

CENTRO UNIVERSITÁRIO DO NORTE - UNINORTE  
CURSO DE TECNOLOGIA EM DESENVOLVIMENTO DE SOFTWARE  
RELATÓRIO FINAL DO TRABALHO DE CONCLUSÃO DE CURSO

Graduação em Tecnologia em Desenvolvimento de Software

**GerPro – Sistema de Geração e Apoio à Correção de PESw**

por

Manfrine Silva Santos  
Manoel da Silva Neto  
Marcelo Silva de Souza  
Rilmar Pereira Gomes

Manaus – AM  
Julho de 2009

**CENTRO UNIVERSITÁRIO DO NORTE - UNINORTE**  
**CURSO DE TECNOLOGIA EM DESENVOLVIMENTO DE SOFTWARE**  
**RELATÓRIO FINAL DO TRABALHO DE CONCLUSÃO DE CURSO**

**MANFRINE SILVA SANTOS**  
**MANOEL SILVA NETO**  
**MARCELO SILVA DE SOUZA**  
**RILMAR PEREIRA GOMES**

**GerPro – Sistema de Geração e Apoio à Correção de PESw**

Este trabalho foi apresentado à Graduação do Curso de Tecnologia em Desenvolvimento de Software do Centro Universitário do Norte como requisito parcial para obtenção do grau de Tecnólogo em Desenvolvimento de Software.

**ORIENTADOR: Prof. MSc. Ricardo da Silva Barboza**

Manaus – AM  
Julho de 2009

TRABALHO DE CONCLUSÃO DE CURSO  
CURSO SUPERIOR DE TECNOLOGIA EM DESENVOLVIMENTO DE SOFTWARE

**TERMO DE APROVAÇÃO**

O Trabalho de Conclusão de Curso “**Sistema de Geração e Apoio a Correção de PESw**”, elaborado pelos alunos **Manfrine Silva Santos, Manoel da Silva Neto, Marcelo Silva de Souza e Rilmar Pereira Gomes**, foi julgado adequado por todos os membros da Banca Examinadora, para a obtenção do grau de **Tecnólogo em Desenvolvimento de Software** e aprovado, em sua forma final, pelo Curso Superior de Tecnologia em Desenvolvimento de Software do Centro Universitário do Norte.

Manaus, 27 de julho de 2009.

---

Prof. Ricardo da Silva Barboza, Msc.

**Orientador do TCC**

Banca Examinadora integrada pelos seguintes professores:

---

Prof. Ricardo da Silva Barboza, MSc.

---

Prof. Anderson Fernandes Esteves, MSc.

Dedico este trabalho à minha família,  
minha esposa, meus amigos,  
à minha cidade natal Tapiramutá  
e todas as pessoas que, direta ou indiretamente,  
contribuíram para a realização desse sonho.  
MANFRINE SILVA SANTOS

À minha família.  
MANOEL DA SILVA NETO

Dedico este trabalho à Deus, à meu filho, meus irmãos,  
à minha mãe, D. Gercina, exemplo de garra e determinação  
e à minha esposa, Marcela, fonte de inspiração.  
MARCELO SILVA DE SOUZA

Ao generoso Deus.  
À minha filha.  
Aos meus pais.  
À minha futura esposa, Thainá Oliveira,  
que sempre acreditou na minha capacidade,  
e a todos que contribuíram para realização desse sonho  
RILMAR PEREIRA GOMES

## **AGRADECIMENTOS**

*Agradecemos primeiramente a Deus por nos proporcionar saúde e força, também a todas as pessoas que direta e indiretamente sempre nos apoiaram e acreditaram que nosso esforço transformaria um sonho em realidade.*

*Gostaríamos de agradecer a todos os professores do curso, pois são eles os nossos grandes exemplos de vitória, em especial a professora Áurea Hiléia, Professor Anderson Esteves e Professor Emmerson Santa Rita, por nos incentivar e nos ajudar na detecção de nossos talentos.*

## **RESUMO**

Esta dissertação relata sobre o processo de desenvolvimento do GerPro – Sistema de Geração e Apoio à Correção de PESw. O GerPro atua na criação e correção de propostas de especificação de Software, visando automatizar parte destes processos. Este aplicativo se mostrou de grande ajuda aos envolvidos na criação e avaliação deste documento, submetido pelos alunos finalistas do Curso de Desenvolvimento de Software do Centro Universitário do Norte.

## **ABSTRACT**

This dissertation reports about the GerPro development process - PESw Generation and Support Correction System. The GerPro acts in creation and correction of software specification proposal, aiming to automate part of these processes. This application proved great help to those involved in the creation and evaluation of this document, sub-ceilings for final year students of the Software Development Course of the Centro Universitário do Norte.

## ÍNDICE

<b>1. INTRODUÇÃO .....</b>	<b>1</b>
1.1. OBJETIVOS .....	2
1.2. JUSTIFICATIVA .....	2
1.3. ORGANIZAÇÃO DO TRABALHO .....	4
<b>2. REVISÃO BIBLIOGRÁFICA.....</b>	<b>5</b>
2.1 METODOLOGIA.....	5
2.2 MODELAGEM .....	6
2.3. FERRAMENTAS .....	7
2.3.1. Banco de Dados .....	7
2.3.2. MySQL .....	8
2.3.3. SQL.....	9
2.3.4. Apache Tomcat.....	9
2.3.5. Hibernate (HIBERNATE.ORG).....	9
2.3.6. Ferramenta de desenvolvimento Eclipse .....	10
2.3.7. UML – Unified Modeling Language (Linguagem de Modelagem Unificada).....	10
2.3.8. Versionamento.....	11
2.3.9. Internet e World Wide Web.....	12
2.3.10. Padrões de Projeto .....	13
2.3.11. Java.....	15
2.3.12. JEE – Java Web .....	15
2.3.13. JSF – Java Server Faces.....	16
2.3.14. Orientação a Objeto .....	17
2.3.15. Ireport.....	17
2.3.16. Jude .....	18
<b>3. ANÁLISE.....</b>	<b>19</b>
3.1 DESCRIÇÃO DO SISTEMA.....	19
3.2 LISTA DE FUNÇÕES .....	23
3.3 DIAGRAMA DE CASOS DE USO .....	24
3.4 DESCRIÇÃO DE CASOS DE USO.....	25
3.4.1. Caso de Uso Manter Proposta .....	25



3.4.2.	Caso de Uso Alocar Propostas .....	26
3.4.3.	Caso de uso Submeter Correção .....	29
3.4.4.	Caso de Uso Gerar Gráfico de Porcentagem de Propostas Aprovadas, Reprovadas e Aprovadas com Ressalvas. ....	32
3.4.5.	Caso de Uso Gerar Relatório de Equipes e Propostas.....	34
3.5	DIAGRAMAS .....	35
3.5.1.	Classes.....	35
3.5.2.	Seqüência Alocar Proposta .....	36
3.5.3.	Seqüência Submeter Correção .....	37
3.5.4.	Seqüência Cadastrar Proposta.....	38
3.5.5.	Seqüência Editar Proposta.....	39
3.5.6.	Seqüência Pesquisar Proposta .....	40
3.5.7.	Seqüência Gerar Relatório de Equipe e Proposta .....	41
3.5.8.	Seqüência Gerar Relatório de Porcentagem de Aprovadas, Reprovadas e Aprovadas com Ressalva .....	42
3.5.9.	Atividade cadastrar proposta.....	43
3.5.10.	Atividade Gerar Relatório Equipe x Proposta .....	44
3.5.11.	Estado Status do Item .....	45
3.5.12.	Estado Status da Proposta .....	46
3.5.13.	Entidade- Relacionamento .....	47
4.	PROJETO.....	48
4.1	CASOS DE USO .....	49
4.1.1.	Diagrama de Casos de Uso .....	49
4.2.	DIAGRAMAS .....	61
4.2.1.	Classes persistentes.....	62
4.2.2.	Classes de Fachada e Dao.....	63
4.2.3.	Classes de Processamento .....	64
4.2.4.	Classes de Fronteira .....	65
4.2.5.	Classe de Relatório .....	66
4.2.6.	Classes – Da Arquitetura do Sistema.....	67
4.2.7.	Seqüência Cadastrar Proposta.....	68
4.2.8.	Seqüência Pesquisar Proposta .....	69
4.2.9.	Seqüência Editar Proposta.....	70
4.2.10.	Seqüência Alocar Proposta .....	71
4.2.11.	Seqüência Submeter Correção .....	72

4.2.12.	Seqüência Relatório Gráfico de Porcentagem de propostas aprovadas, reprovadas e aprovadas com ressalvas .....	73
4.2.13.	Seqüência Relatório Equipe e Proposta.....	74
4.2.14.	Atividade – Cadastrar Proposta .....	75
4.2.15.	Atividade Gerar Relatório de Equipe e Propostas .....	76
4.2.16.	Estado – Status do Item.....	77
4.2.17.	Estado – Status da Proposta.....	78
4.2.18.	Diagrama de Entidade e Relacionamento – Modelo Físico .....	79
4.3.	PROJETO DE BANCO DE DADOS .....	80
4.3.1.	Dicionário de Dados .....	80
4.4.	Mapeamento de Dados .....	86
4.5.	PROJETO DE INTERFACE.....	89
4.5.1.	Interfaces do Tipo Entrada de Dados – Registrar Dados.....	89
4.5.2.	Interface do Tipo Processamento.....	91
4.5.3.	Interface do Tipo Busca Dados. ....	95
4.5.3.1.	Interface Pesquisar Propostas (Pesquisa). ....	95
4.5.4.	Interfaces do Tipo Saída de Dados – Relatórios. ....	97
4.5.5.	Interfaces do Tipo Visualização de Relatórios.....	99
4.6.	Aspectos Ergonômicos.....	100
4.6.1.	Comunicabilidade.....	101
4.7.	Projeto Navegacional .....	105
5.	CONSIDERAÇÕES FINAIS .....	106
6.	REFERÊNCIAS BIBLIOGRÁFICAS .....	108
	ANEXOS.....	112

## LISTA DE FIGURAS

Figura 1 - Diagrama de Casos de Uso .....	24
Figura 2 - Diagrama de Classes de Análise .....	35
Figura 3 - Diagrama de Seqüência Alocar Proposta .....	36
Figura 4 - Diagrama de Seqüência Calcular Status.....	37
Figura 5 - Diagrama de Seqüência Cadastrar Proposta .....	38
Figura 6 - Diagrama de Seqüência Alterar Proposta.....	39
Figura 7 - Diagrama de Seqüência Pesquisar Proposta .....	40
Figura 8 - Diagrama de Seqüência Relatório Proposta x Equipe.....	41
Figura 9 - Diagrama de Seqüência Relatório de Porcentagem .....	42
Figura 10 - Diagrama de Atividade Cadastrar Proposta.....	43
Figura 11 - Diagrama de Atividade Gerar Relatório.....	44
Figura 12 - Diagrama de Estado do Status do Item.....	45
Figura 13 - Diagrama de Estado do Status da Proposta .....	46
Figura 14 - Diagrama de Entidade e Relacionamento.....	47
Figura 15 - Diagrama de Casos de Uso de Projeto .....	49
Figura 16 - Diagrama de Classe de Projeto .....	62
Figura 17 - Diagrama de Classe de Fachada e Dao.....	63
Figura 18 - Classe de Processamento .....	64
Figura 19 - Classes de Fronteira.....	65
Figura 20 - Classe de Relatório.....	66
Figura 21 - Arquitetura do Sistema .....	67
Figura 22 - Diagrama de Seqüência Cadastrar Proposta (Projeto) .....	68
Figura 23 - Diagrama de Seqüência Pesquisar Proposta (Projeto) .....	69
Figura 24 - Diagrama de Seqüência Editar Proposta (Projeto).....	70
Figura 25 - Diagrama de Seqüência Alocar Proposta (Projeto).....	71
Figura 26 - Diagrama de Seqüência Submeter Correção (Projeto).....	72
Figura 27 - Diagrama de Seqüência do Relatório de Porcentagem (Projeto).....	73
Figura 28 - Diagrama de Seqüência do Relatório de Equipe X Proposta (Projeto).....	74
Figura 29 - Diagrama de Atividade Cadastrar Proposta (Projeto) .....	75
Figura 30 - Diagrama de Atividade Gerar Relatório de Equipe X Proposta (Projeto).....	76

Figura 31 - Diagrama de Estados do Status do Item (Projeto) .....	77
Figura 32 - Diagrama de Estados do Status da Proposta (Projeto) .....	78
Figura 33 - Diagrama de Entidade e Relacionamento – Modelo Físico.....	79
Figura 34 - Tela de Cadastro de Proposta.....	89
Figura 35 - Tela de Alocar Proposta .....	91
Figura 36 - Tela de Submeter Correção.....	93
Figura 37 - Tela de Pesquisar Proposta .....	95
Figura 38 - Tela para Chamada de Relatórios .....	97
Figura 39 - Tela de Relatório .....	99
Figura 40 - Tela de Login Mensagem de Erro .....	102
Figura 41 - Tela de Alocar Proposta .....	103
Figura 42 - Tela de Cadastro de Propostas .....	104
Figura 43 - Projeto Navegacional.....	105

## **1. INTRODUÇÃO**

A comScore, empresa de pesquisa e marketing para Web, divulgou que a quantidade de usuários de internet ultrapassou a marca de 1 bilhão em Janeiro deste ano. Isso implica, entre outras coisas, em dizer que existe uma mudança de hábitos das pessoas e que o uso cada vez mais intenso da internet indica uma busca por maior praticidade em resolver problemas, buscar informações e se relacionar com outras pessoas. Assim, buscando sempre pelo dinamismo oferecido pela internet, mais e mais aplicações são criadas ou alteradas para se adequarem a esse novo ambiente, trazendo consigo novas tecnologias e aprimoramentos de seu uso.

Praticidade tem muito a ver com a internet. A facilidade de pagar contas só precisando estar conectado saiu do que era ficção e tornou-se realidade até mesmo nos lugares mais distantes da Terra. Distantes mas não mais isolados. Pessoas compartilham fotos, imagens, sonhos. Falam outras línguas. Conhecem outros lugares. Assim é a Internet. Há espaço para tudo.

Buscando também esta praticidade que nós, alunos do 6º período do Curso de Desenvolvimento de Software do Centro Universitário do Norte - UNINORTE, nos propomos a criar o GerPro - um sistema que tem por finalidade proporcionar maior praticidade no processo de correção de um importante documento no mundo do desenvolvimento de sistemas: a Proposta de Especificação de Software.

## 1.1. OBJETIVOS

No processo de desenvolvimento de software PRAXIS, a Proposta de Especificação de Software – PESw - é o primeiro artefato a ser homologado pelo cliente para tenha início o desenvolvimento do software em questão. Este artefato é alvo de avaliação pelos professores do curso de Tecnologia de Desenvolvimento de Software do Centro Universitário do Norte, como parte do Trabalho de Conclusão de Curso I.

O processo de produção e correção da PESw é feito de forma manual e tende a atrasos devido à grande quantidade de propostas submetidas para correção pelos alunos finalistas. Por este motivo, o GerPro tem por objetivo geral auxiliar os alunos finalistas e professores no processo de geração e correção de PESw.

O sistema aqui proposto poderá ser de grande ajuda à alunos e professores, dinamizando o processo de produção e correção de PESw e ainda reduzindo a necessidade de impressão deste documento.

## 1.2. JUSTIFICATIVA

O Uninorte possui, entre seus cursos da área de Tecnologia, a graduação em Tecnologia de Desenvolvimento de Software. Esse curso oferece ao seu aluno a capacitação para atuar na área de desenvolvimento de sistemas, desde sua análise até sua implantação, passando por todas as fases compreendidas no processo de desenvolvimento de software. O curso tem duração de seis períodos, sendo dois por ano, englobando as disciplinas mais relevantes para que o aluno receba instrução suficiente para se adequar ao mercado de trabalho.

No processo de formação do profissional, é utilizada pelo corpo docente uma grande variedade de avaliações do conhecimento do aluno. Entre tais avaliações existe o Trabalho de Conclusão de Curso – TCC. Elaborado com o objetivo de oportunizar ao aluno aplicar e aprimorar o conhecimento adquirido ao longo do curso, o TCC é realizado a partir do 5º

período do curso e está dividido em duas fases: TCC I, que inclui a fase de Concepção e a parte inicial da fase de Elaboração e a Parte II onde as fases de Elaboração e Construção são concluídas. Ambas definidas pelo Processo para Aplicação eXtensíveis InterativoS - Praxis.

No decorrer do TCC, os alunos devem produzir artefatos para compor a documentação do processo de desenvolvimento de software. Esses artefatos podem ser relatórios, diagramas e/ou documentos exigidos para uma melhor compreensão do produto que se deseja produzir. Entre esses artefatos, a Proposta de Especificação de Software – PESw – é o primeiro documento a ser produzido com base no levantamento de requisitos realizado junto ao cliente. Esse documento visa especificar as funcionalidades do software em questão. Nele estão incluídas informações sobre os objetivos que o software terá no cenário descrito pelo cliente, a motivação para a produção do mesmo, os métodos que serão utilizados, assim como as ferramentas para o apoio à produção. Além dessas informações, há também uma estimativa de duração de cada fase do processo de desenvolvimento.

A PESw é baseada em vários modelos de proposta, sendo que o Curso de Desenvolvimento de Software da Uninorte utiliza apenas dois: o Modelo de Proposta Típica, utilizado em trabalhos específicos de desenvolvimento e o Modelo de Proposta Atípica ou Especial, para trabalhos com fundamentação teórica. Para cada modelo existe um conjunto de perguntas para sua avaliação. Essas perguntas são objetivas e relacionadas a uma parte da PESw ou ao seu todo. Finalizada a avaliação, a proposta recebe um Status com base na combinação das respostas dadas a cada pergunta de avaliação. Esses status são: Proposta Aprovada, Proposta Aprovada com Ressalva e Proposta Reprovada.

O processo de avaliação da PESw é realizado de forma manual desde sua produção até a avaliação pelos professores. Com isso, a duração de cada fase aumenta, podendo ocasionar atrasos na entrega, correção e lançamento de notas. Enfim, prejudicar o cronograma e o calendário disciplinar.

Com base neste contexto, é proposta deste trabalho a criação de um sistema que atue como apoio no processo de geração e correção de propostas de especificação de software, facilitando assim as atividades relacionadas a este processo. Para que isso seja possível, metas menores devem ser alcançadas, entre elas a persistência dos dados dos usuários do sistema, a visualização e impressão de propostas e geração de relatórios, além da correção de proposta com base na pergunta.

### 1.3. ORGANIZAÇÃO DO TRABALHO

Este trabalho é composto deste capítulo introdutório e de outros itens, conforme detalhamento feito a seguir:

No capítulo 2 – Apresentação da metodologia aplicada no processo de desenvolvimento, a modelagem do sistema e as ferramentas utilizadas no processo.

No capítulo 3 – Apresentação do conteúdo produzido na fase de análise (TCC I).

No capítulo 4 - Apresentação do conteúdo produzido na fase de projeto (TCC II)

Finalizando, no capítulo 5 - Apresentação de conclusões obtidas no decorrer do processo de elaboração do sistema e do aprendizado adquirido.



## 2. REVISÃO BIBLIOGRÁFICA

Neste capítulo serão apresentadas: a metodologia aplicada ao desenvolvimento do sistema GerPro, a sua modelagem e uma visão geral das ferramentas utilizadas no decorrer do processo de desenvolvimento.

### 2.1 METODOLOGIA

As aplicações web têm características em seu funcionamento que as diferenciam de outros tipos de aplicações. A arquitetura cliente-servidor é evidenciada em aplicações desenvolvidas para a internet pois esta deve acomodar uma estrutura leve para o cliente, baseada em requisições e que se comporte de maneira dinâmica e segura neste ambiente.

