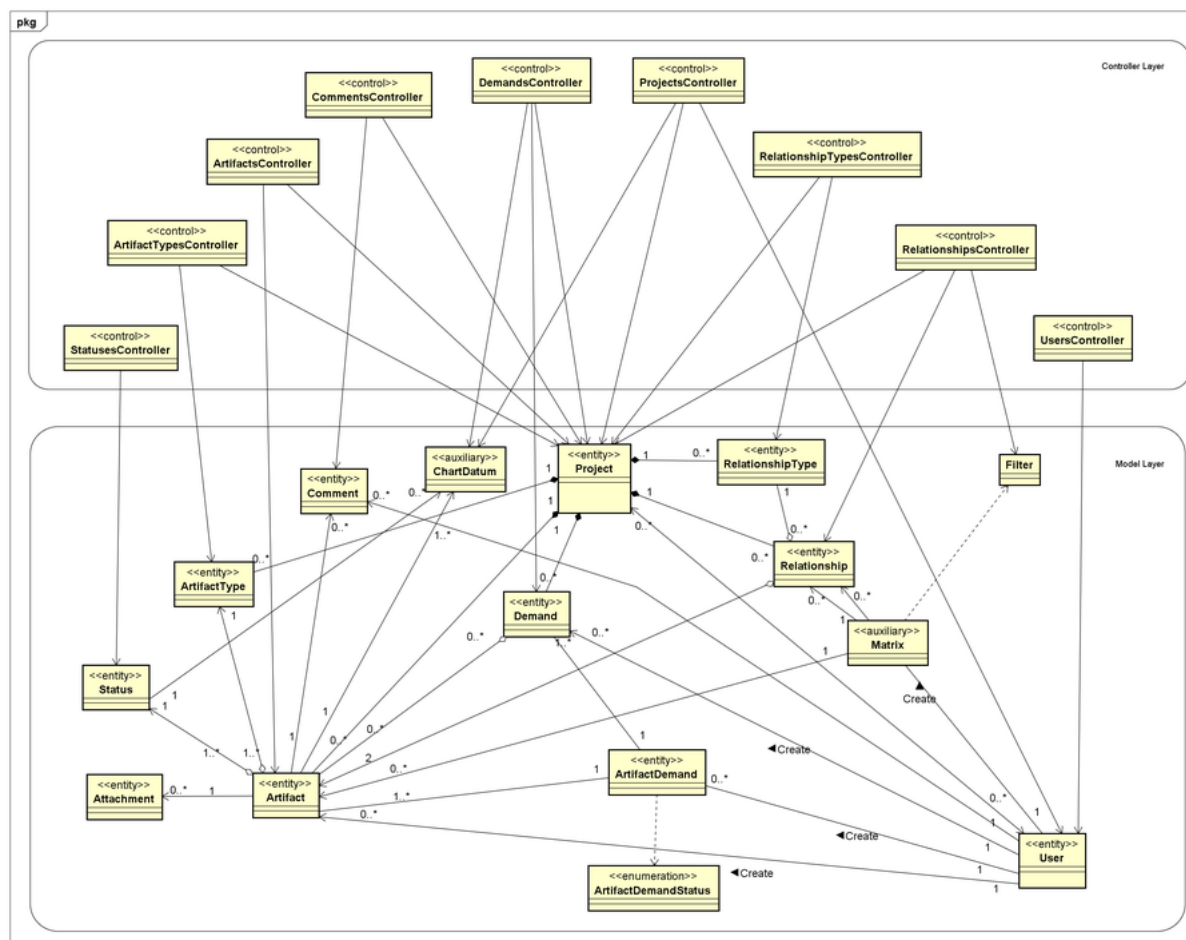


Figura 6 – Diagrama de classe - Parte 2



3.4 Modelo de Dados

Um modelo de dados descreve a representação lógica e física dos dados persistentes no sistema (ROTATIONAL SOFTWARE CORPORATION, 2001). O modelo de dados da FRR, apresentado na Figura 7, contém as seguintes tabelas:

3.4.1 Users

A tabela *users* representa todos os usuários da ferramenta. Nela estão as principais informações do usuário, tais como o nome (*username*), email, papéis (*roles_mask*). As demais informações estão relacionadas com o controle de acesso da ferramenta.

3.4.2 Projects

A tabela *projects* representa todos os projetos da ferramenta. Nela estão armazenados o nome (*name*) do projeto, sua descrição (*description*) e usuário (*user_id*) que o criou. o *user_id* é uma referência a tabela *users*.

3.4.3 ProjectsUsers

A tabela *projects_users* representa os usuários de um projeto. O *user_id* é uma referência a tabela *users* e *project_id*, a tabela *projects*. Essa tabela permite que um usuário possa estar incluído em mais de um projeto, assim como, um projeto conter diversos usuários.

3.4.4 ArtifactTypes

A tabela *artifact_types* representa os tipos de artefatos da ferramenta. Nela estão armazenados o nome (*name*) do artefato e a referência (*project_id*) à tabela *projects*. Um tipo de artefato pertence unicamente a um projeto.

3.4.5 ArtifactStatuses

A tabela *artifact_statuses* representa os status possíveis de serem atribuídos a um artefato. Nela, é armazenada a descrição (*name*) do status e a que projeto ele pertence (*project_id*). Cada status pertence a um único projeto.

3.4.6 Artifacts

A tabela *artifacts* representa os artefatos da ferramenta. Nela estão armazenados o código (*code*) do artefato, seu nome (*name*), descrição (*description*), prioridade (*priority*) e o status (*artifact_status_id*), o qual referencia a tabela *artifact_statuses*. O tipo do artefato (*artifact_type_id*) é uma referência a tabela *artifact_types*, assim como o usuário que o criou (*user_id*) e o projeto a que ele pertence (*project_id*) são referências às tabelas *users* e *projects* respectivamente. Como um artefato pode ter um arquivo anexo, existe uma referência (*attachment_id*) para tabela *attachments*.

3.4.7 Versions

A tabela *versions* pode conter versões de qualquer objeto visto que ele é salvo como uma *string*. Ela é mantida automaticamente por uma biblioteca chamada *PaperTrail*. Para esse projeto, foi escolhido manter apenas as versões dos artefatos. Informações adicionais são salvas, tais como o tipo de item (*item_type*), o identificador do item (*item_id*), o evento que gerou a versão (*event*), podendo ser de criação ou atualização, e o usuário que gerou o evento (*whodunnit*).

3.4.8 Attachments

A tabela *attachments* representa o anexo de um artefato. Nela estão armazenados o artefato a que o anexo pertence (*artifact_id*) e a versão do artefato (*artifact_version_index*). a coluna *artifact_id* é uma referência a tabela *artifacts*. Os anexos físicos são mantidos por um sistema de arquivo.

3.4.9 Comments

A tabela *comments* representa os comentários de um artefato. Como um comentário pertence a um único artefato, existe na tabela uma referência (*artifact_id*) a tabela *artifacts*. Esta tabela armazena também informações de quem criou o comentário (*user_id*), que é uma referência a tabela *users*, e o conteúdo (*content*).

3.4.10 ChartData

A tabela *chart_data* armazena os dados necessários à exibição do gráfico de distribuição dos status dos artefatos ao longo do tempo do projeto (Figura 2). Nela contém uma referência para o artefato (*artifact_id*) e uma referência para o status do artefato (*artifact_status_id*) no momento em que ele foi criado ou teve seu status alterado.

3.4.11 RelationshipTypes

A tabela *relationship_types* representa os tipos de relacionamentos de cada projeto. Nela estão armazenados o nome (*name*) do relacionamento e a referência (*project_id*) à tabela *projects*. Um tipo de relacionamento pertence unicamente a um projeto.

3.4.12 Relationships

A tabela *relationships* representa todos os relacionamentos da ferramenta. Nela temos o tipo de relacionamento (*relationship_type_id*), o artefato de origem (*origin_artifact_id*) e o artefato de destino (*end_artifact_id*) que são referências às tabelas *relationship_types* e *artifacts* respectivamente. Nessa tabela também estão armazenados a descrição do relacionamento (*description*) e a referência ao projeto (*project_id*) que o relacionamento pertence.

3.4.13 Demands

A tabela *demands* representa as demandas dos projetos. Como uma demanda pertence a um único projeto, é armazenada uma referência (*project_id*) à tabela *projects*. Uma demanda tem informações de quem a criou (*responsible_user_id*), que é uma referência a tabela *users*, o nome (*name*) da demanda e a descrição (*description*).

3.4.14 ArtifactDemands

Uma demanda pode conter diversos artefatos, assim como um artefato pode pertencer a diversas demandas. Devido esse cenário, foi necessário a criação da tabela *artifact_demands*. Nela são armazenadas referências as tabelas *artifacts* (*artifact_id*) e *demands* (*demand_id*). Nela também são armazenados o usuário que inseriu o artefato na demanda (*user_id*), a versão do artefato quando foi inserido (*version_index*) e o modo como ele foi inserido na demanda (*status*) - se importado ou criado diretamente na demanda. Esses dados são necessários para identificação de conflitos.

Figura 7 – Modelo de Dados



3.5 Conclusão

A arquitetura da FRR está baseada no padrão MVC. Este padrão agrupa as classes do projeto em três camadas (*model*, *view* e *controller*). As principais funcionalidades da FRR são a geração de uma matriz de rastreabilidade a partir dos relacionamentos entre artefatos, a possibilidade de rastrear os artefatos e o versionamento dos artefatos. Essas funcionalidades fornecem informações essenciais a gestão de requisitos.

4 Prova de Conceito

Neste capítulo são apresentando exemplos de uso da FRR. Para montagem do exemplo, foi carregado na FRR os requisitos e artefatos gerados durante o processo de desenvolvimento deste projeto.

4.1 Pré-rastreabilidade

Através da FRR, é possível determinar o que deu origem a um requisito. Isso é importante, pois caso um requisito não tenha origem, pode ser questionada sua necessidade. Na Figura 8, por exemplo, temos uma regra de negócio¹ que deu origem a um requisito funcional². Já na Figura 9, é possível identificar a origem o requisito a partir dele.

Figura 8 – Exemplo de pré-rastreabilidade



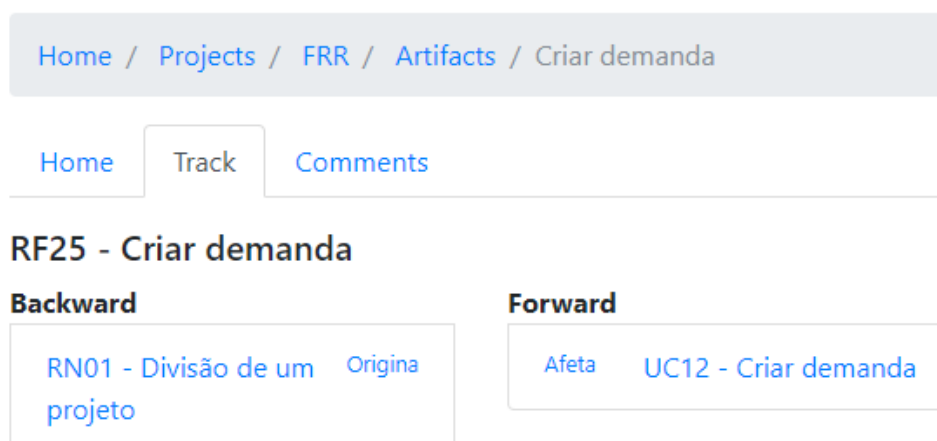
4.2 Pós-rastreabilidade

Um outro exemplo diz respeito a pós-rastreabilidade. Nesse exemplo, ilustrado na Figura 9, é possível rastrear os artefatos gerados a partir de um requisito. Neste caso, o requisito funcional RF25 gerou o caso de uso UC12, o qual explica em mais detalhe o requisito funcional.

¹ RN01 - Um projeto é subdivido em demandas.

² RF25 - A ferramenta deve permitir que os gerentes e analistas de um projeto possam criar demandas no projeto.

Figura 9 – Exemplo de pós-rastreabilidade



4.3 Rastreamento Bidirecional

As Figuras 8 e 9, exemplificam o rastreamento bidirecional, pois através de um único relacionamento, o link “origina”, é possível navegar para origem, RN01, ou para o destino, RF25. O mesmo é possível através do link “afeta”.

4.4 Rastreamento Horizontal

Através da matriz de rastreabilidade da FRR, é possível identificar e criar relacionamentos horizontais filtrando os artefatos pelo tipo na matriz, escolhendo os mesmos artefatos para linhas e colunas. A Figura 10, mostra que os requisitos funcionais RF01³ e o RF05⁴ dependem do requisito funcional RF02⁵. A dependência existe, pois, para completar o cadastro do usuário ou resetar uma senha, é necessário enviar um email para usuário com informações de como confirmar o cadastro ou alterar a senha.

³ RF01 - Cadastrar usuário

⁴ RF05 - Resetar senha

⁵ RF02 - Enviar e-mail

Figura 10 – Exemplo de relacionamento horizontal

Afeta Depende de Origina						
	RF01	RF02	RF03	RF04	RF05	RF06
RF01		☒	☐	☐	☐	☐
RF02	☐		☐	☐	☐	☐
RF03	☐	☐		☐	☐	☐
RF04	☐	☐	☐		☐	☐
RF05	☐	☒	☐	☐		☐
RF06	☐	☐	☐	☐	☐	

4.5 Rastreamento Vertical

Através dos links que representam os relacionamentos nas Figuras 8 e 9, é possível navegar de um requisito até um artefato final que o implementa, como, por exemplo, o código.

4.6 Múltiplas Matrizes de Rastreabilidade

Com a FRR, é possível visualizar diversas matrizes de rastreabilidade. É possível combinar qualquer tipo de artefato dois a dois e visualizar seus relacionamentos, selecionar um tipo de relacionamento específico e visualizar os relacionamentos por demanda ou no projeto como um todo. Na Figura 11, são apresentados os relacionamentos entre os requisitos funcionais e os casos de uso de uma demanda específica. As cores dos relacionamentos identificam o seu tipo.

