

2011

Quid-IDE

Framework de Documentação

Johnys Custódio da Silva Rabelo
Leonardo Eustáquio Pessoa de Oliveira

UNIVERSIDADE ESTADUAL DE GOIÁS
UNIDADE UNIVERSITÁRIA DE CIÊNCIAS EXATAS E TECNOLÓGICAS
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

JOHNYS CUSTÓDIO DA SILVA RABELO
LEONARDO EUSTÁQUIO PESSOA DE OLIVEIRA

Quid-IDE
Framework de Documentação

Anápolis
Dezembro, 2011

UNIVERSIDADE ESTADUAL DE GOIÁS
UNIDADE UNIVERSITÁRIA DE CIÊNCIAS EXATAS E TECNOLÓGICAS
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

JOHNYS CUSTÓDIO DA SILVA RABELO
LEONARDO EUSTÁQUIO PESSOA DE OLIVEIRA

Quid-IDE
Framework de Documentação

Trabalho de Conclusão de Curso apresentado ao Departamento de Sistemas de Informação da Unidade Universitária de Ciências Exatas e Tecnológicas da Universidade Estadual de Goiás como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Esp. Guiliano Rangel

Anápolis

Dezembro, 2011

UNIVERSIDADE ESTADUAL DE GOIÁS
UNIDADE UNIVERSITÁRIA DE CIÊNCIAS EXATAS E TECNOLÓGICAS
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Quid-IDE
Framework de Documentação

Trabalho de Conclusão de Curso apresentada ao Departamento de Sistemas de Informação da Unidade Universitária de Ciências Exatas e Tecnológicas da Universidade Estadual de Goiás, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Aprovado por:

Guiliano Rangel, Especialista, UEG – UnUCET.

(ORIENTADOR)

Márcio Giovane Cunha Fernandes, Mestre, UEG – UnUCET.

(EXAMINADOR)

Anápolis, 26 de Novembro de 2011.

FICHA CATALOGRÁFICA

RABELO, Johnys Custódio da Silva; OLIVEIRA, Leonardo Eustáquio Pessoa de. **Quid-IDE: Framework de Documentação.** Anápolis 2011. UEG / UnUCET, Bacharelado em Sistemas de Informação, 2011.

Projeto de Software. Universidade Estadual de Goiás, Unidade Universitária de Ciências Exatas e Tecnológicas. Departamento de Sistemas de Informação.

1. Framework 2. Documentação 3. Artefato 4. Serviço 5. Dependência 6. Componente

REFERÊNCIAS BIBLIOGRÁFICAS

SANTOS, Jonathan Bruno Silva. **Framework de Documentação: Framework base para a documentação de software.** Anápolis, 2010. 220 p. Documento de Modelagem de Sistema – Curso de Sistemas de Informação, UnUCET, Universidade Estadual de Goiás.

RABELO, Johnys Custódio da Silva; OLIVEIRA, Leonardo Eustáquio Pessoa de. **Quid-IDE: Framework de Documentação.** Anápolis, 2011. xx p. Projeto de Software – Curso de Sistemas de Informação, UnUCET, Universidade Estadual de Goiás.

CESSÃO DE DIREITOS

NOME DO AUTOR: Johnys Custódio da Silva Rabelo, Leonardo Eustáquio Pessoa de Oliveira

TÍTULO DO TRABALHO: Quid-IDE: Framework de Documentação.

GRAU/ANO: Bacharelado em Sistemas de Informação / 2011-2.

É concedida à Universidade Estadual de Goiás permissão para reproduzir cópias deste trabalho, emprestar ou vender tais cópias para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte deste trabalho pode ser reproduzida sem a autorização por escrito do autor.

Johnys Custódio da Silva Rabelo

Rua Assis Brasil Qd: 38 Lt: 559 A
Vila Jaiara – Anápolis – GO – Brasil
CEP: 75064-090

Leonardo Eustáquio Pessoa de Oliveira

Rua Prof.^a Ester C. Amaral Qd: 13 Lt: 17
Bougainville – Anápolis – GO – Brasil
CEP: 75075-640

Dedico este trabalho a você, caro leitor, que viu neste projeto
uma inspiração para novas idéias.

AGRADECIMENTOS

Agradecemos primeiramente a Deus que nos deu esta oportunidade, á nossa família que sempre nos incentivaram a buscar conhecimento e viver em pleno crescimento intelectual. Aos professores que serviram como exemplos e alguns como contraexemplos, aos colegas que nos acompanharam nesta magnífica jornada, ao orientador Prof. Esp. Guiliano Rangel que confiou em nossa capacidade e nos apoiou nesta incrível experiência.

A todos, os nossos sinceros agradecimentos.

LISTA DE ABREVIATURAS E SIGLAS

Siglas	Descrição
DMS	Documento de Modelagem de Software
UEG	Universidade Estadual de Goiás
UnUCET	Unidade Universitária de Ciências Exatas e Tecnológicas
HTML	HyperText Markup Language
XHTML	Extensible HyperText Markup Language
XML	eXtensible Markup Language
PDF	Portable Document Format
CRUD	Create, Retrieve, Update and Delete.
ABNT	Associação Brasileira de Normas Técnicas
LATEX	Software de editoração eletrônica desenvolvido por Leslie Lamport escrevendo documentos em TeX, por sua vez desenvolvido por Donald Knuth.

RESUMO

Este trabalho apresenta o projeto de um framework de documentação, o Quid-IDE. Seu objetivo é facilitar o processo de documentação de um projeto com estrutura fixa, ou seja, um projeto formado em cima de um template, fazendo com que o usuário se concentre apenas no conteúdo e o framework cuide de todo o resto, desde a formatação até a publicação.

O software abrange três atividades distintas: o desenvolvimento de componentes, a criação de templates utilizando tais componentes e o preenchimento dos templates. Para isto foram definidos alguns perfis de usuário: Administrador, Gerente de Projetos, Desenvolvedor, Montador, Preenchedor. Com a infra-estrutura oferecida pelo framework, o desenvolvedor cria componentes (TipoMembro e Serviço) e os mapeia à framework, cada qual com uma função específica. O montador é responsável por identificar os elementos (Artefato e Membro) de um template (Modelo) e criá-los dentro da aplicação, utilizando os componentes disponíveis. O usuário preenchedor encontra o template pronto e concentra todo seu trabalho apenas no preenchimento. Devido à necessidade de uma equipe trabalhar no mesmo projeto, o Gerente de Projeto é responsável por definir equipes, ou seja, selecionar dentre os usuários aqueles que farão parte do projeto. O Administrador é o responsável pela aplicação, algumas de suas responsabilidades são o cadastro de usuários e a definição do perfil de cada um.

O framework controla todas suas entidades, portanto todas as funcionalidades a serem ativadas pelo usuário são acionadas através da camada superior do framework, ou seja, todo o processamento é abstraído do usuário. As funcionalidades documentadas neste trabalho são: a mapeamento (importação) de componentes, o relacionamento entre artefatos (dependência), a montagem de artefatos, a montagem do modelo, o registro de log das operações do sistema e a persistência de todos os dados gerados e a validação de dados.

Palavras-chave: Framework, Componente, Artefato, Serviço, Modelo, TipoMembro, Membro, Documento, Estrutura, Rastreabilidade, Template, DMS, Projeto, Extensibilidade, Plugins.

ABSTRACT

This work presents a framework for the project documentation, the Quid-IDE. Your goal is to facilitate the process of documenting a project with fixed structure, ie, a project formed on top of a template, causing the user to focus only on the content and the framework takes care of everything else, from formatting to publication.

The software comprises three distinct activities: the development of components, creating templates using such components and filling of templates. For some that were defined user profiles: Administrator, Project Manager, Developer, Editor, Form Filler. With the infrastructure provided by the framework, the developer creates components (TipoMembro and Service) and the framework maps, each with a specific function. The assembler is responsible for identifying the elements (and Artifact State) to a template (template) and create them within the application using the components available. The user is fulfilling the template ready and concentrates all his work only in the filling. Due to the necessity of a team working on the same project, the Project Manager is responsible for defining teams, or select from users who are part of the project. The manager is responsible for implementing some of its responsibilities are the user registration and profiling of each.

The framework manages all your bodies, so all the features to be activated by the user are driven through the top layer of the framework, ie, all processing is abstracted from the user. The features documented in this paper are: a mapping (import) of components, the relationship between artifacts (dependence), the assembly of artifacts, assembling the model, the log of the operations of the system and the persistence of all data generated and validation of data.

Keywords: Framework, Component, Artifact, Service Model, TipoMembro, State, Document Structure, Tracking, Template, DMS, Design, Extensibility, Plugins.

PREFÁCIO

Ninguém nunca deve ter pensado que uma documentação de baixa qualidade fosse positiva, mas até onde isso pode prejudicar? Se um projeto for de qualidade, mas tiver erros em sua documentação ele pode ser mantido e evoluir? Na maioria das vezes não, mesmo porque, se o objetivo é a qualidade, o cuidado em cada detalhe é, sem dúvida, essencial!

A documentação interfere diretamente na recepção de um produto. No caso de um software, por exemplo, ela é a comunicação direta com o usuário. Também os artigos merecem cuidados, pois disso depende a boa divulgação de qualquer produto.

O framework Quid-IDE têm por base o auxílio nas tarefas onde é necessário se documentar algo, ou seja, o processo de documentação.

Este documento aborda os princípios básicos, as idéias iniciais, como seus recursos e características fundamentais serão gerenciados para a definição do negócio. Para isto, aqui está registrado todos os procedimentos utilizados para o entendimento e a construção do software proposto, ou seja, sua documentação e implementação.

Este documento está organizado da seguinte forma:

- Gerência do Projeto: parte onde é descrito o processo de desenvolvimento de software, cronograma de atividades, iterações do projeto e os relatórios semanais de progresso;

- Dicionário do Projeto: contém a lista de abreviaturas nos artefatos do documento, elementos de dados e elementos de comportamento;

- Modelagem do Negócio: contêm as atas de reuniões, a descrição do negócio, a lista de requisitos, a lista de regras de negócio, as dificuldades encontradas no negócio, a definição do problema, o objetivo do software, o escopo do software, os clientes alvo do negócio e o glossário do negócio;

- Requisitos: contem todos os detalhes dos requisitos, as ações possíveis e um mapeamento dos tipos de dados relacionado a cada requisito em específico.

- Casos de uso: contem todos os detalhes de cenários e funcionalidades que serão implementadas.

- Arquitetura: contém todas as informações necessárias para o entendimento do funcionamento do framework.

TABELA DE REVISÕES

Versão	Principais Autores	Descrição da Versão	Data de Término
V 0.1	Leonardo Pessoa Johnys Custódio	Criação deste documento para entrega de pré-projeto.	07/07/2011
V 0.2	Leonardo Pessoa Johnys Custódio	Adição das especificações dos requisitos para banca parcial.	18/06/2011
V 0.3	Leonardo Pessoa Johnys Custódio	Adição dos detalhamentos de caso de uso e da especificação da arquitetura projeto.	08/11/2011
V 1.0	Leonardo Pessoa Johnys Custódio	Aplicação das correções exigidas pela banca examinadora.	06/12/2011

Tabela 1 – Tabela de Revisões

SUMÁRIO

1	Lista de Elementos Gráficos.....	19
1.1	Lista de Tabelas.....	19
1.2	Lista de Figuras	23
2	Dicionário do Projeto	26
2.1	Listade Abreviaturas nos Artefatos do Documento.....	26
2.2	Elementos de Dados	26
2.3	Elementos de comportamento	26
3	Gerência do Projeto	26
3.1	Processo de Desenvolvimento do Software	26
3.1.1	<i>Levantamento de Requisitos.....</i>	26
3.1.2	<i>Análise de Requisitos</i>	27
3.1.3	<i>Implementação e Teste de Casos de Uso</i>	27
3.1.4	<i>Finalização da Documentação e Implantação.....</i>	27
3.2	Cronograma de Atividades.....	28
3.3	Iterações do Projeto	29
3.4	Relatório Semanal de Progresso.....	30
3.4.1	<i>Relatório Semanal de Progresso n.º 01.....</i>	30
3.4.2	<i>Relatório Semanal de Progresso n.º 02.....</i>	30
3.4.3	<i>Relatório Semanal de Progresso n.º 03.....</i>	30
3.4.4	<i>Relatório Semanal de Progresso n.º 04.....</i>	31
3.4.5	<i>Relatório Semanal de Progresso n.º 05.....</i>	31
3.4.6	<i>Relatório Semanal de Progresso n.º 06.....</i>	32
4	Modelagem do Negócio	32
4.1	Ata de Reunião	32
4.1.1	<i>Ata n.º 01</i>	32
4.1.2	<i>Ata n.º 02</i>	33
4.1.3	<i>Ata n.º 03</i>	34
4.1.4	<i>Ata n.º 04</i>	36
4.1.5	<i>Ata n.º 05</i>	39
4.1.6	<i>Ata n.º 06</i>	42
4.1.7	<i>Ata n.º 07</i>	43
4.1.8	<i>Ata n.º 08</i>	44
4.1.9	<i>Ata n.º 09</i>	45
4.1.10	<i>Ata n.º 10</i>	47
4.1.11	<i>Ata n.º 11</i>	48
4.1.12	<i>Ata n.º 12</i>	49
4.1.13	<i>Ata n.º 13</i>	50
4.2	Descrição do Negócio	51
4.3	Lista de Requisitos	52
4.4	Lista de Regras de Negócio.....	52
4.5	Dificuldades Encontradas no Negócio	54

4.6	Definição do Problema.....	54
4.7	Objetivo do Software	55
4.8	Escopo do Software.....	56
4.9	Clientes Alvo.....	57
4.10	Glossário do Negócio	57
5	Requisitos	57
5.1	Especificação de Requisitos n.º 01	57
5.1.1	<i>R01 – Manter TipoMembro</i>	57
5.1.2	<i>R01 – Protótipo de Interface</i>	60
5.1.3	<i>R01 – Informações Relacionadas</i>	61
5.1.4	<i>R01 – Funcionalidades Relacionadas</i>	61
5.2	Especificação de Requisitos n.º 02	62
5.2.1	<i>R02 – Manter Membro</i>	62
5.2.2	<i>R02 – Protótipo de Interface</i>	62
5.2.3	<i>R02 – Informações Relacionadas</i>	62
5.2.4	<i>R02 – Funcionalidades Relacionadas</i>	63
5.3	Especificação de Requisitos n.º 03	63
5.3.1	<i>R03 – Manter Artefato</i>	63
5.3.2	<i>R03 – Protótipo de Interface</i>	64
5.3.3	<i>R03 – Informações Relacionadas</i>	64
5.3.4	<i>R03 – Funcionalidades Relacionadas</i>	64
5.4	Especificação de Requisitos n.º 04.....	65
5.4.1	<i>R04 – Montar Artefato</i>	65
5.4.2	<i>R04 – Protótipo de Interface</i>	66
5.4.3	<i>R04 – Informações Relacionadas</i>	69
5.4.4	<i>R04 – Funcionalidades Relacionadas</i>	70
5.5	Especificação de Requisitos n.º 05	70
5.5.1	<i>R05 – Clonar Artefato</i>	70
5.5.2	<i>R05 – Protótipo de Interface</i>	71
5.5.3	<i>R05 – Informações Relacionadas</i>	71
5.5.4	<i>R05 – Funcionalidades Relacionadas</i>	71
5.6	Especificação de Requisitos n.º 06.....	71
5.6.1	<i>R06 – Preencher Artefato</i>	71
5.6.2	<i>R06 – Protótipo de Interface</i>	72
5.6.3	<i>R06 – Informações Relacionadas</i>	74
5.6.4	<i>R06 – Funcionalidades Relacionadas</i>	74
5.7	Especificação de Requisitos n.º 07	75
5.7.1	<i>R07 – Visualizar Artefato</i>	75
5.7.2	<i>R07 – Protótipo de Interface</i>	75
5.7.3	<i>R07 – Informações Relacionadas</i>	75
5.7.4	<i>R07 – Funcionalidades Relacionadas</i>	75
5.8	Especificação de Requisitos n.º 08.....	75
5.8.1	<i>R08 – Publicar Artefato</i>	76
5.8.2	<i>R08 – Protótipo de Interface</i>	76
5.8.3	<i>R08 – Informações Relacionadas</i>	76
5.8.4	<i>R08 – Funcionalidades Relacionadas</i>	76
5.9	Especificação de Requisitos n.º 09	76

5.9.1	<i>R09 – Manter Modelo</i>	76
5.9.2	<i>R09 – Protótipo de Interface</i>	78
5.9.3	<i>R09 – Informações Relacionadas</i>	79
5.9.4	<i>R09 – Funcionalidades Relacionadas</i>	79
5.10	Especificação de Requisitos n.º 10	79
5.10.1	<i>R10 – Manter Projeto</i>	79
5.10.2	<i>R10 – Protótipo de Interface</i>	80
5.10.3	<i>R10 – Informações Relacionadas</i>	81
5.10.4	<i>R10 – Funcionalidades Relacionadas</i>	81
5.11	Especificação de Requisitos n.º 11	81
5.11.1	<i>R11 – Manter Serviços</i>	81
5.11.2	<i>R11 – Protótipo de Interface</i>	82
5.11.3	<i>R11 – Informações Relacionadas</i>	83
5.11.4	<i>R11 – Funcionalidades Relacionadas</i>	83
5.12	Especificação de Requisitos n.º 12	83
5.12.1	<i>R12 – Gerenciar Serviço de Persistência</i>	84
5.12.2	<i>R12 – Protótipo de Interface</i>	84
5.12.3	<i>R12 – Informações Relacionadas</i>	84
5.12.4	<i>R12 – Funcionalidades Relacionadas</i>	84
5.13	Especificação de Requisitos n.º 13	85
5.13.1	<i>R13 – Gerenciar Serviço de Dependências</i>	85
5.13.2	<i>R13 – Protótipo de Interface</i>	86
5.13.3	<i>R13 – Informações Relacionadas</i>	86
5.13.4	<i>R13 – Funcionalidades Relacionadas</i>	86
5.14	Especificação de Requisitos n.º 14	87
5.14.1	<i>R14 – Manter Log das Operações do Sistema</i>	87
5.14.2	<i>R14 – Protótipo de Interface</i>	88
5.14.3	<i>R14 – Informações Relacionadas</i>	88
5.14.4	<i>R14 – Funcionalidades Relacionadas</i>	88
5.15	Especificação de Requisitos n.º 15	89
5.15.1	<i>R15 – Gerenciar Multiusuários</i>	89
5.15.2	<i>R15 – Protótipo de Interface</i>	90
5.15.3	<i>R15 – Informações Relacionadas</i>	92
5.15.4	<i>R15 – Funcionalidades Relacionadas</i>	93
6	Modelagem de Casos de Uso	93
6.1	Relação de Casos de Uso	93
6.2	Relação de Atores.....	94
6.3	Diagrama Geral de Casos de Uso.....	94
6.4	Detalhamento de Casos de Uso	95
6.4.1	<i>Caso de Uso 01 - Manter Artefato</i>	95
6.4.2	<i>Caso de Uso 02 - Montar Artefato</i>	109
6.4.3	<i>Caso de Uso 03 – Clonar Artefato</i>	117
6.4.4	<i>Caso de Uso 04 – Preencher Artefato</i>	119
6.4.5	<i>Caso de Uso 05 – Visualizar Artefato</i>	124
6.4.6	<i>Caso de Uso 06 – Publicar Artefato</i>	126
6.4.6.1	<i>Cenário 01 – Publicar Artefato</i>	126
6.4.7	<i>Caso de Uso R07 - Manter Membro</i>	127

6.4.8	<i>Caso de Uso 08 – Manter TipoMembro.....</i>	133
6.4.9	<i>Caso de Uso 09 – Manter Modelo.....</i>	140
6.4.10	<i>Caso de Uso 10 – Manter Projeto.....</i>	156
6.4.11	<i>Caso de Uso 11 - Manter Serviço</i>	162
6.4.12	<i>Caso de Uso 12 – Gerenciar Serviço</i>	168
6.4.13	<i>Caso de Uso 13 – Realizar Versionamento.....</i>	173
6.4.14	<i>Caso de Uso 14 – Manter Opções de Log das Operações do Sistema.....</i>	174
6.4.15	<i>Caso de Uso 15 – Gerenciar Multusuários.....</i>	175
6.4.16	<i>Caso de Uso 16 – Visualizar Log das Operações do Sistema.....</i>	187
6.4.17	<i>Caso de Uso 17 - Registrar Log das Operações do Sistema.....</i>	189
7	Modelagem de Dados	190
7.1	Projeto Conceitual de Dados	191
7.2	Projeto Lógico de Dados	192
7.3	Script de Criação do Banco de Dados	192
8	Especificação da Arquitetura	196
8.1	Visão Lógica	196
8.1.1	<i>Camadas Lógicas</i>	196
8.2	Visão de Implementação	198
8.2.1	<i>Definição de Herança por Camada</i>	198
8.3	Visão de Implantação	200
8.4	Diagrama de Componentes	201
8.5	Diagrama Geral de Sequência	202
8.6	Diagrama de Pacotes	202
8.6.1	<i>Pacote Configurações</i>	204
8.6.2	<i>Pacote Domínios</i>	204
8.6.3	<i>Pacote Enums.....</i>	204
8.6.4	<i>Pacote Excecoes.....</i>	204
8.6.5	<i>Pacote Interfaces.....</i>	204
8.6.6	<i>Pacote Núcleo</i>	204
8.6.7	<i>Pacote Utilitárias</i>	205
8.6.8	<i>Pacote Versionador.....</i>	205
8.6.9	<i>Pacote Logs.....</i>	205
8.6.10	<i>Pacote Persistência.....</i>	205
8.6.11	<i>Pacote Controladores</i>	205
8.6.12	<i>Pacote Managed.....</i>	205
8.6.13	<i>Pacote Descritor</i>	205
8.7	Padrões de Nomenclatura Utilizados na Codificação	205
8.7.1	<i>Classes.....</i>	205
8.7.2	<i>Métodos</i>	206
8.8	Informações para Extensão do Framework	206
8.8.1	<i>TipoMembro.....</i>	206
8.8.2	<i>Serviço.....</i>	207
8.8.3	<i>Arquivo .JAR</i>	207
9	Realização de Casos de Uso	207
9.1	Diagrama Geral de Classes do Projeto	208

10	Testes	208
10.1	Plano de Testes.....	208
10.2	Casos de Testes	208
10.2.1	<i>Casos de Testes CDU01 - Manter Artefato.....</i>	208
10.2.2	<i>Casos de Testes CDU02 - Montar Artefato.....</i>	215
10.2.3	<i>Casos de Testes CDU01 - Manter Artefato.....</i>	219
10.2.4	<i>Casos de Testes CDU04 - Preencher Artefato.....</i>	220
10.2.5	<i>Casos de Testes CDU05 - Visualizar Artefato</i>	223
10.2.6	<i>Casos de Testes CDU06 - Publicar Artefato.....</i>	223
10.2.7	<i>Casos de Testes CDU07 - Manter Membro</i>	224
10.2.8	<i>Casos de Testes CDU08 - Manter TipoMembro</i>	227
10.2.9	<i>Casos de Testes CDU09 - Manter Membro</i>	229
10.3	Relatório de Testes	233
11	Conclusão.....	234
12	Referências.....	235
13	Apêndices	236
	Apêndice I - Cronograma de Atividades do Trabalho de Conclusão de Curso.....	236
	Apêndice II - Pôster Apresentado no III Simpósio de Tecnologia da Informação e III Semana de Iniciação Científica do Curso de Sistemas de Informação UnUCET-UEG/2011	237
	Apendice III – Javadoc e API	238
14	Anexos	238

1 Lista de Elementos Gráficos

1.1 Lista de Tabelas

Tabela 1 – Tabela de Revisões	13
Tabela 2 – Lista de abreviaturas nos artefatos do documento	26
Tabela 3 – Processo de desenvolvimento do software	27
Tabela 4 – Cronograma de atividades	29
Tabela 5 – Iterações do projeto.....	29
Tabela 6 - Relatório Semanal de Progresso n.º 01.....	30
Tabela 7 - Relatório Semanal de Progresso n.º 02.....	30
Tabela 8 - Relatório Semanal de Progresso n.º 03.....	31
Tabela 9 - Relatório Semanal de Progresso n.º 04.....	31
Tabela 10 - Relatório Semanal de Progresso n.º 05	32
Tabela 11 - Relatório Semanal de Progresso n.º 06	32
Tabela 12 – Ata da Reunião n.º 01	33
Tabela 13 – Ata da Reunião n.º 02	34
Tabela 14 – Ata da Reunião n.º 03	36
Tabela 15 – Ata da Reunião n.º 04	39
Tabela 16 – Ata da Reunião n.º 05	42
Tabela 17 – Ata da Reunião n.º 06	43
Tabela 18 – Ata da Reunião n.º 07	44
Tabela 19 – Ata da Reunião n.º 08	45
Tabela 20 – Ata da Reunião n.º 09	47
Tabela 21 – Ata da Reunião n.º 10	48
Tabela 22 – Ata da Reunião n.º 11	49
Tabela 23 – Ata da Reunião n.º 12	50
Tabela 24 – Ata da Reunião n.º 13	51
Tabela 25 – Lista de Requisitos.....	52
Tabela 26 – Lista de regras de negócio	54
Tabela 27 – Lista de dificuldades encontradas no negócio	54
Tabela 28 – Lista de clientes alvo	57
Tabela 29 – Glossário do negócio	57
Tabela 30 – Tabela de especificação do requisito R01	59
Tabela 31 – Informações relacionadas ao requisito R01	61
Tabela 32 – Funcionalidades relacionadas ao requisito R01.....	61
Tabela 33 – Tabela de especificação do requisito R02	62
Tabela 34 – Informações relacionadas ao requisito R02	63
Tabela 35 – Funcionalidades relacionadas ao requisito R02.....	63
Tabela 36 – Tabela de especificação do requisito R03	64

Tabela 37 – Informações relacionadas ao requisito R03	64
Tabela 38 – Funcionalidades relacionadas ao requisito R03.....	65
Tabela 39 – Tabela de especificação do requisito R04	66
Tabela 40 – Funcionalidades relacionadas ao requisito R04.....	70
Tabela 41 – Tabela de especificação do requisito R05	70
Tabela 42 – Funcionalidades relacionadas ao requisito R05.....	71
Tabela 43 – Tabela de especificação do requisito R06	72
Tabela 44 – Funcionalidades relacionadas ao requisito R06.....	74
Tabela 45 – Tabela de especificação do requisito R07	75
Tabela 46 – Funcionalidades relacionadas ao requisito R07.....	75
Tabela 47 – Tabela de especificação do requisito R08	76
Tabela 48 – Funcionalidades relacionadas ao requisito R08.....	76
Tabela 49 – Tabela de especificação do requisito R09	77
Tabela 50 – Informações relacionadas ao requisito R09	79
Tabela 51 – Funcionalidades relacionadas ao requisito R09.....	79
Tabela 52 – Tabela de especificação do requisito R10	80
Tabela 53 – Informações relacionadas ao requisito R10.....	81
Tabela 54 – Funcionalidades relacionadas ao requisito R10.....	81
Tabela 55 – Tabela de especificação do requisito R11	82
Tabela 56 – Informações relacionadas ao requisito R11	83
Tabela 57 – Funcionalidades relacionadas ao requisito R11.....	83
Tabela 58 – Tabela de especificação do requisito R12	84
Tabela 59 – Funcionalidades relacionadas ao requisito R12.....	85
Tabela 60 – Tabela de especificação do requisito R13	86
Tabela 61 – Funcionalidades relacionadas ao requisito R13.....	87
Tabela 62 – Tabela de especificação do requisito R14	88
Tabela 63 – Informações relacionadas ao requisito R14.....	88
Tabela 64 – Funcionalidades relacionadas ao requisito R14.....	89
Tabela 65 – Tabela de especificação do requisito R15	89
Tabela 66 – Informações relacionadas ao requisito R15	93
Tabela 67 – Funcionalidades relacionadas ao requisito R15.....	93
Tabela 68 – Lista de casos de uso.....	94
Tabela 69 – Lista atores do projeto	94
Tabela 70 – CDU01 – Detalhamento Cenário 01	96
Tabela 71 – CDU01 – Detalhamento Cenário 02	98
Tabela 72 – CDU01 – Detalhamento Cenário 03	100
Tabela 73 – CDU01 – Detalhamento Cenário 04.....	102
Tabela 74 – CDU01 – Detalhamento Cenário 05	102
Tabela 75 – CDU01 – Detalhamento Cenário 06.....	104
Tabela 76 – CDU01 – Detalhamento Cenário 07	105
Tabela 77 – CDU01 – Detalhamento Cenário 08.....	107
Tabela 78 – CDU02 – Detalhamento Cenário 01	110
Tabela 79 – CDU02 – Detalhamento Cenário 02	111
Tabela 80 – CDU02 – Detalhamento Cenário 03	112

Tabela 81 – CDU02 – Detalhamento Cenário 04.....	114
Tabela 82 – CDU02 – Detalhamento Cenário 05.....	115
Tabela 83 – CDU02 – Detalhamento Cenário 06.....	117
Tabela 84 – CDU03 – Detalhamento Cenário 01	118
Tabela 85 – CDU04 – Detalhamento Cenário 01	120
Tabela 86 – CDU04 – Detalhamento Cenário 02.....	122
Tabela 87 – CDU04 – Detalhamento Cenário 03.....	123
Tabela 88 – CDU04 – Detalhamento Cenário 04.....	124
Tabela 89 – CDU05 – Detalhamento Cenário 01	125
Tabela 90 – CDU06 – Detalhamento Cenário 01	126
Tabela 91 – CDU07 – Detalhamento Cenário 01	128
Tabela 92 – CDU07 – Detalhamento Cenário 02.....	130
Tabela 93 – CDU07 – Detalhamento Cenário 03.....	132
Tabela 94 – CDU08 – Detalhamento Cenário 01	134
Tabela 95 – CDU08 – Detalhamento Cenário 02.....	135
Tabela 96 – CDU08 – Detalhamento Cenário 03.....	137
Tabela 97 – CDU08 – Detalhamento Cenário 04.....	139
Tabela 98 – CDU09 – Detalhamento Cenário 01	141
Tabela 99 – CDU09 – Detalhamento Cenário 02.....	143
Tabela 100 – CDU09 – Detalhamento Cenário 03.....	145
Tabela 101 – CDU09 – Detalhamento Cenário 04.....	146
Tabela 102 – CDU09 – Detalhamento Cenário 05.....	147
Tabela 103 – CDU09 – Detalhamento Cenário 06.....	149
Tabela 104 – CDU09 – Detalhamento Cenário 07.....	151
Tabela 105 – CDU09 – Detalhamento Cenário 08.....	153
Tabela 106 – CDU09 – Detalhamento Cenário 09.....	154
Tabela 107 – CDU09 – Detalhamento Cenário 10.....	155
Tabela 108 – CDU10 – Detalhamento Cenário 01	157
Tabela 109 – CDU10 – Detalhamento Cenário 02.....	158
Tabela 110 – CDU10 – Detalhamento Cenário 03.....	159
Tabela 111 – CDU10 – Detalhamento Cenário 04.....	161
Tabela 112 – CDU11 – Detalhamento Cenário 01	163
Tabela 113 – CDU11 – Detalhamento Cenário 02.....	164
Tabela 114 – CDU11 – Detalhamento Cenário 03.....	166
Tabela 115 – CDU11 – Detalhamento Cenário 04.....	167
Tabela 116 – CDU12 – Detalhamento Cenário 01	169
Tabela 117 – CDU12 – Detalhamento Cenário 02.....	171
Tabela 118 – CDU13 – Detalhamento Cenário 01	173
Tabela 119 – CDU14 – Detalhamento Cenário 01	175
Tabela 120 – CDU15 – Detalhamento Cenário 01	176
Tabela 121 – CDU15 – Detalhamento Cenário 02.....	178
Tabela 122 – CDU15 – Detalhamento Cenário 03.....	179
Tabela 123 – CDU15 – Detalhamento Cenário 04.....	180
Tabela 124 – CDU15 – Detalhamento Cenário 05.....	182

Tabela 125 – CDU15 – Detalhamento Cenário 06.....	183
Tabela 126 – CDU15 – Detalhamento Cenário 07.....	184
Tabela 127 – CDU15 – Detalhamento Cenário 08.....	186
Tabela 128 – CDU15 – Detalhamento Cenário 09.....	187
Tabela 129 – CDU16 – Detalhamento Cenário 01.....	188
Tabela 130 – CDU17 – Detalhamento Cenário 01.....	190
Tabela 131 – Casos de Teste da funcionalidade 01	209
Tabela 132 – Casos de Teste da funcionalidade 01	209
Tabela 133 – Casos de Teste da funcionalidade 01	210
Tabela 134 – Casos de Teste da funcionalidade 01	210
Tabela 135 – Casos de Teste da funcionalidade 01	210
Tabela 136 – Casos de Teste da funcionalidade 02.....	211
Tabela 137 – Casos de Teste da funcionalidade 02.....	211
Tabela 138 – Casos de Teste da funcionalidade 03.....	211
Tabela 139 – Casos de Teste da funcionalidade 03.....	212
Tabela 140 – Casos de Teste da funcionalidade 03.....	212
Tabela 141 – Casos de Teste da funcionalidade 03.....	213
Tabela 142 – Casos de Teste da funcionalidade 04.....	213
Tabela 143 – Casos de Teste da funcionalidade 04.....	213
Tabela 144 – Casos de Teste da funcionalidade 04.....	214
Tabela 145 – Casos de Teste da funcionalidade 04.....	214
Tabela 146 – Casos de Teste da funcionalidade 04.....	215
Tabela 147 – Casos de Teste da funcionalidade 01	215
Tabela 148 – Casos de Teste da funcionalidade 01	216
Tabela 149 – Casos de Teste da funcionalidade 02.....	216
Tabela 150 – Casos de Teste da funcionalidade 02.....	216
Tabela 151 – Casos de Teste da funcionalidade 03	217
Tabela 152 – Casos de Teste da funcionalidade 04.....	217
Tabela 153 – Casos de Teste da funcionalidade 04.....	218
Tabela 154 – Casos de Teste da funcionalidade 04.....	218
Tabela 155 – Casos de Teste da funcionalidade 05	218
Tabela 156 – Casos de Teste da funcionalidade 05	219
Tabela 157 – Casos de Teste da funcionalidade 05	219
Tabela 158 – Casos de Teste da funcionalidade 06	219
Tabela 159 – Casos de Teste da funcionalidade 01	220
Tabela 160 – Casos de Teste da funcionalidade 01	220
Tabela 161 – Casos de Teste da funcionalidade 01	221
Tabela 162 – Casos de Teste da funcionalidade 01	221
Tabela 163 – Casos de Teste da funcionalidade 02	221
Tabela 164 – Casos de Teste da funcionalidade 03	222
Tabela 165 – Casos de Teste da funcionalidade 03	222
Tabela 166 – Casos de Teste da funcionalidade 04	222
Tabela 167 – Casos de Teste da funcionalidade 01	223
Tabela 168 – Casos de Teste da funcionalidade 01	223

Tabela 169 – Casos de Teste da funcionalidade 01	224
Tabela 170 – Casos de Teste da funcionalidade 01	224
Tabela 171 – Casos de Teste da funcionalidade 01	224
Tabela 172 – Casos de Teste da funcionalidade 01	225
Tabela 173 – Casos de Teste da funcionalidade 01	225
Tabela 174 – Casos de Teste da funcionalidade 02.....	226
Tabela 175 – Casos de Teste da funcionalidade 02.....	226
Tabela 176 – Casos de Teste da funcionalidade 02.....	226
Tabela 177 – Casos de Teste da funcionalidade 02.....	227
Tabela 178 – Casos de Teste da funcionalidade 02.....	227
Tabela 179 – Casos de Teste da funcionalidade 01	227
Tabela 180 – Casos de Teste da funcionalidade 01	228
Tabela 181 – Casos de Teste da funcionalidade 01	228
Tabela 182 – Casos de Teste da funcionalidade 01	229
Tabela 183 – Casos de Teste da funcionalidade 01	229
Tabela 184 – Casos de Teste da funcionalidade 01	229
Tabela 185 – Casos de Teste da funcionalidade 02.....	230
Tabela 186 – Casos de Teste da funcionalidade 02.....	230
Tabela 187 – Casos de Teste da funcionalidade 03.....	231
Tabela 188 – Casos de Teste da funcionalidade 04.....	231
Tabela 189 – Casos de Teste da funcionalidade 05.....	232
Tabela 190 – Casos de Teste da funcionalidade 06.....	232
Tabela 191 – Casos de Teste da funcionalidade 07	233
Tabela 192 – Cronograma de atividades proposto no projeto do Trabalho de Conclusão de Curso	236

1.2 Lista de Figuras

Figura 1 – Importação de TipoMembro	60
Figura 2 – Exclusão de TipoMembro	60
Figura 3 – Criar Artefato	66
Figura 4 – Montar Artefato – Artefato vazio.....	67
Figura 5 – Montar Artefato – Seleção de membros.....	67
Figura 6 – Montar Artefato – Propriedades do membro selecionado.....	68
Figura 7 – Montar Artefato – Propriedades do membro representando um serviço	68
Figura 8 – Montar Artefato – Seleção de serviço	69
Figura 9 – Montar Artefato – Propriedades do serviço selecionado	69
Figura 10 – Clonar Artefato.....	71
Figura 11 – Preencher Artefato – Seleção do Projeto	72
Figura 12 – Preencher Artefato – Árvore de artefatos (Modelo)	73
Figura 13 – Preencher Artefato – Novo artefato	73
Figura 14 – Preencher Artefato – Tela de preenchimento.....	74

Figura 15 – Montar Modelo – Seleção dos artefatos.....	78
Figura 16 – Montar Modelo – Configuração de cada artefato adicionado.....	78
Figura 17 – Manter Projeto – Criar Projeto.....	81
Figura 18 – Gerenciar Multiusuários – CRUD usuário	90
Figura 19 – Gerenciar Multiusuários – Logon de usuário.....	91
Figura 20 – Gerenciar Multiusuários – CRUD equipe	92
Figura 21 – Diagrama geral de casos de uso	95
Figura 22 – Diagrama de Atividades Cenário 01 Mapear Artefato.....	97
Figura 23 – Diagrama de Atividades Cenário 02 Pesquisar Artefato.....	99
Figura 24 – Diagrama de Atividades Cenário 03 Remover Artefato.	100
Figura 25 – Diagrama de Atividades Cenário 04 Alterar Artefato (Rotina Inicial)	102
Figura 26 – Diagrama de Atividades Cenário 05 Alterar Artefato (Sem utilização em modelo)	103
Figura 27 – Diagrama de Atividades Cenário 06 Alterar Artefato (Utilizado em um modelo e em nenhum projeto)	104
Figura 28 – Diagrama de Atividades Cenário 07 Alterar Artefato (Utilizado em dois ou mais modelos e em nenhum projeto)	106
Figura 29 – Diagrama de Atividades Cenário 08 Alterar Artefato (Utilizado em um ou mais projetos)	108
Figura 30 – Diagrama de Atividades Cenário 01 – Criar Artefato.....	110
Figura 31 – Diagrama de Atividades Cenário 02 – Adicionar membro.....	112
Figura 32 – Diagrama de Atividades Cenário 03 – Remover membro	113
Figura 33 – Diagrama de Atividades Cenário 04 – Adicionar serviço.....	114
Figura 34 – Diagrama de Atividades Cenário 05 – Remover serviço.....	116
Figura 35 – Diagrama de Atividades Cenário 06 – Visualizar Artefato.	117
Figura 36 – Diagrama de Atividades Cenário 01 – Clonar Artefato	119
Figura 37 – Diagrama de Atividades Cenário 01 – Montar Interface de Preenchimento	121
Figura 38 – Diagrama de Atividades Cenário 02 – Interface de Preenchimento do Membro	122
Figura 39 – Diagrama de Atividades Cenário 03 – Armazenamento de Valor informado ao Membro	123
Figura 40 – Diagrama de Atividades Cenário 04 – Cancelar Preenchimento do Artefato....	124
Figura 41 – Diagrama de Atividades Cenário 01 – Visualizar Artefato	125
Figura 42 – Diagrama de Atividades Cenário 01 – Publicar Artefato.....	127
Figura 43 – Diagrama de Atividades Cenário 01 – Incluir Membro.....	129
Figura 44 – Diagrama de Atividades Cenário 02 – Pesquisar Membro	131
Figura 45 – Diagrama de Atividades Cenário 03 – Alterar Membro	132
Figura 46 – Diagrama de Atividades Cenário 01 – Mapear TipoMembro.....	134
Figura 47 – Diagrama de Atividades Cenário 02 – Verificação de Espelho	136
Figura 48 – Diagrama de Atividades Cenário 03 – Executar Rotina de Nova Versão de TipoMembro	138
Figura 49 – Diagrama de Atividades Cenário 04 – Executar Rotina de Nova Revisão de TipoMembro	140
Figura 50 – Diagrama de Atividades Cenário 01 – Montar Modelo	142
Figura 51 – Diagrama de Atividades Cenário 02 – Preencher Modelo	144

Figura 52 – Diagrama de Atividades Cenário 03 – Incluir Artefato no Modelo	145
Figura 53 – Diagrama de Atividades Cenário 04 – Remover Artefato do Modelo	146
Figura 54 – Diagrama de Atividades Cenário 05 – Alterar parâmetros do artefato no modelo	148
Figura 55 – Diagrama de Atividades Cenário 06 – Mapear Modelo	150
Figura 56 – Diagrama de Atividades Cenário 07 – Alterar Modelo	152
Figura 57 – Diagrama de Atividades Cenário 08 – Alterar Artefato no Framework	153
Figura 58 – Diagrama de Atividades Cenário 09 – Remover Modelo	154
Figura 59 – Diagrama de Atividades Cenário 10 – Clonar Modelo	156
Figura 60 – Diagrama de Atividades Cenário 01 – Mapear Projeto	157
Figura 61 – Diagrama de Atividades Cenário 02 – Remover Projeto	158
Figura 62 – Diagrama de Atividades Cenário 03 – Bloquear Projeto	160
Figura 63 – Diagrama de Atividades Cenário 04 – Preencher Projeto	162
Figura 64 – Diagrama de Atividades Cenário 01 – Mapear Serviço	164
Figura 65 – Diagrama de Atividades Cenário 02 – Pesquisar Serviço	165
Figura 66 – Diagrama de Atividades Cenário 03 – Remover Serviço	166
Figura 67 – Diagrama de Atividades Cenário 04 – Verificação dos Serviços Cadastrados ...	168
Figura 68 – Diagrama de Atividades Cenário 01 – Executar Serviço Geral	170
Figura 69 – Diagrama de Atividades Cenário 02 – Executar Serviço Rotina Interna	172
Figura 70 – Diagrama de Atividades Cenário 01 – Versionar	174
Figura 71 – Diagrama de Atividades Cenário 01 – Alterar Opção de Log	175
Figura 72 – Diagrama de Atividades Cenário 01 – Incluir Usuário	177
Figura 73 – Diagrama de Atividades Cenário 02 – Alterar Usuário	178
Figura 74 – Diagrama de Atividades Cenário 03 – Alterar Usuário	179
Figura 75 – Diagrama de Atividades Cenário 04 – Logon de Usuário	181
Figura 76 – Diagrama de Atividades Cenário 05 – Incluir Equipe	182
Figura 77 – Diagrama de Atividades Cenário 06 – Desativar Usuário	184
Figura 78 – Diagrama de Atividades Cenário 07 – Associar Usuários a uma Equipe	185
Figura 79 – Diagrama de Atividades Cenário 08 – Remover Usuários de uma Equipe	186
Figura 80 – Diagrama de Atividades Cenário 09 – Listar Usuários de uma Equipe	187
Figura 81 – Diagrama de Atividades Cenário 01 – Pesquisar Log	189
Figura 82 – Diagrama de Atividades Cenário 01 – Registrar Log	190
Figura 83 – Projeto Conceitual do Banco de Dados	191
Figura 84 – Projeto Lógico do Banco de Dados	192
Figura 85 – Diagrama de Dependência entre as Camadas Lógicas	197
Figura 86 – Diagrama de implantação	201
Figura 87 – Diagrama de componentes	201
Figura 88 – Diagrama de sequencia geral	202
Figura 89 – Diagrama de pacotes do framework	203
Figura 90 – Diagrama de pacotes da extensão	203
Figura 91 – Diagrama de pacotes da aplicação	204
Figura 92 – Diagrama geral de classes do framework	208
Figura 93 – Pôster Quid-IDE: Framework de Documentação	237

2 Dicionário do Projeto

2.1 Lista de Abreviaturas nos Artefatos do Documento

Artefato	Termo ou Sigla	Significado
Processo de Desenvolvimento de Software	V & V	Verificação e Validação
Cronograma	DER	Diagrama de Entidade-Relacionamento
Cronograma	MR	Modelo Relacional
Cronograma	Atividade	Atividade que precisa ser realizada.
Cronograma	Previsão	Data planejada para entrega da atividade concluída.
Cronograma	Entrega	Data em que a atividade foi realmente entregue.

Tabela 2 – Lista de abreviaturas nos artefatos do documento

2.2 Elementos de Dados

Os dados referente a este tópico estão apresentados no Apêndice III.

2.3 Elementos de comportamento

Os dados referente a este tópico estão apresentados no Apêndice III.

3 Gerência do Projeto

3.1 Processo de Desenvolvimento do Software

3.1.1 Levantamento de Requisitos

- a. Reunião com o orientador para o conhecimento do componente a ser desenvolvido;

Descrição da atividade: A cada início de um componente, é feita uma reunião com o orientador para a proposta de criação do componente, explicitando as linhas gerais de suas funcionalidades. Nesta reunião o orientador deverá esclarecer possíveis dúvidas, mudanças ou idéias para a criação do componente.

Nela deverá conter:

<ul style="list-style-type: none"> • Gravação da reunião em áudio; • Anotações das explicações do orientador ou dos orientandos; • Levantamento de idéias sobre o determinado componente. <p>b. Formulação da ata de reunião para documentação;</p> <p><i>Descrição da atividade:</i> Com os dados levantados na reunião, formular uma ata que demonstre todos os assuntos abordados.</p> <p>Nesta ata deverá conter:</p> <ul style="list-style-type: none"> • Participantes da reunião; • Hora e local da reunião; • Descrição do teor da reunião; • Descrição dos requisitos levantados; • Pontos a serem abordados na próxima reunião. <p>c. Identificação dos requisitos e regras de negócio;</p>
3.1.2 Análise de Requisitos
<ul style="list-style-type: none"> a. Formular documentos de especificação dos requisitos; b. Identificação e detalhamento dos casos de uso; c. Proposição da modelagem conceitual dos dados a serem armazenados; d. Proposição dos problemas ligados aos requisitos; e. Formulação dos diagramas classe e sequência; f. Reunião com o orientador para a validação dos artefatos gerados; g. Análise e modificação de possíveis mudanças;
3.1.3 Implementação e Teste de Casos de Uso
<ul style="list-style-type: none"> a. Criação dos casos de Verificação e Validação (V & V); b. Implementação dos casos de uso; c. Aplicação dos Casos de Verificação e Validação (V & V); d. Correção de possíveis erros; e. Reunião com o orientador para a demonstração do componente desenvolvido; f. Correção de possíveis mudanças.
3.1.4 Finalização da Documentação e Implantação
<ul style="list-style-type: none"> a. Agrupamento dos documentos gerados ao DMS; b. Disponibilizar componente no servidor de aplicação.

Tabela 3 – Processo de desenvolvimento do software

3.2 Cronograma de Atividades

Atividade	Responsável	Previsão	Entrega
1º Iteração			
Entendimento do negócio	Johnys Custódio Leonardo Pessoa	01/04	01/04
Levantamento e especificação dos requisitos e regras de negócio	Johnys Custódio Leonardo Pessoa	01/04	01/04
Definição das prioridades entre os casos de uso	Johnys Custódio Leonardo Pessoa	01/04	01/04
Definição da arquitetura	Johnys Custódio Leonardo Pessoa	01/04	01/04
2º Iteração			
Elaboração do diagrama de classe conceitual	Johnys Custódio Leonardo Pessoa	08/11	08/11
Elaboração do DER e do MR	Johnys Custódio Leonardo Pessoa	08/11	08/11
Detalhamento dos casos de uso	Johnys Custódio Leonardo Pessoa	08/11	08/11
Elaboração dos diagramas de classe	Johnys Custódio Leonardo Pessoa	08/11	08/11
Definição dos casos de validação e verificação	Johnys Custódio Leonardo Pessoa	08/11	08/11
Implementação do caso de uso: Manter TipoMembro	Johnys Custódio Leonardo Pessoa	08/11	26/11
Implementação do caso de uso: Manter Membro	Johnys Custódio Leonardo Pessoa	08/11	26/11
Implementação do caso de uso: Manter Serviço	Johnys Custódio Leonardo Pessoa	08/11	26/11
Implementação do caso de uso: Gerenciar Multusuários	Johnys Custódio Leonardo Pessoa	08/11	26/11
Implementação do caso de uso: Manter Artefato	Johnys Custódio Leonardo Pessoa	08/11	26/11
Implementação do caso de uso: Clonar Artefato	Johnys Custódio Leonardo Pessoa	08/11	-
Implementação do caso de uso: Manter Modelo	Johnys Custódio	08/11	26/11

	Leonardo Pessoa		
Implementação do caso de uso: Manter Log das Operações do Sistema	Johnys Custódio Leonardo Pessoa	08/11	-
3º Iteração			
Implementação do caso de uso: Preencher Artefato	Johnys Custódio Leonardo Pessoa	26/11	07/12
Implementação do caso de uso: Visualizar Artefato	Johnys Custódio Leonardo Pessoa	26/11	07/12
Implementação do caso de uso: Manter Documento	Johnys Custódio Leonardo Pessoa	26/11	-
Implementação do caso de uso: Gerenciar Serviço de Persistência	Johnys Custódio Leonardo Pessoa	26/11	-
Implementação do caso de uso: Gerenciar Serviço de Dependência	Johnys Custódio Leonardo Pessoa	26/11	-
Finalização do DMS	Johnys Custódio Leonardo Pessoa	06/12	07/12

Tabela 4 – Cronograma de atividades

3.3 Iterações do Projeto

Incremento	Previsão	Entrega
1ª iteração: Entendimento do negócio, levantamento e especificação de requisitos e regras de negócio, definição dos casos de uso.	03/06/2011	03/06/2011
2ª iteração: Implementação dos seguintes casos de uso: manter tipomembro, manter membro, manter Serviço, gerenciar multiusuários, manter artefato, montar artefato, clonar artefato, manter modelo, manter log de operações do sistema.	10/10/2011	08/11/2011
3ª iteração: Implementação dos seguintes casos de uso: preencher artefato, visualizar artefato, manter documento, gerenciar serviço de persistência, gerenciar serviço de validação de dados, imprimir artefato, gerenciar serviço de dependências, manter workflow.	26/11/2011	

Tabela 5 – Iterações do projeto

3.4 Relatório Semanal de Progresso

3.4.1 Relatório Semanal de Progresso n.º 01

Relatório Semanal de Progresso		N.º 01
Período	08/02/2011 à 11/02/2011	
Atividade Planejada	Reunião com os orientados para o entendimento do projeto. Leitura da documentação antiga do Quid-IDE. Verificação de possíveis melhorias do projeto. Elaboração de atas de reunião.	
Atividade Realizada	Reunião com os orientados para o entendimento do projeto. Leitura da documentação antiga do Quid-IDE. Verificação de possíveis melhorias do projeto. Elaboração de atas de reunião.	
Próximo Período	12/02/2011 à 23/02/2011	
Planejado para o próximo período	Definição dos Componentes da nova arquitetura.	

Tabela 6 - Relatório Semanal de Progresso n.º 01

3.4.2 Relatório Semanal de Progresso n.º 02

Relatório Semanal de Progresso		N.º 02
Período	12/02/2011 à 22/02/2011	
Atividade Planejada	Reunião com Orientador. Detalhamento da documentação do Quid-IDE, verificando pontos a serem melhorados. Proposição de novas idéias. Elaboração das Atas de Reunião.	
Atividade Realizada	Reunião com Orientador. Detalhamento da documentação do Quid-IDE, verificando pontos a serem melhorados. Proposição de novas idéias. Elaboração das Atas de Reunião.	
Próximo Período	23/02/2011 á 25/03/2011	
Planejado para o próximo período	Detalhamentos dos Componentes.	

Tabela 7 - Relatório Semanal de Progresso n.º 02

3.4.3 Relatório Semanal de Progresso n.º 03

Relatório Semanal de Progresso	N.º 03

Período	23/02/2011 á 25/02/2011
Atividade Planejada	Reunião com Orientador. Detalhamento dos componentes TipoMembro, Membro, Artefato e Dependência. Proposição de novas soluções. Confecção das Atas de Reunião.
Atividade Realizada	Reunião com Orientador. Detalhamento dos componentes TipoMembro, Membro, Artefato e Dependência. Proposição de novas soluções. Confecção das Atas de Reunião.
Próximo Período	26/02/2011 á 19/03/2011
Planejado para o próximo período	Definição da Montagem do Modelo.

Tabela 8 - Relatório Semanal de Progresso n.º 03**3.4.4 Relatório Semanal de Progresso n.º 04**

Relatório Semanal de Progresso		N.º 04
Período	26/02/2011 á 19/03/2011	
Atividade Planejada	Reunião com Orientador. Detalhamento dos componentes Modelo, ItemModelo. Especificação dos componentes: Artefatos, Dependência, Membro e Framework. Confecção das Atas de Reunião.	
Atividade Realizada	Reunião com Orientador. Detalhamento dos componentes Modelo, ItemModelo. Especificação dos componentes: Artefatos, Dependência, Membro e Framework. Confecção das Atas de Reunião	
Próximo Período	20/03/2011 á 16/04/2011	
Planejado para o próximo período	Correção das definições dos Componentes.	

Tabela 9 - Relatório Semanal de Progresso n.º 04**3.4.5 Relatório Semanal de Progresso n.º 05**

Relatório Semanal de Progresso		N.º 05
Período	26/02/2011 á 19/03/2011	
Atividade Planejada	Reunião com Orientador. Planejamento da Arquitetura do framework. Descrição do componente: Serviço.	

	Descrição dos serviços fornecidos pelo framework. Confecção das Atas de Reunião.
Atividade Realizada	Reunião com Orientador. Planejamento da Arquitetura do framework. Descrição do componente: Serviço. Descrição dos serviços fornecidos pelo framework. Confecção das Atas de Reunião.
Próximo Período	17/04/2011 á 21/05/2011
Planejado para o próximo período	Especificação dos Requisitos.

Tabela 10 - Relatório Semanal de Progresso n.º 05**3.4.6 Relatório Semanal de Progresso n.º 06**

Relatório Semanal de Progresso		N.º 06
Período	17/04/2011 á 21/05/2011	
Atividade Planejada	Reunião com Orientador. Especificação dos requisitos. Validação dos requisitos com o orientador. Confecção da documentação referente aos requisitos. Confecção dos documentos referentes à banca parcial. Confecção das Atas de reunião.	
Atividade Realizada	Reunião com Orientador. Especificação dos requisitos. Validação dos requisitos com o orientador. Confecção da documentação referente aos requisitos. Confecção dos documentos referentes a banca parcial. Confecção das Atas de reunião.	
Próximo Período	22/05/2011 á 20/06/2011	
Planejado para o próximo período	Confecção do DMS para banca parcial.	

Tabela 11 - Relatório Semanal de Progresso n.º 06**4 Modelagem do Negócio****4.1 Ata de Reunião****4.1.1 Ata n.º 01****Ata da Reunião n.º 01**

Data	Horário	Local						
08/02/2011	Das 20h00m às 21h00m	Casa do Leonardo Pessoa						
Equipe técnica	Johnys Custódio e Leonardo Pessoa							
Equipe usuária	Não se aplica.							
Teor desta reunião								
<p>A idéia iniciada por Jonathan em 2010 foi focar as responsabilidades (ações) na entidade Artefato, forçando assim a necessidade de um desenvolvedor para a criação de novos artefatos.</p> <p>Após este entendimento, os dois participantes da reunião entraram em consenso sobre generalizar os artefatos, ou seja, dar funções ao sistema para a criação e manipulação destes para que não dependessem de um desenvolvedor. Fazendo isto, o primeiro problema encontrado foi a persistência das informações: Como os dados seriam armazenados com artefatos genéricos?</p> <p>A idéia inicial encontrada foi descer um nível, ou seja, Artefatos são compostos por Membros, então os membros têm que ser responsáveis por suas ações. Algumas dúvidas surgiram neste ponto, pois pensando em generalizar os artefatos, foi especificado os membros, ou seja, ainda assim seria necessário um desenvolvedor para a criação de novos membros.</p> <p>Foi então pensado na criação de outra entidade para o projeto: TipoDeMembro. Esta então seria a nossa entidade mais baixa. Seria responsável por definir o tipo de dado, o tamanho máximo, modo de exibição da tela, etc. Membros seriam compostos por TipoDeMembro's.</p> <p>A entidade TipoDeMembro seria baseada em componentes HTML, sendo assim, disponibilizáramos os componentes já prontos, tornando assim os Artefatos e os Membros genéricos.</p> <p>Também surgiram algumas dúvidas:</p> <ol style="list-style-type: none"> 1. Por que o Jonathan focou em colocar as responsabilidades nos artefatos e não nos membros? Existe algum motivo que ainda não foi entendido? 								
Observação Importante								
<ul style="list-style-type: none"> - Técnica utilizada durante a reunião: Entrevista informal. - Contatos: <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">Johnys Custódio:</td> <td style="width: 33%;">email: jjohnys@gmail.com</td> <td style="width: 33%;">telefone: (62) 9339-4670</td> </tr> <tr> <td>Leonardo Pessoa:</td> <td>email: nardopessoa@gmail.com</td> <td>telefone: (62) 9123-7035</td> </tr> </table> 			Johnys Custódio:	email: jjohnys@gmail.com	telefone: (62) 9339-4670	Leonardo Pessoa:	email: nardopessoa@gmail.com	telefone: (62) 9123-7035
Johnys Custódio:	email: jjohnys@gmail.com	telefone: (62) 9339-4670						
Leonardo Pessoa:	email: nardopessoa@gmail.com	telefone: (62) 9123-7035						
Requisitos Registrados								

Tabela 12 – Ata da Reunião n.º 01**4.1.2 Ata n.º 02**

Ata da Reunião n.º 02		
Data	Horário	Local
12/02/2011	Das 14h20m às 16h20m	Casa do orientador Guiliano Rangel
Equipe técnica	Johnys Custódio, Leonardo Pessoa e Guiliano Rangel	
Equipe usuária	Não se aplica.	
Teor desta reunião		

Dado o inicio da reunião o Professor Guiliano nos mostrou o projeto como um todo, focando nas responsabilidades e abrangências. Após a demonstração, questionamos a necessidade do usuário desenvolvedor para a realização da confecção dos artefatos do projeto, este explicado pelo Guiliano que o projeto era um framework, e não um programa final.

Após discussão, foram levantados alguns pontos sobre o projeto:

- Utilização de Interfaces e superclasses para a confecção de artefatos, outros elementos do framework.
- Definir um descritor de tela (impressão / visualização) para poder imprimir um documento já preenchido. Este descritor de tela deverá cuidar de dependências entre os artefatos.
- Definir quais artefatos devem ser confeccionados de modo que se possa utilizar o framework para escrever o DMS atual (2011).
- O framework deverá manter diagramas de teste de arquitetura.
- Fixação de que o artefato se persiste de seu próprio jeito. Ele é responsável por proporcionar uma interface de preenchimento, visualização, e uma maneira para se persistir.
- Como objetivo final, o DMS deste projeto deverá ser impresso pelo framework.
- Tentar dividir a visão do controle do framework de modo que os artefatos sejam capazes de se: visualizar, preencher e imprimir.
- A impressão final poderá ser feita em PDF ou em LATEX.

Observação Importante

- Técnica utilizada durante a reunião: Entrevista informal.

- Contatos:

Johnys Custódio:	email: jjohnys@gmail.com	telefone: (62) 9339-4670
Leonardo Pessoa:	email: nardopessoa@gmail.com	telefone: (62) 9123-7035
Guiliano Rangel:	email: guiliano@gmail.com	telefone:

Requisitos Registrados

Tabela 13 – Ata da Reunião n.º 02

4.1.3 Ata n.º 03

Ata da Reunião n.º 03				
Data	Horário	Local		
23/02/2011	Das 21h00m às 22h00m	Universidade Estadual de Goiás – Biblioteca		
Equipe técnica	Johnys Custódio e Leonardo Pessoa			
Equipe usuária	Não se aplica.			
Teor desta reunião				
Após primeiros esclarecimentos, os alunos Leonardo e Johnys decidiram dividir o projeto em quatro etapas (fases): <ul style="list-style-type: none"> • Criação de Artefatos; • Montagem do Modelo de Documento; • Preenchimento do Modelo; • Impressão do trabalho pronto. 				

Ao partirmos para a primeira etapa (criação), decidiu-se criar responsabilidades para cada componente do framework, ficando por base a divisão abaixo:

TipoMembro (definido na Ata n.º 02 como TipoDeMembro)

- Apresentação no browser (input, textarea, label, etc.);
- Tipo de dado para a persistência (seja ela qual for);
- Domínio e máscara.

Membro

- Obrigatoriedade, visibilidade;
- Formatação de apresentação (impressão e tela).
- Componente envolvido na reutilização de componentes.

Artefato

- Mostrar preenchido (HTML e impressão);
- Se persistir (a partir do tipo de dado definido no membro).

Dependência

- Relacionar os artefatos (quando houver necessidade)
- Ao se criar uma dependência entre artefatos, o artefato carrega essa dependência por toda sua vida no software.
- Premissa para criação de dependência: Os artefatos dependentes já deverão estar construídos.
- Quando um artefato pai fica responsável pela criação de outro artefato filho, (exemplo: Ata de Reunião gerar Requisitos), não será possível o preenchimento de um artefato filho antes do artefato pai.

Ao estudar as dependências, verificou que elas podem ser divididas em duas classes:

- Dependência de Preenchimento e Dependência de Criação.

Para melhor contextualizar, tomemos o exemplo de três artefatos de um projeto: Ata, Requisito e Regra de Negocio.

Supomos que o preenchimento de uma Ata geraria Requisitos e os requisitos são compostos de regras de negócios. A dependência de Preenchimento pode ser explicada pelos requisitos que em seu conteúdo contém regras de negócio. A dependência de criação seria feita ao modo que, quando se cria uma Ata, dela se criaria Requisitos que seriam posteriormente preenchidos. Ou seja, o Artefato Ata obrigaria a criação de artefatos Requisitos de acordo com a proposta do componente dependência.

Ao partir da idéia de criação de artefatos, ficou uma dúvida: O projeto do Quid-IDE deixaria de ser um framework e se transformaria em um projeto final?

Pensamento Individual: Johnys

Na volta para casa, pensei que a parte de criação do artefato, poderia ser tratada como se fosse um componente externo ao framework (um plugin). Este componente faria que a responsabilidade do usuário desenvolvedor ficasse somente na criação do componente TipoMembro. E a confecção do artefato poderia ser feito pelo usuário que monta o documento.

Próxima Reunião:

- Definir melhor a dependência entre os artefatos;
- Esclarecer com o orientador qual proposta devemos seguir em relação ao Framework;
- Dada a proposta, começar a confecção de exemplos de teste.

Observação Importante

- Técnica utilizada durante a reunião: Entrevista informal.

- Contatos:

Johnys Custódio:	email: jjohnys@gmail.com	telefone: (62) 9339-4670
Leonardo Pessoa:	email: nardopessoa@gmail.com	telefone: (62) 9123-7035

Requisitos Registrados

--

Tabela 14 – Ata da Reunião n.º 03

4.1.4 Ata n.º 04

Ata da Reunião n.º 04				
Data	Horário	Local		
26/02/2011	Das 18h00m às 22h10m	Universidade Estadual de Goiás - Biblioteca		
Equipe técnica	Johnys Custódio e Leonardo Pessoa			
Equipe usuária	Não se aplica.			
Teor desta reunião				
Após discussão de alguns assuntos foram retiradas as seguintes proposições:				
<p>Propriedades do TipoMembro</p> <ul style="list-style-type: none"> * Apresentação no Browser: <ul style="list-style-type: none"> • Esta apresentação deve ter no mínimo: imagem ilustrativa, nome do componente e um texto descritivo sobre o que o TipoMembro faz. * Tipo de Dado para Persistência: <ul style="list-style-type: none"> • O TipoMembro deverá ser capaz de dizer qual o tipo de dado ele possui para ser armazenado (varchar, boolean, integer, etc.). • TipoMembro Estático não terão coluna nos artefatos que o utilizam (ver Membro estático abaixo). * Domínio e Máscara <ul style="list-style-type: none"> • O Membro deve ser capaz de informar um tipo padrão de entrada (máscara) caso ele precise, não ficando a parte dele criar esta função, e sim do descritor de tela. Para solucionar este problema pensou-se em definir que o membro fale em qual evento a máscara deve ser inserida, por exemplo, ao se digitar (onKeyPress) ou quando o componente perder o foco (onExit), neste modo, o descritor de tela, saberá o momento de aplicar uma máscara ao componente. Esta seria definida em um padrão comum independente do descritor de tela, que ficaria responsável apenas pela interpretação e execução. • O TipoMembro deverá ser capaz de validar uma entrada de dados. Esta validação é feita a partir de um método padrão que recebe um texto (String) e retorna um valor verdade ou falso (boolean), informando se a entrada é válida ou não. 				
<p>Propriedades do Membro</p> <ul style="list-style-type: none"> * Obrigatoriedade e visibilidade: <ul style="list-style-type: none"> • Deverá informar ao framework se ele é de preenchimento obrigatório ou não, ficando por parte do framework cobrar esta obrigatoriedade do usuário que está preenchendo o documento. • Caso o Membro seja Estático, ou seja, seu texto é único pra todo o software. Ele deve ter base em um TipoMembro Estático. * Característica de pegar o valor já existente no Membro e atribuir um valor preenchido pelo 				

usuário (get / set value);

* Ordem de preenchimento na tela.

- Deverá ser capaz de informar qual a ordem de preenchimento do mesmo no artefato (table order), de modo que possa ser gerenciado pela interface de preenchimento.

Propriedades do Artefato

* Para melhor reutilização de textos, as características de localização serão armazenadas na tabela de ligação entre membro e artefato.

* Outras propriedades dos artefatos serão discutidas nas próximas atas.

Serviços disponibilizados pelo framework

- * Dependência;
- * Persistência;
- * Validação de artefatos com suas tabelas no banco;

Características que ainda faltam:

* Ao ser instalado um TipoMembro, este deverá informar ao framework sua existência, e este deve gravar esta informação para posterior criação do "Descriptor de Tela".

Problemas a serem solucionados

Problema 01:

- Para separar a visão do controle, devemos manter uma interface entre os dois, de modo que a visão conheça o controle, porém o controle seja independente da interface.

Possíveis soluções dos problemas

Solução Problema 01:

- Os componentes devem ser divididos em duas camadas: visão e controle;
- A visão seria compartilhada com o descritor de tela.
- O controle seria independente da visão, disponibilizando informações suficientes de modo que se possa criar uma nova visão a partir dele.

Informações pertinentes ao Descritor de Tela

* Ao se abrir o descritor de tela, ele deve perguntar ao framework quais artefatos e serviços estão disponíveis para a confecção de um novo artefato. Ao conseguir as listas, o descritor de tela obtém a imagem, o nome e a descrição de apresentação de cada um e monta uma interface para que o usuário montador possa selecionar e montar o novo artefato. Após isso, o descritor deve ser capaz de mostrar em um local, todos os artefatos que o artefato que está sendo construído pode fazer uma dependência.

* O descritor de tela deve solicitar ao framework quais os tipos de dependências que podem ser estabelecidas entre o artefato que está sendo construído e os demais artefatos.

* O descritor de tela deve ser capaz de interpretar tipos de funções de máscaras para que os componentes possam utilizar, de acordo com o estabelecido na descrição do TipoMembro.

* Ao ser selecionado um componente, o descritor de tela deve ser capaz de perguntar ao componente quais são suas propriedades e, de posse delas, criar uma interface para que o usuário possa preenchê-las e posteriormente invocar o serviço de armazenamento. Estas propriedades podem ser valores (exemplo: escolhas de um combobox, ou radiobutton, etc.), ou algo que corresponda à confecção de um novo Membro.

* Para cada componente adicionado ao artefato, o descritor de tela deve dar a ele um nome

variável (que possa ser modificada pelo usuário) este nome será a identificação do componente dentro do artefato e posteriormente será utilizado pelo descritor de tela para a confecção da tabela do artefato.

- * Deve ser de responsabilidade do componente, a maneira no qual ele receberá as propriedades preenchidas e atribuí-las no componente visual.

- * O descritor de tela deve ser capaz de capturar a resolução da tela e requisitar ao framework um tamanho para a apresentação (máximo ou mínimo)

- * Na montagem documento o alinhamento dos membros na horizontal será baseado em linhas, para que não fique desorganizado (desalinhado).

- * O descritor de tela deve ser uma interface expansível, ou seja, os membros poderão adicionar novos componentes (funções) à ela.

- * Quando o usuário quiser adicionar uma dependência entre os artefatos, esta dependência deve ser gerenciada a partir de componentes específicos. Estes componentes deverão ser capazes de mostrar ao descritor de tela a maneira que elas devem ser tratadas (um novo campo na tabela, uma nova tabela associativa entre os componentes, etc.). Esta dependência deve posteriormente informar ao framework de modo que ele possa gerenciá-la no momento de preenchimento do Modelo de Documento.

Criação de um novo Artefato

- * Quando o descritor de tela for criar o novo artefato ele deverá seguir alguns passos:

- Perguntará aos componentes adicionados, quais deles devem ter uma coluna na tabela que será criada para artefato.

- Perguntará as dependências adicionadas, quais delas são colunas na tabela que será criada para artefato.

- Com base nos membros que serão adicionados na tabela, ele perguntará ao TipoMembro qual seu tipo de dado de armazenamento (varchar, integer, etc.), e solicitará ao membro qual o seu nome de identificação.

- Framework deve validar a duplicidade de artefatos (comparação do identificador do artefato).

- Após ter essas informações, solicitará ao framework (serviço de persistência) a criação de uma tabela ou alguma outra forma de armazenamento com tais campos e tais dados. O nome da tabela será o nome do identificador único do artefato.

OBS.: Toda tabela criada no banco deverá conter um coluna de identificação única.

- Após a criação da tabela, o descritor de tela solicitará ao framework o registro do novo artefato (geral). Neste registro, serão armazenados os membros estáticos e demais informações correspondentes aos membros.

- Após o registro do artefato, o descritor solicitará às dependências quais delas desejam ter novas tabelas no banco. Estas tabelas vão ser criadas com nomes padrões (exemplo: nome1Artefato_nome2Artefato), que indicarão possíveis relacionamentos.

- Após a definição das tabelas e relacionamentos, o descritor de tela solicitará ao framework a criação destas tabelas e o registro das dependências, para que elas possam ser gerenciadas posteriormente.

Informações referentes a comportamentos de edição:

- * O framework só poderá permitir modificações de artefatos que não estejam sendo usados por algum modelo.

- * Se o usuário quiser aproveitar parte do artefato, o descritor de tela possibilitará a criação de uma copia do artefato (clonagem), e esta poderá ser editada. Esta copia terá todos os componentes e

dependências, porem não terá o mesmo nome e o texto descritivo, ficando a parte do descritor solicitar tais novas informações ao usuário.

Informações para requisitos futuros.

* Montagem do Modelo de DMS.

- Quando um determinado artefato for adicionado ao modelo, ele deverá saber quais dependências ele possui, de modo que o framework possa gerenciá-las e cobrá-las do usuário que esta montando o modelo.

Observação Importante

- Técnica utilizada durante a reunião: Entrevista informal.

- Contatos:

Johnys Custódio:	email: jjohnys@gmail.com	telefone: (62) 9339-4670
Leonardo Pessoa:	email: nardopessoa@gmail.com	telefone: (62) 9123-7035

Requisitos Registrados

Tabela 15 – Ata da Reunião n.º 04

4.1.5 Ata n.º 05

Ata da Reunião n.º 05		
Data	Horário	Local
19/03/2011	Das 14h00m às 15h30m	Casa do orientador Guiliano Rangel
Equipe técnica	Johnys Custódio, Leonardo Pessoa e Guiliano Rangel	
Equipe usuária	Não se aplica.	

Teor desta reunião

Neste momento o pensamento começou a envolver Modelo, Árvore e Seqüência de Preenchimento. A princípio surgiram alguns problemas quando colocou-se em pauta a situação da seqüência de preenchimento. Onde ficariam guardados estes dados? Na dependência entre artefatos, quais partes de um artefato deve ser preenchida no ato da sua criação?

Pensou-se em fazer dois tipos de obrigatoriedade:

- Obrigação de preenchimento.
- Obrigação de criação (levando em conta que um "item" obrigatório de criação também é de obrigação de preenchimento).

Pensou-se em utilizar o atributo título (ver item 2.1 desta ata), para se solucionar a dependência de criação ou utilizar um serviço para controlar essa obrigação.

Pensou-se em unificar a classe dependência, pare que um artefato seja composto por uma lista de dependências, cada dependência apontaria para outro artefato ou para um membro.

Acrônimos

- Grau: nível do artefato na arvore modelo.
- Ordem: dado dois ou mais artefatos de mesmo grau, a ordem garante seqüência dos mesmos.

1 - Definição da classe ItemModelo:

1.1 - Para solucionarmos o problema da montagem da árvore do Modelo, criamos a classe ItemModelo que será composta pelos Artefatos envolvidos no modelo.

1.2 - Nela serão gravadas as informações:

- 1.2.1 - Artefato que representa.
- 1.2.2 - O grau do artefato na árvore do modelo.
- 1.2.3 - Ordem de aparecimento na árvore do modelo.
- 1.2.4 - Multiplicidade (quantidades de vezes que se pode repetir o Artefato dentro do modelo (uma vez ou muitas).

1.2.5 - Ordem de Preenchimento.

1.2.5.1 - A ordem de preenchimento informará ao framework qual deverá ser a ordem de preenchimento dos Artefatos no Modelo.

1.2.5.2 - Artefatos que podem ser preenchidos simultaneamente devem ter a mesma ordem de preenchimento.

1.2.5.3 - O framework permitirá o preenchimento de um artefato desde que todos os artefatos do modelo com ordem menor já estejam preenchidos.

Obs.: Estes artefatos de ordem menor devem ter correlação entre si. Por exemplo: uma Ata, um requisito, um Caso de Uso, caso criemos uma ata que gere um requisito em específico e posteriormente um caso de uso, esta ordem deverá ser seguida. Porém, se criar duas Atas que gerem dois requisitos cada, não necessariamente teremos que preencher todos estes requisitos para fazer um caso de uso.

1.2.5.4- Artefatos que são auto-criados (listas, índices, etc.) não terão ordem de preenchimento, e consequentemente não serão disponibilizados para o preenchimento.

1.3 - Ao se adicionar um artefato no modelo, verifica-se se ele contém uma dependência com outros artefatos. Caso exista a dependência, o framework solicitará que o usuário montador do Modelo utilize os outros artefatos dependentes.

1.4 - Um modelo só poderá ser alterado somente se não estiverem sendo utilizado por nenhum projeto.

2- Características dos Artefatos:

2.1 - Para solucionar o nome de identificação do artefato na TreeView (visualização da árvore), todo artefato deverá conter um "título", ou seja um identificador único que o diferencie dos demais artefatos semelhantes a ele.

2.1.1 - Esse título deverá ser fixo e o framework obrigará o artefato a tê-lo.

2.2 - O artefato deve conhecer todos os TipoMembro's que ele se compõem.

3 - Definição da Classe Dependência (será controlada por uma classe a parte do artefato)

3.1 - Seria a classe genérica de controle de dependência. Nela conteria as seguintes informações:

3.1.1 - Membro que representará a dependência.