Esta arquitetura foi escolhida para o GerPro, seguindo orientações do cliente, visto que este desejava mobilidade, flexibilidade e disponibilidade do aplicativo fora do ambiente acadêmico.

Nesse processo de desenvolvimento, utilizou-se algumas técnicas do processo *eXtreme Process* - XP, como a codificação em par, e do próprio Praxis, devido ao conteúdo ter sido ministrado em ocasiões anteriores neste curso de tecnologia. Assim, mantivemos a base do Praxis do lado da documentação e algumas práticas do XP do lado da codificação.

Para a modelagem das funcionalidades, iterações e estados da aplicação, utilizamos a Linguagem de Modelagem Unificada - UML - devido ao seu alcance e padronização reconhecidas no Paradigma de Orientação à Objetos.

## 2.2 MODELAGEM

A extração de requisitos realizada por meio de entrevista com o cliente forneceu diretrizes para o início do processo de desenvolvimento. A partir deste ponto, todos os artefatos necessários para a documentação e implementação do sistema foram criados, baseando em diagramas e descrições UML.

No início da década de 90, surgiram muitas propostas de técnicas para modelagem de sistemas segundo o paradigma de orientação à objetos. Nesse período, era comum o fato de duas técnicas possuírem diferentes notações gráficas para modelar uma mesma perspectiva de um sistema. Ao mesmo tempo, cada técnica tinha seus pontos fortes e fracos. Neste cenário surgiu a UML. Proposta pelos pesquisadores Grady Booch, James Rumbaugh e Ivar Jacobson, a UML inicial aproveitou as melhores características de cada proposta preexistente e principalmente das técnicas propostas pelos próprios “três amigos”, como ficaram conhecidos. Em 1997, UML foi aceita pelo OMG como padrão de modelagem. Desde seu surgimento, recebeu atualizações para torná-la mais clara e útil e neste momento está na versão 2.0.

Um processo de desenvolvimento de software que utilize a UML padrão de modelagem envolve a criação de diversos documentos que pode ser textuais ou gráficos. Assim, os artefatos, como são conhecidos tais documentos, são classificados pelo tipo e ajudam e compõe as visões do sistema. Segundo [BEZ07], estas visões são assim definidas:

- Visão de Casos de Uso: descreve o sistema do ponto de vista externo como um conjunto de interações entre o sistema e os agentes externos ao sistema. Esta visão é criada inicialmente e direciona o desenvolvimento das demais visões do sistema.
- Visão de Projeto: enfatiza as características do sistema que dão suporte tanto estrutural quanto comportamental, às funcionalidades externamente visíveis ao sistema.

- Visão de Implementação: abrange o gerenciamento de versões do sistema, construídas pelo agrupamento de módulos (componentes) e subsistemas.
- Visão de Implantação: corresponde à distribuição física do sistema em seus subsistemas e à conexão entre essas partes.
- Visão de Processo: esta visão enfatiza as características de concorrência (paralelismo), sincronização e desempenho do sistema.

A implementação ou das visões depende da complexidade do software a ser construído.

Os artefatos da UML estão os diagramas. Esses são assim definidos:

Diagramas UML:

- Estruturais:
  - de Objetos, de Estrutura Composta, de Classes e de Pacotes;
  - Estruturais de Implementação:
    - de Componentes e de Implantação;
- Comportamentais:
  - de Atividades, de Casos de Uso e de Transição de Estado;
  - Comportamentais de Interação:
    - de Seqüência, de Temporização e de Visão Geral de Interação;

## 2.3. FERRAMENTAS

### 2.3.1. Banco de Dados

[EN05A] afirma que:

Os bancos de dados e a sua tecnologia estão provocando um grande impacto no uso de computadores. É viável afirmar que eles representam um papel crítico em que os computadores são utilizados, incluindo negócios, comércio eletrônico, engenharia, medicina, direito, educação e as ciências da informação,

para citar apenas algumas delas. A palavra banco de banco é tão comumente utilizada que, primeiro, devemos defini-la. Nossa definição inicial é bastante genérica.

Os bancos de dados estão inseridos no dia-dia das pessoas, armazenando suas informações, que podemos afirmar que a maioria das informações gerenciais ou simplesmente informativas está armazenada em bancos de dados. Os bancos de dados se tornaram ferramentas fundamentais para a sobrevivência das pessoas que necessitam de informações para o seu trabalho quaisquer que seja ele.

Um banco de dados é uma coleção de dados relacionados”. Os bancos de dados não são necessariamente um SGBD (Sistema de Gerenciamento de Banco de Dados), mas sim qualquer forma de armazenamento de informações. “Esses dados podem ter sido escritos em uma agenda de telefones ou armazenados em um computador, por meio de programas como o Microsoft Access ou Excel. [EN05B].

### 2.3.2. MySQL

[MYS09A] – “O programa MySQL (R) é um servidor robusto de bancos de dados SQL (*Structured Query Language* – Linguagem Estruturada para Pesquisas) muito rápido, multitarefa e multi-usuário.”

O SGBD MySQL é um sistema free, pode ser adquirido gratuitamente no site [mysql.com](http://mysql.com). O MySQL possui milhares de usuários, por ele ser um sistema que oferece uma boa confiabilidade aos usuários e atende a os mais diferenciados tipos de sistemas.

[MYS09B] diz que,

“O programa MySQL é de Licença Dupla. Os usuários podem escolher entre usar o programa MySQL como um produto Open Source/Free Software sob os termos da GNU General Public License (<http://www.fsf.org/licenses/>)”.

### 2.3.3. SQL

[OLI02] afirma que:

SQL (*Structured Query Language*) é um conjunto de comandos de manipulação de banco de dados utilizado para criar e manter a estrutura desse banco de dados, além de incluir modificar e pesquisar informações nas tabelas dele.

O SQL é a linguagem de manipulação de banco de dados mais conhecida e mais utilizada por todos os SGBDs comerciais existentes no mercado. É uma linguagem poderosa e que a partir dela surgiram outras linguagens específicas para manipulação de SGBDs tipo Oracle que utiliza a PL/SQL dentre outros.

### 2.3.4. Apache Tomcat

Segundo [HEM07], “o Tomcat é talvez o mais popular servidor Web baseado em Java e contêiner de servlet. Ele é um contêiner de servlets relativamente leve que cresceu em popularidade nos últimos anos.”

O Tomcat pode ser adquirido gratuitamente, ele é muito popular entre os desenvolvedores de software, que trabalham com tecnologias free. O Tomcat é um sistema de armazenagem de Servlets. [COU02] afirma que: “O Tomcat é um Servlet Container, ou seja, é um servidor onde são instaladas Servlets para tratar as requisições que o servidor receber”.

### 2.3.5. Hibernate (HIBERNATE.ORG)

O Hibernate é um framework (conjunto de código que tem como função, conectar aplicações ou até mesmo tecnologias diferentes), que faz um mapeamento do banco de dados e é utilizado para apoiar e simplificar a comunicação da base de dados com os códigos de manipulação de sistemas no desenvolvimento de sistemas.

### 2.3.6. Ferramenta de desenvolvimento Eclipse

Segundo [HEM07A], “O Eclipse é uma das melhores coisas que aconteceu ao Java nos últimos anos. Em minha opinião Java ganhou uma vida longa como uma tecnologia dominante”.

O Eclipse é uma ferramenta de desenvolvimento de fácil manipulação, e compreensão, como também muito utilizada principalmente por profissionais desenvolvedores Java. Ela pode ser adquirida gratuitamente no site eclipse.org.

O Eclipse, por ser uma ferramenta livre e por ter uma grande quantidade de desenvolvedores que trabalham no seu desenvolvimento e melhoramento, possui inúmeros plugins (programas que funcionam anexados a outros programas), que facilitam a vida dos desenvolvedores, necessitando assim de apenas um programa com várias funcionalidades.

[HEM07B] afirma que:

As ferramentas Java do Eclipse incluem a formatação de fonte, geração de executáveis (*building*), depuração de integração com Ant. Entretanto, há literalmente centenas de plug-ins gratuitos e comerciais disponíveis para o Eclipse. De diagramas UML às ferramentas de banco de dados se houver demanda por alguma funcionalidade, é provável que você encontre um plug-in apropriado.

### 2.3.7. UML – Unified Modeling Language (Linguagem de Modelagem Unificada)

A linguagem UML, permite aos usuários visualizarem um sistema de forma gráfica, expressa através dos seus vários diagramas. Ela é apenas uma linguagem, independente de processo e seu foco é voltado para a representação conceitual e física de um sistema. “A UML não é uma linguagem de programação e sim uma linguagem de modelagem, cujo objetivo é auxiliar os engenheiros de software a definir características do software.” [GUE08]

[DEI05A] afirma que, “A *Unified Modeling Language* é agora o esquema de representação gráfica mais amplamente utilizado para modelar sistemas orientado a objetos”. E define uma notação padrão que pode ser utilizada por desenvolvedores de software

orientados a objetos. A orientação a objetos hoje em dia é um dos muitos conceitos de programação existente, veio no intuito de melhorar e agilizar o método procedural.

A UML é aplicada através dos seus diagramas, que têm a função de modelar cada perspectiva do sistema, tornando um meio de visualização do sistema em sua toda totalidade. A versão 2.0 é composta por treze diagramas, divididos em Diagramas Estruturais e Diagramas Comportamentais.

“O objetivo é fornecer múltiplas visões do sistema a ser modelado, analisando-o e modelando sob diversos aspectos, procurando-se, assim, atingir a completitude da modelagem, permitindo que cada diagrama complemente os outros.” [GUE08].

Os Estruturais são: Diagrama de Objetos, Diagrama de Estrutura Composta, Diagrama de Classes, Diagrama de Pacotes, Diagrama de Implantação que são dois: Diagrama de Componentes, Diagrama de Implantação.

Os Comportamentais são: Diagrama Atividades, Diagrama de Casos de Uso, Diagrama de Transição e Estados. Diagrama de Interação que são divididos em quatro: Diagrama de Seqüência, Diagrama de Colaboração, Diagrama de Temporização e Diagrama de Visão Geral da Interação.

Seguindo um padrão de desenvolvimento, um sistema tem a extração de requisitos, análise, implementação, implantação e testes. A UML também se adéqua a cada ciclo do software, na análise temos as modelagem à nível de análise, os pretende-se saber o que o sistema de fazer. E na implementação ou codificação e implantação por diante, temos as modelagens em nível de projeto que se pretende mostrar, como deve ser feito ou como foi implementado respectiva função.

### **2.3.8. Versionamento**

[LOC09] afirma que:

“Uma coisa não é mais sequer discutível: todo projeto de software, independente do tamanho, precisa de um sistema de versionamento de código. Isso é imperativo. Mesmo com apenas um desenvolvedor, controlar as revisões, versões e modificações no projeto não é algo que se possa desconsiderar.”

O hoje a questão do versionamento não se restringe ao software, mas sim a todos os ramais de um projeto, como a documentação. O versionamento cria um somatório de vantagens no desenvolvimento de um projeto. Por exemplo um mesmo sistema pode ser manipulado por várias pessoas ao mesmo tempo, tornando sua produção mais rápida, produtiva e segura, produtiva pelo fato de várias pessoas trabalhando diretamente numa mesma funcionalidade, a mesma está concluída em menos tempo e segura pelo fato de todos os envolvidos saberem qual usuário manipulou qual parte do código.

### **2.3.9. Internet e World Wide Web**

[DEI05B] diz que:

“[...] Originalmente projetada para conectar os principais sistemas de computadores de cerca de uma dúzia de universidades e organizações de pesquisas, a Internet atualmente é acessível a centenas de milhões de computadores no mundo.”

A internet hoje é um meio de comunicação tão popular quanto o rádio e a televisão, vem se expandindo a aumentando cada vez mais o número de usuários navegando nessa tecnologia. Ela nos traz um volume de informação nunca visto antes e de forma tão rápida, que não era de se imaginar tamanho aumento e transformação há tempos atrás.

[DEI05C] “Com o surgimento da World Wide Web (www) – os usuários de computador podem localizar e visualizar documentos baseados em multimídia sobre qualquer assunto pela Internet.”

A WWW permitiu com sua tecnologia integrar os vários tipos de mídias disponíveis na web, tornando-se assim uma tecnologia multimídia.



### 2.3.10. Padrões de Projeto

Padrões de projetos de software descrevem soluções para problemas que ocorrem repetidamente em diversos sistemas. Essas soluções são descritas de forma que possam ser reutilizadas em vários sistemas diferentes.

Conforme afirma [MAR09A],

Padrões de projeto podem ser vistos como uma solução que já foi testada para um problema. Desta forma, um padrão de projeto geralmente descreve uma solução ou uma instância da solução que foi utilizada para resolver um problema específico. Padrões de projetos são soluções para problemas que alguém um dia teve e resolveu aplicando um modelo que foi documentado e que você pode adaptar integralmente ou de acordo com necessidade de sua solução.

#### 2.3.10.1. Mvc

MVC (vem do inglês, *Control – View – Model*) em português Controlador – Visão - Modelo. É um *Design Partner*, traduzindo é um Padrão de Projeto, [GON07] afirma que “MVC é um conceito de desenvolvimento e design que tenta separar uma aplicação em três partes distintas”.

[MAR09B] diz, “A arquitetura MVC - (Modelo Visualização Controle) fornece uma maneira de dividir a funcionalidade envolvida na manutenção e apresentação dos dados de uma aplicação”. Com a MVC o aplicativo fica dividido em camada, permitindo uma melhor manutenibilidade, uma melhor leitura e entendimento do código do aplicativo. As atualizações em aplicativos que implementa este padrão fica mais independente, pois é separada a camada de regras de negócio, da camada armazenamento e a camada de visão também é separada, isso dá pra atualizar constantemente as interfaces sem modificar a funcionalidade principal do sistema por exemplo.

A arquitetura MVC foi desenvolvida para ser usada no projeto interface visual em Smalltalk. [PER08] “O MVC foi um dos primeiros padrões identificados, surgiu na

comunidade de Smalltalk em 1979 e após 23 anos de sua criação (em 2003) ainda é um pattern aplicável”.

Mas ele pode ser usado em qualquer aplicação interativa.

Com o padrão MVC torna possível o desenvolvimento em paralelo das três camadas, ou seja, uma equipe desenvolve o layout de telas, camadas view ou de visão, ou projeta o banco de dados, camada de modelo, e outra equipe desenvolve a camada de controle, as regras de negócio, tornando assim mais prático, fácil, dinâmico e produtivo.

#### 2.3.10.2. Padrão Facade

[MAR09C] diz que: “O padrão Facade é um padrão estrutural que tem como objetivo fornecer um conjunto de interfaces em um sistema é utilizado para reduzir a complexidade de um sistema”.

Com o padrão Facade permite-se com que o sistema tenha um conceito de baixo acoplamento tornando-o assim, mais portátil, independente de tecnologia e lógica utilizada.

[BEZ07A] afirma que, ”a solução fornecida por este padrão é a seguinte: cria uma fachada para o subsistema fornecedor, de tal forma que o subsistema cliente se comunique com o primeiro por intermédio desta fachada.” O Facade é utilizado como um comunicador que recebe uma ordem em que é repassada para quem vai executar a ordem, o qual é encapsulado, por tanto o requisitante da ordem não precisa saber como será executado a ordem, mas sim que ela será atendida.

#### 2.3.10.3. Padrão DAO

Segundo [CAR09],

devido à sua qualidade de tradutor, o DAO abstrai a origem e o modo de obtenção / gravação dos dados, de modo que o restante do sistema manipula os dados de forma transparente, sem se preocupar com o que acontece por trás dos panos. Isso ajuda muito em processos de migrações de fonte de dados e testes unitários.

O Padrão Dao (*Data Access Objects*) como o nome diz, é um objeto de acesso a dados, ele tem a função de isolar a camada de acesso a dados, facilitando o reuso e a manutenção.

[BEZ07B] – “O Padrão DAO é uma forma de desacoplar as classes do negócio os aspectos relativos ao acesso ao(s) mecanismos de armazenamento persistente [...]”.

### **2.3.11. Java**

Segundo [JAV09A],

Java é uma tecnologia. É um mundo tão grande que ninguém se arrisca a dizer: ? Eu sei Java ?. Basicamente constitui-se de uma linguagem de programação e um programa para execução chamado de máquina virtual ou virtual machine. Quando programa-se em Java usa-se a linguagem de programação Java e um ambiente de desenvolvimento Java para gerar um software que será executado em um ambiente de distribuição Java. Tudo isso é a tecnologia Java.

A Linguagem Java se popularizou por ser livre, de acesso fácil aos desenvolvedores que buscam uma linguagem de alto poder, de confiança, que torna o código mais seguro. O Java é utilizado por muitos desenvolvedores, o que torna uma aplicação feita em Java fácil de manter, realizar atualizações e manutenções em seu código.

O Java é uma linguagem orientada a objetos que falaremos mais na próxima seção, tem em sua abrangência para plataformas Java Desktop J2SE, Java WEB JEE, Java para dispositivos embaçados e dispositivos móveis J2ME.

### **2.3.12. JEE – Java Web**

[INF09] diz que:

O JEE (*Java Enterprise Edition*) é a plataforma Java voltada para redes, internet, intranets e afins. Assim, ela contém bibliotecas especialmente desenvolvidas para o acesso a servidores, a sistemas de e-mail, a banco de dados, etc. Por essas características, o JEE foi desenvolvido para suportar uma grande quantidade de usuários simultâneos.

A plataforma JEE contém uma série de especificações, cada uma com funcionalidades distintas. Entre elas, tem-se:

- JDBC (*Java Database Connectivity*), utilizado no acesso a banco de dados;
- JSP (*Java Server Pages*), um tipo de servidor Web. Grossamente falando, servidores Web são as aplicações que permitem a você acessar um site na internet;
- Servlets, para o desenvolvimento de aplicações Web, isto é, esse recurso "estende" o funcionamento dos servidores Web, permitindo a geração de conteúdo dinâmico nos sites.

O Java Web tem várias funcionalidades, uma delas é por proporcionar uma segurança maior das informações, devido as sãs inúmeras configurações que deixam o código bem separado, dividido deixando um baixo acoplamento.

A tecnologia Java Web, está muito difundida no mercado, por tanto quem adquire um produto com Java Web, certamente achará pessoas com bom conhecimento para possíveis correções e melhoramentos nas suas aplicações.

### **2.3.13. JSF – Java Server Faces**

“*Java Server Faces* (JSF) é uma estrutura para criar aplicações baseadas em WEB utilizando interfaces” [GEA02]. O JSF é um framework de aplicações para ambiente Web, que possibilita ao desenvolvedor, construir telas dinamicamente com em NetBeans ou Delphi arrastando e soltando.

JSF aplica o conceito de interface com aplicação de padrão MVC, tornando sua arquitetura muito simples e objetiva, que simplifica a complexidade do código para Java WEB, mas preservado seus conceitos de segurança e aplicabilidade.

### 2.3.14. Orientação a Objeto

[JAV09B] Diz que:

O avanço das tecnologias na área de análise, projeto, desenvolvimento e qualidade de software permitem que aplicações mais flexíveis e poderosas sejam construídas. A necessidade de integrar estas aplicações criou uma nova metodologia de análise e desenvolvimento: a Orientação a Objetos.

A Orientação a Objeto ou apenas OO, é uma tecnologia de desenvolvimento de software em que as linguagem de programação estão se adaptando e se adequando a ela, pelo fato de seus conceitos serem vistos com bons olhos por críticos em desenvolvimento de sistemas.