Figura 11 – Relacionamentos entre requisitos funcionais e casos de uso

Afeta Depende de Origina Satisfeito por								
	UC15	UC16	UC17	UC18	UC19	UC20	UC21	UC24
RF11	☒	☐	☐	☐	☐	☐	☐	☐
RF12	☐	☒	☐	☐	☐	☐	☐	☐
RF13	☐	☐	☒	☐	☐	☐	☐	☐
RF14	☐	☐	☐	☒	☐	☐	☐	☐
RF15	☐	☐	☐	☐	☒	☒	☐	☐
RF16	☐	☐	☐	☐	☐	☐	☒	☐
RF17	☐	☐	☐	☒	☒	☐	☐	☐
RF18	☐	☐	☐	☒	☒	☒	☐	☐

4.7 Conflito Entre Demandas

Como um artefato pode fazer parte de mais de uma demanda, a FRR identifica esses artefatos, alertando o usuário da possibilidade de conflitos. Ao acessar um artefato com possibilidade de conflito, a FRR lista as demandas da qual o artefato faz parte e os responsáveis por cada uma delas. Na Figura 12, o caso de uso UC03 está destacado com um fundo amarelo, identificando que ele pertence a mais de uma demanda. Na tela do artefato, existe uma exclamação de alerta, e ao clicar nela, o sistema exibe todas as demandas da qual o artefato faz parte.

Figura 12 – Exemplo de conflito entre demandas

Code	Name	Type	Description	Status	Priority	Attachment	Created by
RF30	Alterar senha	Requisito	A ferramenta deve alertar ao analista				
RF31	Verificar senha	Requisito	A ferramenta deve alertar ao analista				
RF32	Verificar senha	Requisito	A ferramenta deve alertar ao analista				
RF33	Verificar senha	Requisito	A ferramenta deve alertar ao analista				
UC03	Confirmar cadastro	Caso de Uso		Proposto			Administrador

Name	Description	Repo
Análise de Impactos	Módulo com funcionalidades de auxílio à análise de impactos	Admi
CRUD - Usuário	Módulo de para gestão de usuário	Admi

4.8 Controle de Versão

Sempre que um artefato é alterado, a FRR versiona esse artefato mantendo um histórico de alteração, assim o usuário pode reverter alguma alteração. A Figura 13 exibe o histórico de alteração do requisito funcional RF02. Nela é possível identificar que a versão 1 do requisito teve seu status alterada para aprovado.

Figura 13 – Histórico de alterações de um artefato

Version	Code	Name	Type	Description	Status	Priority	Attachment	Updated at	Upadated by
Live	2	RF02	Enviar e-mail	Requisito Funcional	A ferramenta deve enviar um e-mail para o usuário cadastrado com as instruções para cadastrar uma senha.	Aprovado		2018-07-04	Administrador
1	RF02	Enviar e-mail	Requisito Funcional	A ferramenta deve enviar um e-mail para o	Proposto			2018-07-03	Administrador

4.9 Acompanhamento do Status da Demanda/Projeto

A FRR permite o acompanhamento do status de uma demanda ou projeto através dos status dos artefatos. Ela exibe um gráfico com a distribuição dos status dos artefatos ao longo do tempo de desenvolvimento do projeto. Assim, quando 100% dos artefatos tiverem sido implementados e verificados, a demanda ou projeto foi finalizado.

Na Figura 14 é possível verificar que apenas 2% dos artefatos foram aprovados no projeto e 98% foram propostos e estão pendentes de aprovação.

Figura 14 – Distribuição de status dos artefatos



5 Comparativo entre Ferramentas

Nesta seção, será apresentado um comparativo entre a FRR com as ferramentas citadas no Seção 2.3 deste trabalho (OpenReq, SIGERA).

Para comparar as ferramentas, foram utilizados os seguintes critérios: tipos de requisitos e seus atributos, cadastro e edição de requisitos, controle e histórico de alterações, rastreabilidade, baseline e controle de versão, fórum de discussões, integração com outras ferramentas, segurança, acesso concorrente, acesso Web e restrições de ambiente.

Os critérios utilizados para comparação foram baseados no estudo “*Requirements for Requirements Management Tools*” (HOFFMANN et al., 2004). Este estudo tem como propósito ajudar os usuários a comparar e selecionar ferramentas de gerenciamento de requisitos, além de ajudar os fornecedores de ferramentas a direcionar futuros desenvolvimentos de ferramentas.

- a) **Tipos de requisitos e seus atributos:** No OpenReq e SIGERA, os requisitos são agrupados por tipo. O usuário pode configurar os tipos a serem utilizados em cada projeto. Na FRR, é possível cadastrar diversos tipos de requisitos e artefatos.
- b) **Cadastro e edição de requisitos:** No OpenReq e SIGERA, os requisitos são identificados pelo tipo, seguido de uma numeração sequencial por projeto. O OpenReq oferece uma interface própria para edição de requisitos. a FRR permite ao usuário determinar o identificador dos requisitos e assim utilizar um padrão determinado antes de adoção da ferramenta.
- c) **Controle e histórico de alterações:** todas as ferramentas geram um novo registro de histórico a cada alteração nos requisitos. O OpenReq utiliza números com casa decimal para representar a versão. Quando o texto do requisito é alterado, a unidade do número da versão é incrementada para o próximo número inteiro, caso contrário, apenas a parte decimal do número é incrementada. O SIGERA exige, após o cadastro de um requisito, que seja criada uma versão para incluir informações adicionais como, impacto, custo e descrição. Ele também só permite que um requisito seja alterado se não possuir uma versão registrada ou tenha status “Implementado”. A FRR permite que requisitos seja alterado em qualquer momento. Ela registra a data da alteração, quem alterou e mantém os valores novos e antigos de cada atributo.
- d) **Rastreabilidade:** O openReq não tem a funcionalidade de rastreabilidade. O SIGERA implementa a rastreabilidade, porém num único sentido e não permite

o cadastro de diferentes tipos relacionamentos. Seus relacionamentos são exibidos em uma lista no detalhamento de cada requisito. A FRR permite o cadastro de diferentes tipos de relacionamentos agregando mais informações e implementa a rastreabilidade bidirecional. Também permite diferentes tipos de visualização dos relacionamentos, numa matriz de rastreabilidade ou a partir de um artefato específico.

- e) **Baseline e controle de versões:** Nenhuma das ferramentas permite a criação de uma *baseline*. Contudo a FRR permite agrupar os requisitos relacionados a uma demanda, definindo de forma indireta uma *baseline*. O openReq possui um software de controle de versão que permite a comparação entre os textos dos requisitos. A FRR e o SIGERA permitem a comparação entre as versões dos requisitos através dos históricos de alterações.
- f) **Fórum de discussões:** A FRR e o SIGERA possuem fórum de discussão permitindo que requisitos e artefatos sejam discutidos entre as partes envolvidas. O openReq não dispõe dessa funcionalidade.
- g) **Integração com outras ferramentas:** Nenhuma das ferramentas possui integração com outras ferramentas.
- h) **Segurança:** todas as ferramentas utilizam o conceito de “perfis”, onde são definidas as ações que podem ser efetuadas pelos usuários em cada projeto. Na FRR, os usuários podem ter mais de um perfil no projeto. Existe também um perfil que permite apenas leitura.
- i) **Acesso concorrente:** Nenhuma das ferramentas tem tratamento para concorrência ou edição compartilhada dos requisitos.
- j) **Acesso Web:** Todas as ferramentas possuem acesso web, porém a FRR utiliza tecnologias atuais amplamente discutidas na Web.
- k) **Restrições de ambiente:** O OpenReq e o SIGERA foram desenvolvidos em Java, e por isso oferece suporte a todos os sistemas operacionais suportados pela linguagem. Além disso, oferecem suporte aos principais bancos de dados existentes. Porém, por serem projetos antigos que não têm sido atualizados, podem haver incompatibilidade com *browsers* modernos. A FRR foi implementado usando o framework Ruby on Rails, e tem total compatibilidade com diversos sistemas operacionais (Linux, iOS, Windows) e os browsers atuais. Podem ser utilizados diferentes banco de dados (MySQL, Postgres, Oracle, entre outros) sem necessidade alteração do código.

6 Conclusão

Desenvolver softwares de qualidade, dentro do orçamento e no tempo estimado é um dos grandes desafios de toda empresa de TI. Para que o projeto seja bem-sucedido, é necessário um bom processo de desenvolvimento e gestão de requisitos, visto que erros nos requisitos consomem maior parte do custo de reparo de um sistema. A rastreabilidade de requisitos é o coração da gestão de requisitos. Ela é importante pois permite a identificação de artefatos relacionados, o que assiste de sobremodo na atividade de análise de impactos. Existem diversas técnicas de visualização de relacionamentos entre requisitos, porém a mais utilizada é a matriz de rastreabilidade. Ela permite identificar rapidamente os relacionamentos, porém quando muito grande, dificulta a identificação dos relacionamentos.

Este trabalho teve como objetivo desenvolver uma ferramenta de apoio à gestão de requisitos, com ênfase na rastreabilidade. Além da matriz de rastreabilidade, foi implementada a possibilidade de rastrear os artefatos a partir de um artefato específico, o que facilita a análise de impacto de alteração de um artefato. Também foi implementado a funcionalidade de versionamento dos artefatos, permitindo assim analisar o histórico de um artefato e revertê-lo. Esta ferramenta pode ser de grande ajuda na gerência de uma grande quantidade de requisitos.

Este projeto proporcionou o aprofundamento no conhecimento da engenharia de software, mais especificamente, na gestão de requisitos. Foram identificados diversos conceitos sobre gestão e rastreabilidade de requisitos desconhecidos pelo autor. Este trabalho também proporcionou grande aprendizado na área de programação e o aperfeiçoamento do conhecimento do framework Rails.

6.1 Trabalhos Futuros

Como possíveis trabalhos futuros pode-se apontar:

- Integrar à FRR mais técnicas de visualização de relacionamentos para atender contextos diferentes;
- Fazer integração da FRR com um repositório de código, a fim de coletar dados para manter os artefatos atualizados;
- Fazer integração da FRR com um servidor de arquivo como o dropbox, google drive, etc.;
- Possibilitar a geração de documentos automaticamente a partir dos artefatos cadastrados;

- Permitir a geração automática de relatórios dos status dos projetos e demandas;
- Identificar automaticamente os artefatos que precisam ser analisados após uma alteração em um artefato relacionado.

Referências

- AHA! LABS INC. *Introduction to Requirements Management*. 2018. Disponível em: <<https://www.aha.io/roadmapping/guide/requirements-management>>. Acesso em: 24/06/2018.
- COVENTRY, T. Requirements management – planning for success! In: *PMI Global Congress Proceedings*. London: [s.n.], 2015. Disponível em: <<https://www.pmi.org/learning/library/requirements-management-planning-for-success-9669>>. Acesso em: 27/06/2018.
- FORD, D. N.; STERMAN, J. D. Overcoming the 90% Syndrome: Iteration Management in Concurrent Development Projects. 2003. Disponível em: <<http://dx.doi.org/10.1177/106329303038031>>. Acesso em: 27/06/2018.
- GOTEL, O. et al. Traceability Fundamentals. In: . Londres: Spring, 2012. p. 3 – 22.
- GRANDE, J. I. D.; MARTINS, L. E. G. SIGERAR: Uma Ferramenta para Gerenciamento de Requisitos. In: *Workshop em Engenharia de Requisitos*. Rio de Janeiro: [s.n.], 2006.
- GRGEC, M.; MUŽAR, R. Role of UML Sequence Diagram Constructs in Object Lifecycle Concept. *Journal of Information and Organizational Sciences*, v. 31, n. 1, 2007.
- GRINGS, C. L.; SAYÃO, M. OpenReq: uma Ferramenta para Auxílio à Gerência de Requisitos. In: *Workshop em Engenharia de Requisitos*. Valparaíso, Chile: [s.n.], 2009.
- HANIF, M. Q.; RIZVI, U. S. *Issues and Challenges of Requirement Elicitation in Large Web Projects: Practical Concepts of Requirement Engineering for Successful Web Projects Development*. [S.l.]: LAP LAMBERT Academic Publishing, 2010. 68 p.
- HARKER, S. D. P.; EASON, K. D. The Change and Evolution of Requirements as a Challenge to the Practice of Software Engineering. *Proceedings of the IEEE International Symposium on Requirements Engineering*, p. 266 – 272, 1993.
- HOFFMANN, M. et al. Requirements for Requirements Management Tools. In: *IEEE International Requirements Engineering Conference*. Quioto: IEEE, 2004. ISSN 1090-705X.
- IEEE. IEEE Standard Glossary of Software Engineering Terminology. 1990.
- LEFFINGWELL, D. Calculating Your Return on Investment from More Effective Requirements Management. *American Programmer*, v. 10, n. 4, p. 13 – 16, 1997.
- LI, Y.; MAALEJ, W. Which Traceability Visualization Is Suitable in This Context? A Comparative Study. In: REGNELL, B.; DAMIAN, D. (Ed.). *Requirements Engineering: Foundation for software quality*. Essen: Springer, Berlin, Heidelberg, 2012. (Lecture Notes in Computer Science, v. 7195), p. 194 – 210.
- NUSEIBEH, B.; EASTERBROOK, S. Requirements Engineering: A Roadmap. 2000. Disponível em: <<http://www.inf.fu-berlin.de/inst/ag-se/teaching/V-SWT-2014/ICSE2000.pdf>>. Acesso em: 24/06/2018.