3.1.2 - Nome e descrição.

3.1.4 - Seria regida por uma interface padrão (pois haverá especialização de dependências).

3.1.5 - Em cada dependência teria:

3.1.5.1- O artefato dependente (Artefato filho);

3.1.5.2- O tipo de dependência realizada (criação, composição, etc.);

3.1.5.3- O artefato que esta recebendo a dependência (Artefato pai);

3.2 - Especialização da Classe Dependência:

3.2.0 - O framework deve solucionar o problema de dependência circular, pai tem dependência com filho e filho tem a mesma dependência com o pai.

3.2.1 - Dependência de Criação (Extends Dependência)

3.2.1.1- Seria capaz de solucionar a dependência de quando um artefato obriga a criação de outro artefato.

3.2.1.2- Ela seria composta por um Membro “lista de preenchimento” com entrada de texto.

3.2.1.3- Cada entrada seria o título do artefato a ser criado.

3.2.1.4- Ao se salvar esta lista, seriam criados os artefatos dependentes dele.

3.2.1.5- Este tipo de dependência só poderá ser feita com um artefato de multiplicidade N (muitas).

3.2.1.6- Esta dependência deverá dar opções de seleção para que se escolha um filho já criado. O framework tem que validar a duplicidade de dados, informar o

usuário e liberar para que ele escolha o artefato que já está armazenado.

3.2.2 - Dependência de Composição (Extends Dependência)

3.2.2.1- Seria capaz de solucionar a dependência de quando um artefato contém um membro que é composto de um campo de outro artefato.

3.2.2.2- Ela seria composta por um Membro “lista de preenchimento” com selecionador de um comboBox;

3.2.2.3- O usuário montador poderá escolher qual membro do artefato provedor será mostrado no combo de seleção.

3.2.2.4- Ela permitirá definir um limite de quantos elementos poderão ser selecionados.

3.2.2.5- Ela permitirá definir se o campo é obrigatório, entre outras características de preenchimento.

3.2.3 - Dependência de Listagem (Extends Dependência).

3.2.3.1 - Seria capaz de solucionar a dependência de quando um artefato necessita de uma lista de um determinado artefato. Exemplo: Um artefato Lista de Regra de Negócio que necessita de todas as regras de negócio preenchidas no sistema;

3.2.3.2 - Esta dependência seria composta de um Membro “lista de campos” de artefato;

3.2.3.3 - Ela seria uma dependência que poderia ser adicionada em artefatos auto montáveis, ou seja, seu preenchimento tem origem em outros artefatos;

3.2.4 - Dependência de Ordem de Preenchimento (Extends Dependência)

3.2.4.1 - Seria capaz de solucionar a obrigatoriedade entre artefatos, ou seja, um artefato não pode ser preenchido antes de outro;

3.2.4.2 - Esta dependência resolveria questões de preenchimento onde um artefato não depende de outro, forçando o usuário a seguir a ordem definida pelo usuário montador;

3.2.4.3 - Seria composto por um Membro “bloqueador de preenchimento” ou “mensagem de erro” (a definir).

4- Características Membro

4.1 - O atributo obrigatoriedade de um membro define se ele é de preenchimento obrigatório ou não durante o preenchimento do artefato em específico.

4.2 - Deste modo, a dependência de criação pode criar um novo artefato, preenchendo-lhe somente o nome de identificação.

5 - Serviços do Framework

5.1 - A dependência será um serviço do framework, totalmente independente e reutilizável por outros serviços.

5.2 - O framework será capaz de gerenciar serviços. Estes serviços serão especializados em eventos e ações, tais como dependências.

5.3 - O framework será capaz de realizar o gerenciamento de Ações / Eventos.

5.4 - Como o problema proposto está centralizado no artefato, este utilizará os serviços oferecidos pelo framework, como persistência e dependência já apresentados.

6 - Dúvidas para a próxima ata

6.1 - Como o montador de tela vai montar o formulário de preenchimento?

6.2 - Esses dados serão passados via XML?

6.3 - Quais os métodos principais que deve ter a interface do controle do framework?

6.4 - Como ficariam divididas as responsabilidades dentro do framework? Tudo controlado pelo IControle?

6.5 - Como faremos essa interface ser expansível à medida que se adiciona novas dependências e novos artefatos?

6.6 - Quem controlaria as instâncias dos objetos na interface? Ela solicitará ao framework como um FactoryObject?

6.7 - Criar um protótipo de arquitetura para framework no padrão multicamadas.

Observação Importante

- Técnica utilizada durante a reunião: Entrevista informal.

- Contatos:		
Johnys Custódio:	email: jjohnys@gmail.com	telefone: (62) 9339-4670
Leonardo Pessoa:	email: nardopessoa@gmail.com	telefone: (62) 9123-7035
Guiliano Rangel:	email: guiliano@gmail.com	telefone:
Requisitos Registrados		

Tabela 16 – Ata da Reunião n.º 05

4.1.6 Ata n.º 06

Ata da Reunião n.º 06				
Data	Horário	Local		
16/04/2011	Das 14h30m às 15h45m	Casa do orientador Guiliano Rangel		
Equipe técnica	Johnys Custódio, Leonardo Pessoa e Guiliano Rangel			
Equipe usuária	Não se aplica.			
Teor desta reunião				
<p>Neste momento do projeto, começou-se a pensar em como os elementos do framework iriam se comunicar. Baseando no modelo arquitetural MVC (model-vision-control) e na proposta atual, fez-se um pouco de confusão entre Artefato e Controle. O artefato poderia ser usado apenas como estrutura e transporte de dados, sendo assim o controle responsável pelas chamadas aos serviços ou o artefato executaria os serviços vinculados a ele sem a necessidade do controle conhecer os serviços. O orientador explicou a situação dizendo que o software final seria composto por dois projetos: um Framework e uma Aplicação que utilizará o framework, concluindo que o artefato possui características de um controle, porém do lado do framework.</p> <p>Disto, foram acertadas as camadas do software:</p> <ul style="list-style-type: none"> - Visão - Descritor de Tela (Aplicação) - Controle da aplicação (Aplicação) - Controle do framework - Artefato (Framework) - Camada de extensão - Serviços e TipoMembros (Framework) <p>Fixado esta idéia, levantou-se a dúvida de como os serviços seriam executados, pois alguns serviços necessitam que outros sejam executados previamente, por exemplo, o serviço de persistência não pode ser executado antes do serviço de dependência. Como solução, propôs que o artefato possuirá uma lista ordenada de serviços, ou seja, uma idéia de fila, onde um serviço só pode ser executado ao término de outro. Caso seja possível que dois serviços executem paralelamente, eles terão o mesmo número de ordem.</p> <p>O esboço da arquitetura foi apresentado ao orientador, alguns pontos foram levantados, porém decidiu-se que a arquitetura deverá ser finalizada após a definição de requisitos, tornando este um assunto para as próximas reuniões.</p> <p>Para apresentação da banca parcial o orientador exigiu que todos os requisitos estejam bem especificados. O entendimento do projeto é primordial para a prova de conceito. Por ser um framework, não será cobrado a implementação da primeira iteração do projeto.</p>				
Observação Importante				
<ul style="list-style-type: none"> - Técnica utilizada durante a reunião: Entrevista informal. - Contatos: 				

Johnys Custódio:	email: jjohnys@gmail.com	telefone: (62) 9339-4670
Leonardo Pessoa:	email: nardopessoa@gmail.com	telefone: (62) 9123-7035
Guiliano Rangel:	email: guiliano@gmail.com	telefone:
Requisitos Registrados		

Tabela 17 – Ata da Reunião n.º 06

4.1.7 Ata n.º 07

Ata da Reunião n.º 07		
Data	Horário	Local
21/05/2011	Das 14h13m às 15h45m	Casa do orientador Guiliano Rangel
Equipe técnica	Johnys Custódio, Leonardo Pessoa e Guiliano Rangel	
Equipe usuária	Não se aplica.	
Teor desta reunião		
<p>Dado o início da reunião, Johnys expôs a definição do requisito TipoMembro para ser verificado com o orientador Guiliano. Desta verificação, foram levantadas as seguintes alterações:</p> <ul style="list-style-type: none"> 1- O TipoMembro terá duas divisões lógicas: Visão e Modelo, sendo que o modelo é independente da visão. Para um determinado modelo poderá existir diversas visões relacionada com seu descritor de tela. 2- Definir quais são os eventos em que a máscara deve ser inserido. Tais eventos devem estar correlacionados com o descritor de tela da visão em específico. 3- Dar mais exemplos de TipoMembro estático, melhorando sua definição (nome da faculdade, nome do orientando, data). 4- Os TipoMembro deverão ter um controle de versionamento dividido em dois níveis. <ul style="list-style-type: none"> a. Nível 1 (Versão): Nível que garante a compatibilidade de um TipoMembro. Um mesmo TipoMembro, que contém a <i>versão</i> diferente é encarado pelo framework como um novo TipoMembro. b. Nível 2 (Revisão): Garante a compatibilidade com o TipoMembro anterior. Este tipo de versão é utilizado quando são feitas alterações menores no TipoMembro (correções de bugs, manutenções adaptativas, etc). Com <i>revisões</i> iguais, o framework fará uma atualização do TipoMembro cadastrado no sistema. 5- A remoção de um TipoMembro poderá ocorrer de duas maneiras: <ul style="list-style-type: none"> a. Durante o mapeamento dos TipoMembro cadastrados no framework: Nesta maneira, o framework verificará os arquivos do TipoMembro no diretório com as informações cadastradas no framework. b. Através de uma funcionalidade fornecida pelo framework. <p>Nos dois casos, o framework verificará se existe algum componente utilizando o TipoMembro. Caso exista, o framework informará ao usuário onde este TipoMembro está sendo utilizado, caso contrário excluirá o cadastro do TipoMembro do framework.</p> <ul style="list-style-type: none"> 6- A definição das funcionalidades dos TipoMembro específicas do framework deverão ser informadas na especificação de classes, junto ao detalhamento da arquitetura do software. <p>Após a verificação do TipoMembro, começamos a verificar a especificação do Membro, dela</p>		

foram levantadas as seguintes observações:

- 1- Mudança da característica do Membro. Anteriormente víamos o membro como uma classe separada, o que limitava as características que poderiam ser inferidas sobre o TipoMembro (formatação). Com a nova modelagem, o Membro passa a ser uma representação do TipoMembro, que traz as suas propriedades já previamente preenchidas. Com isso, o membro não é mais especificado no diagrama de classes, e somente no modelo entidade relacional.
- 2- Os requisitos associados ao Membro continuam, porém a sua especificação de requisitos deve contemplar essa nova abordagem.

Após a verificação do Membro, Leonardo expôs a especificação dos serviços e da dependência. Durante a leitura foram acrescentadas as seguintes características:

- 1- O serviço genérico deve ter o TipoMembro que o representará na visão, não sendo esta uma característica obrigatória.
- 2- O serviço deve ser capaz de fornecer sua lista de parâmetros de entradas para a execução de sua ação. Esta lista deve ser compatível com o resultado da ação, tornando possível utilizar a ação de um serviço como entrada de outro serviço.
- 3- Para melhor padronização do serviço, será permitida apenas uma ação para cada serviço, ou seja, uma única responsabilidade.

Para próxima reunião:

Trazer os requisitos Membro, Artefato, Modelo, Modelo e demais serviços especificados.

Verificar como será feita a execução dos serviços dentro de um artefato.

Verificar como será definido o workflow do Modelo.

Observação Importante

- Técnica utilizada durante a reunião: Entrevista informal.

- Contatos:

Johnys Custódio:	email: jjohnys@gmail.com	telefone: (62) 9339-4670
Leonardo Pessoa:	email: nardopessoa@gmail.com	telefone: (62) 9123-7035
Guiliano Rangel:	email: guiliano@gmail.com	telefone:

Requisitos Registrados

Tabela 18 – Ata da Reunião n.º 07

4.1.8 Ata n.º 08

Ata da Reunião n.º 08				
Data	Horário	Local		
27/05/2011	Das 19h30m às 21h30m	UEG - UnUCET		
Equipe técnica	Johnys Custódio, Leonardo Pessoa			
Equipe usuária	Não se aplica.			
Teor desta reunião				
Dado o inicio da reunião, foram discutidos como os atores se relacionariam com os documentos. Disto foi levantado que, com freqüência, um projeto pode ser regido por um grupo, ou seja, várias				

pessoas criando e atualizando os documentos simultaneamente. Com isso o sistema deve possibilitar a criação de grupos para os usuários e permitir que apenas um usuário por grupo seja responsável por criar documentos e definir quais usuários farão parte desse grupo. Isto levou a criação de um novo perfil de usuário, o *Gerente de Documentos*. Somente este perfil de usuário poderá criar um documento, porém a inclusão de novos artefatos poderá ser feita por usuários *Preenchedores*. Como recurso de privacidade, o gerente pode definir se os documentos gerados serão públicos (acessado por qualquer usuário do sistema) ou se serão privados (acessado somente pelos usuários do grupo envolvido no documento), porém a edição dos artefatos envolvidos só deve ser feita por integrantes do grupo.

Visto que muitos usuários atuarão sobre os mesmos artefatos, percebe-se o problema de edição simultânea. Como solução, o framework deve permitir que apenas um usuário tenha acesso à edição do artefato por vez, bloqueando os demais acessos. Ao se tentar acessar o artefato em edição o framework deve exibir quem está editando o artefato. Ele deve oferecer também o cálculo do tempo de inatividade em relação ao artefato, permitindo que com tempo de inatividade maior que 30 minutos, outro usuário pode desbloquear o artefato e iniciar sua edição.

Ao acessar o sistema o usuário terá interfaces referentes a cada perfil, sendo possível selecionar que tipo de perfil ele deseja acessar, mantendo somente as funcionalidades possíveis de cada perfil disponíveis e tornando a interface mais limpa.

O sistema deve manter em log de texto (TXT) todas as ações do usuário perante o sistema classificadas por perfil de usuário e dia da ação. Deve-se permitir ativar e desativar os logs do sistema para os perfis de gerente e preenchedor. Para um controle mais seguro da estrutura do documento, o sistema não permitirá o desativamento dos logs referentes aos perfis de montador e desenvolvedor, tornando possível a identificação do responsável de cada alteração, inserção ou exclusão na estrutura de documento ou componente utilizado.

Observação Importante

- Técnica utilizada durante a reunião: Entrevista informal.

- Contatos:

Johnys Custódio:	email: jjohnys@gmail.com	telefone: (62) 9339-4670
Leonardo Pessoa:	email: nardopessoa@gmail.com	telefone: (62) 9123-7035

Requisitos Registrados

Tabela 19 – Ata da Reunião n.º 08

4.1.9 Ata n.º 09

Ata da Reunião n.º 09				
Data	Horário	Local		
10/08/2011	Das 20h30m às 21h30m	UEG - UnUCET		
Equipe técnica	Johnys Custódio, Guiliano Rangel			
Equipe usuária	Não se aplica.			
Teor desta reunião				
Dado o inicio da reunião, Johnys esboçou os problemas encontrados na confecção do framework.				
1- Onde colocar a forma dos arquivos do TipoMembro, a maneira como é feito o espelho dos arquivos e a forma que o versionamento é feito?				
Todas essas informações da forma dos arquivos e o espelho devem aparecer na especificação de requisitos. O versionamento deve ser inserido no anexo da documentação, podendo ser feito através de				

um diagrama de atividade ou texto escrito.

2- O framework deve disparar exceções na execução de suas rotinas?

Não, por causa da camada intermediaria que existe entre a aplicação e o framework (Ex um WebService, uma arquitetura swing).

A solução proposta é que toda execução do framework deve retornar algo para a camada superior, de modo que ela saiba identificar se ouve algum "problema" na execução, e de posse desse conhecimento, solicitar ao framework o que aconteceu de errado.

3- Problema da falta do arquivo TipoMembro no diretório.

3.1 Primeiro Caso: Existe versão / revisão antiga do arquivo.

O framework deve realizar o rastreamento do TipoMembro, e informar ao usuário em quais Artefatos ele estava sendo utilizado. Após isso ele deve solicitar ao usuário se ele deseja voltar uma versão / revisão antiga. Caso deseje, ele voltará a versão / revisão do arquivo, se não irá inativar todos os modelos que o TipoMembro estava sendo utilizado.

3.2 Segundo Caso: Não existe versão / revisão antiga do arquivo.

O framework deve realizar o rastreamento do TipoMembro e informar ao usuário em quais artefatos ele estava sendo utilizado. Após isso o usuário poderá escolher realizar duas operações:

- 1 - Informar o arquivo do TipoMembro que falta no framework.
- 2 - Bloquear todos os modelos em que o TipoMembro estava sendo utilizado.

4- Especificar como será feito o mapeamento do TipoMembro-Modelo para cada visão específica, partindo da premissa que existe vários TipoMembro-Visão para um determinado TipoMembro-Modelo. (Utilizar a idéia do descritor de tela)

5 - Forma de Persistência:

5.1 - O artefato será um gerenciador do serviço de persistência, sendo que o serviço atuara sobre os Membros de um artefato. Logo não será o artefato que irá se persistir, e sim os Membros que estão dentro dele.

5.2 - O Membro poderá se associar a serviços de persistência de modo que o serviço especifique o tipo de persistência que será realizado (Arquivo / Banco de dados). (Mudança da maneira que os serviços são gerenciados: Os serviços se associam aos membros e são gerenciados pelo artefato como um serviço geral).

5.3 - Para o armazenamento dos parâmetros dos TipoMembro, o próprio TipoMembro ,quando for mapeado pelo framework, deverá criar a sua estrutura de armazenamento de seus parâmetros, podendo ele utilizar o serviço de persistência para isso, isso deixa mais flexível a maneira que os TiposMembros serão parametrizados. O Membro ainda continua sendo a instância do TipoMembro com seus parâmetros definidos.

O TipoMembro-Visão será responsável por fornecer uma interface de preenchimento dos parâmetros do TipoMembro-Modelo, mas será o TipoMembro-Modelo que será responsável pelo armazenamento de tais informações.

6- Existirá o serviço de validação (CPF, CNPJ, etc.), que poderá ser relacionado com os membros de um framework.

7- Alterações:

No diagrama de atividade, deve existir somente um ponto de inicio e um ponto final da execução das atividades, pois o framework sempre deverá dar um retorno da execução de uma determinada ação, até os fluxos alternativos.

No framework não existira fluxos de exceção, pelo fato da forma que eles devem ser tratados.
Observação Importante
- Técnica utilizada durante a reunião: Entrevista informal. - Contatos: Johnys Custódio: email: jjohnys@gmail.com telefone: (62) 9339-4670 Guiliano Rangel: email: guiliano@gmail.com telefone:
Requisitos Registrados

Tabela 20 – Ata da Reunião n.º 09

4.1.10 Ata n.º 10

Ata da Reunião n.º 10				
Data	Horário	Local		
13/08/2011	Das 14h30m às 15h30m	Casa do orientador Guiliano Rangel		
Equipe técnica	Johnys Custódio, Leonardo Pessoa e Guiliano Rangel			
Equipe usuária	Não se aplica.			
Teor desta reunião				
<p>Dada a orientação as dúvidas sobre a forma de persistência ainda foram levantadas. Em reuniões anteriores já haviam definido que a persistência estaria diretamente com o artefato, definido pelo usuário como o artefato se persistiria, voltando a responsabilidade do tipo de dado a ser persistido para o TipoMembro. Esta idéia foi mantida, contrariando a discussão da ata anterior. Caso haja a necessidade de que dentro de um artefato X um TipoMembro ser persista diferente, deve-se criar um novo artefato para isto e assim utilizar o serviço de dependência para que o mesmo seja persistido diferente.</p> <p>Em seguida iniciou-se a validação dos requisitos por base nos protótipos criados. O primeiro debatido foi o requisito Manter TipoMembro ação mapear TipoMembro. Foi abordado a necessidade de se importar arquivos TipoMembro-Visão para TipoMembro-Modelo já existente e também e importar vários TipoMembro-Visão para o único TipoMembro-Modelo ao mesmo tempo. Para maior facilidade na importação, será exigido dois arquivos com extensão JAR. Um contendo os arquivos do TipoMembro-Visão de uma visão específica e outro contendo os arquivos do TipoMembro-Modelo. Os JAR's devem seguir o padrão definido pela Oracle, contendo a estrutura de pacotes correta. No momento de importação do TipoMembro o formato será validado. Pelo menos uma das classes importadas deverá herdar a classe representante de um TipoMembro, pois só assim o framework conseguirá reconhecê-lo como um componente.</p> <p>O atributo nome do TipoMembro (visão/modelo) será seu identificador único no framework para a verificação de espelhamento do mesmo. Para a diferenciação da plataforma de cada TipoMembro-Visão, ele deverá informar o nome da sua plataforma para a qual ele foi desenvolvido. O usuário desenvolvedor terá acesso a este nome caso necessite criar novos TipoMembro-Visão, pois o descritor de tela só reconhecerá TipoMembro de sua plataforma específica.</p> <p>O próximo requisito validado foi o Montar Artefato. Até o momento foi desenvolvido apenas um protótipo para representar todo o requisito. Na validação viu-se a necessidade de se criar vários protótipos envolvendo cada situação das dependências entre artefatos e as visões de preenchimento e impressão. Além disso notou-se a falta de:</p> <ul style="list-style-type: none"> - lista dos serviços mapeados; 				

- opção de criação de um novo Membro (Novo);
- definição de um layout para a organização dos componentes do artefato (tabela, grid ou outro tipo);
- cada entrada de texto ser linkada a seu rótulo (label);
- reorganização do quadro de componentes (TipoMembro, Serviços, Artefatos);
- reorganização do quadro de propriedades dos Membros (fonte, posicionamento, propriedades específicas).

Para a próxima reunião, pretende-se validar os protótipos dos demais requisitos.

Observação Importante

- Técnica utilizada durante a reunião: Entrevista informal.

- Contatos:

Johnys Custódio:	email: jjohnys@gmail.com	telefone: (62) 9339-4670
Leonardo Pessoa:	email: nardopessoa@gmail.com	telefone: (62) 9123-7035
Guiliano Rangel:	email: guiliano@gmail.com	telefone:

Requisitos Registrados

Tabela 21 – Ata da Reunião n.º 10

4.1.11 Ata n.º 11

Ata da Reunião n.º 11				
Data	Horário	Local		
24/09/2011	Das 14h35m às 15h45m	Casa do orientador Guiliano Rangel		
Equipe técnica	Johnys Custódio da Silva Rabelo, Guiliano Rangel.			
Equipe usuária	Não se aplica			
Teor desta reunião				
Arquivo jar, arquivo “manifest” faz a funcionalidade de descobrir qual a classe principal e a estrutura de pacote.				
O arquivo de importação dos TipoMembro e serviço deverão conter na sua estrutura do pacote algo que diferencie duas versões diferentes.				
Dado o inicio da reunião foi demonstrado a correção do caso de uso manter TipoMembro.				
1. Ficou definido que a rotina de versionamento será transformada em um cenário e linkada no caso de uso manter artefato.				
2. O cenário verificar espelho, será transformado em uma rotina e não deverá ter caso de uso especificando ela.				
3. Não há necessidade de explicar rotinas internas nos casos de uso. Somente fazer um diagrama de atividades caso haja necessidade de visualizar as atividades que serão realizadas.				
4. Para a importação dos arquivos, verificamos que o padrão da Oracle possui um arquivo manifest que indica a estrutura de pacotes, e classe principal. Essa informação deverá ser utilizada no momento de mapeamento de novos tipomembros e serviços para o framework,				
Apos a validação do caso de uso, iniciamos a discussão de como será feito o preenchimento dos parâmetros dos serviços e dos tipomembros.				

Assuntos abordados:

5. A aplicação deverá fornecer objetos padrões(artefato, membros, serviço, etc.) para a entrada de parâmetros de serviços e tipomembros.

6. Como o serviço não terá representação na visão, ficamos com o problema de como será feita o preenchimento destes parâmetros na visão.

6.1. Para uma possível solução, observamos que o descritor de tela poderia informar um formato padrão de entrada de parâmetros, o qual seria interpretado pela aplicação.

7. Com a observação desse problema, visualizamos outro problema: Como faremos o preenchimento de parâmetros compostos (parâmetros que não tem como entrada um tipo primitivo).

Para próxima reunião, ficamos de validar os outros casos de uso que foram feitos, e debater a solução do problema 6 e 7.

Observação Importante

- Técnica utilizada durante a reunião: Entrevista informal.

- Contatos:

Johnys Custódio:	email: jjohnys@gmail.com	telefone: (62) 9339-4670
Guiliano Rangel:	email: guiliano@gmail.com	telefone:

Requisitos Registrados

Tabela 22 – Ata da Reunião n.º 11

4.1.12 Ata n.º 12

Ata da Reunião n.º 12		
Data	Horário	Local
29/11/2011	Das 14h35m às 15h35m	Casa do orientador Guiliano Rangel
Equipe técnica	Johnys Custódio, Leonardo Pessoa e Guiliano Rangel	
Equipe usuária	Não se aplica.	

Teor desta reunião

Data o inicio da reunião foi demonstrados alguns problemas para serem solucionados:

- Como as versões antigas serão mantidas?
- Qual o numero máximo de componentes por jar?
- URLClassLoader não consegue carregar 2 classes com o mesmo nome (pacote + nome)
- Qual classe concreta do arquivo .jar clamar. Usar parâmetros no manifest?
- Verificação do arquivo .jar derruba a VM caso a classe contenha algum erro.

Soluções propostas:

Remover Membro: Remover membro no ato de remover um artefato. Chamar caso no caso de uso Remover artefato.

Como as versões antigas serão mantidas: Cada TipoMembro deve ser implementado dentro de um pacote com o seu nome. Ou seja, o ultimo pacote definido tem que ser o nome do TipoMembro seguido de sua versão.

Número máximo de componentes por jar: 1 componente por jar.

Qual classe concreta do arquivo .jar clamar: Usar parâmetros no manifest: Utilizar o parâmetro Main-Class, para definir a classe principal do pacote que será utilizada pelo framework.

Verificação do arquivo .jar derruba a VM caso a classe contenha algum erro: Não precisa escrever no requisito só especificar no diagrama de caso de uso.

A persistência dos parâmetros do tipo será feita a partir de serialização dos arquivos.

Dividir os casos de uso.

Criar Ator administrador, ele será pré-cadastrado no banco de dados.

Não tem workflow.

Caso o usuário seja removido da equipe, caso ele esteja logado, ele será deslogado do framework.

Caso o usuário seja inativado, e caso ele esteja logado, ele será deslogado do framework.

Requisitos futuros: Publicar artefato, worflow, serviço de validação,

O artefato terá sua interface para que o núcleo framework seja extensível.

Mudar versão e revisão só quando tiver em multi usuário.

Jar ter o nome do componente + numero de versão

Observação Importante

- Técnica utilizada durante a reunião: Entrevista informal.

- Contatos:

Johnys Custódio:	email: jjohnys@gmail.com	telefone: (62) 9339-4670
Leonardo Pessoa:	email: nardopessoa@gmail.com	telefone: (62) 9123-7035
Guiliano Rangel:	email: guiliano@gmail.com	telefone:

Requisitos Registrados

Tabela 23 – Ata da Reunião n.º 12

4.1.13 Ata n.º 13

Ata da Reunião n.º 13				
Data	Horário	Local		
03/12/2011	Das 14h35m às 15h55m	Casa do orientador Guiliano Rangel		
Equipe técnica	Johnys Custódio, Leonardo Pessoa e Guiliano Rangel			
Equipe usuária	Não se aplica.			
Teor desta reunião				
Dado o início da reunião, o orientador revisou as correções feitas no projeto, e sugeriu algumas pequenas modificações. Solicitou também uma redução do escopo, onde deve ser colocado somente as funcionalidades implementadas, e adicionar no não escopo os requisitos que não foram implementados e ficaram como requisitos futuros.				
Na implementação surgiram algumas dúvidas sobre os TipoMembro disparadores: O serviço deve conhecer o TipoMembro disparador? O TipoMembro disparador conhece o serviço?				
- O serviço não conhece o TipoMembro que o dispara, somente o TipoMembro disparador que faz chamada para execução do serviço através do artefato, porém o serviço acionado é um parâmetro				

do TipoMembro.

Também levantou-se questões sobre a categorização dos Serviços, pois mesmo os serviços de persistência e dependência estando descritos separadamente, não há nenhum atributo que os identifica como persistência ou dependência.

Guiliano nos orientou a fazer interfaces que herdam da interface IServico e que estas possuam superclasses, onde o framework identificaria o tipo de serviço através destas interfaces e/ou superclasses.

Foi solicitado também que as superclasses utilizadas no projeto sejam especificadas na arquitetura.

Observação Importante

- Técnica utilizada durante a reunião: Entrevista informal.

- Contatos:

Johnys Custódio:	email: jjohnys@gmail.com	telefone: (62) 9339-4670
Leonardo Pessoa:	email: nardopessoa@gmail.com	telefone: (62) 9123-7035
Guiliano Rangel:	email: guiliano@gmail.com	telefone:

Requisitos Registrados

Tabela 24 – Ata da Reunião n.º 13

4.2 Descrição do Negócio

A documentação de um trabalho baseado em um padrão de documento, ou seja, em que se exista um template a ser preenchido, engloba uma coleção de vários artefatos (figuras, textos, gráficos, etc.) que se relacionam a fim de servir como elemento de comunicação durante um projeto e auxiliar o entendimento das atividades que estão ocorrendo.

Um dos aspectos importantes na documentação é a maneira como ela é construída. Nela deve se definir os fluxos de trabalhos, quais produtos serão construídos e o tempo estimado de cada etapa, a fim de se manter a organização durante o desenvolvimento do trabalho e a coerência das informações.

A documentação serve como um repositório de informações de todo o processo de desenvolvimento. Com ela é possível verificar pontos vulneráveis no sistema, aprofundar em uma análise dos riscos que determinada alteração poderá causar.

O desenvolvimento rápido de um projeto às vezes gera uma documentação fraca e que não atenderá as necessidades futuras.

No caso específico da documentação de software, existe uma estrutura que faz a ligação entre os artefatos. Esta estrutura é definida a partir de um modelo que contêm suas características, a ordem que estas serão produzidas e suas possíveis dependências. Com este modelo é possível definir o processo geral para a execução do projeto.

Podemos dividir a documentação de sistemas em quatro etapas: escolha do modelo (nela se encontram o conjunto de artefatos e suas peculiaridades), o preenchimento, a visualização do documento e sua impressão. Em toda a estrutura existem cargos específicos (arquiteto, programador, testador, etc.) que criaram ou executarão determinada etapa da documentação.

Com base nestas informações formula-se um processo, que descreve as fases da metodologia que será utilizada, os elementos que serão construídos e a sua finalidade, para que ao final do projeto,

se tenha todo um conjunto de artefatos que possibilitem outra equipe entenderem o problema e trabalharem de forma coerente para realizar possíveis alterações e/ou evoluções.

4.3 Lista de Requisitos

Identificação	Requisito
R01	Manter TipoMembro
R02	Manter Membro
R03	Manter Artefato
R04	Montar Artefato
R05	Clonar Artefato
R06	Preencher Artefato
R07	Visualizar Artefato
R08	Imprimir Artefato
R09	Manter Modelo
R10	Manter Documento
R11	Manter Serviços
R12	Gerenciar Serviço de Persistência
R13	Gerenciar Serviço de Dependências
R14	Manter Log das Operações do Sistema
R15	Gerenciar Multiusuários
R16	Gerenciar Serviço de Validação de Dados

Tabela 25 – Lista de Requisitos

4.4 Lista de Regras de Negócio

Identificação	Regras de Negócio	Requisito Associado
RN01	Na inclusão do TipoMembro, deve-se conter dois arquivos que representam a parte da visão e a do controle do TipoMembro.	R01
RN02	O Nome do Membro é único, e o identifica dentro do Framework.	R02
RN03	Os membros cadastrados no sistemas serão agrupados através de seus respectivos TipoMembro, ordenados alfabeticamente.	R02
RN04	O Nome do Artefato é único, e o identifica dentro do Framework.	R03, R04, R05

RN05	O artefato ao ser criado pela dependência de criação deverá preencher seu Título.	R03
RN06	Os artefatos deverão ser agrupados em categorias, para facilitar a busca no momento de criação do modelo.	R03
RN07	Todo artefato deve conter o serviço de verificação de campos obrigatórios.	R03
RN08	Não pode existir dois artefatos com mesmo título em um documento.	R03
RN09	O log das operações exercidas pelo usuário Desenvolvedor não pode ser desativado por nenhum usuário do sistema.	R14
RN10	A execução dos serviços é feita de modo hierárquica, do maior nível até o serviço que foi solicitado.	R06
RN11	Para se gerar a impressão de um artefato, o mesmo deve estar preenchido.	R08
RN12	Artefatos que podem ser preenchidos simultaneamente devem ter a mesma ordem de preenchimento.	R09
RN13	O framework permitirá o preenchimento de um artefato desde que todos os artefatos do modelo com ordem menor já estejam preenchidos. Obs.: Estes artefatos de ordem menor devem ter correlação entre si. Por exemplo: uma Ata, um requisito, um Caso de Uso, caso criemos uma ata que gere um requisito em específico e posteriormente um caso de uso, esta ordem deverá ser seguida. Porém, se criar duas Atas que gerem dois requisitos cada, não necessariamente teremos que preencher todos estes requisitos para fazer um caso de uso.	R09
RN15	O framework só permitira salvar um Modelo se semente todos os artefatos dependentes forem utilizados.	R09
RN16	Um modelo pode ser alterado/removido desde que não exista nenhum documento baseado neste modelo.	R09
RN17	O nome do Modelo é único e o identifica dentro do framework.	R09
RN18	Somente os usuários do grupo poderão preenche/alterar o projeto.	R10
RN19	Dois usuários não poderão preencher simultaneamente um mesmo artefato do projeto.	R10
RN20	O nome do Serviço é único e o identifica dentro do framework.	R11
RN21	O nome do projeto é único e o identifica dentro do framework.	R10
RN22	Não deve ser possível a ocorrência de "dependência circular", ou seja, que o Artefato-A seja dependente do Artefato-B e o Artefato-B seja dependente do Artefato-A (em se tratando da mesma dependência).	R13

Tabela 26 – Lista de regras de negócio

4.5 Dificuldades Encontradas no Negócio

Dificuldades	Propostas de Soluções
As alterações na formatação do documento de forma manual acarretam o gasto de muito tempo; (Alteração de formatação em todo o documento, padronização)	Criar uma ferramenta que gerencie a formatação do documento como um todo.
Os documentos confeccionados por dois usuários necessitam de serem unidos manualmente, não mantendo o mesmo formato ou estilo de formatação.	Criação de um repositório único da documentação, não necessitando da união de dois documentos.
Como a documentação é feita em computadores pessoais, e não a o armazenamento em um repositório único, fica difícil o acesso de toda a documentação em sua ultima versão, a perda de informações.	Criação de um repositório único que mantenha sempre atualizada a versão da documentação.
Algumas partes do documento são de certa forma abandonadas pelo estudante.	Criar uma ferramenta capaz de reconhecer a obrigatoriedade das partes garantindo assim a integridade e coesão das informações
O tratamento manual do documento faz com que as dependências também sejam tratadas manualmente, essas dependências podem conter erros. Ex: Um requisito alterado ou excluído pode estar sendo referenciado na realização de casos de uso.	Criar uma funcionalidade que garanta a integridade de dependência entre as partes distintas do documento, podendo ser flexível ou obrigatória.
Os controles que ocorrem nos documentos devem ser feitos conceitualmente e manualmente, isso pode gerar erros no layout.	Criar uma ferramenta que trate o layout, o modelo do documento para que não haja erro no mesmo.
Seguir as regras que o modelo define de forma manual (Obrigatoriedade entre os elementos do documento).	Criar uma ferramenta que controle as regras definidas pelo modelo.

Tabela 27 – Lista de dificuldades encontradas no negócio

4.6 Definição do Problema

Uma comparação é feita por [VALERIANO, 1998] para relatar os problemas em se documentar trabalhos: "Muitos projetos não nos legaram sua documentação nem qualquer de seus registros. De alguns, não se conhece nem mesmo seu objetivo, como os megalitos de Stonehenge, por exemplo. De outros, não se sabe como foram executados: as pirâmides egípcias, as das civilizações pré-colombianas e os monumentos da Ilha de Páscoa, apesar de serem muito admirados. Entretanto, da Arca de Noé, quer tenha sido real ou não, conhecem-se suas 'especificações' bíblicas, o que permitiria construir uma 'réplica', com razoável segurança".

Hoje em dia, existe uma grande dificuldade por parte das pessoas em cumprirem as metas de prazo e orçamento causando muita resistência em se fazer uma boa documentação. Muitas das vezes

isto está diretamente ligado a satisfação do cliente e, consequentemente, ao sucesso do prestador de serviço. Deste modo, a documentação é vista como perda de tempo é facilmente deixada de lado, pois a pressão de curtos prazos e custos define a velocidade de construção do produto, e atrasos não são vistos com bons olhos.

A criação de documentos está ligada diretamente com a qualidade do produto que se entrega. Em um grupo de pessoas, para evitar discussões e complicações futuras, os documentos armazenados são de grande importância. Ninguém pode garantir que o conhecimento que está na cabeça de um integrante está claro para toda a equipe e se manterá vivo caso ocorra o desmembramento da mesma. Neste caso, a falta de documentação ou a documentação incorreta afeta diretamente a compreensão do problema proposto.

Normalmente, documentos são baseados em templates, ou seja, um conjunto de padrões aplicados na distribuição dos componentes que o formam, algumas relações entre eles e, em alguns casos, até um fluxo de trabalho onde certos componentes devem ser produzidos antes de outros. Um documento pode ser dividido em vários outros documentos, cada qual com seu objetivo específico, que na conclusão do trabalho precisam ser mesclados para uma possível impressão, distribuição ou armazenamento destes dados, respeitando uma ordem e uma interdependência entre eles.

Outro problema comum são templates mal produzidos onde ocorre um excesso de documentação sem propósito, a repetição de textos comuns para vários componentes e o desrespeito aos padrões de formatação recomendados pela ABNT. Junto a isto está a necessidade de se manter o documento atualizado, pois, para acompanhar o mercado, um bom produto deve estar em constante evolução para melhor satisfazer seus clientes.

Por outro lado há uma gama de softwares editores de texto incompatíveis entre si, trazendo a tona o problema da visualização e edição do trabalho, apresentando erros de layout, fonte, concordância entre elementos e etc.

Tamanha é a complexidade para que o documento proposto seja de qualidade que o desenvolvedor do mesmo gasta a maior parte do seu tempo preocupado com o trabalho manual de formatação e organização, sendo que poderia usar este tempo no desenvolvimento do conteúdo, provavelmente melhorando o documento como um todo.

O mercado tem sentido na pele a diferença que se faz uma boa documentação, mesmo assim ainda a considera um processo demorado e um tanto caro, porém é nela que se consegue prever riscos no projeto e assim antecipar planos de prevenção. É incrível como empresas dizem não ter dinheiro nem tempo para documentar seus produtos mas, consequentemente, disponibilizam milhões para o recall dos mesmos.

4.7 Objetivo do Software

O projeto em questão tem como objetivo a criação de um framework que almeja a flexibilidade em relação aos tipos de documentos tratáveis, ou seja, qualquer documento estruturado que pode ser baseado em um modelo (template). Para este gerenciamento, é disponibilizado funcionalidades e ferramentas que envolvem todo o processo de documentação, desde a segmentação do modelo até seu preenchimento e publicação.

Devido à evolução constante dos modelos utilizados para estes fins, o framework está preparado para a inserção de novos componentes sem que seja necessária uma alteração em sua estrutura lógica e/ou arquitetural, desde que estes sigam os padrões definidos neste projeto.

Para a “virtualização” do modelo, ou seja, sua construção dentro do framework, é necessário identificar todos os artefatos que o compõe (figura, tabela, texto livre, gráfico, etc.). Para isso é exigida uma análise detalhada e em seguida uma segmentação do template. Com os artefatos identificados, o framework disponibiliza recursos para a montagem e para o relacionamento dos mesmos.

Com o modelo construído, o framework possibilita o preenchimento do mesmo, fazendo todo o controle de obrigatoriedade e multiplicidade e respeitando as dependências definidas no ato da montagem. Enfim, permite alcançar a qualidade máxima do documento, focando a atenção do preenchedor no conteúdo produzido e abstraindo todo o processamento.

Um projeto a ser documentado pode envolver uma equipe durante todo seu ciclo de vida. O framework controla as informações pertinentes ao projeto somente àquela equipe relacionada e mantém em logs as atividades de todos os usuários.

4.8 Escopo do Software

A base para o bom funcionamento de um framework é uma arquitetura que atenda todos os requisitos levantados, e pensando nisso, para esta versão, será disponibilizada uma arquitetura que possa atender os principais requisitos: flexibilidade e extensibilidade. Uma arquitetura multicamadas com camadas definidas e responsabilidades distintas, o que gera um desacoplamento entre as mesmas.

O Quid-IDE disponibiliza alguns perfis de usuário (Administrador, Gerente de Projetos, Desenvolvedor, Montador e Preenchedor), cada qual envolvendo uma atividade específica. A criação de novos componentes e serviços, ou seja, a extensibilidade do framework, está relacionada com o usuário desenvolvedor. A identificação e criação dos artefatos e a organização da estrutura do documento é tarefa do usuário montador. Cabe ao usuário preenchedor apenas a preocupação do preenchimento dos modelos definidos pelo usuário montador. Devido à necessidade de uma equipe trabalhar no mesmo projeto, o Gerente de Projeto é responsável por definir equipes, ou seja, selecionar dentre os usuários aqueles que farão parte do projeto. O Administrador é o responsável pela aplicação, algumas de suas responsabilidades são o cadastro de usuários e a definição do perfil de cada um.

O Quid-IDE é capaz de receber os novos componentes (arquivos JAR), importá-los e disponibilizá-los para uso em tempo de execução, sejam eles TipoMembros ou Serviços.

Na montagem do artefato, define-se os parâmetros para TipoMembro's escolhidos, criando assim os Membros. Vale lembrar que a visão de montagem e preenchimento disponibilizadas pelos TipoMembro's estão voltadas para a tecnologia de visão escolhida.

O Modelo é formado por artefatos, entretanto, para esta versão, não há um controle de ordem ou dependência entre os artefatos, somente são escolhidos os artefatos que compõe o modelo, o índice será tratado de forma diferente, com um serviço de indexação, porém não será disponibilizado nesta versão.

O framework não disponibiliza uma interface para programação (IDE) para a criação de novos componentes, somente permitirá a importação de arquivos (classes) com a implementação desenvolvida na linguagem Java.

Nesta versão não está implementado a publicação do artefato, o controle de logs das operações do framework, o gerenciamento dos serviços de dependência e persistência ou de qualquer outro serviço mapeado no framework e a clonagem de artefato.

4.9 Clientes Alvo

Nome	Definição
Acadêmicos	Pessoa que aprende uma ciência, arte ou técnica em qualquer área e que seja capaz de desenvolver documentos.
Orientadores	Pessoa que ensina uma ciência, arte ou técnica em qualquer área e que esteja ligada ao desenvolvimento de documentos.
Desenvolvedores	Pessoa capacitada a desenvolver software utilizando a biblioteca extensível do framework Quid-IDE, escrita na linguagem Java.

Tabela 28 – Lista de clientes alvo

4.10 Glossário do Negócio

Termo	Definição	Sinônimo
Framework	Conjunto de classes estendível para um determinado fim.	Ferramenta de suporte
Workflow	Maneira temporal como um trabalho é realizado e controlado	Fluxo de trabalho
Template	Documento com formatação e tópicos pré-configurados, porém sem conteúdo	Modelo pré-definido
Recall	Ocorre pela descoberta de problemas relativos à segurança do produto (do inglês "chamar de volta")	Solicitação de troca obrigatória
Implementação	Escrever o código de um programa.	Programar
Layout	Desenho padrão	Template
Artefato	Qualquer item identificado em um modelo de documento	Elemento de um documento

Tabela 29 – Glossário do negócio

5 Requisitos

5.1 Especificação de Requisitos n.º 01

5.1.1 R01 – Manter TipoMembro

Incremento	Primeiro Incremento
Nome	Manter TipoMembro
Tipo	Funcional

Definição	<p>Requisito responsável por mapear TipoMembro para o framework. O TipoMembro é a entidade de representação mais baixa do framework, nele ficarão armazenados informações pertinentes de visão e persistência.</p> <p>O TipoMembro deve ter duas divisões (Lógicas): Visão e Modelo. Sendo que o modelo é independente da visão, e poderá ter varias visões (tecnologia) para um mesmo modelo.</p> <p>TipoMembro - Visão terá as seguintes características:</p> <ul style="list-style-type: none"> a) Interpretar o TipoMembro - Modelo, e se desenhar na Interface. Neste desenho deverão ser aplicadas as propriedades de Impressão e visualização definido no requisito Manter Membro. b) Apresentação ao usuário: Esta apresentação consiste em imagem ilustrativa, Nome do TipoMembro e um texto descritivo. c) Validação de tipos básicos. d) Divisão em visualização, onde o usuário poderá ver a descrição do Membro, e digitação onde o usuário irá preencher os dados do documento. Quando este dado for digitado ele substituirá a descrição do Membro. e) Fornecer uma lista dos parâmetros do TipoMembro Modelo a serem desenhados pelo Descritor de Tela para o preenchimento. f) Evento no qual a máscara deve ser inserida: Informar em qual evento a máscara deve ser inserida. Estes eventos serão definidos pelo TipoMembro Visão. Nele deverá conter quais eventos a máscara poderá ser inserida, de acordo com a tecnologia da visão que se está utilizando. <p>TipoMembro - Modelo terá as seguintes características:</p> <ul style="list-style-type: none"> a) Tipo de dado para persistência: Informar qual é o tipo de dado (varchar, integer, float, etc.), para sua persistência. b) TipoMembro estático: TipoMembros estáticos, são aqueles que não necessitam de coluna na tabela de criação de um artefato, pois seu conteúdo não é preenchido pelo usuário. Ex (Título, uma linha de divisão, nome da instituição, etc.). c) Tipo de Máscara de Preenchimento: Informar através de um layout, o padrão de máscara de entrada de dados caso possua. d) Domínio: Informar qual tipo de dado para a entrada de dados. (Alfanumérico, número ou somente letras, expressão de linguagem, etc.) e) Validação de Entrada de Dados: Ele deverá informar se determinada entrada de dados contém um valor correto. f) Indexado: Deve informar se ele é um componente que deve ser indexado pelo serviço de indexação. g) Parâmetros: Cada TipoMembro contém uma lista de parâmetros que servirão para sua formatação para a visualização / Impressão, e para outras funcionalidades que o TipoMembro necessite. h) Validação dos Parâmetros: Cada TipoMembro deverá validar os parâmetros obrigatórios para seu funcionamento. <p>O TipoMembro deve ter o controle de versões em dois níveis:</p> <p>Nível 1 (Versão): Garante a versão de um TipoMembro, com compatibilidade. Caso seja uma nova versão, deve-se realizar um novo</p>
------------------	--

	<p>mapeamento, como se fosse um novo TipoMembro, com o mesmo nome porem com uma versão diferente.</p> <p>Nível 2 (Revisão): Garante a compatibilidade da versão anterior (Atualização menores). Caso seja uma nova revisão, haverá somente a atualização do TipoMembro no cadastro do FrameWork.</p> <p>O TipoMembro deverá conter um tipo abstrato que determina as possíveis operações sobre seu valor.</p> <p>TiposMembros disparadores, são TiposMembros associados a serviços. Eles são responsáveis por chamar a execução de um servido de um Artefato.</p> <p>O formato dos arquivos de importação deverá seguir o padrão definido pela Oracle, contendo a estrutura de pacotes correta. Dentro do arquivo MANIFEST.MF deverá ter um parametro chamado Main-Class, que identifica a classe de utilização do TipoMembro (Classe Concreta), está classe deverá implementar as interfaces definidas na arquitetura. No momento de importação do TipoMembro o formato será validado.</p> <p>O atributo nome do TipoMembro (visão/modelo) será seu identificador único no framework para a verificação de espelhamento do mesmo. Para a diferenciação da plataforma de cada TipoMembro-Visão, ele deverá informar o nome da sua plataforma para a qual ele foi desenvolvido, sendo possível ao usuário desenvolvedor o acesso a teste nome caso necessite de criar novos TipoMembros-Visao.</p> <p>Cada TipoMembro deverá conter dois números de identificação da versão e da revisão, que sera utilizado pelo serviço de versionamento do framework. Caso o usuário tente importar um TipoMembro ja cadastrado, com mesmo número de versão e revisão, o framework deverá informar que o TipoMembro já está cadastrado, e não realizar a importação do arquivo.</p> <p>Cada TipoMembro Visão deverá conter um TipoMembro Modelo associado a ele no framework.</p>
Regras de negócio associadas	RN01

Tabela 30 – Tabela de especificação do requisito R01

5.1.2 R01 – Protótipo de Interface

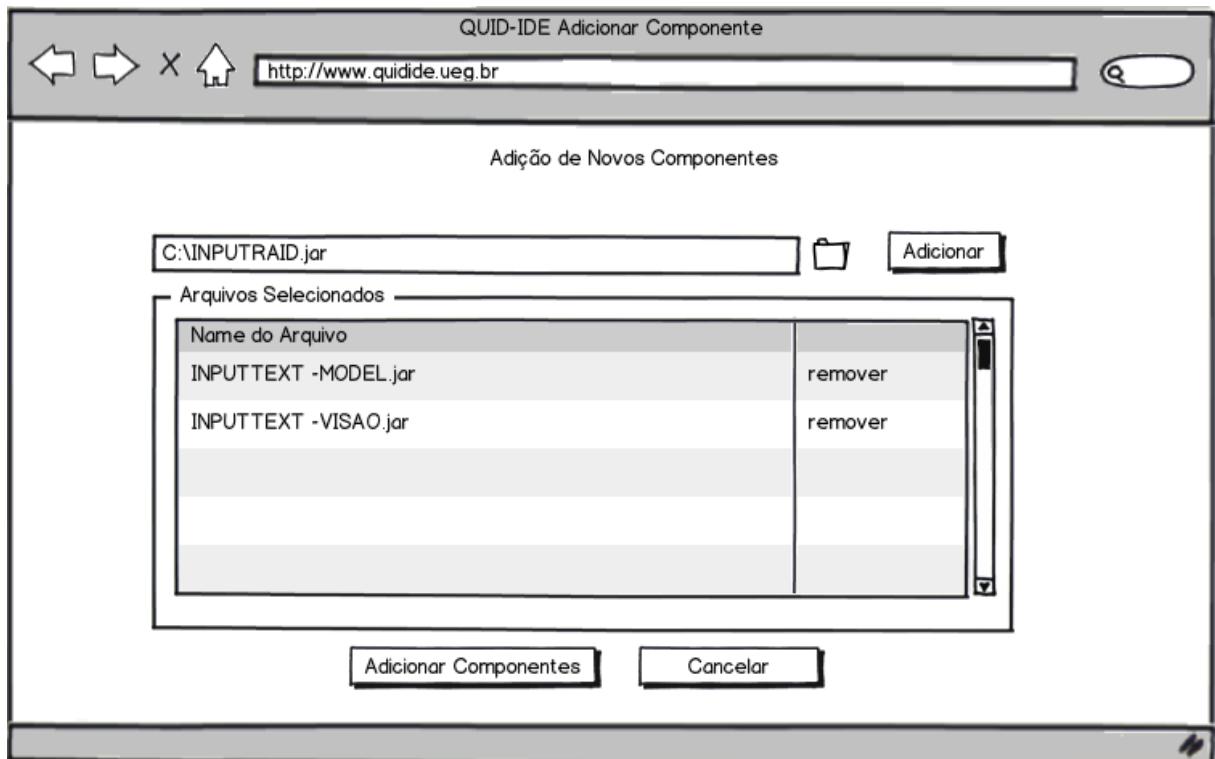


Figura 1 – Importação de TipoMembro

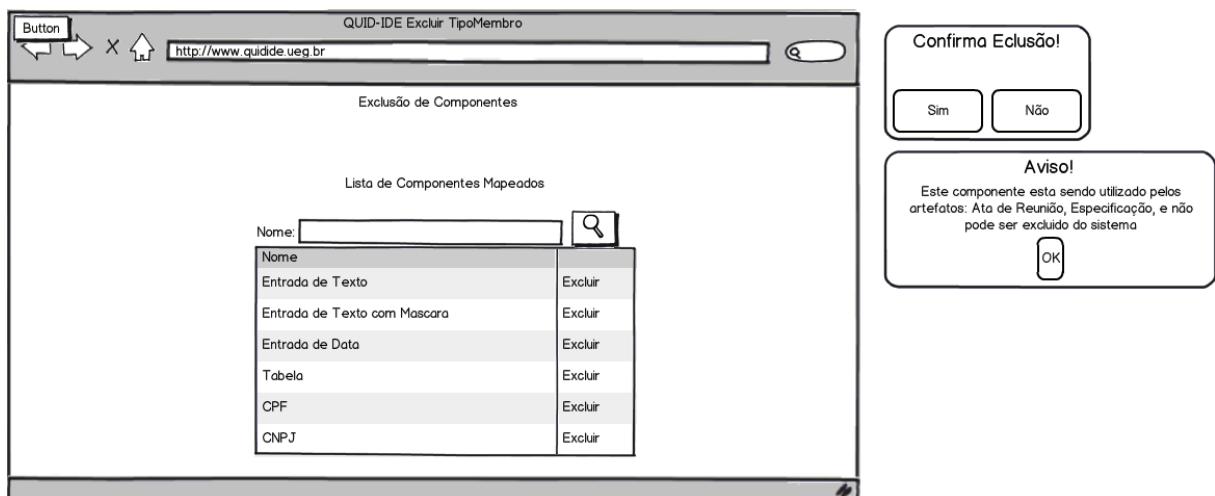


Figura 2 – Exclusão de TipoMembro

5.1.3 R01 – Informações Relacionadas

Item	Tipo de Dado	Tamanho	Definido pelo Sistema	Obrigatório	Valor inicial	Domínio	Máscara
codigo	Integer	-	Não	Sim	-	-	-
descricao	String	255	Não	Não	-	-	-
nome	String	255	Não	Sim	-	-	-
versao	Integer	-	Não	Sim	-	-	-
revisao	Integer	-	Não	Sim	-	-	-
ativo	Boolean	-	Não	Sim	true	-	-
aguardandoexclusao	Boolean	-	Não	Não	false	-	-
tipotipomembro	String	255	Não	Sim	-	VISAO / MODELO	-

Tabela 31 – Informações relacionadas ao requisito R01

5.1.4 R01 – Funcionalidades Relacionadas

Ação	Funcionalidade
Mapear TipoMembro	Realiza o mapeamento TipoMembro para o framework. Este mapeamento será composto por : a) Gravação do Arquivo: Inclui um novo TipoMembro a partir de seu arquivo, o armazenando em um local específico.
Remover TipoMembro	Pode ser realizada durante o Mapeamento dos TiposMembros. Caso haja o cadastro do TipoMembro no framework, e uma ausência do arquivo no diretório, poderá ocorrer duas ações: a) Remover o TipoMembro caso não esteja sendo utilizado por nenhum membro ou serviço. b) Realizar a rastreabilidade dos locais em que o TipoMembro estava sendo utilizado, e informar para o usuário a ausência do TipoMembro nos locais rastreados (Modelos, e locais no Modelo). A remoção do TipoMembro também poderá ser realizada através de uma funcionalidade fornecida pelo framework.
Pesquisar TipoMembro	Realiza a pesquisar um determinado TipoMembro mapeado no framework. Esta pesquisa retornará um TipoMembro - Modelo, que será representado na visão.
Verificar Espelho	Verificação de correspondência entre o TipoMembro Modelo e sua representação no TipoMembro Visão.

Tabela 32 – Funcionalidades relacionadas ao requisito R01

5.2 Especificação de Requisitos n.º 02

5.2.1 R02 – Manter Membro

Incremento	Primeiro Incremento
Nome	Manter Membro
Tipo	Funcional
Definição	<p>Requisito responsável por manter um Membro no Framework. O membro é a representação do TipoMembro preenchidos os seus parâmetros. Para este preenchimento, no momento de criação do Membro ele deve armazenar os parâmetros do TipoMembro fornecidos pela Visão.</p> <p>O membro deve conter as seguintes informações:</p> <ul style="list-style-type: none"> a) Nome e descrição do Membro. b) Lista parâmetros do TipoMembro. c) O TipoMembro que ele representa. <p>O framework sera responsável por armazenar os parametros do TipoMembro.</p> <p>Os parametros do Membro deverão seguir um padrão definido pelo framework, para que esses possam ser persistidos pelo mesmo.</p>
Regras de negócio associadas	RN02, RN03

Tabela 33 – Tabela de especificação do requisito R02

5.2.2 R02 – Protótipo de Interface

Os protótipos deste requisito estão relacionados com os protótipos apresentados no requisito R04 – Montar Arfefato.

5.2.3 R02 – Informações Relacionadas

Item	Tipo de Dado	Tamanho	Definido pelo Sistema	Obrigatório	Valor inicial	Domínio	Máscara
codigo	Integer	-	Não	Sim	-	-	-
descricao	String	255	Não	Não	-	-	-
nome	String	255	Não	Sim	-	-	-
x	Integer	-	Não	Sim	-	-	-
y	Integer	-	Não	Sim	-	-	-
comprimento	Integer	-	Não	Sim	-	-	-
altura	Integer	-	Não	Sim	-	-	-

tipoMembroModelo	TipoMembro	-	Não	Sim	-	-	-
------------------	------------	---	-----	-----	---	---	---

Tabela 34 – Informações relacionadas ao requisito R02**5.2.4 R02 – Funcionalidades Relacionadas**

Ação	Funcionalidade
Incluir Membro	Funcionalidade responsável por cadastrar um Membro dentro do FrameWork. Este cadastro é composto pelo armazenamento dos parâmetros do TipoMembro e de seu Título e descrição. No momento da criação de um membro, ele deverá verificar se todos os parâmetros obrigatórios do TipoMembro foram informados.
Pesquisar Membro	Funcionalidade responsável por pesquisar Membros no framework. Esta pesquisa poderá ser feita a partir do TipoMembro, retornando todos os membros que são compostos por ele (categorização a partir do TipoMembro), ou pelo nome do Membro.
Alterar Membro	Funcionalidade responsável por alterar um membro cadastrado no sistema. Esta alteração poderá ser realizada somente se o Membro não estiver sendo utilizado por nenhum Artefato do Framework. Caso esteja, o sistema deverá clonar o Membro, solicitando a alteração do Nome do mesmo.

Tabela 35 – Funcionalidades relacionadas ao requisito R02**5.3 Especificação de Requisitos n.º 03****5.3.1 R03 – Manter Artefato**

Incremento	Primeiro Incremento
Nome	Manter Artefato
Tipo	Funcional
Definição	<p>Requisito responsável por mapear um Artefato para o framework. O artefato e a entidade que representa uma parte do documento que está sendo montado (Capa, Contracapa, Ata, Glossário, etc.) que siga as especificações descritas no Escopo do Software.</p> <p>Cada artefato deverá conter:</p> <ul style="list-style-type: none"> a) Nome e descrição: O nome do artefato será utilizado no momento da montagem do Modelo. Sua descrição informa o que o artefato representa como parte de um documento. b) Título: Atributo do artefato que diferencia dois artefatos do mesmo Tipo (Ex: Duas Atas), preenchidos em um documento. <p>Cada artefato tem a responsabilidade de:</p> <ul style="list-style-type: none"> a) Identificar os Membros que ele é composto. b) Identificar os serviços que serão executados. c) Determinar a ordem em que os serviços deverão ser executados. d) Pegar o resultado da execução de um serviço e encaminhar para outro serviço. e) Os serviços que serão executados a partir dos eventos disparados pelo

	<p>usuário são associados aos TiposMembros disparadores (Botões, links), porém nem todos os serviços possuem uma representação visual.</p> <ul style="list-style-type: none"> • Ao se disparar um evento, o artefato é responsável por executar todos os serviços anteriores a o serviço chamado, (determinado na ordem de eventos), passando se necessários os resultados da execução dos serviços. (Ver Requisito Manter Servico)
Regras de negócio associadas	RN04, RN05, RN06, RN07, RN08

Tabela 36 – Tabela de especificação do requisito R03

5.3.2 R03 – Protótipo de Interface

A manutenção de artefatos é feita através das telas relacionadas aos requisitos:

- R04 – Montar Artefato
- R05 – Clonar Artefato
- R06 – Preencher Artefato

5.3.3 R03 – Informações Relacionadas

Item	Tipo de Dado	Tamanho	Definido pelo Sistema	Obrigatório	Valor inicial	Domínio	Máscara
codigo	Integer	-	Não	Sim	-	-	-
descricao	String	255	Não	Não	-	-	-
nome	String	255	Não	Sim	-	-	-
titulo	String	255	Não	Sim	-	-	-
categoria	String	255	Não	Sim	-	-	-
membros	Collection<Membro>	-	Não	Sim	-	-	-
servicos	Collection<Servicos>	-	Não	Sim	-	-	-

Tabela 37 – Informações relacionadas ao requisito R03

5.3.4 R03 – Funcionalidades Relacionadas

Ação	Funcionalidade
Pesquisar Artefato	Funcionalidade que pesquisa artefatos dentro do Framework. Esta pesquisa poderá ser feita a partir da categoria e do titulo de um artefato. Quando a pesquisa não encontrar nenhum artefato, o framework deverá retornar uma lista vazia.
Remover Artefato	Funcionalidade que remove um artefato dentro do Framework. Para ser removido, um artefato não poderá estar sendo utilizado por nenhum

	Modelo cadastrado no Framework. Caso esteja o framework informará uma lista contendo os modelos em que o artefato está sendo utilizado.
Alterar Artefato	<p>Funcionalidade que permite alterar um artefato dentro do Framework. Um artefato só poderá ser alterado caso não esteja sendo utilizado por nenhum modelo no sistema. Caso esteja, poderão ocorrer as seguintes ocasiões:</p> <p>O artefato estar sendo utilizado em somente um modelo, e este não está sendo preenchido em nenhum projeto: O framework informará ao usuário em qual modelo o artefato está sendo utilizado, e solicitará uma confirmação da alteração do artefato.</p> <p>O artefato estar sendo utilizado por dois modelos e não está sendo preenchido em nenhum projeto: O Framework solicitará a confirmação da clonagem do artefato (Ver Requisito Clonar Artefato), e solicitará em qual modelo o clone deverá substituir.</p> <p>O artefato estar sendo utilizado em um ou mais projetos: O Framework informará em quais projetos o artefato está sendo preenchido. Caso o usuário montador ainda deseje alterar o artefato, o Framework permitirá somente a alteração dos parâmetros dos Membros, não permitindo a inclusão ou alteração de nenhum Membro ou Serviço do artefato.</p>
Mapear Artefato	Funcionalidade que insere um artefato ao cadastro do framework. Nesta inserção é verificada a duplicidade do artefato e o mapeamento de membros não cadastrados no framework..

Tabela 38 – Funcionalidades relacionadas ao requisito R03

5.4 Especificação de Requisitos n.º 04

5.4.1 R04 – Montar Artefato

Incremento	Primeiro Incremento
Nome	Montar Artefato
Tipo	Funcional
Definição	<p>A montagem do artefato consiste na adição dos membros e serviços que constituirão o artefato.</p> <p>O sistema deve exibir uma paleta contendo todos os TipoMembros cadastrados. Ao selecionar um determinado TipoMembro, o sistema deverá exibir uma lista com todos os membros cadastrados, e uma opção de criar um novo membro (Ver Requisito Manter Membro).</p> <p>Para a adição dos Membros ao Artefato o usuário montador deverá fornecer as seguintes informações:</p> <ul style="list-style-type: none"> a) O Membro a ser inserido. b) Caso não seja um Membro estático, o usuário montador poderá informar texto com a finalidade do Membro dentro do artefato. Este texto será mostrado ao usuário preenchedor, no momento da digitação do Artefato. c) Obrigatoriedade de Preenchimento: Determina se o membro e de

	<p>preenchimento obrigatório.</p> <p>d) Informa os parâmetros de indexação e tabulação (Ver Requisito "R01" Manter TipoMembro).</p> <p>e) Posiciona o membro no local desejado no artefato.</p> <p>Para a adição dos serviços em um artefato, o usuário deverá selecionar o serviço a ser inserido, informando se necessário os parâmetros de entrada e o local do seu respectivo TipoMembro. Cada serviço deverá ser organizado em uma lista ordenada, em que o Framework realizara a execução. Nesta lista o usuário poderá configurar que determinado resultado de um serviço, poderá ser entrada de outro serviço (Ver Requisito Manter Servico).</p> <p>Em qualquer momento da montagem, o usuário poderá ver uma previa do Artefato, que informa como será a versão de preenchimento / visualização.</p>
Regras de negócio associadas	RN04

Tabela 39 – Tabela de especificação do requisito R04

5.4.2 R04 – Protótipo de Interface

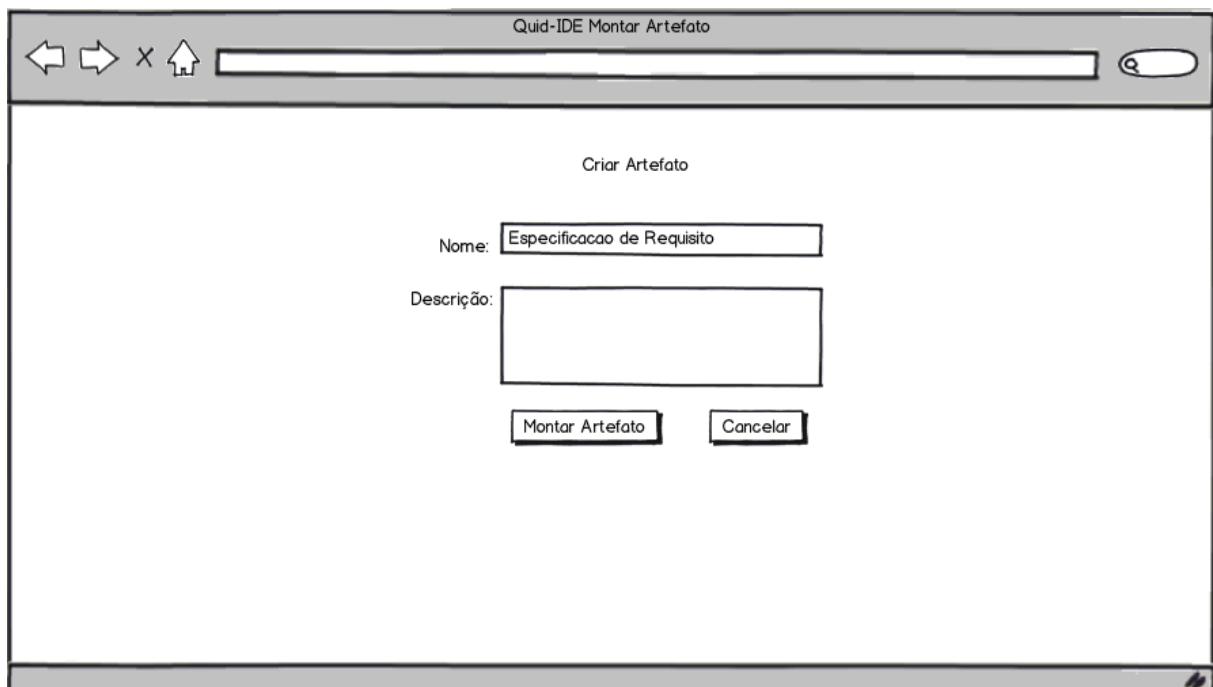


Figura 3 – Criar Artefato

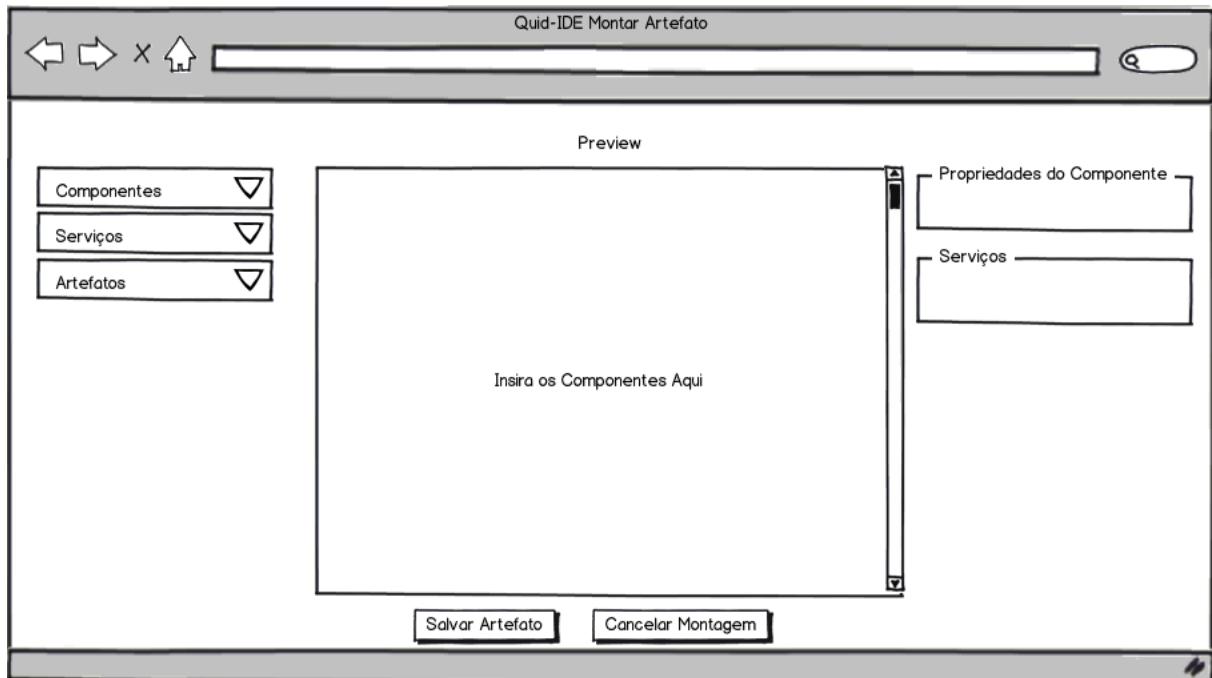


Figura 4 – Montar Artefato – Artefato vazio

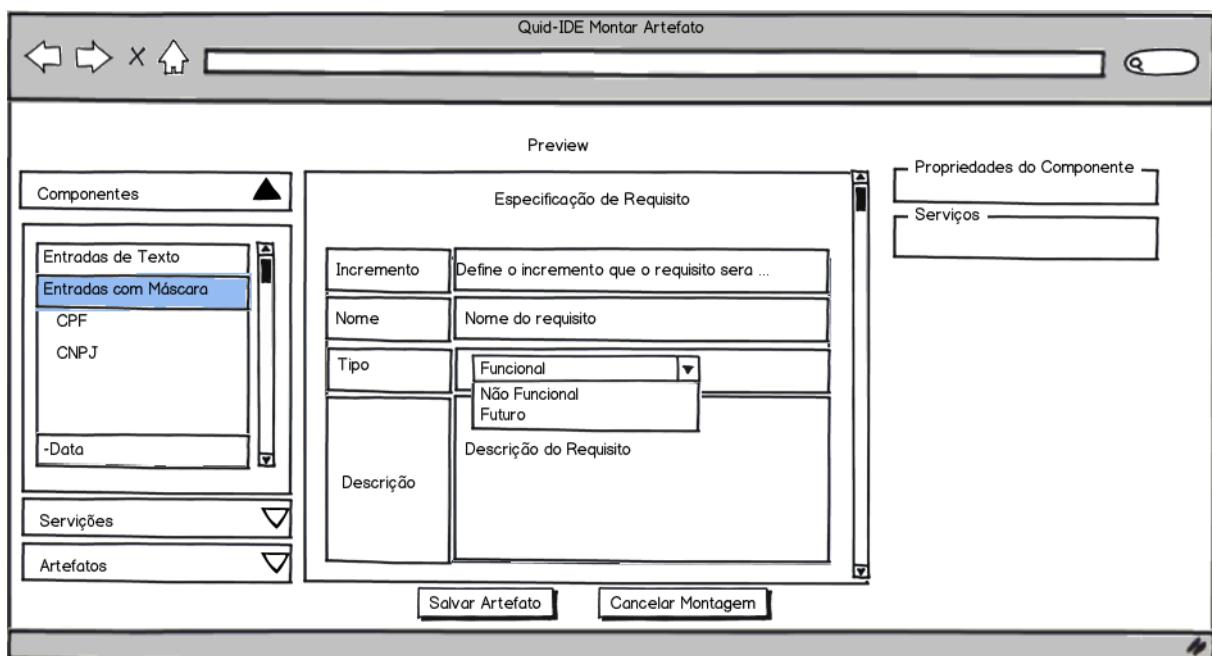


Figura 5 – Montar Artefato – Seleção de membros

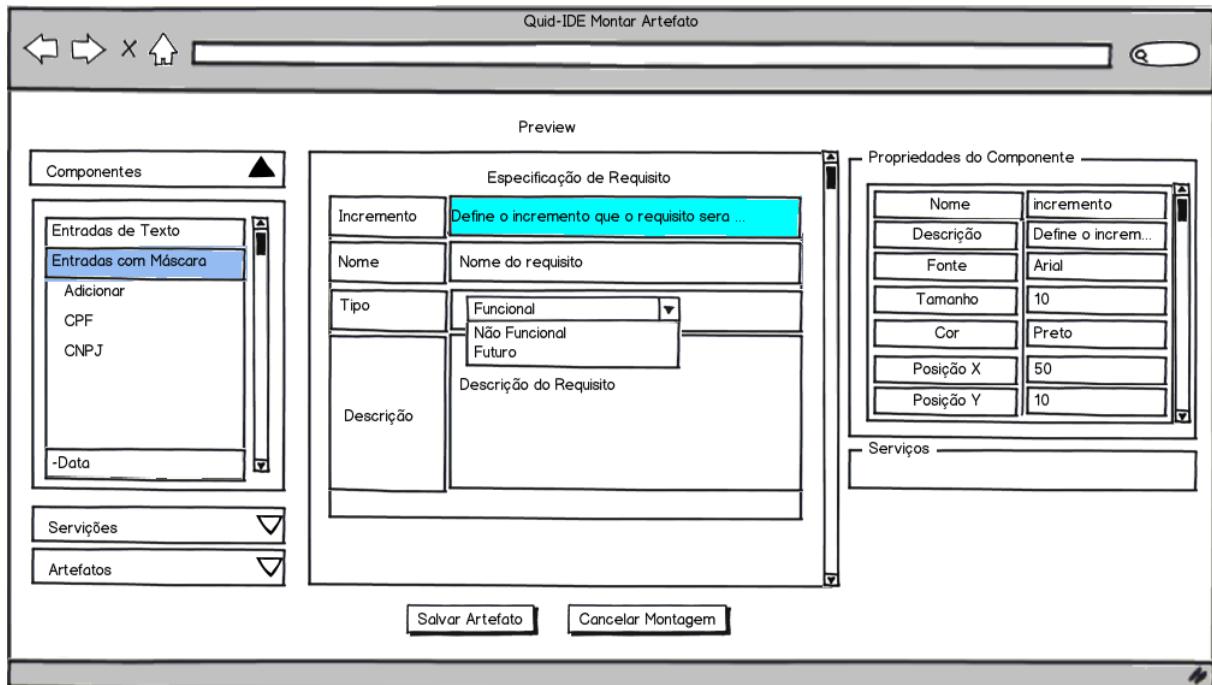


Figura 6 – Montar Artefato – Propriedades do membro selecionado

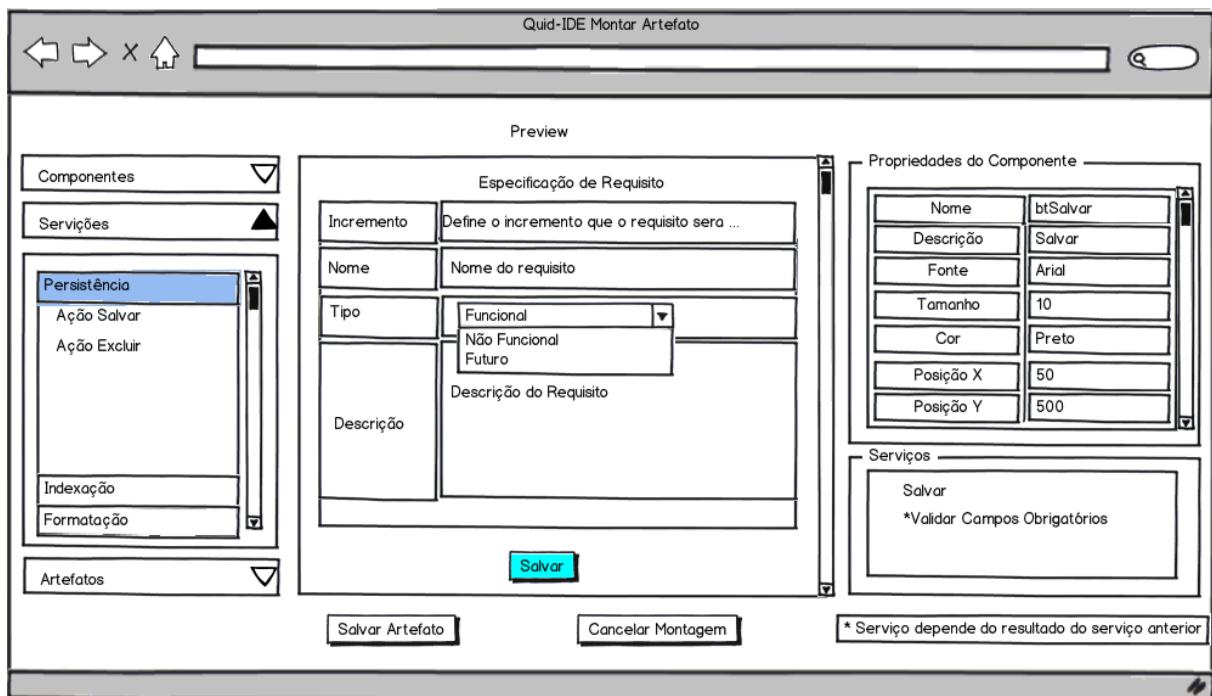


Figura 7 – Montar Artefato – Propriedades do membro representando um serviço

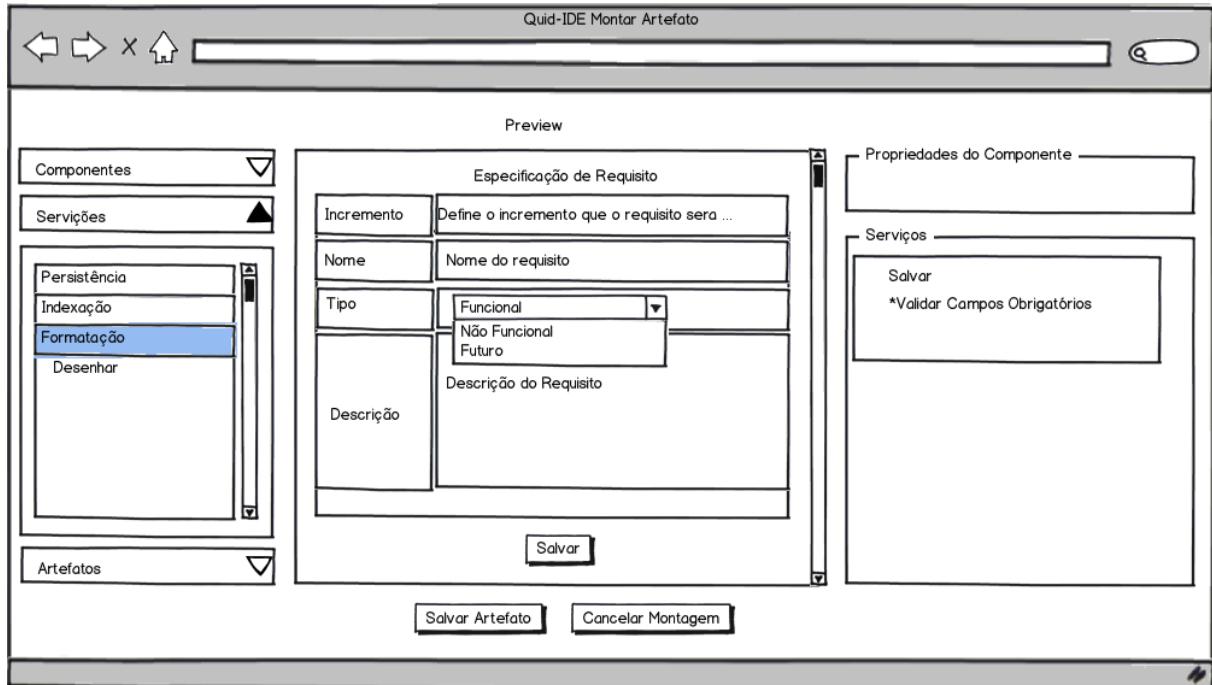


Figura 8 – Montar Artefato – Seleção de serviço

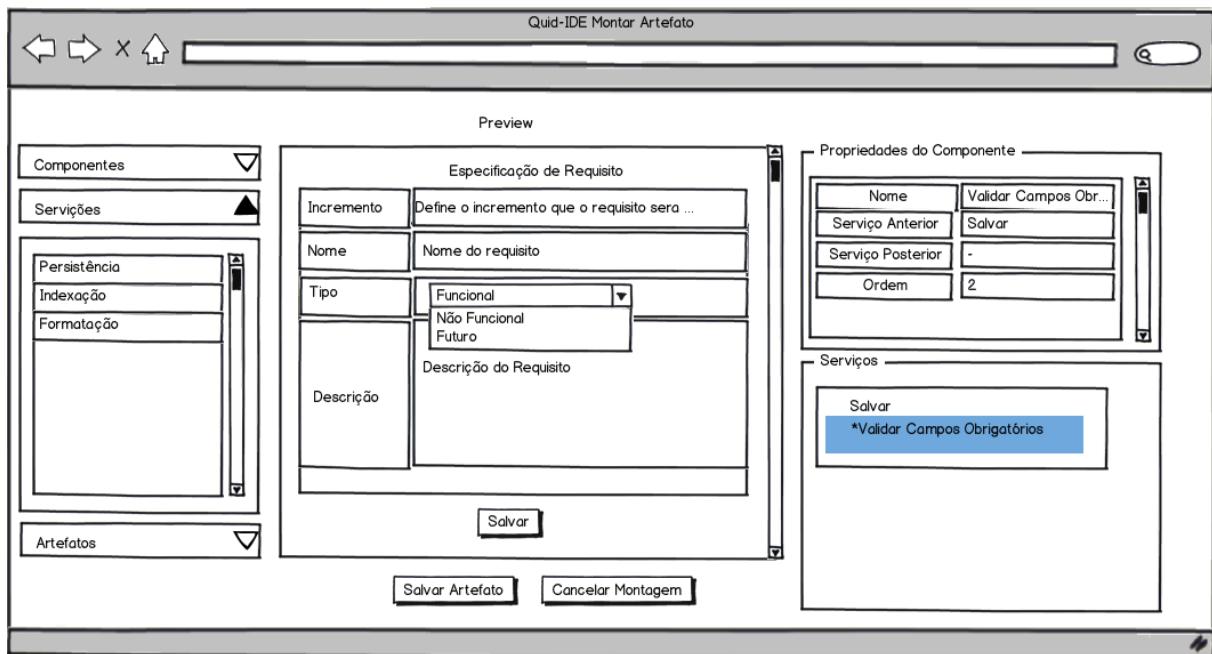


Figura 9 – Montar Artefato – Propriedades do serviço selecionado

5.4.3 R04 – Informações Relacionadas

As informações referentes a este requisitos foram dispostas nos requisitos:

- R03 – Manter Artefato
- R02 – Manter Membro

- R11 – Manter Serviço

5.4.4 R04 – Funcionalidades Relacionadas

Ação	Funcionalidade
Adicionar Membro	Adiciona um membro ao artefato que esta sendo criado.
Remover Membro	Remove o membro selecionado do artefato que esta sendo criado.
Adicionar Serviço	Adiciona um serviço ao artefato que esta sendo criado.
Remover Serviço	Remove o serviço selecionado do artefato que esta sendo criado.
Visualizar Artefato	Exibe o artefato que esta sendo montado na visao de visualização.