A OO tem vários conceitos que devem ser obedecidos em sua aplicação, mas das mais conhecidas é a herança múltipla que não é permitida no conceito de orientação a objetos, outros também importantes são, herança, polimorfismos entre outros.

### 2.3.15. Ireport

[JAV09C] diz que,

O IReport é uma ferramenta que visa facilitar a construção de relatórios utilizando a biblioteca JasperReports (<http://jasperreports.sourceforge.net>) através de uma interface gráfica desenvolvida em Swing. Ele dispõe de importantes ferramentas para desenvolver relatórios complexos e demorados.

O IReport é uma ferramenta de geração de relatórios que é comumente utilizada em sistemas com linguagem Java, por ser uma ferramenta livre também, e seu código ser feito em

Java. A Simplicidade de sua aplicação torna uma ferramenta para geração de relatórios de boa usabilidade, praticidade e confiabilidade.

### **2.3.16. Jude**

O JUDE é um sistema de modelagem de diagramas, comumente usado para modelar os diagramas da UML, mas não se restringe só a UML, com o JUDE modelam-se diagramas de ciclos de vida e também de árvores genealógicas.

O JUDE é mais utilizado com a UML, por realizar várias funções de interação com desenvolvimento de sistemas tipo: após criado o diagrama de Classes da UML o JUDE oferece uma função de gerar o código a partir do seu diagrama, a ferramenta exporta o código a para algum tipo de linguagem orientada a objetos, tipo C#, Java, C++. O mesmo acontece fazendo uma engenharia reversa do código para diagrama, mas neste caso o usuário monta os diagramas, como afirma [INF08].

JUDE é uma ferramenta profissional de modelagem para sistemas a qual suporta UML, diagrama entidade relacionamento, Flowchart, CRUD, Mini Mapas e Diagrama de Fluxo de Dados. Permite também a conversão entre modelos UML, ER Diagramas, Flowcharts, fluxo de dados e mini mapas.

### **3. ANÁLISE**

Este capítulo trata dos artefatos construídos na fase de análise.

#### **3.1 DESCRIÇÃO DO SISTEMA**

O sistema GerPro possui dois objetivos principais: auxiliar o aluno quanto à geração da Proposta de Especificação de Software - PESw, assim como o apoio à correção deste artefato pelos professores responsáveis. Para que tais funcionalidades sejam implementadas, se fazem necessários os cadastros de usuários do sistema, as equipes e as propostas.

Alunos e professores deverão ser cadastrados pelo coordenador para obter acesso ao sistema. Após o cadastro do aluno, o coordenador relaciona-o com uma equipe no momento em que cadastra uma proposta para a equipe, então o GerPro disponibiliza os itens, notando que uma proposta não deve ser aproveitada para outro período e sim criar uma nova. O preenchimento de cada item será obrigatório e de inteira responsabilidade do aluno. Um status é acrescido à proposta pelo sistema para informar que a mesma está em fase de construção.

Após o preenchimento de todos os itens da PESw, o aluno poderá submeter sua proposta para correção. Após a proposta submetida, nenhuma alteração poderá ser realizada

pelo aluno no documento, estando este disponível apenas para visualização. Nesse momento, o status da proposta é alterado para Proposta Em Correção.

Após a submissão da proposta para a correção o coordenador deve alocar as propostas para os professores corrigi-las, o sistema aloca as proposta de forma dinâmica, para um ou até dois professores corrigirem as propostas.

Para avaliação do professor, o sistema disponibiliza campos como apoio à correção de cada item a ser analisado. O professor pode inserir observações. As respostas dadas às perguntas são armazenadas e compõem o status da proposta analisada. Abaixo segue a lista de Perguntas:

- 1) *Apresentação e estilo (bem organizado, claro, correção gramatical e ortográfica), coesão e coerência textual?*
- 2) *Trabalho de acordo com os padrões de escrita definidos?*
- 3) *Justificativa clara, com texto bem organizado e, enfatizando a relevância e a problemática que implica na necessidade do desenvolvimento do Sistema?*
- 4) *Missão do produto de acordo com a abrangência do sistema?*
- 5) *Lista de Funções bem definidas, demonstrando, no mínimo, 7 (sete) requisitos do sistema (entrada, processamento e saída). Sendo pelo menos 3 de processamento?*
- 6) *Metodologia coerente com os objetivos definidos e adequada abrangência do tema?*
- 7) *Sistema pertencente ao grupo de trabalhos aceitos?*

O professor poderá escolher um item qualquer para correção, corrigi-lo e salvar a correção. Após todos os itens serem corrigidos, o sistema disponibiliza a opção de submeter correção, sendo que após submeter, o professor não poderá mais alterar a correção. Neste momento, o sistema atualiza o status da proposta com base na combinação dos resultados das perguntas das correções. Como é mostrado à baixo:

Se Resposta NÃO para a Pergunta 7 ou Resposta NÃO para as Perguntas 3 ou 5 ou 6, a Proposta está Reprovada.

Se Reposta NÃO para as Perguntas 3 ou 5 ou 6 e Reposta SIM para a Pergunta 7, a Proposta está Aprovada com Ressalva.

Qualquer resultado que não sejam iguais as 2 combinações anteriores, a proposta está Aprovada.



Após a combinação de resultados o sistema gera os possíveis os seguintes status: Proposta Aprovada, Proposta Aprovada com Ressalva ou Proposta Reprovada. A correção é realizada por até dois professores por proposta, sendo que cada resultado final de cada correção sendo essa a segunda correção, os status serão somado o qual gera um status final. O cálculo é:

- Se a primeira correção o status final da Proposta for *Aprovada* e na segunda correção a Proposta recebe o status *Aprovada* o status final será *Aprovada*.
- Se a primeira correção o status final da Proposta for *Aprovada* e na segunda correção a Proposta recebe o status *Aprovada com Ressalva* o status final será *Aprovada*.
- Se a primeira correção o status final da Proposta for *Aprovada* e na segunda correção a Proposta recebe o status *Reprovada* o status final será *Aprovada com Ressalva*.
- Se a primeira correção o status final da Proposta for *Aprovada com Ressalva* e na segunda correção a Proposta recebe o status *Aprovada com Ressalva* o status final será *Aprovada com Ressalva*.
- Se a primeira correção o status final da Proposta for *Reprovada* e na segunda correção a Proposta recebe o status *Aprovada com Ressalva* o status final será *Reprovada*.
- Se a primeira correção o status final da Proposta for *Reprovada* e na segunda correção a Proposta recebe o status *Reprovada* o status final será *Reprovada*.

O GerPro dispõem de relatórios para o gerenciamento do processo, dentre eles a lista de Equipes e suas propostas e seus respectivos status e o relatórios de porcentagem de propostas aprovadas e reprovadas que tem o seguinte calculo: ((lista propostas por Status \* 100)/lista todas as Proposta).

Para aumentar a segurança da aplicação o GerPro dispõe uma aplicação de criptografia de senha no banco de dados o qual torna as informações armazenadas mais segura do que as senhas padrões.

Para melhorar o processo de desenvolvimento foi utilizado um sistema de versionamento de codificação e também de documentação, o qual deixou a todo momento o sistema seguro no quesito de todos os desenvolvedores tinham acesso e todas as modificações eram refletido em todo o projeto, foi utilizados um servidor da empresa Google o qual ficou disponível na web a todo tempo.

Com a utilização do GerPro, acredita-se em uma melhora substancial no que diz respeito a agilidade e praticidade na construção e correção de PESw, contribuindo para o bom desenvolvimento das atividades relacionadas a este processo.

## 3.2 LISTA DE FUNÇÕES

Número de ordem	Caso de uso	Descrição
1	Manter Proposta	Processo de inclusão, alteração e exclusão de Proposta.
2	Pesquisar Proposta	Processo de consultar de Proposta.
3	Manter Usuários	Processo de inclusão, pesquisa e alteração de Aluno, Professor e Coordenador.
4	Construir Proposta	Preenche o conteúdo dos itens da proposta.
5	Submeter Proposta	Submete a proposta para correção e atualiza o status e desabilita a proposta para alteração.
6	Alocar Proposta	Aloca proposta que estão com status igual a concluído, dinamicamente para os professores selecionados.
7	Corrigir Proposta	Correção dos itens da proposta de acordo com a pergunta.
8	Submeter Correção	Submete a correção da proposta e excuta o Calcular status da proposta e desabilita a proposta para alteração na correção.
9	Calcular Status Final da Proposta.	Calcula o status final da proposta atualizado o status da proposta para o status calculado, que podem ser Aprovado, Reprovado ou Aprovado com ressalvas.
10	Gerar lista de Equipe e Proposta.	Gerar relatório das equipes e suas propostas, com status e data de criação e de submissão da proposta.
11	Gerar Lista de Professores e Propostas	Gerar relatório de professores e as proposta que foram alocadas para o mesmo, mostrando o status e o período da proposta.
12	Gerar gráfico com a Porcentagem de Propostas Aprovadas, Reprovadas, Aprovadas com Ressalvas	Gerar o relatório de porcentagem de propostas Aprovadas, Reprovadas e Aprovadas com ressalvas e mostrar em um gráfico pizza.

## 3.3 DIAGRAMA DE CASOS DE USO

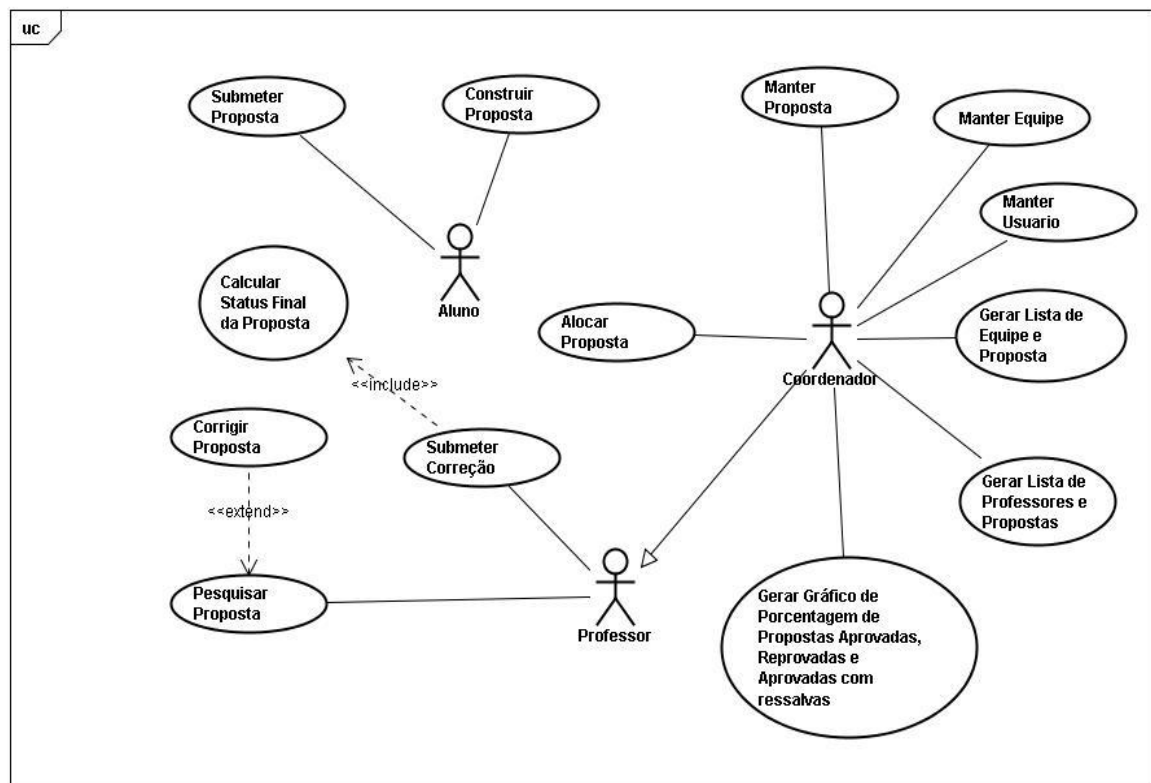


Figura 1 - Diagrama de Casos de Uso

### 3.4 DESCRIÇÃO DE CASOS DE USO

#### 3.4.1. Caso de Uso Manter Proposta

<b>Caso de Uso:</b> Manter Propostas	<b>Sigla:</b> UC01
<b>Objetivo:</b> Utilizar as operações de cadastrar, editar ou pesquisa.	
<b>Ator:</b> Coordenador.	
<b>Pré-condições:</b> Não aplicável	
<b>Pós-condições:</b> Não aplicável	
<b>Fluxo Principal</b> <ol style="list-style-type: none"> <li>Este caso de uso se inicia quando o ator aciona a opção cadastrar na tela homeCoordenador.</li> <li>O sistema exibe a tela PesquisarProposta com uma lista de propostas carregadas em uma tabela com os campos Nome, Equipe, Status e Opções.</li> <li>O ator pode optar por Pesquisar, Nova Proposta, Editar ou Sair.</li> <li>Se a opção for Pesquisar, o sistema aciona o fluxo alternativo Pesquisar (FA(01)).</li> <li>Se a opção for Nova Proposta, o sistema aciona o fluxo alternativo Nova Proposta (FA(02)).</li> <li>Se a opção for Editar, o sistema aciona o fluxo alternativo Editar (FA(03)).</li> <li>Se a opção for Sair, o sistema aciona o fluxo alternativo Sair (FA(04)).</li> <li>Este caso de uso se encerra.</li> </ol>	
<b>Fluxo alternativo Pesquisar (FA01) ref. UC01(4)</b> <b>Pré condição:</b> Não aplicável. <b>Passos:</b> <ol style="list-style-type: none"> <li>O ator pode optar por pesquisar por Nome ou Equipe.</li> <li>O ator preenche campo nome ou equipe.</li> <li>O ator aciona o botão Pesquisar.</li> <li>O sistema exibe na tabela, a Proposta, equipe, status e o botão Editar ao lado da proposta.</li> <li>Este fluxo se encerra.</li> </ol>	
<b>Fluxo alternativo Nova Proposta (FA02) ref. UC01(5)</b> <b>Pré condição:</b> O ator acionou o botão Nova Proposta. <b>Passos:</b> <ol style="list-style-type: none"> <li>O sistema filtra as equipes que não possuem relação com propostas.</li> <li>O sistema exibe a tela CadastroProposta com as equipes carregada no campo equipe.</li> <li>O ator preenche o campo Nome.</li> <li>O ator seleciona a equipe no campo Equipe.</li> <li>O ator preenche campo Período.</li> <li>O ator aciona o botão Salvar.</li> <li>O sistema lê a data corrente do servidor.</li> </ol>	

<p>8. O sistema salva os dados.</p> <p>9. O sistema exibe a mensagem na mesma tela: Parabéns: Cadastro Salvo com sucesso.</p> <p>10. O Este fluxo se encerra.</p>	
<p><b>Fluxo alternativo Editar (FA03) ref. UC01(6)</b></p> <p><b>Pré condição:</b> O ator acionou o botão Editar.</p> <p><b>Passos:</b></p> <ol style="list-style-type: none"> <li>1. O sistema exibe a tela AlterarProposta com os campos nome, equipe, data_criacao e período carregados, sendo que somente o campo Nome estará em modo de edição.</li> <li>2. O ator altera o campo nome.</li> <li>3. O ator aciona o botão Salvar.</li> <li>4. O sistema atualiza na tabela PROPOSTA o campo nome.</li> <li>5. O sistema exibe a mensagem: Parabéns: Cadastro Salvo com sucesso.</li> <li>6. O sistema retorna para a tela PesquisarProposta.</li> </ol>	
<p><b>Fluxo alternativo Sair (FA03) ref. UC01(1,3,4,5,6,7), FA01(1,2,3), FA02(3,4,5,6) e FA03(2, 3).</b></p> <p><b>Pré condição:</b> Não aplicável.</p> <p><b>Passos:</b></p> <ol style="list-style-type: none"> <li>4. O ator aciona a opção Sair.</li> <li>5. O sistema retorna para a tela Login.</li> <li>6. Este fluxo se encerra.</li> </ol>	
<p><b>Fluxo de exceção Campos Obrigatórios. (FE01) ref. FA02(3,5), FA03(2).</b></p> <p><b>Pré-condição:</b> Não Preencher campos obrigatórios, ou seja, campos com asterisco (*).</p> <p><b>Passos:</b></p> <ol style="list-style-type: none"> <li>1. O ator aciona o botão Salvar.</li> <li>2. O sistema exibe na tela a mensagem, Preenchimento Obrigatório.</li> <li>3. Este fluxo se encerra.</li> </ol>	
<p><b>Mensagens:</b></p>	
<b>ID:</b>	<b>Descrição:</b>

### 3.4.2. Caso de Uso Alocar Propostas

<b>Caso de Uso:</b> Alocar Propostas.	<b>Sigla:</b> UC02
<b>Objetivo:</b> Alocar proposta dinamicamente para professores selecionados.	
<b>Ator:</b> Coordenador.	
<b>Pré-condições:</b> Todos os itens devem estar com status igual a corrigido.	
<b>Pós-condições:</b> Proposta desabilitada para usuário aluno.	

**Fluxo Principal**

1. Este caso de uso se inicia quando o ator aciona a opção Alocar, na tela homeCoordenador.
2. O sistema exibe a tela AlocarProposta com todos o professores cadastrados.
3. O ator seleciona a opção Grupo.
4. O ator seleciona um Professor.
5. O ator aciona o botão Adicionar.
6. O ator aciona o botão Alocar.
7. O sistema aloca as propostas para o professor selecionado.
8. O sistema atualiza o status da proposta para Em correção.
9. Este fluxo se encerra.

**Fluxo Alternativo Grupo e Dois Professores (FA01) ref. UC02(3,4,5,6)**

**Pré condição:** O ator seleciona dois professores.

**Passos:**

1. O sistema aloca dinamicamente as propostas para os professores que foram selecionados, sendo que cada proposta é alocada para os dois professores.
2. O sistema salva as alocações.
3. O sistema atualiza o status da proposta para Em correção.
4. Este fluxo se encerra.

**Fluxo Alternativo Individual (FA02) ref. UC02(5)**

**Pré condição:** Não aplicável.

**Passos:**

1. O ator seleciona a opção Individual.
2. O ator seleciona dois Professores.
3. O ator aciona o botão Adicionar.
4. O ator aciona o botão Alocar.
5. O sistema aloca as propostas para o professor selecionado.
6. O sistema atualiza o status da proposta para Em correção.
7. Este fluxo se encerra.

**Fluxo Alternativo Individual e Dois Professores (FA03) ref. UC02(5)**

**Pré condição:** Não aplicável.

**Passos:**

1. O ator seleciona a opção Individual.
2. O ator seleciona dois Professores.
3. O ator aciona o botão Adicionar.
4. O ator aciona o botão Alocar.
5. O sistema verifica se a quantidade de proposta  
Se quantidade de proposta igual a par.

O sistema percorre a lista de proposta alocando as proposta alternando os professores,  
ou seja:

Proposta1 para o professor1.

Proposta2 para o professor2.

Proposta3 para o professor1.