- OPERATIONAL EXCELLENCE PROGRAM OFFICE. *Change Control Process*. Disponível em: <http://vcaf.berkeley.edu/sites/default/files/change_control_process_aa.pdf>. Acesso em: 01/07/2018.
- PIGOSKI, T. M. Practical software maintenance: Best practices for managing your software investment. In: _____. 1. ed. Nova York: Wiley, 1996. cap. 6. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.582.5421&rep=rep1&type=pdf>>. Acesso em: 30/06/2018.
- PMI. *Requirements Management a Core Competency for Project And Program Succes*. [S.l.], 2014. Disponível em: <<https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/requirements-management.pdf>>. Acesso em: 24/06/2018.
- POHL, K. *PRO-ART: Enabling Requirements Pre-Traceability*. 1996. Disponível em: <<https://pdfs.semanticscholar.org/cce2/2f053ddb4d38f3e8c4bdc63c7e57fcc89618.pdf>>. Acesso em: 04/06/2018.
- PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de Software: Uma Abordagem Profissional*. 8. ed. Porto Alegre: Amgh Editora, 2016. 968 p.
- ROTATIONAL SOFTWARE CORPORATION. *Artifact: Data Model*. 2001. Disponível em: <http://sce.uhcl.edu/helm/rationalunifiedprocess/process/artifact/ar_datmd.htm>. Acesso em: 22/06/2018.
- RUBY ON RAILS ORGANIZATION. *Welcome to Rails*. 2018. Disponível em: <<http://api.rubyonrails.org/>>. Acesso em: 17/06/2018.
- SEI. *CMMI for Development, Version 1.3*. Pittsburg, 2010. Disponível em: <https://resources.sei.cmu.edu/asset_files/TechnicalReport/2010_005_001_15287.pdf>. Acesso em: 24/06/2018.
- SOFTTEX. *Guia de Implementação – Parte 1: Fundamentação para Implementação do Nível G do MR-MPS-SW:2012*. [S.l.], 2013. Disponível em: <http://www.softex.br/wp-content/uploads/2013/07/MPS.BR_Guia_de_Implementacao_Parte_1_2013..pdf>. Acesso em: 24/06/2018.
- SOMMERVILLE, I. *Software engineering*. 9. ed. Massachusetts: Pearson, 2010. 792 p.
- STANDISH GROUP. *The CHAOS Report*. 1995. Disponível em: <<https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>>. Acesso em: 24/06/2018.
- TAYLOR, J. Project Scheduling and Cost Control: Planning, Monitoring and Controlling the Baseline. In: _____. [S.l.]: J. Ross Publishing, 2007. cap. 10.
- WIEGERS, K.; BEATTY, J. *Software Requiriments*. 3. ed. Redmond: Microsoft Press, 2013. (Developer Best Practices). 672 p.

Apêndices

APÊNDICE A – Requisitos Funcionais

Tabela 3 – Lista de Requisitos Funcionais

ID	Título	Descrição
RF01	Cadastrar usuário	A ferramenta deve permitir que um administrador cadastre usuários. Os tipos de usuário que podem ser cadastrados são gerentes de projeto, analistas, desenvolvedores, testadores, convidados e administradores.
RF02	Enviar e-mail	A ferramenta deve enviar um e-mail para o usuário cadastrado com as instruções para cadastrar uma senha.
RF03	Cadastrar senha	A ferramenta deve permitir que um usuário cadastrado escolha uma senha antes do primeiro acesso.
RF04	Inativar usuário	A ferramenta deve permitir que um administrador inative um usuário.
RF05	Resetar senha	A ferramenta deve permitir que o usuário possa resetar sua senha. A ferramenta deve enviar um email para cadastrar uma nova senha.

ID	Título	Descrição
RF06	Editar perfil	A ferramenta deve permitir que os usuários editem seu perfil.
RF07	Criar projeto	A ferramenta deve permitir que os gerentes possam criar projetos.
RF08	Editar projeto	A ferramenta deve permitir que o gerente do projeto possa editar um projeto.
RF09	Adicionar usuário ao projeto	A ferramenta deve permitir que usuários cadastrados possam ser adicionados em projetos. Os usuários só visualizarão os projetos nos quais foram adicionados.
RF10	Excluir projeto	A ferramenta deve permitir que o gerente do projeto possa excluir um projeto. Todas as demandas, artefatos, anexos, relacionamentos e configurações do projeto serão excluídas.
RF11	Criar tipo de artefato	A ferramenta deve permitir que o gerente possa criar, dentro de um projeto, os tipos de artefatos que serão rastreados.
RF12	Editar tipo de artefato	A ferramenta deve permitir que o gerente possa editar um tipo de artefato.
RF13	Excluir tipo de artefato	A ferramenta deve permitir que o gerente exclua um tipo de artefato. Esta ação exclua todos os artefatos do tipo excluído e os relacionamentos que chegam e saem deles, inclusive os anexos.

ID	Título	Descrição
RF14	Criar artefato	A ferramenta deve permitir que os analistas criem artefatos dos tipos existentes no projeto.
RF15	Editar artefato	A ferramenta deve permitir que os analistas editem artefatos dentro de uma demanda.
RF16	Excluir artefato	A ferramenta deve permitir que os analistas possam excluir um artefato. Todos os relacionamentos que chegam ou saem do artefato e os anexos serão excluídos.
RF17	Anexar arquivo	A ferramenta deve permitir que arquivos sejam anexados aos artefatos.
RF18	Versionar artefato	A ferramenta deve criar uma versão do artefato sempre que ele for alterado.
RF19	Criar tipo de relacionamento	A ferramenta deve permitir que o gerente crie tipos de relacionamentos dentro de um projeto.
RF20	Editar tipo de relacionamento	A ferramenta deve permitir que o gerente possa editar um tipo de relacionamento.
RF21	Excluir tipo de relacionamento	A ferramenta deve permitir que o gerente exclua um tipo de relacionamento. Esta ação exclua todos os relacionamentos do tipo excluído.

ID	Título	Descrição
RF22	Criar relacionamento	A ferramenta deve permitir que um analista relacione dois artefatos.
RF23	Editar relacionamento	A ferramenta deve permitir que os analistas editem os relacionamentos.
RF24	Excluir relacionamento	A ferramenta deve permitir que os analistas excluam um relacionamento entre dois artefatos.
RF25	Criar demanda	A ferramenta deve permitir que os gerentes e analistas de um projeto possam criar demandas no projeto.
RF26	Editar demanda	A ferramenta deve permitir que os gerentes do projeto e analista da demanda edite a demanda.
RF27	Excluir demanda	A ferramenta deve permitir que o os gerentes do projeto e analista da demanda possam excluir uma demanda. o usuário poderá manter as alterações realizadas nos artefatos e relacionamentos da demanda ou descartá-las. Ao descartá-las, os artefatos e relacionamentos criados na demanda serão excluídos e as alterações nos artefatos importados serão desfeitas.

ID	Título	Descrição
RF28	Importar artefato	A ferramenta deve permitir que o analista importe um artefato do projeto para demanda.
RF29	Remover artefato	A ferramenta deve permitir que o analista remova um artefato de uma demanda.
RF30	Alertar concorrência	A ferramenta deve alertar ao analista quando um artefato estiver sendo utilizado em outra demanda.
RF31	Visualizar relacionamentos	A ferramenta deve permitir que os gerentes, analistas, desenvolvedores, testadores e convidados visualizem os relacionamentos entre dois tipos de artefatos por demanda ou por projeto.
RF32	Verificar impactos	A ferramenta deve permitir que os gerentes, analistas, desenvolvedores e testadores verifiquem impacto de alteração de um artefato, ou seja, permitir que visualizem os relacionamentos que chegam e saem dele.

APÊNDICE B – Requisitos Não Funcionais

Tabela 4 – Lista de Requisitos Não Funcionais

ID	Descrição
RNF01	A ferramenta deve ter uma interface Web.
RNF02	A ferramenta deve possuir design responsivo em suas interfaces gráficas.
RNF03	A ferramenta deve encerrar a sessão do usuário após 30 minutos de inatividade.
RNF04	A ferramenta deve bloquear por 30 minutos os usuários que errarem a senha duas vezes.
RNF05	A ferramenta deve ser construída utilizando o framework Rails na linguagem Ruby

APÊNDICE C – Atores e Permissões

Os usuários da ferramenta são divididos em 6 atores representando os administradores (Admin) gerentes de projeto (GP), analistas de sistemas (AS), desenvolvedores (Dev), testadores (Tes) e convidados (Con). Na tabela abaixo, são listadas as permissões de cada ator:

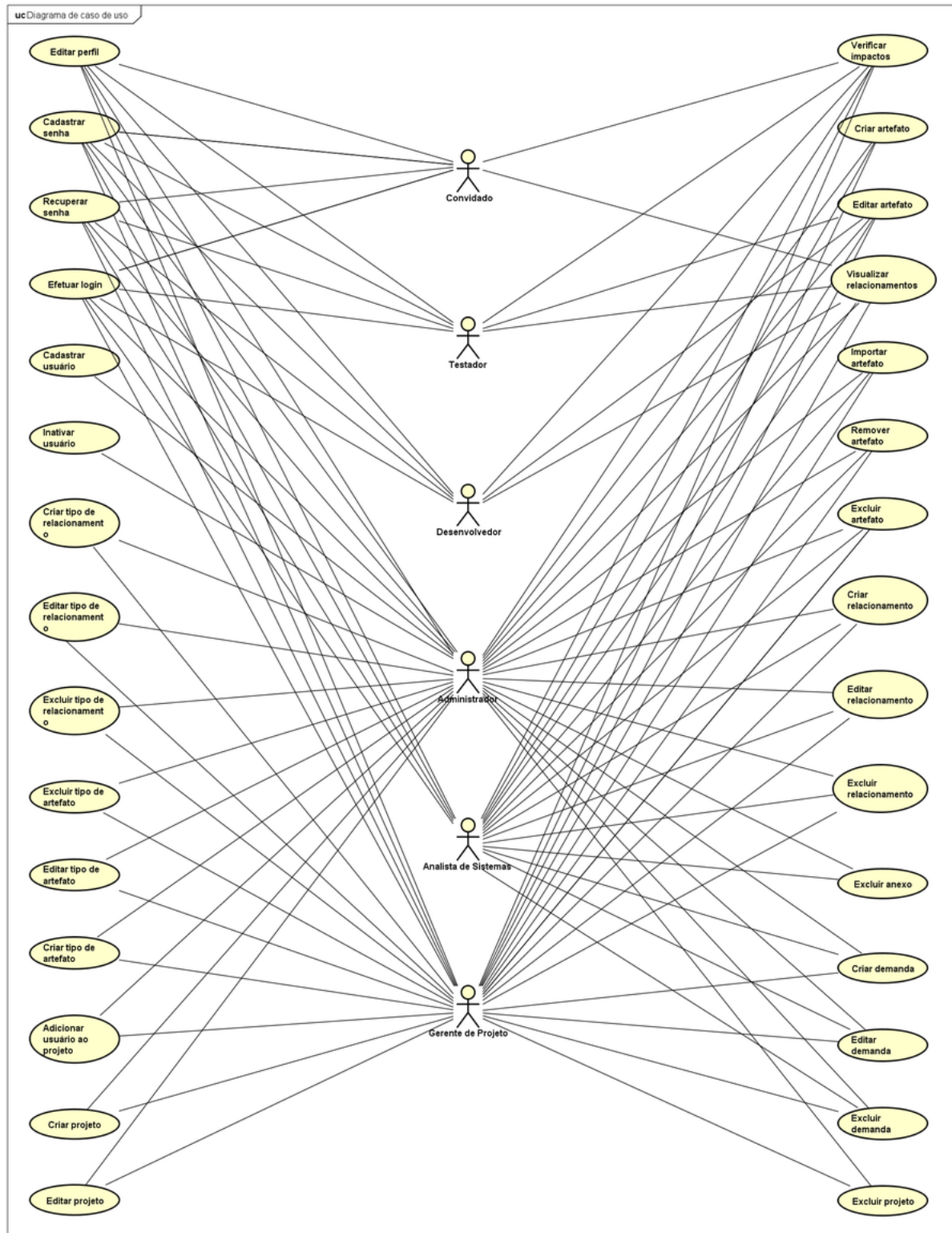
Tabela 5 – Permissões dos Atores

Requisito	Admin	GP	AS	Dev	Tes	Con
Cadastrar usuário	X					
Cadastrar senha	X	X	X	X	X	X
Inativar usuário	X					
Recuperar senha	X	X	X	X	X	X
Editar perfil	X	X	X	X	X	X
Criar projeto	X	X				
Editar projeto	X	X				
Adicionar usuário ao projeto	X	X				
Excluir projeto	X	X				
Criar demanda	X	X	X			
Editar demanda	X	X	X			
Excluir demanda	X	X	X			
Criar tipo de artefato	X	X				
Editar tipo de artefato	X	X				

Requisito	Admin	GP	AS	Dev	Tes	Con
Excluir tipo de artefato	X	X				
Criar artefato	X	X	X			
Editar artefato	X	X	X	X	X	
Excluir artefato	X	X	X			
Anexar arquivo	X	X	X	X	X	
Excluir anexo	X	X	X	X	X	
Importar artefato	X	X	X			
Remover artefato	X	X	X			
Criar tipo de relacionamento	X	X				
Editar tipo de relacionamento	X	X				
Excluir tipo de relacionamento	X	X				
Criar relacionamento	X	X	X			
Editar relacionamento	X	X	X			
Excluir relacionamento	X	X	X			
Visualizar artefatos	X	X	X	X	X	X
Visualizar relacionamentos	X	X	X	X	X	X
Verificar impactos	X	X	X	X	X	X

APÊNDICE D – Diagrama de Casos de Uso

Figura 15 – Diagrama de casos de uso



APÊNDICE E – Descrição dos Casos de Uso

E.1 UC - Cadastrar Usuário

Tabela 6 – UC - Cadastrar Usuário

ITEM	VALUE
UseCase	Cadastrar usuário
Summary	-
Actor	Administrador
Precondition	Estar logado
Postcondition	Cadastro do usuário pendente de confirmação
Base Sequence	1) O ator aciona a opção para cadastrar um novo usuário; 2) O sistema exibe a tela de cadastro de usuário; 3) O ator informa os dados do usuário (e-mail e papéis); 4) O sistema valida os dados; 5) O sistema envia um email para o usuário com instruções para confirmar o cadastro.
Branch Sequence	-
Exception Sequence	• O sistema informa que o email é inválido. • O sistema informa que o email já foi cadastrado. • O sistema informa que os campos obrigatórios (email e papéis) não foram preenchidos.
Sub UseCase	-
Note	-

E.2 UC - Inativar Usuário**Tabela 7 – UC - Inativar Usuário**

ITEM	VALUE
UseCase	Inativar usuário
Summary	-
Actor	Administrador
Precondition	Estar logado
Postcondition	Usuário inativado perde o acesso ao sistema
Base Sequence	1) Ator aciona a opção para visualizar a lista de usuários; 2) O sistema exibe a lista de usuários cadastrados; 3) O ator aciona a opção para inativa um usuário; 4) O sistema solicita confirmação da operação; 5) O ator confirma a operação; 6) O sistema inativa o usuário.
Branch Sequence	-
Exception Sequence	-
Sub UseCase	-
Note	-

E.3 UC - Confirmar Cadastro

Tabela 8 – UC - Confirmar Cadastro

ITEM	VALUE
UseCase	Confirmar cadastro
Summary	-
Actor	Administrador, Gerente de Projeto, Analista de Sistemas, Desenvolvedor, Testador e Convidado
Precondition	Tempo para confirmação não expirado
Postcondition	Ator pode acessar o sistema
Base Sequence	1) O ator acessa o link enviado por email para confirmação; 2) O sistema exibe um formulário; 3) O ator informa o nome do usuário, uma senha e a confirmação da senha; 4) O sistema valida os dados informados; 5) O sistema o sistema confirma o cadastro do ator; 6) O sistema loga o ator.
Branch Sequence	-
Exception Sequence	• O sistema informa que o token de confirmação é inválido. • O sistema informa que o toke de confirmação expirou. • O sistema informa que o email já foi confirmado. • O sistema informa que os campos obrigatórios não foram preenchidos (nome do usuário, senha e confirmação de senha). • O sistema informa que a senha não atende os critérios.
Sub UseCase	-
Note	O token expira em 3 dias e a senha deve ter no mínimo 8 caracteres.