Tabela 40 – Funcionalidades relacionadas ao requisito R04

5.5 Especificação de Requisitos n.º 05

5.5.1 R05 – Clonar Artefato

Incremento	Primeiro Incremento
Nome	Clonar Artefato
Tipo	Futuro
Definição	<p>Requisito responsável por realizar uma copia de um artefato A para um artefato B. Esta copia mantêm os mesmos membros, serviços e ordem de execução dos serviços do artefato gerador, modificando somente o nome e a descrição do artefato clonado.</p> <p>Depois de clonado, o usuário montador poderá realizar qualquer alteração dos membros e serviços do artefato.</p> <p>O artefato clonado não terá vinculo com nenhum modelo cadastrado no framework, mesmo que o artefato <gerador> este em algum modelo.</p>
Regras de negócio associadas	RN04

Tabela 41 – Tabela de especificação do requisito R05

5.5.2 R05 – Protótipo de Interface

Figura 10 – Clonar Artefato

5.5.3 R05 – Informações Relacionadas

As informações referentes a este requisitos foram dispostas nos requisitos:

- R03 – Manter Artefato

5.5.4 R05 – Funcionalidades Relacionadas

Ação	Funcionalidade
Clonar	Clona um artefato selecionado pelo usuário.

Tabela 42 – Funcionalidades relacionadas ao requisito R05

5.6 Especificação de Requisitos n.º 06

5.6.1 R06 – Preencher Artefato

Incremento	Primeiro Incremento
Nome	Preencher Artefato
Tipo	Funcional
Definição	Consiste no preenchimento dos Membros não estáticos do artefato do documento, pelo usuário preenchedor. Em cada membro a ser preenchido, deverá aparecer a descrição

	<p>informada pelo usuário montador (Ver Montar Artefato). Quando o usuário preenchedor clicar sobre o membro, o sistema deve informar uma interface para o preenchimento do Membro (Ex: uma entrada de texto, um combo de seleção, etc.). Após este preenchimento, o artefato valida a entrada de dados e substitui a descrição do membro pelo valor que o usuário preencheu, aplicada as propriedades definidas pelo membro (formatação, posicionamento).</p> <p>Ao ser disparada qualquer evento, através dos TipoMembro (Ver TipoMembro disparadores de ações, em Manter TipoMembro), o artefato deverá executar a seqüência dos serviços definidos na montagem do artefato.</p> <p>Existem serviços que finalizam o preenchimento do artefato. Tais serviços serão denominados serviços finalizadores como (Persistencia). Ao finalizar o preenchimento o framework deverá informar uma prévia do artefato preenchido.</p>
Regras de negócio associadas	RN10

Tabela 43 – Tabela de especificação do requisito R06

5.6.2 R06 – Protótipo de Interface

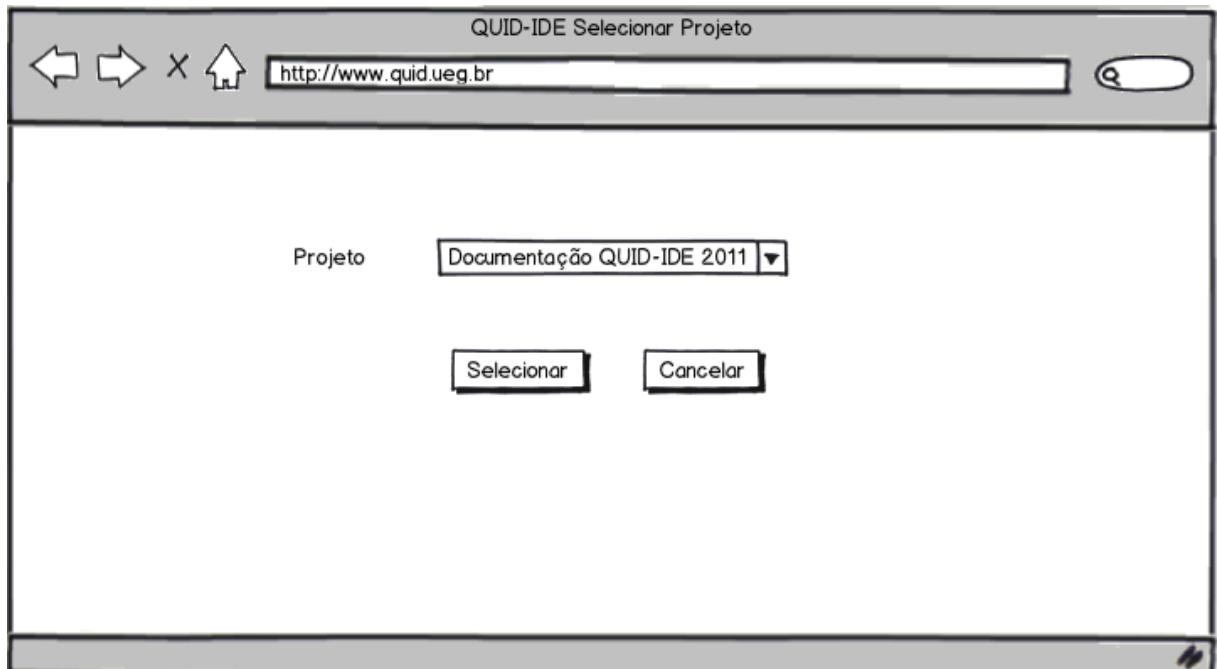


Figura 11 – Preencher Artefato – Seleção do Projeto

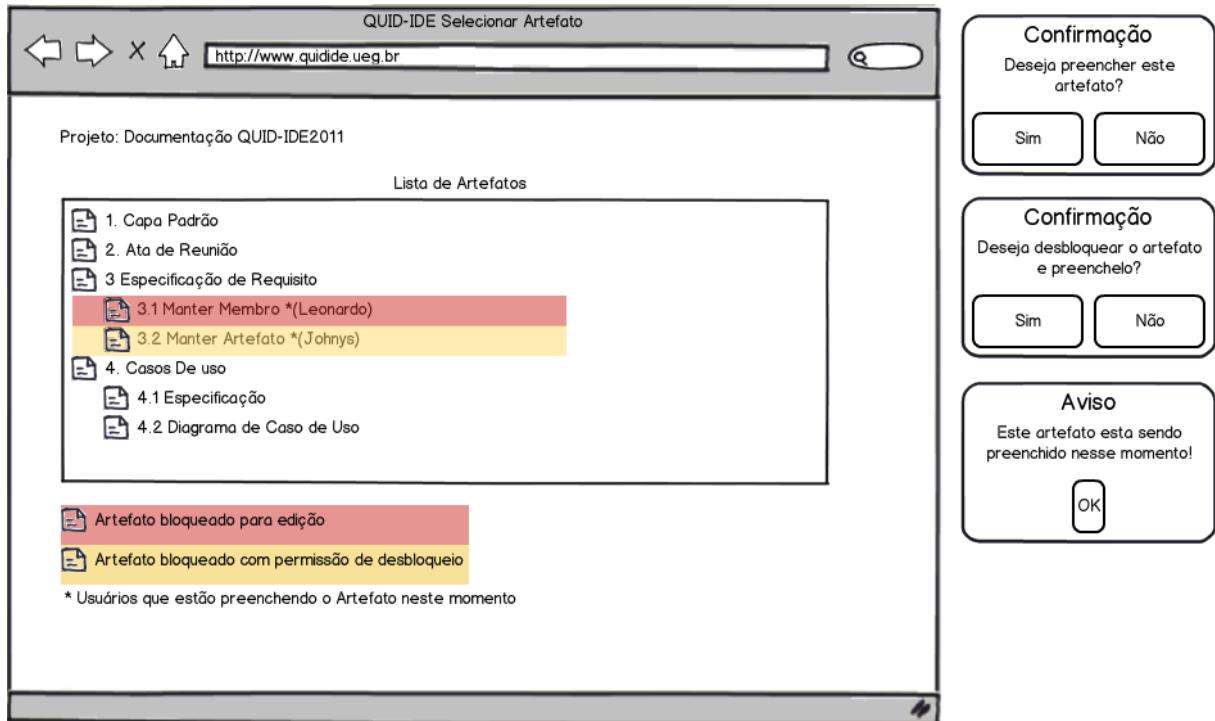


Figura 12 – Preencher Artefato – Árvore de artefatos (Modelo)

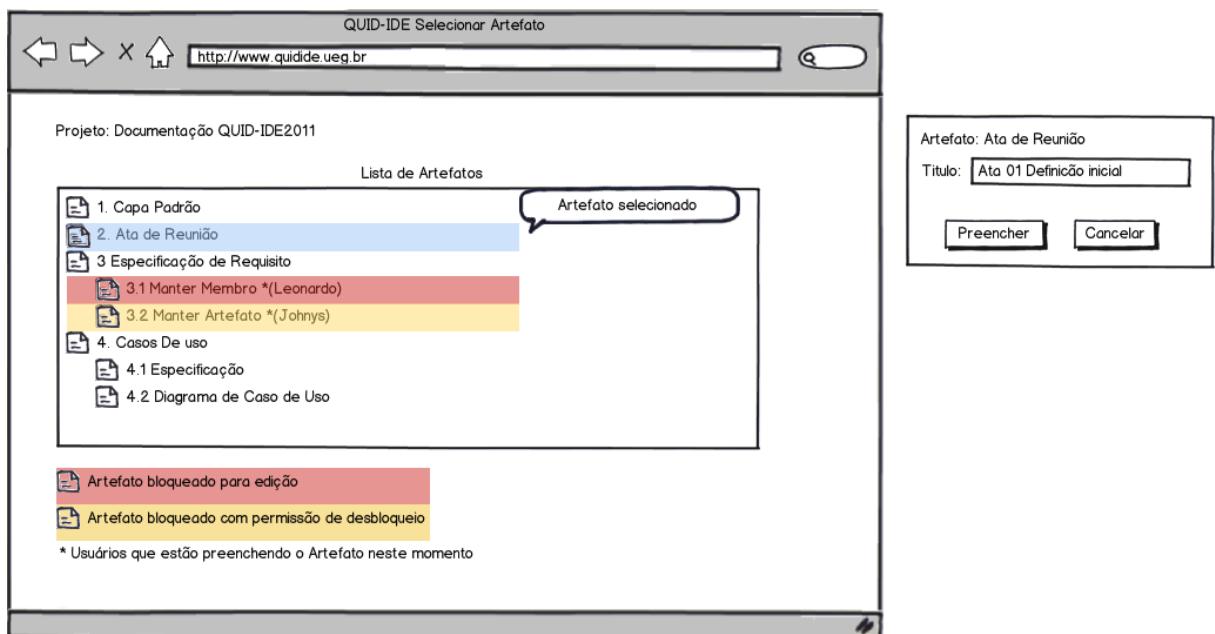


Figura 13 – Preencher Artefato – Novo artefato

Manter Usuario
http://www.quidide.ueg.br

Visualizar

Titulo: Ata 01 Definição inicial

Equipe	Johnys / Leonardo		
Data:	10/10/2011	Local	UNUCET
Clientes:	Clientes envolvidos		
Teor:	Descrição da Ata de Reunião		

Salvar Cancelar

Figura 14 – Preencher Artefato – Tela de preenchimento

5.6.3 R06 – Informações Relacionadas

As informações referentes a este requisitos foram dispostas nos requisitos:

- R03 – Manter Artefato
- R01 – Manter TipoMembro

5.6.4 R06 – Funcionalidades Relacionadas

Ação	Funcionalidade
Cancelar Preenchimento	Cancela o preenchimento do artefato.
Visualizar Artefato	Visualiza o artefato em modo de impressão. (Ver Imprimir Artefato)
Visualizar Interface de Preenchimento	Informa ao usuário a interface de preenchimento do artefato.
Executar Ação TipoMembro disparador	Executa uma ação disparada pelo TipoMembro disparador.

Tabela 44 – Funcionalidades relacionadas ao requisito R06

5.7 Especificação de Requisitos n.º 07

5.7.1 R07 – Visualizar Artefato

Incremento	Primeiro Incremento
Nome	Visualizar Artefato
Tipo	Funcional
Definição	Exibe ao usuário preenchedor a previa do modelo de impressão do artefato preenchido ou uma previa do artefato montado ao usuário montador (Nota: no caso do usuário montador, os valores dos membros serão as descrições informadas no momento de montagem do artefato). Esta visualização será feita de acordo com a tecnologia de visão que esta sendo utilizada (Ex: Visão WEB: utilizara HTML, Visão desktop: um editor de relatório).
Regras de negócio associadas	

Tabela 45 – Tabela de especificação do requisito R07

5.7.2 R07 – Protótipo de Interface

A manutenção de artefatos é feita através das telas relacionadas aos requisitos:

- R04 – Montar Artefato
- R06 – Preencher Artefato

5.7.3 R07 – Informações Relacionadas

As informações referentes a este requisitos foram dispostas nos requisitos:

- R03 – Manter Artefato
- R01 – Manter TipoMembro

5.7.4 R07 – Funcionalidades Relacionadas

Ação	Funcionalidade
Visualizar Artefato	Visualiza uma prévia do artefato que esta sendo montado/preenchido.

Tabela 46 – Funcionalidades relacionadas ao requisito R07

5.8 Especificação de Requisitos n.º 08

5.8.1 R08 – Publicar Artefato

Incremento	Primeiro Incremento
Nome	Publicar Artefato
Tipo	Futuro
Definição	<p>Requisito responsável por gerar um arquivo para a impressão do artefato para o usuário preenchedor.</p> <p>O framework disponibilizara o arquivo no formato LATEX ou em texto UNICODE. Caso o usuário deseje outros padrões, deve-se adicionar um novo serviço de impressão ao framework.</p>
Regras de negócio associadas	RN11

Tabela 47 – Tabela de especificação do requisito R08

5.8.2 R08 – Protótipo de Interface

Os protótipos deste requisito estão relacionados com os protótipos apresentados no requisito R05 – Preencher Artefato.

5.8.3 R08 – Informações Relacionadas

As informações referentes a este requisito foram dispostas nos requisitos:

- R03 – Manter Artefato
- R01 – Manter TipoMembro

5.8.4 R08 – Funcionalidades Relacionadas

Ação	Funcionalidade
Publicar Artefato	Gera um arquivo para a impressão do artefato.

Tabela 48 – Funcionalidades relacionadas ao requisito R08

5.9 Especificação de Requisitos n.º 09

5.9.1 R09 – Manter Modelo

Incremento	Primeiro incremento
Nome	Manter Modelo
Tipo	Funcional

Definição	<p>Requisito responsável por gerenciar os Modelos dentro do framework.</p> <p>O modelo é composto por um conjunto de artefatos que constituem o “template” ou molde do documento.</p> <p>Cada Modelo tem um nome de identificação e um texto descritivo.</p> <p>A montagem do modelo deve ser representada a partir de um “Tree View” (Arvore de visualização). Em cada nó da arvore deverá conter as seguintes informações:</p> <ul style="list-style-type: none"> a) O artefato que representa. b) O grau do artefato na árvore do modelo. O grau indica o nível do artefato dentro do Modelo. c) A ordem de aparecimento na árvore do modelo. A ordem define a posição de dois artefatos de mesmo grau dentro do Modelo. d) Multiplicidade (quantidades de vezes que se pode repetir o Artefato dentro do modelo (uma vez ou muitas). e) Ordem de Preenchimento: informará ao framework qual deverá ser a ordem de preenchimento dos Artefatos no Modelo. f) Artefatos que são auto-criados (listas, índices, etc.) não terão ordem de preenchimento, e consequentemente não serão disponibilizados para o preenchimento. <p>Ao se adicionar um artefato no modelo, verifica-se se ele contém uma dependência com outros artefatos. Caso exista a dependência, o framework solicitará que o usuário montador do Modelo que utilize os outros artefatos dependentes.</p> <p>Em cada no da arvore de visualização, aparecerá o atributo título do artefato.</p> <p>Após o preenchimento dos no da arvore, o usuário poderá realizar o workflow do Modelo (Ver requisito Manter Workflow).</p>
Regras de negócio associadas	RN12, RN13, RN14, RN15, RN16, RN17

Tabela 49 – Tabela de especificação do requisito R09

5.9.2 R09 – Protótipo de Interface

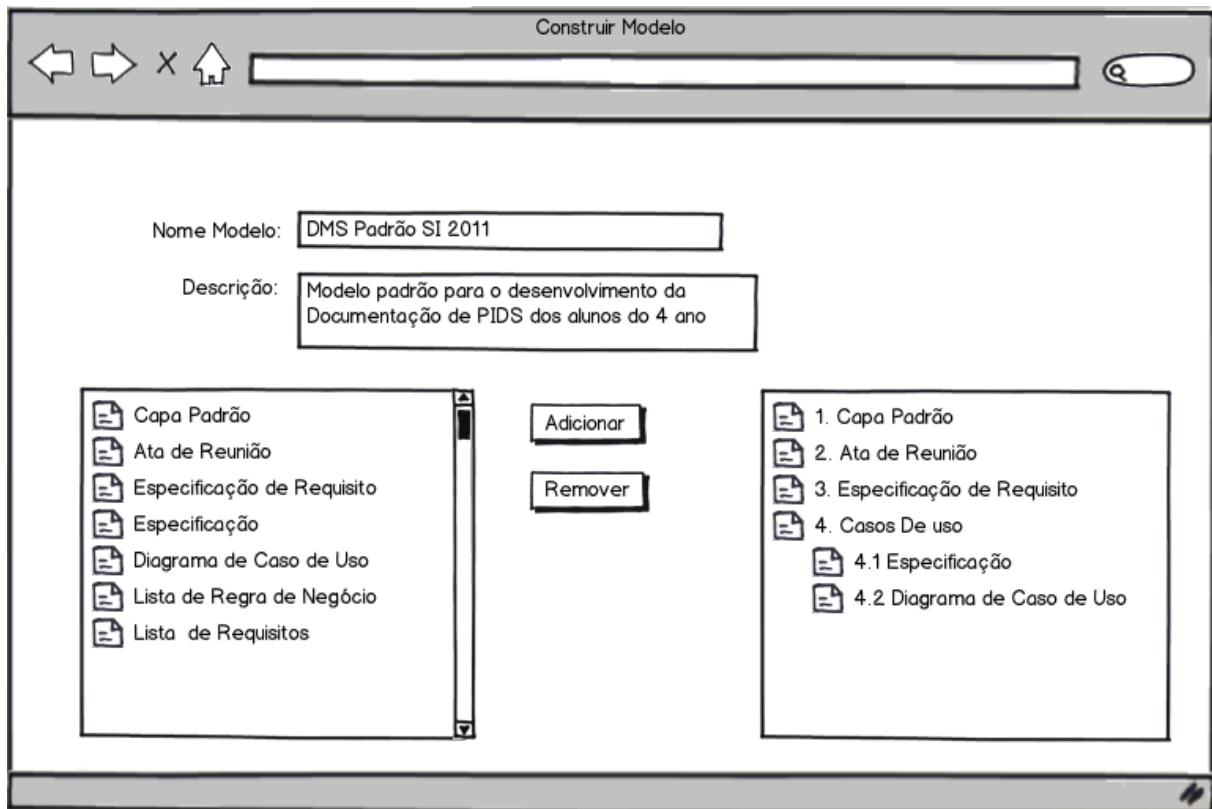


Figura 15 – Montar Modelo – Seleção dos artefatos

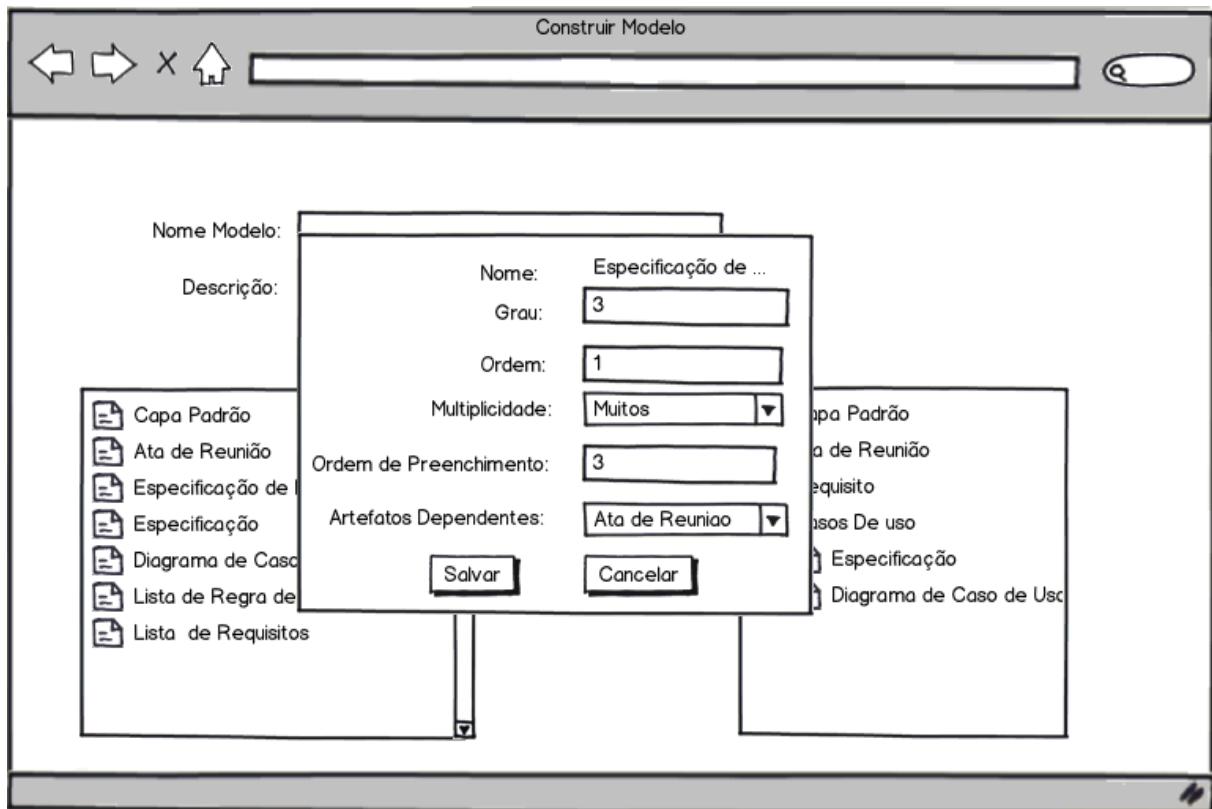


Figura 16 – Montar Modelo – Configuração de cada artefato adicionado

5.9.3 R09 – Informações Relacionadas

Item	Tipo de Dado	Tamanho	Definido pelo Sistema	Obrigatório	Valor inicial	Domínio	Máscara
codigo	Integer	-	Não	Sim	-	-	-
descricao	String	255	Não	Não	-	-	-
nome	String	255	Não	Sim	-	-	-
artefatos	Collection<Artefato>	-	Não	Sim	-	-	-

Tabela 50 – Informações relacionadas ao requisito R09

5.9.4 R09 – Funcionalidades Relacionadas

Ação	Funcionalidade
Montar Modelo	Cria um modelo a partir do adicionamento dos artefatos na estrutura da arvore de visualização.
Remover Modelo	Remove um Modelo do framework.
Clonar Modelo	Cria um novo Modelo a partir de um modelo cadastrado no sistema, copiando toda a sua estrutura (Nos da Arvore), bem como sua definição do workflow.
Alterar Modelo	Inclui/Exclui artefatos dentro do framework, e possibilita a alteração dos valores dos nos da arvore de visualização.
Mapear Modelo	Salva um modelo no framework.

Tabela 51 – Funcionalidades relacionadas ao requisito R09

5.10 Especificação de Requisitos n.º 10

5.10.1 R10 – Manter Projeto

Incremento	Primeiro Incremento
Nome	Manter Projeto
Tipo	Funcional
Definição	Requisito que gerencia os projetos cadastrados dentro do framework. Um projeto é o preenchimento de um Modelo por um grupo de usuários cadastrados no sistema (Ver requisito gerenciamento de Usuário).

	<p>Cada projeto terá um nome que o identifica dentro do framework.</p> <p>Cada usuário do grupo poderá preencher/editar os artefatos dentro do modelo, seguindo as regras definidas no workflow.</p> <p>O framework deverá manter a integridade referencial dos artefatos preenchidos, possibilitando o travamento de um artefato (bloqueio de um artefato para que nenhum outro usuário do grupo edite o documento), mostrando a informação de quem está editando o documento.</p> <p>O framework deverá estabelecer um limite de tempo (parametrizável), que um artefato poderá ficar aberto e sem alteração por um determinado usuário. Passado este tempo, o framework deverá desbloquear o artefato e notificar ao usuário que estava modificando o documento.</p> <p>O projeto só poderá ser removido do framework caso ele não tenha nenhum artefato preenchido. Caso contrário, o usuário poderá inativar o projeto, impedindo que os usuários do grupo continuem preenchendo o projeto. Caso algum usuário estiver preenchendo o documento, o framework desbloquearia o artefato e bloquearia o documento, notificando o usuário preenchedor do artefato que o projeto foi bloqueado.</p>
Regras de negócio associadas	RN18, RN19, RN21

Tabela 52 – Tabela de especificação do requisito R10

5.10.2 R10 – Protótipo de Interface

Quid-IDE Criar Projeto

http://quidide.ueg.br

Criar Novo Projeto

Modelo: Template SI 2010

Equipe: GRUPO GED

Nome Projeto: Documentação SI 2011

Criar Projeto Cancelar

Figura 17 – Manter Projeto – Criar Projeto

5.10.3 R10 – Informações Relacionadas

Item	Tipo de Dado	Tamanho	Definido pelo Sistema	Obrigatório	Valor inicial	Domínio	Máscara
codigo	Integer	-	Não	Sim	-	-	-
descricao	String	255	Não	Não	-	-	-
nome	String	255	Não	Sim	-	-	-
modelo	Modelo	-	Não	Sim	-	-	-
equipe	Equipe	-	Não	Sim	-	-	-
status	String	255	Não	Sim	ATIVO	ATIVO / INATIVO	-

Tabela 53 – Informações relacionadas ao requisito R10

5.10.4 R10 – Funcionalidades Relacionadas

Ação	Funcionalidade
Mapear projeto	Realiza o cadastro de um projeto no framework.
Remover projeto	Remove um projeto do framework.
Preencher projeto	Permite seleção do artefato que será preenchido pelo usuário.

Tabela 54 – Funcionalidades relacionadas ao requisito R10

5.11 Especificação de Requisitos n.º 11

5.11.1 R11 – Manter Serviços

Incremento	Primeiro Incremento
Nome	Manter Serviços
Tipo	Futuro
Definição	Os serviços representam as ações dos artefatos. Tudo que um artefato faz será especificado em um serviço. Isto implica que um artefato possuirá um conjunto de serviços, executados de acordo com a definição da sequência do usuário (detalhes sobre a ordem de execução está melhor detalhada na definição do requisito MANTER ARTEFATO). Exemplos de serviços são a persistência de artefatos e a dependência entre artefatos. Estes serão

	<p>melhores explicados na definição dos requisitos 12 e 13 respectivamente. Cada serviço será capaz de executar uma única ação, ou seja, uma responsabilidade. Serviços que passam idéia de várias ações diferentes devem ser quebrados em vários serviços.</p> <p>Para padronização, serviços devem ser capaz de fornecer sua lista de tipos parâmetros de entradas para a execução de sua ação. Esta lista deve ser compatível com o resultado da ação, tornando possível utilizar a ação de um serviço como entrada de outro serviço.</p> <p>Este requisito tem característica genérica visando atender o objetivo de extensibilidade do software tornando possível a adição de qualquer serviço desde que este possua as características abaixo:</p> <ul style="list-style-type: none"> • Nome (identificador único no framework) • Descrição • TipoMembro • Ação • Versão • Revisão • Lista de tipos de parâmetros de entrada para execução da ação. • Resultado da ação executada compatível com a lista de entrada e incluindo uma lista de mensagens. <p>É de responsabilidade de cada serviço validar os parâmetros de entrada passados para execução de sua funcionalidade.</p> <p>Assim como o TipoMembro, o serviço deve ter o controle de versões em dois níveis:</p> <ul style="list-style-type: none"> • Nível 1 (Versão): Garante a versão de um Serviço, com compatibilidade. Caso seja uma nova versão, deve-se realizar um novo mapeamento, como se fosse um novo Serviço, com o mesmo nome, porém com uma versão diferente. • Nível 2 (Revisão): Garante a compatibilidade com a versão anterior (atualizações menores). Caso seja uma nova revisão, haverá somente a atualização do Serviço no cadastro do framework, substituindo a implementação anterior.
Regras de negócio associadas	RN20

Tabela 55 – Tabela de especificação do requisito R11

5.11.2 R11 – Protótipo de Interface

Os protótipos relacionados a este requisito são os mesmos apresentados no requisito R01 – Manter TipoMembro.

5.11.3 R11 – Informações Relacionadas

Item	Tipo de Dado	Tamanho	Definido pelo Sistema	Obrigatório	Valor inicial	Domínio	Máscara
codigo	Integer	-	Não	Sim	-	-	-
descricao	String	255	Não	Não	-	-	-
nome	String	255	Não	Sim	-	-	-
versao	Integer	-	Não	Sim	-	-	-
revisao	Integer	-	Não	Sim	-	-	-
ativo	Boolean	-	Não	Sim	true	-	-
aguardandoRemocao	Boolean	-	Não	Não	false	-	-

Tabela 56 – Informações relacionadas ao requisito R11

5.11.4 R11 – Funcionalidades Relacionadas

Ação	Funcionalidade
Mapear Serviço	O framework deve ser capaz de mapear todos os serviços existentes, ou seja, identificar todos os serviços existentes no repositório do framework e relacioná-los para futura utilização. Para isto, deve-se permitir a gravação do arquivo onde está implementado o serviço, armazenando-o em um local específico.
Remover Serviço	A remoção de serviços será feita de duas maneiras: <ul style="list-style-type: none"> remover serviços que não estejam sendo utilizados por nenhum artefatos através da rastreabilidade, onde o arquivo que está implementado o serviço está ausente do repositório, informando ao usuário os locais de impacto de sua ausência (Artefatos, Modelos). A remoção de Serviço também poderá ser realizada pelo usuário, através de uma funcionalidade fornecida pelo framework.
Pesquisar Serviço	Com a necessidade do usuário de ter conhecimento sobre os serviços existentes, o framework deve ser capaz de pesquisar-los.

Tabela 57 – Funcionalidades relacionadas ao requisito R11

5.12 Especificação de Requisitos n.º 12

5.12.1 R12 – Gerenciar Serviço de Persistência

Incremento	Primeiro Incremento
Nome	Gerenciar Serviço de Persistência
Tipo	Futuro
Definição	<p>Persistência é uma especialização de serviço, obrigando-a que implemente as funcionalidades de tal. Persistir é o ato de armazenar dados para manipulação posterior, portanto, este requisito se responsabiliza por criar e validar a estrutura de armazenamento (tabelas, arquivos), caso seja necessário, e armazenar os dados baseando-se na estrutura dos artefatos. O framework contará como serviço nativo a persistência em banco de dados modelo relacional e em arquivo, seguindo o padrão XML v1.0.</p> <p>Para que ocorra a portabilidade dos dados entre máquinas distintas, a estrutura do artefato (atributos, lista de membros e serviços) também deve ser salva, impedindo que ocorra erros na compatibilidade de artefatos.</p> <p>Além das propriedades herdadas do serviço, adiciona-se:</p> <ul style="list-style-type: none"> • Artefato • Tipo (inserir, alterar, remover, pesquisar)
Regras de negócio associadas	

Tabela 58 – Tabela de especificação do requisito R12

5.12.2 R12 – Protótipo de Interface

Não se aplica a este requisito.

5.12.3 R12 – Informações Relacionadas

As informações referentes a este requisitos foram dispostas nos requisitos:

- R11 – Manter Artefato

5.12.4 R12 – Funcionalidades Relacionadas

Ação	Funcionalidade
Incluir Componente	Realiza a inserção de um novo artefato na estrutura de armazenamento.
Alterar Componente	Altera o artefato específico na estrutura de armazenamento.
Remover Componente	Exclui o artefato da estrutura de armazenamento.
Pesquisar Componente	Retorna todos os artefatos atendem as condições do artefato passado como

	parâmetro.
--	------------

Tabela 59 – Funcionalidades relacionadas ao requisito R12

5.13 Especificação de Requisitos n.º 13

5.13.1 R13 – Gerenciar Serviço de Dependências

Incremento	Primeiro Incremento
Nome	Gerenciar Serviço de Dependências
Tipo	Futuro
Definição	<p>Dependência é uma especialização de serviço. As dependências são as relações entre os artefatos, portanto, este requisito visa obrigar que cada dependência seja respeitada no ato da montagem e/ou preenchimento de um artefato. Como cada tipo de dependência exige que uma ação específica seja executada, ela é considerada um serviço do artefato, impondo assim que ela implemente as funcionalidades de um serviço.</p> <p>Além das propriedades herdadas do serviço, adiciona-se:</p> <ul style="list-style-type: none"> • Artefato gestor da dependencia (artefato que possui o serviço) • Artefato que depende (dependente) • Tipo (criação, composição, listagem, preenchimentos) <p>Há a necessidade de descobrir quais artefatos se relacionam com um artefato em específico. Determinar isto é essencial para a montagem da TreeView e avaliar o impacto de alterações em artefatos.</p> <p>Até o momento foram identificados cinco tipos de dependências, os nomes foram baseados em sua responsabilidade:</p> <ul style="list-style-type: none"> • <i>Dependência de criação</i>: Acontece quando um Artefato-A obriga a criação de um Artefato-B, sendo que este Artefato-B só pode ser criado a partir de um Artefato-A. O TipoMembro que o representa deve disponibilizar a escolha de um dos Artefato-B já existentes ou a criação de um novo Artefato-B, validando a duplicidade dos dados. Esta dependência implica que a multiplicidade entre os artefatos é de muitos para muitos. • <i>Dependência de composição</i>: Acontece quando um Artefato-A possui um Artefato-B já existente. O TipoMembro que o representa deve disponibilizar todos os Artefatos-B existentes para a escolha de um. O limite de um artefato escolhido define a multiplicidade entre os

	<p>artefatos, de um para um. Características de preenchimento, como obrigatoriedade, também são definidas na dependência.</p> <ul style="list-style-type: none"> • <i>Dependência de listagem-parcial:</i> Acontece quando um Artefato-A possui muitos Artefato-B já existentes, sem opção de criação de um novo Artefato-B. Assim a multiplicidade é definida de um para muitos. • <i>Dependência de listagem-total:</i> Acontece quando um Artefato-A possui todos os Artefato-B existentes. Assim a multiplicidade seria de um para muitos, porém obrigando o relacionamento com todos todos os Artefato-B existentes. • <i>Dependência de ordem de preenchimento:</i> Acontece quando um Artefato-A só pode ser preenchido após o preenchimento do Artefato-B. Esta dependência resolveria questões de preenchimento onde um artefato não depende em termos de multiplicidade, forçando o usuário a seguir a ordem definida pelo usuário montador através desta dependência. Não há necessidade de um TipoMembro para representar esta dependência, sendo que sua ação é simplesmente verificar a possibilidade de preenchimento e, em caso de impedimento, retornar uma mensagem de erro.
Regras de negócio associadas	RN22

Tabela 60 – Tabela de especificação do requisito R13

5.13.2 R13 – Protótipo de Interface

Não se aplica a este requisito.

5.13.3 R13 – Informações Relacionadas

As informações referentes a este requisitos foram dispostas nos requisitos:

- R11 – Manter Artefato

5.13.4 R13 – Funcionalidades Relacionadas

Ação	Funcionalidade
Pesquisar Dependência	Pesquisa as dependências disponíveis no framework.
Validar Possibilidade de Dependência	Valida a possibilidade de existência da dependência entre artefatos.

Criar Depêndencia entre Artefatos	Define que um artefato depende de outro de acordo com as características de cada especialização de dependência descritas na definição do requisito, disponibilizada acima.
-----------------------------------	--

Tabela 61 – Funcionalidades relacionadas ao requisito R13

5.14 Especificação de Requisitos n.º 14

5.14.1 R14 – Manter Log das Operações do Sistema

Incremento	Primeiro Incremento
Nome	Manter Log das Operações do Sistema
Tipo	Futuro
Definição	<p>Log são registros de eventos relevantes num sistema com finalidade de conhecimento de comportamento dos usuários e diagnóstico de problemas que ocorra na execução de componentes.</p> <p>O framework oferecerá quatro tipos de log, baseados nos níveis de operação:</p> <ul style="list-style-type: none"> • <i>Desenvolvimento</i>: envolve as ações do usuário Desenvolvedor, ou seja, a manipulação de TipoMembros e Serviços. Este nível não poderá ser desativado pelo gerente. • <i>Montagem</i>: envolve as ações do usuário Montador, ou seja, a manipulação de Membros, Artefatos e Modelos. • <i>Preenchimento</i>: envolve as ações do usuário Preenchedor, ou seja, a manipulação de Documentos. • <i>Exceção</i>: envolve os erros emitidos pelo framework, como falha na conexão com o banco de dados, falha ao salvar um arquivo ou qualquer falha que seja lançada a partir do framework, ou seja, ações que não estão ligadas diretamente à um usuário. Logs deste tipo não serão desativados em momento algum e serão gravados a parte dos demais. <p>O gerente poderá selecionar quais níveis ele deseja gravar no framework, podendo ele ativar e desativar somente os níveis Montagem e Preenchimento.</p> <p>Para melhor gerenciamento e usabilidade, os logs devem ser armazenados de acordo com características abaixo:</p> <ul style="list-style-type: none"> • Tipo • Usuário

	<ul style="list-style-type: none"> • Data • Mensagem • Elemento manipulado (documento, modelo, artefato, membro, tipomembro ou serviço) <p>Somente os usuários do tipo gerente poderão alterar as opções e visualizar os logs do sistema.</p>
Regras de negócio associadas	RN09

Tabela 62 – Tabela de especificação do requisito R14

5.14.2 R14 – Protótipo de Interface

Não se aplica a este requisito.

5.14.3 R14 – Informações Relacionadas

Item	Tipo de Dado	Tamanho	Definido pelo Sistema	Obrigatório	Valor inicial	Domínio	Máscara
codigo	Integer	-	Não	Sim	-	-	-
usuario	Integer	-	Não	Sim	-	-	-
tipoElemento	String	255	Não	Sim	-	-	-
codigoElemento	Integer	-	Não	Sim	-	-	-
mensagem	String	255	Não	Sim	-	-	-
data	Date	-	Não	Sim	-	-	-

Tabela 63 – Informações relacionadas ao requisito R14

5.14.4 R14 – Funcionalidades Relacionadas

Ação	Funcionalidade
Registrar Log	Registra em arquivo texto as características citadas acima, organizados pela data.
Ativar Log	Ativa o registro de logs do framework.
Desativar Log	Desativa o registro de logs do framework.
Exibir Log	Exibe ao usuário todas os logs registrados por elemento, data ou usuário.

Tabela 64 – Funcionalidades relacionadas ao requisito R14

5.15 Especificação de Requisitos n.º 15

5.15.1 R15 – Gerenciar Multusuários

Incremento	Primeiro Incremento
Nome	Gerenciar Multusuários
Tipo	Funcional
Definição	<p>Usuários são os atores que integram com o framework. Este requisito tem como responsabilidade fazer a persistência de usuários bem como controlar o acesso no sistema por meio de login e senha.</p> <p>Para manutenção dos usuários serão necessários como dados obrigatórios:</p> <ul style="list-style-type: none"> • Nome • Email • Senha <p>Para a equipe deverá ser informado um nome único e os integrantes da mesma, ou seja, não poderão existir equipes com o mesmo nome.</p> <p>No preenchimento de um documento há a possibilidade do envolvimento de uma equipe ou um grupo, portanto, além da manipulação de usuários, este requisito gerencia a criação de equipes.</p> <p>O sistema não permitirá a exclusão de usuários, apenas seu desativamento da equipe, para que os documentos preenchidos por ele no sistema não percam suas referências.</p>
Regras de negócio associadas	

Tabela 65 – Tabela de especificação do requisito R15

5.15.2 R15 – Protótipo de Interface

Quid-IDE Gerenciar Multusuários

Incluir / Alterar / Pesquisar Usuário

Nome:

E-mail:

Senha:

Administrador
 Gerente de Projetos
 Desenvolvedor
 Montador
 Preenchedor

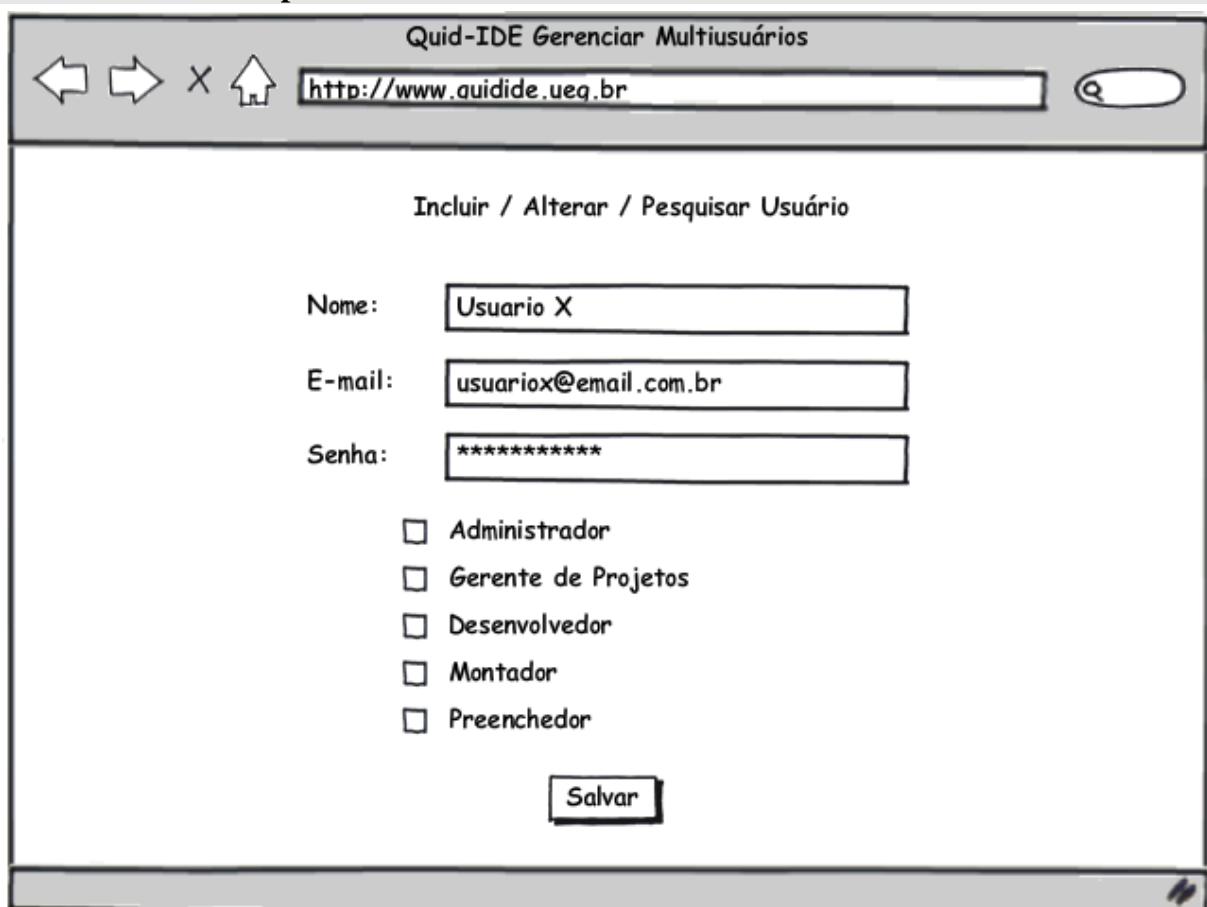


Figura 18 – Gerenciar Multusuários – CRUD usuário

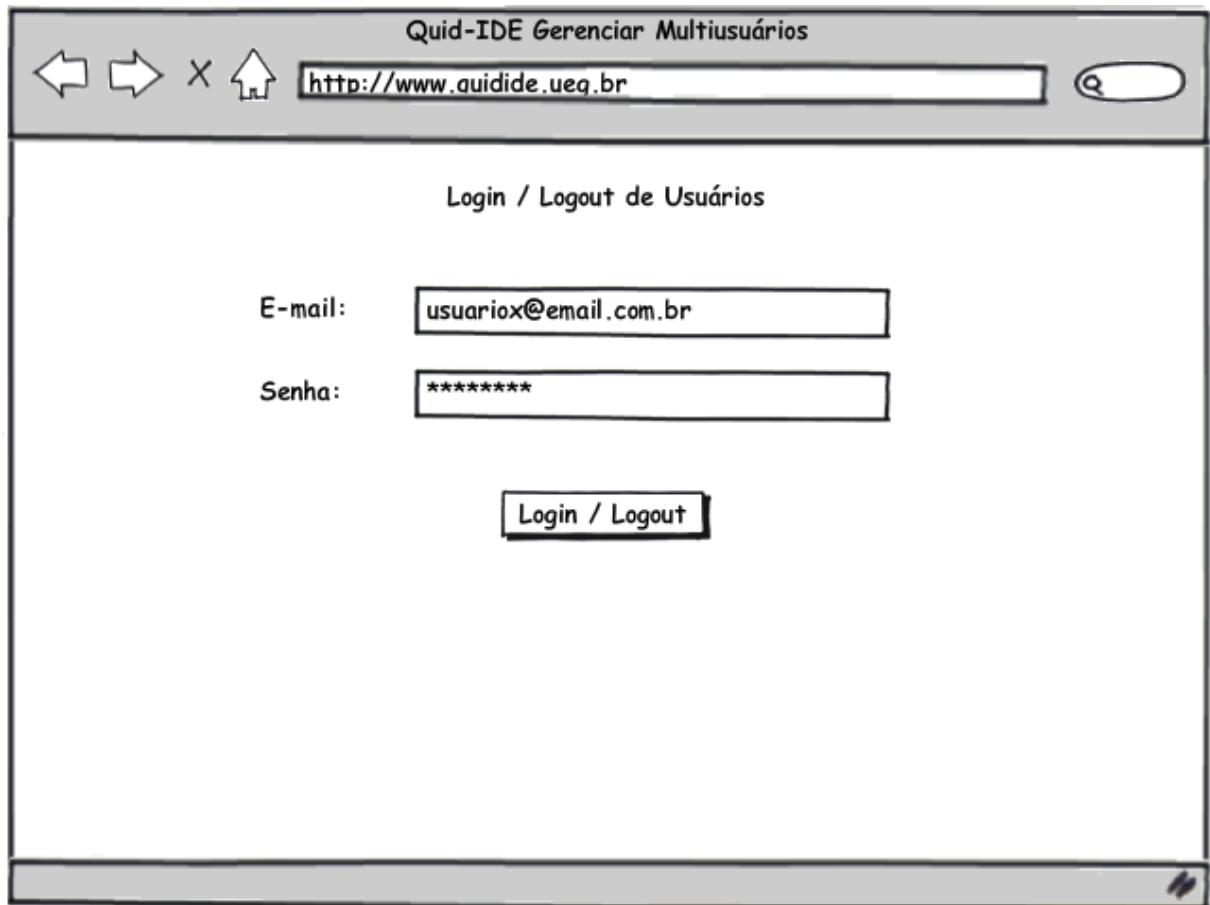


Figura 19 – Gerenciar Multusuários – Logon de usuário

Quid-IDE Gerenciar Multusuários

Incluir / Alterar / Pesquisar Equipe

Nome:

Perfil de Usuário: ▾

Nome	Perfil de Usuário	
Usuario X	Desenvolvedor	<input checked="" type="checkbox"/>
Usuario Y	Desenvolvedor	<input type="checkbox"/>
Usuario Z	Desenvolvedor	<input checked="" type="checkbox"/>

Figura 20 – Gerenciar Multusuários – CRUD equipe

5.15.3 R15 – Informações Relacionadas

Item	Tipo de Dado	Tamanho	Definido pelo Sistema	Obrigatório	Valor inicial	Domínio	Máscara
codigoEquipe	Integer	-	Não	Sim	-	-	-
nomeEquipe	String	25 5	Não	Sim	-	-	-
usuarios	Collection <Usuario>	-	Não	Sim	-	-	-
codigoUsuario	Integer	-	Não	Sim	-	-	-
nomeUsuario	String	25 5	Não	Sim	-	-	-
email	String	25	Não	Sim	-	-	-

		5					
status	String	25 5	Não	Sim	ATIVO	ATIVO / INATIVO	-
ativo	Boolean	-	Não	Sim	true	-	-
perfilAcesso	String	-	Não	Não	-	ADMINISTRADOR / GERENTE / DESENVOLVEDOR / MONTADOR / PREENCHEDOR	-

Tabela 66 – Informações relacionadas ao requisito R15**5.15.4 R15 – Funcionalidades Relacionadas**

Ação	Funcionalidade
Controle de Acesso	Requisito voltado para a segurança, autenticidade e sigilo dos dados inseridos no framework. Por base em um login e senha, permite o acesso ao sistema.
Inserir Usuário	Permite o cadastramento de usuários no sistema.
Desativar Usuário	Permite a pesquisa de usuários cadastrados no sistema.
Alterar Usuário	Desativa um usuário do sistema.
Gerenciar Grupos	Permite a edição dos dados de um usuário.

Tabela 67 – Funcionalidades relacionadas ao requisito R15**6 Modelagem de Casos de Uso****6.1 Relação de Casos de Uso**

Identificação	Caso de Uso
CDU01	Manter Artefato
CDU02	Montar Artefato
CDU03	Clonar Artefato
CDU04	Preencher Artefato
CDU05	Visualizar Artefato
CDU06	Publicar Artefato
CDU07	Manter Membro
CDU08	Manter TipoMembro
CDU09	Manter Modelo

CDU10	Manter Projeto
CDU11	Manter Serviço
CDU12	Gerenciar Serviço
CDU13	Realizar Versionamento
CDU14	Manter Opções de Log das Operações do Sistema
CDU15	Gerenciar Multusuários
CDU16	Visualizar Log das Operações do Sistema
CDU17	Registrar Log das Operações do Sistema

Tabela 68 – Lista de casos de uso

6.2 Relação de Atores

Nome	Definição
Preenchedor	Usuário que aplica o conhecimento para preencher os projetos já existentes.
Montador	Usuário que passa o modelo (template) do papel para o sistema.
Desenvolvedor	Usuário responsável pela criação de novos componentes (TipoMembro e Serviço) para o framework.
Gerente de Projeto	Usuário responsável pela criação de equipes e manipulação dos logs de operações do sistema.
Administrador	Usuário responsável por cadastrar outros usuários no sistema.

Tabela 69 – Lista atores do projeto

6.3 Diagrama Geral de Casos de Uso

Mesmo se tratando de um framework com uma complexidade entre média e alta, não houve necessidade de se dividir os casos de uso em categorias ou subsistemas.

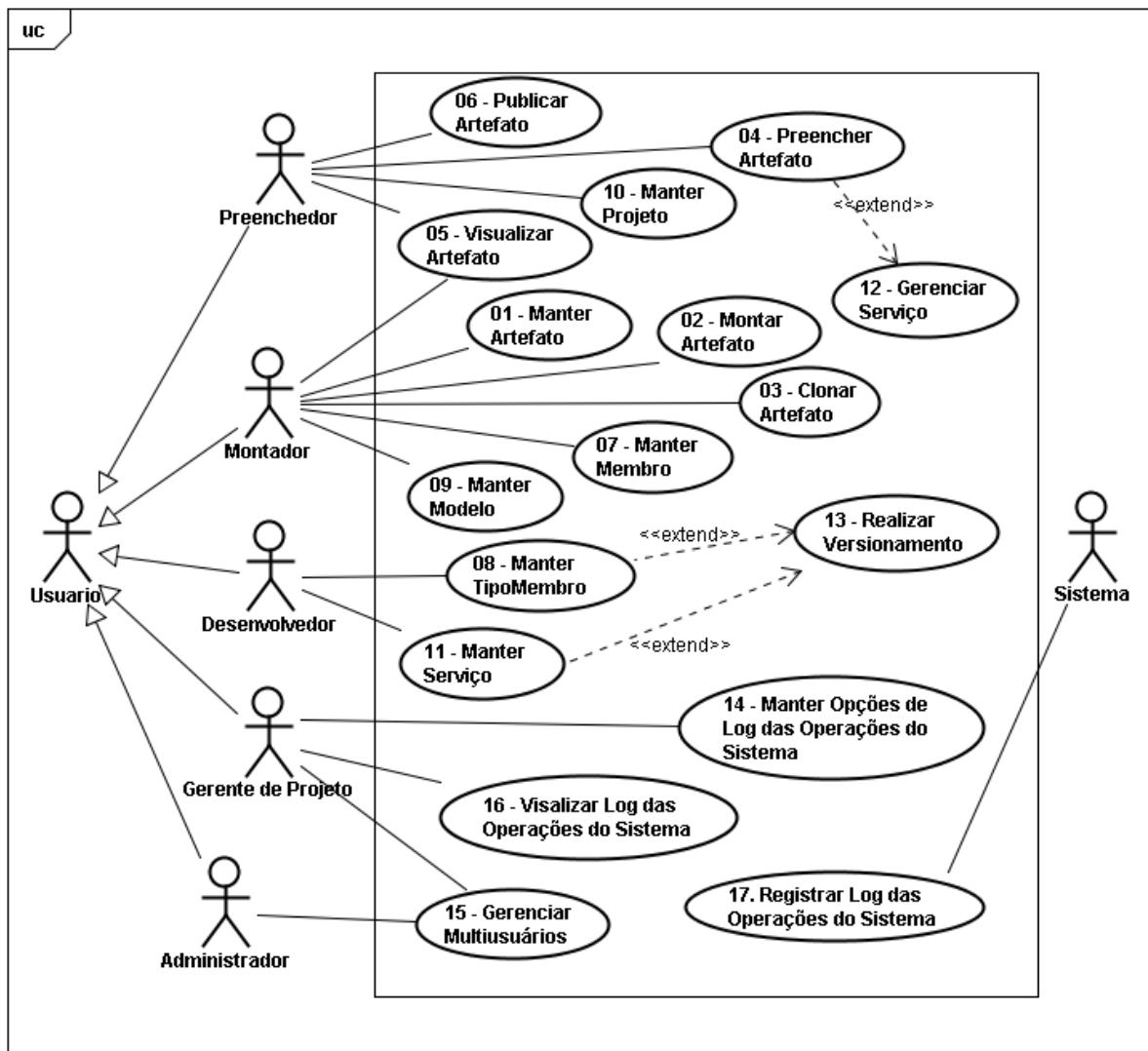


Figura 21 – Diagrama geral de casos de uso

6.4 Detalhamento de Casos de Uso

6.4.1 Caso de Uso 01 - Manter Artefato

6.4.1.1 Cenário 01 - Mapear Artefato

Descrição do Cenário	Cenário responsável por mapear um novo artefato no framework.
Requisitos Associados	R03 - Manter Artefato
Pré Condições	O usuário deverá estar logado no sistema com permissão de Montador.
Pós Condições	O Artefato mapeado no framework.
Atores	Montador
Fluxo Principal	
Ações Recebidas	Ações Realizadas

1. Solicita mapeamento do artefato.	2. Verifica se os atributos nome e descrição do artefato estão preenchidos. 3. Verifica duplicidade de artefato. 4. Executa CDU07 Manter Membro (Mapear membro) para os membros não cadastrados. 5. Realiza o cadastro do artefato. 6. Realiza o mapeamento Artefato-Membro. 7. Realiza o mapeamento Artefato-Servico. 8. Armazena sucesso de execução
9. Retorno da execução.	
Fluxo Alternativo 1 - Atributos nome/descrição do artefato não preenchidos.	
Ações Recebidas	Ações Realizadas
2.1. Atributos nome/descrição do artefato não foram preenchidos.	2.2. Armazena o nome dos atributos não preenchidos. 2.3. Armazena erro de atributos não preenchidos.
2.4. Retorno fluxo principal 9.	
Fluxo Alternativo 2 - Artefato duplicado no framework.	
Ações Recebidas	Ações Realizadas
3.1. O artefato informado já se encontra cadastrado no framework.	3.2. Armazena erro de artefato duplicado no framework.
3.3. Retorno fluxo principal 9.	
Fluxo Alternativo 3 -Erro de execução CDU07 Manter Membro (Mapear Membro).	
Ações Recebidas	Ações Realizadas
4.2. Retorno fluxo principal 9.	4.1. Erro na execução do CDU07 Manter Membro (Mapear Membro).

Tabela 70 – CDU01 – Detalhamento Cenário 01

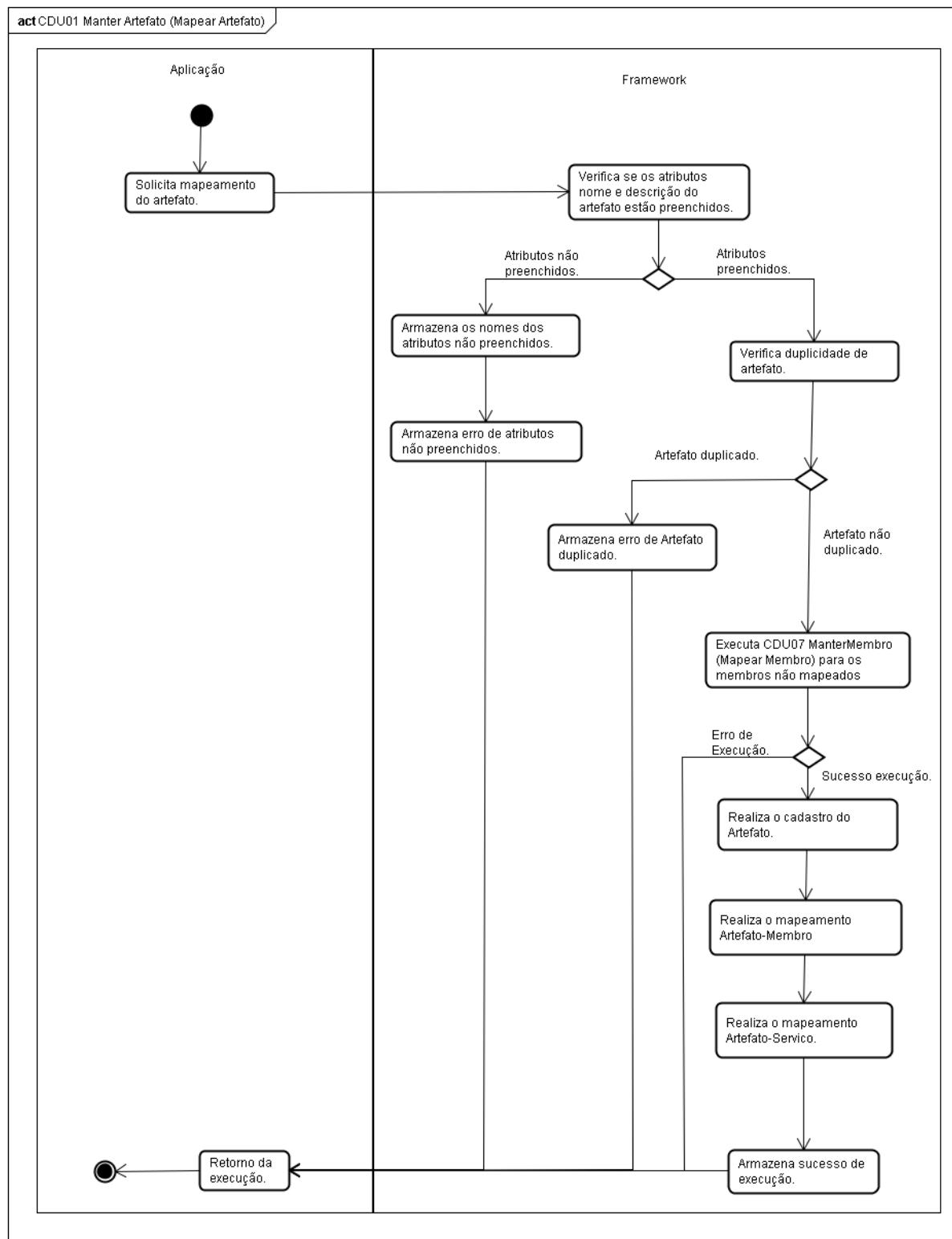


Figura 22 – Diagrama de Atividades Cenário 01 Mapear Artefato

6.4.1.2 Cenário 02 - Pesquisar Artefato

Descrição do Cenário	Cenário responsável por realizar a pesquisa de artefatos cadastrados no framework.
Requisitos Associados	R03 - Manter Artefato
Pré Condições	O usuário deverá estar logado no sistema com permissão de Montador.
Pós Condições	Lista de artefatos.
Atores	Montador
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1. Solicita pesquisa de artefatos.	2. Realiza pesquisa de artefatos filtrando a partir da categoria, nome e descrição informados. 3. Armazena lista de artefatos encontrados. 4. Armazena sucesso de execução. 5. Retorna da execução.
Fluxo Alternativo 1 - Nenhum artefato encontrado.	
Ações Recebidas	Ações Realizadas
2.1. Nenhum artefato encontrado.	2.2. Armazena erro de nenhum artefato cadastrado para o filtro informado. 2.3. Armazena lista vazia. 2.4. Retorno fluxo principal 5.

Tabela 71 – CDU01 – Detalhamento Cenário 02

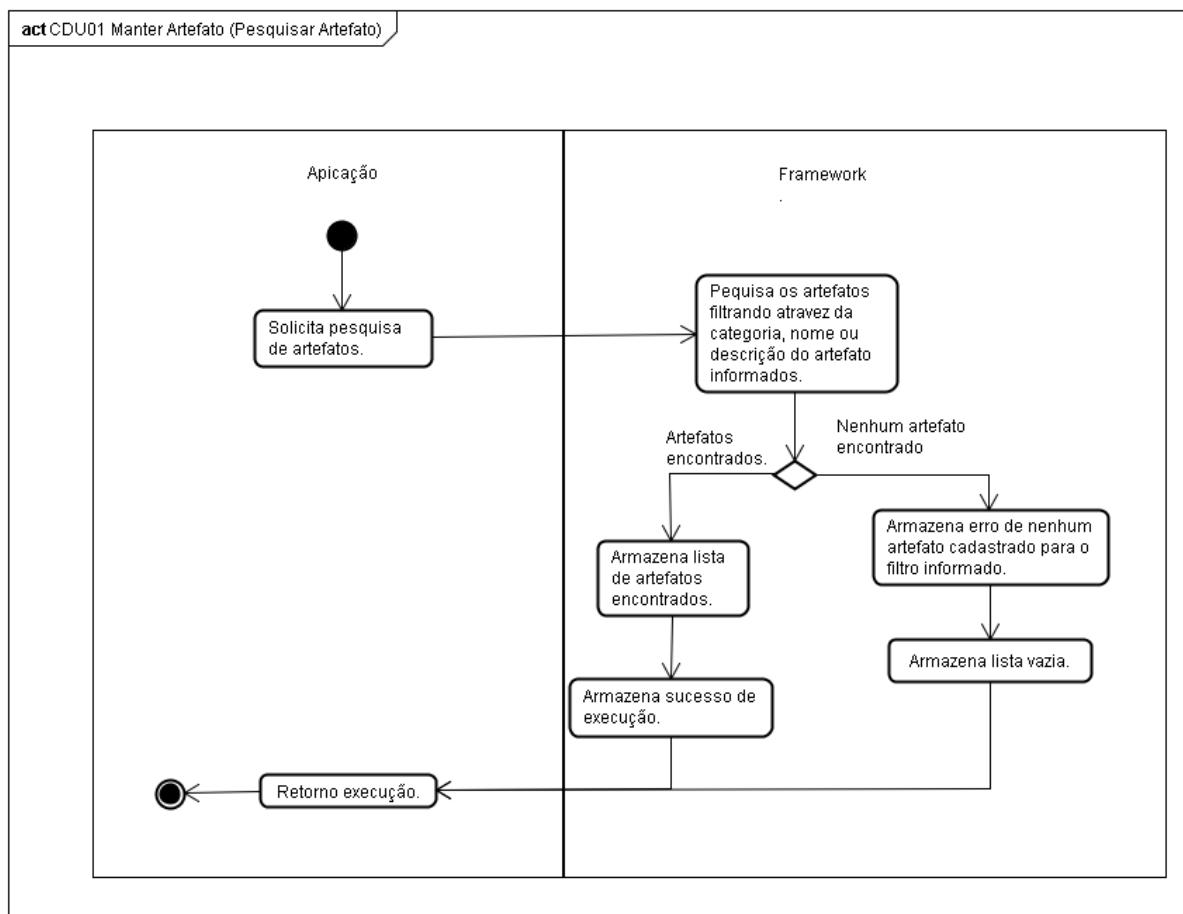


Figura 23 – Diagrama de Atividades Cenário 02 Pesquisar Artefato.

6.4.1.3 Cenário 03 - Remover Artefato

Descrição do Cenário	Cenário responsável por remover um artefato do framework.
Requisitos Associados	R03 - Manter Artefato
Pré Condições	O usuário deverá estar logado no sistema com permissão de Montador.
Pós Condições	O Artefato removido do framework.
Atores	Montador
Fluxo Principal	
Ações Recebidas	
1. Solicita remoção do artefato.	2. Realiza a rastreabilidade do artefato no framework. 3. Remove cadastro do artefato do framework. 4. Armazena sucesso de execução.
5. Retorno da execução.	
Fluxo Alternativo 1 - Artefato utilizado em um modelo e em nenhum projeto	
Ações Recebidas	
2.1. Artefato utilizado em um modelo e em nenhum projeto.	2.2. Armazena o nome do modelo em que o

	artefato esta sendo utilizado. 2.3. Armazenamento do erro de falha de remoção.
2.4. Retorno fluxo principal 5.	
Fluxo Alternativo 2 - Artefato utilizado em dois ou mais modelos e em nenhum projeto	
Ações Recebidas	Ações Realizadas
2.1. Utilizado em dois ou mais modelos e em nenhum projeto.	2.2. Armazena o nome dos modelos em que o artefato está sendo utilizado. 2.3. Armazenamento do erro de falha de remoção.
2.4. Retorno fluxo principal 5.	
Fluxo Alternativo 3 - Artefato utilizado em um ou mais projetos	
Ações Recebidas	Ações Realizadas
2.1. Artefato utilizado em um ou mais projetos.	2.2. Armazena o nome dos projetos em que o artefato está sendo utilizado. 2.3. Armazenamento do erro de falha de remoção.
2.4. Retorno fluxo principal 5.	