<p>Proposta4 para o professor2.</p> <p>Senão</p> <p>O sistema verifica</p> <p>Se</p> <p>Existe só uma proposta na lista.</p> <p>O sistema aloca a proposta para o primeiro professor da lista</p> <p>Senão</p> <p>O sistema retira a primeira proposta da lista.</p> <p>O sistema percorre a lista de proposta alocando as proposta alternando os professores, ou seja:</p> <p>Proposta1 para o professor1.</p> <p>Proposta2 para o professor2.</p> <p>Proposta3 para o professor1.</p> <p>Proposta4 para o professor2.</p> <p>O sistema aloca randomicamente a proposta retirada para um professor.</p> <p>6. O sistema atualiza o status da proposta para Em correção.</p> <p>7. Este fluxo se encerra.</p>
<p><b>Fluxo Alternativo Sair (FA04) ref. UC02(1,3,4,5,6), FA02(1,2,3,4) e FA03(1,2,3,4)</b></p> <p><b>Pré condição:</b> O ator seleciona a opção Sair.</p> <p><b>Passos:</b></p> <ol style="list-style-type: none"> <li>1. O sistema retorna para a tela Login.</li> <li>2. Este fluxo se encerra.</li> </ol>
<p><b>Fluxo Exceção Selecionar mais de dois Professores para alocar (FE01) ref. UC02(5), FA02(3) e FA03(3).</b></p> <p><b>Pré condição:</b> O ator seleciona mais de dois professores.</p> <p><b>Passos:</b></p> <ol style="list-style-type: none"> <li>1. O sistema exibe a mensagem: Selecione no máximo dois professores.</li> <li>2. Este fluxo se encerra.</li> </ol>
<p><b>Fluxo Exceção Não Selecionou nenhum professor (FE02) ref. UC02(),5 FA02(3) e FA03(3).</b></p> <p><b>Pré condição:</b> O ator não selecionou nenhum professor.</p> <p><b>Passos:</b></p> <ol style="list-style-type: none"> <li>1. O sistema exibe a mensagem: Selecione ao menos um professor.</li> </ol>



2. Este fluxo se encerra.

### 3.4.3. Caso de uso Submeter Correção

<b>Caso de Uso:</b> Submeter Correção.	<b>Sigla:</b> UC03
<b>Objetivo:</b> Processar status da proposta, alterando para Aprovado, Reprovado ou Aprovado com Ressalva.	
<b>Ator:</b> Professor.	
<b>Pré-condições:</b> Todos os itens devem estar com status igual a corrigido.	
<b>Pós-condições:</b> Proposta desabilitada para correção.	
<p><b>Fluxo Principal</b></p> <ol style="list-style-type: none"> <li>1. Este caso de uso se inicia quando o ator aciona o botão Submeter, na tela SubmeterCorrecao.</li> <li>2. O sistema verifica.</li> </ol> <p>Se a resposta for igual a Não para as perguntas: (Sistema pertencente ao grupo de trabalhos aceitos)</p> <p><b>ou</b></p> <p>(Justificativa clara, com texto bem organizado e, enfatizando a relevância e a problemática que implica na necessidade do desenvolvimento do Sistema)</p> <p><b>e</b></p> <p>(Lista de Funções bem definidas, demonstrando, no mínimo, 7(sete) requisitos do sistema (entrada, processamento e saída). Sendo pelo menos 3 de processamento)</p> <p><b>e</b></p> <p>(Metodologia coerente com os objetivos definidos e adequada abrangência do tema).</p> <p>O sistema armazena o status da proposta que é igual a Reprovado em um objeto proposta em tempo de execução.</p> <p>O sistema executa o subfluxo Calcula Status Final da proposta.</p>	

Senão

Se a resposta for igual Não para as perguntas:

(Justificativa clara, com texto bem organizado e, enfatizando a relevância e a problemática que implica na necessidade do desenvolvimento do Sistema)

**ou**

(Lista de Funções bem definidas, demonstrando, no mínimo, 7 (sete) requisitos do sistema (entrada, processamento e saída). Sendo pelo menos 3 de processamento)

**ou**

(Metodologia coerente com os objetivos definidos e adequada abrangência do tema)

**e**

Resposta igual a Sim para a pergunta:

(Sistema pertencente ao grupo de trabalhos aceitos), ou seja:

O sistema armazena o status da proposta que é igual a Aprovado com Ressalvas em um objeto proposta em tempo de execução.

O sistema executa o subfluxo Calcula Status Final da proposta.

Senão

O sistema armazena o status da proposta que é igual a Aprovado com Ressalvas em um objeto proposta em tempo de execução.

O sistema executa o subfluxo Calcula Status Final da proposta.

3. O sistema desabilita a proposta submetida.
4. O sistema exibe a mensagem: Proposta Submetida com Sucesso.
5. Este fluxo se encerra.

### **Subfluxo calcular Status Final Proposta (FA02) ref. UC03(2)**

**Pré condição:** Não aplicável.

#### **Passos:**

1. O sistema lê o status da proposta, armazenado no objeto proposta em tempo de execução.
2. O sistema verifica o status da proposta que esta salvo;  
Se o status da proposta estiver igual à Em correção;  
O sistema atualiza o status da proposta para o status armazenado no objeto proposta;

Senão se( status = Aprovado)

<p>Se status do objeto proposta=Aprovado    status da proposta=Aprovado com ressalvas</p> <p>O sistema atualiza o status da proposta para Aprovado;</p> <p>Senão</p> <p>O sistema atualiza o status da proposta para Aprovado com ressalvas;</p> <p>Senão se( status proposta = Aprovado com ressalvas )</p> <p>Se status do objeto proposta =Aprovado;</p> <p>O sistema atualiza o status da proposta para Aprovado;</p> <p>Senão se status do objeto proposta = Aprovado com ressalvas.</p> <p>O sistema atualiza o status da proposta para Aprovado com ressalvas;</p> <p>Senão</p> <p>O sistema atualiza o status da proposta para Reprovado;</p> <p>Senão se( status proposta = Reprovado)</p> <p>Se status do objeto proposta=Aprovado com ressalvas    status da proposta=Reprovado;</p> <p>O sistema atualiza o status da proposta para Reprovado;</p> <p>Senão</p> <p>O sistema atualiza o status da proposta para Aprovado com ressalvas.</p> <p><b>3. Este fluxo se encerra.</b></p>
<p><b>Fluxo Alternativo Sair (FA01) ref. UC03(1)</b></p> <p><b>Pré condição:</b> Não aplicável.</p> <p><b>Passos:</b></p> <ol style="list-style-type: none"> <li>1. O ator aciona a opção Sair.</li> <li>2. O sistema retorna para tela Login.</li> <li>3. Este fluxo se encerra.</li> </ol>
<p><b>Fluxo de Exceção Itens não Corrigido (FE01) ref. UC03(2)</b></p> <p><b>Pré condição:</b> Item com status igual à em correção.</p> <p><b>Passos:</b></p> <ol style="list-style-type: none"> <li>1. O sistema verifica se existe algum item com status = Em correção.</li> <li>2. O sistema exibe msg: Existem itens não corrigidos.</li> <li>3. Este fluxo se encerra.</li> </ol>

### 3.4.4. Caso de Uso Gerar Gráfico de Porcentagem de Propostas Aprovadas, Reprovadas e Aprovadas com Ressalvas.

<p><b>Caso de Uso:</b> Gerar Gráfico de Porcentagem de Propostas aprovadas, Reprovadas e Aprovadas com ressalvas</p> <p style="text-align: right;"><b>Sigla:</b> UC04</p>
<p><b>Objetivo:</b> Obter uma visão geral da porcentagem de Propostas Aprovadas, Aprovadas com Ressalvas e Reprovadas.</p>
<p><b>Ator:</b> Coordenador.</p>
<p><b>Pré-condições:</b> O ator deve ter acionado a opção Relatórios.</p>
<p><b>Pós-condições:</b> Não aplicável</p>
<p><b>Fluxo Principal</b></p> <ol style="list-style-type: none"> <li>1. Este caso de uso se inicia quando o ator aciona a opção Relatórios na tela homeCoordenador.</li> <li>2. O sistema exibe a tela Relatórios.</li> <li>3. O ator seleciona a opção Relatório Resultados de propostas.</li> <li>4. O ator preenche campo Período.</li> <li>5. O ator aciona o botão Visualizar.</li> <li>6. O sistema lê o status de todas as propostas.</li> <li>7. O sistema gera uma lista de proposta do período preenchido no passo 4.</li> <li>8. O sistema percorre a lista.</li> <li>9. O sistema verifica na lista.</li> </ol> <p style="padding-left: 40px;">Se</p> <p style="padding-left: 80px;">status proposta igual a Aprovado.</p> <p style="padding-left: 40px;">O sistema lê a quantidade de proposta aprovadas (Qtd AP) e divide pela quantidade de proposta da lista (Qtd total) e multiplica por 100, ou seja:</p> <p style="padding-left: 80px;"><math display="block">\text{PorcentagemAp} = \text{Qtd AP} / \text{Qtd Total}.</math></p> <p style="padding-left: 40px;">Senão</p> <p style="padding-left: 80px;">Se status proposta igual a Reprovado.</p> <p style="padding-left: 40px;">O sistema lê a quantidade de proposta reprovado (Qtd RP) e divide pela quantidade de proposta da lista (Qtd total) e multiplica por 100, ou seja:</p> <p style="padding-left: 80px;"><math display="block">\text{PorcentagemRp} = \text{Qtd RP} / \text{Qtd Total}.</math></p> <p style="padding-left: 40px;">Senão</p>

O sistema lê a quantidade de proposta aprovado com ressalvas (Qtd APR) e divide pela quantidade de proposta da lista (Qtd total) e multiplica por 100, ou seja:

$$\text{PorcentagemAPR} = \text{Qtd APR} / \text{Qtd Total}.$$

10. O sistema cria uma lista de valores (PorcentagenAp, PorcentagemRp e PorcentagemAPR).
11. O sistema cria uma lista de nomes (Aprovados, Reprovados e Aprovados com Ressalvas).
12. O sistema lê a lista de valores e nomes.
13. O sistema cria o gráfico pizza.
14. O sistema atribui valores com seus respectivos nomes no gráfico.
15. O sistema exibe a tela Resultado Propostas em forma PDF no gráfico pizza.
16. Este fluxo se encerra.

### 3.4.5. Caso de Uso Gerar Relatório de Equipes e Propostas

<b>Caso de Uso:</b> Gerar Lista de Equipes e Propostas	<b>Sigla:</b> UC05
<b>Objetivo:</b> Listar as equipes, suas propostas e os status das propostas.	
<b>Ator:</b> Coordenador.	
<b>Pré-condições:</b> O ator deve ter acionado a opção Relatórios.	
<b>Pós-condições:</b> Não aplicável	
<b>Fluxo Principal</b> <ol style="list-style-type: none"><li>1. Este caso de uso se inicia quando o ator aciona a opção Relatórios na tela homeCoordenador.</li><li>2. O sistema exibe a tela Relatórios.</li><li>3. O ator seleciona o Relatório Proposta x Equipe.</li><li>4. O ator preenche campo período.</li><li>5. O ator aciona o botão Visualizar.</li><li>6. O sistema busca todas as equipes com suas propostas, e as propostas com suas referidas datas e status.</li><li>7. O sistema exibe a tela Relatório Equipe x Proposta em forma de PDF.</li><li>8. Este fluxo se encerra.</li></ol>	