E.4 UC - Reenviar Email de Confirmação

Tabela 9 – UC - Reenviar Email de Confirmação

ITEM	VALUE
UseCase	Reenviar email de confirmação
Summary	-
Actor	Administrador, Gerente de Projeto, Analista de Sistemas, Desenvolvedor, Testador e Convidado
Precondition	Cadastro não confirmado
Postcondition	Gerado novo token de confirmação
Base Sequence	1) O ator solicita o reenvio das instruções de confirmação; 2) O sistema exibe um formulário; 3) O ator informa o email cadastrado; 4) O sistema valida o email; 5) O sistema reenvia as instruções de confirmação por email.
Branch Sequence	-
Exception Sequence	• O sistema informa que o email não foi encontrado. • O sistema informa que o email já foi confirmado. • O sistema informa que o email não foi preenchido.
Sub UseCase	-
Note	-

E.5 UC - Resetar Senha

Tabela 10 – UC - Resetar Senha

ITEM	VALUE
------	-------

ITEM	VALUE
UseCase	Resetar Senha
Actor	Administrador, Gerente de Projeto, Analista de Sistemas, Desenvolvedor, Testador e Convidado
Precondition	Estar cadastrado no sistema
Postcondition	Senha alterada
Base Sequence	1) O ator aciona a opção para acessar o sistema; 2) O sistema exibe a tela de login; 3) O ator aciona a opção para recuperar a senha; 4) O sistema exibe a tela de recuperação de senha; 5) O ator informa o e-mail cadastrado; 6) O sistema valida os dados; [E1, E2, E3] 7) O sistema envia um email ao ator com informações para recuperar senha. 8) O ator acessar o sistema através do link enviado por email; 9) O sistema exibe formulário para cadastro de nova senha; 10) O ator informa os dados (senha e confirmação de senha); 11) O sistema valida os dados; [E4, E5] 12) O sistema altera senha do ator.

ITEM	VALUE
Exception Sequence	E1 - O sistema informa que o email não foi encontrado. E2 - O sistema informa que o email já foi confirmado. E3 - O sistema informa que o email não foi preenchido. E4 - Campos obrigatórios não preenchidos (senha e confirmação de senha) E5 - Confirmação de senha e senha não são iguais

E.6 UC - Efetuar Login

Tabela 11 – UC - Efetuar Login

ITEM	VALUE
UseCase	Efetuar login
Summary	-
Actor	Todos
Precondition	Estar cadastrado no sistema
Postcondition	Ator acessa o sistema
Base Sequence	1) O ator aciona a opção para fazer login; 2) O sistema exibe a tela de login; 3) O ator informa os dados (e-mail, senha); 4) O sistema valida os dados; [A1] 5) O sistema exibe a tela inicial do sistema.
Branch Sequence	A1 - Password ou senha inválidos e uma ou mais tentativas restante 1) O sistema informa que o e-mail ou senha são inválidos; 2) O caso de uso retorna ao passo 3 do fluxo base.
Exception Sequence	• O sistema informar que a conta do ator foi bloqueada. • O sistema informa que a conta do ator foi deletada (inativada).
Sub UseCase	-
Note	A conta é bloqueada na terceira tentativa mal sucedida por uma hora.

E.7 UC - Editar Perfil**Tabela 12 – UC - Editar Perfil**

ITEM	VALUE
UseCase	Editar perfil
Summary	-
Actor	Todos
Precondition	Estar logado
Postcondition	Estar logado
Base Sequence	1) O ator aciona opção para visualizar seu perfil; 2) O sistema exibe o perfil do ator; 3) O ator aciona opção para editar seu perfil; 4) O sistema exibe o formulário de edição; 5) O ator edita seu perfil (nome, imagem, senha, confirmação de senha e senha atual); 6) O sistema valida os dados; 7) O sistema atualiza o perfil do ator.
Branch Sequence	-
Exception Sequence	• O sistema informa que para alterar a senha é necessário repeti-la na confirmação. • O sistema informa que os dados obrigatórios (nome, senha atual) não foram preenchidos. • O sistema informa que o formato da imagem carregado é inválido.
Sub UseCase	-
Note	Os formatos de imagem válidos são png, jpg e jpeg.

E.8 UC - Criar Projeto**Tabela 13 – UC - Criar Projeto**

	VALUE
UseCase	Criar projeto
Summary	-
Actor	Administrador e Gerente de Projeto
Precondition	Estar logado
Postcondition	Projeto cadastrado com o ator adicionado ao projeto.
Base Sequence	1) O ator aciona a opção para cadastrar um projeto; 2) O sistema exibe a tela de cadastro de projeto; 3) O ator informa os dados do projeto (nome, descrição); 4) O sistema valida os dados; 5) O sistema cadastra e adiciona o ator ao projeto; 6) O sistema exibe a tela do projeto.
Branch Sequence	-
Exception Sequence	• O sistema informa que o nome já está em uso. • O sistema informa que o campo obrigatório (nome) não foi preenchido.
Sub UseCase	-
Note	-

E.9 UC - Editar Projeto**Tabela 14 – UC - Editar Projeto**

	VALUE
UseCase	Editar projeto
Summary	-
Actor	Administrador e Gerente de Projeto
Precondition	Estar logado
Postcondition	Projeto atualizado.
Base Sequence	1) O ator seleciona um projeto; 2) O sistema exibe a tela do projeto; 3) O ator aciona a opção para editar o projeto; 4) O sistema exibe a tela de edição de projeto; 5) O ator altera os dados do projeto (nome, descrição); 6) O sistema valida os dados; 7) O sistema atualiza o projeto.
Branch Sequence	-
Exception Sequence	• O sistema informa que o nome já está em uso. • O sistema informa que o campo obrigatório (nome) não foi preenchido.
Sub UseCase	-
Note	-

E.10 UC - Excluir Projeto**Tabela 15 – UC - Excluir Projeto**

	VALUE
UseCase	Excluir projeto
Summary	-
Actor	Administrador e Gerente de Projeto
Precondition	Gerente de Projeto adicionado no projeto
Postcondition	Projeto excluído
Base Sequence	1) O ator seleciona a opção “Projetos” 2) O sistema exibe a lista de projetos que ator tem acesso; 3) O ator seleciona o projeto que deseja excluir; 4) O sistema solicita confirmação da operação; 5) O ator confirma a operação; 6) O sistema exclui o projeto e suas dependências.
Branch Sequence	-
Exception Sequence	-
Sub UseCase	-
Note	-

E.11 UC - Adicionar Usuário ao Projeto**Tabela 16 – UC - Adicionar Usuário ao Projeto**

ITEM	VALUE
UseCase	Adicionar usuário ao projeto
Summary	-
Actor	Administrador e Gerente de Projeto
Precondition	Estar logado e existir usuário não adicionado ao projeto selecionado
Postcondition	Lista de usuários integrantes do projeto atualizada
Base Sequence	1) O ator seleciona um projeto; 2) O sistema exibe a tela do projeto; 3) O ator aciona a opção para adicionar um usuário ao projeto; 4) O sistema busca a lista de usuários cadastrados que não foram adicionados ao projeto e a exibe; 5) O ator escolhe um usuário; 6) O sistema adiciona o usuário ao projeto.
Branch Sequence	-
Exception Sequence	-
Sub UseCase	-
Note	-

E.12 UC - Criar Demanda**Tabela 17 – UC - Criar Demanda**

ITEM	VALUE
UseCase	Criar demanda
Summary	-
Actor	Administrador, Gerente de Projeto e Analista de Sistema
Precondition	Estar logado e estar em um projeto
Postcondition	Demanda criada
Base Sequence	1) O ator seleciona um projeto; 2) O sistema exibe a tela do projeto; 3) O ator aciona a opção para criar uma demanda; 4) O sistema exibe a tela de criação de demanda; 5) O ator informa os dados da demanda (nome, descrição, release, responsável); 6) O sistema valida os dados; 7) O sistema cria a demanda e adiciona o ator a demanda.
Branch Sequence	-
Exception Sequence	• O sistema informa que o nome já está em uso. • O sistema informa que os dados obrigatórios (nome, responsável) não foram preenchidos.
Sub UseCase	-
Note	-

E.13 Editar Demanda**Tabela 18 – UC - Editar Demanda**

ITEM	VALUE
UseCase	Editar demanda
Summary	-
Actor	Administrador, Gerente de Projeto, Analista de Sistemas
Precondition	Estar logado
Postcondition	Demanda atualizada
Base Sequence	1) O ator seleciona um projeto; 2) O sistema exibe a tela do projeto; 3) O ator seleciona uma demanda; 4) O sistema exibe a tela da demanda; 5) O ator aciona a opção para editar a demanda; 6) O sistema exibe a tela de edição da demanda; 7) O ator altera os dados da demanda (nome, descrição, release, responsável); 8) O sistema valida os dados; 9) O sistema atualiza a demanda.
Branch Sequence	-
Exception Sequence	• O sistema verifica que o nome já está em uso. • O sistema verifica que os dados obrigatórios (nome, responsável) não foram preenchidos.
Sub UseCase	-
Note	-

E.14 UC - Excluir Demanda**Tabela 19 – UC - Excluir Demanda**

ITEM	VALUE
UseCase	Excluir demanda
Actor	Administrador, Gerente de Projeto, Analista de Sistemas
Precondition	Selecionada uma demanda
Postcondition	Demanda e artefatos
Base Sequence	1) O ator aciona a opção para excluir a demanda; 2) O sistema solicita confirmação da operação; 3) O ator confirma a operação; 4) O sistema exclui todos os artefatos criados na demanda e remove os que foram importados; [A1] 5) O sistema exclui a demanda.
Branch Sequence	A1 - Demanda com artefatos conflitantes 1) O sistema informa que existem artefatos conflitantes na demanda e exibe uma lista com esses artefatos; 2) O ator informa os artefatos que deseja reverter selecionando um versão anterior e aciona a opção para excluir a demanda; 3) O sistema solicita confirmação; 4) O ator confirma a operação; 5) O sistema reverte todos os artefatos conflitantes que tiveram uma versão selecionada e os remove da demanda; 6) O sistema remove os artefatos conflitantes que não tiveram uma versão selecionada; 7) O sistema exclui os artefatos criados na demanda sem conflitos e remove os que foram importados também sem conflito; 8) O sistema exclui a demanda.

E.15 UC - Criar Tipo de Artefato**Tabela 20 – UC - Criar Tipo de Artefato**

ITEM	VALUE
UseCase	Criar tipo de artefato
Summary	-
Actor	Administrador, Gerente de Projeto
Precondition	Estar logado
Postcondition	Tipo de artefato disponível para cadastro de um artefato
Base Sequence	1) O ator seleciona um projeto; 2) O sistema exibe a tela do projeto; 3) O ator aciona a opção para cadastrar um tipo de artefato; 4) O sistema exibe a tela de cadastro de tipo de artefato; 5) O ator informa os dados do tipo de artefato (nome); 6) O sistema valida os dados; 7) O sistema cadastra o tipo de artefato.
Branch Sequence	-
Exception Sequence	• O sistema informa que o nome já está em uso. • O sistema informa que os dados obrigatórios (nome) não foram preenchidos.
Sub UseCase	-
Note	-

E.16 UC - Editar Tipo de Artefato**Tabela 21 – UC - Editar Tipo de Artefato**

ITEM	VALUE
UseCase	Editar tipo de artefato
Summary	-
Actor	Administrador, Gerente de Projeto
Precondition	Estar logado
Postcondition	Tipo de artefato atualizado
Base Sequence	1) O ator seleciona um projeto; 2) O sistema exibe a tela do projeto; 3) O ator aciona opção para visualizar os tipos de artefatos do projeto; 4) O sistema exibe a lista de tipos de artefatos do projeto; 5) O ator seleciona um tipo de artefato; 6) O sistema exibe os dados do tipo de artefato; 7) O ator aciona a opção para editar um tipo de artefato; 8) O sistema exibe a tela de edição do tipo de artefato; 9) O ator altera os dados do tipo de artefato (nome); 10) O sistema valida os dados; 11) O sistema atualiza o tipo de artefato.
Branch Sequence	-
Exception Sequence	• O sistema informa que o nome já está em uso. • O sistema informa que os dados obrigatórios (nome) não foram preenchidos.
Sub UseCase	-
Note	-

E.17 UC - Excluir Tipo de Artefa**Tabela 22 – UC - Excluir Tipo de Artefa**

ITEM	VALUE
UseCase	Excluir tipo de artefato
Summary	-
Actor	Administrador, Gerente de Projeto
Precondition	Projeto selecionados
Postcondition	Tipo de artefato e os artefatos do mesmo tipo excluídos
Base Sequence	1) O ator aciona a opção visualizar os tipos de artefatos; 2) O sistema exibe a tela com a lista de tipos de artefatos cadastrados; 3) O ator aciona a opção para excluir um tipo de artefato; 4) O sistema alerta que todos os artefatos criados com o tipo serão excluídos; 5) O ator confirma; 6) O sistema exclui todos os artefatos do tipo e o tipo de artefato.
Branch Sequence	-
Exception Sequence	-
Sub UseCase	-
Note	-

E.18 UC - Criar Artefato**Tabela 23 – UC - Criar Artefato**

ITEM	VALUE
------	-------

ITEM	VALUE
UseCase	Criar artefato
Summary	-
Actor	Administrador, Gestor de Projeto, Analista de Sistemas
Precondition	Estar logado e pertencer a um projeto
Postcondition	Artefato criado na demanda pode ser visualizado no projeto
Base Sequence	1) O ator seleciona um projeto; 2) O sistema exibe a tela do projeto; 3) O ator aciona opção para ver a lista de artefatos; 4) O sistema exibe a lista de artefatos do projeto; [A1] 5) O ator aciona a opção para criar um artefato; 6) O sistema exibe a tela de criação de artefato; 7) O ator informa os dados do artefato (código, nome, tipo, descrição, status, prioridade); 8) O sistema valida os dados; 9) O sistema cria o artefato.