Tabela 72 – CDU01 – Detalhamento Cenário 03

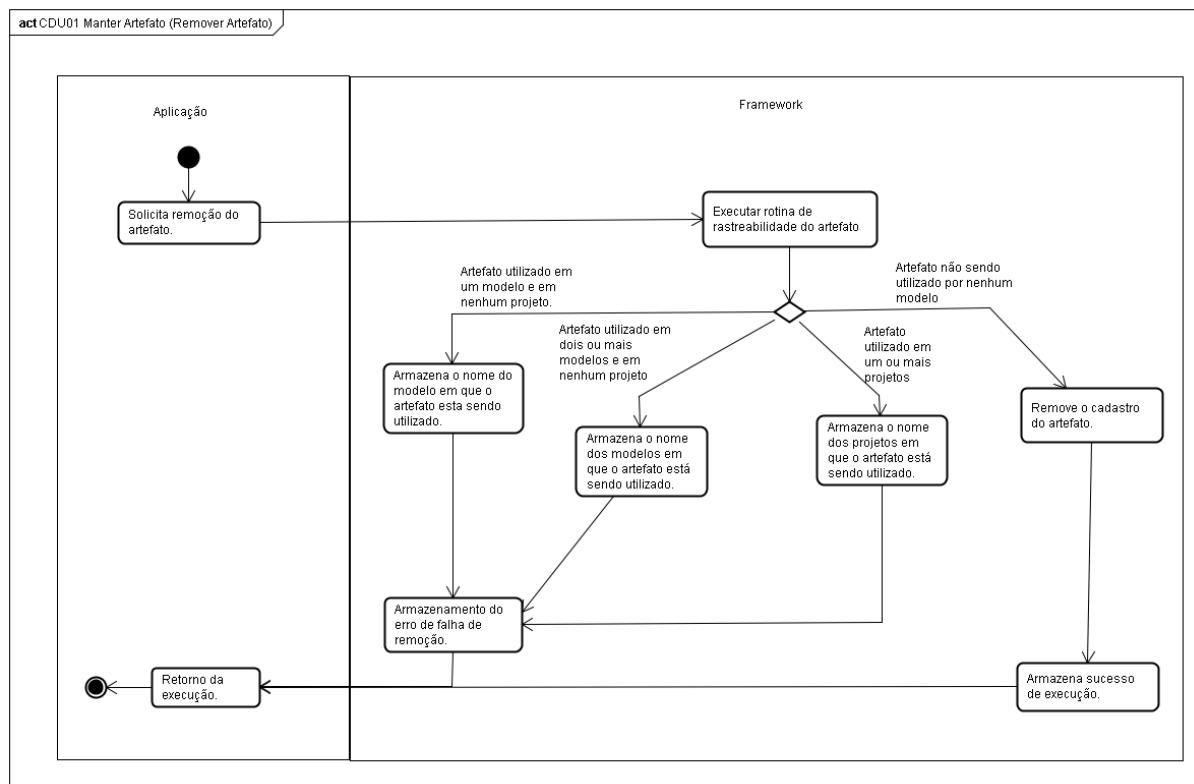


Figura 24 – Diagrama de Atividades Cenário 03 Remover Artefato.

6.4.1.4 Cenário 04 - Alterar Artefato (Rotina Inicial)

Descrição do Cenário	Cenário responsável por realizar a alteração de um artefato.
Requisitos Associados	R03 - Manter Artefato
Pré Condições	O usuário deverá estar logado no sistema com permissão de Montador.
Pós Condições	Artefato alterado no framework.
Atores	Montador
Fluxo Principal	
Ações Recebidas Ações Realizadas	
1. Solicita alteração de artefato. 4. Retorno da execução.	2. Verifica se o nome e a descrição do artefato estão preenchidos. 3. Executar rotina de rastreabilidade do artefato.
Fluxo Alternativo 1 - Atributos nome/descrição do artefato não preenchidos	
Ações Recebidas Ações Realizadas	
2.1. Atributos nome/descrição do artefato não foram preenchidos. 2.4. Retorno fluxo principal .	2.2. Armazena o nome dos atributos não preenchidos. 2.3. Armazena erro de atributos não preenchidos.
Fluxo Alternativo 1 - Artefato utilizado em um modelo e em nenhum projeto	
Ações Recebidas Ações Realizadas	
3.1. Artefato utilizado em um modelo e em nenhum projeto. 3.3. Retorno fluxo principal 4.	3.2. Executa cenário Alterar Artefato - Utilizado em um modelo e em nenhum projeto.
Fluxo Alternativo 2 - Artefato utilizado em dois ou mais modelos e em nenhum projeto	
Ações Recebidas Ações Realizadas	
3.1. Artefato utilizado em dois ou mais modelos e em nenhum projeto 3.3. Retorno fluxo principal 4.	3.2. Executa cenário Alterar Artefato - Utilizado em dois ou mais modelos e em nenhum projeto.
Fluxo Alternativo 3 - Artefato utilizado em um ou mais projetos	
Ações Recebidas Ações Realizadas	
3.1. Artefato utilizado em um ou mais projetos 3.3. Retorno fluxo principal 4.	3.2. Executa cenário Alterar Artefato - Utilizado em um ou mais projetos.
Fluxo Alternativo 4 - Artefato não sendo utilizado por nenhum modelo	
Ações Recebidas Ações Realizadas	
3.1. Artefato não sendo utilizado por nenhum modelo. 3.3. Retorno fluxo principal 4.	3.2. Executa cenário Alterar Artefato - Sem Utilização em modelo.

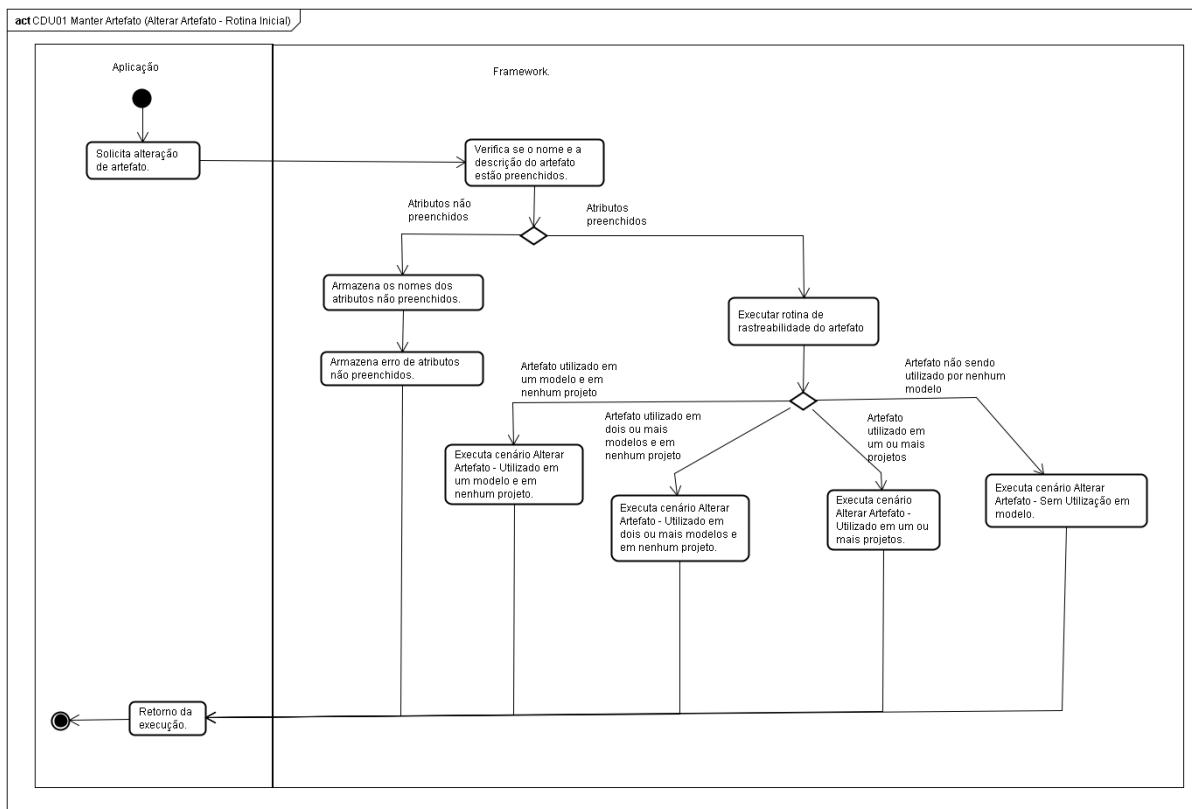
Tabela 73 – CDU01 – Detalhamento Cenário 04

Figura 25 – Diagrama de Atividades Cenário 04 Alterar Artefato
(Rotina Inicial)

6.4.1.5 Cenário 05 - Alterar Artefato (Sem utilização em modelo)

Descrição do Cenário	Cenário responsável alterar um artefato que não esteja sendo utilizado por nenhum modelo cadastrado no framework.
Requisitos Associados	R03 - Manter Artefato
Pré Condições	O usuário deverá estar logado no sistema com permissão de Montador.
Pós Condições	Artefato alterado no framework.
Atores	Montador
Fluxo Principal	
Ações Recebidas	
1. Recebe instância do artefato. 4. Retorno da execução.	
Ações Realizadas	
2. Realiza alteração do artefato no framework. 3. Armazena sucesso de execução.	

Tabela 74 – CDU01 – Detalhamento Cenário 05

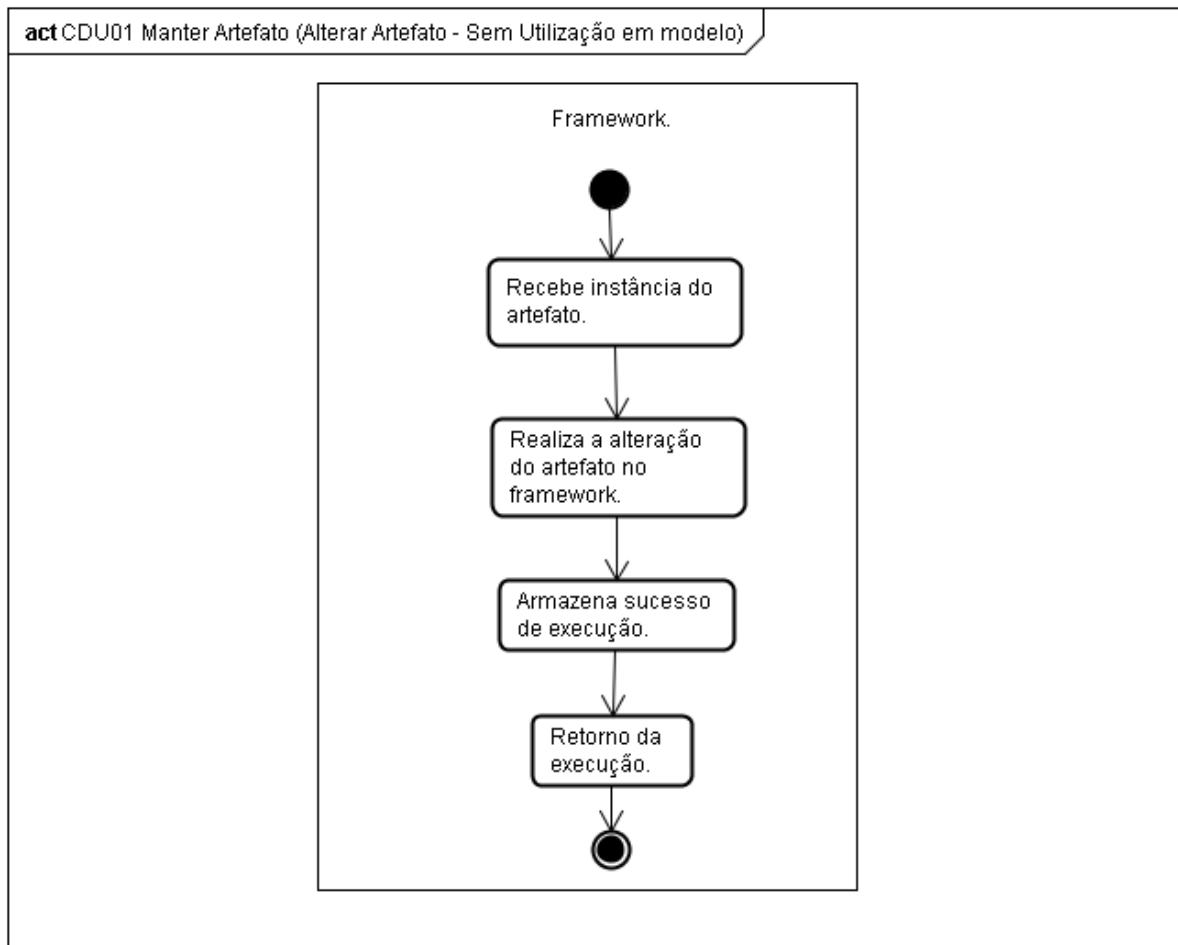


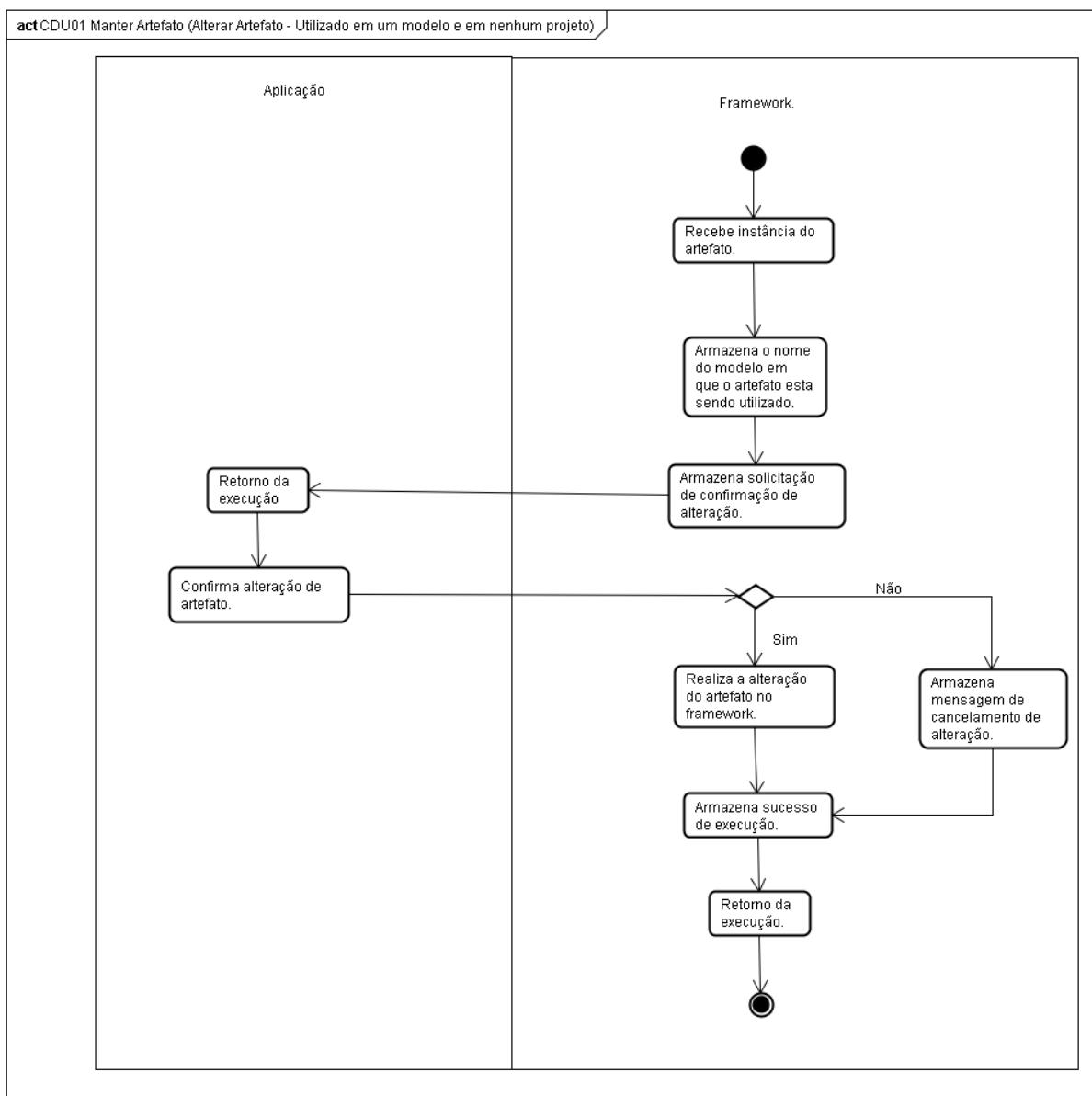
Figura 26 – Diagrama de Atividades Cenário 05 Alterar Artefato
(Sem utilização em modelo)

6.4.1.6 Cenário 06 - Alterar Artefato (Utilizado em um modelo e em nenhum projeto)

Descrição do Cenário	Cenário responsável alterar um artefato que esteja sendo utilizado em um modelo mas em nenhum projeto cadastrado.
Requisitos Associados	R03 - Manter Artefato
Pré Condições	O usuário deverá estar logado no sistema com permissão de Montador.
Pós Condições	Artefato alterado no framework.
Atores	Montador
Fluxo Principal	
Ações Recebidas	
1. Recebe instância do artefato.	
4. Retorno da execução.	
5. Confirma alteração de artefato.	
Ações Realizadas	
2. Armazenamento do nome do modelo em que o artefato esta sendo utilizado.	
3. Armazena solicitação de confirmação de alteração.	
6. Realiza alteração do artefato no framework.	
7. Armazena sucesso de execução.	

8. Retorno da execução.	
Fluxo Alternativo 1 - Não confirmação da alteração do artefato.	
Ações Recebidas	Ações Realizadas
5.1. Não confirmação da alteração do artefato. 5.3. Retorno fluxo principal 7.	5.2. Armazenamento da mensagem de cancelamento de alteração.

Tabela 75 – CDU01 – Detalhamento Cenário 06

Figura 27 – Diagrama de Atividades Cenário 06 Alterar Artefato
(Utilizado em um modelo e em nenhum projeto)

6.4.1.7 Cenário 07 - Alterar Artefato (Utilizado em dois ou mais modelos e em nenhum projeto)

Descrição do Cenário	Cenário responsável alterar um artefato que esteja sendo utilizado em dois ou mais modelo mas em nenhum projeto cadastrado
Requisitos Associados	R03 - Manter Artefato
Pré Condições	O usuário deverá estar logado no sistema com permissão de Montador.
Pós Condições	Artefato alterado no framework.
Atores	Montador
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1. Recebe instância do artefato. 3. Retorno da execução. 4. Seleciona o modelo. 9. Retorno da execução.	2. Armazenamento os modelos em que o artefato esta sendo utilizado. Armazena solicitação de escolha de modelo onde o artefato será clonado. 5. Executa CDU03 clonar artefato. 6. Realiza a alteração do artefato clonado no framework. 7. Altera artefato no modelo selecionado. 8. Armazena sucesso de execução.
Fluxo Alternativo 1 - Nenhum modelo selecionado.	
Ações Recebidas	Ações Realizadas
5.1. Nenhum modelo selecionado para alteração do artefato.. 5.3. Retorno fluxo principal 9.	5.2. Armazenamento da mensagem de cancelamento de alteração.
Fluxo Alternativo 2 - Falha da execução do CDU03 clonar artefato.	
Ações Recebidas	Ações Realizadas
6.1. Falha na execução do CDU03 clonar artefato.	6.2. Retorno fluxo principal 10.

Tabela 76 – CDU01 – Detalhamento Cenário 07

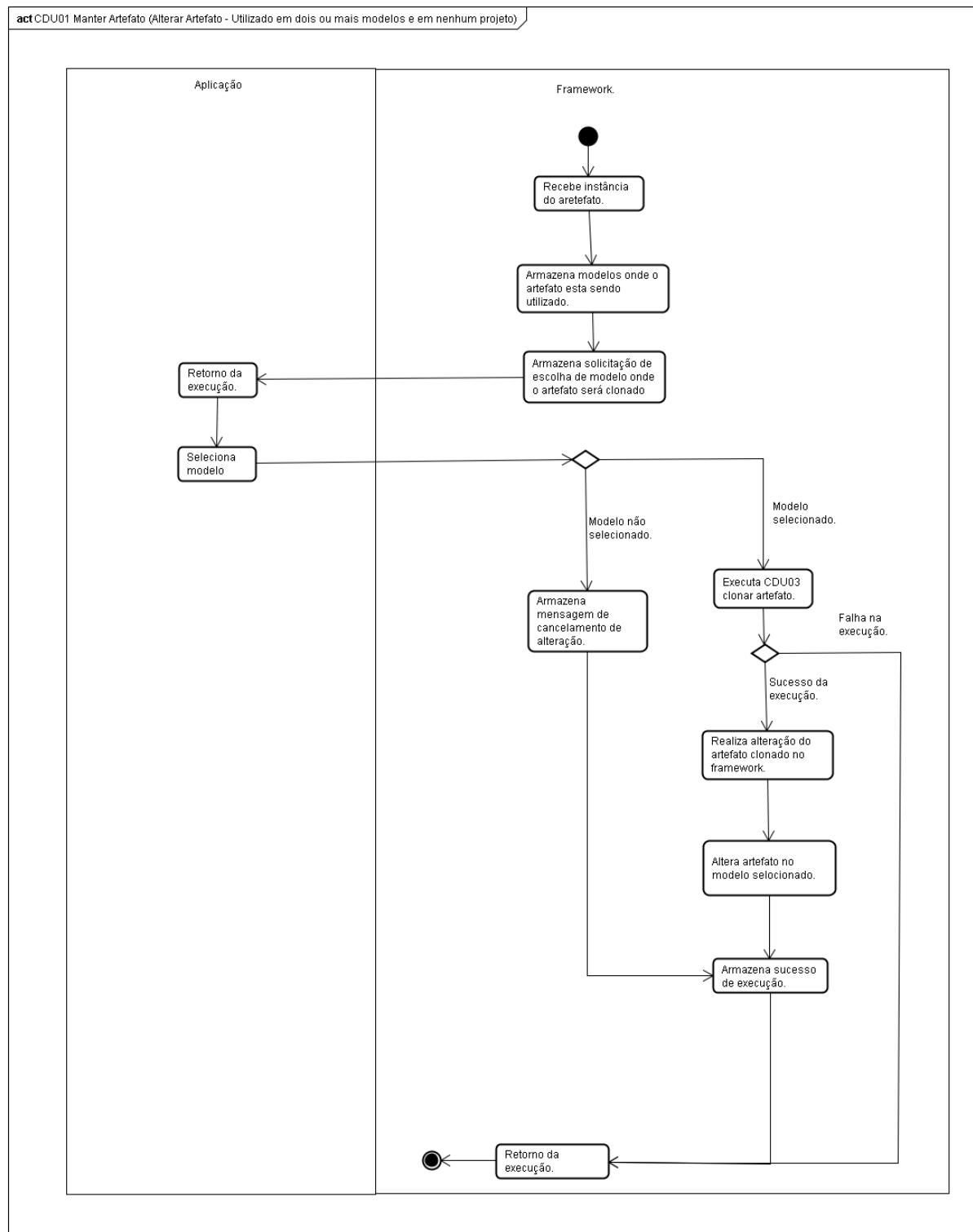


Figura 28 – Diagrama de Atividades Cenário 07 Alterar Artefato
 (Utilizado em dois ou mais modelos e em nenhum projeto)

6.4.1.8 Cenário 08 - Alterar Artefato (Utilizado em um ou mais projetos)

Descrição do Cenário	Cenário responsável alterar um artefato que esteja sendo utilizado em um ou mais documentos cadastrados no framework.
----------------------	---

Requisitos Associados	R03 - Manter Artefato
Pré Condições	O usuário deverá estar logado no sistema com permissão de Montador.
Pós Condições	Artefato alterado no framework.
Atores	Montador
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1. Recebe instância do artefato. 6. Retorno da execução. 7. Confirma alteração de artefato. 10. Retorno da execução.	2. Armazenamento os projetos em que o artefato esta sendo utilizado. 3. Verifica se somente os parâmetros dos membros foram alterados. 4. Armazena os nomes dos projetos onde o artefato está sendo utilizado. 5. Armazena solicitação de alteração de artefato informando os projetos onde ele está sendo utilizado. 8. Realiza alteração do artefato no framework. 9. Armazena sucesso de execução.
Fluxo Alternativo 1 - Algum Membro/Serviço foi adicionado ou removido do artefato.	
Ações Recebidas	Ações Realizadas
3.1. Algum Membro / Serviço foi adicionado ou removido do artefato. 3.4. Retorno execução. 3.5. Informa novo nome e nova descrição e confirma clonagem.	3.2. Armazenamento do nome dos projetos onde o artefato está sendo utilizado. 3.3. Solicita confirmação de clonagem de artefato, informando os projetos onde ele esta sendo utilizado, solicitando o novo nome e a nova descrição do artefato. 3.6. Executa CDU03 Clonar Artefato (Clonar). 3.7. Retorno fluxo principal 10.
Subfluxo Alternativo 2 -Não confirmação de clonagem de artefato.	
Ações Recebidas	Ações Realizadas
3.4.1. Cancela clonagem de artefato. 3.4.3. Retorno fluxo principal 10.	3.4.2. Armazena cancelamento de alteração de artefato.
Fluxo Alternativo 3 -Não confirmação de alteração de artefato.	
Ações Recebidas	Ações Realizadas
6.1. Não confirmação da alteração de artefato. 6.3. Retorno fluxo principal 9.	6.2. Armazena mensagem de cancelamento de alteração de artefato.

Tabela 77 – CDU01 – Detalhamento Cenário 08

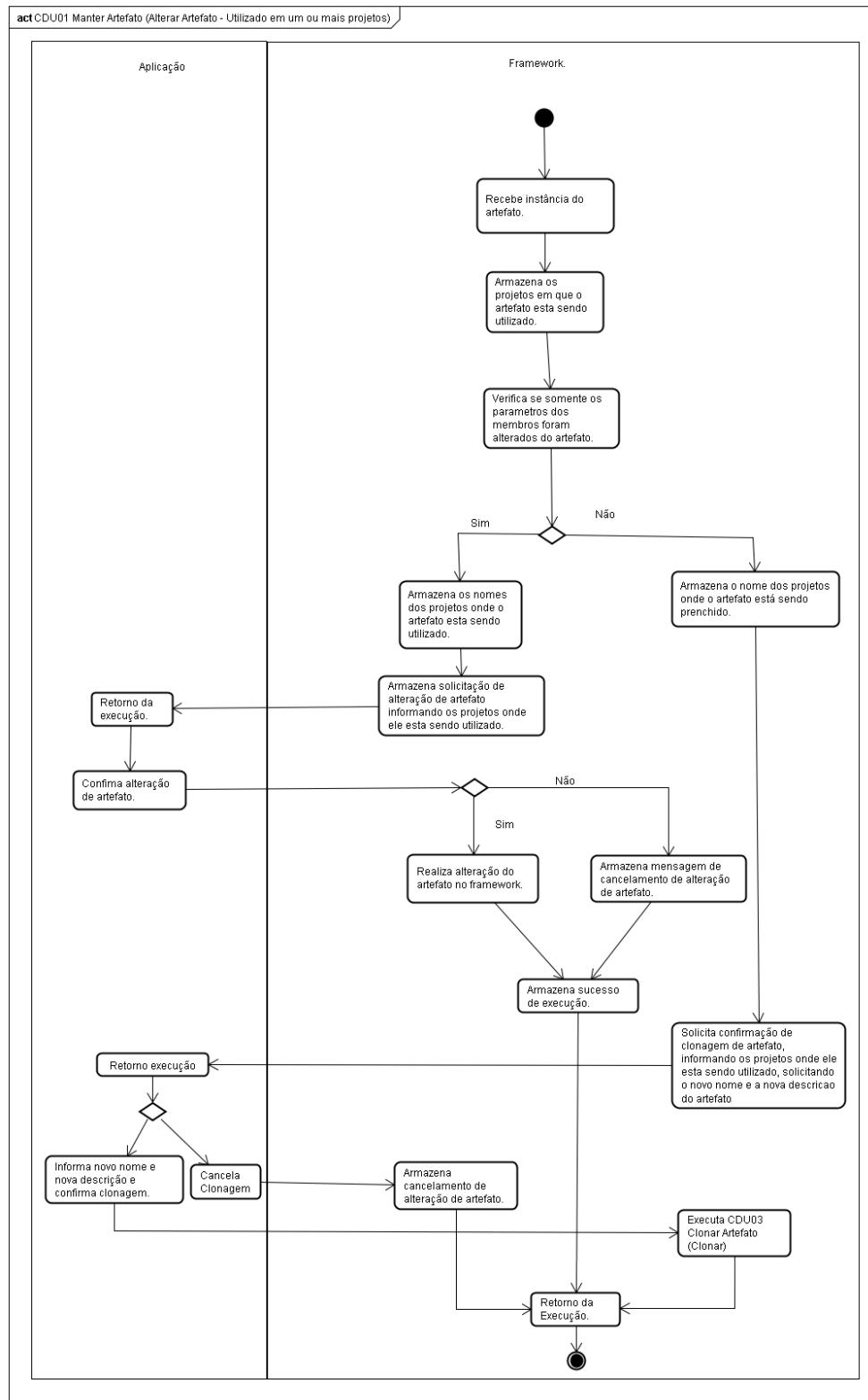


Figura 29 – Diagrama de Atividades Cenário 08 Alterar Artefato
(Utilizado em um ou mais projetos)

6.4.2 Caso de Uso 02 - Montar Artefato

6.4.2.1 Cenário 01 - Criar Artefato

Descrição do Cenário	Cenário responsável por realizar a montagem de um artefato no framework.
Requisitos Associados	R04 - Montar Artefato. R03 Manter Artefato
Pré Condições	O usuário deverá estar logado no sistema com permissão de usuário montador.
Pós Condições	O artefato persistido no framework.
Atores	Montador
Fluxo Principal	
Ações Recebidas	
10. Solicita criação de novo artefato.	
12. Preenche o nome e a descrição do artefato.	11. Instancia um novo artefato.
4. Seleciona a ação de adicionar / remover membros e serviços no artefato ou visualiza artefato ou cancelamento do artefato.	
5. Solicita mapeamento de artefato.	6. Executa CDU01 Manter Artefato (Mapear Artefato)
7. Retorna resultado da execução.	
Fluxo Alternativo 1 – Adiciona novo membro ao artefato.	
Ações Recebidas	
2.5. Adiciona um novo membro no artefato.	2.6. Executa cenário adicionar membro.
2.7. Retorno fluxo principal 4.	
Fluxo Exceção 2 – Remove membro do artefato.	
Ações Recebidas	
4.1. Remove membro do artefato .	4.2. Executa cenário remover membro.
4.3. Retorno fluxo principal 4.	
Fluxo Exceção 3 – Adiciona serviço ao artefato.	
Ações Recebidas	
4.1. Adiciona um novo serviço ao artefato.	4.2. Executa cenário adicionar serviço.
4.3. Retorno fluxo principal 4.	
Fluxo Exceção 4 – Remove serviço do artefato.	
Ações Recebidas	
4.1. Remove serviço do artefato	4.2. Executa cenário remover serviço.
4.3. Retorno fluxo principal 4.	
Fluxo Exceção 5 – Solicita visualização do artefato.	
Ações Recebidas	
4.1. Solicita visualização do artefato.	4.2. Executa cenário visualizar artefato.
4.3. Retorno fluxo principal 4.	
Fluxo Exceção 6 – Solicita cancelamento de montagem de artefato.	
Ações Recebidas	
4.1. Cancelamento da montagem do artefato.	4.2. Cancela montagem do artefato.

4.4. Retorno fluxo principal 7.

4.3. Armazena sucesso de execução.

Tabela 78 – CDU02 – Detalhamento Cenário 01

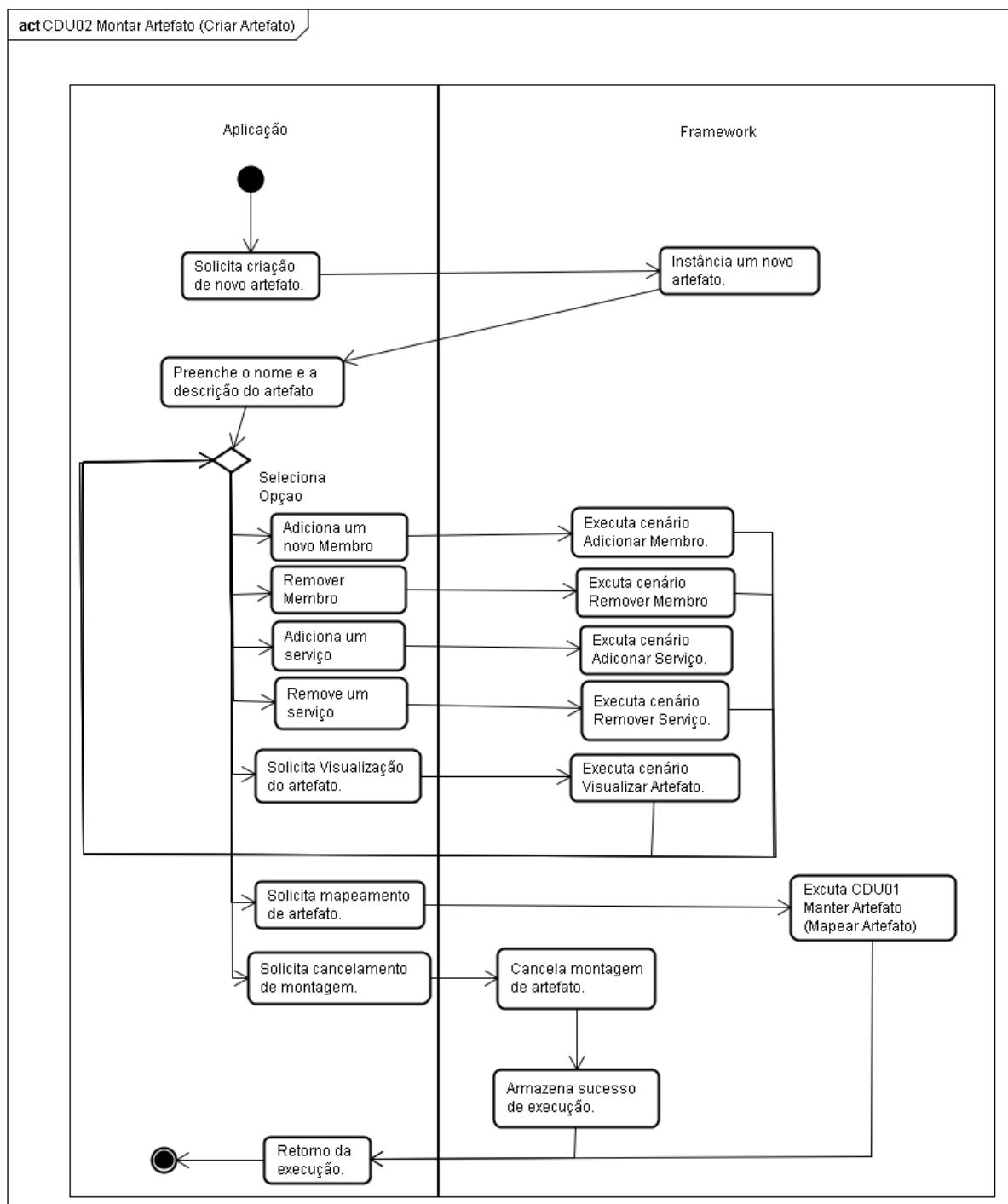


Figura 30 – Diagrama de Atividades Cenário 01 – Criar Artefato.

6.4.2.2 Cenário 02 - Adicionar Membro.

Descrição do Cenário	Cenário responsável por acrescentar um membro ao artefato.
Requisitos Associados	R04 - Montar Artefato.
Pré Condições	O usuário deverá estar logado no sistema com permissão de usuário montador.
Pós Condições	Membro adicionado ao artefato.
Atores	Montador
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1. Seleciona membro a ser inserido. 3. Posiciona o membro no artefato e altera os parâmetros do membro caso necessário. 4. Adiciona membro no artefato. 8. Retorno da execução.	2. Retorna lista de parâmetros do TipoMembro correspondente ao membro selecionado. 5. Verifica os parâmetros do membro. 6. Armazena o membro no artefato. 7. Armazena sucesso de execução.
Fluxo Alternativo 1 - Cancelamento de adição do membro.	
Ações Recebidas	Ações Realizadas
2.1. Cancelamento da adição do membro no artefato. 2.3. Retorno fluxo principal 7.	2.2. Armazena cancelamento de adição do membro.
Fluxo Alternativo 2 - Parâmetros incorretos do membro.	
Ações Recebidas	Ações Realizadas
5.1. Parâmetros do membro estão incorretos. 5.3. Retorno fluxo principal 3.	5.2. Armazena erro de parâmetros incorretos, com os nomes dos parâmetros inválidos.

Tabela 79 – CDU02 – Detalhamento Cenário 02

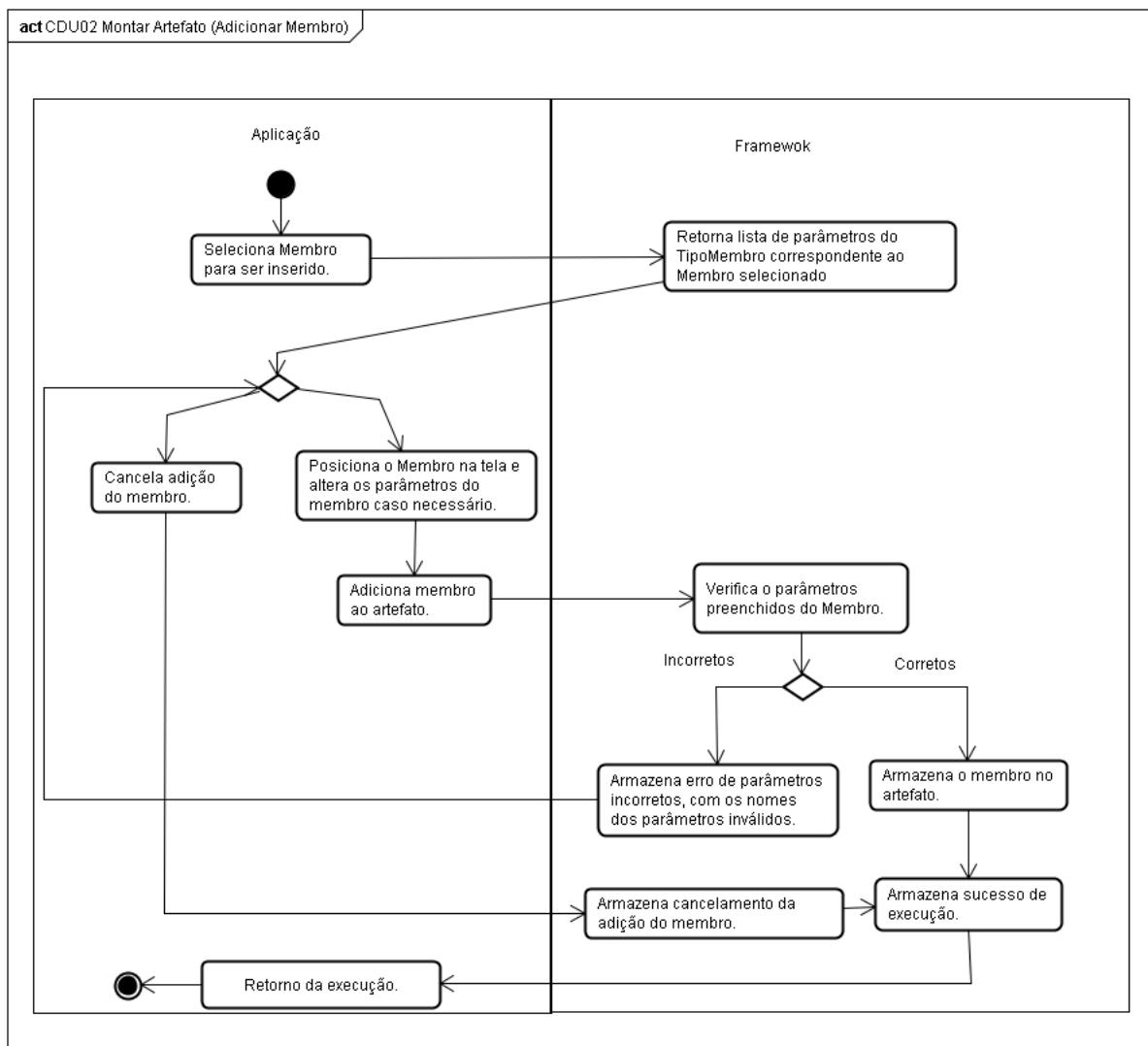


Figura 31 – Diagrama de Atividades Cenário 02 – Adicionar membro.

6.4.2.3 Cenário 03 - Remover Membro

Descrição do Cenário	Cenário responsável por remover um membro do artefato.
Requisitos Associados	R04 - Montar Artefato
Pré Condições	O usuário deverá estar logado no sistema com permissão de usuário montador.
Pós Condições	O membro estará removido do artefato.
Atores	Montador
Fluxo Principal	
Ações Recebidas	
1. Solicita remoção do membro do artefato.	2. Remove membro da lista de membros do artefato.
4. Retorna da execução.	3. Armazena sucesso de execução.

Tabela 80 – CDU02 – Detalhamento Cenário 03

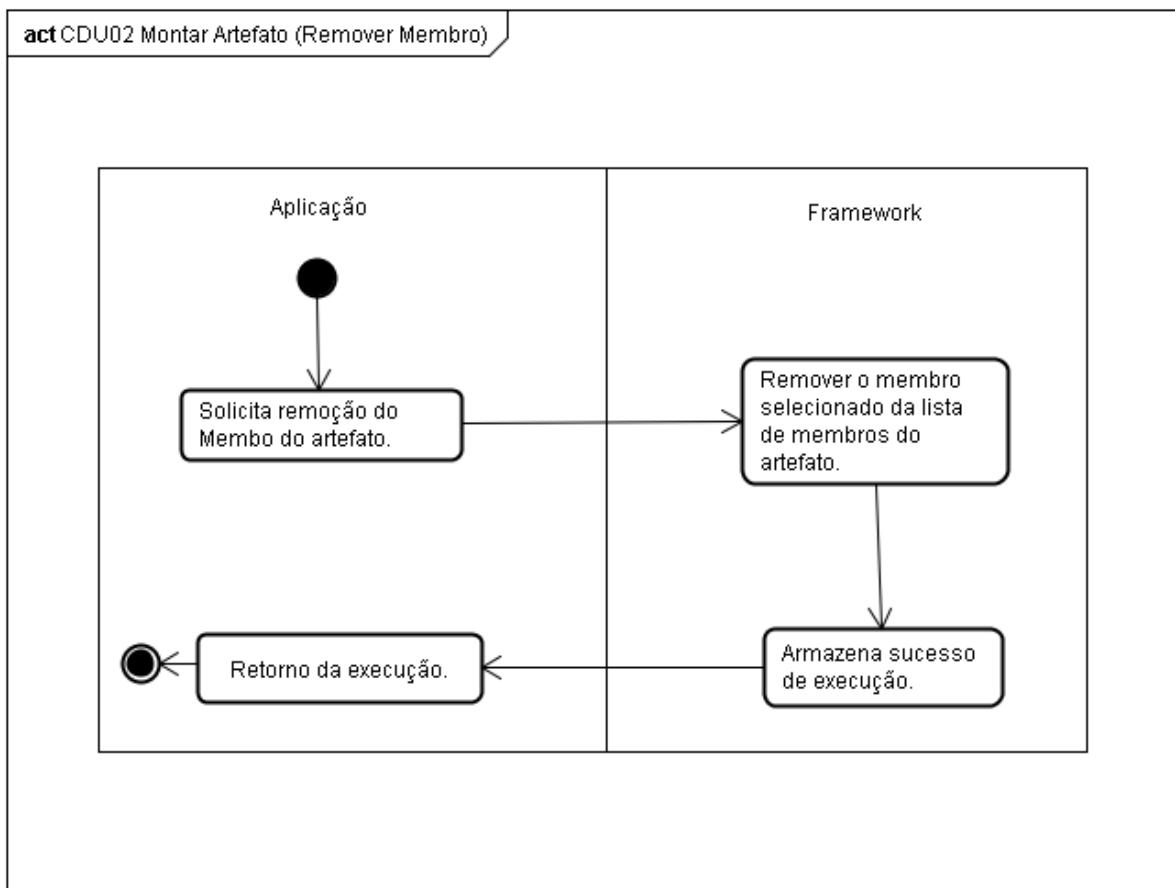


Figura 32 – Diagrama de Atividades Cenário 03 – Remover membro

6.4.2.4 Cenário 04 - Adicionar Serviço.

Descrição do Cenário	Cenário responsável por acrescentar um serviço ao artefato.
Requisitos Associados	R04 - Montar Artefato.
Pré Condições	O usuário deverá estar logado no sistema com permissão de usuário montador.
Pós Condições	Serviço adicionado ao artefato.
Atores	Montador
Fluxo Principal	
Ações Recebidas	
1. Seleciona serviço a ser inserido. 3. Preenche a lista de parâmetros do serviço. 4. Adiciona serviço no artefato. 8. Retorno da execução.	2. Retorna lista de parâmetros serviço. 5. Verifica os parâmetros do serviço 6. Adicionar serviço ao artefato. 7. Armazena sucesso de execução.
Fluxo Alternativo 1 - Serviço com TipoMembro associado.	
Ações Recebidas	

2.1. Serviço associado a um TipoMembro.	2.2. Instancia TipoMembro, acrescentando a lista de parâmetros do serviço, a lista de parâmetros do TipoMembro.
2.3. Retorno fluxo principal 3.	
Fluxo Alternativo 2 - Parâmetros do serviço incorretos.	
Ações Recebidas	Ações Realizadas
5.1. Parâmetros do serviço incorretos.	5.2. Armazena erro de parâmetros incorretos, com os nomes dos parâmetros inválidos.
5.3. Retorno fluxo principal 8.	

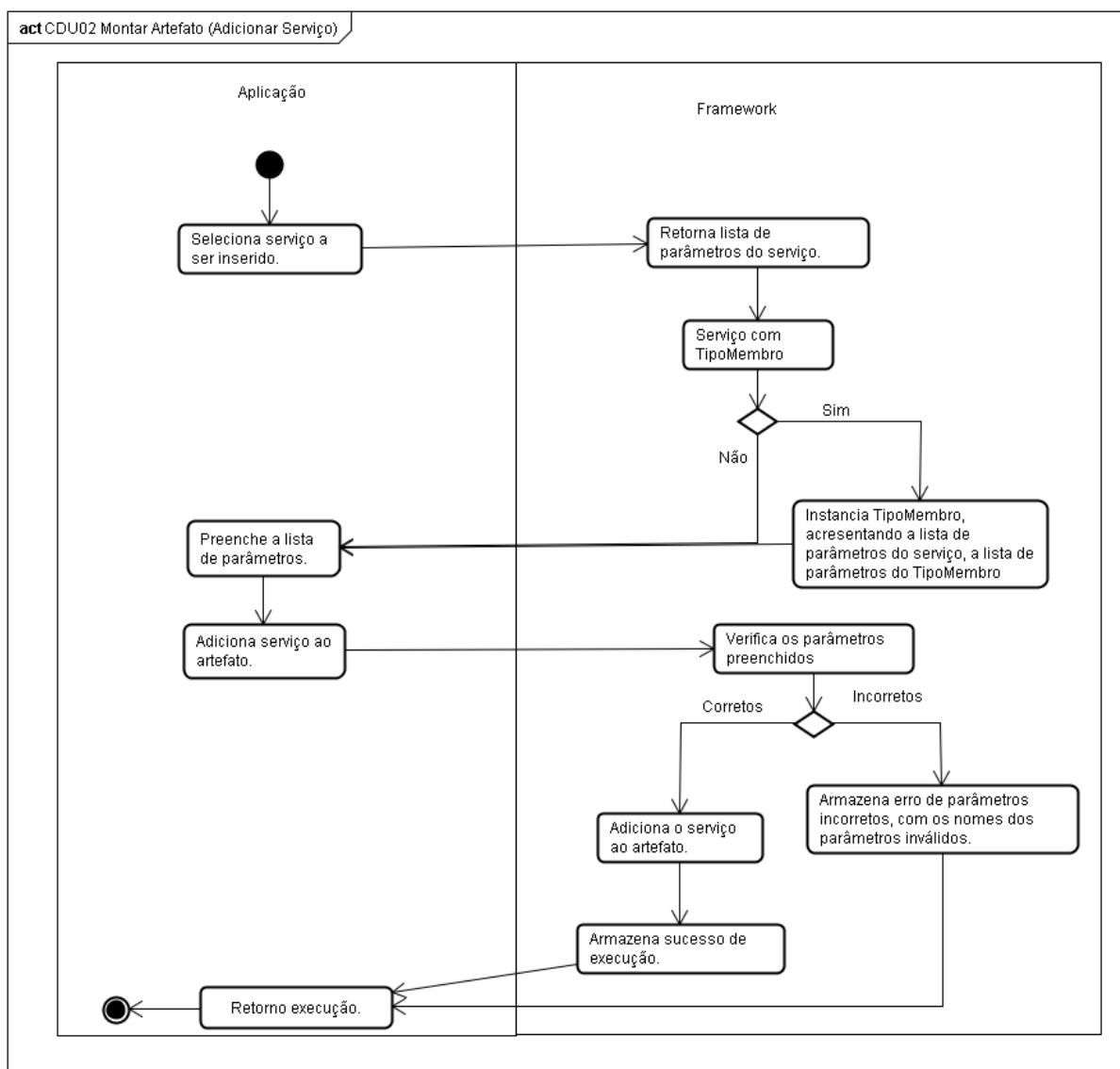
Tabela 81 – CDU02 – Detalhamento Cenário 04

Figura 33 – Diagrama de Atividades Cenário 04 – Adicionar serviço.

6.4.2.5 Cenário 05 - Remover Serviço

Descrição do Cenário	Cenário responsável por remover um serviço do artefato.
Requisitos Associados	R04 - Montar Artefato
Pré Condições	O usuário deverá estar logado no sistema com permissão de usuário montador.
Pós Condições	O serviço estar removido do artefato.
Atores	Montador
Fluxo Principal	
Ações Recebidas	
1. Solicita remoção de serviço do artefato.	2. Remove serviço do artefato. 3. Verificação se o serviço está servindo de parametrização de outro serviço. 4. Verificação se o serviço tem um TipoMembro. 5. Armazena sucesso de execução.
6. Retorna da execução.	
Fluxo Alternativo 1 - Serviço sendo utilizado como entrada de outro serviço	
Ações Recebidas	
2.1. Serviço sendo utilizado como entrada de parâmetros de outro serviço.	2.2. Armazena mensagem de serviço servindo de entrada de outro serviço. 2.3. Remove referência dos parâmetros dos serviços.
2.4. Retorno fluxo principal 4.	
Fluxo Alternativo 2 - Serviço com TipoMembro associado.	
Ações Recebidas	
3.1. Serviço com TipoMembro associado.	3.2. Remove TipoMembro do artefato.
3.3. Retorno fluxo principal 4.	

Tabela 82 – CDU02 – Detalhamento Cenário 05

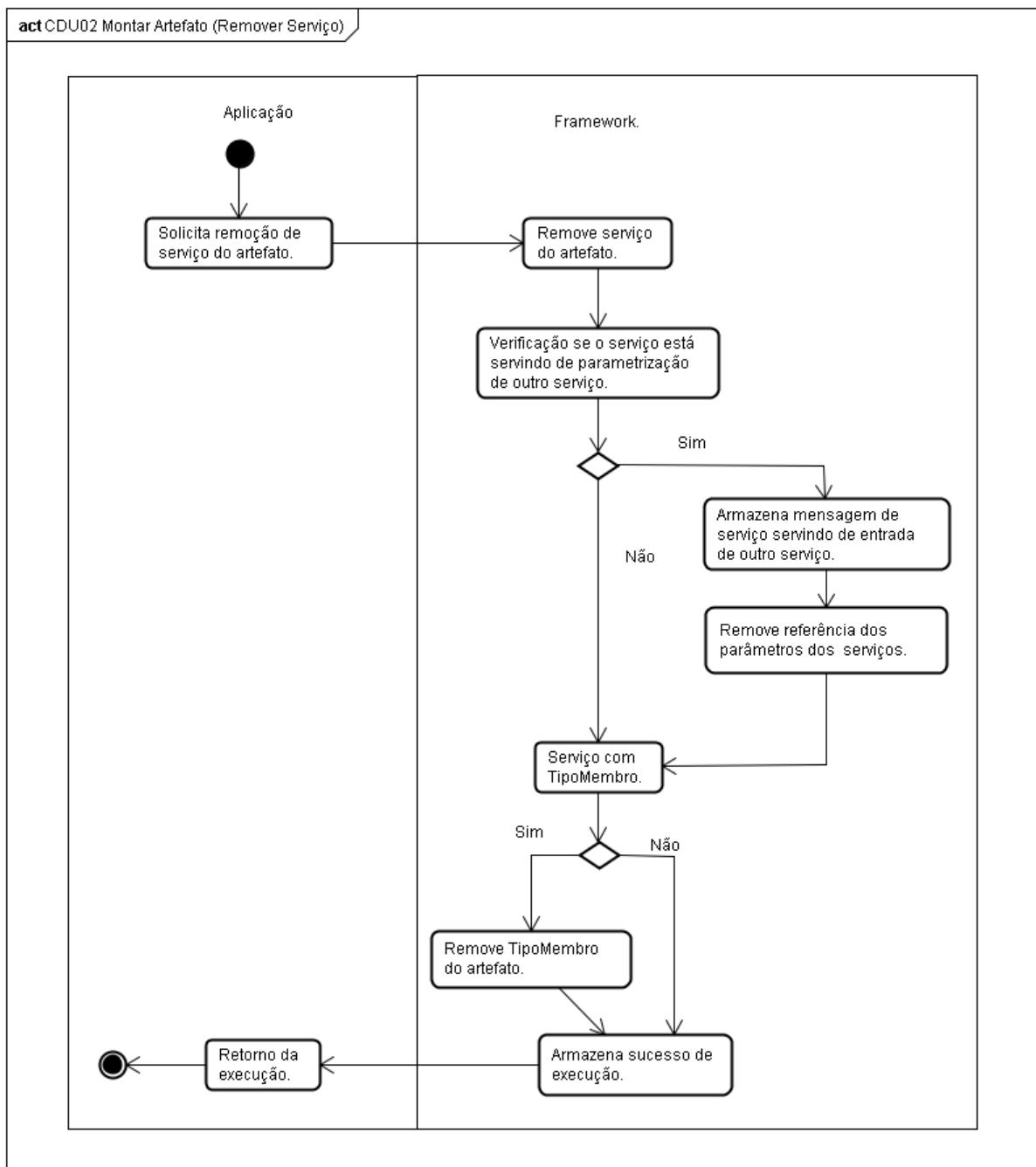


Figura 34 – Diagrama de Atividades Cenário 05 – Remover serviço.

6.4.2.6 Cenário 06 - Visualizar Artefato

Descrição do Cenário	Cenário responsável por fornecer a visão de visualização do artefato que está sendo montado.
Requisitos Associados	R04 - Montar Artefato, R06 Preencher artefato.
Pré Condições	O usuário deverá estar logado no sistema com permissão de usuário montador.
Pós Condições	Pré visualização do artefato.
Atores	Montador
Fluxo Principal	

Ações Recebidas	Ações Realizadas
<p>1. Solicita prévia de visualização de artefato.</p> <p>4. Exibe interface de visualização</p> <p>5. Cancela interface de visualização e retorna interface de montagem.</p> <p>6. Retorno execução.</p>	<p>2. Executa CDU05 (Visualizar Artefato) do artefato que esta sendo construído.</p> <p>3. Retorna visão de visualização do artefato.</p>

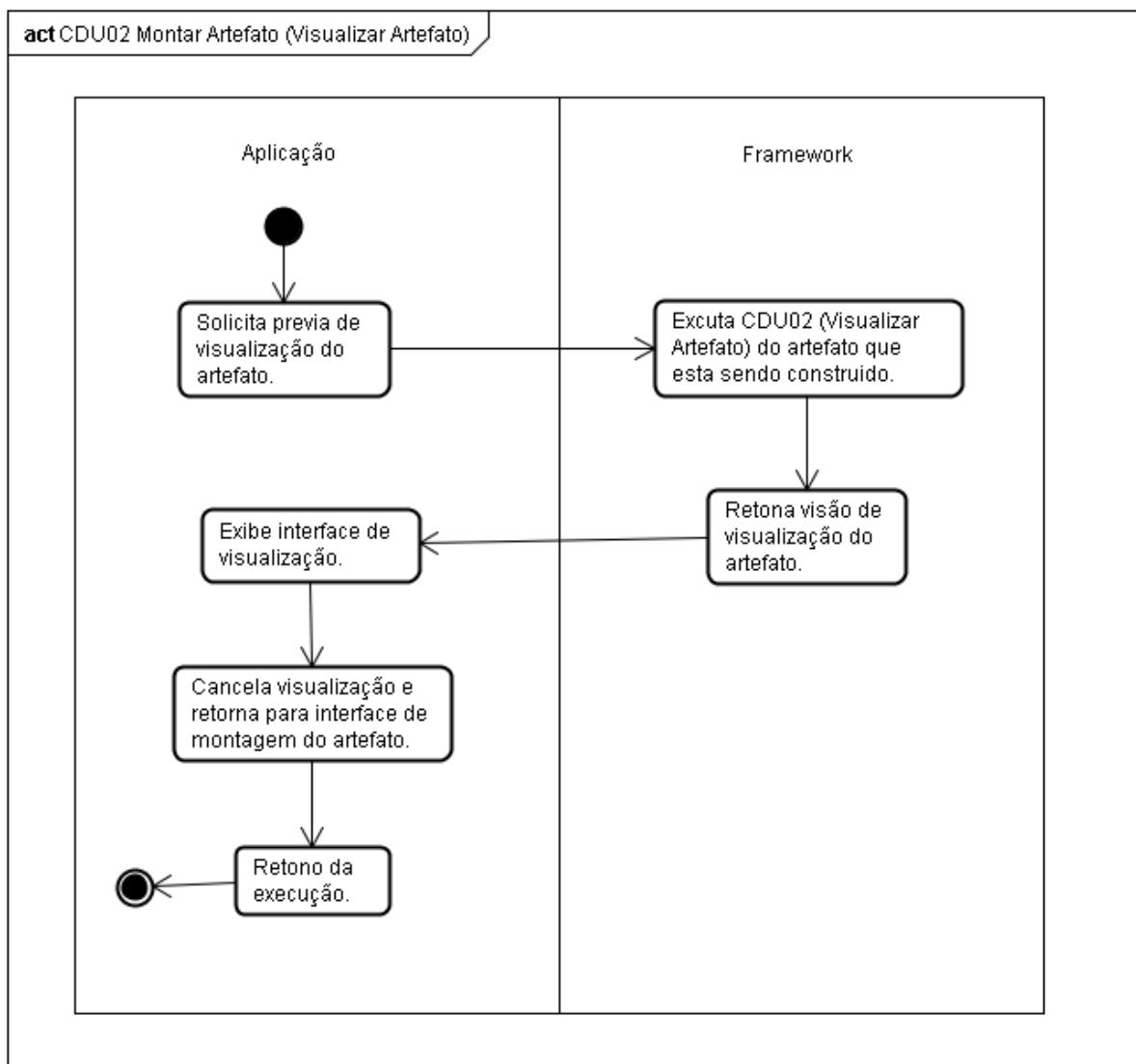
Tabela 83 – CDU02 – Detalhamento Cenário 06

Figura 35 – Diagrama de Atividades Cenário 06 – Visualizar Artefato.

6.4.3 Caso de Uso 03 – Clonar Artefato

6.4.3.1 Cenário 01 – Clonar Artefato

Descrição do Cenário	Cenário responsável por realizar a clonagem de um artefato no framework.
Requisitos Associados	R05 – Clonar Artefato, R03 - Manter Artefato.
Pré Condições	O usuário deverá estar logado no sistema com permissão de montador.
Pós Condições	O artefato clonado no framework.
Atores	Montador
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1. Solicita clonagem do artefato, informando o novo nome e a nova descrição do artefato. 6. Retorno da execução.	2. Recebe o artefato a ser clonado. 3. Instancia um novo artefato e atribui o nome e a descrição informados. 4. Copia os membros e os serviços de um artefato para o outro. 5. Executa CDU01 Manter Artefato (Mapear Artefato).
Fluxo Alternativo 01 - Erro na Execução CDU01 Manter Artefato (Mapear Artefato)	
Ações Recebidas	Ações Realizadas
5.1. Erro na execução CDU01 Manter Artefato (Mapear Artefato)	5.2. Armazena erro de falha de clonagem. 5.3. Retorna fluxo principal 6.

Tabela 84 – CDU03 – Detalhamento Cenário 01

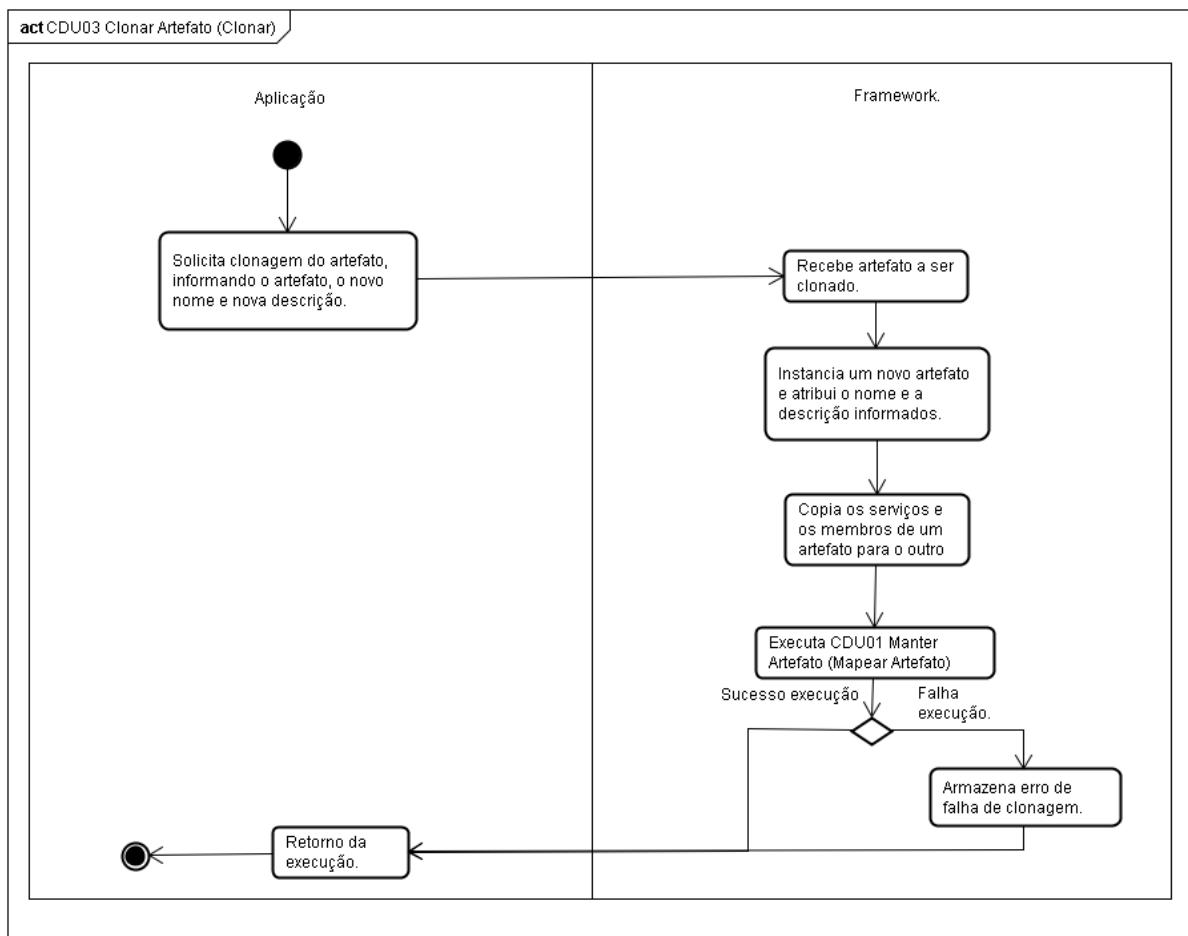


Figura 36 – Diagrama de Atividades Cenário 01 – Clonar Artefato

6.4.4 Caso de Uso 04 – Preencher Artefato

6.4.4.1 Cenário 01 – Montar Interface de Preenchimento.

Descrição do Cenário	Cenário responsável por montar a interface de preenchimento do artefato.
Requisitos Associados	R06 - Preencher artefato
Pré Condições	O usuário deverá estar logado no sistema com permissão de Preenchedor.
Pós Condições	A interface de preenchimento do artefato.
Atores	Preenchedor
Fluxo Principal	
Ações Recebidas	Ações Realizadas
7. Solicita interface de preenchimento do artefato, informando o projeto em que o artefato está sendo preenchido e a tecnologia da visão.	8. Carrega membros e serviços do artefato. 9. Carrega os TipoMembro visão da tecnologia informada dos membros do artefato. 10. Monta interface de preenchimento,

	aplicando as propriedades dos TipoMembro modelos, a partir dos membros, nos TipoMembro visão.
11. Retorno da interface de preenchimento.	
Fluxo Alternativo 01 - Projeto informado	
Ações Recebidas	Ações Realizadas
5.4. Projeto informado	<p>5.5. Carrega as informações preenchidas do artefato no projeto caso existam.</p> <p>5.6. Retorno fluxo principal 4.</p>
Fluxo Alternativo 02 - Artefato sendo preenchido no projeto.	
Ações Recebidas	Ações Realizadas
4.1. Artefato sendo preenchido no projeto.	<p>4.2. Bloqueia interface para o não preenchimento.</p> <p>4.3. Retorno fluxo principal 5.</p>
Fluxo Alternativo 03 - Artefato não está sendo preenchido no projeto.	
Ações Recebidas	Ações Realizadas
4.1. Artefato não está sendo preenchido no projeto.	<p>4.2. Bloqueia artefato no projeto.</p> <p>4.3. Retorno fluxo principal 5.</p>

Tabela 85 – CDU04 – Detalhamento Cenário 01

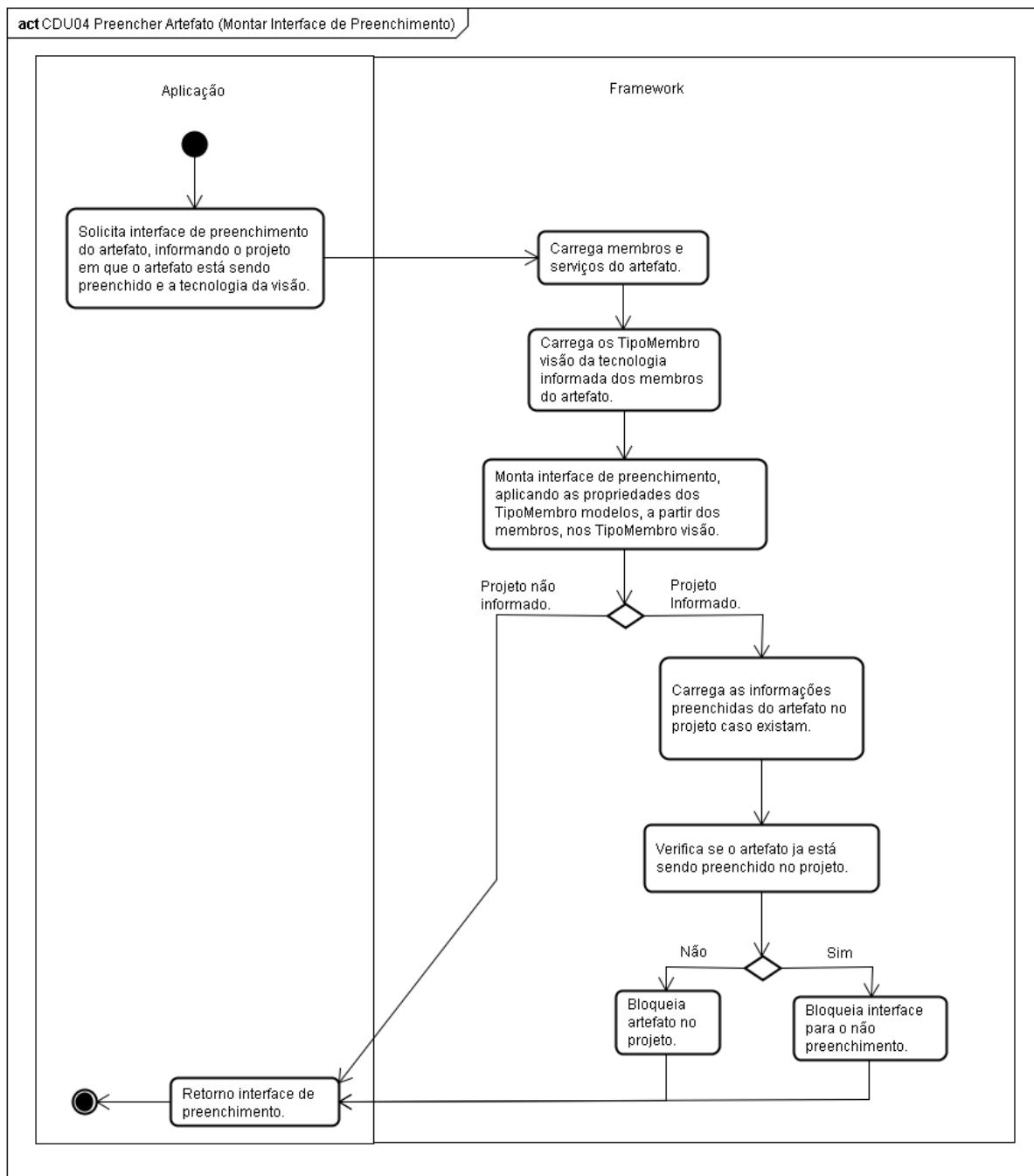


Figura 37 – Diagrama de Atividades Cenário 01 – Montar Interface de Preenchimento

6.4.4.2 Cenário 02 – Interface de Preenchimento do Membro.

Descrição do Cenário	Cenário responsável por retornar a interface de preenchimento do membro.
Requisitos Associados	R06 - Preencher artefato
Pré Condições	O usuário deverá estar logado no sistema com permissão de Preenchedor.
Pós Condições	A interface de preenchimento do membro.
Atores	Preenchedor
Fluxo Principal	

Ações Recebidas	Ações Realizadas
<p>1. Solicita interface de preenchimento do membro.</p> <p>3. Retorno interface de preenchimento do membro.</p>	<p>2. Solicita interface de preenchimento do TipoMembro visão.</p>

Tabela 86 – CDU04 – Detalhamento Cenário 02

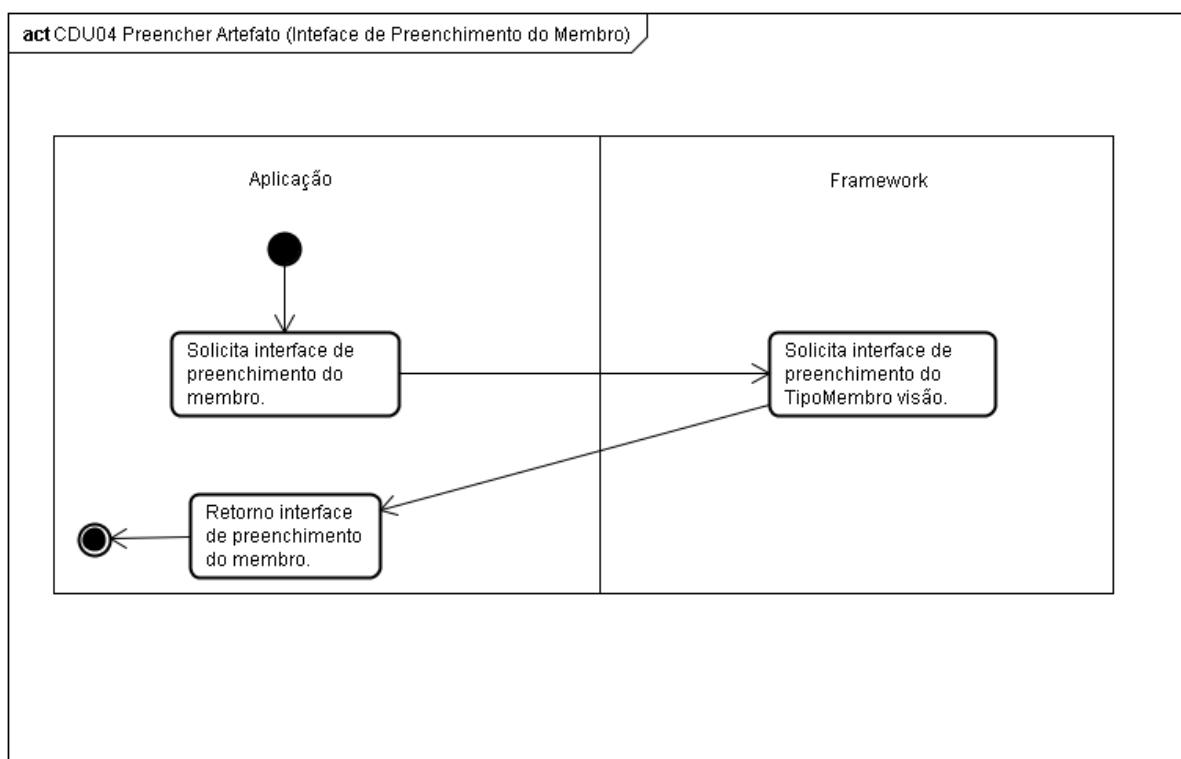


Figura 38 – Diagrama de Atividades Cenário 02 – Interface de Preenchimento do Membro

6.4.4.3 Cenário 03 – Armazenamento de Valor informado ao Membro.

Descrição do Cenário	Cenário responsável por armazenar um valor informado a um membro em sua interface de preenchimento.
Requisitos Associados	R06 - Preencher artefato
Pré Condições	O usuário deverá estar logado no sistema com permissão de Preenchedor.
Pós Condições	A interface de visualização do membro.
Atores	Preenchedor
Fluxo Principal	
Ações Recebidas	Ações Realizadas
<p>1. Solicita armazenamento de valor informado do membro.</p>	<p>2. Solicita validação do valor ao TipoMembro modelo.</p> <p>3. Armazena o valor informado no membro.</p>

	4. Solicita interface de visualização ao TipoMembro visão. 5. Armazena sucesso de execução.
6. Retorno da interface de preenchimento.	
Fluxo Alternativo 01 - Valor informado incorreto.	
Ações Recebidas	Ações Realizadas
2.1. Valor informado invalido. 2.3. Retorno fluxo principal 6.	2.2. Armazena erro de valor incorreto.