## 3.5 DIAGRAMAS

### 3.5.1. Classes

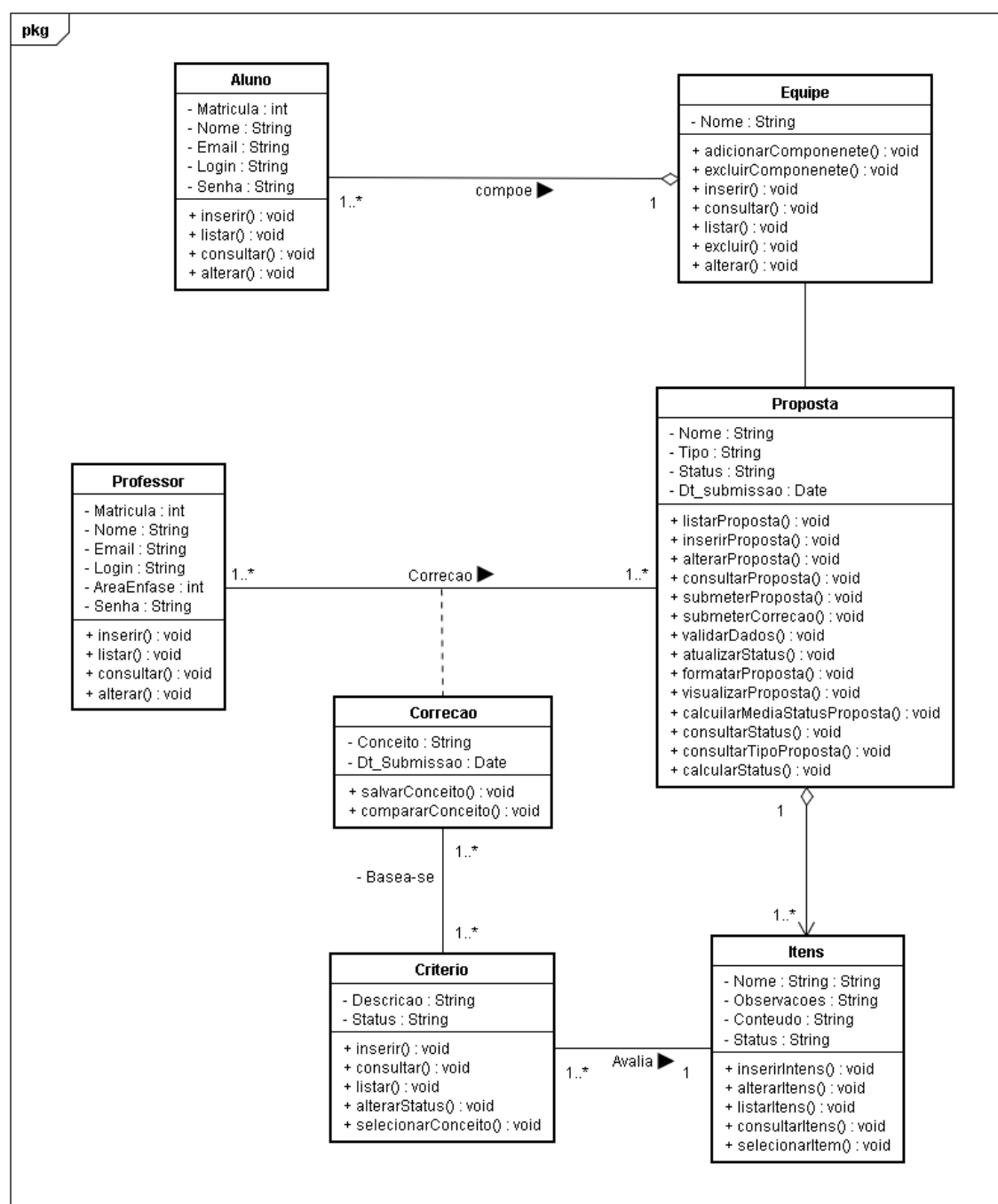


Figura 2 - Diagrama de Classes de Análise

### 3.5.2. Sequência Alocar Proposta

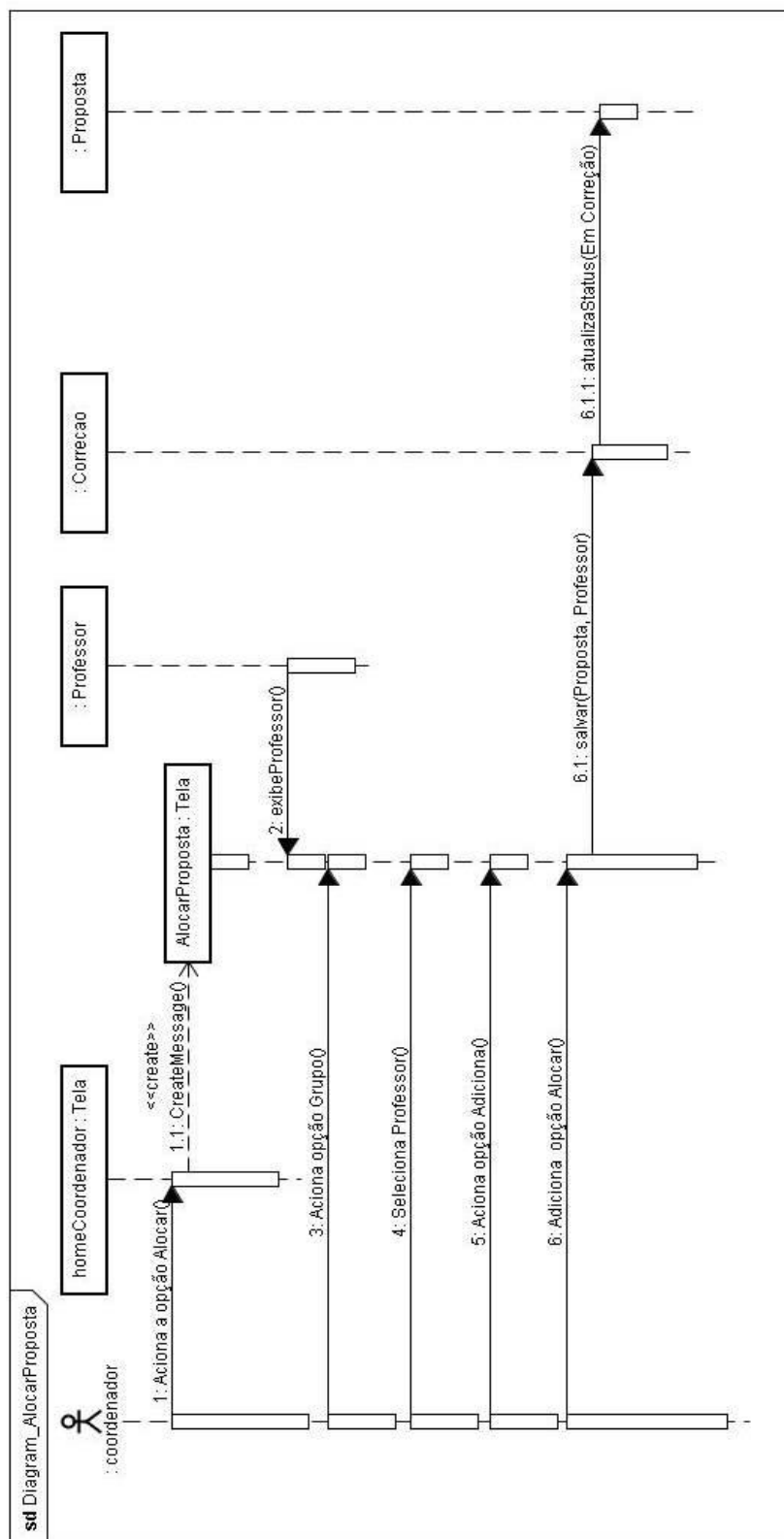


Figura 3 - Diagrama de Sequência Alocar Proposta



### 3.5.3. Sequência Submeter Correção

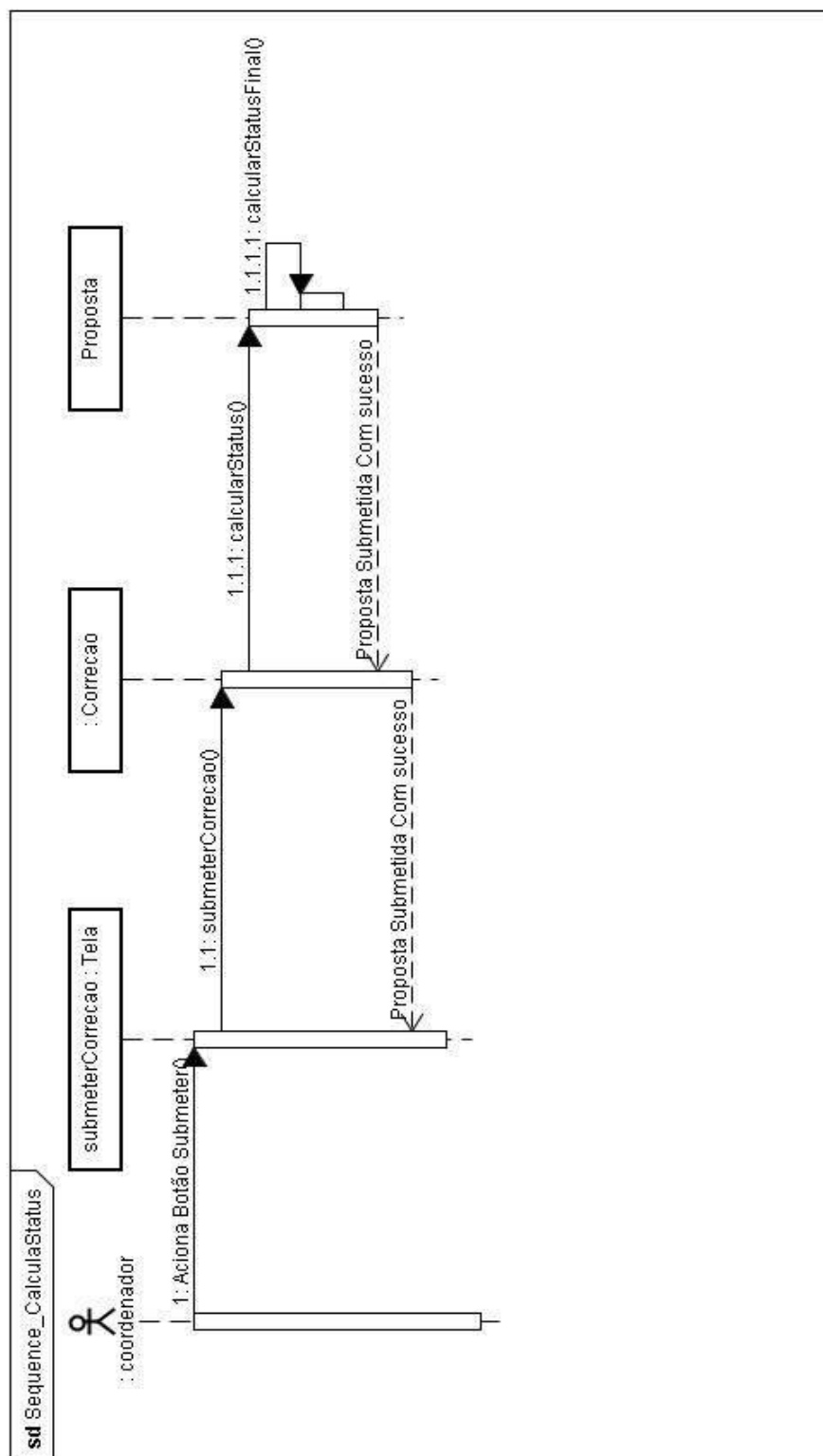


Figura 4 - Diagrama de Sequência Submeter Correção

### 3.5.4. Sequência Cadastrar Proposta

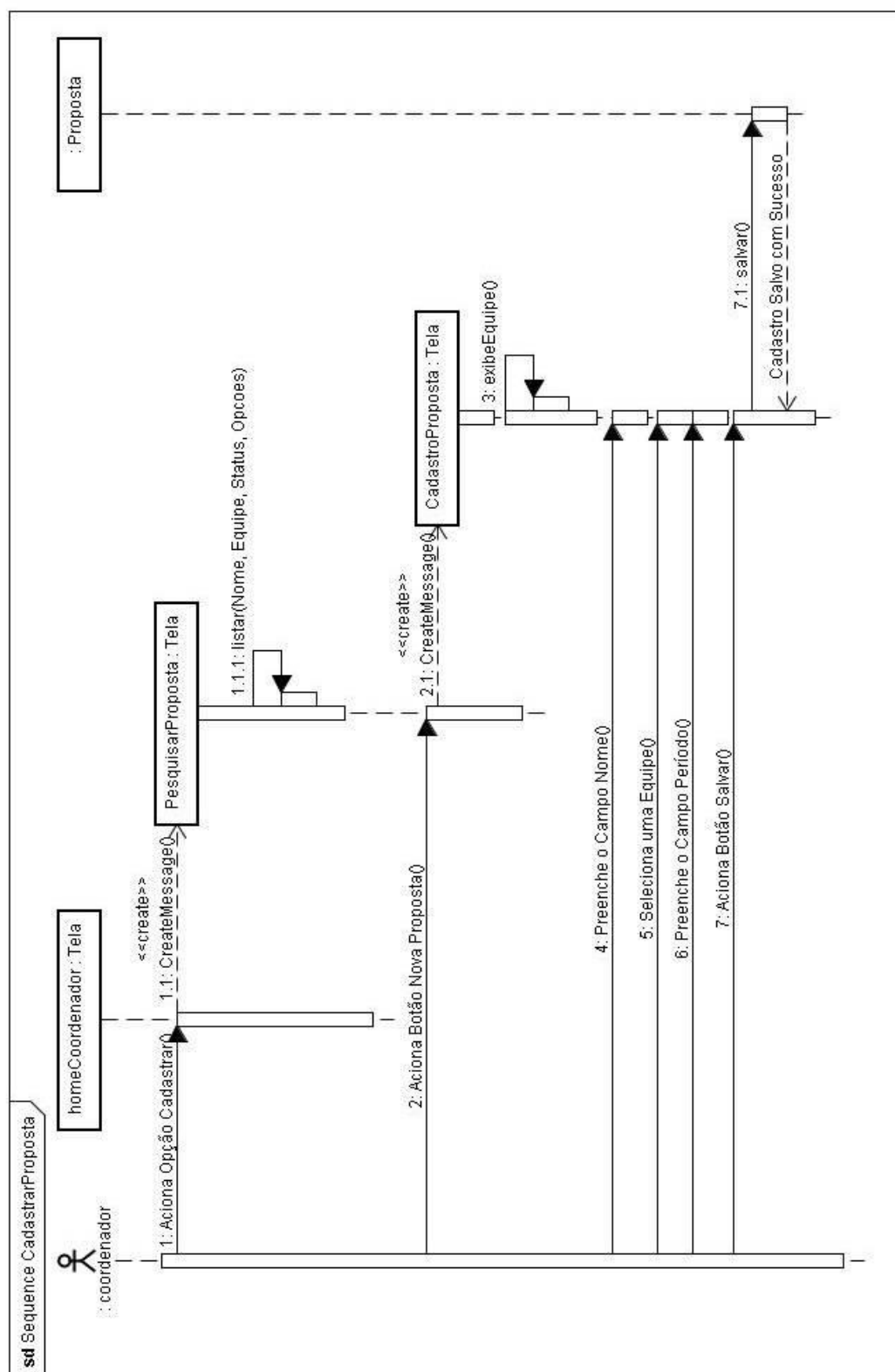


Figura 5 - Diagrama de Sequência Cadastrar Proposta

### 3.5.5. Sequência Editar Proposta

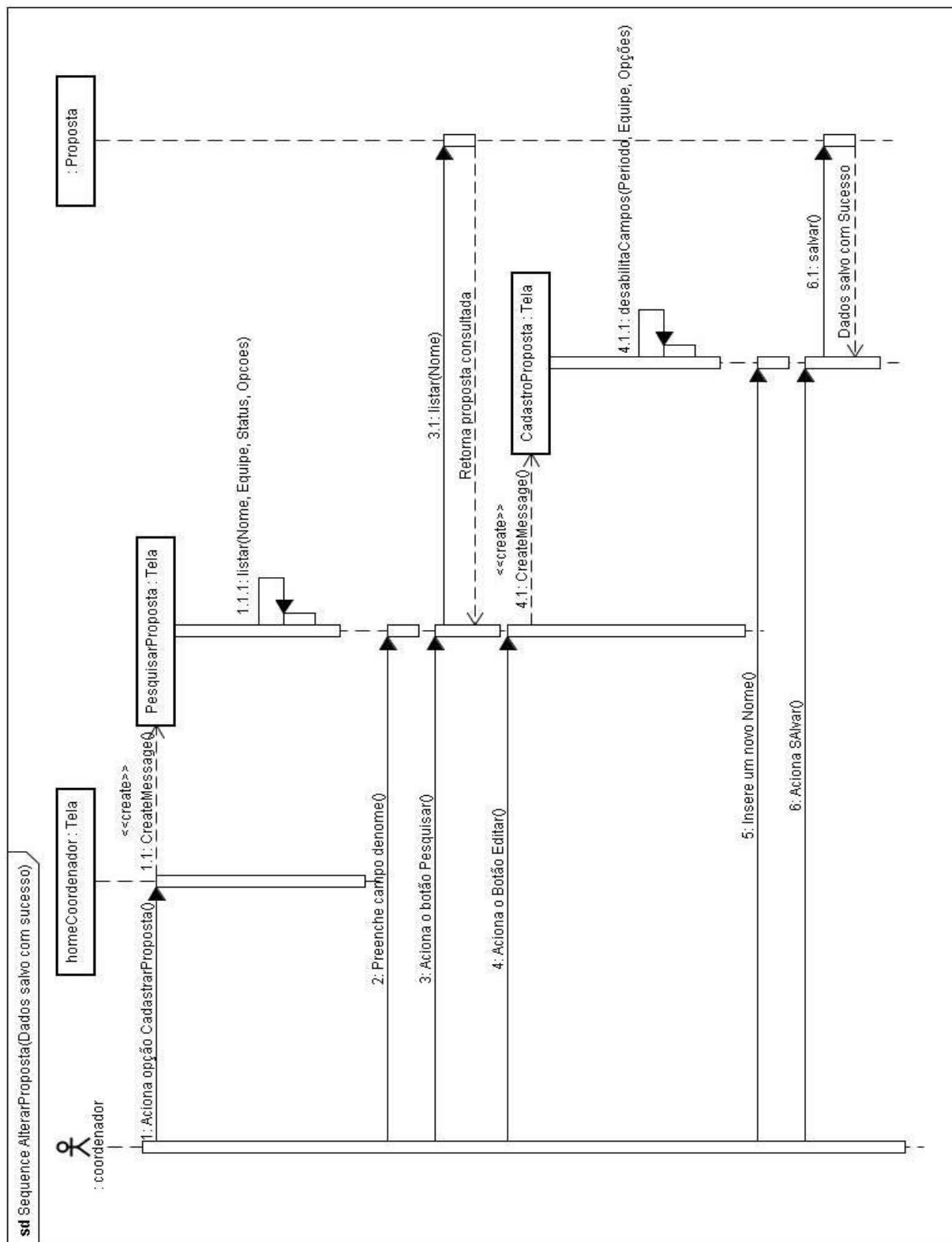


Figura 6 - Diagrama de Sequência Alterar Proposta

### 3.5.6. Sequência Pesquisar Proposta

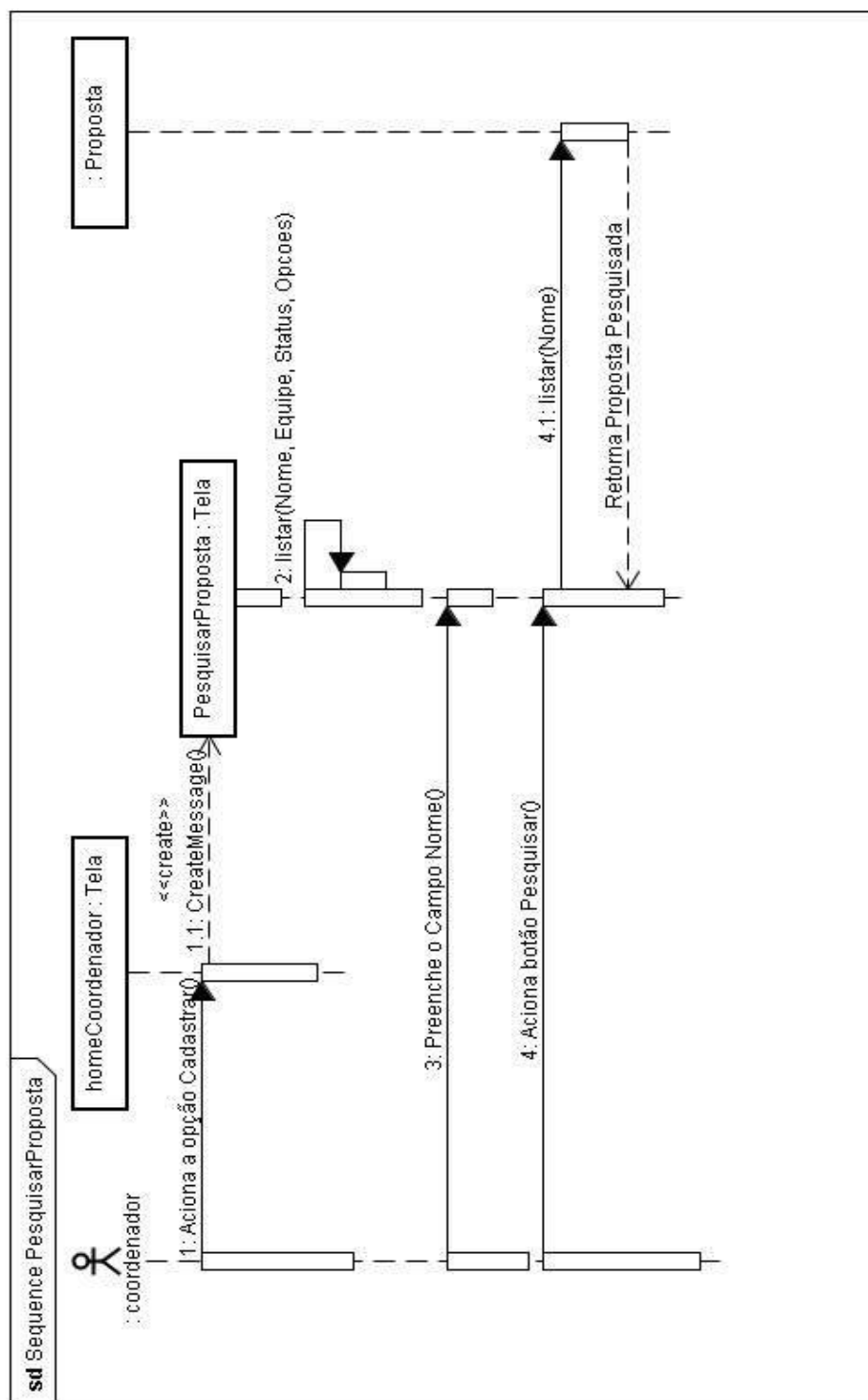


Figura 7 - Diagrama de Sequência Pesquisar Proposta

### 3.5.7. Sequência Gerar Relatório de Equipe e Proposta

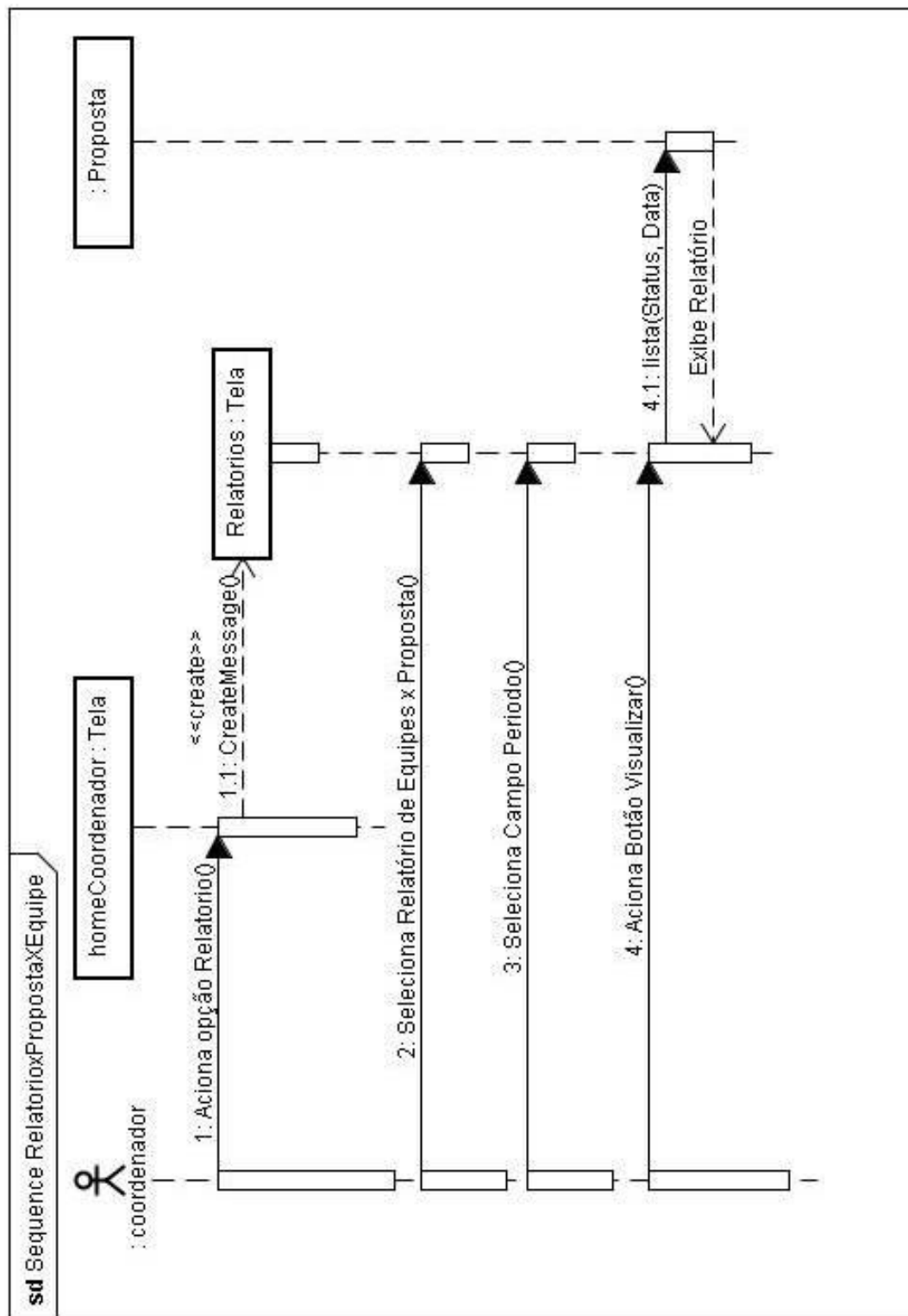


Figura 8 - Diagrama de Sequência Relatório Proposta x Equipe