ITEM	VALUE
Branch Sequence	A1 - Criar artefato em uma demanda 1) O ator seleciona uma demanda do projeto; 2) O sistema exibe a tela da demanda; 3) O ator aciona a opção para criar um artefato; 4) O sistema exibe a tela de criação de artefato; 5) O ator informa os dados do artefato (código, nome, descrição, status, prioridade) 6) O sistema valida os dados; 7) O sistema cria o artefato.
Exception Sequence	• O sistema verifica que o código já está em uso. • O sistema verifica que o nome já está em uso. • O sistema verifica que os campos obrigatórios (código, nome, status e prioridade) não foram preenchidos.
Sub UseCase	UC0
Note	Os status possíveis são: proposto, aprovado, implementado, verificado, deletado, rejeitado.

E.19 UC - Editar Artefato

Tabela 24 – UC - Editar Artefato

ITEM	VALUE
UseCase	Editar artefato
Actor	Administrador, Gestor de Projeto, Analista de Sistemas, Desenvolvedor e Testador

ITEM	VALUE
Precondition	Estar logado e pertencer a um projeto
Postcondition	Artefato atualizado
Base Sequence	1) O ator seleciona um projeto; 2) O sistema exibe a tela do projeto; 3) O ator aciona opção para ver a lista de artefatos; 4. O sistema exibe a lista de artefatos do projeto; [A1] 4) O ator seleciona um artefato; 5) O sistema exibe os dados do artefato 6) O ator aciona a opção para editar um artefato; 7) O sistema exibe a tela de edição de artefato; 8) O ator altera os dados do artefato (código, nome, tipo, descrição, status, prioridade); 9) O sistema valida os dados; 10) O sistema atualiza o artefato.

ITEM	VALUE
Branch Sequence	A1 - Editar artefato em uma demanda 1) O ator seleciona uma demanda do projeto; 2) O sistema exibe a tela da demanda; 3) O ator aciona a opção para criar um artefato; 4) O sistema exibe a tela de criação de artefato; 5) O ator informa os dados do artefato (código, nome, descrição, status, prioridade) 6) O sistema valida os dados; 7) O sistema cria o artefato.
Exception Sequence	• O sistema informa que o código já está em uso. • O sistema informa que o nome já está em uso. • O sistema informa que os campos obrigatórios (código, nome, status e prioridade) não foram preenchidos.

E.20 UC - Excluir Arquivo**Tabela 25 – UC - Excluir Arquivo**

ITEM	VALUE
UseCase	Excluir arquivo
Summary	-
Actor	Administrador, Gerente de Projeto, Analista de Sistemas, Desenvolvedor, Testador
Precondition	Artefato selecionado
Postcondition	Artefato versionado
Base Sequence	1) O ator aciona a opção para excluir o anexo do artefato; 2) O sistema solicita confirmação da operação; 3) O ator confirma a operação; 4) O sistema salva a versão atual do artefato e cria uma nova sem anexo;
Branch Sequence	-
Exception Sequence	-
Sub UseCase	-
Note	-

E.21 UC - Excluir Artefato**Tabela 26 – UC - Excluir Artefato**

ITEM	VALUE
UseCase	Excluir artefato
Summary	-
Actor	Administrador, Gerente de Projeto, Analista de Sistemas
Precondition	Selecionado um projeto ou uma demanda
Postcondition	Artefato, suas versões, anexos e relacionamentos excluídos
Base Sequence	1) O ator seleciona a opção “Artefatos”; 2) O sistema exibe os artefatos do projeto; 3) O ator seleciona um ou mais artefatos que deseja excluir; 4) O sistema solicita confirmação da operação; 5) O ator confirma a operação; 6) O sistema exclui os artefatos e suas dependências.
Branch Sequence	-
Exception Sequence	-
Sub UseCase	-
Note	-

E.22 UC - Importar Artefato**Tabela 27 – UC - Importar Artefato**

ITEM	VALUE
UseCase	Importar artefato
Summary	-
Actor	Administrador, Gerente de Projeto, Analista de Sistemas
Precondition	Selecionada uma demanda
Postcondition	Artefato importado em uma demanda
Base Sequence	1) O ator aciona a opção para importar um artefato do projeto para demanda; 2) O sistema exibe a tela de importação com a lista dos artefatos do projeto que não estão na demanda; 3) O ator seleciona um artefato para importação; 4) O sistema importa o artefato para demanda.
Branch Sequence	-
Exception Sequence	-
Sub UseCase	-
Note	-

E.23 UC - Remover Artefato**Tabela 28 – UC - Remover Artefato**

ITEM	VALUE
UseCase	Remover artefato
Summary	-
Actor	Administrador, Gerente de Projeto, Analista de Sistemas
Precondition	Selecionada uma demanda
Postcondition	Artefato removido da demanda
Base Sequence	1) O ator aciona a opção visualizar os artefatos da demanda; 2) O sistema exibe a tela com a lista dos artefatos da demanda; 3) O ator seleciona um artefato para remoção; 4) O sistema remove o artefato da demanda.
Branch Sequence	-
Exception Sequence	-
Sub UseCase	-
Note	-

E.24 UC - Reverter Artefato**Tabela 29 – UC - Reverter Artefato**

ITEM	VALUE
UseCase	Reverter artefato
Summary	-
Actor	Administrador, Gerente de Projeto, Analista de Sistemas
Precondition	Selecionado um projeto ou uma demanda
Postcondition	Artefato revertido
Base Sequence	1) O ator aciona a opção visualizar os artefatos; 2) O sistema exibe a tela com a lista dos artefatos; 3) O ator seleciona um artefato; 4) O sistema exibe a tela do artefato; 5) O ator aciona a opção para reverter o artefato; 6) O sistema exibe todas as versões do artefato; 7) O ator seleciona uma versão anterior e aciona a opção para reverter o artefato; 8) O sistema solicita confirmação da operação; 9) O ator confirma a operação; 10) O sistema reverte o artefato para versão selecionada e deleta as versões posteriores. [E1]
Branch Sequence	-
Exception Sequence	E1 - O sistema informa que nenhuma versão foi selecionada
Sub UseCase	-
Note	-

E.25 UC - Criar Tipo de Relacionamento**Tabela 30 – UC - Criar Tipo de Relacionamento**

ITEM	VALUE
UseCase	Criar tipo de relacionamento
Summary	-
Actor	Administrador
Precondition	Estar logado
Postcondition	Tipo de relacionamento disponível para ligar os artefatos
Base Sequence	1) O ator seleciona um projeto; 2) O sistema exibe a tela do projeto; 3) O ator aciona a opção para cadastrar um tipo de relacionamento; 4) O sistema exibe a tela de cadastro de um tipo de relacionamento; 5) O ator informa os dados do tipo de relacionamento (nome, cor); 6) O sistema valida os dados; 7) O sistema cadastra o tipo de relacionamento.
Branch Sequence	-
Exception Sequence	• O sistema informa que o nome já está em uso. • O sistema informa que a cor já está em uso. • O sistema informa que os campos obrigatórios (nome, cor) não foram preenchidos.
Sub UseCase	-
Note	-

E.26 UC - Editar Tipo de Relacionamento

Tabela 31 – UC - Editar Tipo de Artefato

ITEM	VALUE
UseCase	Editar tipo de relacionamento
Summary	-
Actor	Administrador, Gerente de Projeto
Precondition	Estar logado
Postcondition	Tipo de relacionamento atualizado
Base Sequence	1) O ator seleciona um projeto; 2) O sistema exibe a tela do projeto; 3) O ator aciona opção para visualizar os tipos de relacionamentos do projeto; 4) O sistema exibe a lista de tipos de relacionamentos do projeto; 5) O ator seleciona um tipo de relacionamento; 6) O sistema exibe os dados do tipo de relacionamento; 7) O ator aciona a opção para editar um tipo de relacionamento; 8) O sistema exibe a tela de edição do tipo de relacionamento; 9) O ator altera os dados do tipo de relacionamento (nome, cor); 10) O sistema valida os dados; 11) O sistema atualiza o tipo de relacionamento.

ITEM	VALUE
Branch Sequence	-
Exception Sequence	• O sistema informa que o nome já está em uso. • O sistema informa que os dados obrigatórios (nome, cor) não foram preenchidos.
Sub UseCase	-
Note	-

E.27 UC - Excluir Tipo de Relacionamento**Tabela 32 – UC - Excluir Tipo de Relacionamento**

ITEM	VALUE
UseCase	Excluir tipo de relacionamento
Summary	-
Actor	Administrador, Gerente de Projeto
Precondition	Projeto selecionados
Postcondition	Tipo de relacionamento e os relacionamentos do mesmo tipo excluídos
Base Sequence	1) O ator aciona a opção visualizar os tipos de relacionamentos; 2) O sistema exibe a tela com a lista de tipos de relacionamentos cadastrados; 3) O ator aciona a opção para excluir um tipo de relacionamento; 4) O sistema alerta que todos os relacionamentos criados com o tipo serão excluídos; 5) O ator confirma; 6) O sistema exclui todos os relacionamentos do tipo e o tipo de relacionamento.
Branch Sequence	-
Exception Sequence	-
Sub UseCase	-
Note	-

E.28 UC - Criar Relacionamento**Tabela 33 – UC - Criar Relacionamento**

ITEM	VALUE
------	-------

ITEM	VALUE
UseCase	Criar relacionamento
Summary	-
Actor	Administrador, Gerente de Projeto e Analista de Sistemas
Precondition	Estar logado, existir tipo de relacionamento e ao menos dois artefatos no projeto
Postcondition	Artefatos relacionados
Base Sequence	1) O ator seleciona um projeto; 2) O sistema exibe a tela do projeto; 3) O ator aciona a opção visualizar os relacionamentos; [A1] 4) O sistema exibe a matriz de rastreabilidade com todos os artefatos do projeto; 5) O ator informa os dados do relacionamento (artefato de origem, artefato de destino, tipo, descrição); 6) O sistema valida os dados; 7) O sistema cria o relacionamento.

ITEM	VALUE
Branch Sequence	A1 - Criar relacionamento dentro de uma demanda 1) O ator seleciona uma demanda; 2) O sistema exibe a lista de artefatos da demanda; 3) O ator aciona a opção de visualizar os relacionamentos; 4) O sistema exibe a matriz de rastreabilidade com todos os artefatos da demanda; 5) O ator informa os dados do relacionamento (artefato de origem, artefato de destino, tipo, descrição); 6) O sistema valida os dados; 7) O sistema cria o relacionamento.
Exception Sequence	• O sistema informa que o artefato de origem já está relacionado ao artefato de destino. • O sistema informa que os campos obrigatórios (artefato de origem, artefato de destino, tipo) não foram preenchidos.
Sub UseCase	-
Note	Relacionamentos criados dentro de uma também ficam visíveis em todo o projeto.

E.29 UC - Editar Relacionamento

Tabela 34 – UC - Editar Relacionamento

ITEM	VALUE
UseCase	Editar relacionamento
Summary	-
Actor	Administrador, Gerente de Projeto e Analista de Sistemas

ITEM	VALUE
Precondition	Estar logado
Postcondition	Relacionamento atualizado
Base Sequence	1) O ator seleciona um projeto; 2) O sistema exibe a lista de demandas do projeto; 3) O ator aciona a opção para visualizar os relacionamentos; [A1] 4) O sistema exibe a matriz de rastreabilidade com todos os artefatos do projeto; 5) O ator seleciona um relacionamento; 6) O sistema exibe a tela de edição do relacionamento; 7) O ator informa os dados do relacionamento (artefato de origem, artefato de destino, tipo, descrição); 8) O sistema valida os dados; 9) O sistema atualiza o relacionamento.