Tabela 87 – CDU04 – Detalhamento Cenário 03

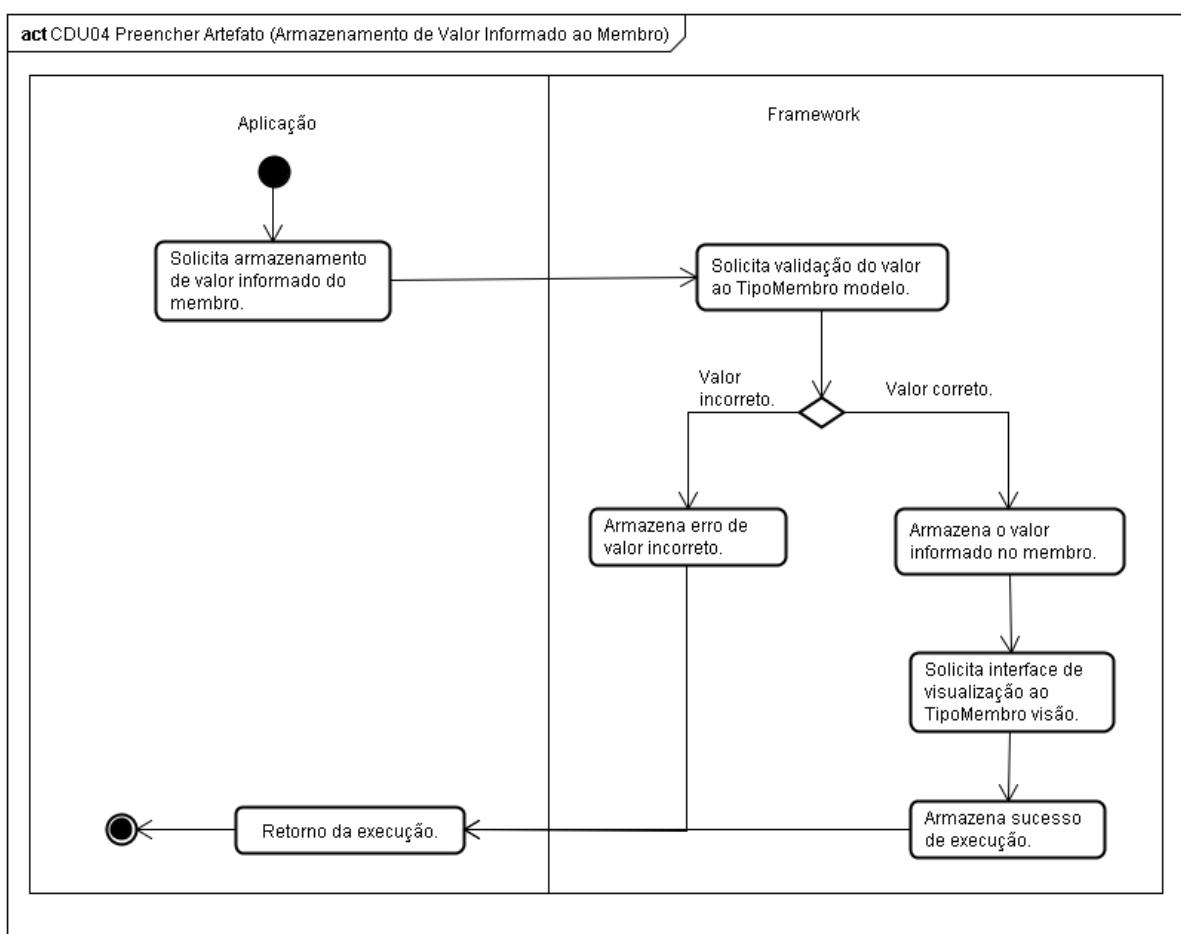


Figura 39 – Diagrama de Atividades Cenário 03 – Armazenamento de Valor informado ao Membro

6.4.4.4 Cenário 04 – Cancelar Preenchimento do Artefato.

Descrição do Cenário	Cenário responsável por cancelar o preenchimento do artefato.
Requisitos Associados	R06 - Preencher artefato
Pré Condições	O usuário deverá estar logado no sistema com permissão de Preenchedor.
Pós Condições	O cancelamento do preenchimento do artefato.
Atores	Preenchedor
Fluxo Principal	
Ações Recebidas	Ações Realizadas

1. Solicita cancelamento de preenchimento de artefato.	2. Remove todos os valores informados dos membros do artefato.
3. Desbloqueia artefato do projeto.	4. Armazena sucesso de execução.
5. Retorno de execução.	

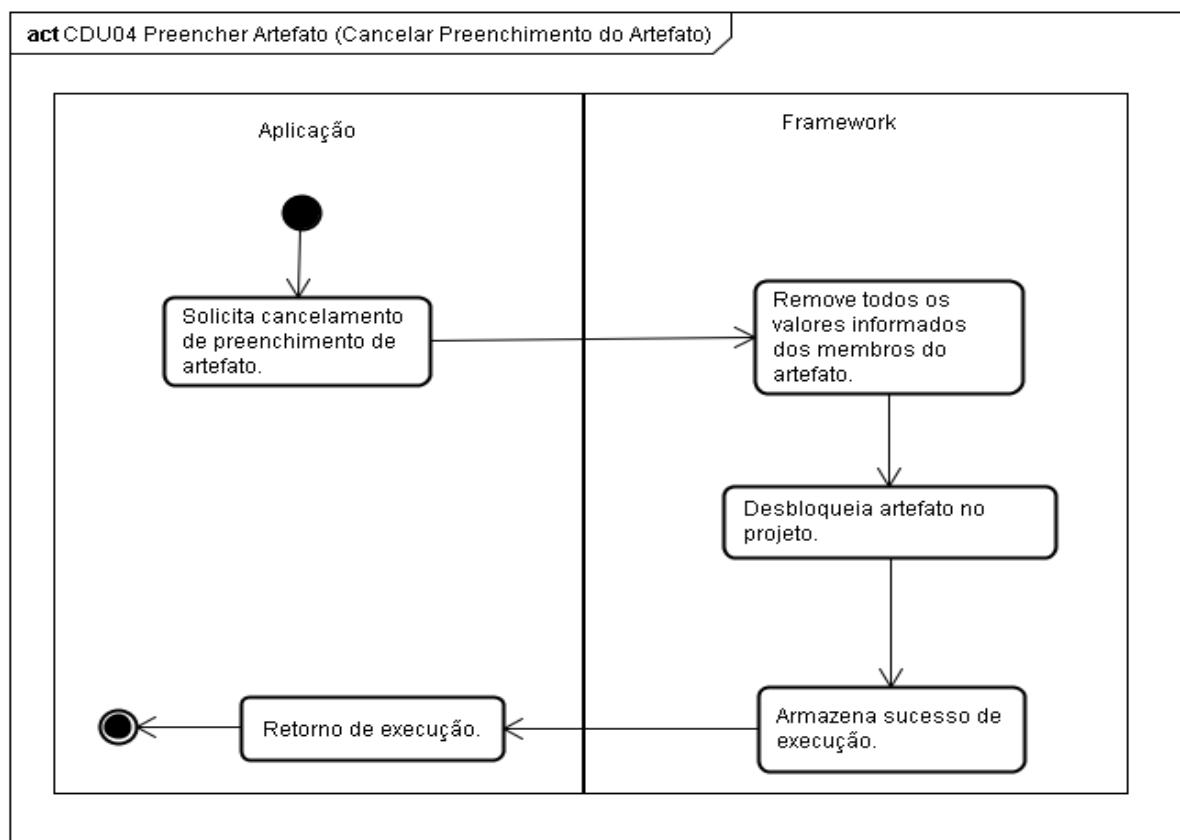
Tabela 88 – CDU04 – Detalhamento Cenário 04

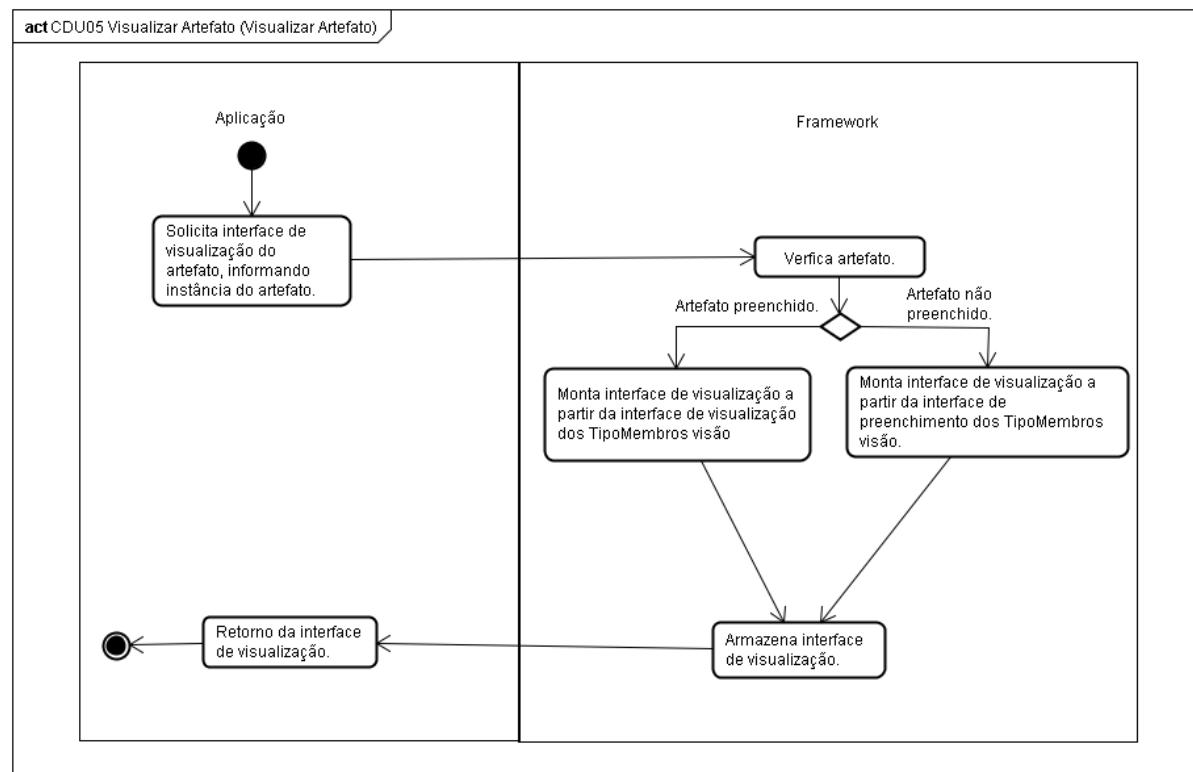
Figura 40 – Diagrama de Atividades Cenário 04 – Cancelar Preenchimento do Artefato

6.4.5 Caso de Uso 05 – Visualizar Artefato

6.4.5.1 Cenário 01 – Visualizar Artefato

Descrição do Cenário	Cenário responsável por montar a interface de visualização do artefato.
Requisitos Associados	R07 - Visualizar artefato
Pré Condições	O usuário deverá estar logado no sistema com permissão de Preenchedor.
Pós Condições	A interface de preenchimento do artefato.
Atores	Preenchedor, Montador
Fluxo Principal	
Ações Recebidas	Ações Realizadas

1. Solicita interface de visualização do artefato, informando a instância do artefato.	2. Verifica o Artefato. 3. Monta interface de visualização a partir dos TipoMembros visão. 4. Armazena interface de visualização.
5. Retorno da interface de visualização.	
Fluxo Alternativo 01 - Artefato preenchido.	
Ações Recebidas	Ações Realizadas
5.7. Artefato preenchido.	5.8. Monta interface de visualização a partir da interface de visualização dos TipoMembros visão. 5.9. Retorna fluxo principal 4.
Fluxo Alternativo 01 - Artefato não preenchido.	
Ações Recebidas	Ações Realizadas
3.1. Artefato não preenchido.	3.2. Monta interface de visualização a partir da interface de preenchimento dos TipoMembros visão. 3.3. Retorna fluxo principal 4.

Tabela 89 – CDU05 – Detalhamento Cenário 01**Figura 41 – Diagrama de Atividades Cenário 01 – Visualizar Artefato**

6.4.6 Caso de Uso 06 – Publicar Artefato

6.4.6.1 Cenário 01 – Publicar Artefato

Descrição do Cenário	Caso de uso responsável por realizar a publicação do artefato.
Requisitos Associados	R08 - Publicar Artefato
Pré Condições	O usuário deverá estar logado no sistema com permissão de Preenchedor.
Pós Condições	A interface de impressão do artefato.
Atores	Preenchedor
Fluxo Principal	
Ações Recebidas	
12. Solicita publicação do artefato.	13. Busca serviços de publicação cadastrados no framework. 14. Armazena solicitação de escolha de serviço de publicação. 16. Executa serviço de publicação. 17. Armazena interface de impressão do artefato. 18. Armazena sucesso de execução.
15. Seleciona serviço de publicação.	
19. Retorno da execução.	
Fluxo Alternativo 01 - Serviço de publicação não selecionado.	
Ações Recebidas	
5.10. Serviço de publicação não selecionado.	5.11. Cancelamento de publicação.
5.12. Retorno fluxo principal 7.	
Fluxo Alternativo 02 - Erro de execução do serviço de publicação.	
Ações Recebidas	
5.1. Erro de execução do serviço de publicação.	5.2. Armazenamento do erro de execução do serviço.
5.3. Retorno fluxo principal 7.	

Tabela 90 – CDU06 – Detalhamento Cenário 01

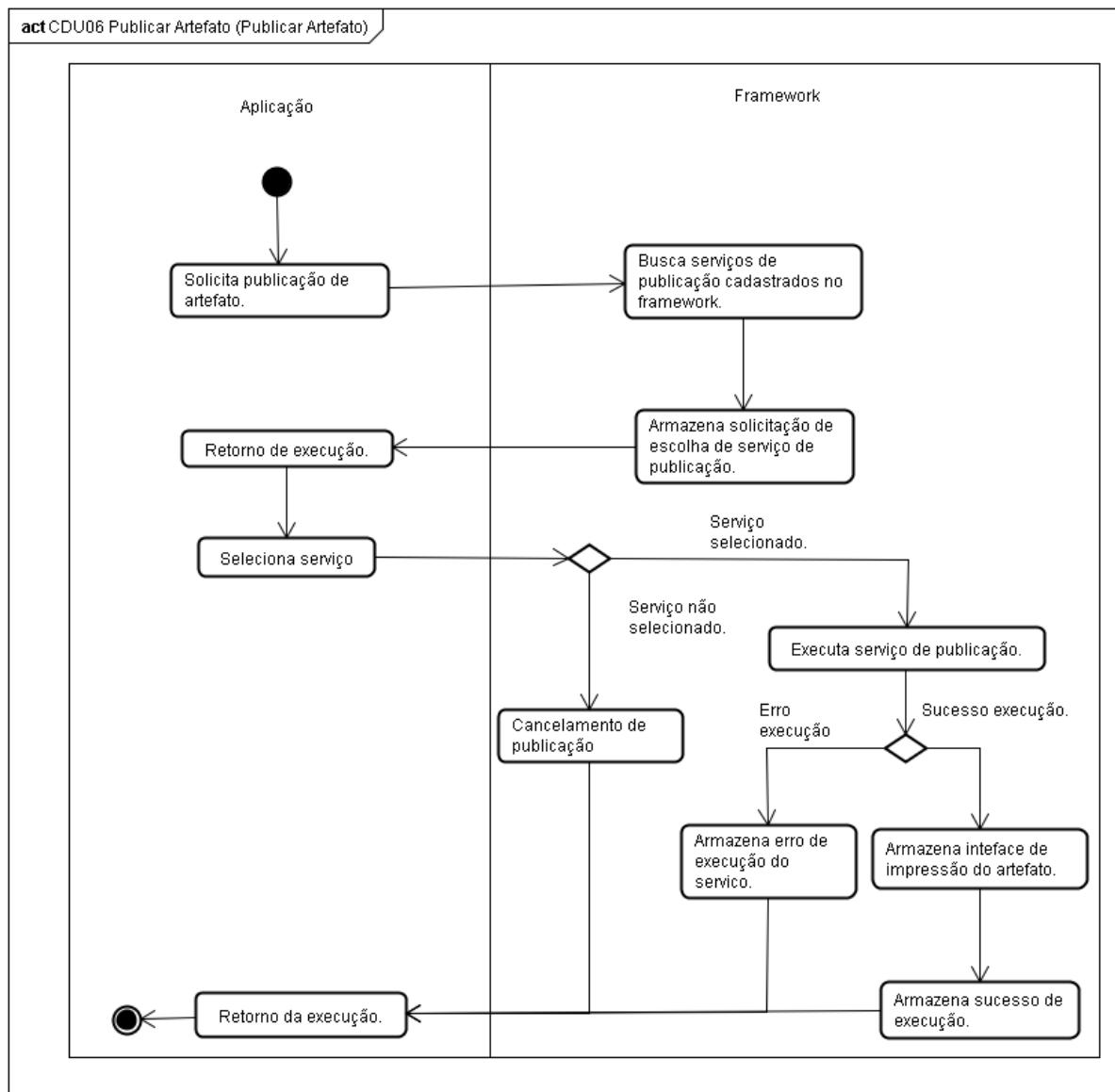


Figura 42 – Diagrama de Atividades Cenário 01 – Publicar Artefato

6.4.7 Caso de Uso R07 - Manter Membro

6.4.7.1 Cenário 01 - Mapear Membro

Descrição do Cenário	Cenário responsável por realizar a persistência de um Membro no framework.
Requisitos Associados	R02 - Manter Membro
Pré Condições	O usuário deverá estar logado no sistema com permissão de usuário montador.
Pós Condições	O Membro persistido no framework.
Atores	Montador
Fluxo Principal	
Ações Recebidas	
13. Recebe uma instância do Membro para	Ações Realizadas

ser persistida.	14. Verifica se os atributos nome e descrição do Membro foram informados. 15. Solicita o TipoMembro a verificação dos parâmetros. 16. Cadastro do Membro e dos parâmetros no framework. 17. Armazena sucesso da execução.
18. Retorna resultado da execução.	
Fluxo Alternativo 1 – Atributos nome ou descrição do Membro não informados	
Ações Recebidas	Ações Realizadas
2.8. Atributos nome ou descrição do framework não formam informados. 2.11. Retorno fluxo principal 6.	2.9. Armazenamento do nome dos campos não informados. 2.10. Armazenamento do erro de falha de execução.
Fluxo Exceção 2 – Parâmetros do TipoMembro ausentes ou incorretos.	
Ações Recebidas	Ações Realizadas
4.4. Parâmetros do TipoMembro ausentes ou incorretos. 4.7. Retorno fluxo principal 6.	4.5. Armazenamento dos nomes dos parâmetros ausentes ou incorretos. 4.6. Armazenamento do erro de falha de execução.

Tabela 91 – CDU07 – Detalhamento Cenário 01

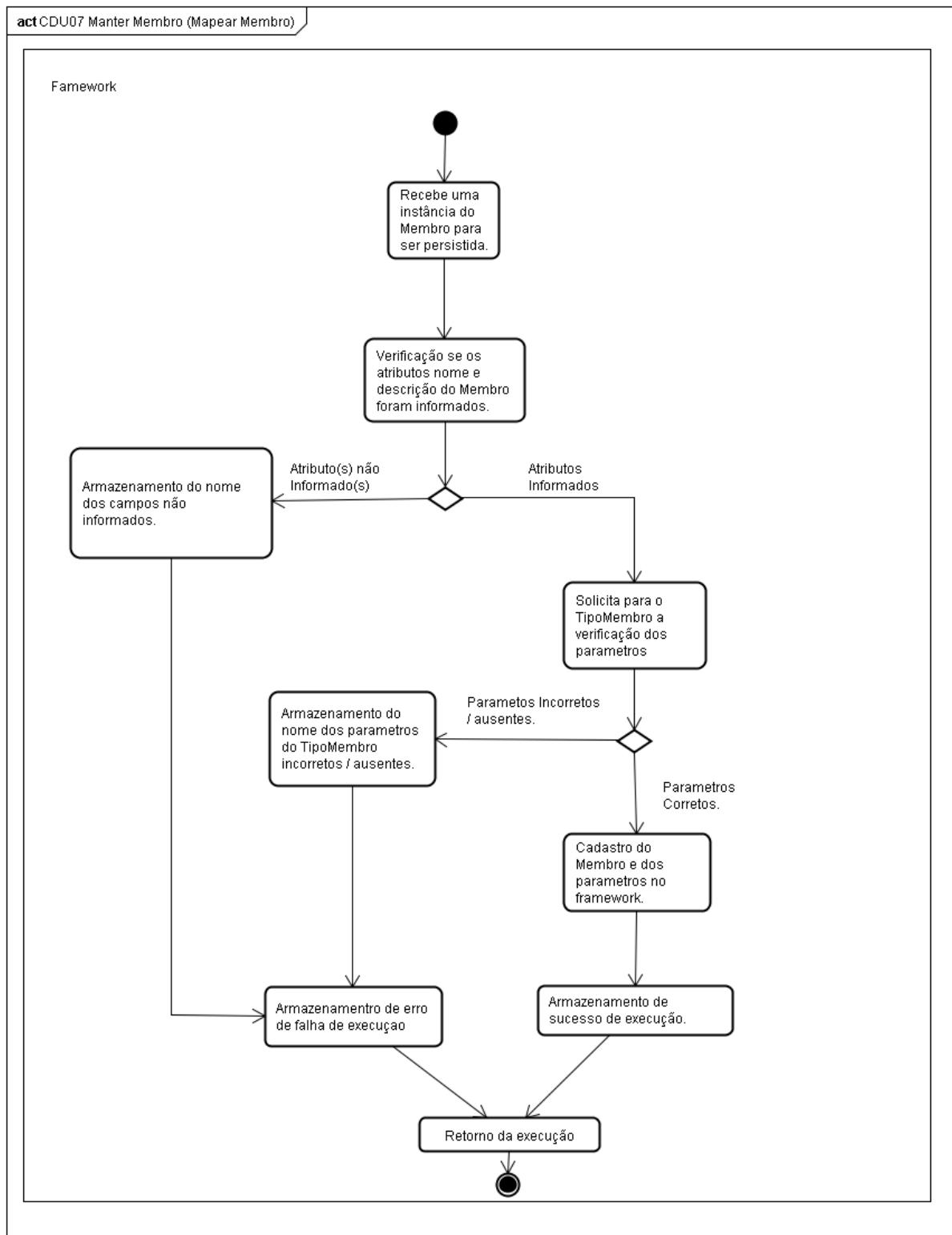


Figura 43 – Diagrama de Atividades Cenário 01 – Incluir Membro

6.4.7.2 Cenário 02 - Pesquisar Membro

Descrição do Cenário	Cenário responsável por realizar a pesquisa de Membros no framework.
----------------------	--

Requisitos Associados	R02 - Manter Membro
Pré Condições	O usuário deverá estar logado no sistema com permissão de usuário montador.
Pós Condições	Uma lista de Membros.
Atores	Montador
Fluxo Principal	
Ações Recebidas	Ações Realizadas
5. Solicita pesquisa de Membro informando o nome ou um TipoMembro.	6. Verifica o tipo de pesquisa informado. 7. Realiza pesquisa. 8. Armazena lista de membros encontrados. 9. Armazena sucesso de execução.
10. Retorno da execução.	
Fluxo Alternativo 1 - Pesquisa por nome do Membro	
Ações Recebidas	Ações Realizadas
2.1. Nome informado para pesquisa de Membro.	2.2. Pesquisa todos os Membros cadastrados no framework que possuem nome parcial ao informado 2.3. Retorna uma lista de Membro.
2.4. Retorno fluxo principal 4.	
Fluxo Alternativo 2 - Pesquisa por TipoMembro do Membro.	
Ações Recebidas	Ações Realizadas
2.1. TipoMembro informado para pesquisa de Membro.	2.2. Pesquisa todos os Membros cadastrados no framework que são instâncias do TipoMembro informado. 2.3. Retorna uma lista de Membro.
2.4. Retorno fluxo principal 4.	
Fluxo Alternativo 3 - Nenhum Membro encontrado.	
Ações Recebidas	Ações Realizadas
3.1. Nenhum membro encontrado.	3.2. Armazena lista vazia. 3.3. Armazena erro de nenhum membro encontrado.
3.4. Retorno fluxo principal 6.	

Tabela 92 – CDU07 – Detalhamento Cenário 02

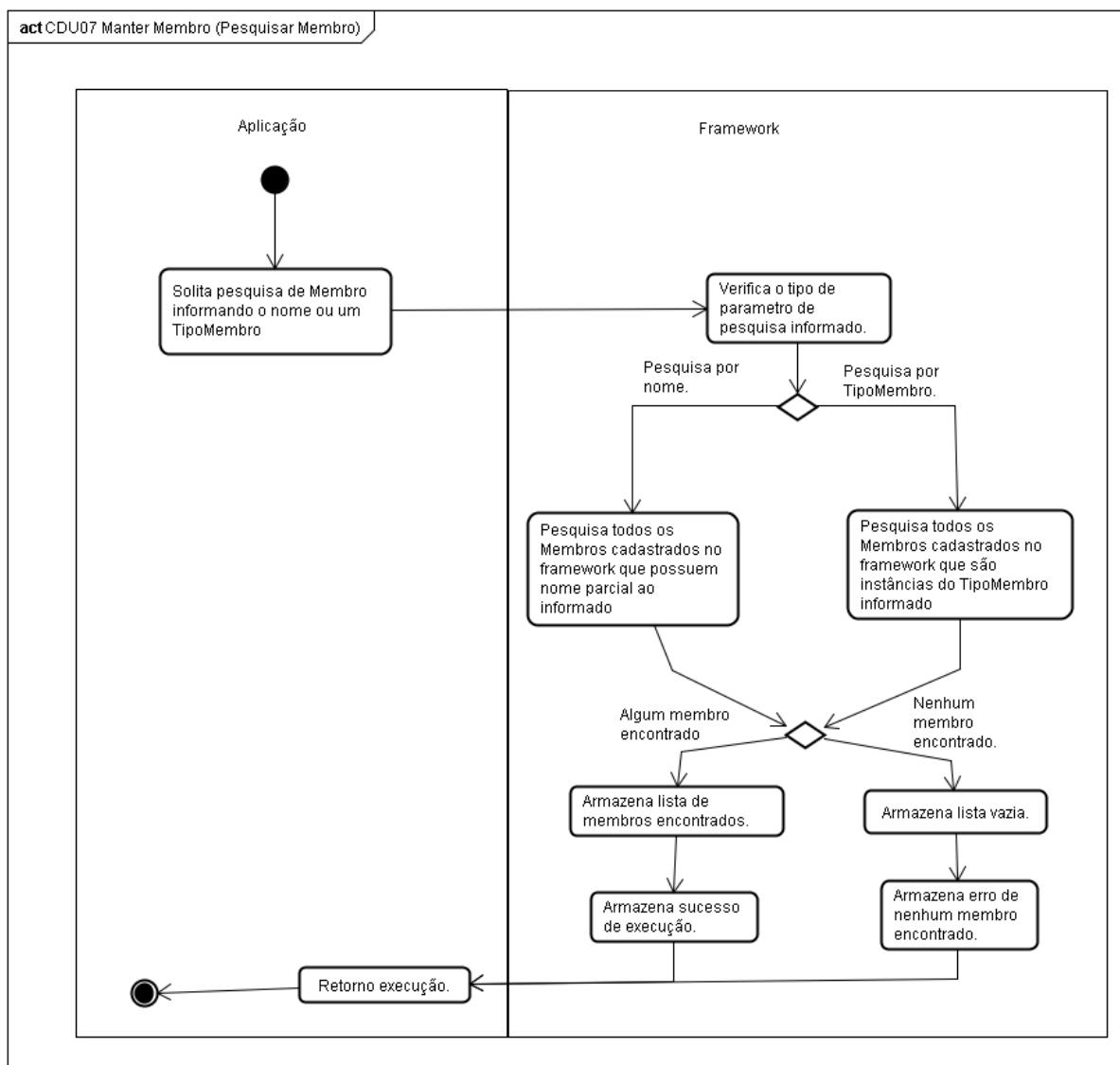


Figura 44 – Diagrama de Atividades Cenário 02 – Pesquisar Membro

6.4.7.3 Cenário 03 - Alterar Membro

Descrição do Cenário	Cenário responsável por realizar a alteração de um membro no framework.
Requisitos Associados	R02 - Manter Membro
Pré Condições	O usuário deverá estar logado no sistema com permissão de usuário montador.
Pós Condições	Os dados do Membro alterado no framework.
Atores	Montador
Fluxo Principal	
Ações Recebidas	
5. Recebe instância do Membro para ser alterada.	6. Verifica se o membro esta sendo utilizado por outro artefato. 7. Realiza a alteração dos parâmetros do
Ações Realizadas	

8. Retorna instância do Membro	Membro no framework.
Fluxo Alternativo 1 - Membro sendo utilizado por outro artefato	
Ações Recebidas	Ações Realizadas
2.1. O Membro está sendo utilizado por outro artefato.	2.2. Cria outra instância do Membro, com os mesmos parâmetros, apagando o nome, a descrição e o código do membro.
2.3. Retorno fluxo principal 4.	

Tabela 93 – CDU07 – Detalhamento Cenário 03

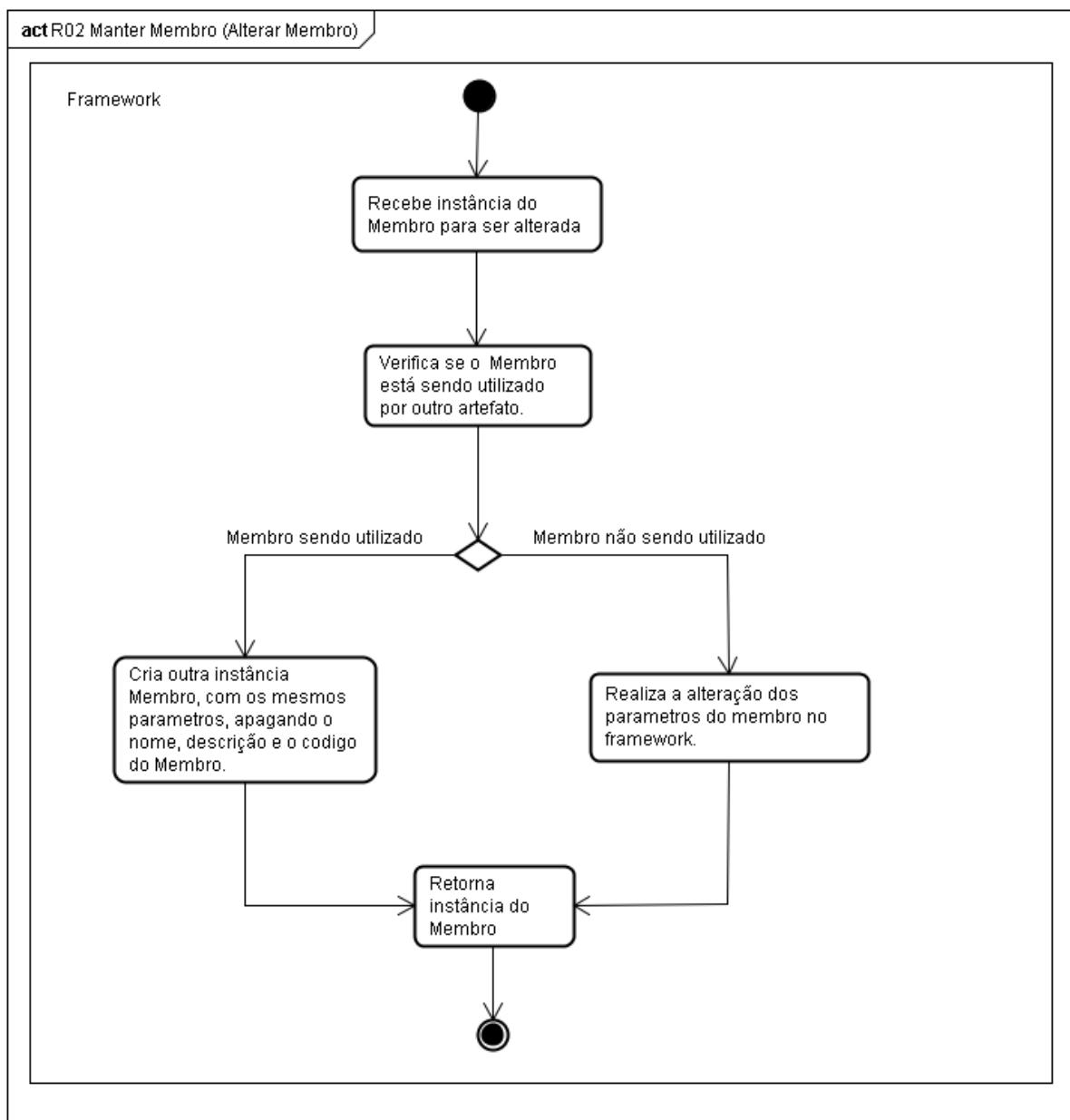


Figura 45 – Diagrama de Atividades Cenário 03 – Alterar Membro

6.4.8 Caso de Uso 08 – Manter TipoMembro

6.4.8.1 Cenário 01 – Mapear TipoMembro

Descrição do Cenário	Cenário responsável por inserir, remover e versionar os TipoMembros do framework.
Requisitos Associados	R01 – Manter TipoMembro
Pré Condições	O usuário deverá estar logado no sistema com permissão de desenvolvedor. Os arquivos do TipoMembro a serem importados deverão estar conforme especificação.
Pós Condições	Os TipoMembros mapeados no framework.
Atores	Desenvolvedor
Fluxo Principal	
Ações Recebidas	Ações Realizadas
20. Aplicação captura os arquivos dos TipoMembros visão e modelo. 21. Execução do mapeamento de TipoMembros. 28. Retorno da execução.	22. Verificação se os arquivos seguem a especificação. 23. Executa cenário 02 Verificar Espelho. 24. Execução do cenário CDU13 Realizar Versionamento (Versionar). 25. Executa mapeamento do TipoMembro. 26. Lista de arquivos de importação esta vazia 27. Armazenamento de sucesso de execução
Fluxo Alternativo 01 - Arquivos não seguem a especificação	
Ações Recebidas	Ações Realizadas
3.1. Arquivos não seguem a especificação.	3.2. Armazenamento dos nomes dos arquivos que não seguem a especificação. 3.3. Armazenamento do erro de formato incorreto. 3.4. Retorno ao item 9 do fluxo principal.
Fluxo Alternativo 02 – Verificação de espelhos incorreta (Lista de espelhos não vazia)	
Ações Recebidas	Ações Realizadas
4.1. Verificação de espelhos incorreta.	4.2. Armazenamento do(s) nome(s) do(s) arquivo(s) que não possuem espelho. 4.3. Armazenamento do erro de espelho incorreto 4.4. Retorno fluxo principal 9.
Fluxo Alternativo 04 – Nova Versão do TipoMembro	
Ações Recebidas	Ações Realizadas
5.1. Nova versão do TipoMembro.	5.2. Executa Rotina de Nova Versão de TipoMembro. 5.3. Retorno fluxo principal 7.
Fluxo Alternativo 05 – Nova Revisão do TipoMembro	
Ações Recebidas	Ações Realizadas
5.1. Nova revisão do TipoMembro.	

	<p>5.2. Executa Rotina de Nova Revisão de TipoMembro.</p> <p>5.3. Retorno fluxo principal 7.</p>
Fluxo Alternativo 06 – Mesma Versão do TipoMembro	
Ações Recebidas	Ações Realizadas
5.1. Mesma versão/revisão do TipoMembro.	<p>5.2. Armazena mensagem de TipoMembro já mapeado no framework, informando o nome do arquivo.</p> <p>5.3. Retorno fluxo principal 7.</p>
Fluxo Alternativo 03 – Lista de arquivos a ser importada não esta vazia	
Ações Recebidas	Ações Realizadas
7.1. Lista de arquivos a ser importada não esta vazia.	7.2. Retorno fluxo principal 5.

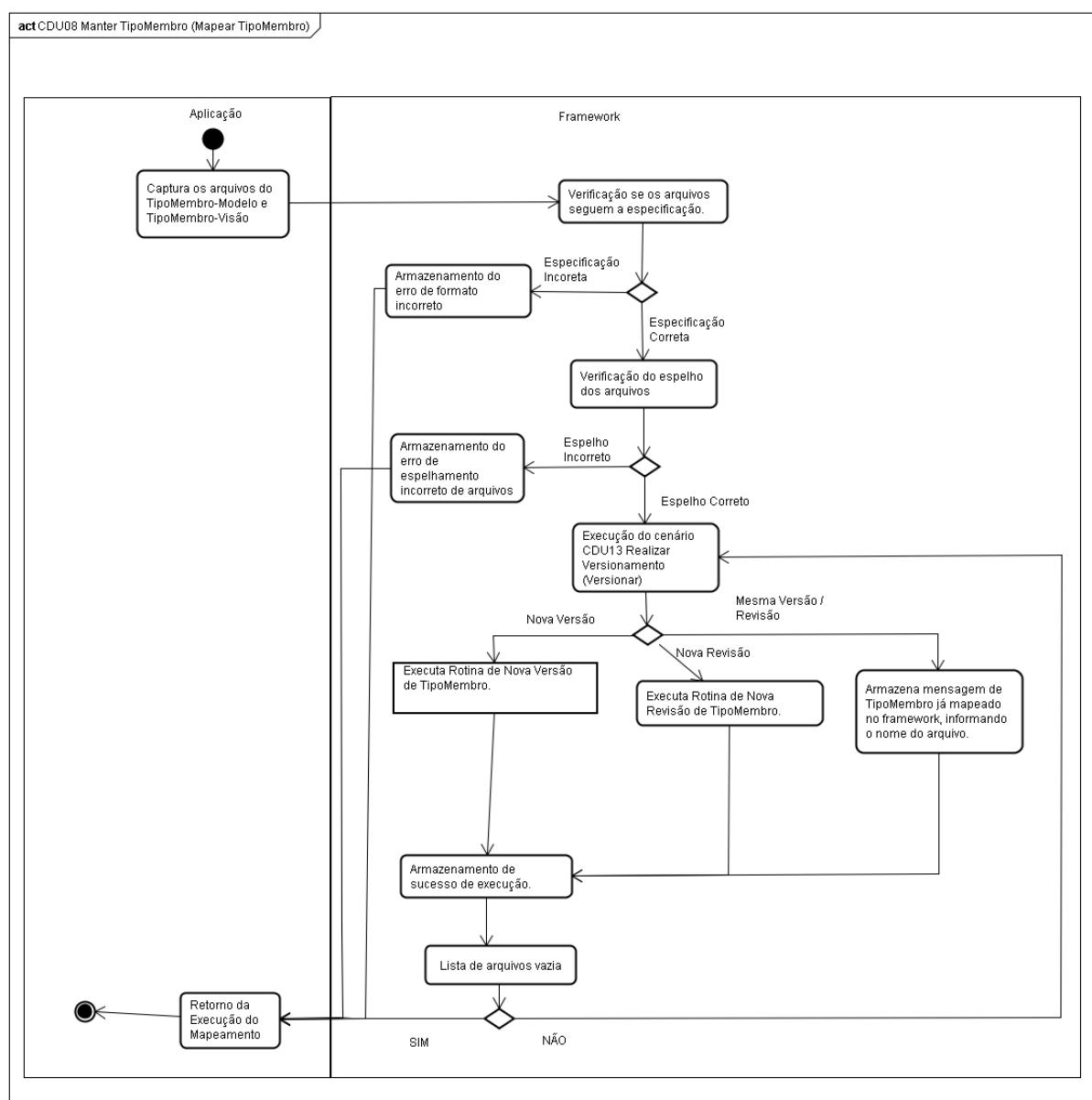
Tabela 94 – CDU08 – Detalhamento Cenário 01

Figura 46 – Diagrama de Atividades Cenário 01 – Mapear TipoMembro

6.4.8.2 Cenário 02 – Verificação de Espelho

Descrição do Cenário	Cenário responsável por verificar se para cada arquivo TipoMembro visão existe um arquivo
Requisitos Associados	R01 Manter TipoMembro
Pré Condições	O usuário deverá estar logado no sistema com permissão de desenvolvedor.
Pós Condições	Espelho verificado
Atores	Desenvolvedor
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1. Verificação de espelho dos arquivos TipoMembros. 5. Retornar lista de nomes dos TipoMembros com espelho incorreto.	2. Selecionar os arquivos correspondentes aos TipoMembro visão. 3. Verificar se existe um TipoMembro modelo cadastrado no framework ou informado na lista de arquivos a ser importada. 4. Não existe TipoMembro visão não verificados.
Fluxo Alternativo 01 - Não existe TipoMembro modelo cadastrado no framework ou informado na lista de arquivos a ser importada	
Ações Recebidas	Ações Realizadas
3.1. Não existe arquivo TipoMembro modelo cadastrado no framework ou arquivo informado na lista de arquivos a ser importada para um determinados TipoMembros visão.	3.2. Armazenamento do nome do TipoMembro visão na lista de nomes de TipoMembros com espelho incorretos. 3.3. Retorno fluxo principal 4.
Fluxo Alternativo 02 - Ainda existe TipoMembro visão não verificado.	
Ações Recebidas	Ações Realizadas
4.1. Ainda existe TipoMembro visão não verificado.	4.2. Retorno fluxo principal 3.

Tabela 95 – CDU08 – Detalhamento Cenário 02

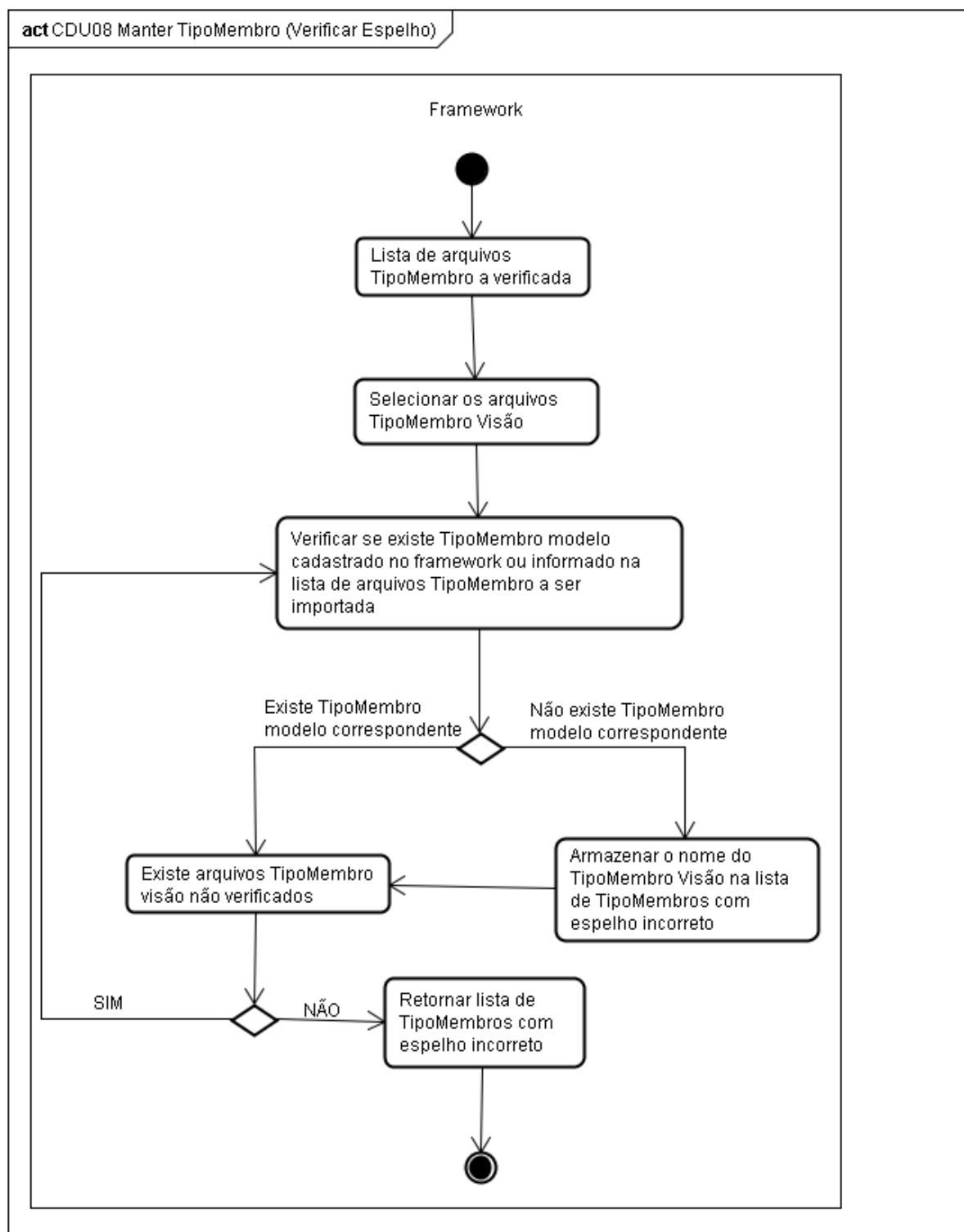


Figura 47 – Diagrama de Atividades Cenário 02 – Verificação de Espelho

6.4.8.3 Cenário 03 – Executar Rotina de Nova Versão de TipoMembro

Descrição do Cenário	Cenário responsável por armazenar uma nova versão do TipoMembro
Requisitos Associados	R01 Manter TipoMembro

Pré Condições	O usuário deverá estar logado no sistema com permissão de desenvolvedor.
Pós Condições	Arquivo armazenado no framework
Atores	Desenvolvedor
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1. Nova versão do TipoMembro.	2. Armazenamento do arquivo referente ao TipoMembro no diretório específico. 3. Cadastramento do TipoMembro no framework.
4. Retorno da operação.	
Fluxo Alternativo 01 -Falha na gravação do arquivo	
Ações Recebidas	Ações Realizadas
2.1. Falha na gravação do arquivo	2.2. Armazenamento do erro de falha de gravação. 2.3. Retorno fluxo principal 4

Tabela 96 – CDU08 – Detalhamento Cenário 03

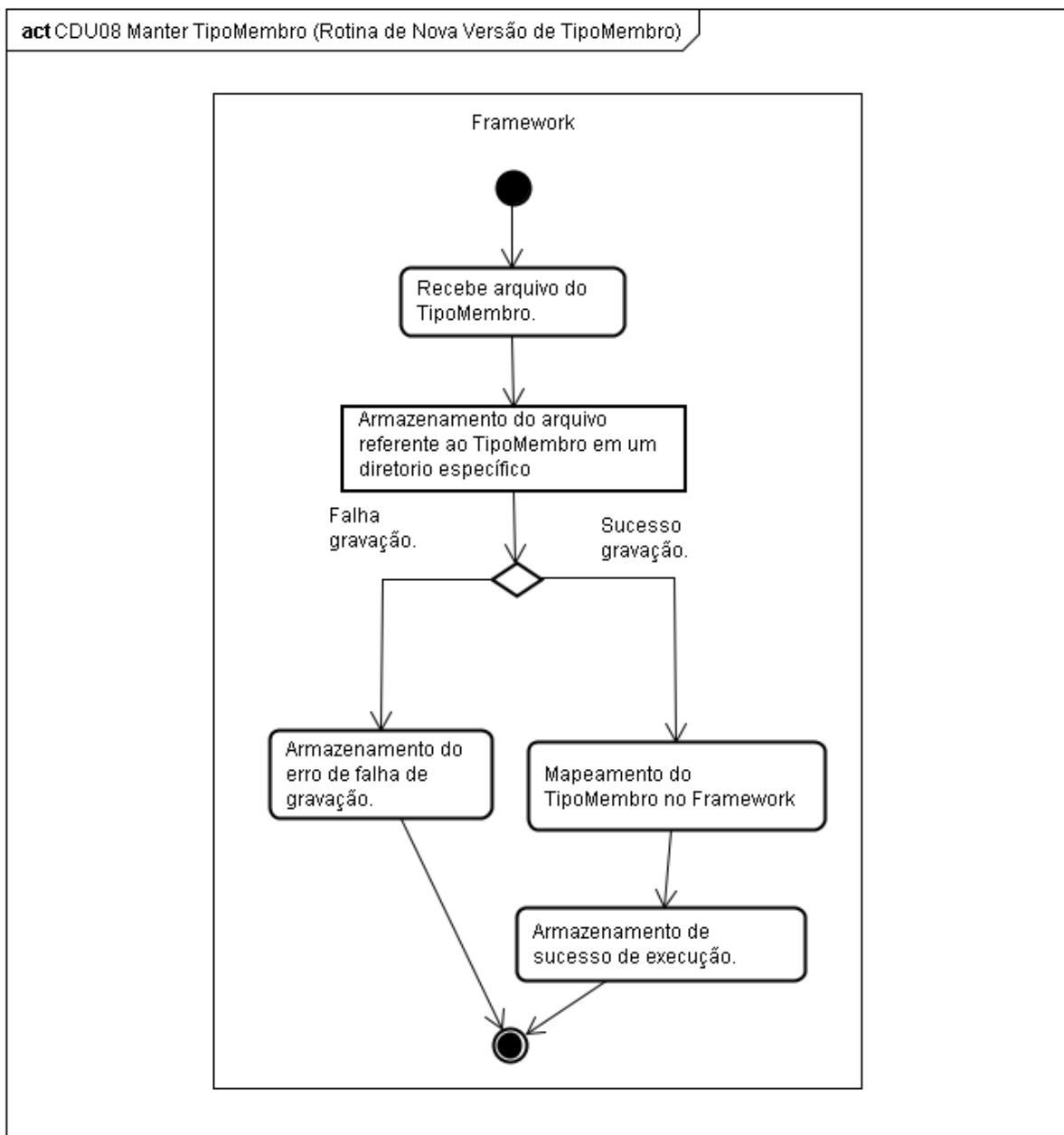


Figura 48 – Diagrama de Atividades Cenário 03 – Executar Rotina de Nova Versão de TipoMembro

6.4.8.4 Cenário 04 – Executar Rotina de Nova Revisão de TipoMembro

Descrição do Cenário	Cenário responsável por armazenar uma nova revisão do TipoMembro.	
Requisitos Associados	R01 Manter TipoMembro	
Pré Condições	O usuário deverá estar logado no sistema com permissão de desenvolvedor	
Pós Condições	Arquivo armazenado no framework	
Atores	Desenvolvedor	
Fluxo Principal		
Ações Recebidas		
1. Nova revisão do TipoMembro	Ações Realizadas	

	<p>2. Transferência do arquivo de revisão antiga do TipoMembro para o diretório de armazenamento de revisão.</p> <p>3. Armazenamento do arquivo referente ao TipoMembro de revisão nova no diretório específico.</p>
4. Retorno da operação.	
	Fluxo Alternativo 01 - Falha na gravação do arquivo
Ações Recebidas	Ações Realizadas
3.1. Falha na gravação do arquivo	<p>3.2. Armazenamento do erro de falha de gravação.</p> <p>3.3. Retorno fluxo principal 4.</p>

Tabela 97 – CDU08 – Detalhamento Cenário 04

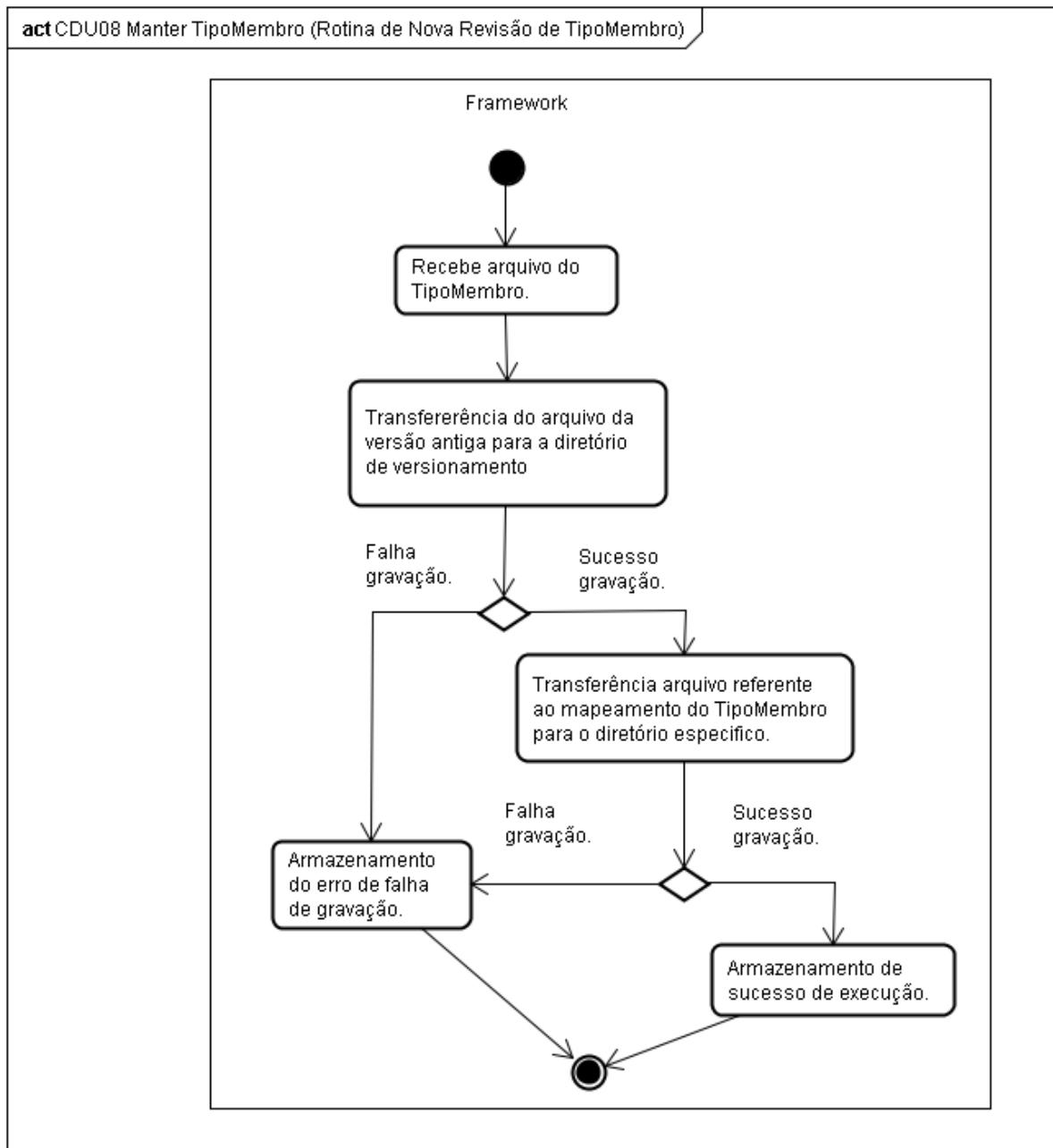


Figura 49 – Diagrama de Atividades Cenário 04 – Executar Rotina de Nova Revisão de TipoMembro

6.4.9 Caso de Uso 09 – Manter Modelo

6.4.9.1 Cenário 01 – Montar Modelo

Descrição do Cenário	Cenário responsável por montar um modelo.
Requisitos Associados	R09 - Manter Modelo
Pré Condições	O usuário deverá estar logado no sistema com permissão de Montador
Pós Condições	O modelo mapeado no framework

Atores	Montador
Fluxo Principal	
Ações Recebidas	Ações Realizadas
29. Solicita criação do modelo informando seu nome e sua descrição. 5. Retorno da execução. 6. Executa cenário Preencher Modelo 7. Solicita persistência de modelo. 9. Retorno execução.	30. Verifica duplicidade de nome de modelo. 31. Instancia novo modelo, atribuindo seu nome e sua descrição. 32. Armazena sucesso de execução. 8. Executa cenário manter modelo
Fluxo Alternativo 01 - Nome do modelo já cadastrado no framework	
Ações Recebidas	Ações Realizadas
5.13. Nome do modelo já cadastrado no framework 5.15. Retorno fluxo principal 9.	5.14. Armazena erro de nome de modelo já cadastrado no framework.

Tabela 98 – CDU09 – Detalhamento Cenário 01

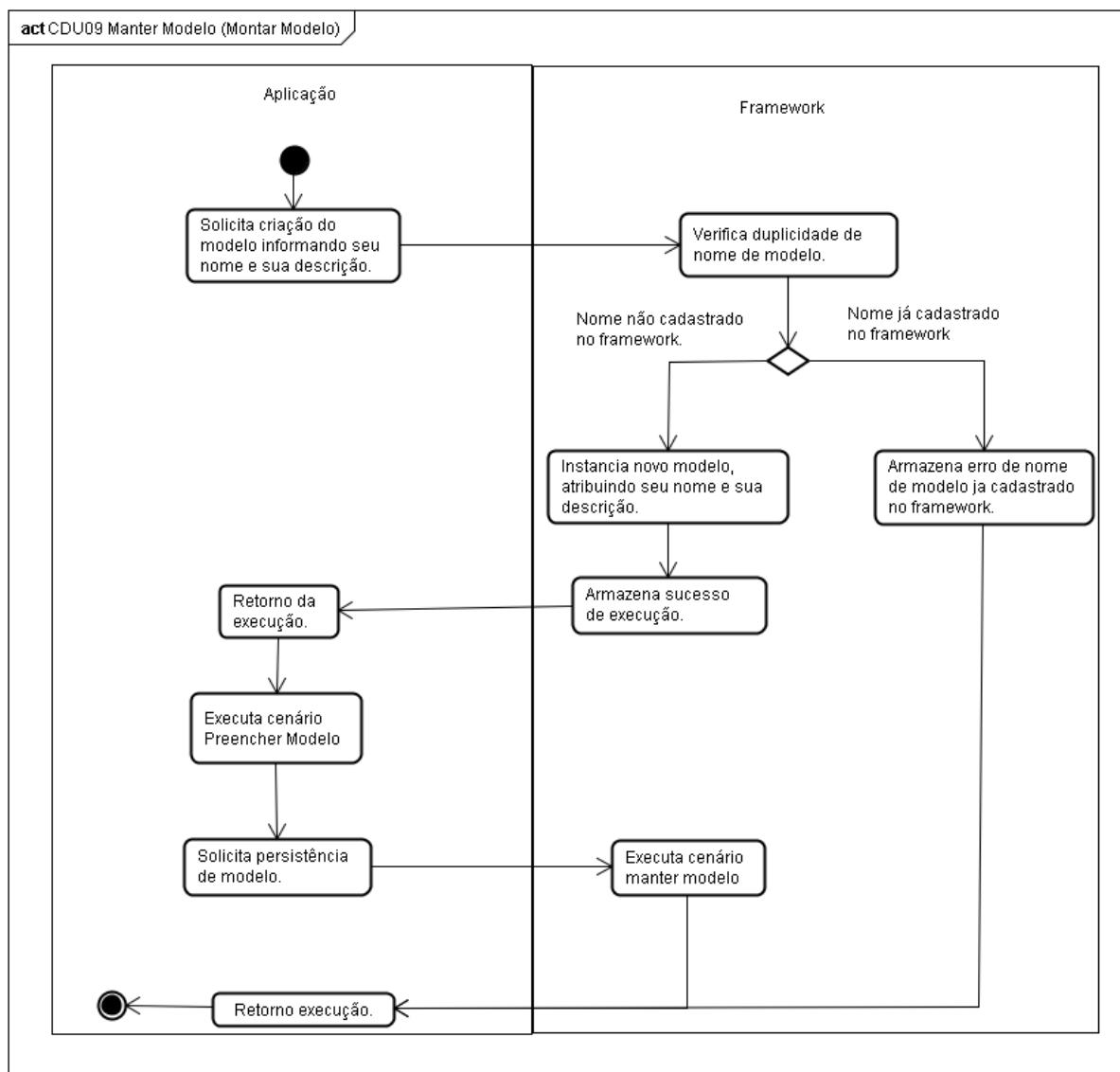


Figura 50 – Diagrama de Atividades Cenário 01 – Montar Modelo

6.4.9.2 Cenário 02 – Preencher Modelo

Descrição do Cenário	Cenário responsável por adicionar, remover e alterar os artefatos no modelo.
Requisitos Associados	R09 - Manter Modelo
Pré Condições	O usuário deverá estar logado no sistema com permissão de Preenchedor
Pós Condições	O modelo preenchido
Atores	Preenchedor
Fluxo Principal	
Ações Recebidas	
1. Recebe instância do modelo a ser preenchida. 2. Adiciona, remove ou altera um artefato no modelo.	3. Executa ação.
Ações Realizadas	

4. Retorno execução.	
Fluxo Alternativo 01 - Executa ação de inserir artefato no modelo.	
Ações Recebidas	Ações Realizadas
3.1. Novo artefato a ser inserido no modelo.	3.2. Executa cenário incluir artefato.
3.3. Retorno fluxo principal 2.	
Fluxo Alternativo 02 - Executa ação de remover artefato no modelo.	
Ações Recebidas	Ações Realizadas
3.1. Artefato a ser removido do modelo.	3.2. Executa cenário remover artefato.
3.3. Retorno fluxo principal 2.	
Fluxo Alternativo 03 - Executa ação de alterar parâmetros do artefato no modelo.	
Ações Recebidas	Ações Realizadas
3.1. Parâmetros do artefato a ser alterado do modelo.	3.2. Executa cenário alterar parâmetros de artefato..
3.3. Retorno fluxo principal 2.	
Fluxo Alternativo 04- Solicita Persistência do modelo.	
Ações Recebidas	Ações Realizadas
3.1. Solicita persistência do modelo.	3.2. Executa cenário 06 Mapear Modelo. 3.3. Retorno fluxo principal 4.

Tabela 99 – CDU09 – Detalhamento Cenário 02

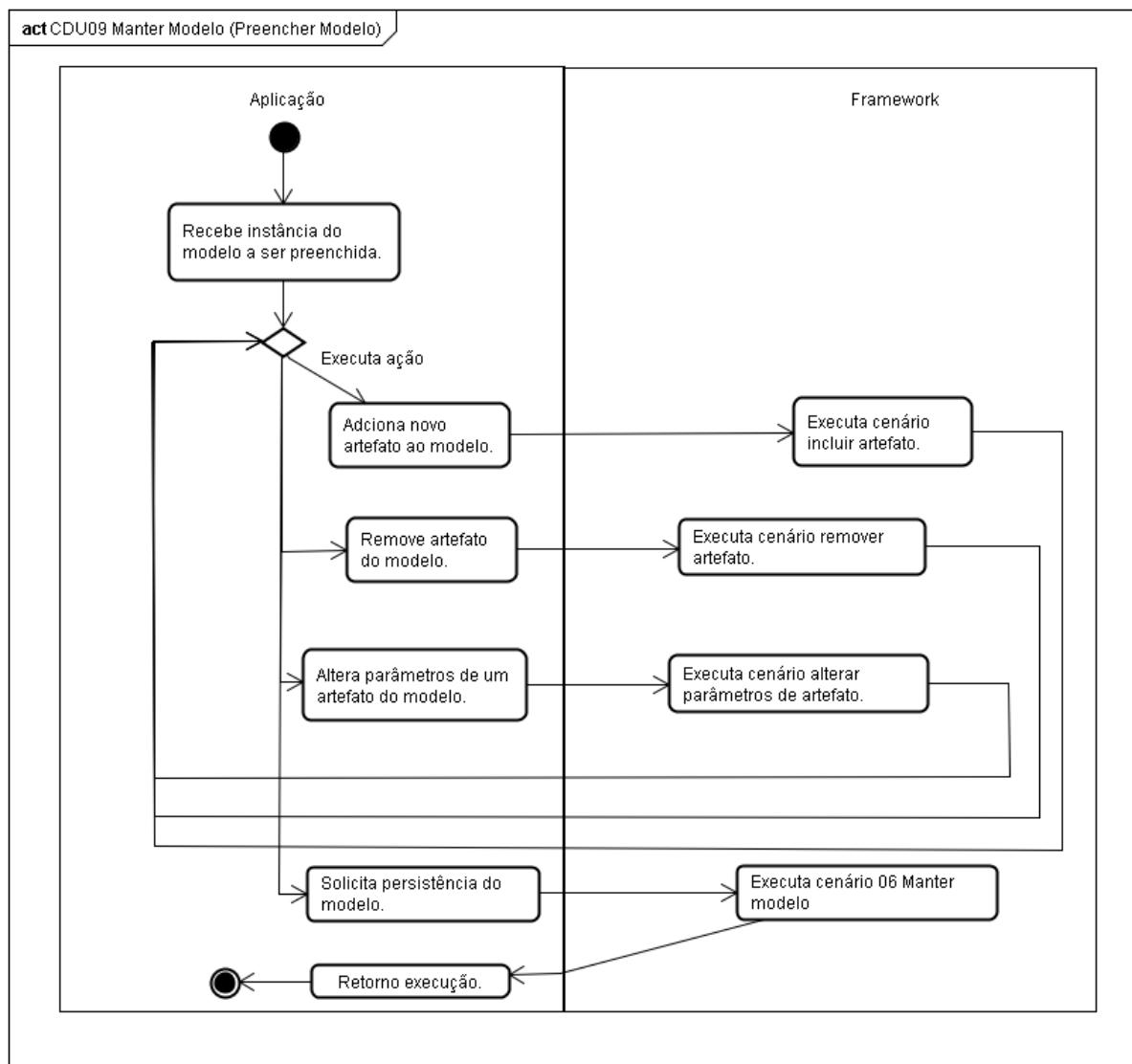


Figura 51 – Diagrama de Atividades Cenário 02 – Preencher Modelo

6.4.9.3 Cenário 03 – Incluir Artefato no Modelo

Descrição do Cenário	Cenário responsável por incluir um artefato no modelo.
Requisitos Associados	R09 - Manter Modelo
Pré Condições	O usuário deverá estar logado no sistema com permissão de Preenchedor
Pós Condições	O artefato incluído no modelo.
Atores	Preenchedor
Fluxo Principal	
Ações Recebidas	
3. Retorno execução. 4. Preenche os parâmetros.	
Ações Realizadas	
1. Recebe instância do artefato a ser inserido no modelo. 2. Solicita preenchimento dos parâmetros de inclusão do artefato no modelo. 5. Valida os parâmetros informados.	

7. Retorno execução.	6. Insere artefato dentro do modelo.
Fluxo Alternativo 01 - Parâmetros de inclusão do artefato no modelo incorretos.	
Ações Recebidas	Ações Realizadas
5.1. Parâmetros de inclusão do artefato no modelo incorretos.	5.2. Armazena erro de parâmetros inválidos, informando os nomes dos parâmetros inválidos.
5.3. Retorno fluxo principal 7.	

Tabela 100 – CDU09 – Detalhamento Cenário 03

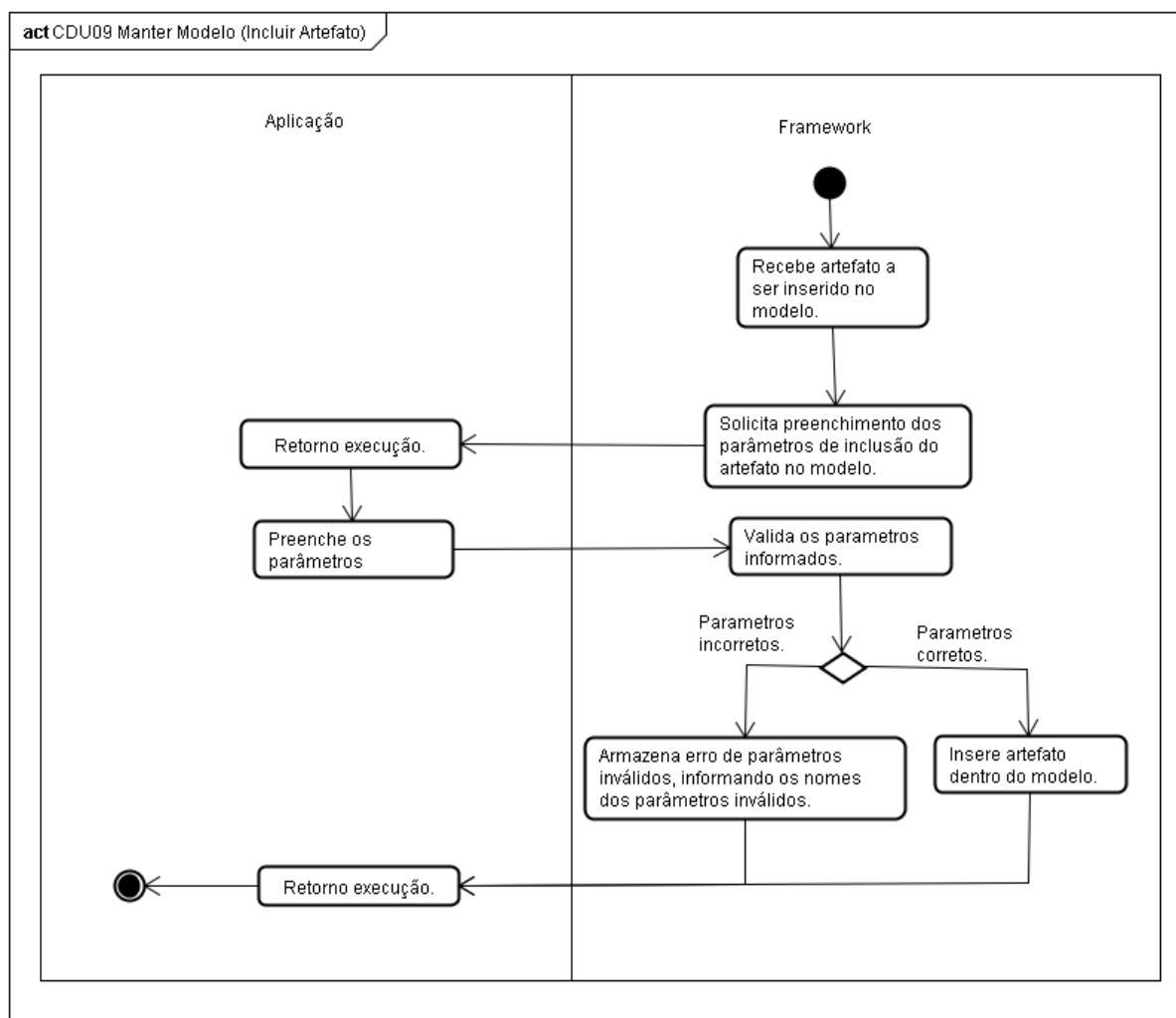


Figura 52 – Diagrama de Atividades Cenário 03 – Incluir Artefato no Modelo

6.4.9.4 Cenário 04 – Remover Artefato do Modelo

Descrição do Cenário	Cenário responsável por remover um artefato no modelo.
Requisitos Associados	R09 - Manter Modelo

Pré Condições	O usuário deverá estar logado no sistema com permissão de Preenchedor
Pós Condições	O artefato removido no modelo.
Atores	Preenchedor
Fluxo Principal	
Ações Recebidas	Ações Realizadas
	<ol style="list-style-type: none"> 1. Recebe artefato a ser removido do modelo. 2. Remove artefato do modelo. 3. Reorganiza grau e ordem de preenchimento dos artefatos do modelo. 4. Armazena sucesso de execução.

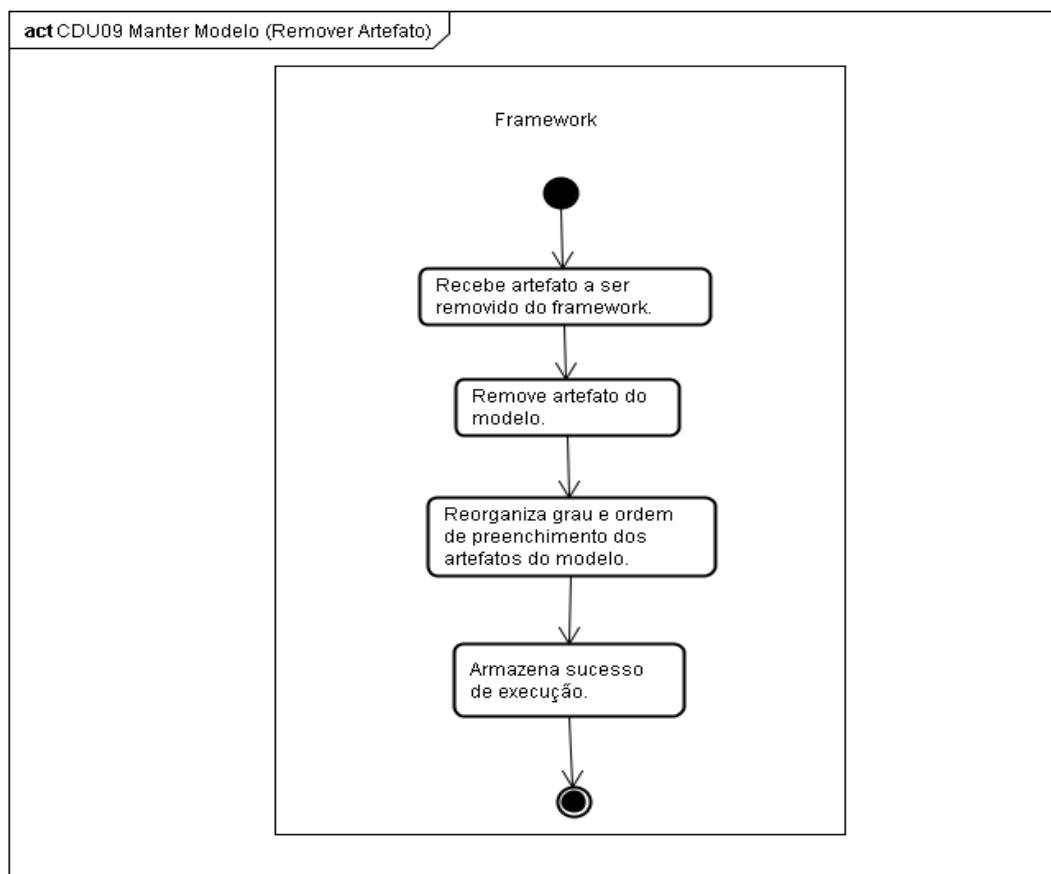
Tabela 101 – CDU09 – Detalhamento Cenário 04

Figura 53 – Diagrama de Atividades Cenário 04 – Remover Artefato do Modelo

6.4.9.5 Cenário 05 – Alterar Parâmetros do Artefato no Modelo.

Descrição do Cenário	Cenário responsável por alterar os parâmetros do artefato no modelo.
Requisitos Associados	R09 - Manter Modelo
Pré Condições	O usuário deverá estar logado no sistema com permissão de Preenchedor
Pós Condições	Os parâmetros do artefato alterados no modelo.
Atores	Preenchedor

Fluxo Principal	
Ações Recebidas	Ações Realizadas
	<ol style="list-style-type: none"> 1. Recebe item do modelo para ser alterado seus parâmetros.. 2. Valida parâmetros informados. 3. Altera os parâmetros do item do modelo no modelo. 4. Armazena sucesso de execução. 5. Retorno execução.
Fluxo Alternativo 01 - Parâmetros de alteração inválidos.	
Ações Recebidas	Ações Realizadas
2.1. Parâmetros de alteração inválidos. 2.3. Retorno fluxo principal 5.	2.2. Armazena erro de parâmetros inválidos, informando os parâmetros incorretos.

Tabela 102 – CDU09 – Detalhamento Cenário 05

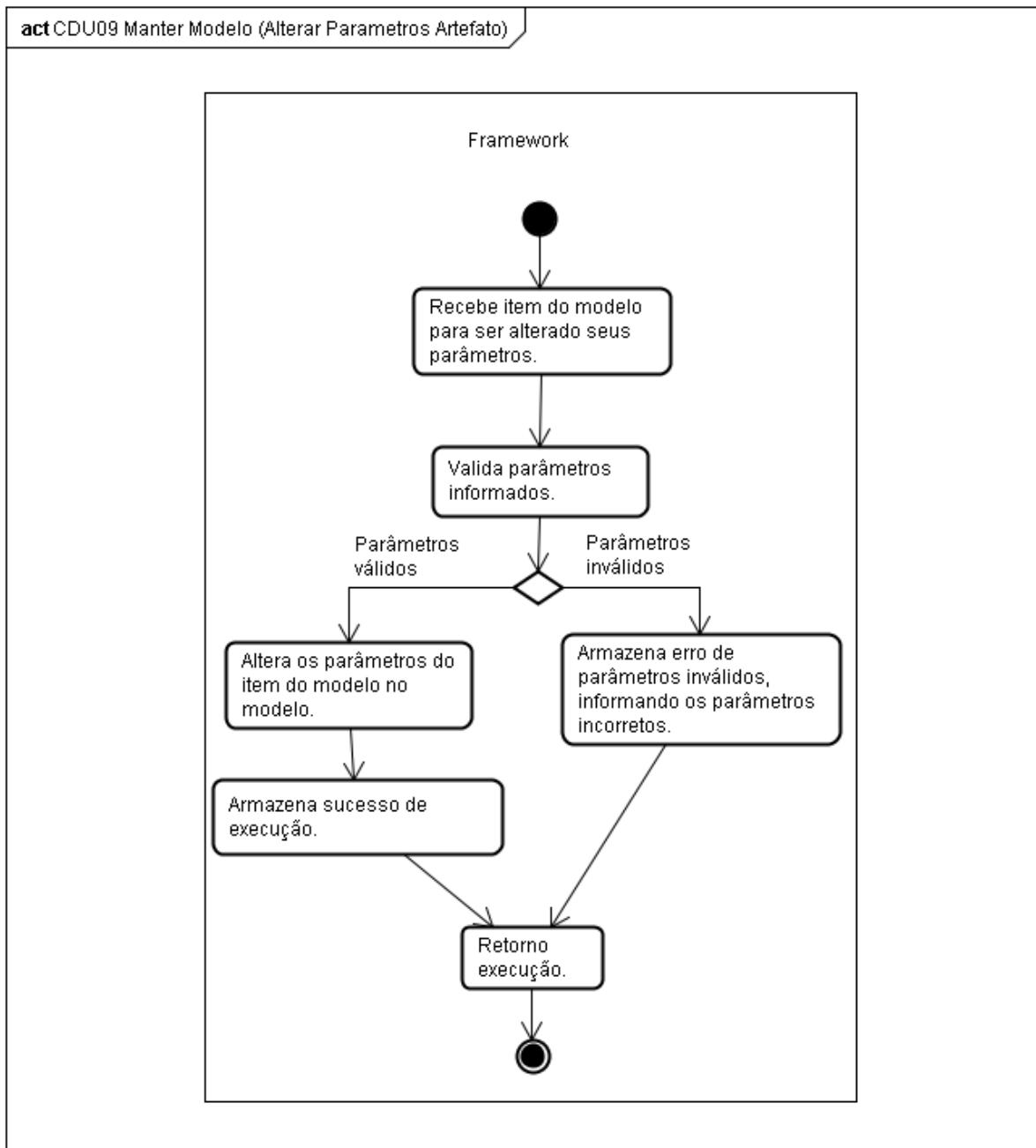


Figura 54 – Diagrama de Atividades Cenário 05 – Alterar parâmetros do artefato no modelo

6.4.9.6 Cenário 06 – Mapear Modelo

Descrição do Cenário	Cenário responsável por mapear um modelo para o framework.
Requisitos Associados	R09 - Manter Modelo
Pré Condições	O usuário deverá estar logado no sistema com permissão de Preenchedor
Pós Condições	O modelo mapeado no framework.
Atores	Preenchedor
Fluxo Principal	
Ações Recebidas	Ações Realizadas
	1. Recebe modelo a ser mapeado.