### 3.5.8. Sequência Gerar Relatório de Porcentagem de Aprovadas, Reprovadas e Aprovadas com Ressalva

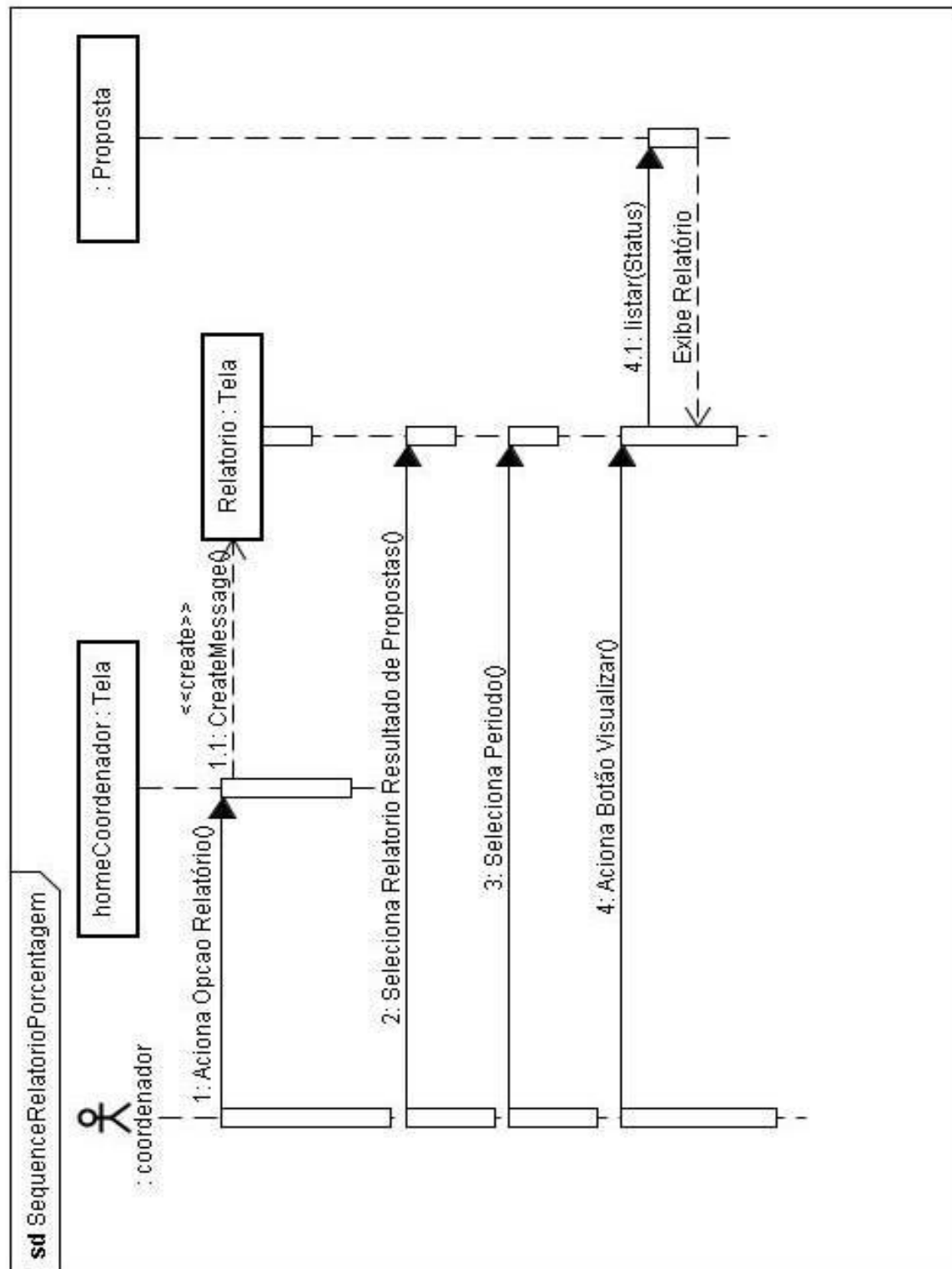


Figura 9 - Diagrama de Sequência Relatório de Porcentagem

### 3.5.9. Atividade cadastrar proposta

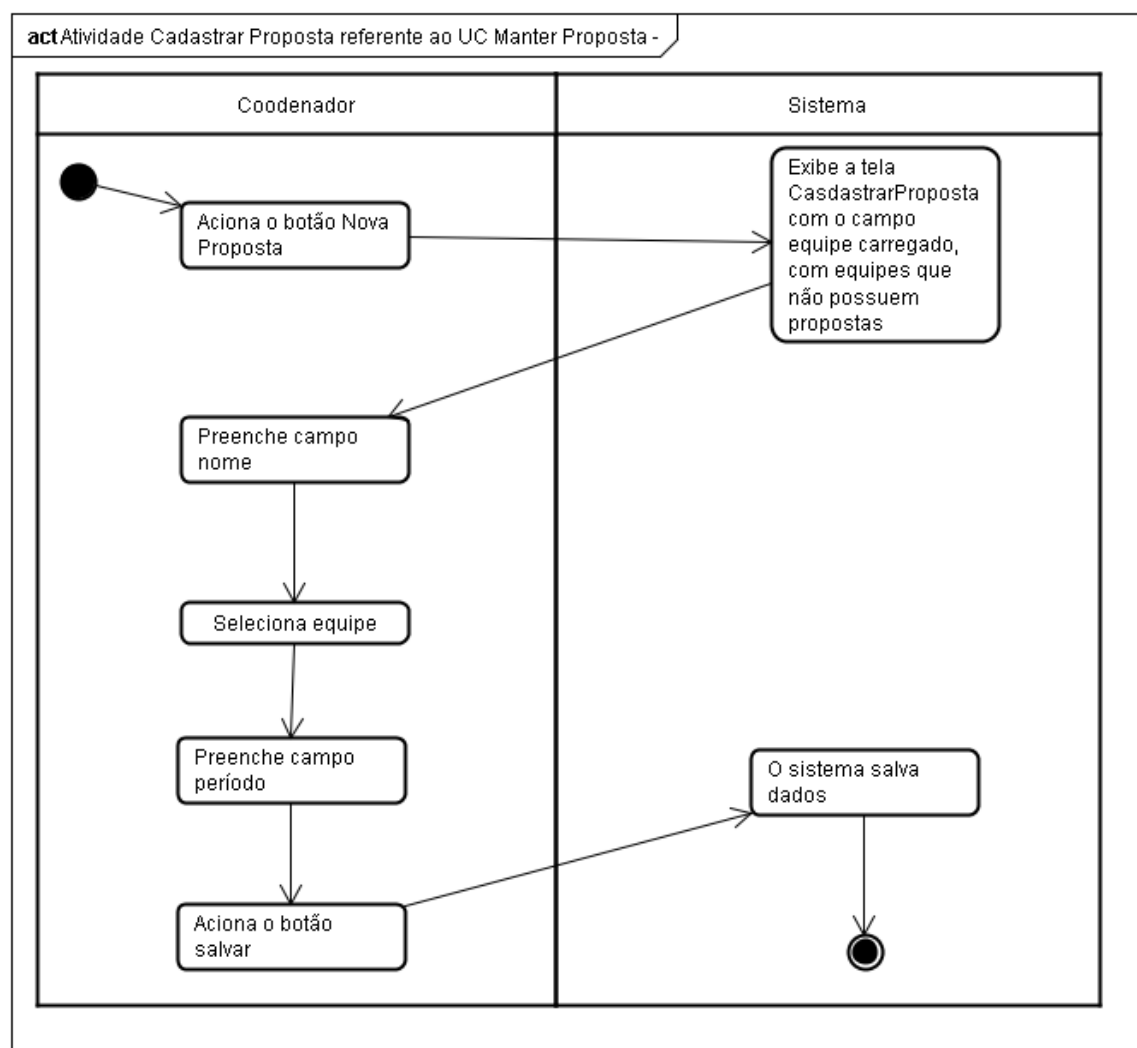
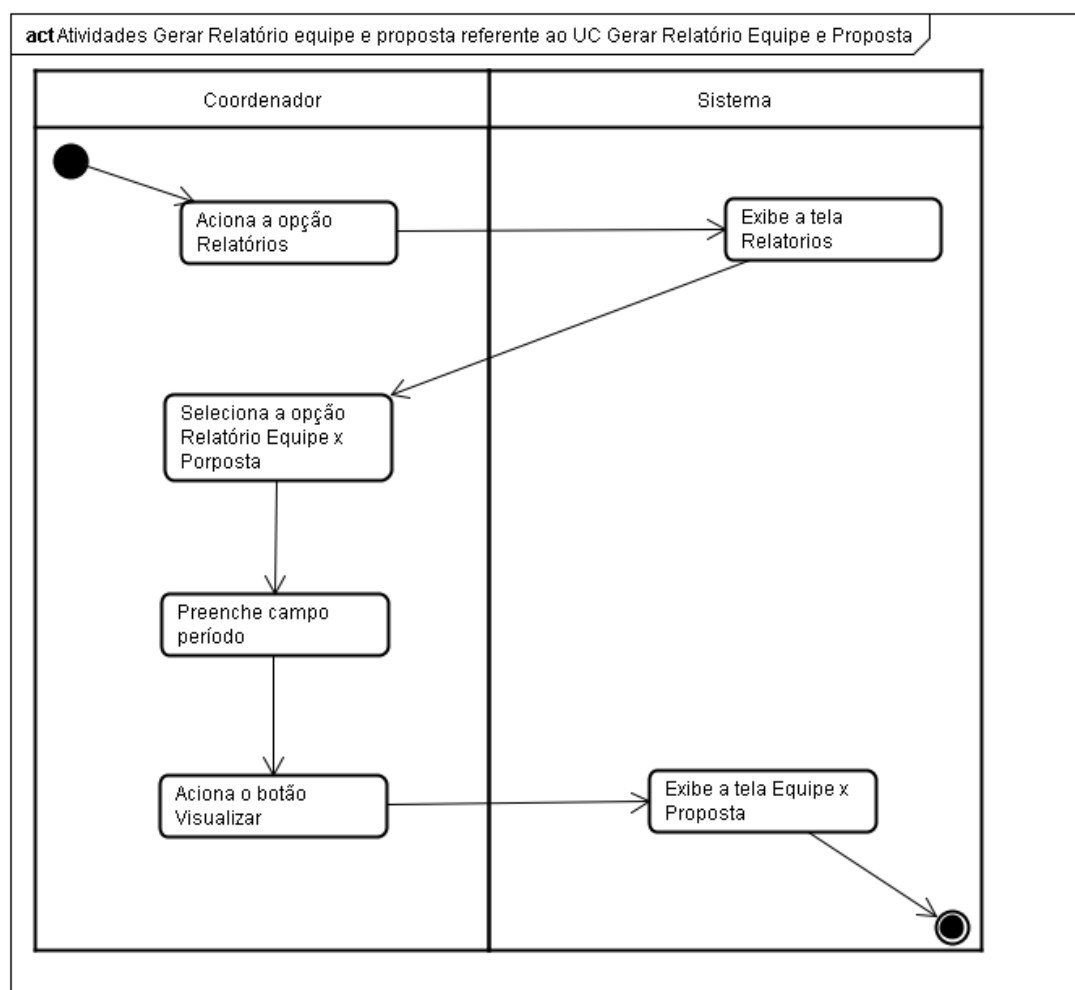


Figura 10 - Diagrama de Atividade Cadastrar Proposta

### 3.5.10. Atividade Gerar Relatório Equipe x Proposta



**Figura 11 - Diagrama de Atividade Gerar Relatório**



### 3.5.11. Estado Status do Item

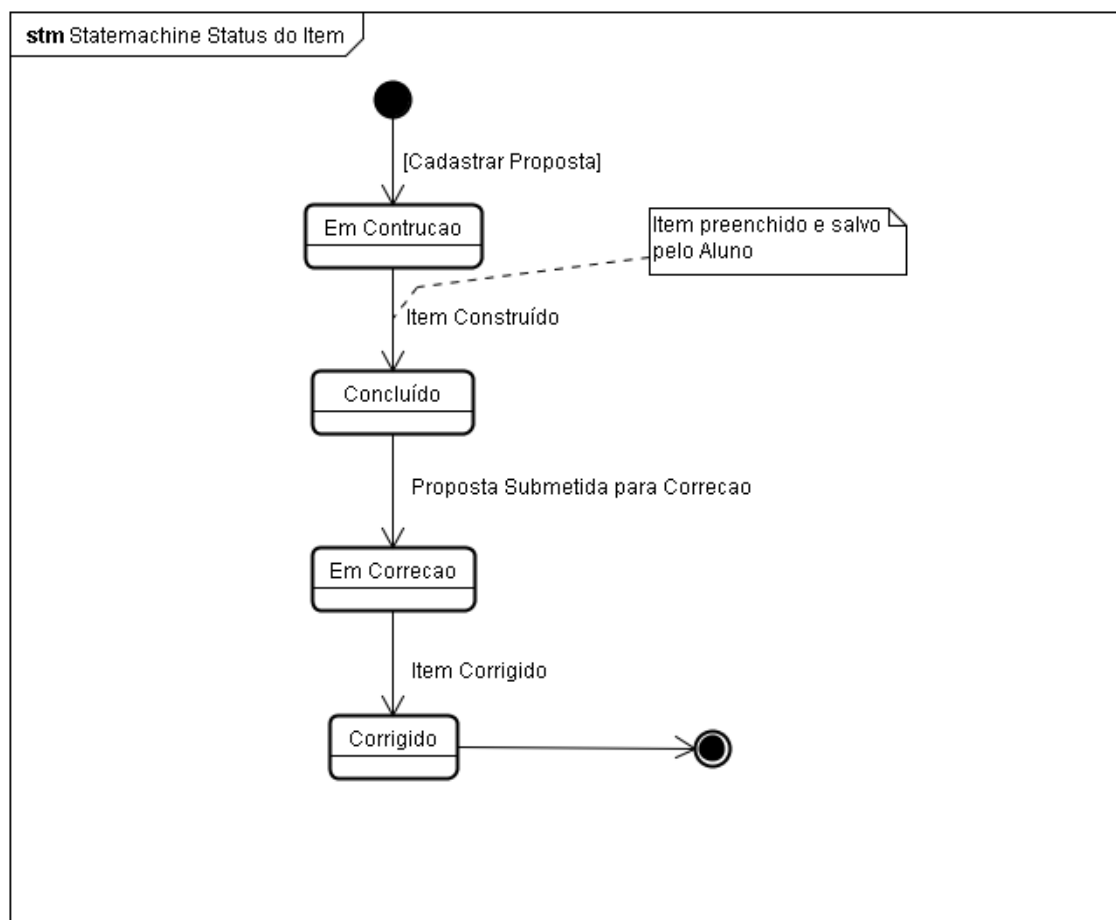


Figura 12 - Diagrama de Estado do Status do Item

### 3.5.12. Estado Status da Proposta

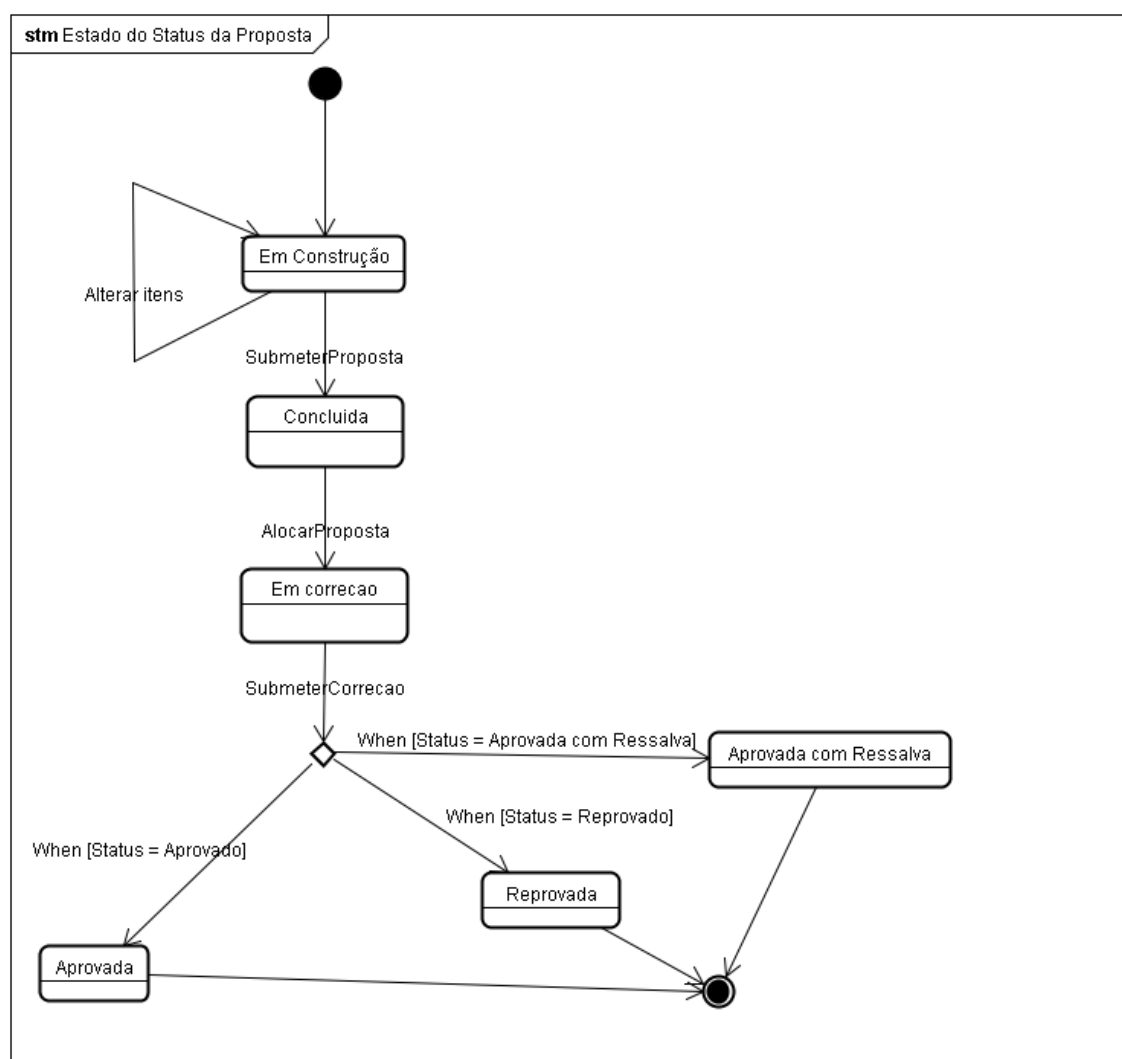


Figura 13 - Diagrama de Estado do Status da Proposta

## 3.5.13. Entidade - Relacionamento

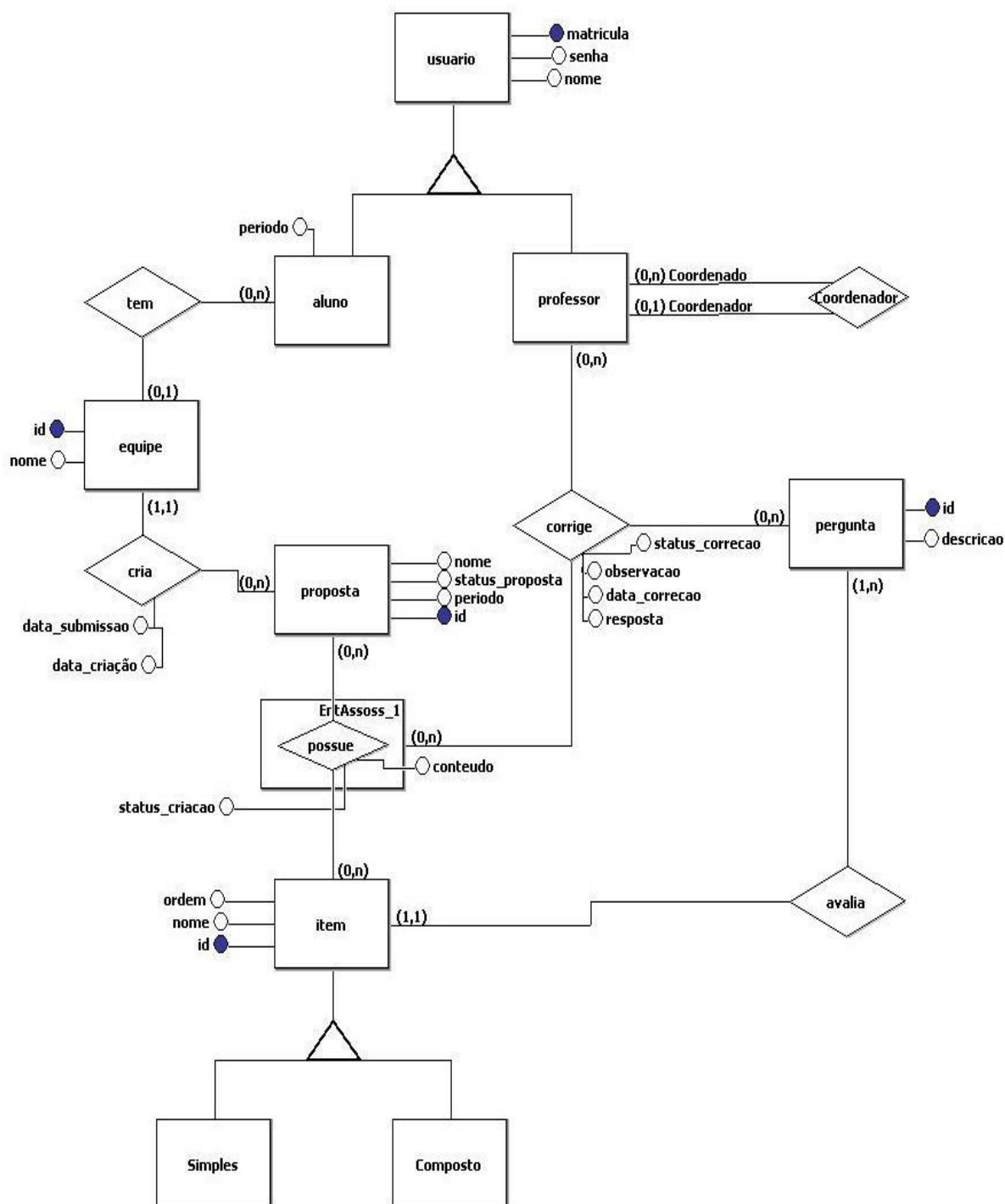


Figura 14 - Diagrama de Entidade e Relacionamento

## **4. PROJETO**

Este capítulo apresenta os artefatos criados na fase de projeto. Nesta fase, os documentos apresentados têm seu grau de detalhamento mais próximo do que será o sistema em si.