ITEM	VALUE
Branch Sequence	A1 - Editar relacionamento dentro de uma demanda 1) O ator seleciona uma demanda; 2) O sistema exibe a lista de artefatos da demanda; 3) O ator aciona a opção de visualizar os relacionamentos; 4) O sistema exibe a matriz de rastreabilidade com todos os artefatos da demanda; 5) O ator seleciona um relacionamento; 6) O sistema exibe a tela de edição do relacionamento; 7) O ator informa os dados do relacionamento (artefato de origem, artefato de destino, tipo, descrição); 8) O sistema valida os dados; 9) O sistema atualiza o relacionamento.
Exception Sequence	• O sistema informa que o artefato de origem já está relacionado ao artefato de destino. • O sistema informa que os campos obrigatórios (artefato de origem, artefato de destino, tipo) não foram preenchidos.
Sub UseCase	-
Note	-

E.30 Excluir Relacionamento

Tabela 35 – UC - Excluir Relacionamento

ITEM	VALUE
UseCase	Excluir relacionamento
Summary	-

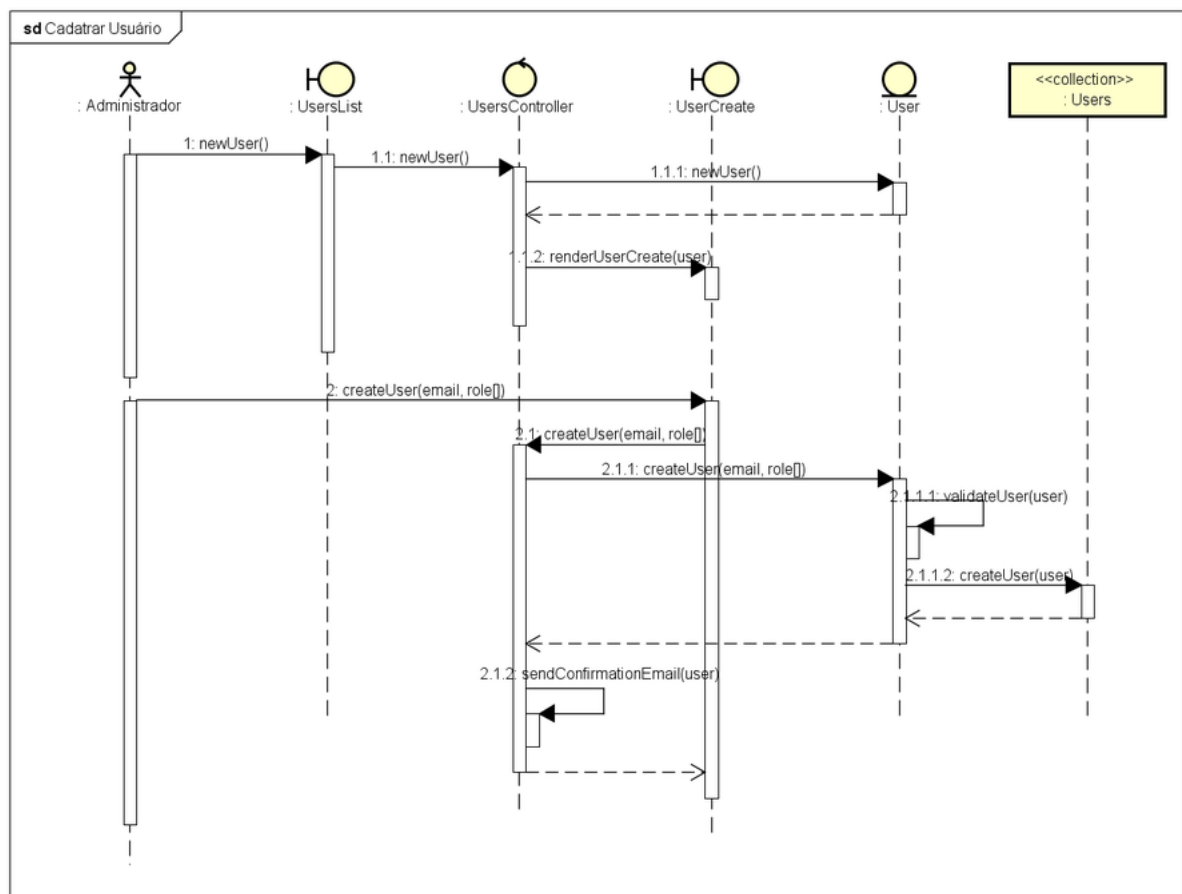
ITEM	VALUE
Actor	Administrador, Gerente de Projeto e Analista de Sistemas
Precondition	Projeto ou demanda selecionados
Postcondition	Relacionamento atualizado
Base Sequence	1) O ator aciona a opção para visualizar os relacionamentos; 2) O sistema exibe a matriz de rastreabilidade com todos os artefatos do projeto; 3) O ator seleciona o relacionamento que deseja excluir; 4) O sistema exclui relacionamento.
Branch Sequence	-
Exception Sequence	-
Sub UseCase	-
Note	-

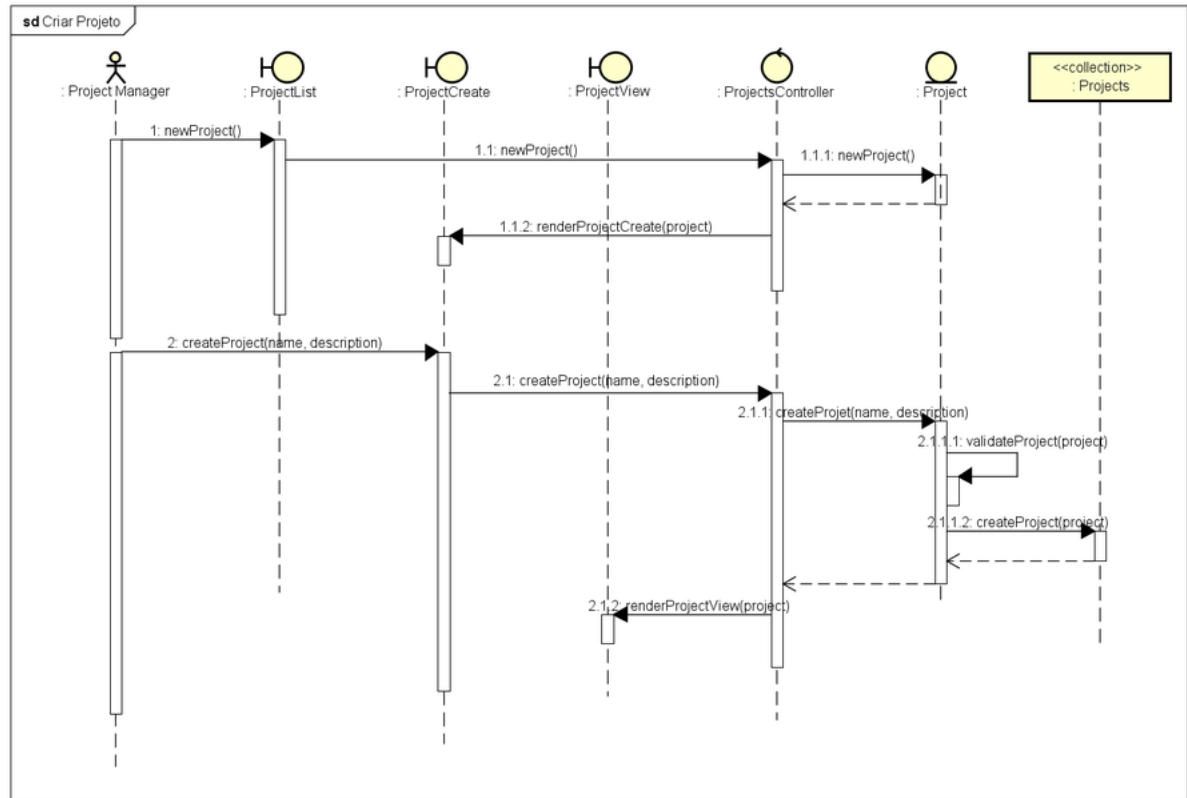
APÊNDICE F – Diagramas de Sequência

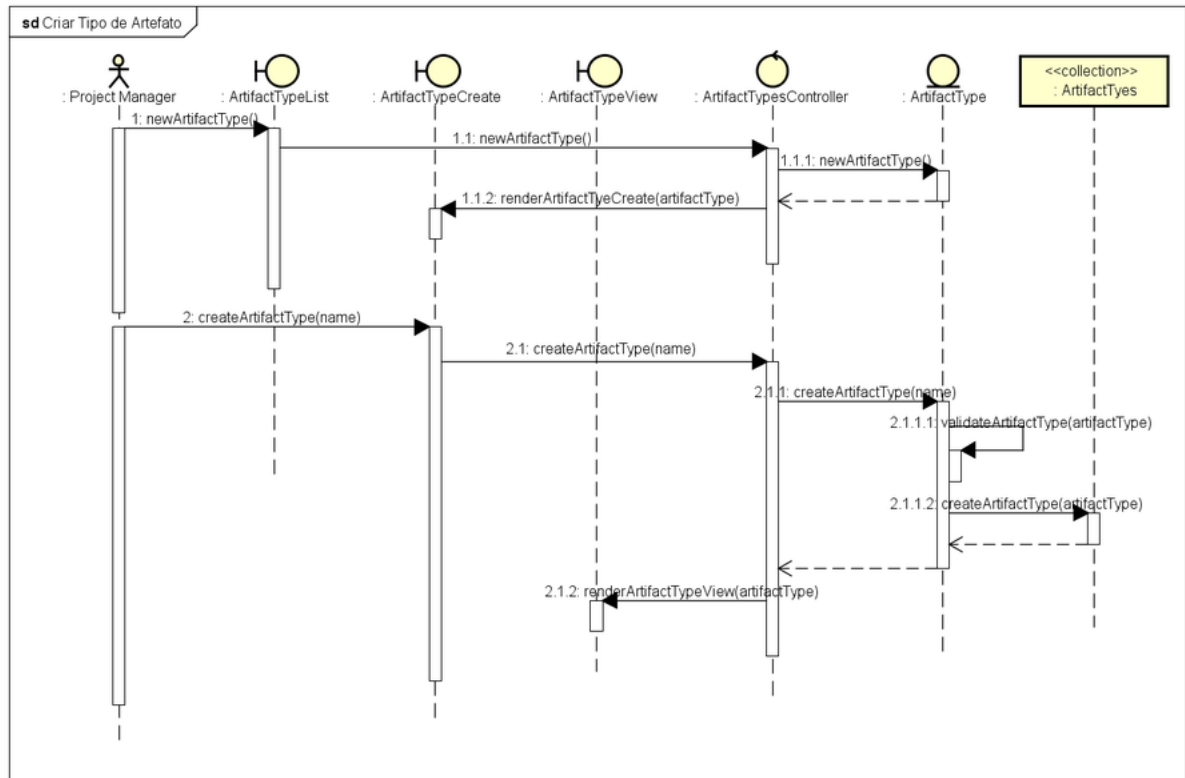
Os diagramas de seqüência UML são usados para representar ou modelar o fluxo de mensagens, eventos e ações entre objetos ou componentes de um sistema. Os diagramas de seqüência são usados principalmente para projetar, documentar e validar a arquitetura, as interfaces e a lógica do sistema, descrevendo a seqüência de ações que precisam ser executadas para concluir uma tarefa ou um cenário. São ferramentas de design úteis porque fornecem uma visão dinâmica do comportamento do sistema, que pode ser difícil de extrair dos diagramas estáticos de especificações (GRGEC; MUŽAR, 2007).