	<ol style="list-style-type: none"> 2. Verifica se os artefatos dependentes foram inseridos no modelo. 3. Realiza cadastro do modelo. 4. Armazena sucesso de execução. 5. Retorno de execução.
Fluxo Alternativo 01 - Artefatos dependentes não foram inseridos no modelo.	
Ações Recebidas	Ações Realizadas
2.1. Artefatos dependentes não foram inseridos no modelo. 2.3. Retorno fluxo principal 5.	2.2. Armazena erro de dependência de artefato, informando o nomes dos artefatos dependentes que não foram inseridos no modelo.

Tabela 103 – CDU09 – Detalhamento Cenário 06

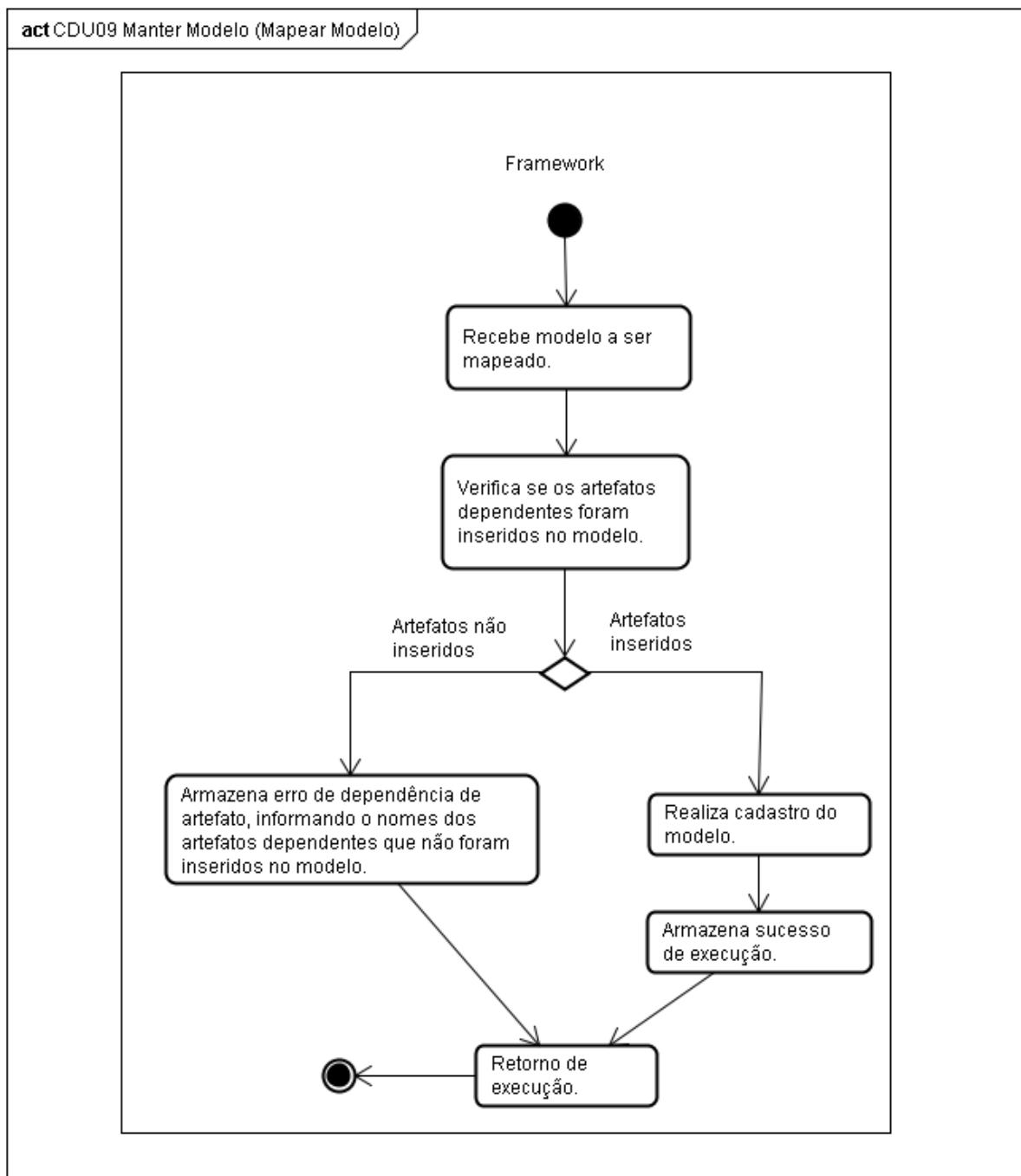


Figura 55 – Diagrama de Atividades Cenário 06 – Mapear Modelo

6.4.9.7 Cenário 07 – Alterar Modelo

Descrição do Cenário	Cenário responsável por realizar a alteração de um modelo.
Requisitos Associados	R09 - Manter Modelo (Ata Reunião 5)
Pré Condições	O usuário deverá estar logado no sistema com permissão de Preenchedor
Pós Condições	O modelo alterado no framework.
Atores	Preenchedor
Fluxo Principal	

Ações Recebidas	Ações Realizadas
1. Solicita alteração de modelo. 5. Retorno execução. 6. Executa cenário Preencher modelo. 7. Solicita alteração do modelo. 9. Retorno execução.	2. Verifica se o modelo está sendo utilizado por algum projeto. 3. Carrega modelo a ser alterado. 4. Armazena sucesso de execução. 8. Executa cenário Alterar Modelo no Framework.
Fluxo Alternativo 01 - Modelo sendo utilizado por algum projeto.	
Ações Recebidas	Ações Realizadas
2.1. Modelo sendo utilizado por algum projeto. 2.3. Retorno fluxo principal 9.	2.2. Armazena erro de modelo sendo utilizado projeto, informando os nomes dos projetos em que o modelo esta sendo utilizado.

Tabela 104 – CDU09 – Detalhamento Cenário 07

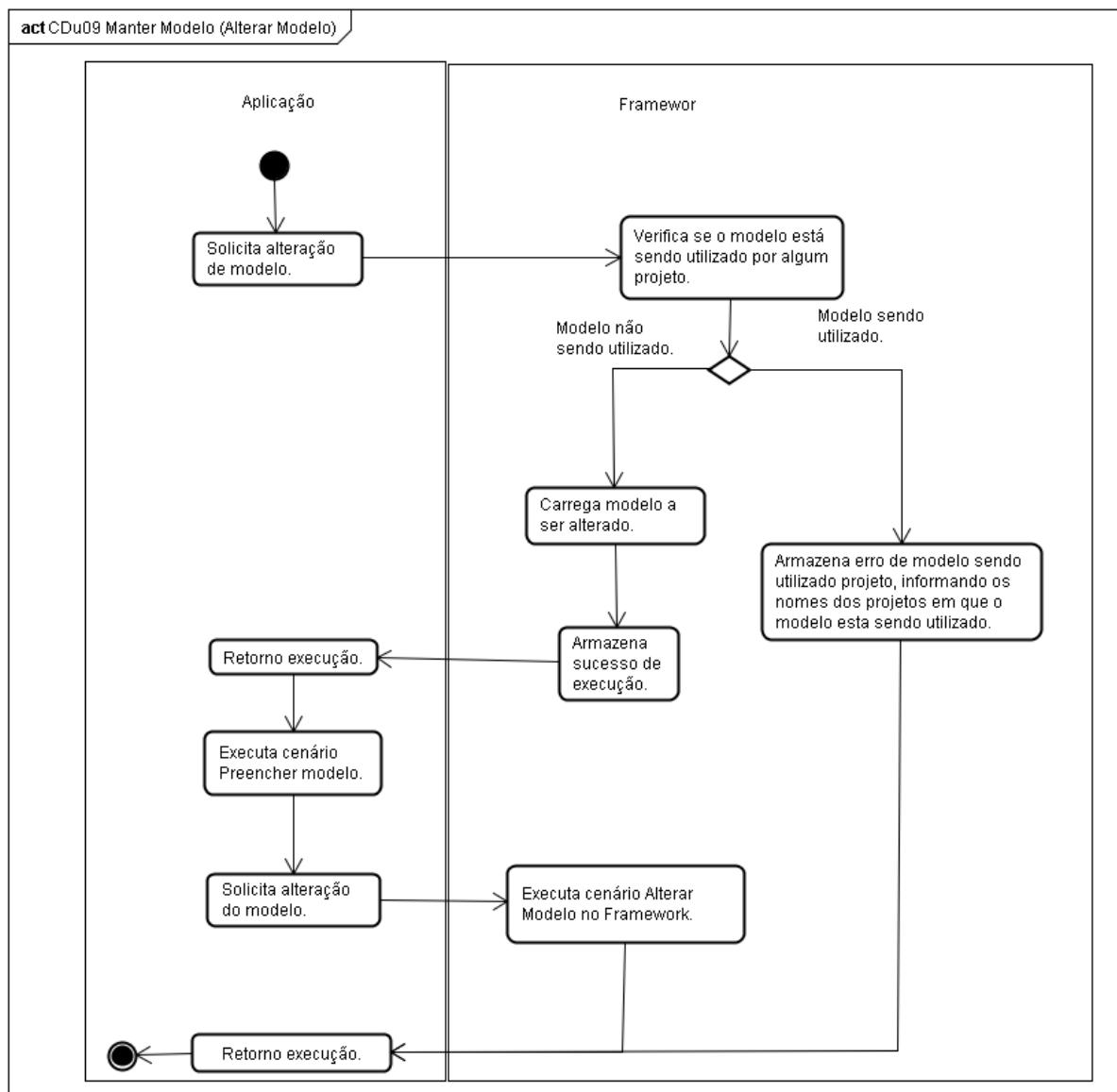


Figura 56 – Diagrama de Atividades Cenário 07 – Alterar Modelo

6.4.9.8 Cenário 08 – Alterar Modelo no Framework

Descrição do Cenário	Cenário responsável por realizar a alteração de um modelo no framework.
Requisitos Associados	R09 - Manter Modelo
Pré Condições	O usuário deverá estar logado no sistema com permissão de Preenchedor
Pós Condições	O modelo alterado no framework.
Atores	Preenchedor
Fluxo Principal	
Ações Recebidas	Ações Realizadas
	<ol style="list-style-type: none"> 1. Recebe modelo a ser alterado. 2. Verifica se os artefatos dependentes foram inseridos no modelo. 3. Realiza alteração do modelo. 4. Armazena sucesso de execução.

	5. Retorno de execução.
Fluxo Alternativo 01 - Artefatos dependentes não inseridos no modelo.	
Ações Recebidas	Ações Realizadas
2.1. Artefatos dependentes não inseridos no modelo. 2.3. Retorno fluxo principal 5.	2.2. Armazena erro de dependência de artefato, informando o nomes dos artefatos dependentes que não foram inseridos no modelo.

Tabela 105 – CDU09 – Detalhamento Cenário 08

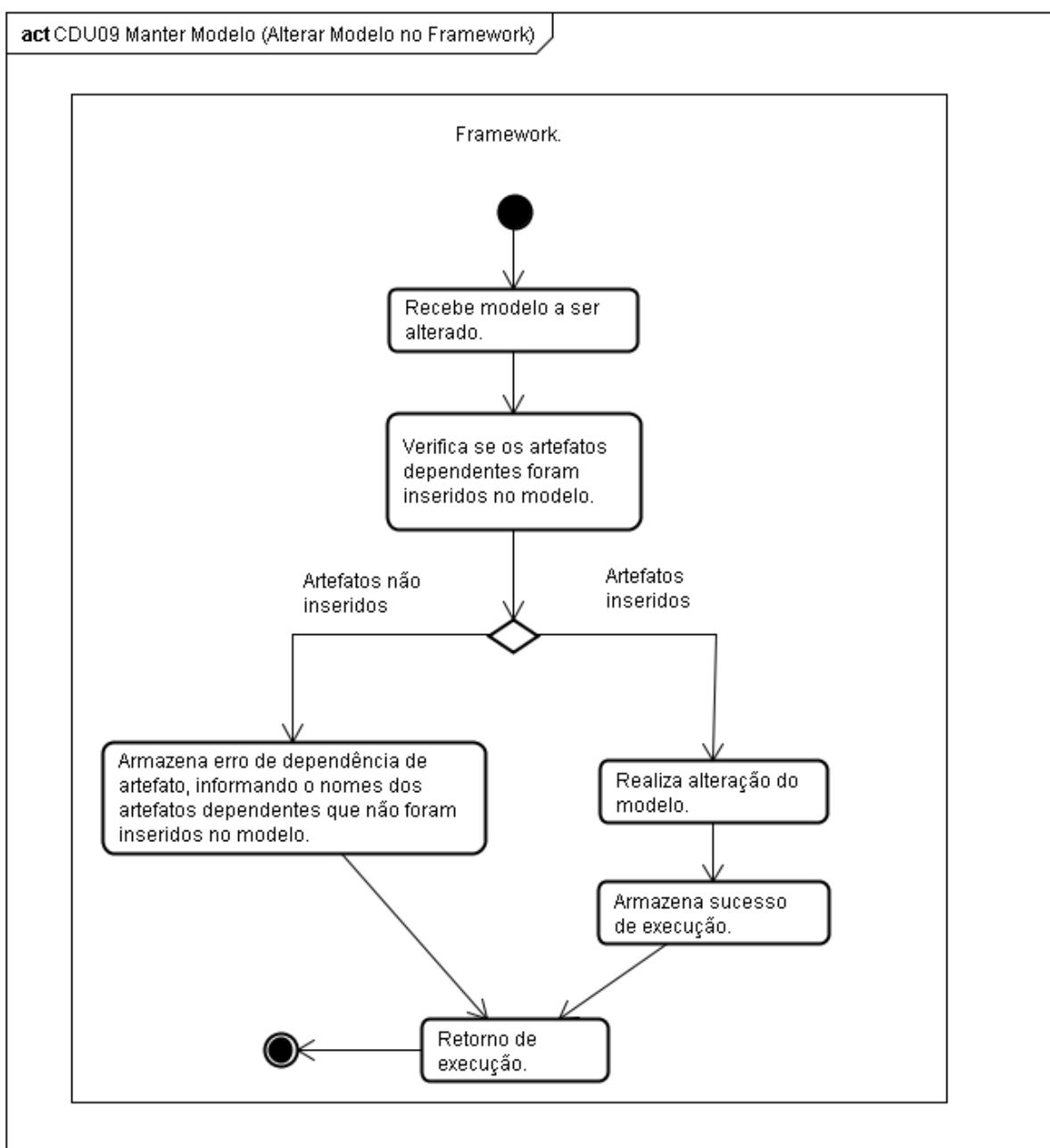


Figura 57 – Diagrama de Atividades Cenário 08 – Alterar Artefato no Framework

6.4.9.9 Cenário 09 – Remover Modelo

Descrição do Cenário	Cenário responsável por remover um modelo do framework.
Requisitos Associados	R09 - Manter Modelo
Pré Condições	O usuário deverá estar logado no sistema com permissão de Preenchedor
Pós Condições	O modelo removido do framework.
Atores	Preenchedor
Fluxo Principal	
Ações Recebidas	
1. Solicita remoção de modelo.	2. Verifica se o modelo está sendo utilizado por algum projeto. 3. Remove modelo. 4. Armazena sucesso de execução. 5. Retorno execução.
Fluxo Alternativo 01 - Modelo sendo utilizado em um projeto	
Ações Recebidas	
2.1. Modelo sendo utilizado em um projeto.	2.2. Armazena erro de exclusão, informando o nome dos projetos em que o modelo esta sendo utilizado. 2.3. Retorno fluxo principal 5.

Tabela 106 – CDU09 – Detalhamento Cenário 09

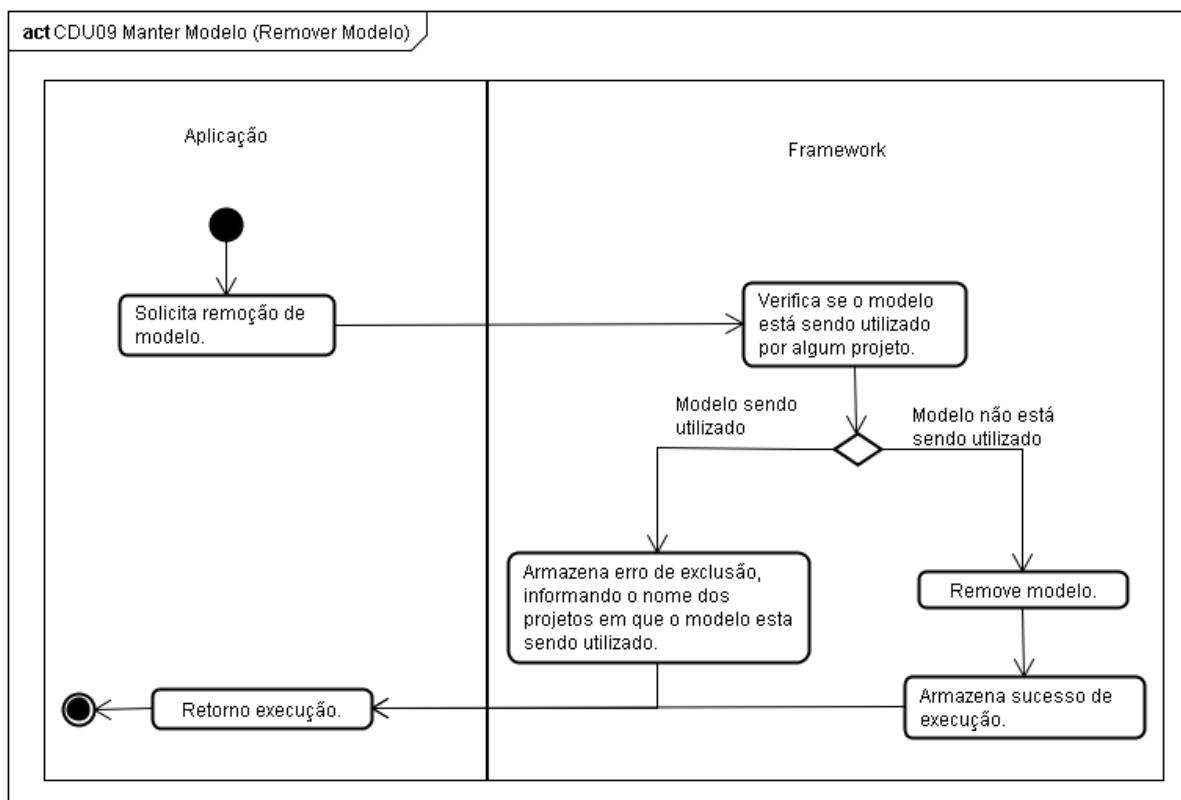


Figura 58 – Diagrama de Atividades Cenário 09 – Remover Modelo

6.4.9.10 Cenário 10 – Clonar Modelo

Descrição do Cenário	Cenário responsável por clonar um modelo.
Requisitos Associados	R09 - Manter Modelo
Pré Condições	O usuário deverá estar logado no sistema com permissão de Preenchedor
Pós Condições	O modelo clonado
Atores	Preenchedor
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1. Solicita clonagem de modelo, informando o novo nome e a nova descrição. 8. Retorno execução.	2. Verifica duplicidade de nome de modelo. 3. Instância um novo modelo. 4. Seta o nome e a descrição do modelo. 5. Copia os itens de um modelo para o outro. 6. Realiza o cadastro do modelo. 7. Armazena sucesso de execução.
Fluxo Alternativo 01 - Nome do modelo cadastrado no framework.	
Ações Recebidas	Ações Realizadas
2.1. Nome do modelo cadastrado no framework. 2.3. Retorno fluxo principal 8.	2.2. Armazena erro de nome de modelo já cadastrado no framework.

Tabela 107 – CDU09 – Detalhamento Cenário 10

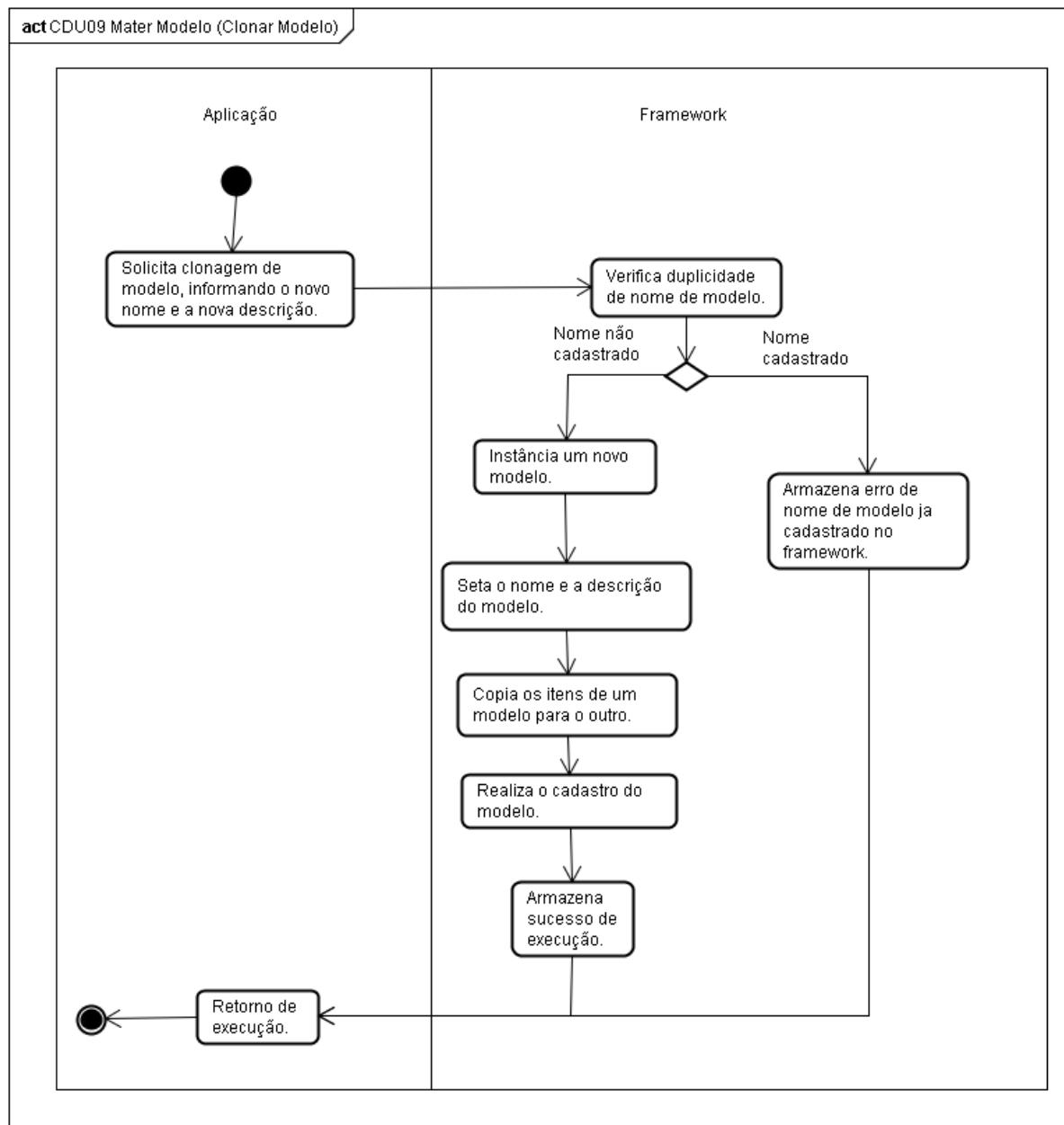


Figura 59 – Diagrama de Atividades Cenário 10 – Clonar Modelo

6.4.10 Caso de Uso 10 – Manter Projeto

6.4.10.1 Cenário 01 – Mapear Projeto

Descrição do Cenário	Cenário responsável por realizar o cadastro de um novo projeto
Requisitos Associados	R10 - Manter Projeto
Pré Condições	O usuário deverá estar logado no sistema com permissão de gerente de projeto.
Pós Condições	O projeto mapeado no framework.
Atores	Gerente de Projeto
Fluxo Principal	

Ações Recebidas	Ações Realizadas
33. Solicita cadastro de projeto, informando o nome do projeto, o modelo e o grupo de usuários que ele pertence.	
37. Retorno da execução.	34. Valida duplicidade de nome de cadastro. 35. Realiza o cadastro do projeto. 36. Armazena sucesso de execução.
Fluxo Alternativo 01 - Nome do projeto já cadastrado no framework	
Ações Recebidas	Ações Realizadas
5.16. Nome do projeto já cadastrado no framework.	
5.18. Retorno fluxo principal 5.	5.17. Armazena erro de nome de projeto cadastrado no framework.

Tabela 108 – CDU10 – Detalhamento Cenário 01

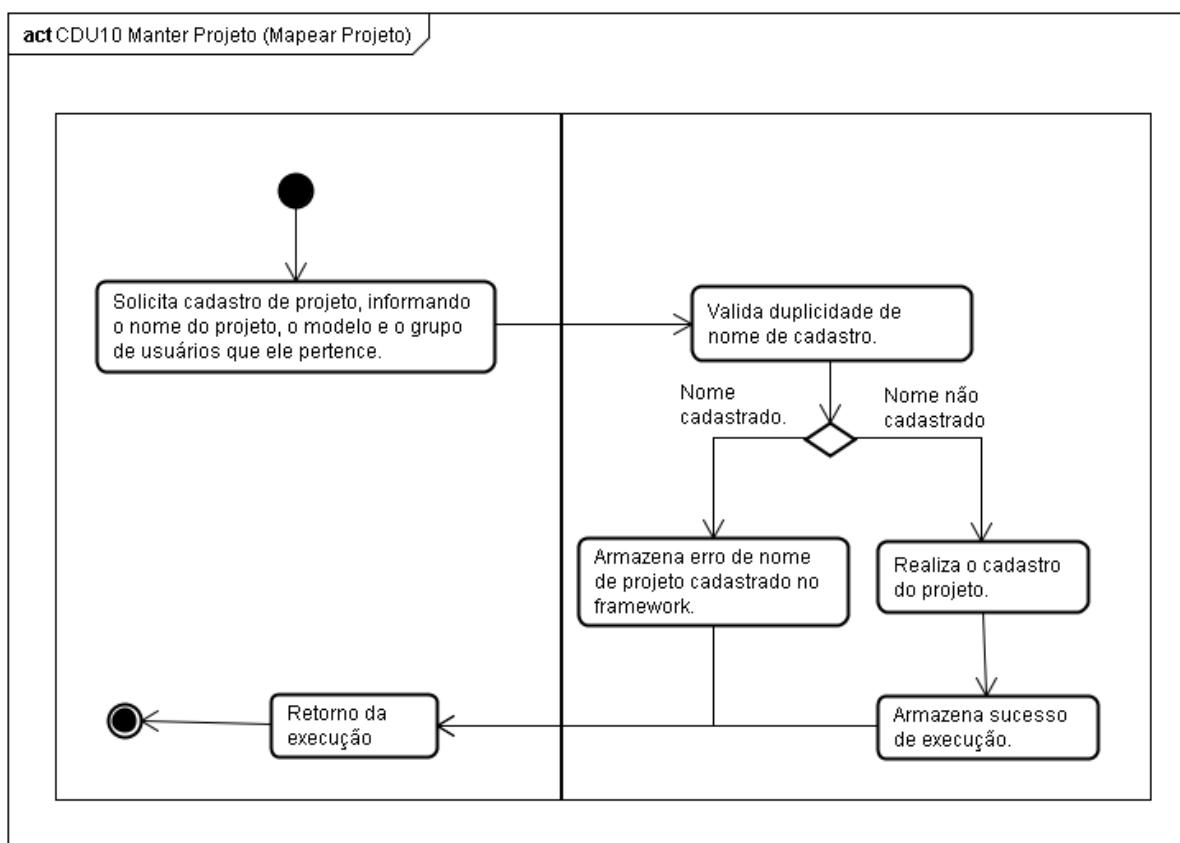
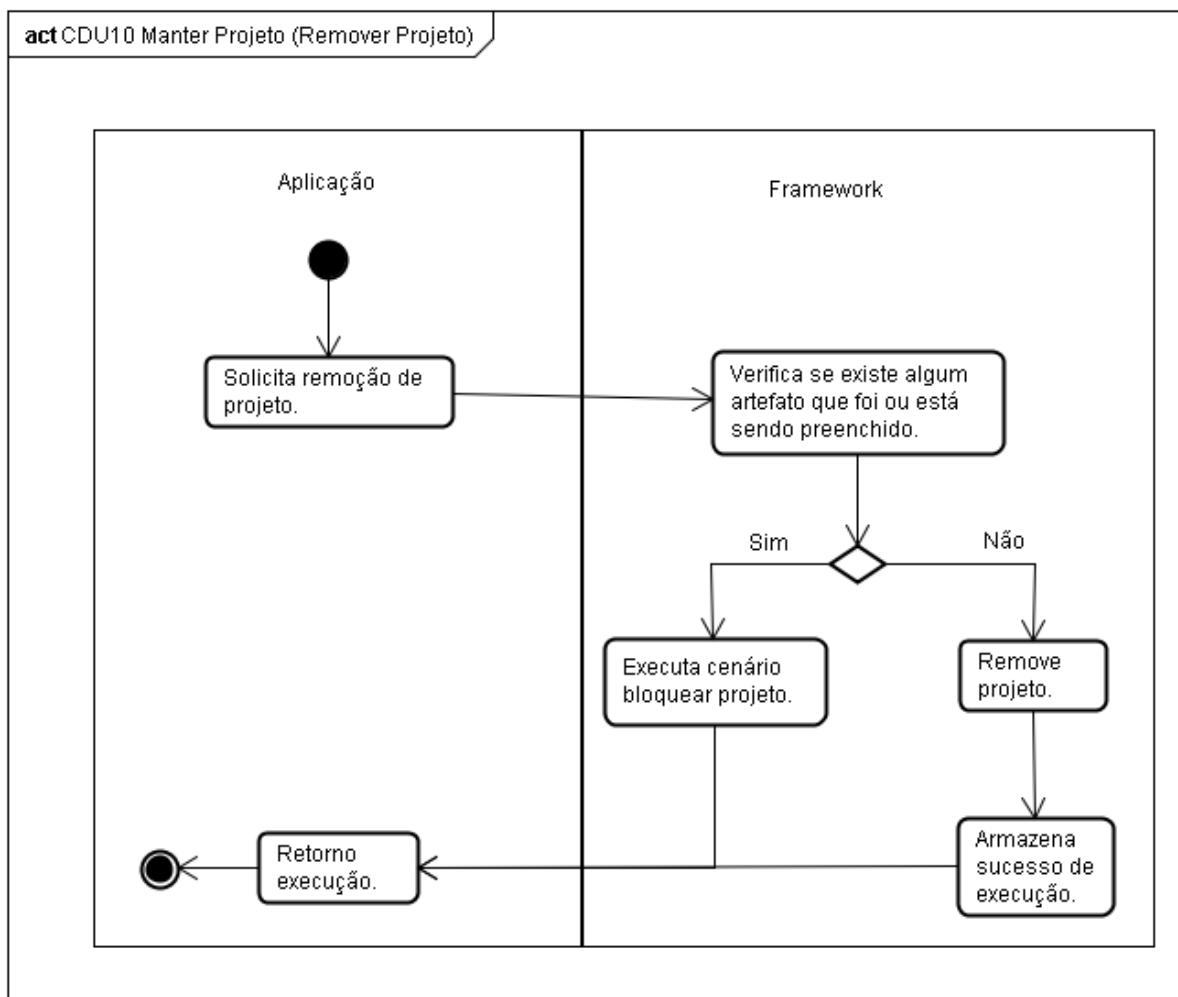


Figura 60 – Diagrama de Atividades Cenário 01 – Mapear Projeto

6.4.10.2 Cenário 02 – Remover Projeto

Descrição do Cenário	Cenário responsável por remover um projeto do framework.
Requisitos Associados	R10 - Manter Projeto
Pré Condições	O usuário deverá estar logado no sistema com permissão de gerente de projeto

Pós Condições	O projeto removido do framework.
Atores	Gerente de Projeto
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1. Solicita remoção de projeto. 5. Retorno da execução.	2. Verifica se existe algum artefato que foi ou está sendo preenchido. 3. Remove projeto. 4. Armazena sucesso de execução.
Fluxo Alternativo 01 - Existe artefato que foi ou está sendo preenchido	
Ações Recebidas	Ações Realizadas
2.1. Existe artefato que foi ou está sendo preenchido. 2.3. Retorno fluxo principal 5.	2.2. Executa cenário bloquear projeto.

Tabela 109 – CDU10 – Detalhamento Cenário 02**Figura 61 – Diagrama de Atividades Cenário 02 – Remover Projeto**

6.4.10.3 Cenário 03 – Bloquear Projeto

Descrição do Cenário	Cenário responsável por bloquear um projeto no framework.
Requisitos Associados	R10 - Manter Projeto
Pré Condições	O usuário deverá estar logado no sistema com permissão de gerente de projeto
Pós Condições	O projeto bloqueado no framework.
Atores	Gerente de projeto.
Fluxo Principal	
Ações Recebidas	Ações Realizadas
3. Retorno execução. 4. Confirma bloqueio.	1. Recebe projeto a ser bloqueado. 2. Armazena solicitação de confirmação de bloqueio de projeto. 5. Desbloqueia artefatos que estão sendo preenchidos por usuários, armazenando notificação de bloqueio de projeto. 6. Bloqueia projeto no framework, não permitindo que ninguém o preencha. 7. Armazena sucesso de execução. 8. Retorno execução.
Fluxo Alternativo 01 - Cancelamento de bloqueio de projeto	
Ações Recebidas	Ações Realizadas
4.1. Cancelamento de bloqueio de projeto 4.3. Retorno fluxo principal 8.	4.2. Armazena cancelamento de bloqueio do projeto.

Tabela 110 – CDU10 – Detalhamento Cenário 03

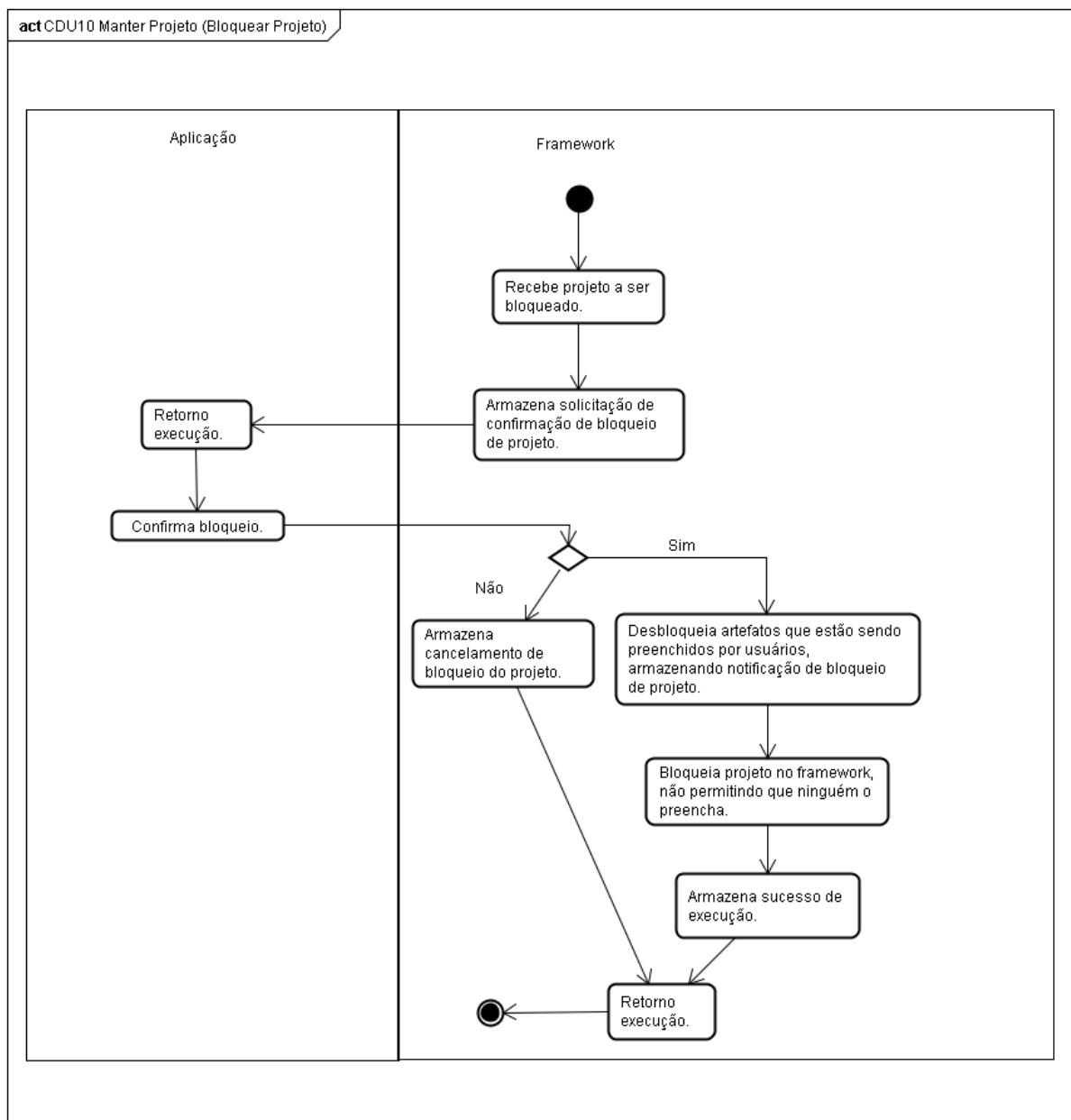


Figura 62 – Diagrama de Atividades Cenário 03 – Bloquear Projeto

6.4.10.4 Cenário 04 – Preencher Projeto

Descrição do Cenário	Cenário responsável por selecionar artefato que será preenchido pelo usuário.
Requisitos Associados	R10 - Manter Projeto
Pré Condições	O usuário deverá estar logado no sistema com permissão de Preenchedor
Pós Condições	Artefato a ser preenchido pelo usuário
Atores	Preenchedor
Fluxo Principal	
Ações Recebidas	
1. Solicita projetos que o usuário logado participa.	Ações Realizadas

	<p>2. Verifica se usuário esta ativo.</p> <p>3. Armazena lista de projetos que o usuário participa.</p> <p>4. Armazena sucesso de execução.</p> <p>5. Retorno execução.</p> <p>6. Solicita lista de artefatos do modelo do projeto selecionado.</p> <p>10. Retorno execução.</p> <p>11. Seleciona artefato a ser preenchido.</p> <p>13. Retorno execução.</p>
Fluxo Alternativo 01 - Usuário inativo no framework	
Ações Recebidas	Ações Realizadas
2.1. Usuário inativo no framework. 2.3. Retorno fluxo principal 13.	2.2. Armazena erro de usuário inativado no framework.

Tabela 111 – CDU10 – Detalhamento Cenário 04

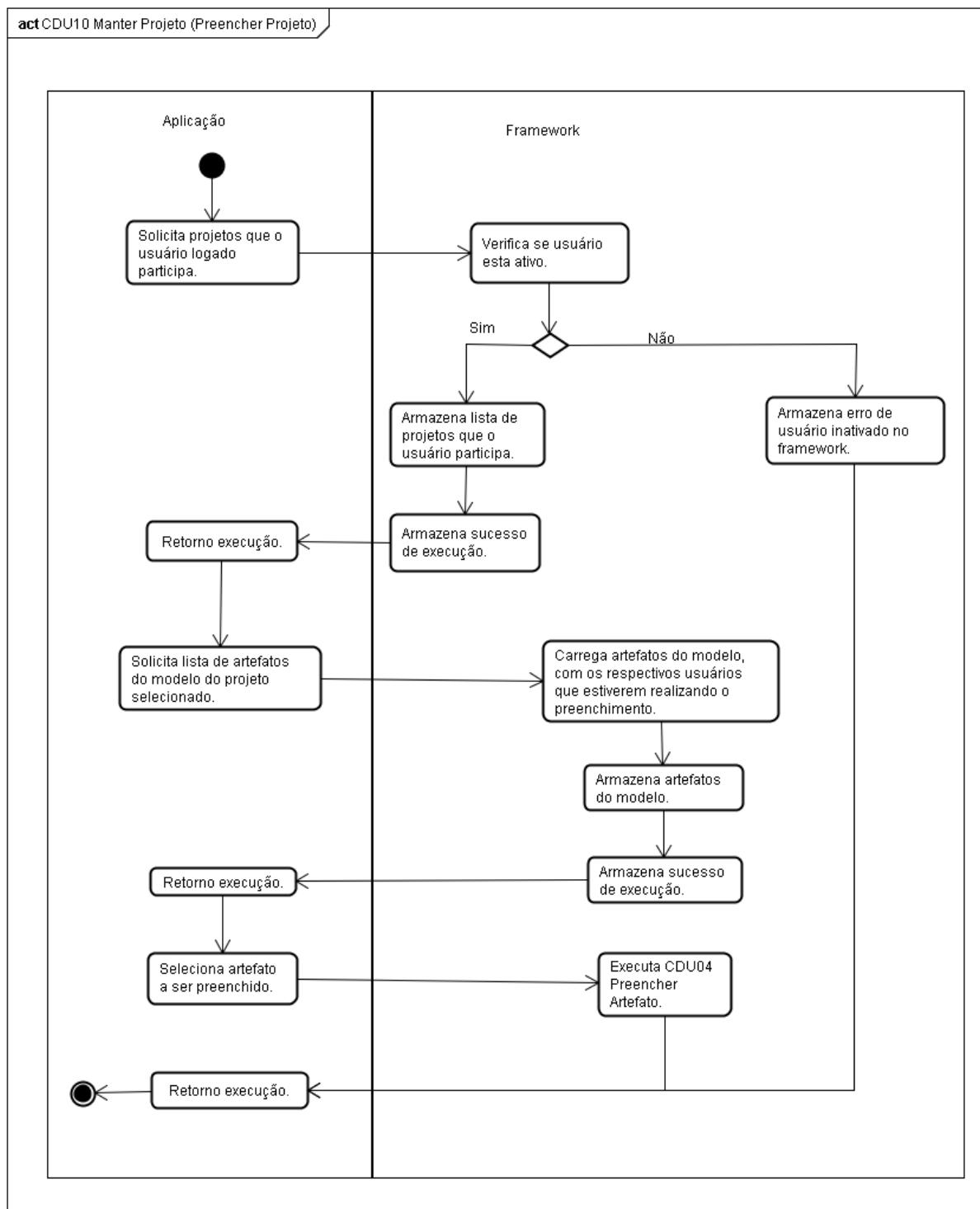


Figura 63 – Diagrama de Atividades Cenário 04 – Preencher Projeto

6.4.11 Caso de Uso 11 - Manter Serviço

6.4.11.1 Cenário 01 - Mapear Serviço

Descrição do Cenário	Cenário responsável por mapear serviços para o framework.
----------------------	---

Requisitos Associados	R11 Manter Serviço
Pré Condições	O usuário deverá estar logado no sistema com permissão de desenvolvedor.
Pós Condições	O serviço(s) estar mapeados no framework.
Atores	Desenvolvedor
Fluxo Principal	
Ações Recebidas	
19. Lista de serviços para ser mapeada.	20. Verifica se o formato dos arquivos estão corretos. 21. Executa cenário CDU13 Realizar Versionamento (Versionar) 22. Armazena sucesso de execução. 23. Lista de arquivos vazia.
24. Retorno da execução.	
Fluxo Alternativo 1 - Formato incorreto de arquivos.	
Ações Recebidas	
3.1. Arquivos com formato incorreto.	3.2. Armazena o nome dos arquivos incorretos. 3.3. Armazena erro de formato incorreto de arquivo.
3.4. Retorno fluxo principal 5.	
Fluxo Exceção 2 - Versionamento por nova versão	
Ações Recebidas	
3.1. Arquivo de serviço para ser cadastrado no framework.	3.2. Armazenamento do arquivo no diretório específico. 3.3. Mapeamento do serviço no framework.
3.4. Retorno fluxo principal 4.	
Fluxo Exceção 3 - Versionamento por revisão	
Ações Recebidas	
3.1. Arquivo de serviço para ser revisionado no framework.	3.2. Transferência do arquivo antigo para o diretório de versionamento. 3.3. Transferência do arquivo referente ao mapeamento ao diretório específico.
3.4. Retorno fluxo principal 4.	
Fluxo Exceção 4 - Mesma versão de arquivo	
Ações Recebidas	
3.5. Mesmo arquivo informado.	3.6. Armazena mensagem de arquivo já versionado.
3.7. Retorno fluxo principal 4.	
Fluxo Exceção 4 - Lista de arquivos não vazia.	
Ações Recebidas	
2.1. Lista de arquivos não vazia.	2.2. Retorno fluxo principal 3.

Tabela 112 – CDU11 – Detalhamento Cenário 01

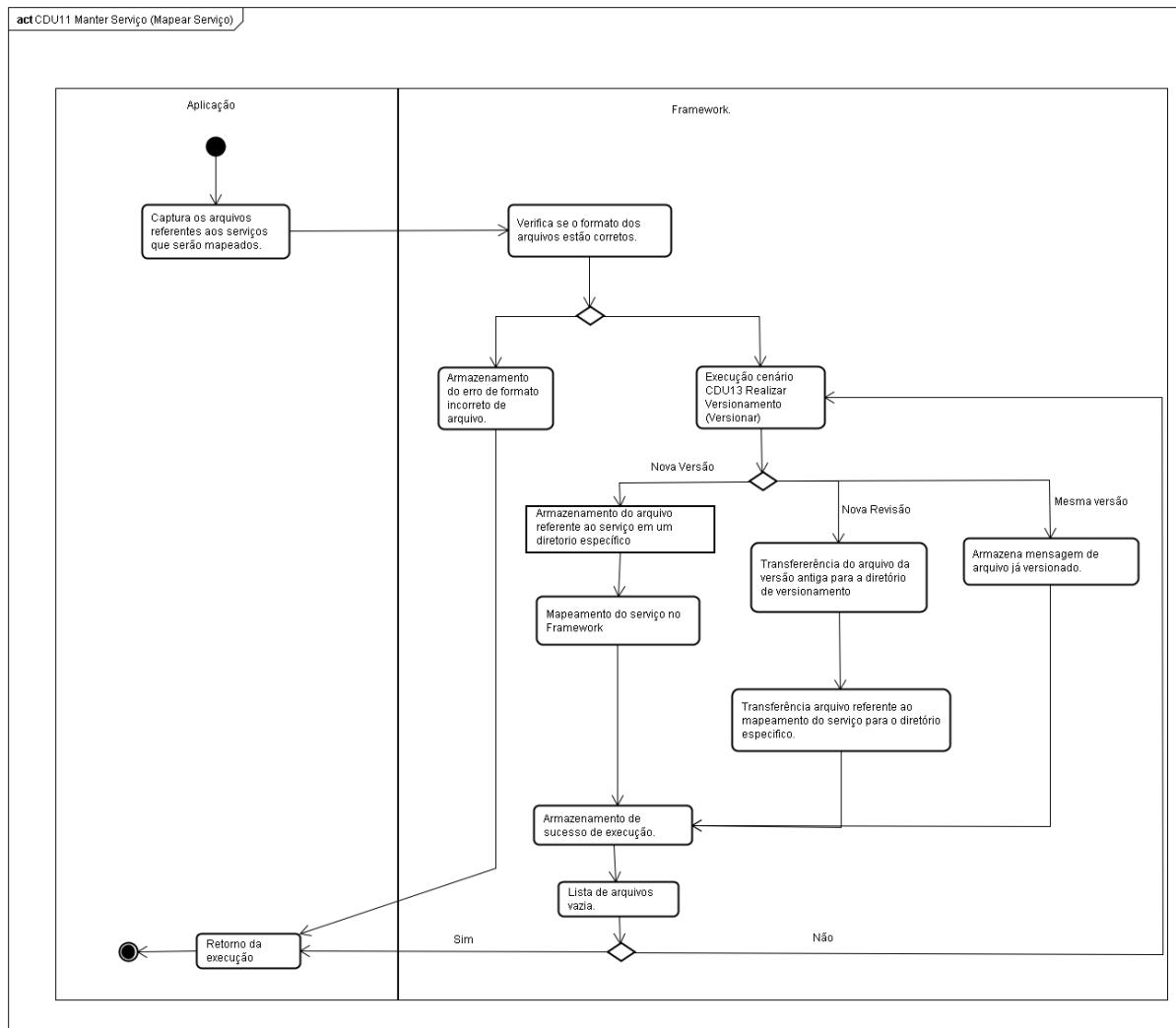


Figura 64 – Diagrama de Atividades Cenário 01 – Mapear Serviço.

6.4.11.2 Cenário 02 - Pesquisar Serviço

Descrição do Cenário	Cenário responsável por pesquisar os serviços cadastrados no framework.
Requisitos Associados	R11 Manter Serviço
Pré Condições	O usuário deverá estar logado no sistema com permissão de desenvolvedor.
Pós Condições	Lista de serviços cadastrados no framework.
Atores	Desenvolvedor / Montador
Fluxo Principal	
Ações Recebidas	
Ações Realizadas	
1. Solicita lista de serviços. 4. Retorna lista de serviços.	2. Pesquisa serviços cadastrados. 3. Realiza instância dos serviços cadastrados.

Tabela 113 – CDU11 – Detalhamento Cenário 02

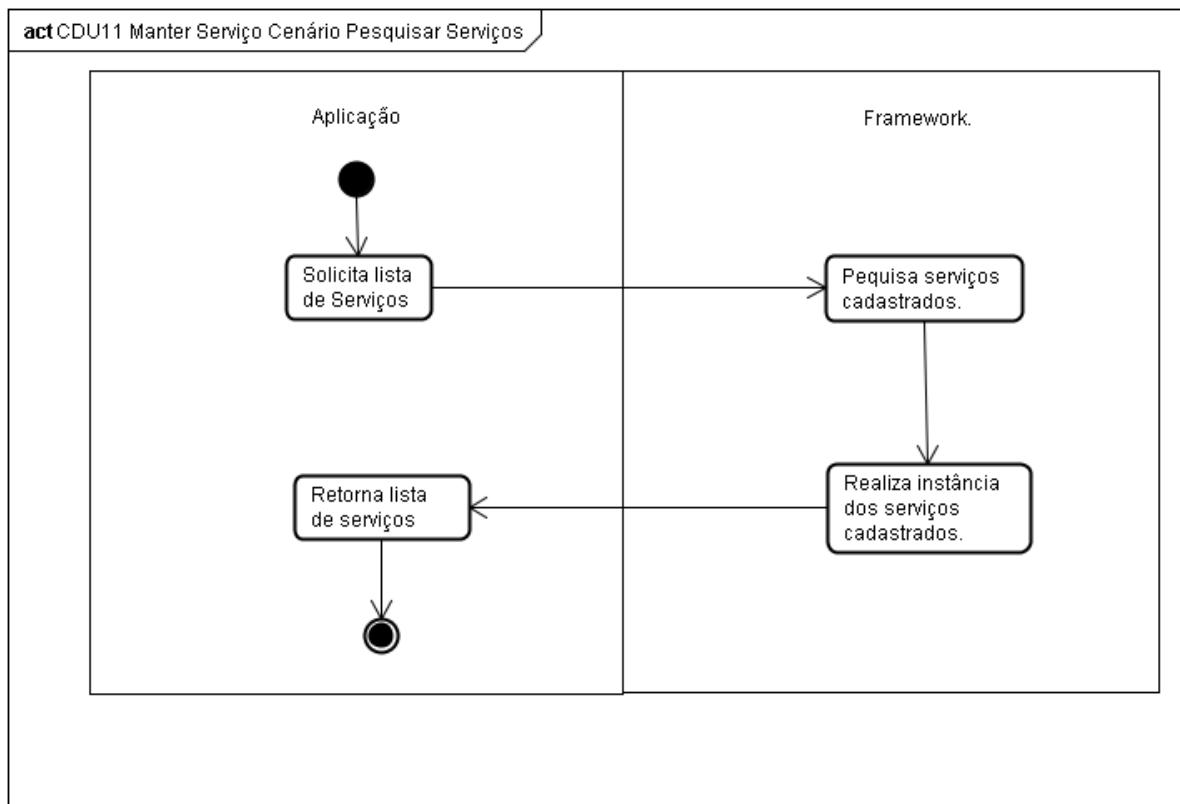


Figura 65 – Diagrama de Atividades Cenário 02 – Pesquisar Serviço.

6.4.11.3 Cenário 03 - Remover Serviço

Descrição do Cenário	Cenário responsável por remover um serviço do framework.
Requisitos Associados	R11 Manter Serviço
Pré Condições	O usuário deverá estar logado no sistema com permissão de desenvolvedor.
Pós Condições	O serviço(s) estar removido do framework.
Atores	Desenvolvedor
Fluxo Principal	
Ações Recebidas	
1. Solicita remoção de serviço.	2. Realiza a rastreabilidade do serviço no framework. 3. Descadastra o serviço do framework. 4. Transfere o arquivo do serviço para o diretório de versionamento. 5. Armazena sucesso de execução.
6. Retorno da execução.	
Fluxo Alternativo 1 - Serviço sendo utilizado no framework.	
Ações Recebidas	
2.3. Serviço está sendo utilizado no framework.	2.4. Armazena os nomes dos artefatos onde o serviço estava sendo utilizado. 2.5. Armazena erro de falha de remoção.

2.6. Retorno fluxo principal 6.

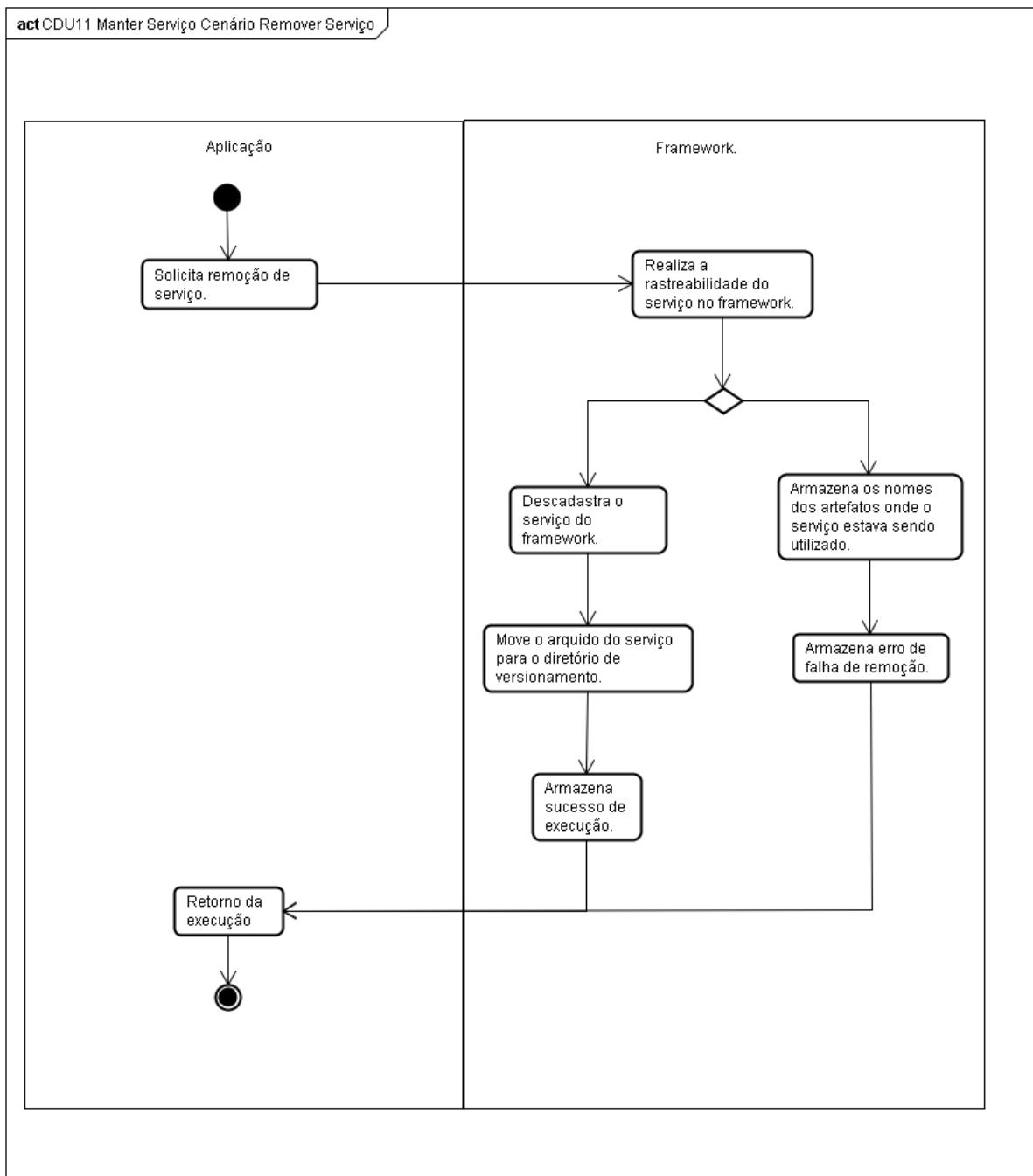
Tabela 114 – CDU11 – Detalhamento Cenário 03

Figura 66 – Diagrama de Atividades Cenário 03 – Remover Serviço.

6.4.11.4 Cenário 04 - Verificação dos Serviços Cadastrados

Descrição do Cenário	Cenário responsável por verificar o cadastro dos serviços do framework.
Requisitos Associados	R11 Manter Serviço

Pré Condições	O usuário deverá estar logado no sistema com permissão de desenvolvedor.
Pós Condições	Verificação dos serviços cadastrados no framework.
Atores	Desenvolvedor
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1. Solicita instância do framework.	2. Pesquisa lista de serviços cadastrados no framework. 3. Verifica se o arquivo se encontra no diretório especificado. 4. Lista de serviços vazia.
5. Retorno da instância do framework.	
Fluxo Alternativo 1 - Arquivo ausente e sendo utilizado no framework.	
Ações Recebidas	Ações Realizadas
3.1. Arquivo do serviço ausente no diretório e sendo utilizado pelo framework.	3.2. Inativa os modelos onde o serviço estava sendo utilizado. 3.3. Armazena erro de serviço não encontrado, armazenando o nome dos artefatos onde ele estava sendo utilizado.
3.4. Retorno fluxo principal 4.	
Fluxo Alternativo 2 - Arquivo ausente e não sendo utilizado no framework.	
Ações Recebidas	Ações Realizadas
3.1. Arquivo do serviço ausente no diretório e não sendo utilizado pelo framework.	3.2. Remove o cadastro do serviço do framework.
3.3. Retorno fluxo principal 4.	
Fluxo Alternativo 3 - Lista de serviços não vazia	
Ações Recebidas	Ações Realizadas
4.1. Lista de serviços não vazia.	4.2. Retorno fluxo principal 3.

Tabela 115 – CDU11 – Detalhamento Cenário 04

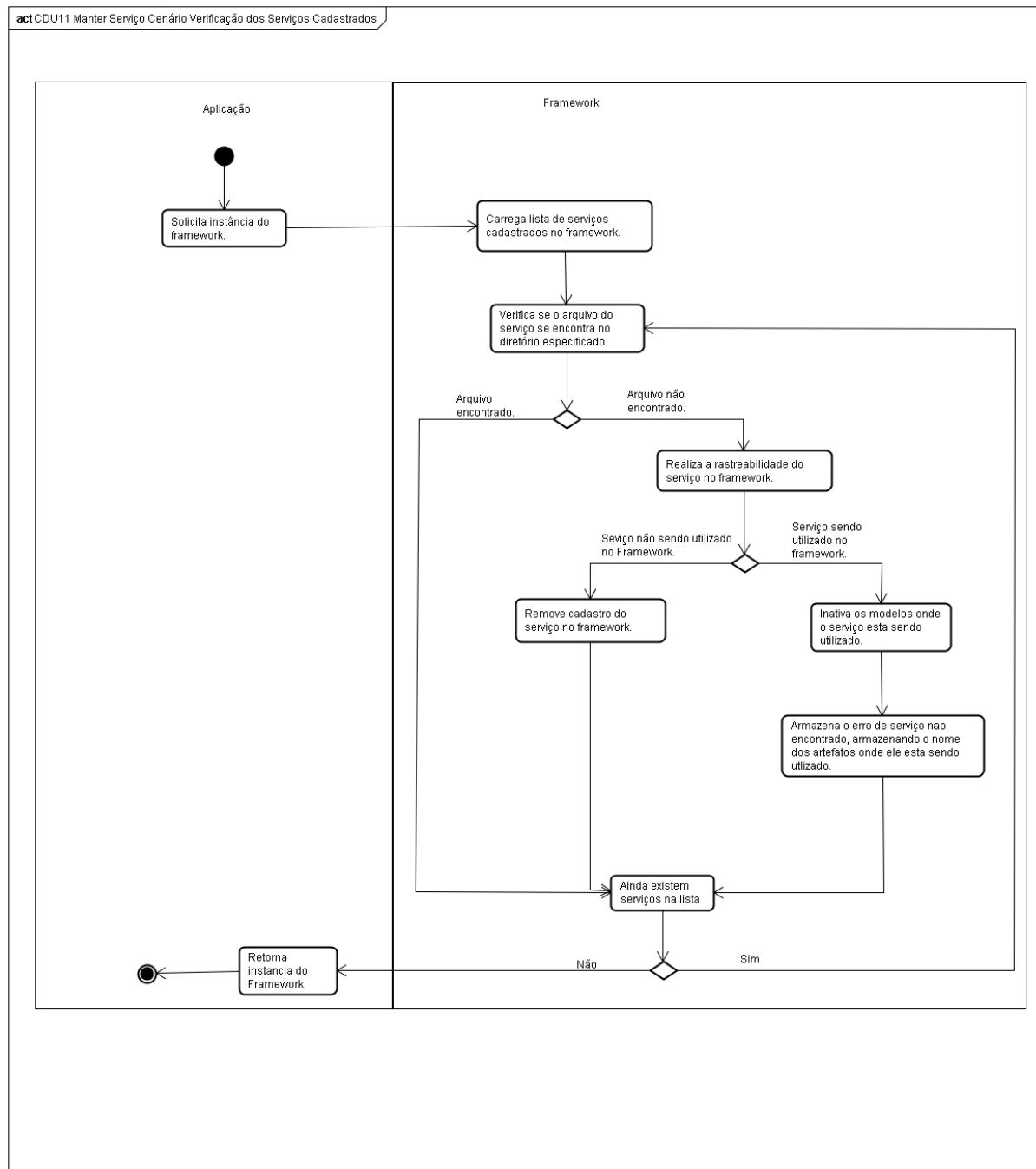


Figura 67 – Diagrama de Atividades Cenário 04 – Verificação dos Serviços Cadastrados.

6.4.12 Caso de Uso 12 – Gerenciar Serviço

6.4.12.1 Cenário 01 – Executar Serviço Geral

Descrição do Cenário	Cenário responsável realizar a execução de um serviço disparado por um TipoMembro disparador.
----------------------	---

Requisitos Associados	R06 - Preencher artefato, R12 - Gerenciar Serviço de Persistência, R13 Gerenciar Serviço de Dependência, R16 Gerenciar Serviço de Validação de dados.
Pré Condições	O usuário deverá estar logado no sistema com permissão de Preenchedor.
Pós Condições	O retorno da execução do serviço.
Atores	Preenchedor
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1. TipoMembro dispara execução de serviço.	2. Executa cenário 02 Executar Serviço Rotina Interna. 3. Verifica se o serviço é finalizador. 4. Armazena sucesso de execução. 5. Retorno execução.
Fluxo Alternativo 01 - Erro na execução do cenário Executar Serviço Rotina Interna	
Ações Recebidas	Ações Realizadas
2.1. Erro na execução do cenário Executar Serviço Rotina Interna.	2.2. Retorno fluxo principal 5.
Fluxo Alternativo 02 - Serviço finalizador	
Ações Recebidas	Ações Realizadas
3.1. Serviço executado é finalizador.	3.2. Executa cenário Montar Interface de visualização. 3.3. Retorno fluxo principal 4.

Tabela 116 – CDU12 – Detalhamento Cenário 01

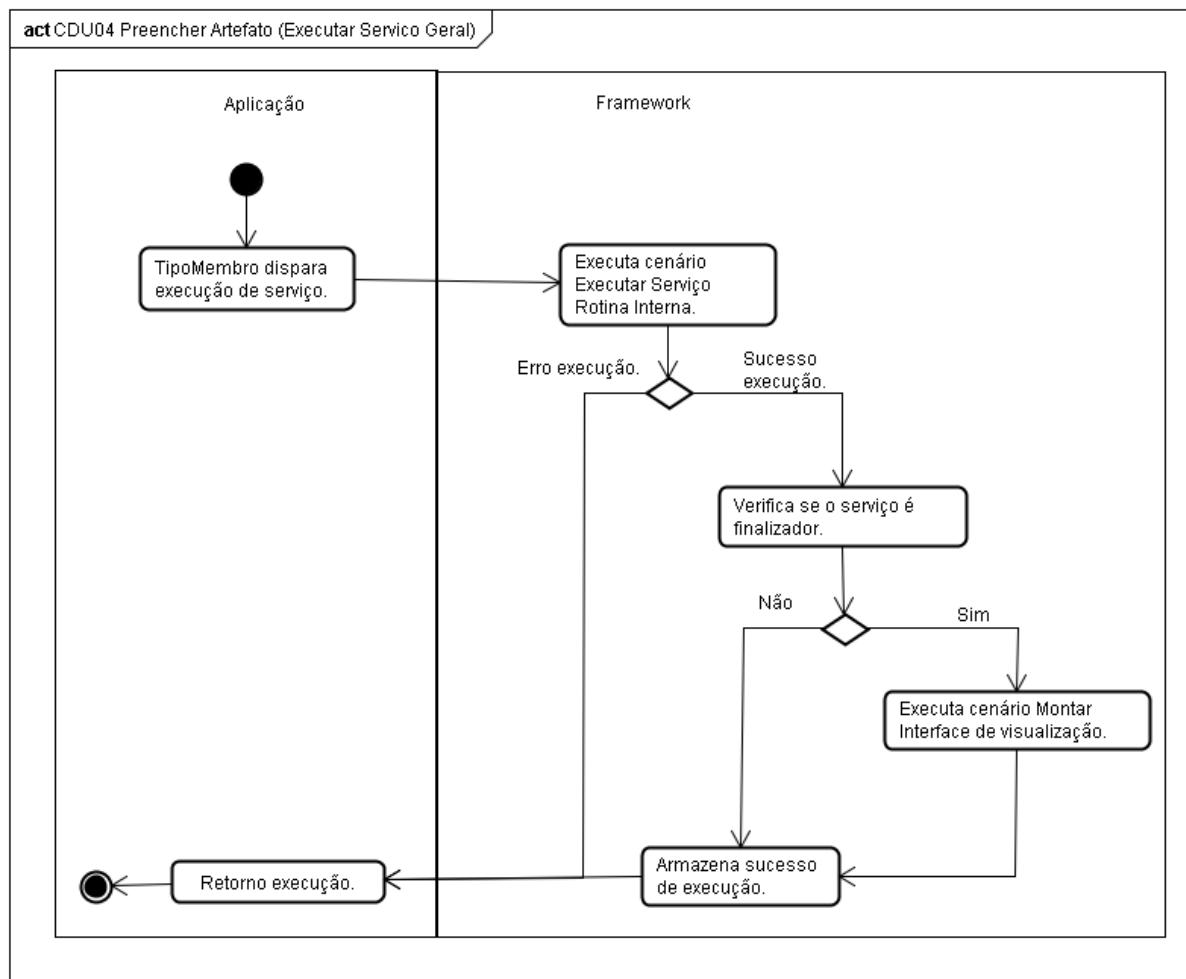


Figura 68 – Diagrama de Atividades Cenário 01 – Executar Serviço Geral.

6.4.12.2 Cenário 02 – Executar Serviço Rotina Interna.

Descrição do Cenário	Cenário responsável por realizar a execução de um serviço.
Requisitos Associados	R06 - Preencher artefato, R12 - Gerenciar Serviço de Persistência, R13 Gerenciar Serviço de Dependência, R16 Gerenciar Serviço de Validação de dados.
Pré Condições	O usuário deverá estar logado no sistema com permissão de Preenchedor.
Pós Condições	O retorno da execução do serviço.
Atores	Preenchedor
Fluxo Principal	
Ações Recebidas	
1. Recebe serviço a ser executado.	2. Identifica serviço na lista de serviços do artefato. 3. Identifica serviços anteriores ao serviço selecionado e forma uma lista de serviços a serem executados. 4. Executa serviço da lista. 5. Verifica se ainda existe serviço na lista de execução.

	<p>6. Armazena sucesso de execução com o retorno da execução dos serviços.</p> <p>7. Retorno da execução.</p>
Fluxo Alternativo 01 - Falha na execução do serviço.	
Ações Recebidas	Ações Realizadas
4.1. Falha na execução do serviço.	<p>4.2. Armazena erro de falha de execução de serviço com a descrição da falha do serviço executado.</p>
4.3. Retorno fluxo principal 7.	
Fluxo Alternativo 02 - Ainda existem serviços na lista de execução.	
Ações Recebidas	Ações Realizadas
5.1. Ainda existem serviços na lista de execução.	<p>5.2. Realiza passagem de parâmetros entre os serviços, caso necessário.</p> <p>5.3. Retorno fluxo principal 4.</p>

Tabela 117 – CDU12 – Detalhamento Cenário 02

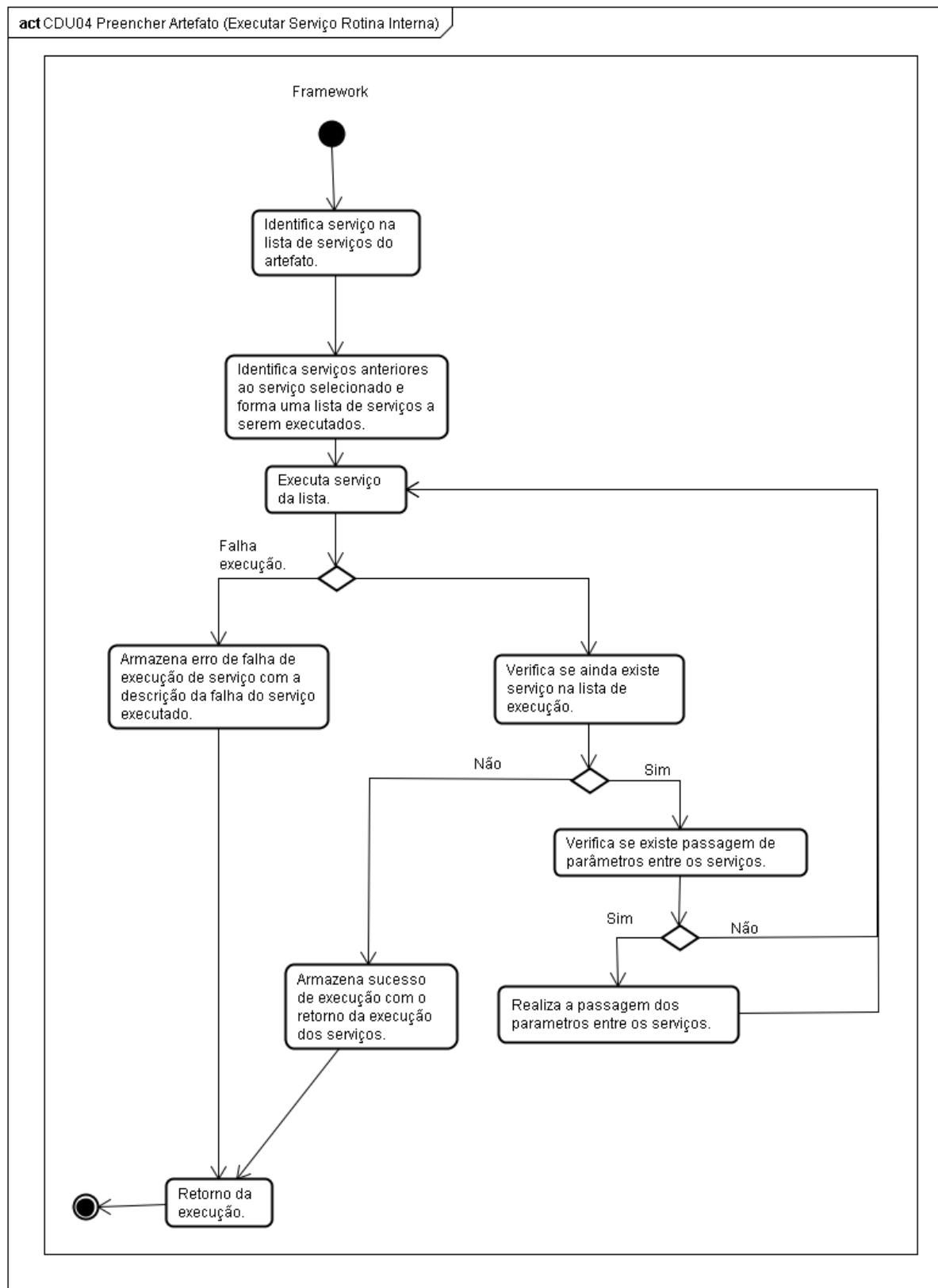


Figura 69 – Diagrama de Atividades Cenário 02 – Executar Serviço Rotina Interna.