## 4.1 CASOS DE USO

### 4.1.1. Diagrama de Casos de Uso

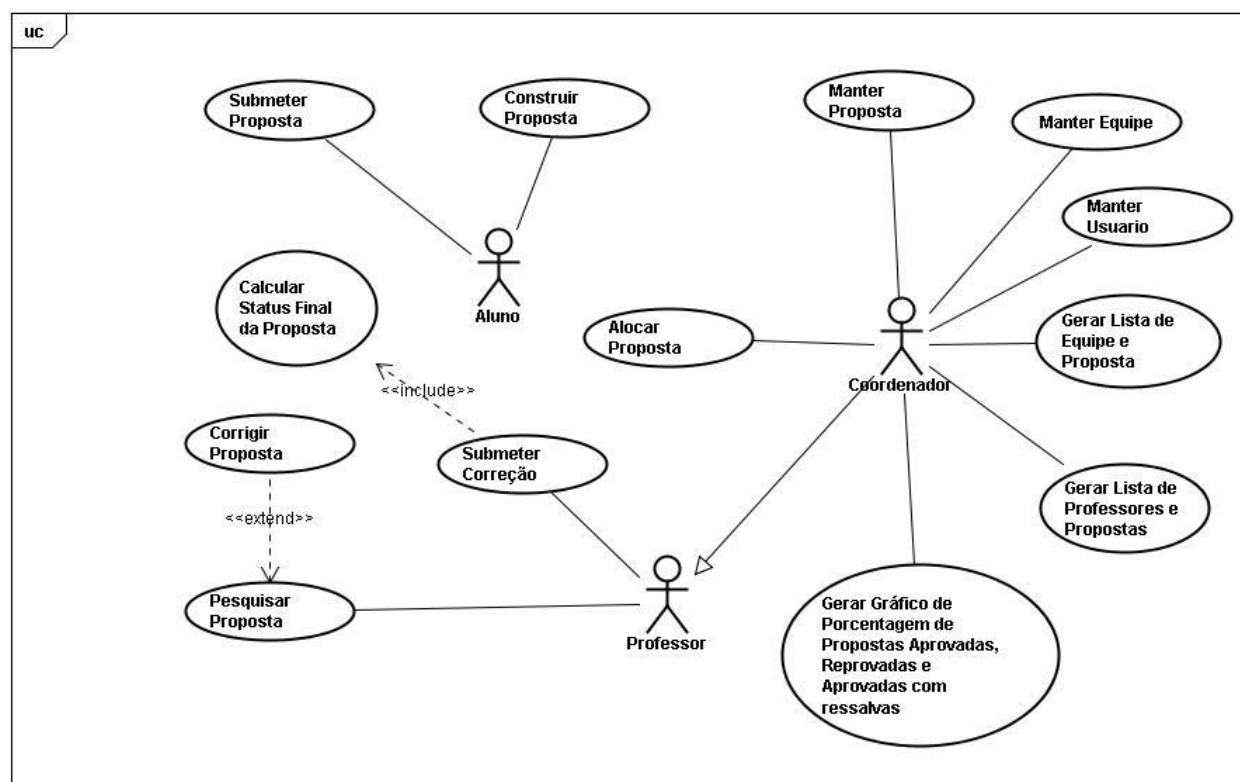


Figura 15 - Diagrama de Casos de Uso de Projeto

## 4.1.1.1.Descrição de Casos de Uso

## 4.1.1.1.1. Manter Proposta

<b>Caso de Uso:</b> Manter Propostas	<b>Sigla:</b> UC01
<b>Objetivo:</b> Utilizar as operações de cadastrar, editar ou pesquisa.	
<b>Ator:</b> Coordenador.	
<b>Pré-condições:</b> Não aplicável	
<b>Pós-condições:</b> Não aplicável	
<b>Fluxo Principal</b> <ol style="list-style-type: none"> <li>1. Este caso de uso se inicia quando o ator aciona a opção cadastrar na tela homeCoordenador.</li> <li>2. O sistema lê na tabela PROPOSTA os campos id, nome, id_equipe e id_status_proposta.</li> <li>3. O sistema exibe a tela PesquisarProposta com uma lista de propostas carregadas em uma tabela com os campos Nome, Equipe, Status e Opções.</li> <li>4. O ator pode optar por Pesquisar, Nova Proposta, Editar ou Sair.</li> <li>5. Se a opção for Pesquisar, o sistema aciona o fluxo alternativo Pesquisar (FA(01)).</li> <li>6. Se a opção for Nova Proposta, o sistema aciona o fluxo alternativo Nova Proposta (FA(02)).</li> <li>7. Se a opção for Editar, o sistema aciona o fluxo alternativo Editar (FA(03)).</li> <li>8. Se a opção for Sair, o sistema aciona o fluxo alternativo Sair (FA(04)).</li> <li>9. Este caso de uso se encerra.</li> </ol>	
<b>Fluxo alternativo Pesquisar (FA01) ref. UC01(5)</b> <b>Pré condição:</b> Não aplicável. <b>Passos:</b> <ol style="list-style-type: none"> <li>1. O ator pode optar por pesquisar por Nome ou Equipe.</li> <li>2. O ator preenche campo nome ou equipe.</li> <li>3. O ator aciona o botão Pesquisar.</li> <li>4. O sistema lê na tabela PROPOSTA os campos id, nome, id_equipe e id_status_proposta.</li> <li>5. O sistema exibe na tabela a Proposta, equipe, status e o botão Editar ao lado da proposta.</li> <li>6. Este fluxo se encerra.</li> </ol>	
<b>Fluxo alternativo Nova Proposta (FA02) ref. UC01(6)</b> <b>Pré condição:</b> O ator acionou o botão Nova Proposta. <b>Passos:</b> <ol style="list-style-type: none"> <li>1. O sistema lê na tabela EQUIPE o campo nome.</li> <li>2. O sistema filtra as equipes que não possuem relação com propostas, ou seja.   <div style="text-align: center;">           ( select * from equipe e +            where e.id not in (select id_equipe from proposta)         </div> </li> </ol>	

<ol style="list-style-type: none"> <li>3. O sistema exibe a tela CadastroProposta com as equipes carregada no campo equipe.</li> <li>4. O ator preenche o campo Nome.</li> <li>5. O ator seleciona a equipe no campo Equipe.</li> <li>6. O ator preenche campo Período.</li> <li>7. O ator aciona o botão Salvar.</li> <li>8. O sistema lê a data corrente do servidor.</li> <li>9. O sistema salva os dados na tabela PROPOSTA nos campos id, nome, id_status recebe os status Em construção, data_criacao, id_equipe e periodo.</li> <li>10. O sistema exibe a mensagem: sucesso.cadastro.proposta_detail.</li> <li>11. O Este fluxo se encerra.</li> </ol>
<p><b>Fluxo alternativo Editar (FA03) ref. UC01(7)</b></p> <p><b>Pré condição:</b> O ator acionou o botão Editar.</p> <p><b>Passos:</b></p> <ol style="list-style-type: none"> <li>1. O sistema exibe a tela AlterarProposta com os campos nome, equipe, data_criacao e periodo carregados, sendo que somente o campo Nome estará em modo de edição.</li> <li>2. O ator altera o campo nome.</li> <li>3. O ator aciona o botão Salvar.</li> <li>4. O sistema atualiza na tabela PROPOSTA o campo nome.</li> <li>5. O sistema exibe a mensagem: sucesso.cadastro.proposta_detail.</li> <li>6. O sistema retorna para a tela PesquisarProposta.</li> </ol>
<p><b>Fluxo alternativo Sair (FA03) ref. UC01(1,3,4,5,6,7), FA01(1,2,3), FA02(4,5,6,7) e FA03(2,3).</b></p> <p><b>Pré condição:</b> Não aplicável.</p> <p><b>Passos:</b></p> <ol style="list-style-type: none"> <li>1. O ator aciona a opção Sair.</li> <li>2. O sistema retorna para a tela Login.</li> <li>3. Este fluxo se encerra.</li> </ol>
<p><b>Fluxo de exceção Campos Obrigatórios. (FE01) ref. FA02(4,6), FA03(2).</b></p> <p><b>Pré-condição:</b> Não Preencher campos obrigatórios, ou seja, campos com asterisco (*).</p> <p><b>Passos:</b></p> <ol style="list-style-type: none"> <li>1. O ator aciona o botão Salvar.</li> <li>2. O sistema exibe na tela a mensagem: Preenchimento Obrigatório.</li> <li>3. Este fluxo se encerra.</li> </ol>

#### 4.1.1.1.1. Alocar Propostas (Processamento 1)

<b>Caso de Uso:</b> Alocar Propostas.	<b>Sigla:</b> UC02
<b>Objetivo:</b> Alocar proposta dinamicamente para professores selecionados.	
<b>Ator:</b> Professor.	

**Pré-condições:** Todos os itens devem estar com status igual a corrigido.

**Pós-condições:** Proposta desabilitada para usuário aluno.

### Fluxo Principal

1. Este caso de uso se inicia quando o ator aciona a opção Alocar, na tela homeCoordenador.
2. O sistema lê na tabela USUARIO o campo nome, ou seja:

```
Select from Usuario as user
where user.tipoUsuario.id = 2
      or user.tipoUsuario.id = 3
order by user.nome asc
```

3. O sistema lê na tabela PROPOSTA o campo com id\_status=Concluido, ou seja:

```
Select * from Proposta as proposta
where proposta.status.id = 6
```

4. O sistema exibe a tela AlocarProposta com todos o professores cadastrados.
5. O ator seleciona a opção Grupo.
6. O ator seleciona um Professor.
7. O ator aciona o botão Adicionar.
8. O ator aciona o botão Alocar.
9. O sistema lê na tabela USUARIO os campos id e nome.
10. O sistema lê na tabela PERGUNTA os campos id e descricao.
11. O sistema lê na tabela ITEM os campos id e nome.
12. O sistema lê na tabela PROPOSTA campo id\_status\_proposta=concluido.
13. O sistema aloca as propostas para o professor selecionado.
14. O sistema salva na tabela CORRECAO os campos id\_proposta, id\_item, matricula\_professor, id\_pergunta, id\_status\_correcao=Em correcao.
15. O sistema atualiza na tabela PROPOSTA o campo id\_status\_proposta=Em correcao.
16. Este fluxo se encerra.

### Fluxo Alternativo Grupo e Dois Professores (FA01) ref. UC02(5,7,8,9,10,11,12)

**Pré condição:** O ator seleciona dois professores.

#### Passos:

1. O sistema aloca dinamicamente as propostas para os professores que foram selecionados, sendo que cada proposta é alocada para os dois professores.
2. O sistema salva na tabela CORRECAO os campos id\_proposta, id\_item, matricula\_professor, id\_pergunta, id\_status\_correcao=Em correcao.
3. O sistema atualiza na tabela PROPOSTA o campo id\_status\_proposta=Em correcao.
4. Este fluxo se encerra.

### Fluxo Alternativo Individual (FA02) ref. UC02(5)

**Pré condição:** Não aplicavel.



**Passos:**

1. O ator seleciona a opção Individual.
2. O ator seleciona dois Professores.
3. O ator aciona o botão Adicionar.
4. O ator aciona o botão Alocar.
5. O sistema lê na tabela USUARIO os campos id e nome.
6. O sistema lê na tabela PERGUNTA os campos id e descricao.
7. O sistema lê na tabela ITEM os campos id e nome.
8. O sistema lê na tabela PROPOSTA campo id\_status\_proposta=concluido.
9. O sistema aloca as propostas para o professor selecionado.
10. O sistema salva na tabela CORRECAO os campos id\_proposta, id\_item, matricula\_professor, id\_pergunta, id\_status\_correcao=Em correcao.
11. O sistema atualiza na tabela PROPOSTA o campo id\_status\_proposta=Em correcao.
12. Este fluxo se encerra.

**Fluxo Alternativo Individual (FA03) ref. UC02(5)**

**Pré condição:** Não aplicavel.

**Passos:**

1. O ator seleciona a opção Individual.
2. O ator seleciona dois Professores.
3. O ator aciona o botão Adicionar.
4. O ator aciona o botão Alocar.
5. O sistema lê na tabela USUARIO os campos id e nome.
6. O sistema lê na tabela PERGUNTA os campos id e descricao.
7. O sistema lê na tabela ITEM os campos id e nome.
8. O sistema lê na tabela PROPOSTA campo id\_status\_proposta=concluido.
9. O sistema verifica se a quantidade de proposta  
Se quantidade de proposta igual a par.

O sistema percorre a lista de proposta alocando as proposta alternando os professores,  
ou seja:

Proposta1 para o professor1.

Proposta2 para o professor2.

Proposta3 para o professor1.

Proposta4 para o professor2.

Senão

O sistema verifica

Se

Existe só uma proposta na lista.

O sistema aloca a proposta para o primeiro professor da lista

Senão

O sistema retira a primeira proposta da lista.

<p>O sistema percorre a lista de proposta alocando as proposta alternando os professores, ou seja:</p> <p>Proposta1 para o professor1.</p> <p>Proposta2 para o professor2.</p> <p>Proposta3 para o professor1.</p> <p>Proposta4 para o professor2.</p> <p>O sistema aloca randomicamente a proposta retirada para um professor.</p> <p>10. O sistema salva na tabela CORRECAO os campos id_proposta, id_item, matricula_professor, id_pergunta, id_status_correcao=Em correcao.</p> <p>11. O sistema atualiza na tabela PROPOSTA o campo id_status_proposta=Em correcao.</p> <p>12. Este fluxo se encerra.</p>
<p><b>Fluxo Alternativo Sair (FA04) ref. UC02(1,5,6,7,8), FA02(1,2,3,4) e FA03(1,2,3,4)</b></p> <p><b>Pré condição:</b> O ator seleciona a opção Sair.</p> <p><b>Passos:</b></p> <ol style="list-style-type: none"> <li>1. O sistema retorna para a tela Login.</li> <li>2. Este fluxo se encerra.</li> </ol>
<p><b>Fluxo Exceção Selecionar mais de dois Professores para alocar (FE01) ref. UC02(6), FA02(2) e FA03(2).</b></p> <p><b>Pré condição:</b> O ator seleciona mais de dois professores.</p> <p><b>Passos:</b></p> <ol style="list-style-type: none"> <li>1. O sistema exibe a mensagem: erro.maximo.professor.selecionado_detail.</li> <li>2. Este fluxo se encerra.</li> </ol>
<p><b>Fluxo Exceção Não Selecionou nenhum professor (FE02) ref. UC02(6), FA02(2) e FA03(2).</b></p> <p><b>Pré condição:</b> O ator não selecionou nenhum professor.</p> <p><b>Passos:</b></p> <ol style="list-style-type: none"> <li>1. O sistema exibe a mensagem: erro.professor.nao.selecionado_detail.</li> <li>2. Este fluxo se encerra.</li> </ol>

#### 4.1.1.1.2. Submeter Correção

**Caso de Uso:** Submeter Correção.

**Sigla:** UC03

**Objetivo:** Processar status da proposta, alterando para Aprovado, Reprovado ou Aprovado com Ressalva.

**Ator:** Professor.

**Pré-condições:** Todos os itens devem estar com status igual a corrigido.

**Pós-condições:** Proposta desabilitada para correção.

### **Fluxo Principal**

1. Este caso de uso se inicia quando o ator aciona o botão Submeter, na tela SubmeterCorrecao.
2. O sistema lê na tabela CORRECAO os campos id\_proposta, id\_item, matricula\_professor, id\_pergunta, resposta e id\_status\_correcao.

Se a resposta for igual a Não para as perguntas:

(Sistema pertencente ao grupo de trabalhos aceitos)

**ou**

(Justificativa clara, com texto bem organizado e, enfatizando a relevância e a problemática que implica na necessidade do desenvolvimento do Sistema)

**e**

(Lista de Funções bem definidas, demonstrando, no mínimo, 7(sete) requisitos do sistema (entrada, processamento e saída). Sendo pelo menos 3 de processamento)

**e**

(Metodologia coerente com os objetivos definidos e adequada abrangência do tema), ou seja:

((id\_resposta=2 && id\_pergunta=7) || ((id\_resposta=2 && id\_pergunta=3) && (id\_resposta=2 && id\_pergunta=5) && (id\_resposta=2 && id\_pergunta=6))).

O sistema armazena id\_status\_proposta=Reprovado em um objeto proposta em tempo de execução.

O sistema executa o subfluxo Calcula Status Final da proposta.

Senão

Se a resposta for igual Não para as perguntas:

(Justificativa clara, com texto bem organizado e, enfatizando a relevância e a problemática que implica na necessidade do desenvolvimento do Sistema)

**ou**

(Lista de Funções bem definidas, demonstrando, no mínimo, 7 (sete) requisitos do sistema (entrada, processamento e saída). Sendo pelo menos 3 de processamento)

**ou**

(Metodologia coerente com os objetivos definidos e adequada abrangência do tema)

**e**

Resposta igual a Sim para a pergunta:

(Sistema pertencente ao grupo de trabalhos aceitos), ou seja:

((id\_resposta=2 && id\_pergunta=3) || id\_resposta=2 && id\_pergunta=5 || id\_resposta=2 && id\_pergunta=6) && (id\_resposta=1 && id\_pergunta=7)).

O sistema armazena id\_status\_proposta=Aprovado com Ressalvas em um objeto proposta em tempo de execução.

O sistema executa o subfluxo Calcula Status Final da proposta.

Senão

O sistema armazena id\_status\_proposta=Aprovado em um objeto proposta em tempo de execução

O sistema executa o subfluxo Calcula Status Final da proposta.

3. O sistema desabilita a proposta submetida.
4. O sistema exibe a mensagem: sucesso.submeter.proposta\_detail
5. Este fluxo se encerra.