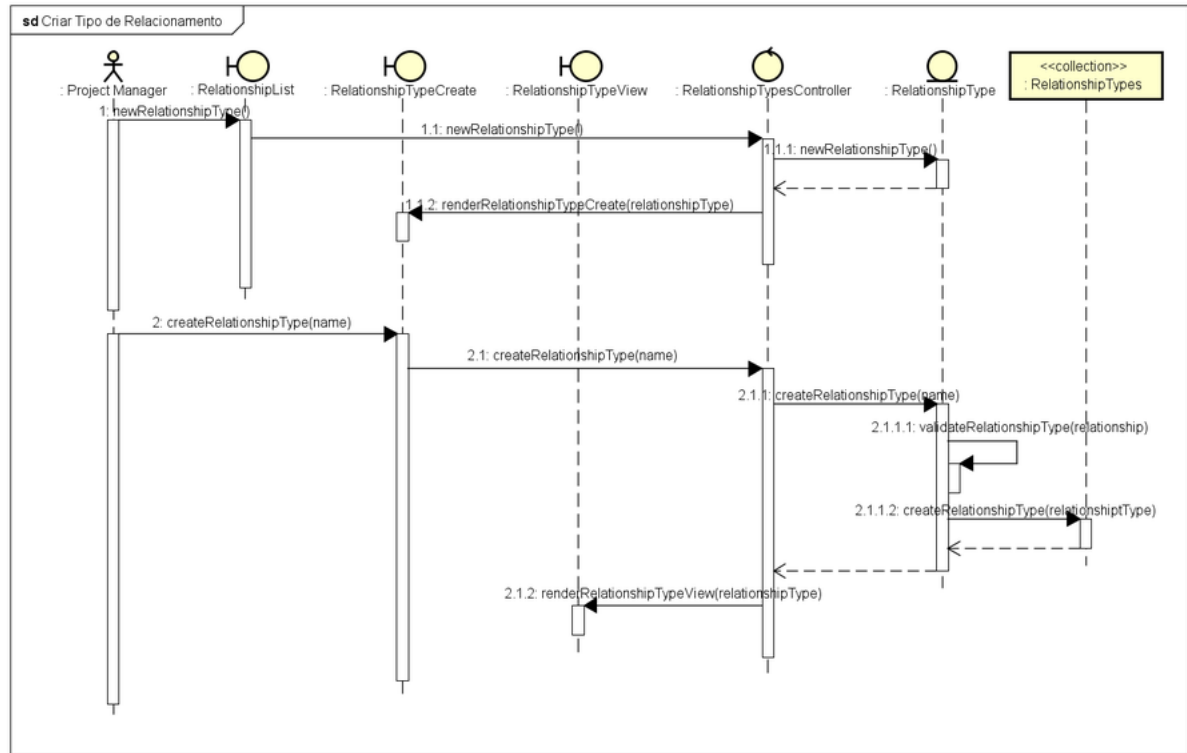
F.1 DS1 - Cadastrar Usuário

Figura 16 – Diagrama de Sequência - Cadastrar Usuário



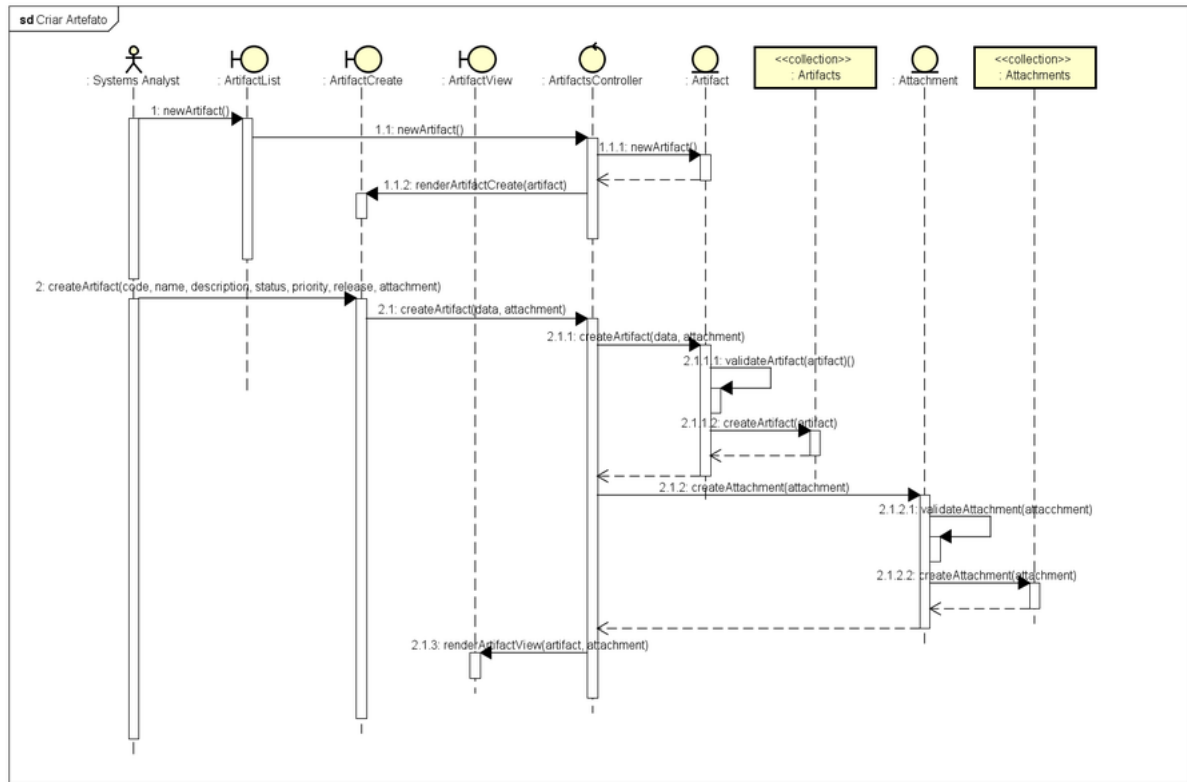
F.2 DS2 - Criar Projeto**Figura 17 – Diagrama de Sequência - Criar Projeto**

F.3 DS3 - Criar Tipo de Artefato**Figura 18 – Diagrama de Sequência - Criar Tipo de Artefato**

F.4 DS4 - Criar Tipo de Relacionamento**Figura 19 – Diagrama de Sequência - Criar Tipo de Relacionamento**