6.4.13 Caso de Uso 13 – Realizar Versionamento

6.4.13.1 Cenário 01 – Versionar

Descrição do Cenário	Cenário responsável por verificar se um determinado componente tem uma nova verão, uma nova revisão ou um arquivo já cadastrado no framework com mesma versão e revisão.
Requisitos Associados	R01 - Manter TipoMembro, R11 - Manter Serviço
Pré Condições	O usuário deverá estar logado no sistema com permissão de desenvolvedor.
Pós Condições	O versionamento do arquivo
Atores	Desenvolvedor
Fluxo Principal	
Ações Recebidas	Ações Realizadas
38. Recebe arquivo a ser versionado.	39. Captura o código de versão e revisão do arquivo. 40. Recupera o último código da versão e da revisão do arquivo. 41. Códigos encontrados. 42. Verificação se o código de versão é igual ao código de versão do arquivo. 43. Verifica se o código de revisão é igual ao código de revisão do arquivo. 44. Armazena tipo de versionamento: Mesma versão e revisão de arquivo.
45. Retorno da execução.	
Fluxo Alternativo 01 - Códigos não recuperados	
Ações Recebidas	Ações Realizadas
5.19. Códigos não recuperados.	5.20. Armazena tipo de versionamento: Nova Versão de Arquivo.
5.21. Retorno fluxo principal 8.	
Fluxo Alternativo 02 - Código de versão do arquivo diferente do código recuperado	
Ações Recebidas	Ações Realizadas
6.1. Código de versão arquivo diferente do código recuperado.	6.2. Armazena tipo de versionamento: Nova Versão de Arquivo.
6.3. Retorno fluxo principal 8.	
Fluxo Alternativo 03 - Código de revisão do arquivo diferente do código recuperado	
Ações Recebidas	Ações Realizadas
6.1. Código de revisão arquivo diferente do código recuperado.	6.2. Armazena tipo de versionamento: Nova Revisão de Arquivo.
6.3. Retorno fluxo principal 8.	

Tabela 118 – CDU13 – Detalhamento Cenário 01

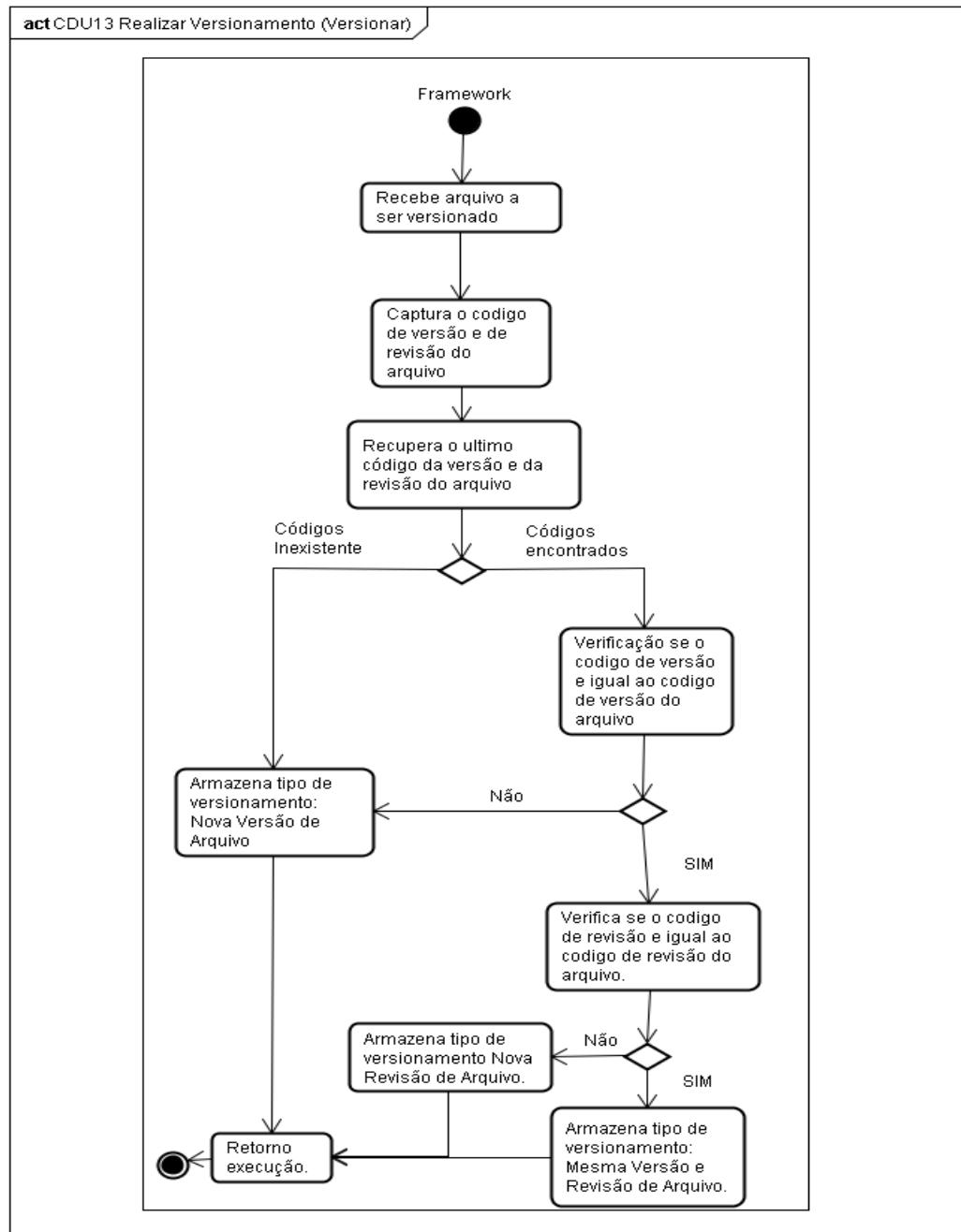


Figura 70 – Diagrama de Atividades Cenário 01 – Versionar

6.4.14 Caso de Uso 14 – Manter Opções de Log das Operações do Sistema

6.4.14.1 Cenário 1 – Alterar Opção de Log

Descrição do Cenário	Cenário responsável por alterar as opções de log do framework
Requisitos Associados	R14 – Manter Log das Operações do Sistema
Pré Condições	O usuário deverá estar logado no sistema com permissão de gerente.
Pós Condições	Opções de log alterados no framework.
Atores	Gerente de Projeto

Fluxo Principal	
Ações Recebidas	Ações Realizadas
25. Solicita alteração das opções de Log do framework.	
28. Retorno execução.	26. Altera opções de log. 27. Armazena sucesso de execução.

Tabela 119 – CDU14 – Detalhamento Cenário 01

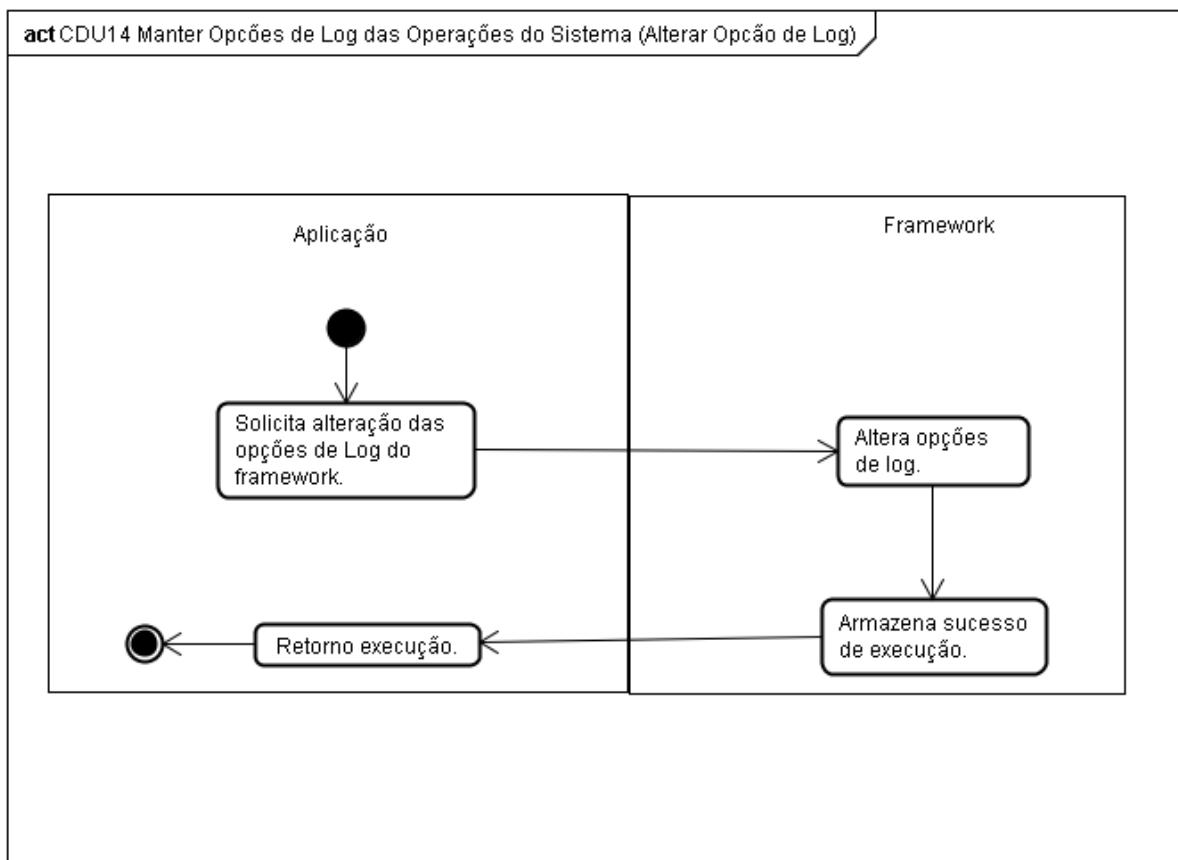


Figura 71 – Diagrama de Atividades Cenário 01 – Alterar Opção de Log

6.4.15 Caso de Uso 15 – Gerenciar Multusuários

6.4.15.1 Cenário 1 – Incluir Usuário

Descrição do Cenário	Cenário responsável por incluir novos usuários no framework.
Requisitos Associados	R15 – Gerenciar Multusuários
Pré Condições	O usuário deverá estar logado no sistema com permissão de usuário gerente de projeto.
Pós Condições	O usuário cadastrado no framework
Atores	Gerente de Projeto
Fluxo Principal	

Ações Recebidas	Ações Realizadas
29. Solicita cadastro de usuário no framework. 32. Retorno execução. 33. Preenche os atributos do usuário. 37. Retorno execução.	30. Armazena instância do novo usuário. 31. Armazena sucesso de execução. 34. Recebe instancia do usuário e verifica a obrigatoriedade dos campos preenchidos e duplicidade de nome de usuário. 35. Cadastra usuário no framework. 36. Armazena sucesso de execução.
Fluxo Alternativo 1 – Campo(s) obrigatório(s) não preenchido(s)	
Ações Recebidas	Ações Realizadas
2.1. Um ou mais campos obrigatórios não foram preenchidos. 2.3. Retorno fluxo principal 9.	2.2. Armazena erro de campos obrigatórios, informando os campos que não foram informados.
Fluxo Alternativo 2 – Nome de usuário já cadastrado	
Ações Recebidas	Ações Realizadas
6.1. Nome de usuário já cadastrado no framework. 6.3. Retorno fluxo principal 9.	6.2. Armazena erro de nome de usuário já cadastrado.

Tabela 120 – CDU15 – Detalhamento Cenário 01

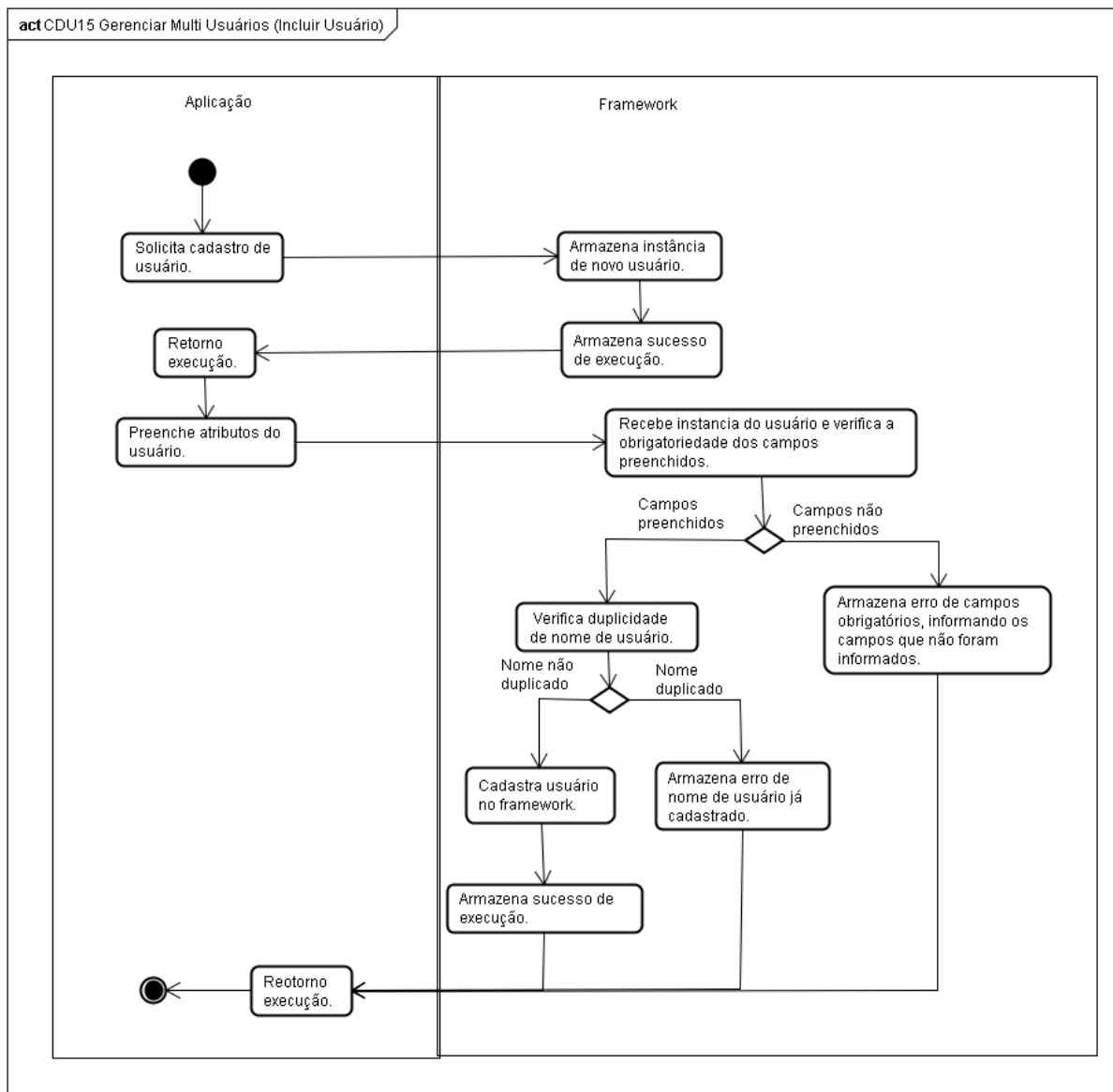


Figura 72 – Diagrama de Atividades Cenário 01 – Incluir Usuário

6.4.15.2 Cenário 2 – Alterar Usuário

Descrição do Cenário	Cenário responsável pela alteração de um usuário.	
Requisitos Associados	R15 – Gerenciar Multiusuários	
Pré Condições	O usuário deve estar logado com qualquer perfil ao sistema	
Pós Condições	O usuário alterado no sistema.	
Atores	Usuário	
Fluxo Principal		
Ações Recebidas		
1. Solicita alteração de usuário.	2. Armazena instancia do usuário logado.	
1. Retorno execução.	3. Armazena sucesso de execução.	
5. Realiza alterações no usuário.	6. Recebe instancia do usuário e verifica a	

	obrigatoriedade dos campos preenchidos e duplicidade de nome de usuário. 7. Altera usuário no framework. 8. Armazena sucesso de execução.
9. Retorno execução.	

Tabela 121 – CDU15 – Detalhamento Cenário 02

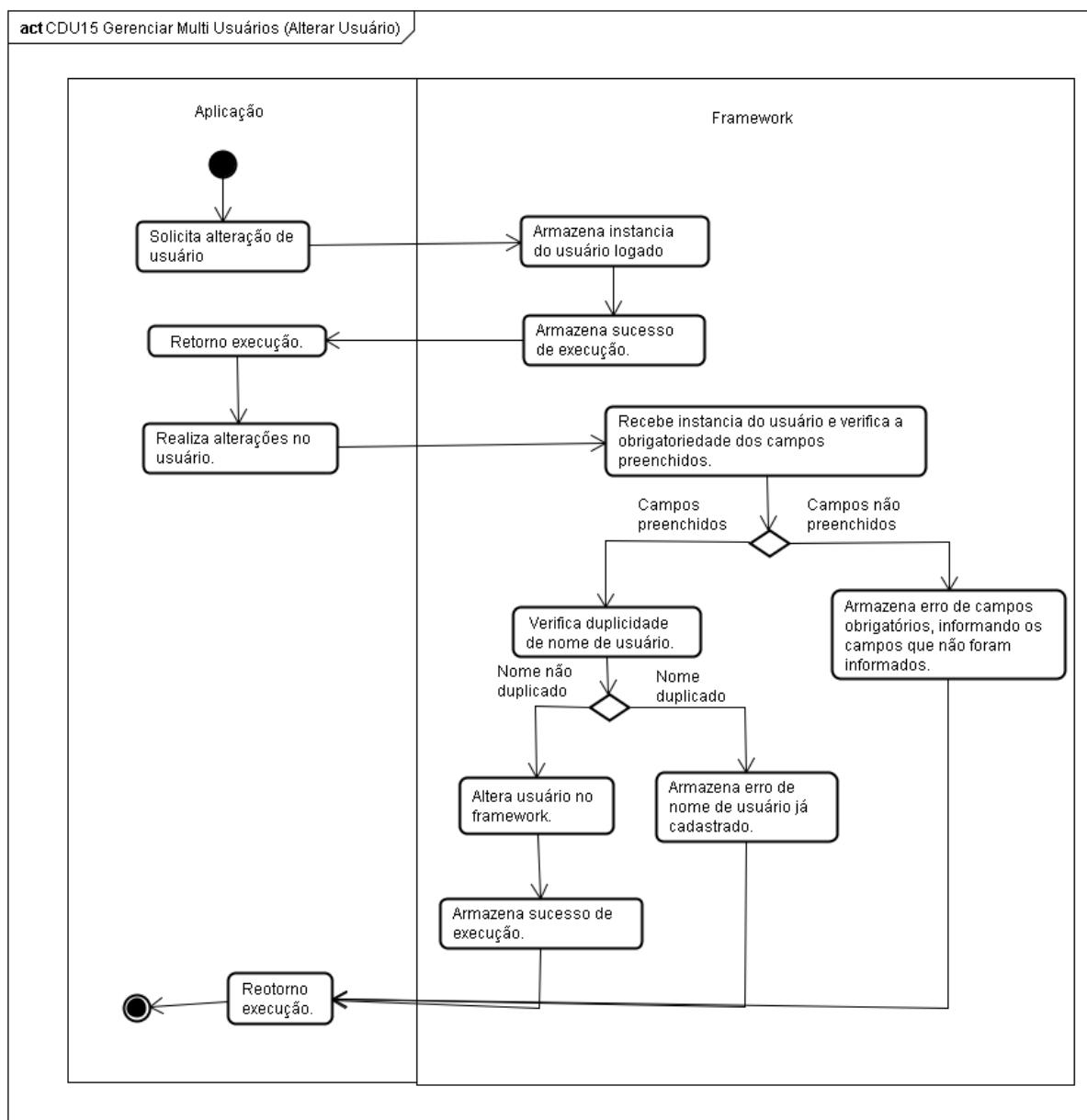
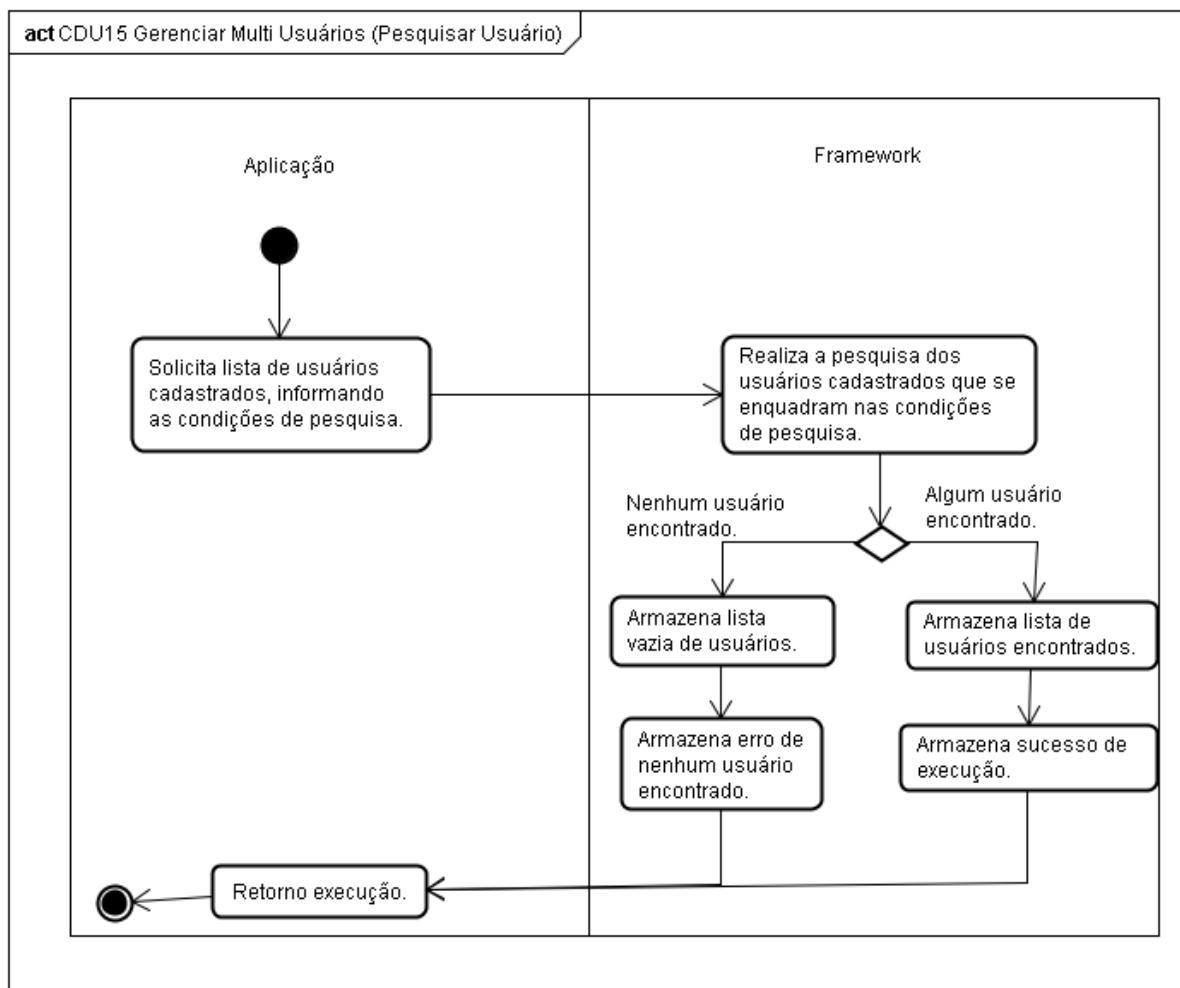


Figura 73 – Diagrama de Atividades Cenário 02 – Alterar Usuário

6.4.15.3 Cenário 3 – Pesquisar Usuário

Descrição do Cenário	Cenário responsável por pesquisar usuários cadastrados.
Requisitos Associados	R15 – Gerenciar Multusuários
Pré Condições	O usuário deve estar logado com permissão de Gerente de Projeto.

Pós Condições	Lista de usuários.
Atores	Gerente de Projeto
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1. Solicita lista de usuários cadastrados, informando as condições de pesquisa. 5. Retorno execução.	2. Realiza a pesquisa dos usuários cadastrados que se enquadram nas condições de pesquisa. 3. Armazena lista de usuários encontrados. 4. Armazena sucesso de execução.
Fluxo Alternativo 1 – Nenhum usuário encontrado	
Ações Recebidas	Ações Realizadas
2.1. Nenhum usuário encontrado. 2.4. Retorno fluxo principal 5.	2.2. Armazena lista de usuários vazia. 2.3. Armazena erro de nenhum usuário encontrado.

Tabela 122 – CDU15 – Detalhamento Cenário 03**Figura 74 – Diagrama de Atividades Cenário 03 – Alterar Usuário**

6.4.15.4 Cenário 4 – Logon de Usuário

Descrição do Cenário	Cenário responsável por realizar o logon dos usuários.
Requisitos Associados	R15 – Gerenciar Multusuários
Pré Condições	
Pós Condições	O usuário logado no sistema.
Atores	Usuário
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1. Solicita logon de usuário.	2. Verifica se o usuário está cadastrado. 3. Verifica se o usuário está ativo. 4. Realiza logon do usuário. 5. Armazena sucesso de execução.
6. Retorno da execução.	
Fluxo Alternativo 1 – Usuário não cadastrado	
Ações Recebidas	Ações Realizadas
2.1. Usuário informado não esta cadastrado.	2.2. Armazena erro de usuário não cadastrado.
2.3. Retorno fluxo principal 5.	
Fluxo Alternativo 2 – Usuário está inativo	
Ações Recebidas	Ações Realizadas
3.1. Usuário informado esta inativo.	3.2. Armazena erro de usuário inativo.
3.3. Retorno fluxo principal 5.	

Tabela 123 – CDU15 – Detalhamento Cenário 04

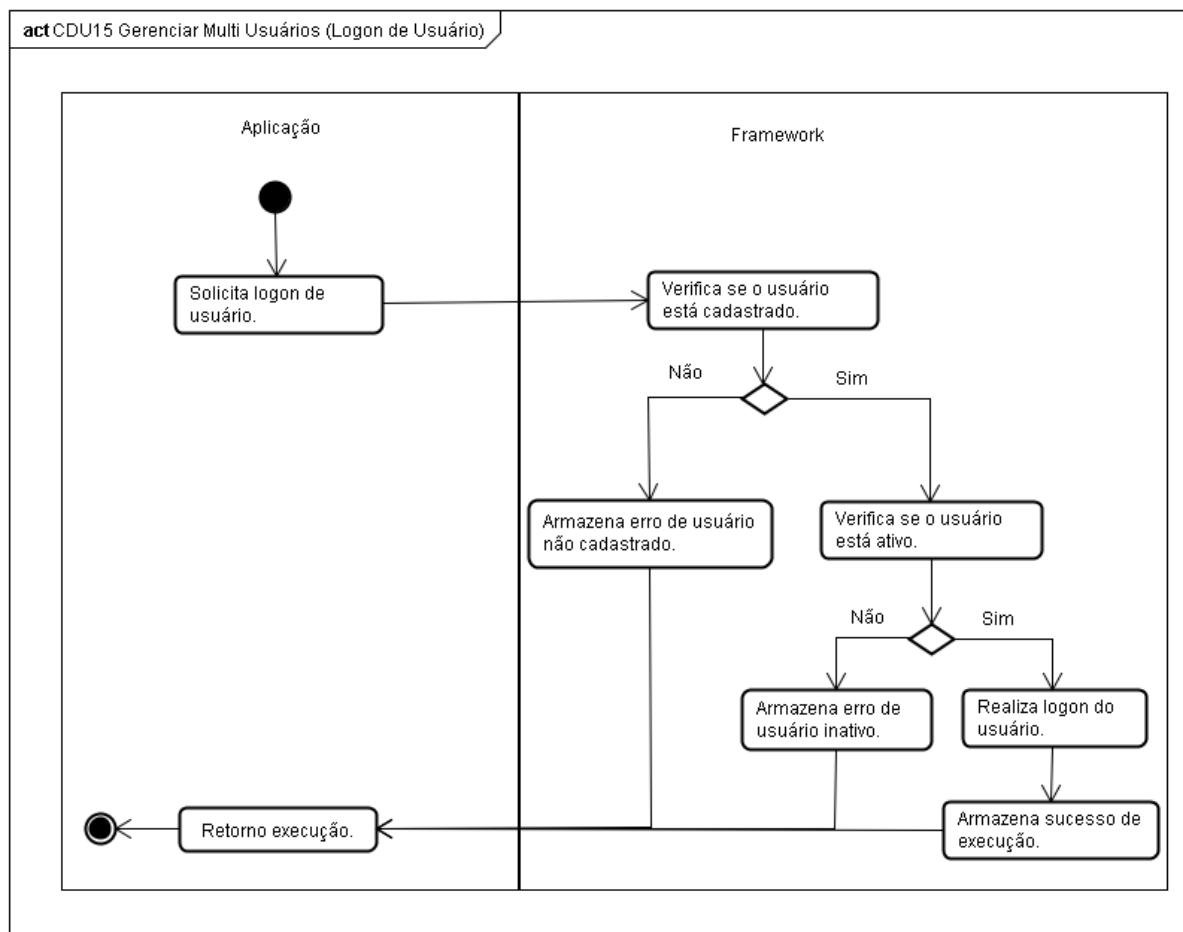


Figura 75 – Diagrama de Atividades Cenário 04 – Logon de Usuário

6.4.15.5 Cenário 5 – Incluir Equipe

Descrição do Cenário	Cenário responsável pela criação de uma equipe.
Requisitos Associados	R15 – Gerenciar Multusuários.
Pré Condições	O usuário deve estar logado com perfil de gerente de projeto.
Pós Condições	A equipe cadastrada .
Atores	Gerente de Projeto
Fluxo Principal	
Ações Recebidas	
1. Solicita cadastro de equipe.	2. Instancia uma nova equipe.
4. Retorno execução.	3. Armazena sucesso de execução.
5. Preenche os atributos da equipe.	6. Verifica os campos obrigatórios e a duplicidade de cadastro.
	7. Cadastra equipe.
	8. Armazena sucesso de execução.
9. Retorno execução.	
Fluxo Alternativo 1 – Campo(s) obrigatório(s) não preenchido(s)	
Ações Recebidas	
6.1. Campos obrigatórios não informados.	

	6.2. Armazena erro de campos obrigatórios, informando os campos não informados.
6.3. Retorno fluxo principal 9.	
Fluxo Alternativo 2 – Duplicidade de cadastro de equipe	
Ações Recebidas	Ações Realizadas
6.1. Equipe já cadastrada	6.2. Armazena erro de duplicidade de cadastro de equipe.
6.3. Retorno fluxo principal 9.	

Tabela 124 – CDU15 – Detalhamento Cenário 05

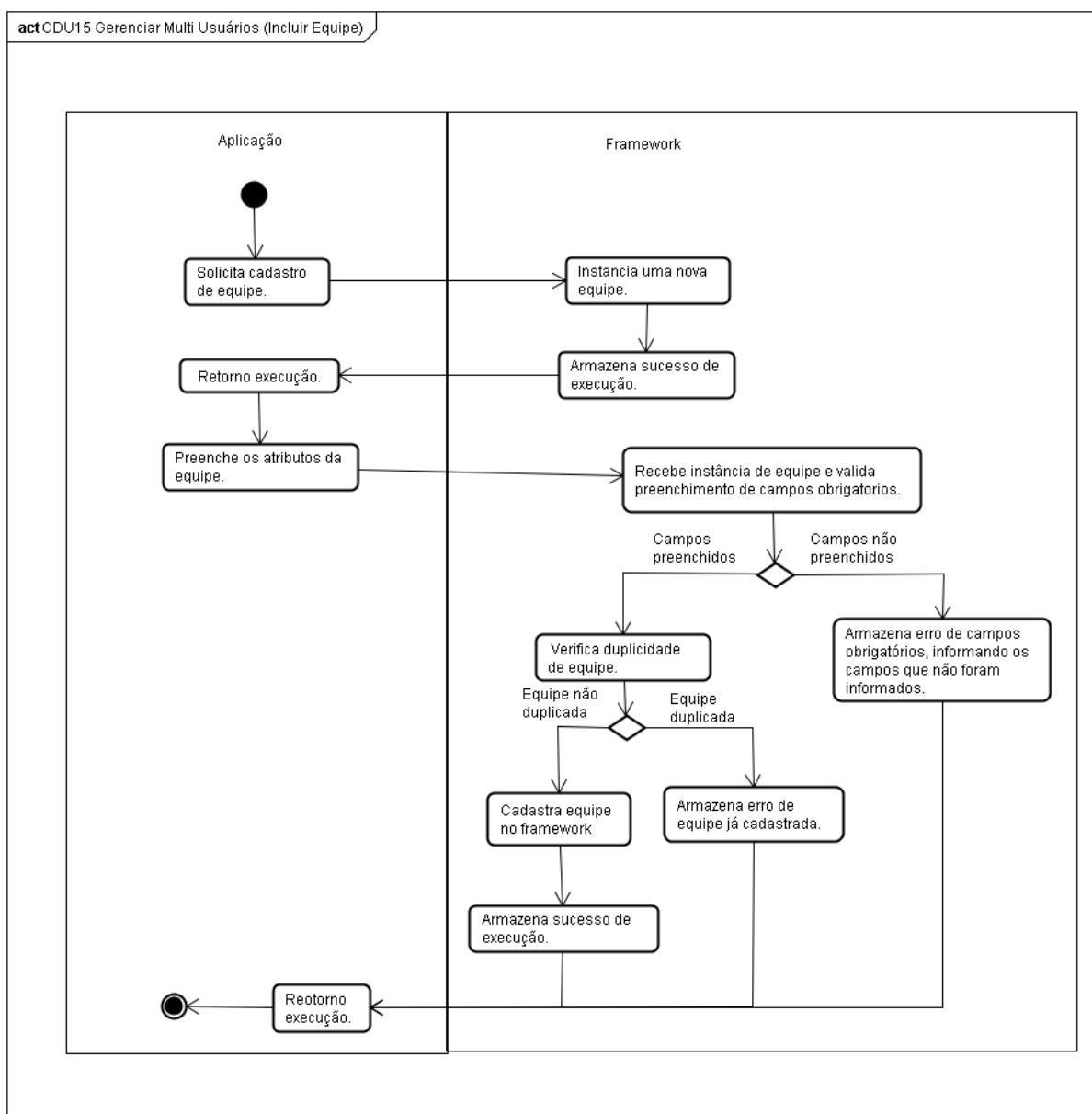


Figura 76 – Diagrama de Atividades Cenário 05 – Incluir Equipe

6.4.15.6 Cenário 6 – Desativar Usuário

Descrição do Cenário	Cenário responsável por desativar um usuário.
Requisitos Associados	R15 – Gerenciar Multiusuários
Pré Condições	O usuário deve estar logado com permissão de gerente de projeto.
Pós Condições	Usuário desativado.
Atores	Gerente de Projeto
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1. Solicita desativação de um usuário. 5. Retorno execução.	2. Verifica se usuário esta logado. 3. Desativa o usuário. 4. Armazena sucesso de execução.
Fluxo Alternativo 1 – Usuário logado	
Ações Recebidas	Ações Realizadas
2.1. Usuário logado 2.3. Retorno fluxo principal 3.	2.2. Desloga usuário.

Tabela 125 – CDU15 – Detalhamento Cenário 06

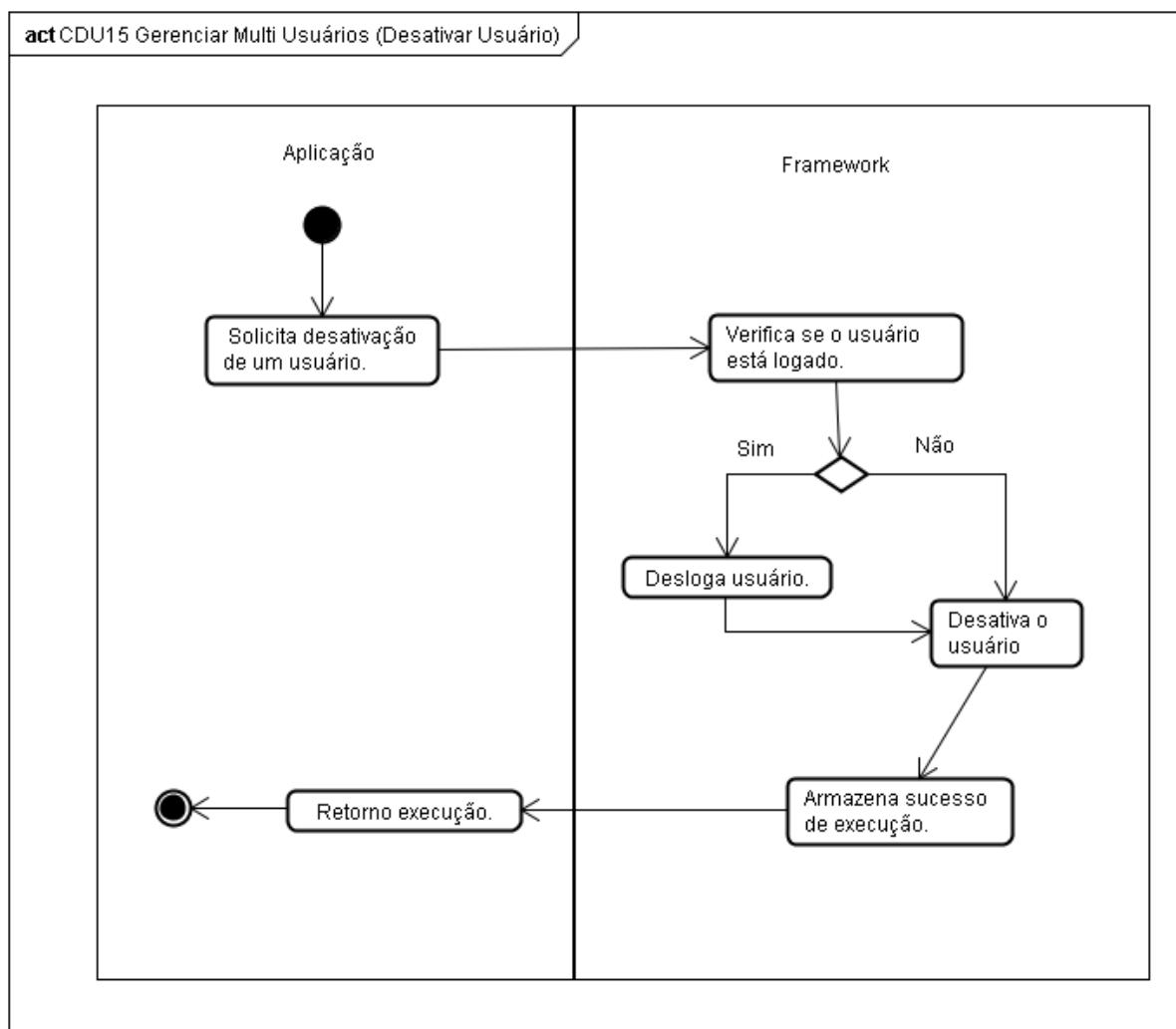


Figura 77 – Diagrama de Atividades Cenário 06 – Desativar Usuário

6.4.15.7 Cenário 7 – Associar Usuários à uma Equipe

Descrição do Cenário	Cenário responsável pela ligação de usuários a uma equipe.
Requisitos Associados	R15 – Gerenciar Multiusuários
Pré Condições	O usuário deve estar logado com permissão de gerente de projeto.
Pós Condições	Usuários associados a uma equipe.
Atores	Gerente de Projeto.
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1. Solicita inclusão de usuários em uma equipe, informando os usuários e a equipe onde eles serão inseridos. 6. Retorno da execução.	2. Verifica se os usuários selecionados estão ativos. 3. Verifica duplicidade de cadastro de usuário em na equipe. 4. Cadastra usuários na equipe. 5. Armazena sucesso de execução.
Fluxo Alternativo 1 – Usuários informados estão inativos	
Ações Recebidas	Ações Realizadas
2.1. Usuários informados estão inativos. 2.3. Retorno fluxo principal 6.	2.2. Armazena erro de usuários inativos, informando os usuários que estão inativos.
Fluxo Alternativo 2 – Duplicidade de cadastro de usuário na equipe	
Ações Recebidas	Ações Realizadas
3.1. Usuário informado já esta cadastrado na equipe. 3.3. Retorno fluxo principal 6.	3.2. Armazena erro de duplicidade de cadastro de usuário na equipe, informando os usuários duplicados.

Tabela 126 – CDU15 – Detalhamento Cenário 07

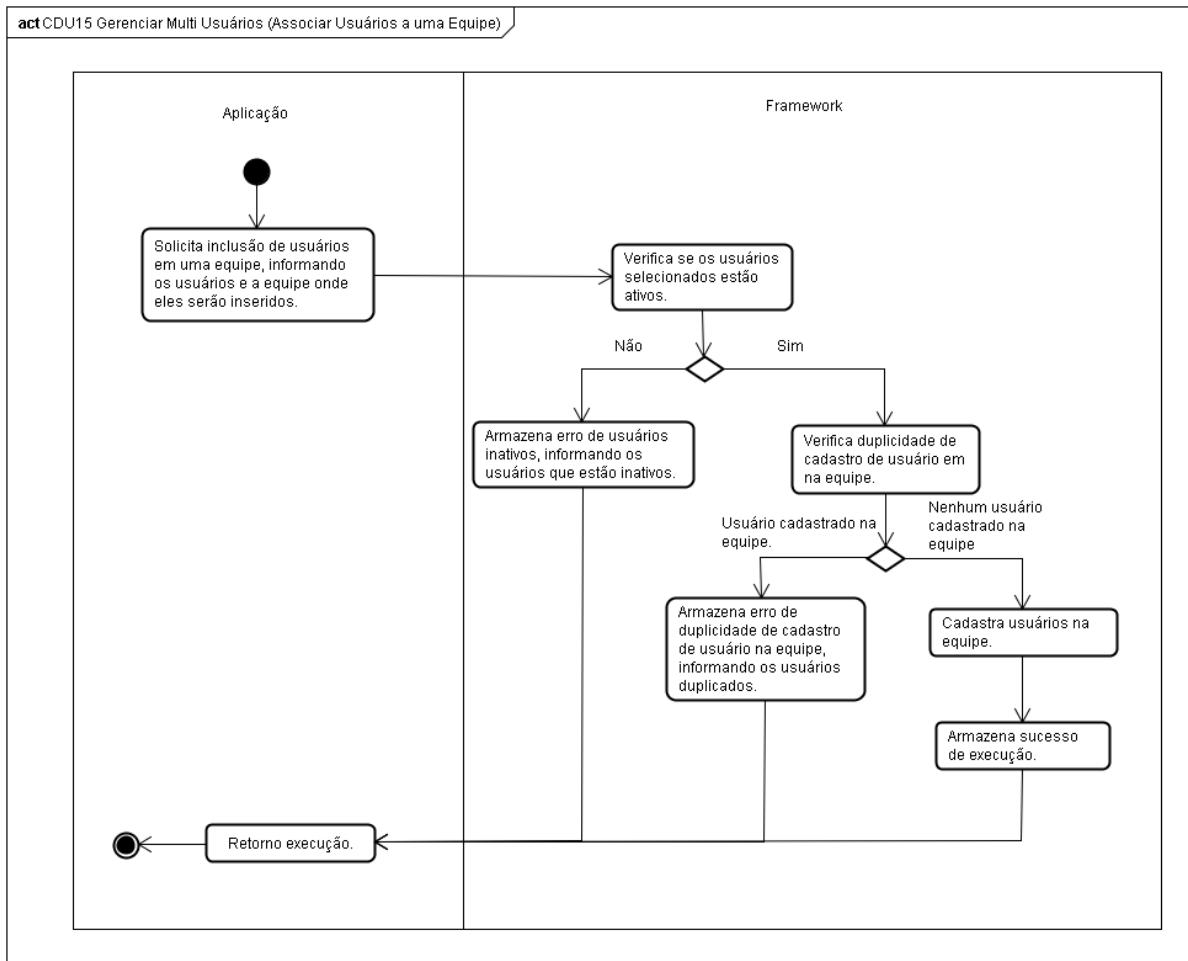


Figura 78 – Diagrama de Atividades Cenário 07 – Associar Usuários a uma Equipe

6.4.15.8 Cenário 8 – Remover Usuários de uma Equipe.

Descrição do Cenário	Cenário responsável pela remoção de usuários de uma equipe.
Requisitos Associados	R15 – Gerenciar Multusuários
Pré Condições	O usuário deve estar logado com permissão de gerente de projeto.
Pós Condições	Usuários removidos de uma equipe.
Atores	Gerente de Projeto.
Fluxo Principal	
Ações Recebidas	
1. Solicita remoção de usuários em uma equipe, informando os usuários os usuários que serão removidos, e a equipe.	2. Verifique se todos os usuários informados pertencem a equipe. 3. Verifica se existe algum projeto sendo preenchido por algum usuário a ser removido. 4. Remove usuários da equipe. 5. Armazena sucesso de execução.
6. Retorno da execução.	
Fluxo Alternativo 1 – Usuários não pertencem a equipe informada	
Ações Recebidas	
Ações Realizadas	

2.1. Usuários não pertencem a equipe informada.	2.2. Armazena erro de exclusão de usuários da equipe, informando os usuários que não pertencem a equipe.
2.3. Retorno fluxo principal 6.	
Fluxo Alternativo 2 – Projeto sendo preenchido por algum usuário que será removido	
Ações Recebidas	Ações Realizadas
3.1. Projeto sendo preenchido por algum usuário que será removido	3.2. Armazena erro de usuário preenchendo projeto, informando os usuários que estão preenchendo projeto.
3.3. Retorno fluxo principal 6.	

Tabela 127 – CDU15 – Detalhamento Cenário 08

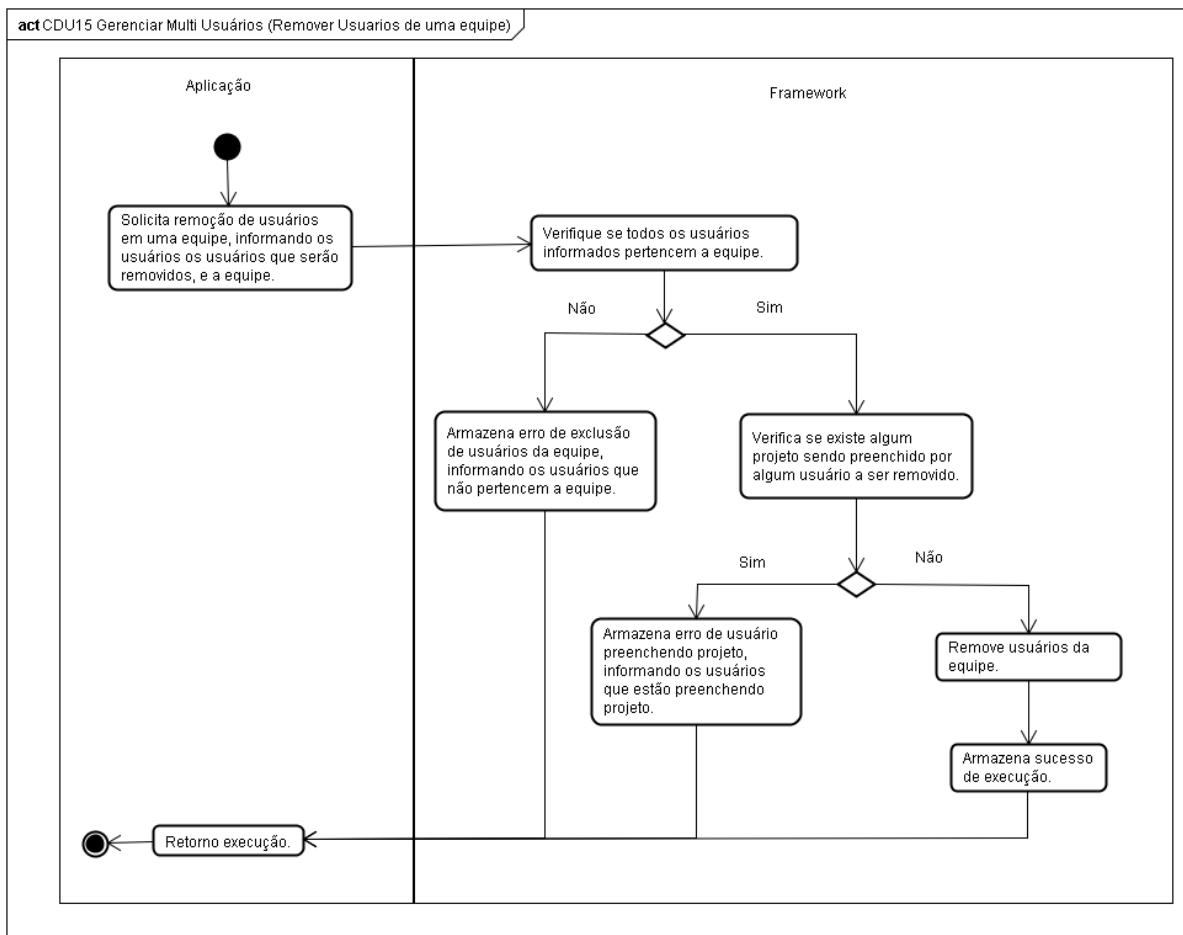


Figura 79 – Diagrama de Atividades Cenário 08 – Remover Usuários de uma Equipe

6.4.15.9 Cenário 9 – Listar usuários de uma Equipe.

Descrição do Cenário	Cenário responsável por listar os usuários que pertencem a uma equipe.
-----------------------------	--

Requisitos Associados	R15 – Gerenciar Multiusuários
Pré Condições	O usuário deve estar logado com permissão de gerente de projeto.
Pós Condições	Lista de usuários de uma equipe.
Atores	Gerente de Projeto.
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1. Solicita lista de usuários que pertencem à equipe. 5. Retorno da execução.	2. Consulta todos os usuários que pertencem à equipe informada. 3. Armazena lista de usuários. 4. Armazena sucesso de execução.

Tabela 128 – CDU15 – Detalhamento Cenário 09

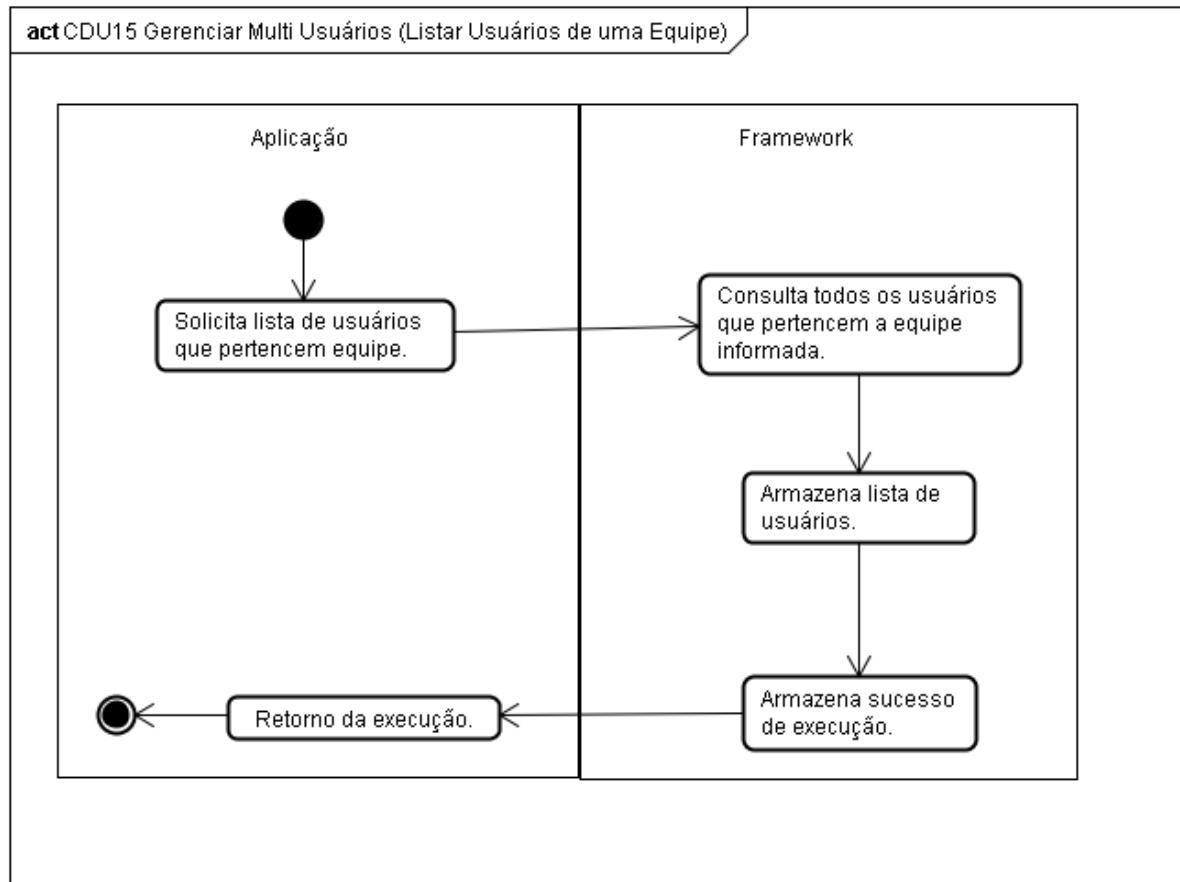


Figura 80 – Diagrama de Atividades Cenário 09 – Listar Usuários de uma Equipe

6.4.16 Caso de Uso 16 – Visualizar Log das Operações do Sistema

6.4.16.1 Cenário 1 – Pesquisar Log

Descrição do Cenário	Cenário responsável por pesquisar os logs das operações do framework gravados anteriormente.
Requisitos Associados	R14 – Manter Log das Operações do Sistema
Pré Condições	O usuário deve estar logado no sistema com permissão de gerente de projeto.
Pós Condições	Lista de logs do sistema.
Atores	Usuário
Fluxo Principal	
Ações Recebidas	
38. Solicita lista de logs, informando o período e o nível de log.	39. Pesquisa logs do nível e do período informado. 40. Armazena lista de logs. 41. Armazena sucesso de execução.
42. Retorno da execução.	
Fluxo Alternativo 01 - Nenhum log encontrado	
Ações Recebidas	
5.22. Nenhum log encontrado.	5.23. Armazena lista de logs vazia. 5.24. Armazena erro de nenhum log encontrado.
5.25. Retorno fluxo principal 5.	

Tabela 129 – CDU16 – Detalhamento Cenário 01

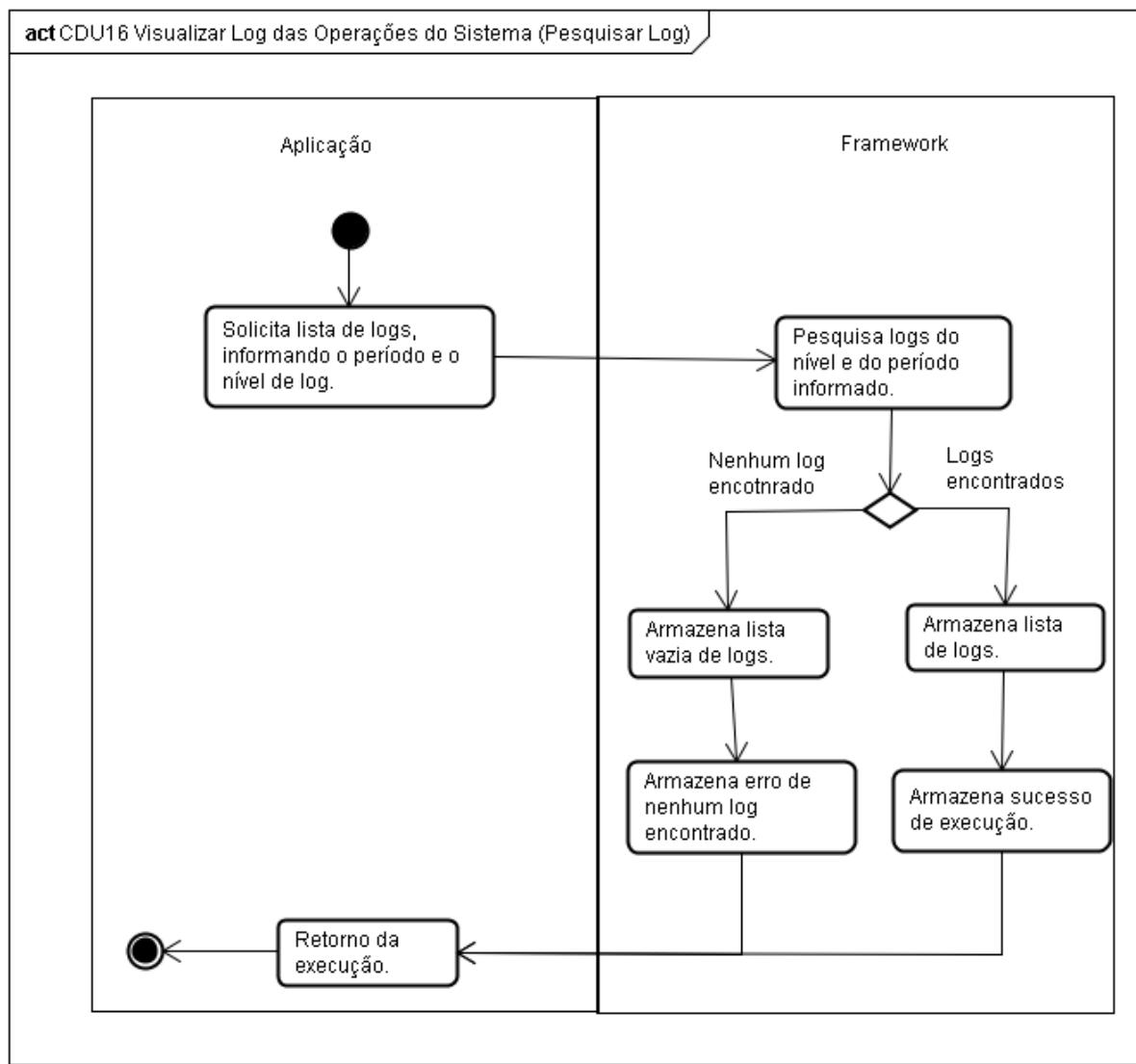


Figura 81 – Diagrama de Atividades Cenário 01 – Pesquisar Log

6.4.17 Caso de Uso 17 - Registrar Log das Operações do Sistema

6.4.17.1 Cenário 01 - Registrar Log

Descrição do Cenário	Cenário responsável por registrar os logs das operações que ocorrerem no framework.
Requisitos Associados	R14 - Manter log das operações do sistema.
Pré Condições	Ocorrer alguma operação.
Pós Condições	Log armazenado
Atores	Sistema
Fluxo Principal	
Ações Recebidas	
43. Realiza operação	44. Verifica se esta sendo registrado log para nível de operação efetuada.

	45. Realiza registro do log da operação. 46. Finaliza cenário.
Fluxo Alternativo 01 - Não está sendo registrado log para este nível de operação	
Ações Recebidas	Ações Realizadas
5.26. Não está sendo registrado log para este nível de operação.	5.27. Retorna fluxo principal 4.

Tabela 130 – CDU17 – Detalhamento Cenário 01

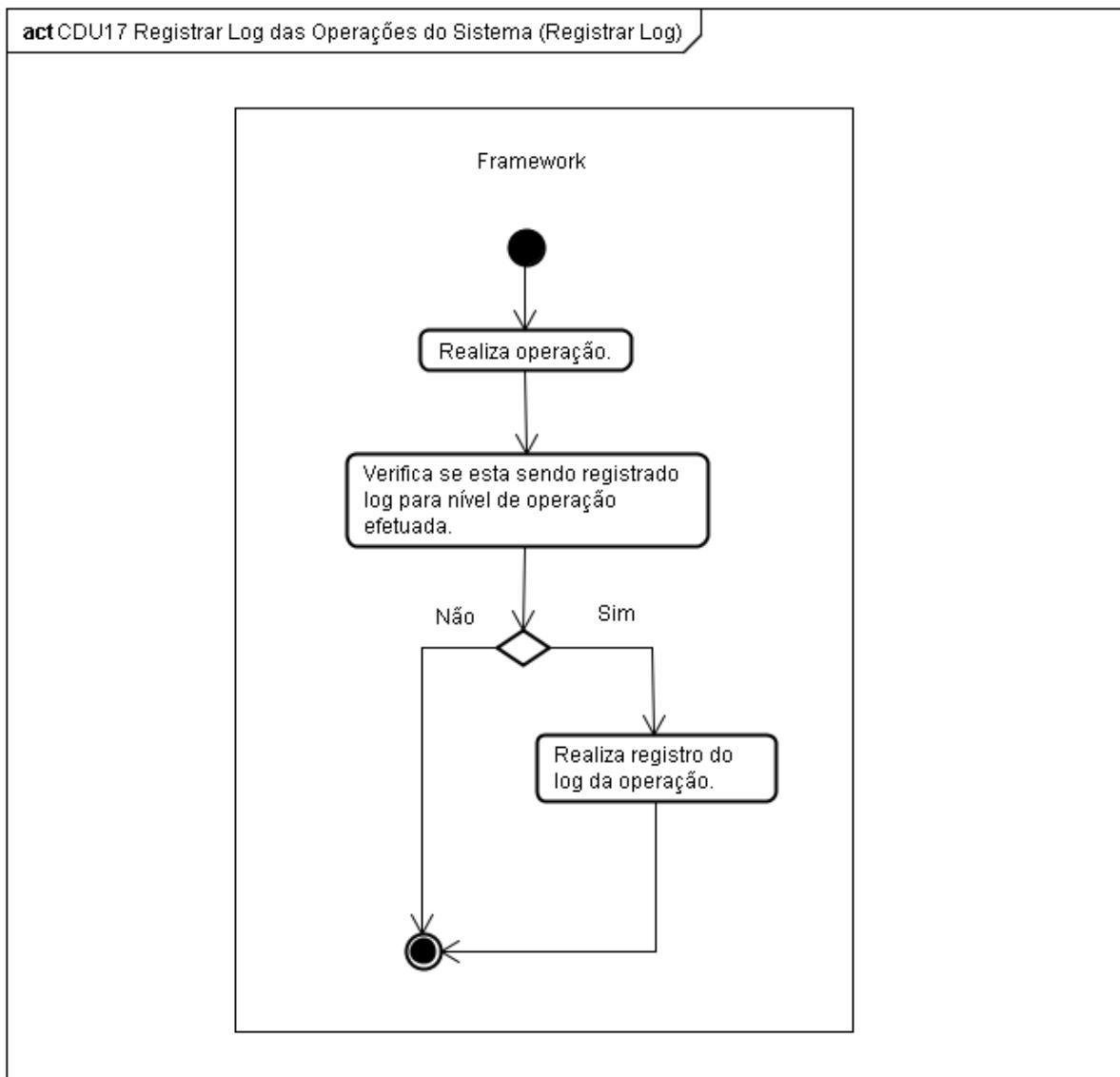


Figura 82 – Diagrama de Atividades Cenário 01 – Registrar Log

7 Modelagem de Dados

7.1 Projeto Conceitual de Dados

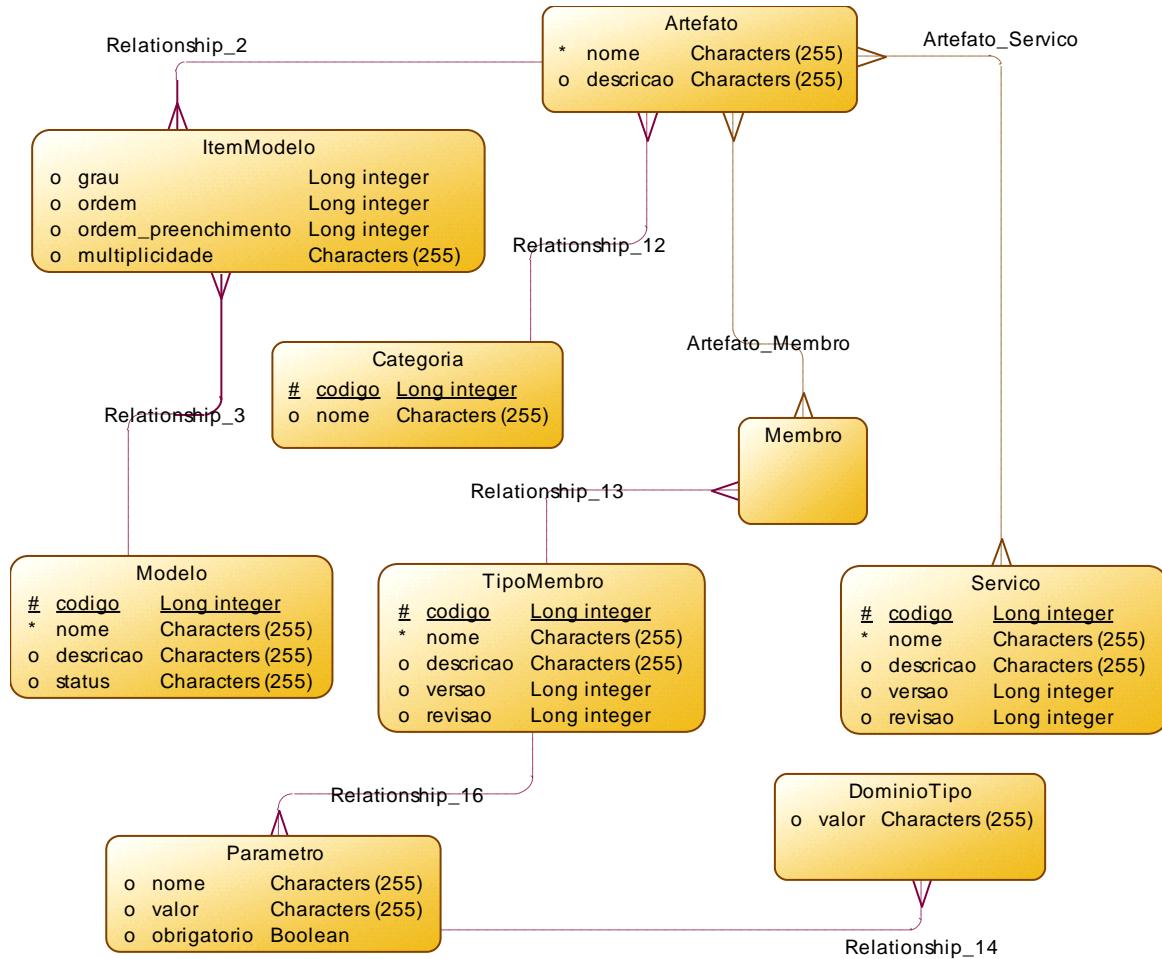


Figura 83 – Projeto Conceitual do Banco de Dados

7.2 Projeto Lógico de Dados

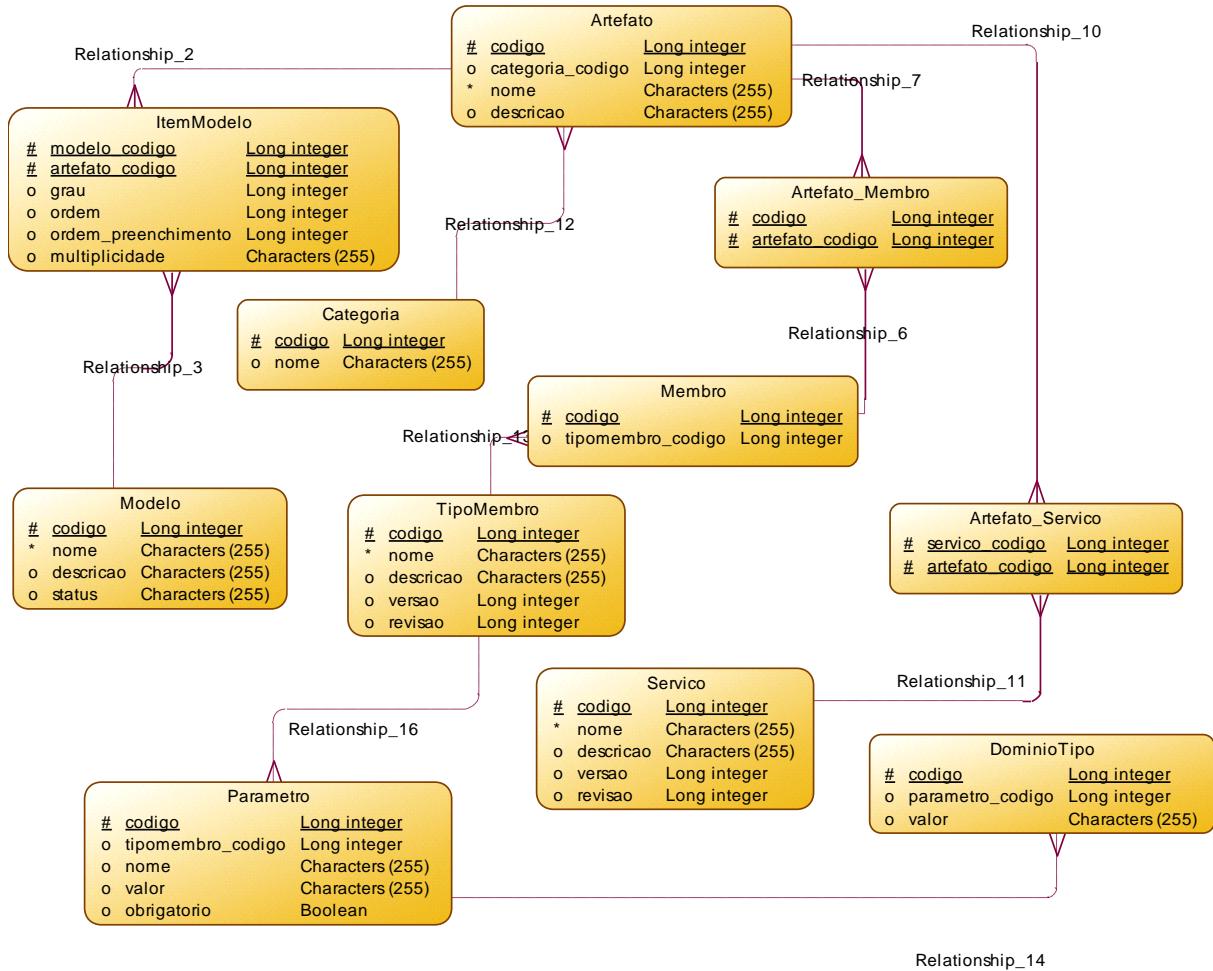


Figura 84 – Projeto Lógico do Banco de Dados

7.3 Script de Criação do Banco de Dados

Mesmo apresentando aqui o script de criação do banco, o framework está capacitado para criar todas as tabelas no banco a partir do framework de persistência Hibernate, porém o banco deve estar criado e ativo.

-- Base de dados

```
CREATE DATABASE quid;
```

-- Usuario

```
CREATE TABLE usuario (
```

```
  codigo SERIAL PRIMARY KEY NOT NULL,
```

```
  email character varying(255),
```

```
  nivelacesso character varying(255),
```

```
    nome character varying(255),  
    senha character varying(255),  
    status character varying(255),  
    perfilacesso character varying(255)  
);
```

-- Equipe

```
CREATE TABLE equipe (  
    codigo SERIAL PRIMARY KEY NOT NULL,  
    nome character varying(255)  
);
```

-- Equipe - Usuario

```
CREATE TABLE equipe_usuario (  
    equipe_codigo bigint NOT NULL REFERENCES equipe(codigo),  
    usuario_codigo bigint NOT NULL REFERENCES usuario(codigo)  
);
```

-- Categoria

```
CREATE TABLE categoria (  
    codigo SERIAL PRIMARY KEY NOT NULL,  
    descricao character varying(255)  
);
```

-- Artefato

```
CREATE TABLE artefato (  
    codigo SERIAL PRIMARY KEY NOT NULL,  
    descricao character varying(255),  
    nome character varying(255),  
    categoria_codigo bigint REFERENCES categoria(codigo)  
);
```

-- Modelo

```
CREATE TABLE modelo (
    codigo SERIAL PRIMARY KEY NOT NULL,
    descricao character varying(255),
    nome character varying(255)
);
```

-- ItemModelo

```
CREATE TABLE item_modelo (
    codigo SERIAL PRIMARY KEY NOT NULL,
    grau integer,
    multiplicidade character varying(255),
    ordem integer,
    ordempreenchimento integer,
    artefato_codigo bigint REFERENCES artefato(codigo),
    modelo_codigo bigint NOT NULL REFERENCES modelo(codigo),
    ordempai integer
);
```

-- Servico

```
CREATE TABLE servico (
    codigo SERIAL PRIMARY KEY NOT NULL,
    descricao character varying(255),
    nome character varying(255),
    revisao integer NOT NULL,
    versao integer NOT NULL,
    aguardandoremocao boolean,
    ativo boolean
);
```

-- TipoMembro

```
CREATE TABLE tipo_membro (
```

```

codigo SERIAL PRIMARY KEY NOT NULL,
descricao character varying(255),
nome character varying(255),
revisao integer NOT NULL,
versao integer NOT NULL,
aguardandoexclusao boolean,
ativo boolean,
tipotipomembro character varying(255) NOT NULL
);

```

-- Membro

```

CREATE TABLE membro (
    codigo SERIAL PRIMARY KEY NOT NULL,
    descricao character varying(255),
    nome character varying(255),
    parametros bytea,
    tipomembro_codigo bigint REFERENCES tipo_membro(codigo)
);

```

-- Artefato - Servico

```

CREATE TABLE artefato_servico (
    codigo SERIAL PRIMARY KEY NOT NULL,
    anteriorobrigatorio boolean NOT NULL,
    parametrosservico bytea,
    parametrostipomembro bytea,
    artefato_codigo bigint REFERENCES artefato(codigo),
    servico_codigo bigint REFERENCES servico(codigo),
    servicoanterior_codigo bigint REFERENCES servico(codigo),
    servicoproximo_codigo bigint REFERENCES servico(codigo)
);

```

-- Artefato - TipoMembro

```

CREATE TABLE artefato_tipomembro (
    codigo SERIAL PRIMARY KEY NOT NULL,
    altura integer,
    comprimento integer,
    x integer,
    y integer,
    artefato_codigo bigint REFERENCES artefato(codigo),
    membroframework_codigo bigint REFERENCES membro(codigo)
);

```

8 Especificação da Arquitetura

Descrever a visão arquitetural de um sistema compreende detalhar mecanismos, componentes e relacionamentos de forma a atender um conjunto de requisitos de um sistema, ou seja, a estrutura ou estruturas do sistema, que contempla elementos de software, as propriedades visíveis externamente desses elementos e o relacionamento entre eles.

A arquitetura dos sistemas será representada através das seguintes visões arquiteturais:

- **Visão Lógica:** Compreende a descrição das classes mais importantes, sua organização em pacotes de serviços e subsistemas, e a organização destes subsistemas em camadas. Apresenta as mais importantes realizações de casos de uso, demonstrando os aspectos dinâmicos da arquitetura;
- **Visão de Implementação:** Descreve a organização estática do software em seu ambiente de desenvolvimento mapeando elementos de design em componentes de implementação no modelo de implementação;
- **Visão de Implantação:** Descreve como o software será disposto e distribuído no ambiente de execução. Compreende os equipamentos físicos e a organização dos componentes de software nesses equipamentos;

8.1 Visão Lógica

O sistema utilizará uma configuração de desenvolvimento multicamada dividindo-o em dois subsistemas, aplicação e framework.

8.1.1 Camadas Lógicas

A arquitetura está baseada em camadas lógicas. Cada camada possui uma responsabilidade específica e só depende da camada imediatamente subjacente. Isto garante o maior desacoplamento do sistema, ou seja, alterações em uma camada têm pouco ou nenhum impacto nas demais.