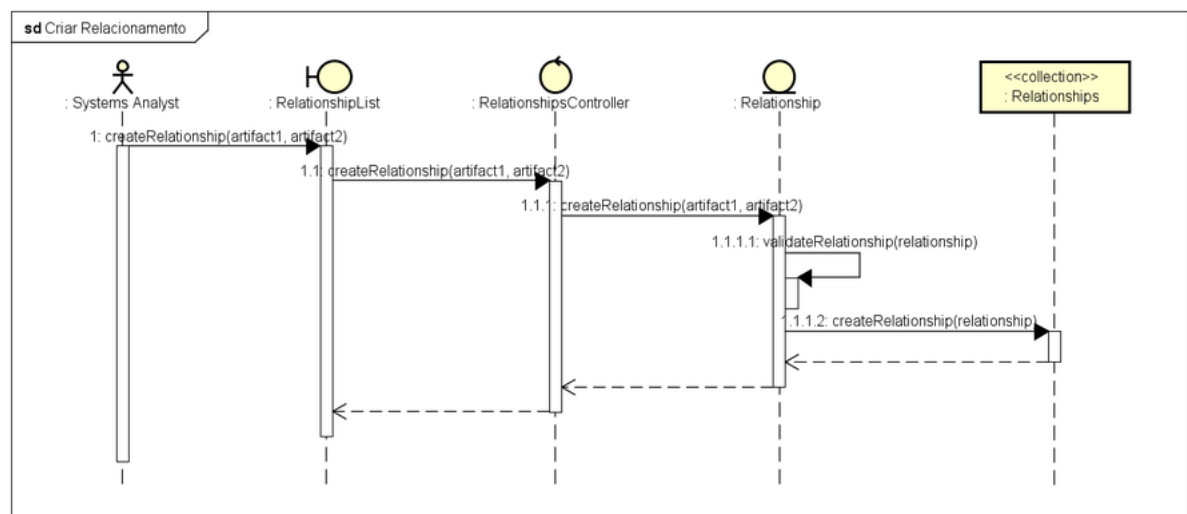
F.5 DS5 - Criar Artefato

Figura 20 – Diagrama de Sequência - Criar Artefato



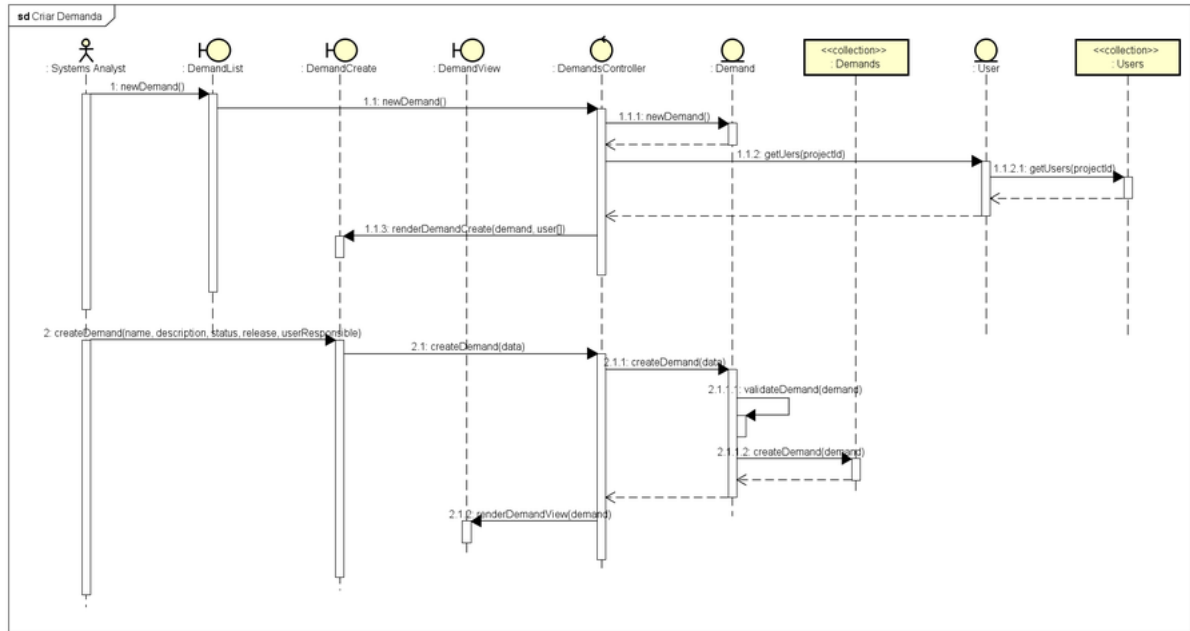
F.6 DS6 - Criar Relacionamento

Figura 21 – Diagrama de Sequência - Criar Relacionamento



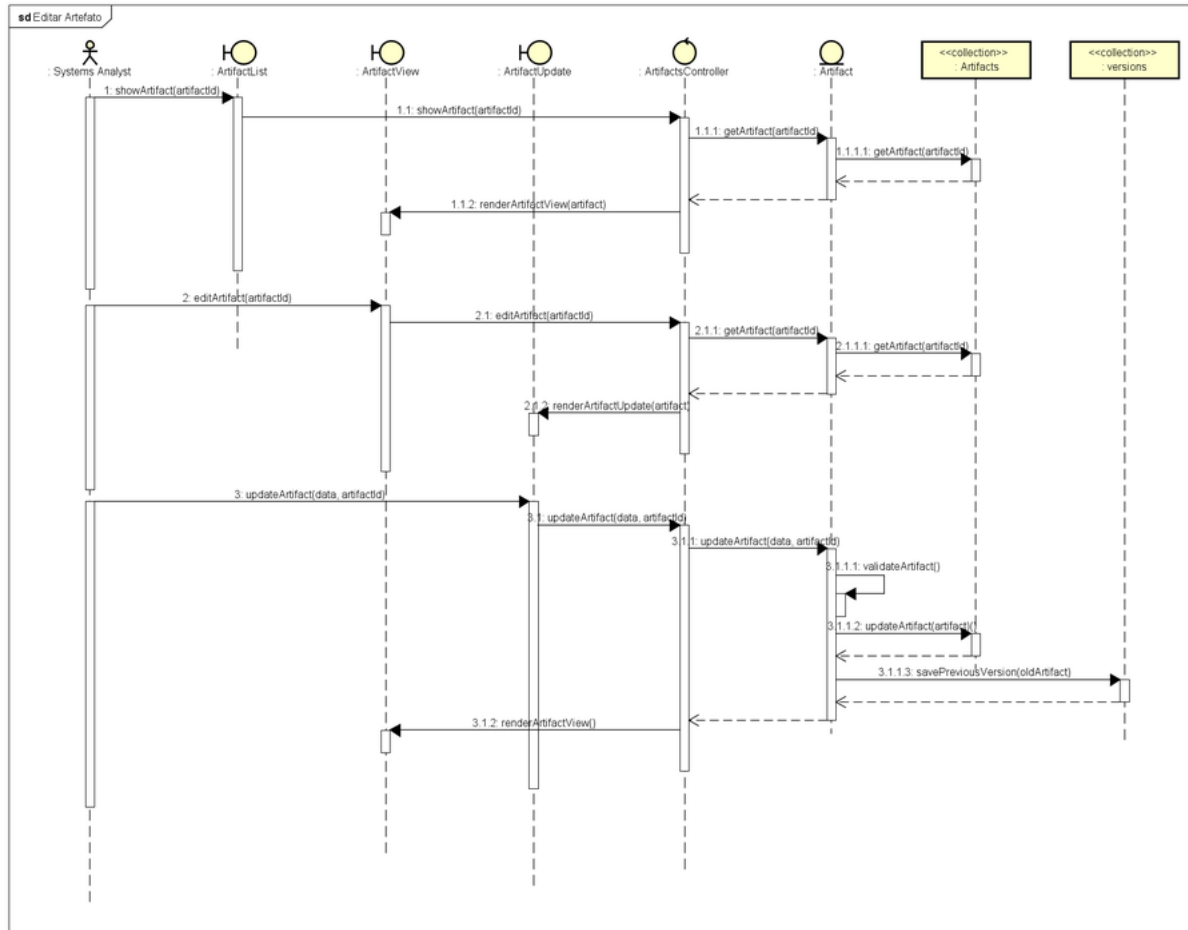
F.7 DS7 - Criar Demanda

Figura 22 – Diagrama de Sequência - Criar Demanda



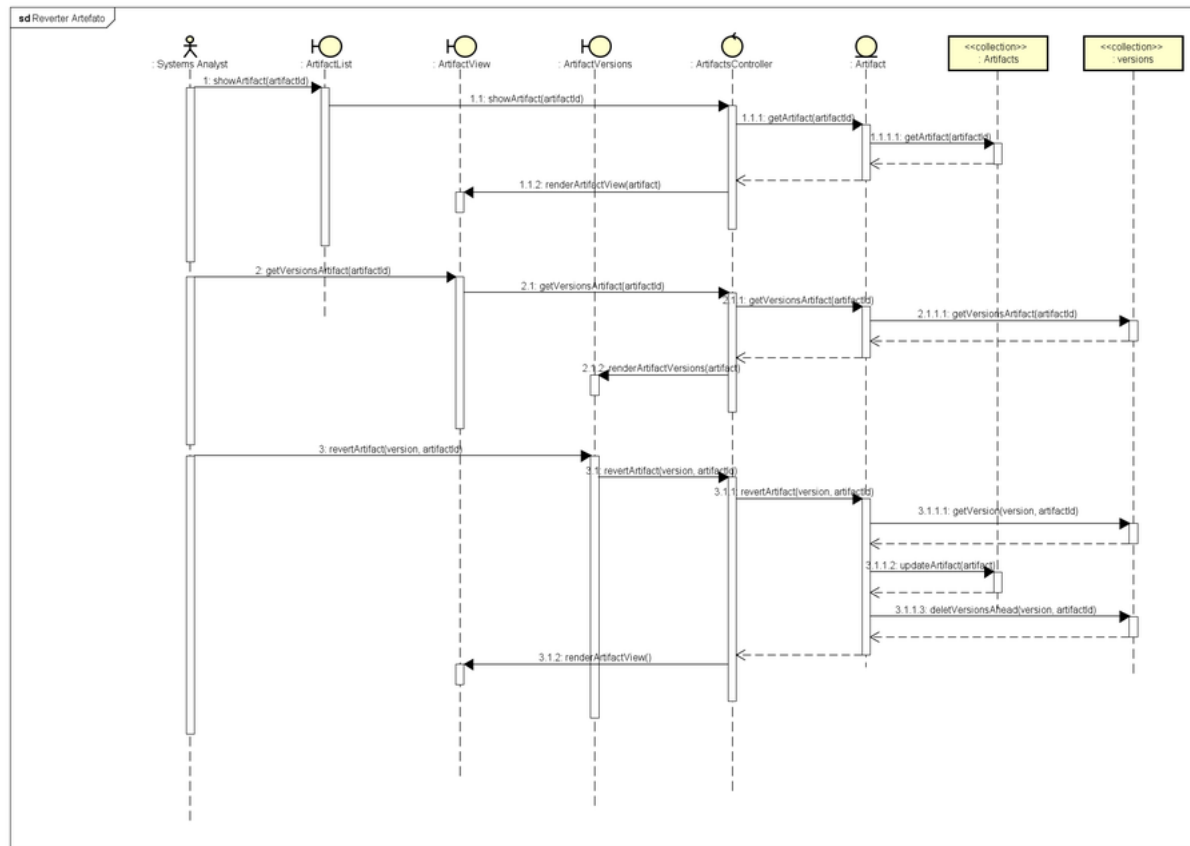
F.8 DS8 - Editar Artefato

Figura 23 – Diagrama de Sequência - Editar Artefato



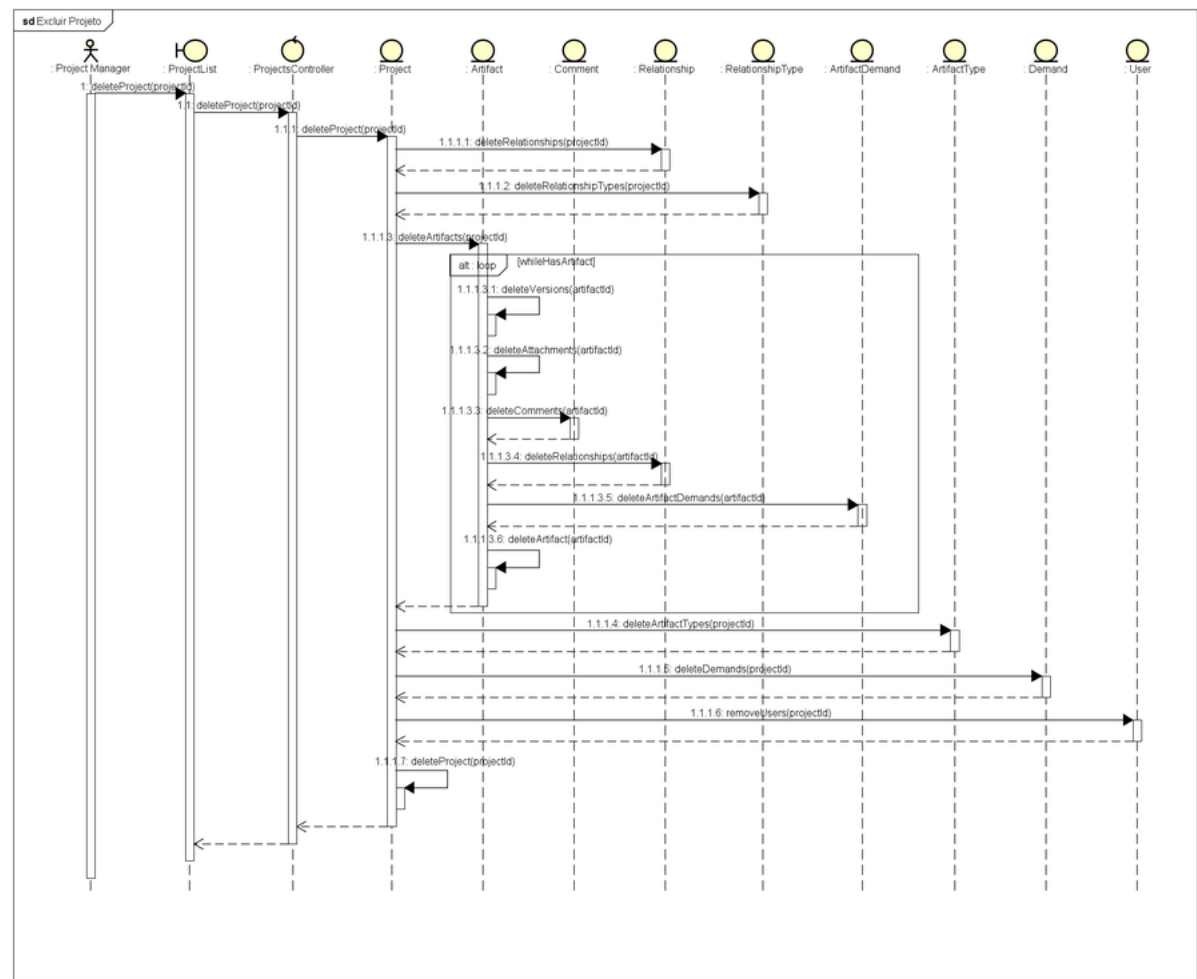
F.9 DS9 - Reverter Artefato

Figura 24 – Diagrama de Sequência - Reverter Artefato



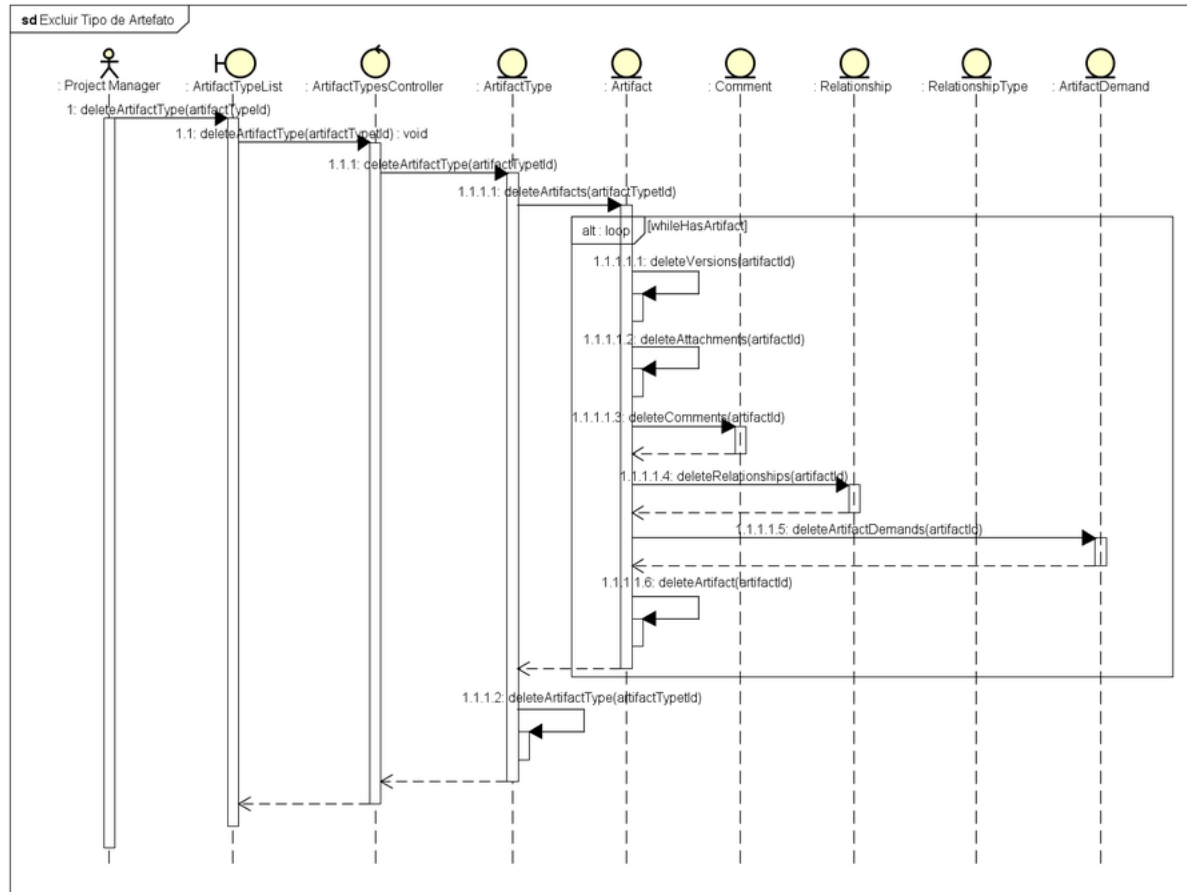
F.10 DS10 - Excluir Projeto

Figura 25 – Diagrama de Sequência - Excluir Projeto



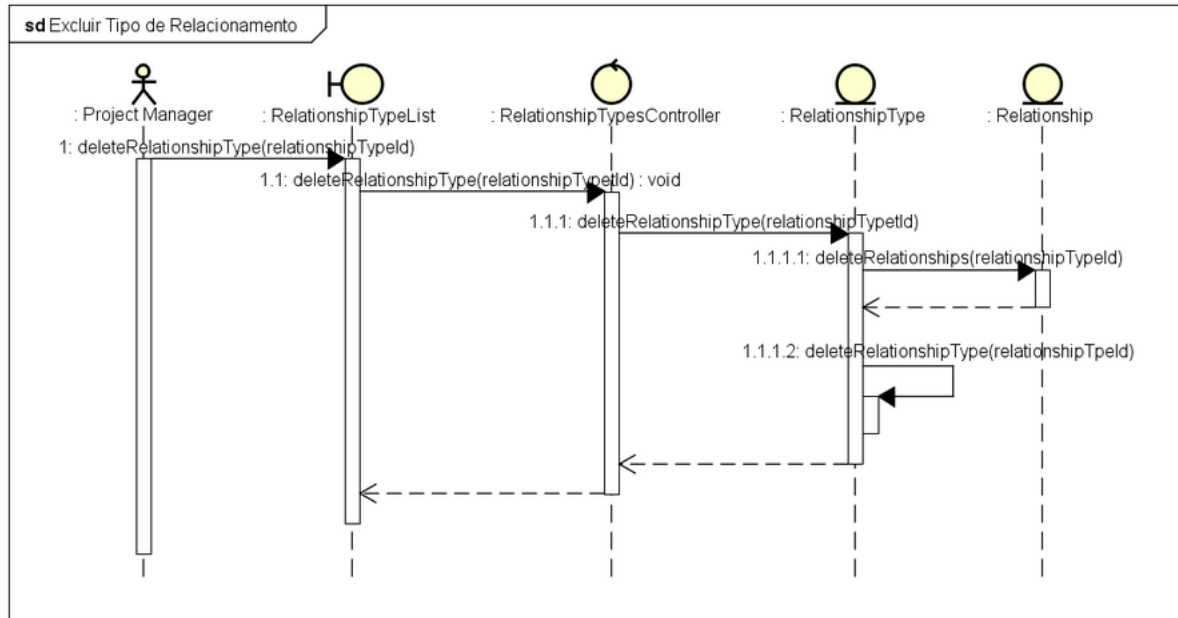
F.11 DS11 - Excluir Tipo de Artefato

Figura 26 – Diagrama de Sequência - Excluir Tipo de Artefato



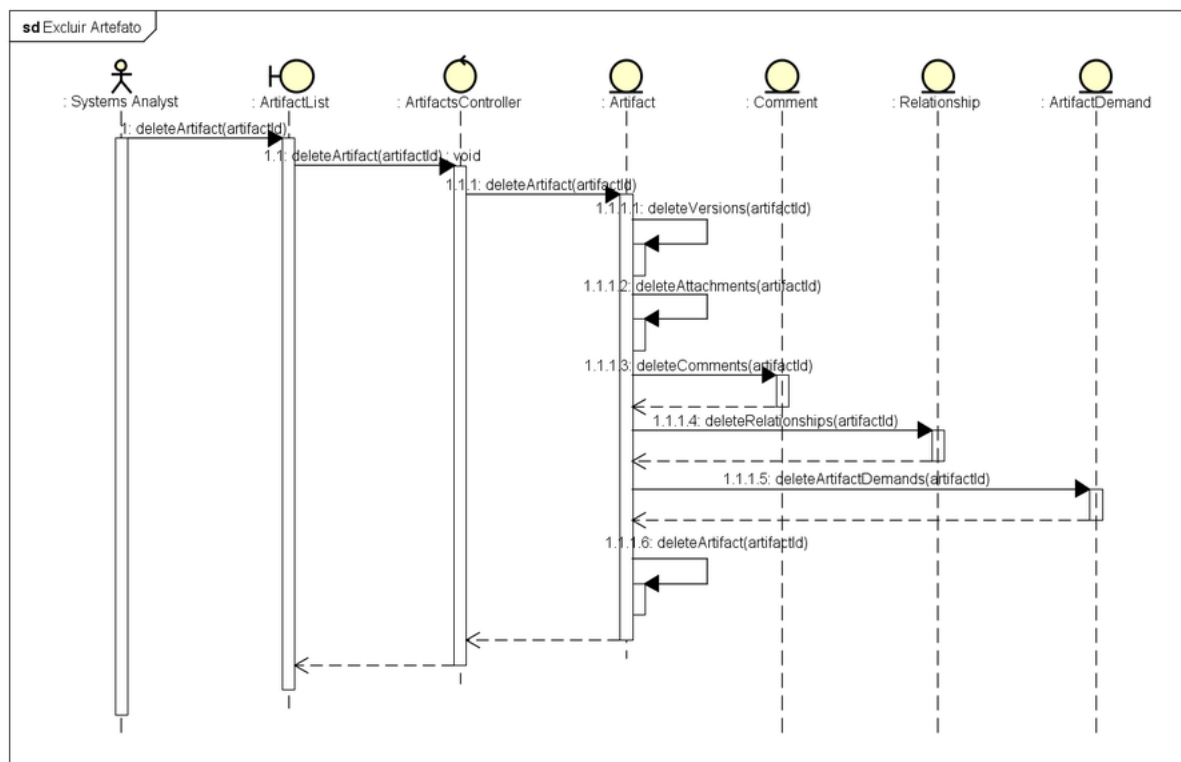
F.12 DS12 - Excluir Tipo de Relacionamento

Figura 27 – Diagrama de Sequência - Excluir Tipo de Relacionamento



F.13 DS13 - Excluir Artefato

Figura 28 – Diagrama de Sequência - Excluir Artefato



F.14 DS14 - Excluir Demanda

Figura 29 – Diagrama de Sequência - Excluir Demanda