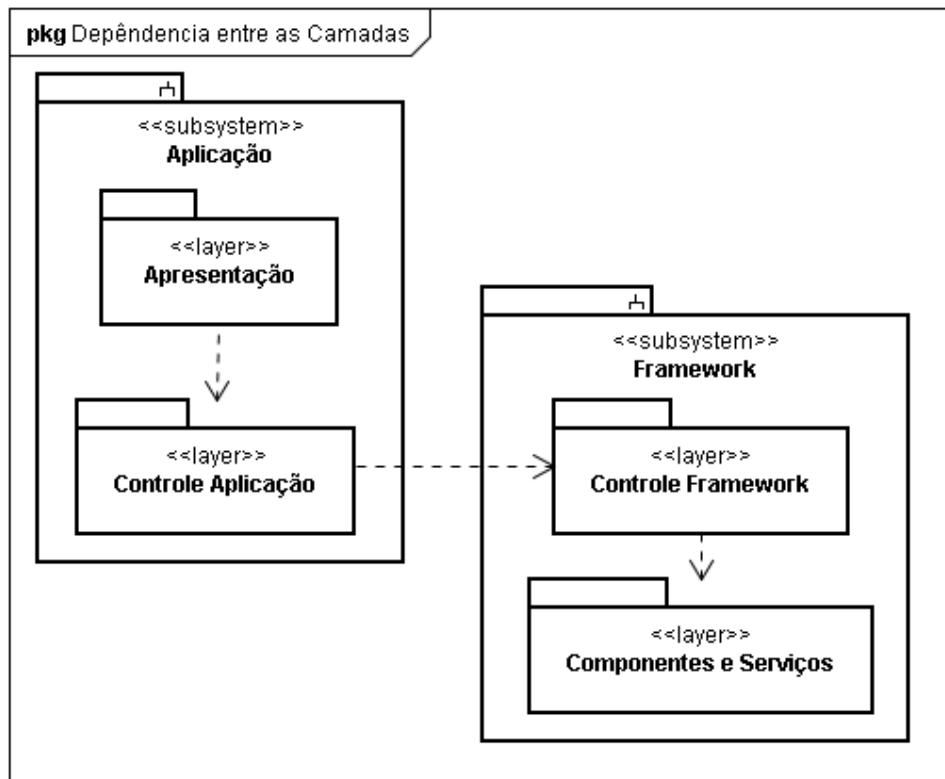


Figura 85 – Diagrama de Dependência entre as Camadas Lógicas

8.1.1.1 Camada de Apresentação

Compreende os elementos visuais, composta por arquivos representantes das telas do sistema (formato xhtml) e arquivos voltados para o controle da tela, mais precisamente chamados de ManagedBean (componente responsáveis por receber e tratar as requisições da tela), disponibilizando o acesso as funcionalidades para o usuário.

O descritor de tela surge como um dos principais componentes desta camada e é responsável pelo gerenciamento de informações entre a tela (xhtml), os ManagedBean e o TipoMembro Visão. O intuito deste componente é generalizar o modo como os componentes são desenhados na tela, ou seja, tornando possível que qualquer novo componente inserido seja desenhado sem alterações no código.

A tecnologia escolhida para desenhar as telas deveria ter a capacidade de cria-las dinamicamente a partir de uma tela base. Esta característica é de suma importância, pois a entidade TipoMembro Visão contem todo o código necessário para se desenhar, porém utiliza os dados ofertados pelo TipoMembro Modelo através do Artefato.

Em hipótese alguma esta camada deve conter regras de negócio da aplicação ou do framework. Ela deve ser simples e deve conseguir gerenciar todos os dados obtidos através da camada adjacente, a Camada de Controle da Aplicação.

8.1.1.2 Camada de Controle da Aplicação

Possui classes responsáveis pelo controle do workflow do projeto. Há uma seqüência de ações a serem respeitadas durante a execução das requisições do usuário sendo estas reencaminhadas para os artefatos da camada subjacente para que o framework execute ou não a ação requisitada.

Todas as regras de negócio voltadas para aplicação deverão ser aplicadas nesta camada, como a liberação das funcionalidades por perfil de usuário.

8.1.1.3 Camada de Controle do Framework

Possui a principal classe do framework, o Artefato. No ato da montagem do artefato são definidas um conjunto de funcionalidades possíveis para o mesmo, limitando-o e obrigando a aplicação das regras de negócio.

O Artefato é o agrupamento das informações, ou seja, é um conjunto de TipoMembros e Serviços que individualmente não passam uma informação concreta ao usuário, portanto ao serem organizados no Artefato passam uma informação consistente.

Um conjunto de Artefatos reunidos formam um Modelo, ou seja, o template propriamente dito. Se o Artefato já passava uma informação consistente, o Modelo representa a evolução e relacionamento destas informações.

8.1.1.4 Camada de Componentes e Serviços

Possui classes que representam os componentes e os serviços criados pelo usuário desenvolvedor, ou seja, todas as funcionalidades e componentes implementados. Compreende os elementos de integração responsáveis pela definição dos dados a serem persistidos, bem como as conversões necessárias. É imprescindível que cada componente criado nesta camada, o TipoMembro Modelo, tenha seu respectivo “espelho” na Camada de Apresentação, o TipoMembro Visão. O TipoMembro Modelo fornece uma estrutura única e específica pra cada componente e esta estrutura é utilizada pela camada de apresentação, seja qual for a tecnologia utilizada em sua implementação, isto é, esta entidade é a base para a entidade TipoMembro Visão, sendo possível que tenha vários TipoMembro Visão cadastrados para um mesmo TipoMembro Modelo.

O escopo desta camada está voltado apenas para ações e dados, portanto em hipótese alguma as regras de negócio do framework devem se concentrar nesta camada.

8.2 Visão de Implementação

Os fatores que influenciaram na escolha da linguagem Java para implementação do projeto foram: orientação a objetos, open source, portabilidade, documentação clara e acessível, afinidade por partes dos desenvolvedores e disponibilidade de recursos ricos e de fácil utilização (bibliotecas).

8.2.1 Definição de Herança por Camada

Abaixo serão citadas as classes de maior impacto na arquitetura do projeto, ou seja, o projeto não se resume apenas a elas.

8.2.1.1 Camada de Apresentação

Para melhor compatibilidade com o componente do sistema descritor de tela, será utilizado o framework JSF 2.x (Java Server Faces), facilitando assim o mapeamento e a manipulação de componentes no browser.

Templates serão utilizados para o desenho das telas. A partir da versão 2, o JSF integrou o Facelets a suas funcionalidades. Este framework alternativo faz uso de arquivos xhtml que, comparado

à arquivos JSP, aumenta a velocidade da aplicação de 30% à 50% pois, ao contrário dos JSP, não precisam ser compilados na primeira execução [SOUZA, 2007].

O template base para todas as páginas do projeto está localizado em WebContent/template/template_base.xhtml. Caso seja necessário a criação de novos templates, novas pastas devem ser criadas de acordo com a funcionalidade envolvida no template.

Para o controle da dinamicidade de inserção e remoção de componentes da tela, será utilizado a classe abstrata SuperManagedBean. Esta será a classe pai de todas os Managed Bean disponíveis no projeto contendo métodos de tratamento de exceções e mensagens, captura de atributos da requisição e da sessão, manipulação de arquivos dentre outros. Esta classe é de grande importância para o bom funcionamento do componente descritor de tela.

O descritor de tela trabalhará diretamente com os TipoMembro Visão, portanto haverá uma interface (contrato) que identificará e obrigará os componentes inseridos a seguirem o padrão definido. Esta interface será a ITipoMembroVisao e conterá todos os recursos necessários para se desenhar na tela e conseguir ler os dados do TipoMembro Modelo.

Esta camada deve ser totalmente individual atendendo aos requisitos da programação web, tendo uma camada no cliente(xhtml) e uma camada no servidor com características de apresentação (ManagedBean), com componentes voltados diretamente para a visão adotada.

8.2.1.2 Camada de Controle da Aplicação

Esta camada é responsável por delegar as requisições do usuário para a execução pelos artefatos da camada subjacente.

Existirão quatro classes principais caracterizadas pelos perfis possíveis do sistema: DesenvolvedorControle, MontadorControle, PreenchedorControle, GerenteProjetoControle.

As ações de cada classe estão relacionadas com as possibilidades de cada perfil.

8.2.1.3 Camada de Controle do Framework – Artefato

Esta camada é responsável por aplicar todas as regras de negócio definidas para o framework.

A classe que está totalmente envolvida e dedicada para este processo é o Artefato. Por ser a classe principal do framework não é permitido que ela seja alterada, e para isto ela será declarada como final, para que não seja possível que outra classe a tome como herança para que seus métodos sejam sobreescritos.

Outra classe de grande importância é a ItemModelo. Esta é responsável por organizar os artefatos de um Modelo, criando assim a estrutura propriamente dita do template tomado como modelo. Estas duas classes também são declaradas como final pelo mesmo motivo da classe Artefato.

8.2.1.4 Camada de Componentes e Serviços

Esta camada é onde todos os Serviços e TipoMembros Modelo implementados se localizam. Somente o Artefato pode ter acesso direto à estes componentes e caso uma camada superior necessite destas informações, esta deverá requisitar ao Artefato.

Para padronizar a execução dos Serviços e a utilização dos TipoMembro Modelo, os mesmos devem implementar a interfaces (contratos) IServico e ITipoMembroModelo, respectivamente.

8.2.1.5 Camada de Integração – DAO

Esta camada é responsável pelo armazenamento dos dados de saída obtidos durante o processamento dos dados de entrada. Também é responsável pela estrutura de armazenamento, ou seja, esta camada contém recursos para a criação da estrutura de armazenamento, caso seja necessário.

Para tais funcionalidades desenvolve-se uma persistência genérica baseada em annotations e na API JDBC oferecida pela Oracle®.

Para a persistência em um SGBD de dados provenientes dos TipoMembros Modelo, será utilizada a classe DadosJDBCDAO e para a persistência das estruturas e variação dos componentes, ou seja, a criação de tabelas e o armazenamento dos Membros, será utilizada a classe EstruturaJDBCDAO.

Para a persistência em arquivo XML, serão utilizadas as classes DadosXMLDAO e EstruturaXMLDAO.

8.3 Visão de Implantação

Para disponibilizar o sistema a configuração adotada será um servidor web, mais precisamente um container de servlets executado dentro de um servidor de aplicação.

O upload das iterações do sistema será feito remotamente através de um browser, funcionalidade esta disponibilizada pelo servidor web escolhido, Apache Tomcat v7.x.

Para o armazenamento dos dados referente ao framework será utilizado o SGBD open-source PostgreSQL v9.x. A persistência de tais dados será feita com a utilização de JPA e Hibernate, dois framework voltados para a camada de persistência.

A princípio, haverá apenas uma máquina como servidor. O sistema operacional é irrelevante, já que não impacta na arquitetura e nem no funcionamento do sistema, desde que ele suporte o SGBD e o servidor web mencionados anteriormente e que estes estejam configurados corretamente e disponibilizados (ativos) através de uma rede.

Como configuração mínima é exigido um servidor com processador de velocidade de 2,86Ghz e cache L2 de 8MB, memória RAM de 4GB e placa de rede 10/100/1000, ou seja, com capacidade de 1 GB/s. Para o cliente a única exigência é a utilização de um dos seguintes browsers: Mozilla Firefox v4.0 ou superior, Internet Explorer v8.0 ou superior, Google Chrome v10.0 ou superior. Qualquer um dos browsers escolhidos deve suportar Javascript.

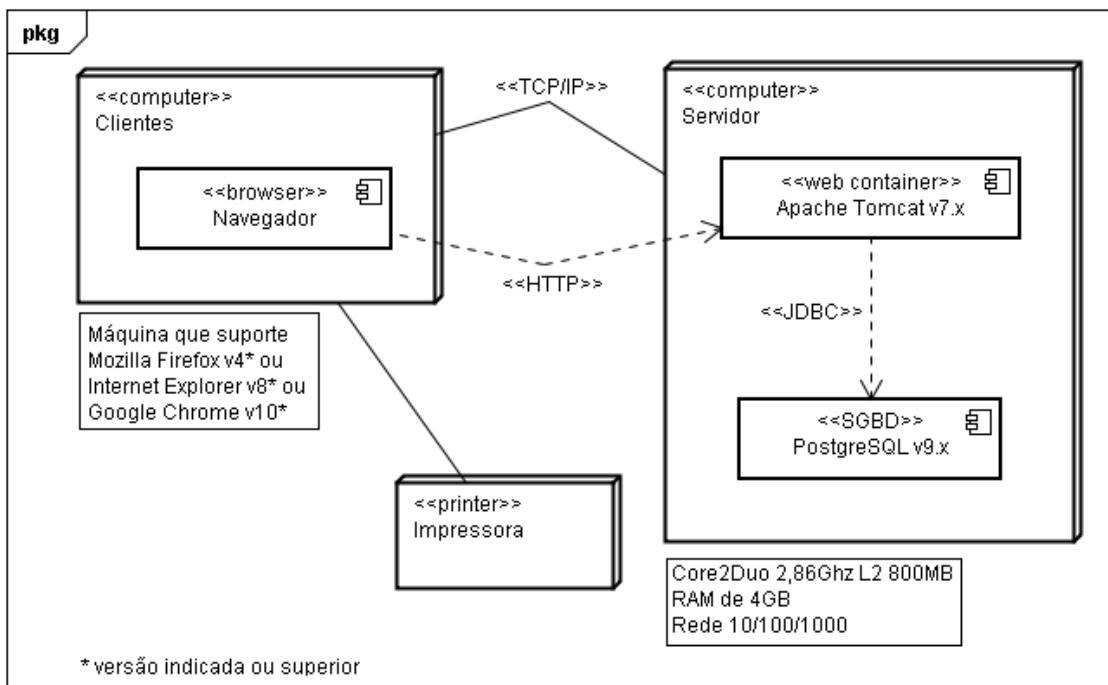


Figura 86 – Diagrama de implantação

8.4 Diagrama de Componentes

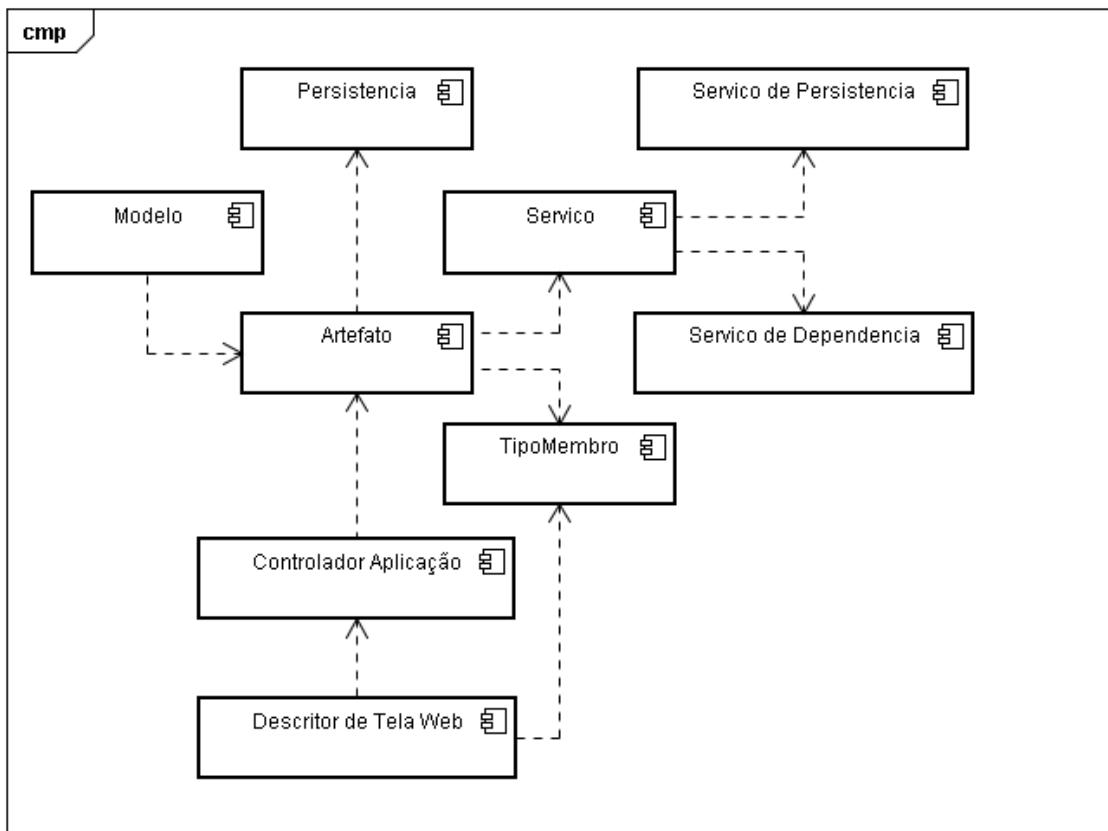


Figura 87 – Diagrama de componentes

8.5 Diagrama Geral de Sequência

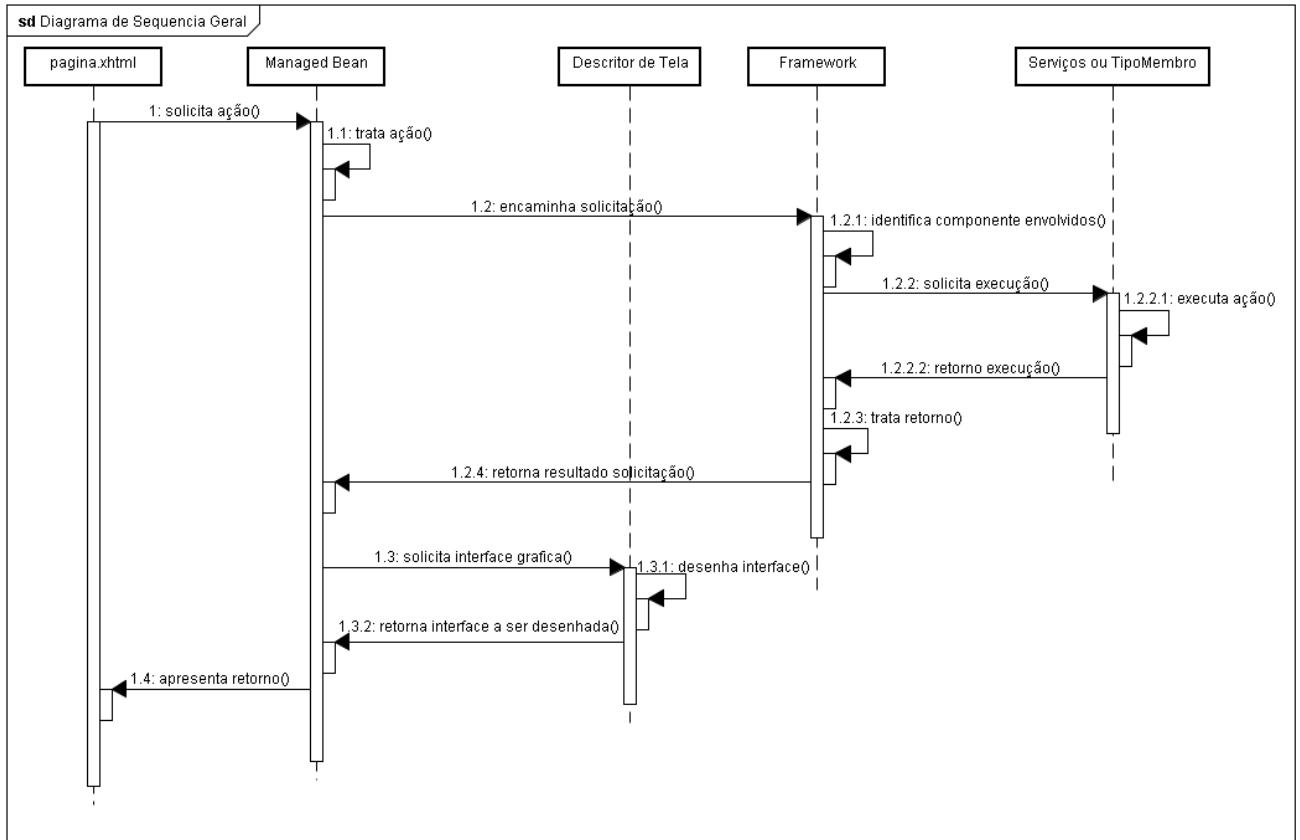


Figura 88 – Diagrama de sequencia geral

8.6 Diagrama de Pacotes

A nomenclatura base dos pacotes está dividida pelo escopo em que serão trabalhados:

- **br.ueg.unucet.quid**: contém sub-pacotes utilizados no framework;
- **br.ueg.unucet.quid.ide**: contém sub-pacotes utilizados na aplicação;
- **br.ueg.unucet.quid.extensao**: contém sub-pacotes utilizados para a extensão do framework.

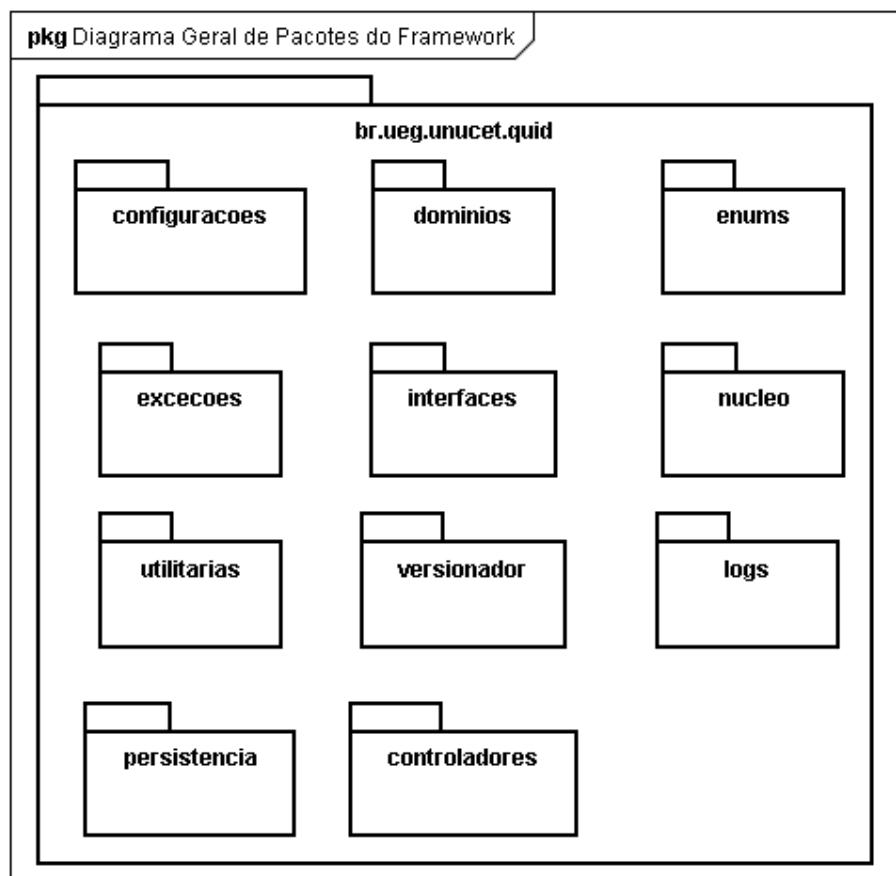


Figura 89 – Diagrama de pacotes do framework

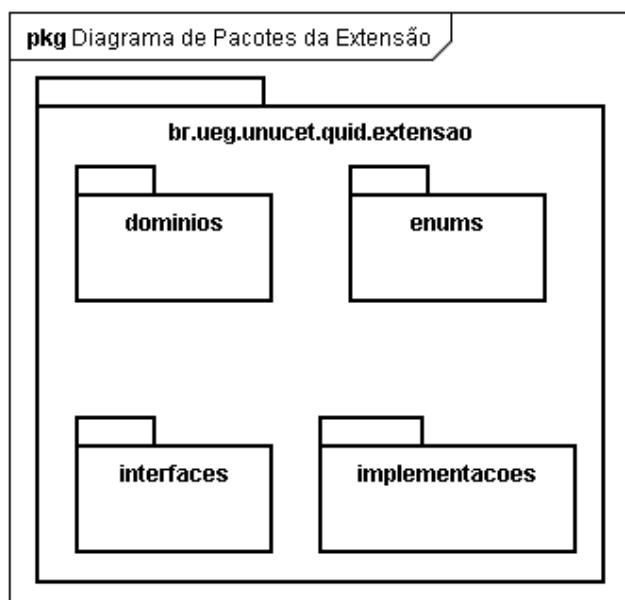


Figura 90 – Diagrama de pacotes da extensão

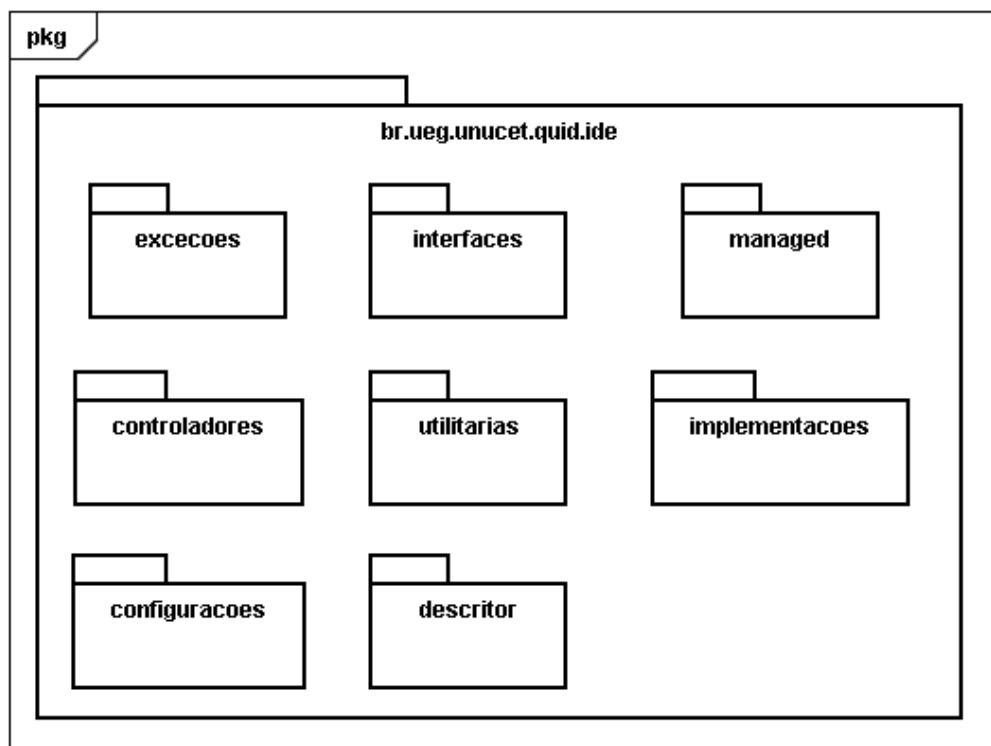


Figura 91 – Diagrama de pacotes da aplicação

8.6.1 Pacote Configurações

Possui os arquivos de configuração do funcionamento interno do framework e arquivos de mensagens padrões. Não se deve retirar nenhum registro do arquivo de mensagens, porém sua alteração é permitida.

8.6.2 Pacote Domínios

Possui todos os modelos das classes utilizadas pelo framework, ou seja, as classes relacionadas ao negócio. Exemplos: Modelo, Projeto, Retorno, Permissao.

8.6.3 Pacote Enums

Centraliza todos os Enums utilizados no framework.

8.6.4 Pacote Excecoes

Todas as exceções declaradas para o uso do framework são mantidas neste pacote. Elas são tratadas antes de chegarem a aplicação, para que as mensagens estejam padronizadas.

8.6.5 Pacote Interfaces

Possui todas as interfaces de uso do framework, inclusive as necessárias para a extensão do mesmo.

8.6.6 Pacote Núcleo

As classes aqui mantidas possuem o acoplamento das informações que envolvem as regras de negócio.

8.6.7 Pacote Utilitárias

Possui classes de auxílio a implementação. Exemplos: LeitoraJarUtil, LeitoraPropertiesUtil, StringUtil, ManipuladoraClasseUtil.

8.6.8 Pacote Versionador

Contém a implementação do controle de versionamento dos Serviços e TipoMembros.

8.6.9 Pacote Logs

Mantém as classes responsáveis pelo gerenciamento dos logs do framework.

8.6.10 Pacote Persistência

Possui as classes utilizadas na persistência de dados e estruturas do framework. Estas são acessadas pelo framework diretamente ou pelo Serviço de Persistência, se necessário.

8.6.11 Pacote Controladores

As classes mantidas nesse pacote resolveram as regras de negócio dos componentes do framework. Elas serão subdivididas de acordo com a responsabilidade dos modelos envolvidos no domínio do framework.

8.6.12 Pacote Managed

Contem todas os Managed Bean necessários para a implementação da visão.

8.6.13 Pacote Descritor

Possui as classes voltadas para a implementação do descritor de tela.

8.7 Padrões de Nomenclatura Utilizados na Codificação

Esta seção disponibiliza os padrões adotados para a nomenclatura das classes, métodos, enums, constantes e todos os recursos referentes a implementação do projeto. Seu objetivo é permitir um bom entendimento do código caso este projeto seja continuado posteriormente.

Antes dos padrões aqui expostos, foram adotados os padrões de nomenclatura da Oracle® como base para o desenvolvimento, por exemplo: classes só se iniciam com letra maiúscula, métodos se iniciam com letra minúscula, constantes se são escritas com todos os caracteres em maiúsculo, dentre outros.

8.7.1 Classes

- **Enum:** todos os enum's utilizados devem possuir o sufixo “**Enum**”. Exemplos: DominioEntradaEnum, MultiplicidadeEnum, StatusEnum.
- **Interfaces:** todas as interfaces devem conter o prefixo “**I**”. Exemplos: ITipoMembroModelo, ITipoMembroVisao, ITipoMembroServico.
- **ManagedBean:** todos os Managed Bean devem possuir o sufixo “**MB**”. Exemplos: UsuarioMB, ArtefatoMB, UploadMB.

- **Utilitárias:** todos as classes utilitárias deve possuir o sufixo “**Util**”. Exemplos: DataUtil, StringUtil, GenericoUtil.
- **Controle:** todas as classes que pertencem ao controle da aplicação devem ter o sufixo “**Controle**”. Exemplos: DesenvolvedorControle, PreenchedorControle, MontadorControle.

8.7.2 Métodos

- **Atribuição:** todos os métodos que tem a função de atribuir um valor a um atributo da classe devem conter o prefixo “**set**”. Exemplos: setTitulo, setDescricao, setPlataforma.
- **Obtenção:** todos os métodos que tem o objetivo de retornar o valor de um atributo da classe devem conter o prefixo “**get**”. Exemplos: getTitulo, getDescricao, getPlataforma.
- **Validação:** todos os métodos que tem o objetivo de validar algo, seja um atributo ou um valor de entrada, deve contar o prefixo “**is**”. Exemplos: isDataValida, isEntradaValida, isValorMaiorQueZero.
- **Ação:** todos os métodos criados nos Managed Bean que referenciam ações na tela devem conter o sufixo “**Action**”. Exemplos: salvarArtefatoAction, excluirMembroAction.
- **Inserir:** todos os métodos que passa a idéia de inserção devem conter o prefixo “**inserir**”. Exemplos: inserirArtefato, inserirTipoMembro, inserirMembro.
- **Alterar:** todos os métodos que implementem uma atualização de dados devem conter o prefixo “**alterar**”. Exemplos: alterarArtefato, alterarTipoMembro, alterarProjeto.
- **Remover:** todos os métodos que implementem uma remoção de dados devem conter o prefixo “**remover**”. Exemplos: removerMembro, removerArtefato, removerTipoMembro.
- **Pesquisar:** todos os métodos que retornam os elementos de uma entidade de acordo com os atributos preenchidos devem conter o prefixo “**pesquisar**”. Exemplo: pesquisarArtefato, pesquisarTipoMembro, pesquisarMembro.
- **Listar:** todos os métodos que retornam todos os elementos de uma entidade devem conter o prefixo “**listar**”. Exemplo: listarArtefato, listarTipoMembro, listarModelo.

8.8 Informações para Extensão do Framework

Esta seção é voltada para o usuário com perfil Desenvolvedor e descreve todas as informações necessárias para se criar novos componentes para o framework. Aqui estão descritas todas as interfaces que devem ser preenchidas para que o framework consiga reconhecer os arquivos importados como componentes compatíveis.

Todas as interfaces mencionadas nos tópicos abaixo estão localizadas no pacote: br.ueg.unucet.quid.interfaces. Para conhecimento do desenvolvedor, a funcionalidade de cada método da interface está disponível no Apêndice A, que é o javadoc das interfaces.

8.8.1 TipoMembro

Para a criação de novos componentes o usuário deve compreender que são necessários duas implementações: uma representando o TipoMembro Modelo e outra representando o TipoMembro Visão. Para estes as classes devem implementar as interfaces ITipoMembroModelo e

ITipoMembroVisao respectivamente. O framework só reconhecerá os componentes se estiverem implementando estas interfaces.

Para a criação de uma nova visão para um componente que já existe no framework deve-se somente criar um novo TipoMembro Visão respeitando a implementação das interfaces mencionado no parágrafo superior.

8.8.2 Serviço

Para a implementação de novas funcionalidades para os Artefatos, ou seja, ações a serem executadas a partir de um artefato, deve-se criar classes que implementem a interface IServico. Do mesmo modo que os TipoMembro, o framework só conseguirá reconhecer classes com os métodos descritos na interface IServico.

8.8.3 Arquivo .JAR

O padrão adotado para a criação dos arquivos para importação é o padrão da Oracle®. Só será permitido um componente por jar, e a classe que implementa a interface do componente, seja ITipoMembro ou IServico para novos TipoMembro e Servicos respectivamente, deve estar referenciada no arquivo .MANIFEST do jar.

O componente deve estar em um pacote com seu nome seguido de sua versão e revisão. Independente da quantidade de parâmetros que o componente possua, eles serão armazenados na base de dados como um vetor de bytes.

9 Realização de Casos de Uso

9.1 Diagrama Geral de Classes do Projeto

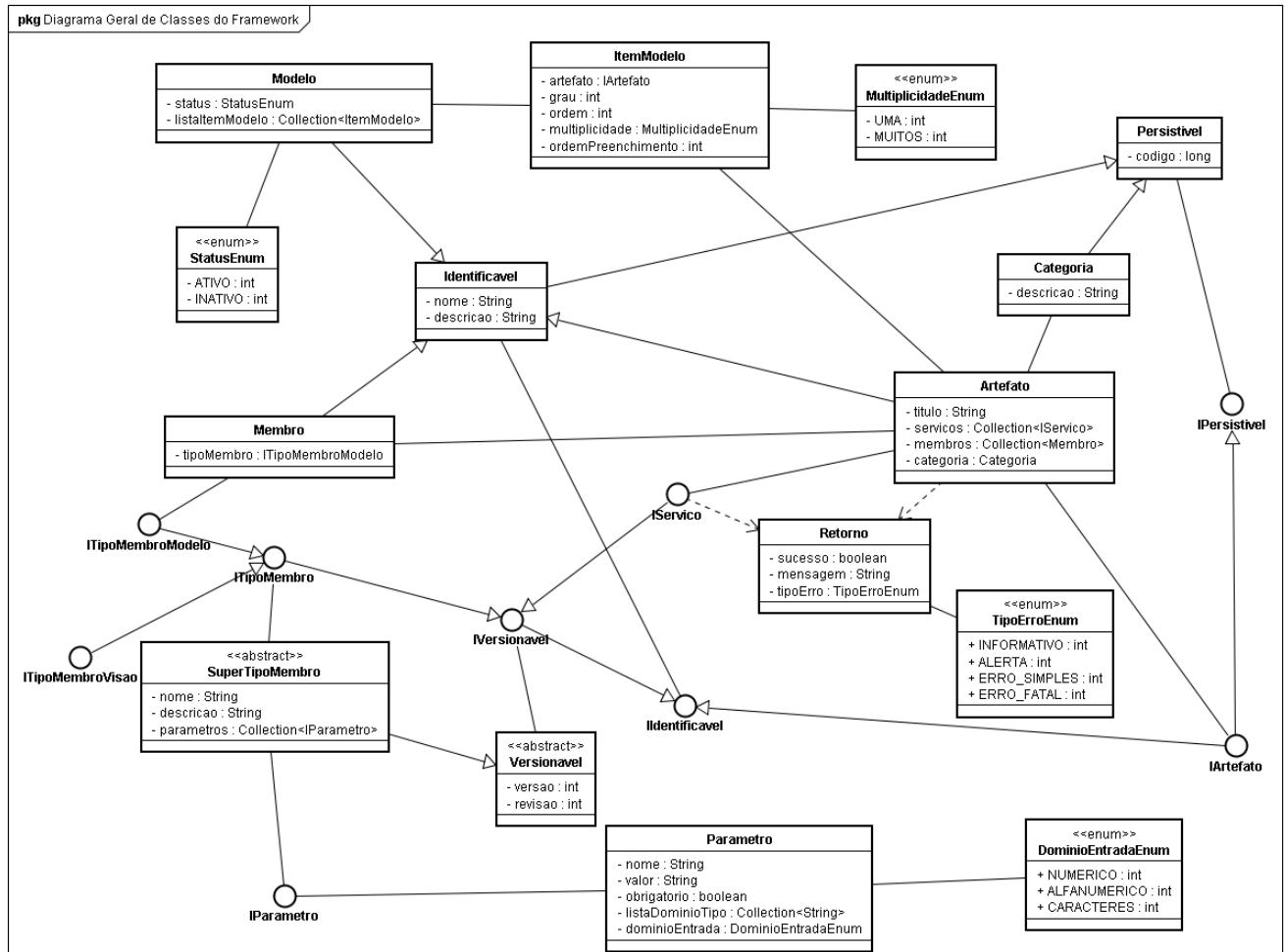


Figura 92 – Diagrama geral de classes do framework

10 Testes

10.1 Plano de Testes

10.2 Casos de Testes

10.2.1 Casos de Testes CDU01 - Manter Artefato

10.2.1.1 Cenário mapear artefato

Caso de Teste	Mapear um novo artefato correto para o framework.
Tipo de Teste	Caixa Preta
Procedimentos	
1.	Preenche informações referentes ao artefato a ser mapeado.
2.	Solicita mapeamento do artefato ao framework.
Resultado Esperado	Retorno de sucesso de execução.
Resultado do Teste	1. Resultado obtido 1.

Tabela 131 – Casos de Teste da funcionalidade 01

Caso de Teste	Mapear um novo artefato não informando o nome do artefato.
Tipo de Teste	Caixa Preta
Procedimentos	
1.	Preenche informações referentes ao artefato a ser mapeado, não informando o nome/descrição do artefato.
2.	Solicita mapeamento do artefato ao framework.
Resultado Esperado	Retorno de erro de execução, armazenando erro de atributos não preenchidos, e retornando a String "Nome", como atributo não preenchido.
Resultado do Teste	1. Resultado obtido 1.

Tabela 132 – Casos de Teste da funcionalidade 01

Caso de Teste	Mapear um novo artefato não informando a descrição do artefato.
Tipo de Teste	Caixa Preta
Procedimentos	
1.	Preenche informações referentes ao artefato a ser mapeado, não informando a descrição do artefato.
2.	Solicita mapeamento do artefato ao framework.
Resultado Esperado	Retorno de erro de execução, armazenando erro de atributos não preenchidos, e retornando a String "Descrição", como atributo não preenchido.

Resultado do Teste	1. Resultado obtido 1.
---------------------------	------------------------

Tabela 133 – Casos de Teste da funcionalidade 01

Caso de Teste	Mapear um novo artefato não informando o nome e nem a descrição do artefato.
Tipo de Teste	Caixa Preta
Procedimentos	
1.	Preenche informações referentes ao artefato a ser mapeado, não informando o nome e nem descrição do artefato.
2.	Solicita mapeamento do artefato ao framework.
Resultado Esperado	Retorno de erro de execução, armazenando erro de atributos não preenchidos, e retornando as Strings "Nome" e "Descrição", como atributos não preenchidos.
Resultado do Teste	1. Resultado obtido 1.

Tabela 134 – Casos de Teste da funcionalidade 01

Caso de Teste	Mapear um novo artefato com um nome de um artefato já cadastrado no framework.
Tipo de Teste	Caixa Preta
Procedimentos	
1.	Preenche informações referentes ao artefato a ser mapeado, informando um nome que já esteja cadastrado no framework.
2.	Solicita mapeamento do artefato ao framework.
Resultado Esperado	Retorno de erro de execução, armazenando erro de artefato duplicado no framework.
Resultado do Teste	1. Resultado obtido 1.

Tabela 135 – Casos de Teste da funcionalidade 01**10.2.1.2 Cenário pesquisar artefato.**

Caso de Teste	Realizar pesquisa de artefatos no framework informando filtros que retornem artefatos cadastrados.
Tipo de Teste	Caixa Preta
Procedimentos	

<ol style="list-style-type: none"> 1. Realizar pesquisa de artefato informando uma categoria um nome e uma descrição de um artefato cadastrado. 2. Solicita pesquisa de artefato. 	
Resultado Esperado	Retorno de sucesso de execução, nos parâmetros uma lista de artefatos encontrados com tais características.
Resultado do Teste	1. Resultado obtido 1.

Tabela 136 – Casos de Teste da funcionalidade 02

Caso de Teste	Realizar pesquisa de artefatos no framework informando filtros que retornem não retornem resultado na pesquisa.
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 1. Realizar pesquisa de artefato informando uma categoria um nome e uma descrição que não contenha em nenhum artefato cadastrado no framework. 2. Solicita pesquisa de artefato. 	
Resultado Esperado	Retorno de erro de execução, armazenando o erro de nenhum artefato encontrado, e nos parâmetros uma lista vazia de artefatos.
Resultado do Teste	1. Resultado obtido 1.

Tabela 137 – Casos de Teste da funcionalidade 02**10.2.1.3 Cenário remover artefato.**

Caso de Teste	Realizar remoção de um artefato que não esteja sendo utilizado por nenhum modelo cadastrado no framework.
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 1. Solicitar remoção de artefato que não esteja sendo utilizado por nenhum modelo cadastrado no framework. 	
Resultado Esperado	Retorno de sucesso de execução, removendo o artefato do framework.
Resultado do Teste	1. Resultado obtido 1.

Tabela 138 – Casos de Teste da funcionalidade 03

Caso de Teste	Realizar remoção de um artefato que esteja sendo utilizado em um modelo e em nenhum projeto do cadastrado no framework.
Tipo de Teste	Caixa Preta
Procedimentos	
	1. Solicitar remoção de artefato que esteja sendo utilizado em um modelo e em nenhum projeto cadastrado no framework.
Resultado Esperado	Retorno de erro de execução, armazenando o erro de falha de remoção, com o nome do modelo onde o artefato esta sendo utilizado.
Resultado do Teste	1. Resultado obtido 1.

Tabela 139 – Casos de Teste da funcionalidade 03

Caso de Teste	Realizar remoção de um artefato que esteja sendo utilizado em dois ou mais modelos e em nenhum projeto do cadastrado no framework.
Tipo de Teste	Caixa Preta
Procedimentos	
	1. Solicitar remoção de artefato que esteja sendo utilizado por dois ou mais modelos e em nenhum projeto cadastrado no framework.
Resultado Esperado	Retorno de erro de execução, armazenando o erro de falha de remoção, com os nomes dos modelos onde o artefato esta sendo utilizado.
Resultado do Teste	1. Resultado obtido 1.

Tabela 140 – Casos de Teste da funcionalidade 03

Caso de Teste	Realizar remoção de um artefato que esteja sendo utilizado em um ou mais projetos cadastrados no framework.
Tipo de Teste	Caixa Preta
Procedimentos	
	1. Solicitar remoção de artefato que não esteja sendo utilizado por um ou mais projetos cadastrados no framework.
Resultado Esperado	Retorno de erro de execução, armazenando o erro de falha de remoção, com os nomes dos projetos onde o artefato esta sendo utilizado.
Resultado do Teste	1. Resultado obtido 1.

Tabela 141 – Casos de Teste da funcionalidade 03**10.2.1.4 Cenário alterar artefato.**

Caso de Teste	Realizar alteração de um artefato que não esteja sendo utilizado em nenhum modelo cadastrado no framework.
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 1. Solicita pesquisa de artefatos. 2. Solicita alteração de um artefato. 3. Modifica as propriedades dos Membros do artefato. 4. Solicita mapeamento do artefato modificado. 	
Resultado Esperado	Retorno de sucesso de execução.
Resultado do Teste	1. Resultado obtido 1.

Tabela 142 – Casos de Teste da funcionalidade 04

Caso de Teste	Realizar alteração de um artefato que esteja sendo utilizado em um modelo e em nenhum projeto cadastrado no framework.
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 1. Solicita pesquisa de artefatos. 2. Solicita alteração de um artefato. 3. Modifica as propriedades dos Membros do artefato. 4. Solicita mapeamento do artefato modificado. 5. Confirma alteração de artefato, do modelo que ele está cadastrado. 	
Resultado Esperado	Retorno de sucesso de execução.
Resultado do Teste	1. Resultado obtido 1.

Tabela 143 – Casos de Teste da funcionalidade 04

Caso de Teste	Realizar alteração de um artefato que esteja sendo utilizado em dois ou mais modelos e em nenhum projeto cadastrado no framework.
----------------------	---

Tipo de Teste	Caixa Preta
Procedimentos	
1.	Solicita pesquisa de artefatos.
2.	Solicita alteração de um artefato.
3.	Modifica as propriedades dos Membros do artefato.
4.	Solicita mapeamento do artefato modificado.
5.	Seleciona modelo em que o artefato clonado será inserido e confirma clonagem.
Resultado Esperado	Retorno de sucesso de execução, com artefato clonado no documento selecionado.
Resultado do Teste	1. Resultado obtido 1.

Tabela 144 – Casos de Teste da funcionalidade 04

Caso de Teste	Realizar alteração de um artefato que esteja sendo utilizado em um ou mais projetos cadastrados no framework.
Tipo de Teste	Caixa Preta
Procedimentos	
1.	Solicita pesquisa de artefatos.
2.	Solicita alteração de um artefato.
3.	Modifica as propriedades dos Membros do artefato.
4.	Solicita mapeamento do artefato modificado.
5.	Confirma alteração de artefato, verificando os projetos onde ele está sendo utilizado.
Resultado Esperado	Retorno de sucesso de execução.
Resultado do Teste	1. Resultado obtido 1.

Tabela 145 – Casos de Teste da funcionalidade 04

Caso de Teste	Realizar alteração de um artefato, inserindo e removendo Membros e serviços, e o artefato esteja sendo utilizado em um ou mais projetos cadastrados no framework.
Tipo de Teste	Caixa Preta
Procedimentos	
1.	Solicita pesquisa de artefatos.

	<p>2. Solicita alteração de um artefato.</p> <p>3. Adiciona / Remove serviços e Membros do artefato.</p> <p>4. Solicita mapeamento do artefato modificado.</p> <p>5. Confirma clonagem do artefato, verificando o nome dos projetos onde o artefato está sendo utilizado.</p>
Resultado Esperado	Retorno de sucesso de execução, com clonagem do artefato.
Resultado do Teste	1. Resultado obtido 1.

Tabela 146 – Casos de Teste da funcionalidade 04**10.2.2 Casos de Testes CDU02 - Montar Artefato****10.2.2.1 Cenário mapear artefato**

Caso de Teste	Solicitar montagem de novo artefato.
Tipo de Teste	Caixa Preta
Procedimentos	
	3. Solicita criação de novo artefato.
Resultado Esperado	Retorno de sucesso de execução, com instância de um artefato em branco(Sem membros ou serviços inseridos nele).
Resultado do Teste	2. Resultado obtido 1.

Tabela 147 – Casos de Teste da funcionalidade 01

Caso de Teste	Solicitar salvar artefato montado.
Tipo de Teste	Caixa Preta
Procedimentos	
	<p>1. Preenche o nome e a descrição do artefato.</p> <p>2. Adiciona membros e serviços no artefato.</p> <p>3. Solicita salvar artefato.</p>
Resultado Esperado	Retorno de sucesso de execução, com artefato armazenado no framework.
Resultado do Teste	1. Resultado obtido 1.

Tabela 148 – Casos de Teste da funcionalidade 01**10.2.2.2 Cenário adicionar membro no artefato.**

Caso de Teste	Adicionar um membro no artefato.
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 1. Seleciona membro a ser inserido no artefato. 2. Posiciona o membro no artefato. 3. Preenche seus parâmetros. 4. Adiciona membro no artefato. 	
Resultado Esperado	Retorno de sucesso de execução, com membro adicionado no artefato.
Resultado do Teste	1. Resultado obtido 1.

Tabela 149 – Casos de Teste da funcionalidade 02

Caso de Teste	Adicionar um membro no artefato, informando parâmetros incorretos do membro.
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 1. Seleciona membro a ser inserido no artefato. 2. Posiciona ele no artefato. 3. Preenche os parâmetros do membro com valores inválidos. 4. Adiciona membro no artefato. 	
Resultado Esperado	Retorno de erro de parâmetros incorretos, informando os nomes dos parâmetros inválidos.
Resultado do Teste	1. Resultado obtido 1.

Tabela 150 – Casos de Teste da funcionalidade 02**10.2.2.3 Cenário remover membro do artefato.**

Caso de Teste	Remover membro do artefato.
----------------------	-----------------------------

Tipo de Teste	Caixa Preta
Procedimentos	
1.	Seleciona membro do artefato a ser removido.
2.	Solicita remoção do membro do artefato.

Tabela 151 – Casos de Teste da funcionalidade 03**10.2.2.4 Cenário adicionar serviço no artefato.**

Caso de Teste	Adicionar um serviço no artefato.
Tipo de Teste	Caixa Preta
Procedimentos	
1.	Seleciona serviço a ser inserido no artefato.
2.	Preenche lista de parâmetros do serviço.
3.	Adiciona serviço ao artefato.

Tabela 152 – Casos de Teste da funcionalidade 04

Caso de Teste	Adicionar um serviço no artefato com parâmetros incorretos.
Tipo de Teste	Caixa Preta
Procedimentos	
1.	Seleciona serviço a ser inserido no artefato.
2.	Preenche lista de parâmetros do serviço, informando parâmetros inválidos.
3.	Adiciona serviço ao artefato.

Tabela 153 – Casos de Teste da funcionalidade 04

Caso de Teste	Adicionar um serviço no artefato que contenha TipoMembro associado.
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 1. Seleciona serviço a ser inserido no artefato, que contenha TipoMembro associado. 2. Preenche lista de parâmetros do serviço, verificando se existe parametrização para o TipoMembro. 3. Posiciona TipoMembro no artefato. 4. Adiciona serviço no artefato. 	
Resultado Esperado	Retorno de sucesso de execução, com o serviço adicionado ao artefato.
Resultado do Teste	1. Resultado obtido 1.

Tabela 154 – Casos de Teste da funcionalidade 04**10.2.2.5 Cenário a remover serviço do artefato.**

Caso de Teste	Remover serviço do artefato.
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 1. Seleciona serviço a ser removido do artefato. 2. Solicitar remoção do serviço selecionado. 	
Resultado Esperado	Retorno de sucesso de execução, com o serviço removido do artefato.
Resultado do Teste	1. Resultado obtido 1.

Tabela 155 – Casos de Teste da funcionalidade 05

Caso de Teste	Remover serviço do artefato que não esteja servindo de parametrização de outro serviço.
Tipo de Teste	Caixa Preta

Procedimentos	
1. Seleciona serviço a ser removido do artefato. 2. Solicitar remoção do serviço selecionado.	
Resultado Esperado	Retorno de sucesso de execução, com o serviço removido do artefato.
Resultado do Teste	1. Resultado obtido 1.

Tabela 156 – Casos de Teste da funcionalidade 05

Caso de Teste	Remover serviço do artefato tenha TipoMembro associado.
Tipo de Teste	Caixa Preta
Procedimentos	
1. Seleciona serviço a ser removido do artefato. 2. Solicitar remoção do serviço selecionado.	
Resultado Esperado	Retorno de sucesso de execução, com o serviço e TipoMembro removidos do artefato.
Resultado do Teste	1. Resultado obtido 1.

Tabela 157 – Casos de Teste da funcionalidade 05**10.2.2.6 Cenário visualizar artefato**

Caso de Teste	Visualizar artefato.
Tipo de Teste	Caixa Preta
Procedimentos	
1. Solicitar visualização do artefato que está sendo montado.	
Resultado Esperado	Retorno de sucesso de execução, com interface de preenchimento do artefato.
Resultado do Teste	1. Resultado obtido 1.

Tabela 158 – Casos de Teste da funcionalidade 06**10.2.3 Casos de Testes CDU01 - Manter Artefato**

10.2.3.1 Cenário clonar artefato

Caso de Teste	Clonar um artefato cadastrado no framework.
Tipo de Teste	Caixa Preta
Procedimentos	
	4. Solicitar clonagem de um artefato, informando o novo nome e a nova descrição do artefato clonado.
Resultado Esperado	Retorno de sucesso de execução, com artefato clonado no framework. .
Resultado do Teste	3. Resultado obtido 1.

Tabela 159 – Casos de Teste da funcionalidade 01

10.2.4 Casos de Testes CDU04 - Preencher Artefato

10.2.4.1 Montar Interface de preenchimento

Caso de Teste	Solicitar interface de preenchimento para um artefato já preenchido em um projeto.
Tipo de Teste	Caixa Preta
Procedimentos	
	5. Solicitar interface de preenchimento do artefato, informando o projeto em que ele está sendo preenchido e a tecnologia de visão.
Resultado Esperado	Retorno de sucesso de execução, armazenado a interface de preenchimento do artefato, com as interfaces de visualização do membro, preenchido as informações que continham no artefato, bloqueando-o no projeto.
Resultado do Teste	4. Resultado obtido 1.

Tabela 160 – Casos de Teste da funcionalidade 01

Caso de Teste	Solicitar interface de preenchimento para um artefato que não foi preenchido no projeto.
Tipo de Teste	Caixa Preta
Procedimentos	
	1. Solicitar interface de preenchimento do artefato, informando um projeto, e um artefato que não está sendo utilizado em um projeto.

Resultado Esperado	Retorno de sucesso de execução, armazenado a interface de preenchimento do artefato, com as interfaces de visualização dos membros, bloqueando o artefato no projeto.
Resultado do Teste	1. Resultado obtido 1.

Tabela 161 – Casos de Teste da funcionalidade 01

Caso de Teste	Solicitar interface de preenchimento para um artefato não informando um projeto.
Tipo de Teste	Caixa Preta
Procedimentos	
1. Solicitar interface de preenchimento do artefato, não informando um projeto.	
Resultado Esperado	Retorno de sucesso de execução, com interface de preenchimento do artefato, com os membros no estado de preenchimento.
Resultado do Teste	1. Resultado obtido 1.

Tabela 162 – Casos de Teste da funcionalidade 01**10.2.4.2 Interface de preenchimento do membro**

Caso de Teste	Solicitar interface de preenchimento para um membro.
Tipo de Teste	Caixa Preta
Procedimentos	
1. Solicitar interface de preenchimento do membro no preenchimento do artefato.	
Resultado Esperado	Retorno de sucesso de execução, com a interface de preenchimento do membro.
Resultado do Teste	1. Resultado obtido 1.

Tabela 163 – Casos de Teste da funcionalidade 02**10.2.4.3 Armazenamento do valor informado ao membro.**

Caso de Teste	Verificação de armazenamento do valor preenchido no membro
Tipo de Teste	Caixa Preta
Procedimentos	
1. Solicitar preenchimento de um artefato.	

	<p>2. Solicitar interface de preenchimento do membro.</p> <p>3. Preencher um valor no membro.</p>
Resultado Esperado	Retorno de sucesso de execução, com o membro na interface de visualização, preenchido com o valor informado.
Resultado do Teste	1. Resultado obtido 1.

Tabela 164 – Casos de Teste da funcionalidade 03

Caso de Teste	Verificação de validação de valor incorreto informado ao membro.
Tipo de Teste	Caixa Preta
Procedimentos	
	<p>1. Solicitar preenchimento de um artefato.</p> <p>2. Solicitar interface de preenchimento do membro.</p> <p>3. Preencher um valor incorreto no membro.</p>
Resultado Esperado	Retorno de erro de execução, com o erro de valor incorreto.
Resultado do Teste	1. Resultado obtido 1.

Tabela 165 – Casos de Teste da funcionalidade 03**10.2.4.4 Cancelar preenchimento do artefato**

Caso de Teste	Solicitar cancelamento de preenchimento de artefato.
Tipo de Teste	Caixa Preta
Procedimentos	
	<p>1. Solicitar preenchimento de artefato.</p> <p>2. Solicitar cancelamento de preenchimento do artefato.</p>
Resultado Esperado	Retorno de sucesso de execução, com o desbloqueio do artefato no projeto.
Resultado do Teste	1. Resultado obtido 1.

Tabela 166 – Casos de Teste da funcionalidade 04

10.2.5 Casos de Testes CDU05 - Visualizar Artefato

10.2.5.1 Cenário visualizar artefato

Caso de Teste	Solicitar interface de visualização do artefato preenchido.
Tipo de Teste	Caixa Preta
Procedimentos	
6.	Solicitar interface de visualização do artefato, informando uma instância do artefato preenchido.
Resultado Esperado	Retorno de sucesso de execução, com interface de visualização dos membros tendo os valores preenchidos.
Resultado do Teste	5. Resultado obtido 1.

Tabela 167 – Casos de Teste da funcionalidade 01

Caso de Teste	Solicitar interface de visualização do artefato não preenchido.
Tipo de Teste	Caixa Preta
Procedimentos	
1.	Solicitar interface de visualização do artefato, informando uma instância do artefato não preenchido.
Resultado Esperado	Retorno de sucesso de execução, com interface de preenchimento dos membros do artefato.
Resultado do Teste	1. Resultado obtido 1.

Tabela 168 – Casos de Teste da funcionalidade 01

10.2.6 Casos de Testes CDU06 - Publicar Artefato

10.2.6.1 Cenário publicar artefato

Caso de Teste	Solicitar publicação de um artefato, informando o serviço de publicação
Tipo de Teste	Caixa Preta
Procedimentos	
7.	Solicita publicação de um artefato.
8.	Seleciona serviço de publicação

9. Confirma publicação de artefato.	
Resultado Esperado	Retorno de sucesso de execução, armazenando o arquivo gerado pelo serviço de publicação.
Resultado do Teste	6. Resultado obtido 1.

Tabela 169 – Casos de Teste da funcionalidade 01

Caso de Teste	Solicitar publicação de um artefato, não informando o serviço de publicação
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 1. Solicita publicação de um artefato. 2. Não seleciona serviço de publicação. 3. Confirma publicação de artefato. 	
Resultado Esperado	Retorno de sucesso de execução, armazenando a mensagem de cancelamento de publicação de artefato.
Resultado do Teste	1. Resultado obtido 1.

Tabela 170 – Casos de Teste da funcionalidade 01**10.2.7 Casos de Testes CDU07 - Manter Membro****10.2.7.1 Cenário mapear membro**

Caso de Teste	Mapear um novo membro correto para o framework
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 10. Selecionar TipoMembro que será criado o membro. 11. Instanciar o membro, e preencher os parâmetros do membro. 12. Solicita mapeamento do membro. 	
Resultado Esperado	Retorno de sucesso de execução, castrando o membro no framework.
Resultado do Teste	7. Resultado obtido 1.

Tabela 171 – Casos de Teste da funcionalidade 01

Caso de Teste	Mapear um novo membro para o framework não informando o nome e a descrição do membro.
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 1. Selecionar TipoMembro que será criado o membro. 2. Instanciar o membro, e preencher os parâmetros do membro, não informando o nome e a descrição do membro. 3. Solicita mapeamento do membro. 	
Resultado Esperado	Retorno de erro de falha de execução, informando as Strings "Nome" e "Descrição", nos parâmetros do retorno.
Resultado do Teste	1. Resultado obtido 1.

Tabela 172 – Casos de Teste da funcionalidade 01

Caso de Teste	Mapear um novo membro para o framework, não informando os parâmetros do membro.
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 1. Selecionar TipoMembro que será criado o membro. 2. Instanciar o membro e não preencher os parâmetros. 3. Solicita mapeamento do membro. 	
Resultado Esperado	Retorno de erro de falha de execução, com os nomes dos parâmetros obrigatórios não informados.
Resultado do Teste	1. Resultado obtido 1.

Tabela 173 – Casos de Teste da funcionalidade 01**10.2.7.2 Pesquisar membro**

Caso de Teste	Pesquisar membros cadastrados no framework informando TipoMembro.
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 1. Pesquisar membros cadastrados informando um TipoMembro. 	

Resultado Esperado	Retorno de sucesso de execução e nos parâmetros do resultado uma lista que contenha membros do TipoMembro informado.
Resultado do Teste	1. Resultado obtido 1.

Tabela 174 – Casos de Teste da funcionalidade 02

Caso de Teste	Pesquisar membros cadastrados no framework informando o nome do membro.
Tipo de Teste	Caixa Preta
Procedimentos	
1. Pesquisar membros cadastrados informando um nome de um membro.	
Resultado Esperado	Retorno de sucesso de execução e nos parâmetros do resultado uma lista que contenha membros que contenham o nome informando.
Resultado do Teste	1. Resultado obtido 1.

Tabela 175 – Casos de Teste da funcionalidade 02

Caso de Teste	Pesquisar membros cadastrados no framework informando um nome e um TipoMembro que não possua membro cadastrado.
Tipo de Teste	Caixa Preta
Procedimentos	
1. Pesquisar membros cadastrados informando um nome e um TipoMembro que não possua membro cadastrado.	
Resultado Esperado	Retorno de erro de nenhum membro encontrado, e uma lista vazia no parâmetro do retorno.
Resultado do Teste	1. Resultado obtido 1.

Tabela 176 – Casos de Teste da funcionalidade 02**10.2.7.3 Alterar Membro**

Caso de Teste	Alterar membro cadastrado no framework, que esteja sendo utilizado por apena um artefato
Tipo de Teste	Caixa Preta
Procedimentos	
1. Alterar os parâmetros de um membro na montagem de um artefato.	

2. Solicita alteração de membro.	
Resultado Esperado	Alteração do membro no framework, e retorno da instância do membro.
Resultado do Teste	1. Resultado obtido 1.

Tabela 177 – Casos de Teste da funcionalidade 02

Caso de Teste	Alterar membro cadastrado no framework, que esteja sendo utilizado por dois ou mais artefatos.
Tipo de Teste	Caixa Preta
Procedimentos	
1. Alterar os parâmetros de um membro na montagem de um artefato.	
2. Solicita alteração de membro.	
Resultado Esperado	Clonagem do membro, retornando instância do membro sem o nome e a descrição do membro informadas.
Resultado do Teste	1. Resultado obtido 1.

Tabela 178 – Casos de Teste da funcionalidade 02

10.2.8 Casos de Testes CDU08 - Manter TipoMembro

10.2.8.1 Cenário mapear TipoMembro

Caso de Teste	Mapear um novo TipoMembro correto para o framework.
Tipo de Teste	Caixa Preta
Procedimentos	
13. Informar local do arquivo onde se encontra o jar do TipoMembro.	
14. Solicita mapeamento do TipoMembro.	
Resultado Esperado	Retorno de sucesso de execução, armazenando o jar no local específico.
Resultado do Teste	8. Resultado obtido 1.

Tabela 179 – Casos de Teste da funcionalidade 01

Caso de Teste	Verificação de formando correto do jar do TipoMembro, no momento do mapeamento.
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 1. Informar local do arquivo onde se encontra o jar do TipoMembro. (O jar deverá conter um erro em seu formato ou conteúdo) 2. Solicita mapeamento do TipoMembro. 	
Resultado Esperado	Retorno de erro de formato incorreto, informando o nome do arquivo jar.
Resultado do Teste	1. Resultado obtido 1.

Tabela 180 – Casos de Teste da funcionalidade 01

Caso de Teste	Mapear um novo TipoMembro correto para o framework, que seja uma nova versão.
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 1. Informar local do arquivo onde se encontra o jar do TipoMembro que é uma nova versão. 2. Solicita mapeamento do TipoMembro. 	
Resultado Esperado	Retorno de sucesso de execução, armazenando o jar no local específico.
Resultado do Teste	1. Resultado obtido 1.

Tabela 181 – Casos de Teste da funcionalidade 01

Caso de Teste	Mapear um novo TipoMembro correto para o framework, que seja uma nova revisão.
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 1. Informar local do arquivo onde se encontra o jar do TipoMembro que é uma nova revisão. 2. Solicita mapeamento do TipoMembro. 	
Resultado Esperado	Retorno de sucesso de execução, armazenando o jar no local específico e armazenando a versão antiga em outro diretório.
Resultado do Teste	1. Resultado obtido 1.

Tabela 182 – Casos de Teste da funcionalidade 01**10.2.9 Casos de Testes CDU09 - Manter Membro****10.2.9.1 Cenário montar modelo**

Caso de Teste	Mapear um novo modelo para o framework.
Tipo de Teste	Caixa Preta
Procedimentos	
15. Solicitar a criação de um modelo informando seu nome e sua descrição. 16. Solicita mapeamento do modelo para o framework. 17. Adiciona artefatos dentro do modelo. 18. Solicita mapeamento do modelo	
Resultado Esperado	Retorno de sucesso de execução, com o modelo mapeado no framework.
Resultado do Teste	9. Resultado obtido 1.

Tabela 183 – Casos de Teste da funcionalidade 01

Caso de Teste	Verificar duplicidade de cadastro de modelo no framework.
Tipo de Teste	Caixa Preta
Procedimentos	
1. Solicitar a criação de um modelo informando um nome de um modelo cadastrado no framework. 2. Solicita mapeamento do modelo para o framework.	
Resultado Esperado	Retorno de erro de modelo já cadastrado no framework.
Resultado do Teste	1. Resultado obtido 1.

Tabela 184 – Casos de Teste da funcionalidade 01**10.2.9.2 Cenário adicionar artefato modelo**

Caso de Teste	Adicionar um novo artefato no modelo.
Tipo de Teste	Caixa Preta

Procedimentos	
1. Seleciona artefato a ser inserido no modelo. 2. Insere artefato no modelo. 3. Informa os parâmetros de inserção do artefato no modelo. 4. Confirma inclusão de artefato no modelo.	
Resultado Esperado	Retorno de sucesso de execução, com o artefato inserido no modelo.
Resultado do Teste	1. Resultado obtido 1.

Tabela 185 – Casos de Teste da funcionalidade 02

Caso de Teste	Verificação de parâmetros inválidos/obrigatórios na inserção do artefato no modelo.
Tipo de Teste	Caixa Preta
Procedimentos	
1. Seleciona artefato a ser inserido no modelo. 2. Insere artefato no modelo. 3. Não informa os parâmetros de inserção do artefato no modelo. 4. Confirma inclusão de artefato no modelo.	
Resultado Esperado	Retorno de erro de parâmetros inválidos/ obrigatórios , com os nomes dos parâmetros inválidos/obrigatórios.
Resultado do Teste	1. Resultado obtido 1.

Tabela 186 – Casos de Teste da funcionalidade 02**10.2.9.3 Cenário remover artefato modelo**

Caso de Teste	Remover um artefato do modelo
Tipo de Teste	Caixa Preta
Procedimentos	
1. Seleciona artefato do modelo. 2. Solicita remoção do artefato do modelo.	

Resultado Esperado	Retorno de sucesso de execução, com artefato removido do modelo e ordem e grau dos outros artefatos do modelo reorganizados.
Resultado do Teste	1. Resultado obtido 1.

Tabela 187 – Casos de Teste da funcionalidade 03**10.2.9.4 Cenário alterar parâmetros do artefato modelo**

Caso de Teste	Remover um artefato do modelo
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 1. Seleciona artefato do modelo. 2. Altera parâmetros do artefato no modelo. 3. Confirma alteração dos parâmetros. 	
Resultado Esperado	Retorno de sucesso de execução, com alteração dos parâmetros do artefato no modelo.
Resultado do Teste	1. Resultado obtido 1.

Tabela 188 – Casos de Teste da funcionalidade 04

10.2.9.5 Cenário alterar parâmetros do artefato modelo

Caso de Teste	Verificação de parâmetros inválidos / obrigatórios na alteração dos parâmetros do artefato no modelo.
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 1. Seleciona artefato do modelo. 2. Altera parâmetros do artefato informando parâmetros inválidos e deixando parâmetros obrigatórios não preenchidos. 3. Confirma alteração dos parâmetros. 	
Resultado Esperado	Retorno de erro de parâmetros inválidos, com os nomes do parâmetros inválidos / não informados.
Resultado do Teste	1. Resultado obtido 1.

Tabela 189 – Casos de Teste da funcionalidade 05

10.2.9.6 Mapear modelo

Caso de Teste	Verificação de artefatos dependentes foram inseridos no modelo.
Tipo de Teste	Caixa Preta
Procedimentos	
<ol style="list-style-type: none"> 1. Realiza montagem do modelo. 2. Não inclui todos os artefatos dependentes no modelo. 3. Solicita mapeamento do modelo. 	
Resultado Esperado	Retorno de erro de dependência de artefato, com os nomes dos artefatos dependentes que não foram inseridos no modelo.
Resultado do Teste	1. Resultado obtido 1.

Tabela 190 – Casos de Teste da funcionalidade 06

10.2.9.7 Alterar modelo

Caso de Teste	Solicitar alteração de modelo no framework.
Tipo de Teste	Caixa Preta
Procedimentos	

1. Pesquisar modelo no framework.	
2. Solicitar alteração de modelo no framework.	
Resultado Esperado	Retorno para interface de preenchimento do modelo.
Resultado do Teste	1. Resultado obtido 1.

Tabela 191 – Casos de Teste da funcionalidade 07

10.3 Relatório de Testes

Nenhum teste realizado até o momento.

11 Conclusão

O uso de modelos (templates) para se documentar projetos é muito usado na atualidade. Muitas normas de qualidade (ISO) exigem um certo nível de documentação para garantir a qualidade da empresa ou projeto. Portanto, é de grande importância ter um total controle sobre as informações geradas no decorrer da vida de um projeto, tanto no armazenamento de novas informações quanto nas alterações necessárias.

Visto que a idéia apresentada neste projeto não possui correspondentes no mercado, tivemos certa dificuldade no levantamento dos requisitos, tomando grande parte do tempo que nos foi cedido. Mesmo não implementando todas as funcionalidades aqui documentadas, nos certificamos de aplicar nossos esforços para que o resultado saísse com qualidade e que este projeto possa ser facilmente entendido, continuado e evoluído.

É fato que o mercado necessita de um software com as características descritas aqui, porém a idéia ainda precisa ser um pouco amadurecida para que o projeto possa conquistar um lugar de destaque e a confiança de clientes e parceiros.

12 Referências

VALERIANO, D.L. **Gerência em Projetos: pesquisa, desenvolvimento e engenharia.** São Paulo, 1998. Markron Books, 1998.

SOUZA, Rafael De Paula, **Facelets**, DevMedia, 2007. <http://www.devmedia.com.br/post-5332-Facelets.html> acessado em: 07/11/2011 às 16:36.

13 Apêndices

Apêndice I - Cronograma de Atividades do Trabalho de Conclusão de Curso

Atividade	Previsão	Entrega
Entrega do Termo de Aceite ao coordenador do TC	11/03/2011	10/03/2011
Entrega do projeto de TC	08/04/2011	08/04/2011
Entrega do(s) produto(s) desenvolvidos para a banca de avaliação parcial	03/06/2011	03/06/2011
Apresentação do projeto e do(s) produto(s) elaborado(s) bem como o andamento do trabalho	18/06/2011	18/06/2011
Entrega do pôster	16/09/2011	16/09/2011
Exposição e comunicação do pôster	28/09/2011	28/09/2011
Entrega do TC a ser apresentado para banca de avaliação final	08/11/2011	08/11/2011
Apresentação do TC para banca de avaliação final - banca de defesa	26/11/2011	26/11/2011
Entrega das correções (recomendadas pela banca de avaliação final) ao orientador	06/12/2011	06/12/2011
Entrega do trabalho de conclusão encadernado ao coordenador do TC	16/12/2011	12/12/2011

Tabela 192 – Cronograma de atividades proposto no projeto do Trabalho de Conclusão de Curso

Apêndice II - Pôster Apresentado no III Simpósio de Tecnologia da Informação e III Semana de Iniciação Científica do Curso de Sistemas de Informação UnUCET-UEG/2011

Universidade Estadual de Goiás
Unidade Universitária de Ciências Exatas e Tecnológicas
Sistemas de Informação

 Johnys Custódio da Silva Rabelo
 johnys@gmail.com
 Leonardo Eustáquio Pessoa de Oliveira
 nardopessoa@gmail.com
 Orientador: Prof. Guiliano Rangel
 guiliano@gmail.com



QUID-IDE: FRAMEWORK DE DOCUMENTAÇÃO

Introdução

Ninguém nunca deve ter pensado que uma documentação de baixa qualidade fosse positiva, mas até onde isso pode prejudicar? Se um projeto for de qualidade, mas tiver erros em sua documentação ele pode ser mantido e evoluir? Na maioria das vezes não, mesmo porque, se o objetivo é a qualidade, o cuidado em cada detalhe é, sem dúvida, essencial!

Mas o que é documentação?

Documentação engloba a coleção de vários artefatos (figuras, textos, gráficos, etc.) que se relacionam a fim de servir como elemento de comunicação durante um projeto e auxiliar para o entendimento das atividades que estão ocorrendo.



Para que serve a documentação?

A documentação serve como um repositório de informações de todo o processo de desenvolvimento. Com ela é possível verificar pontos vulneráveis, aprofundar em uma análise de riscos, realizar o gerenciamento das atividades e da equipe que compõem o projeto.



Problemas referentes a documentação

Hoje em dia, existe uma grande dificuldade por parte das pessoas em cumprirem as metas de prazo e orçamento. A documentação é vista como perda de tempo e facilmente deixada de lado, pois a pressão de curtos prazos e custos define a velocidade de construção do produto e atrasos não são vistos com bons olhos para os clientes.

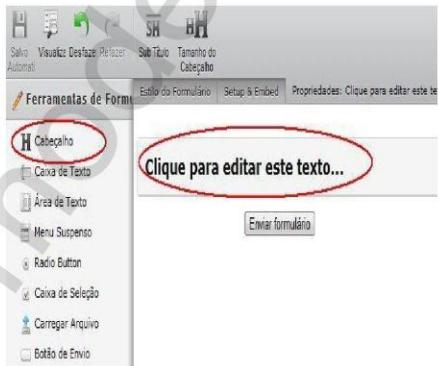


Problemas oriundos da complexidade do documento (excesso de informações repetitivas), da estrutura do documento (alinhamento de texto, fonte, indexação dos capítulos, padrões ABNT), e a ligação de partes dependentes contribuem para que a documentação seja vista como algo que "atrasa" o desenvolvimento de um projeto.

Uma ferramenta que focasse a atenção dos analistas no conteúdo do documento, deixando as partes de formatação, indexação, dependência e que gerenciasse a equipe de preenchimento do projeto auxiliaria na redução do tempo gasto no desenvolvimento da documentação do projeto.

Solução proposta

O projeto em questão tem como objetivo a criação de um framework que almeja a flexibilidade em relação aos tipos de documentos tratáveis, ou seja, qualquer documento estruturado que pode ser baseado em um modelo (template). Com ele é possível a criação das diversas partes do template, através de uma interface intuitiva, criação do template, linkando as partes dependentes, e controle do processo de preenchimento e impressão do trabalho final.



Com isso pretendemos alcançar uma qualidade esperável da documentação, pois foca a atenção do preenchedor no conteúdo do documento, deixando as demais configurações para o framework.

Porque um framework e não uma ferramenta?

Nosso trabalho se define como um framework pois possui as seguintes características:

- Extensibilidade: Permite a criação de novos componentes, que agreguem novas funcionalidades ao framework.
- Reaproveitamento: O framework também poderá ser utilizado para a confecção de outros documentos e formulários de cadastros simples, porém esta visão não está dentro do escopo.

Público alvo

Devido a grande instabilidade e constante evolução da documentação de software, este projeto está voltado para ser utilizado nas faculdades, principalmente na disciplina de projeto de graduação, onde é obrigatório a documentação.

Referências

VALERIANO, D.L. **Gerência em Projetos: pesquisa, desenvolvimento e engenharia**. São Paulo, 1998. Markron Books, 1998.

Figura 93 – Pôster Quid-IDE: Framework de Documentação

Apendice III – Javadoc e API

O arquivo referente ao link abaixo está disponibilizado em uma pasta *javadoc* junto a este arquivo e tanto a pasta quanto este arquivo devem ser descompactados na mesma.

[javadoc\index.html](#)

14 Anexos

Nenhum anexo foi necessário para a confecção deste projeto.