### **Subfluxo calcular Status Final Proposta (FA02) ref. UC03(6)**

**Pré condição:** Não aplicável.

**Passos:**

1. O sistema lê id\_status\_proposta armazenado no objeto proposta em tempo de execução.
2. O sistema lê na tabela PROPOSTA campo id\_status\_proposta.  
Se id\_status\_proposta=Em correcao;

O sistema atualiza na tabela PROPOSTA o campo id\_status\_proposta para o id\_status\_proposta armazenado no objeto proposta.

Senão se( Id\_status\_proposta = 3)(tabela PROPOSTA)

Se id\_status da memória virtual for id\_status\_proposta=3 || id\_status\_proposta=4

O sistema atualiza na tabela PROPOSTA, campo id\_status = 3;

Senao

O sistema atualiza na tabela proposta, campo id\_status\_proposta =3 para id\_status\_proposta=4;

Senão se( Id\_status\_proposta = 4) (tabela PROPOSTA)

Se id\_status da memória virtual for id\_status\_proposta =3;

O sistema atualiza na tabela proposta, campo id\_status\_proposta=4 para id\_status\_proposta=3;

Senão se Id\_status\_proposta da memória virtual for id\_status\_proposta =4

O sistema atualiza na tabela proposta, campo id\_status\_proposta=4 para id\_status\_proposta =4;

Senão

O sistema atualiza na tabela proposta, campo id\_status\_proposta=4 para id\_status\_proposta =5;

Senão se( Id\_status\_proposta = 5) (tabela PROPOSTA)

Se id\_status\_proposta da memória virtual for id\_status\_proposta=4 || id\_status\_proposta=5

O sistema atualiza na tabela proposta, campo id\_status\_proposta=5;

Senao

O sistema atualiza na tabela proposta, campo id\_status\_proposta=3 para id\_status\_proposta=4;

3. Este fluxo se encerra.

### **Fluxo Alternativo Sair (FA01) ref. UC03(1)**

**Pré condição:** Não aplicável.

**Passos:**

1. O ator aciona a opção Sair.
2. O sistema retorna para tela Login.
3. Este fluxo se encerra.

### **Fluxo de Exceção Itens não Corrigido (FE01) ref. UC03(2)**

**Pré condição:** Item com status igual à em correção.

**Passos:**

1. O sistema lê na tabela PROPOSTA\_ITEM o campo id\_status\_criacao = Em correção.
2. O sistema exibe msg: itens.nao.concluidos.
3. Este fluxo se encerra.

#### 4.1.1.1.3. Gerar Gráfico de Porcentagem de Propostas Aprovadas, Reprovadas e Aprovadas com Ressalvas

<b>Caso de Uso:</b> Gerar Gráfico de Porcentagem de Propostas aprovadas, Reprovadas e Aprovadas com ressalvas.	<b>Sigla:</b> UC04
<b>Objetivo:</b> Obter uma visão geral da porcentagem de Propostas Aprovadas, Aprovadas com Ressalvas e Reprovadas.	
<b>Ator:</b> Coordenador.	
<b>Pré-condições:</b> O ator deve ter acionado a opção Relatórios.	
<b>Pós-condições:</b> Não aplicável	
<b>Fluxo Principal</b> <ol style="list-style-type: none"> <li>1. Este caso de uso se inicia quando o ator aciona a opção Relatórios na tela homeCoordenador.</li> <li>2. O sistema exibe a tela Relatórios.</li> <li>3. O ator seleciona a opção Relatório Resultados de propostas.</li> <li>4. O ator preenche campo Período.</li> <li>5. O ator aciona o botão Visualizar.</li> <li>6. O sistema lê na tabela PROPOSTA os campos id_status_proposta que estão iguais a aprovado, aprovado com ressalva ou reprovada e o campo periodo,ou seja: <p>Select from Proposta as proposta</p> <p>where proposta.status.id = 3</p> <p>or proposta.status.id = 4</p> <p>or proposta.status.id = 5</p> <p>and proposta.periodo = periodo</p> <p>Group by proposta.id</p> </li> <li>7. O sistema gera uma lista de proposta do período preenchido no passo 4.</li> <li>8. O sistema percorre a lista.</li> <li>9. O sistema verifica na lista. <p>Se</p> <p>id_status_proposta = Aprovado.</p> <p>O sistema lê a quantidade de proposta aprovadas (Qtd AP) e divide pela quantidade</p> </li> </ol>	

de proposta da lista (Qtd total) e multiplica por 100, ou seja:

$$\text{PorcentagemAp} = \text{Qtd AP} / \text{Qtd Total}.$$

Senão

Se id\_status\_proposta = Reprovado.

O sistema lê a quantidade de proposta reprovado (Qtd RP) e divide pela quantidade de proposta da lista (Qtd total) e multiplica por 100, ou seja:

$$\text{PorcentagemRp} = \text{Qtd RP} / \text{Qtd Total}.$$

Senão

O sistema lê a quantidade de proposta aprovado com ressalvas (Qtd APR) e divide pela quantidade de proposta da lista (Qtd total) e multiplica por 100, ou seja:

$$\text{PorcentagemAPR} = \text{Qtd APR} / \text{Qtd Total}.$$

10. O sistema cria uma lista de valores (PorcentagemAp, PorcentagemRp e PorcentagemAPR).
11. O sistema cria uma lista de nomes (Aprovados, Reprovados e Aprovados com Ressalvas).
12. O sistema lê a lista de valores e nomes.
13. O sistema cria o gráfico pizza.
14. O sistema atribui valores com seus respectivos nomes no gráfico.
15. O sistema exibe a tela Resultado Propostas em forma PDF no gráfico pizza.
16. Este fluxo se encerra.

#### 4.1.1.1.2. Gerar Lista de Equipes e Propostas

<b>Caso de Uso:</b> Gerar Lista de Equipes e Propostas	<b>Sigla:</b> UC05
<b>Objetivo:</b> Listar as equipes, suas propostas e os status das propostas.	
<b>Ator:</b> Coordenador.	
<b>Pré-condições:</b> O ator deve ter acionado a opção Relatórios.	
<b>Pós-condições:</b> Não aplicável	
<b>Fluxo Principal</b> <ol style="list-style-type: none"> <li>1. Este caso de uso se inicia quando o ator aciona a opção Relatórios na tela homeCoordenador.</li> <li>2. O sistema exibe a tela Relatorios.</li> <li>3. O ator seleciona o Relatório Proposta x Equipe.</li> <li>4. O ator preenche campo período.</li> </ol>	

5. O ator aciona o botão Visualizar.
6. O sistema lê na tabela PROPOSTA os campos id, nome, id\_status\_proposta, data\_criacao, data\_submissao, id\_equipe e periodo, ou seja:

SELECT

    proposta.nome AS proposta\_nome,  
    proposta.data\_criacao AS proposta\_data\_criacao,  
    proposta.data\_submissao AS proposta\_data\_submissao,  
    proposta.periodo AS proposta\_periodo,  
    equipe.nome AS equipe\_nome,  
    status.nome AS status\_nome

FROM

    equipe equipe INNER JOIN proposta proposta ON equipe.id = proposta.id\_equipe  
    INNER JOIN status status ON proposta.id\_status\_proposta = status.id  
    AND status.id = proposta.id\_status\_proposta

WHERE

    proposta.periodo = periodo

ORDER BY

    equipe.nome ASC;

7. O sistema exibe a tela Relatório Equipe x Proposta em forma de PDF.
8. Este fluxo se encerra.



## 4.2. DIAGRAMAS

Optamos por apresentar os diagramas de classes separados por pacotes para uma melhor apresentação.

## 4.2.1. Classes persistentes

Esse diagrama visa mostrar as principais classes do sistema onde operam as regras de negócio do sistema.

Este diagrama também está em formato A3 para melhor visualização.

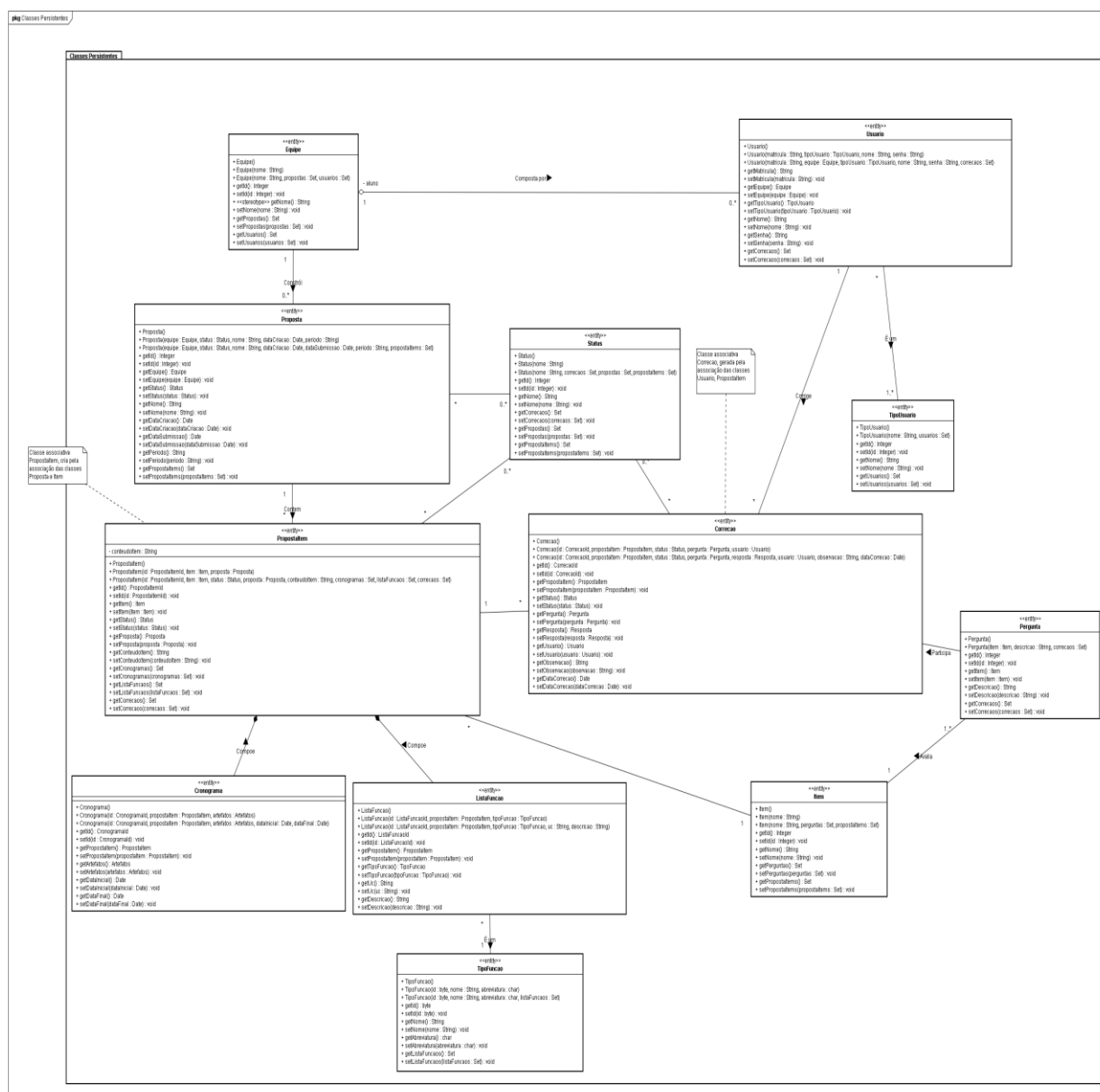


Figura 16 - Diagrama de Classe de Projeto

#### 4.2.2. Classes de Fachada e Dao

Essa figura abaixo visa mostrar as principais classes Dao do sistema juntamente com suas interface. As classes Dao são responsáveis por fazer as transações no banco de dados e as interfaces são criadas para servirem de comunicação para outras classes do sistema acessarem as classes Dao, encapsulando assim a forma como são feitas as transações no banco de dados.

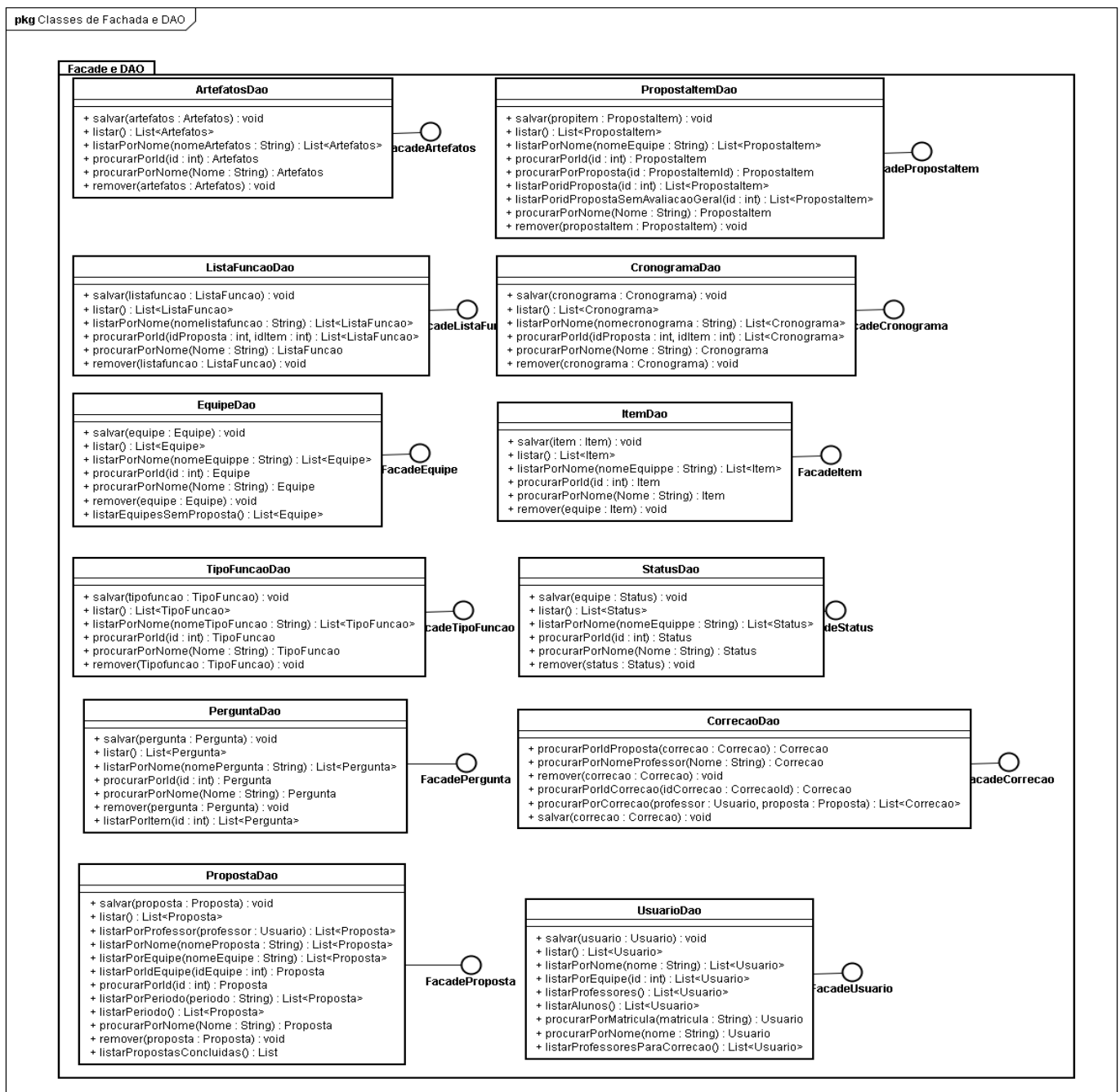


Figura 17 - Diagrama de Classe de Fachada e Dao

### 4.2.3. Classes de Processamento

Esse diagrama visa mostrar as classes que contém os métodos de processamento do sistema GerPro.

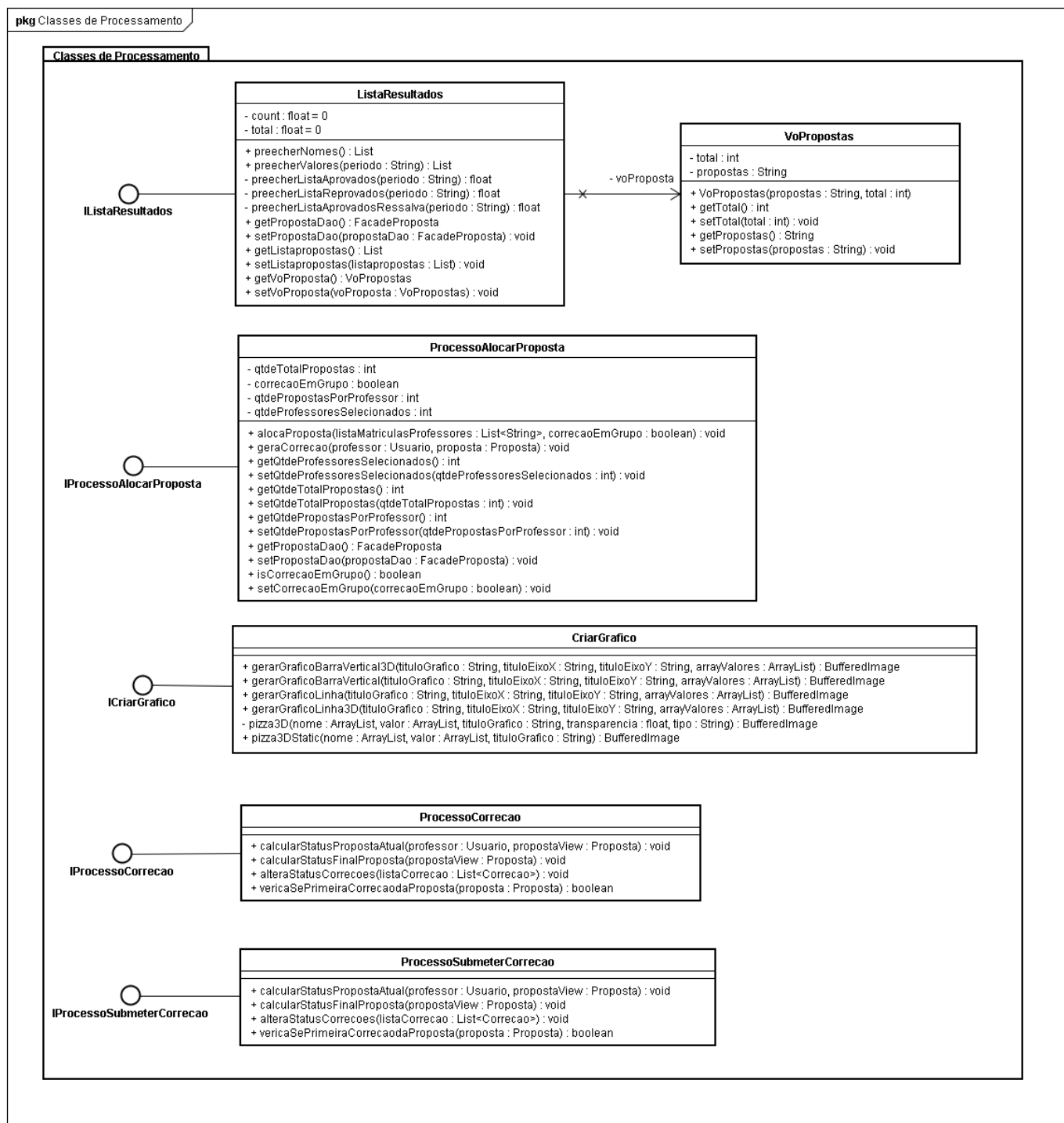
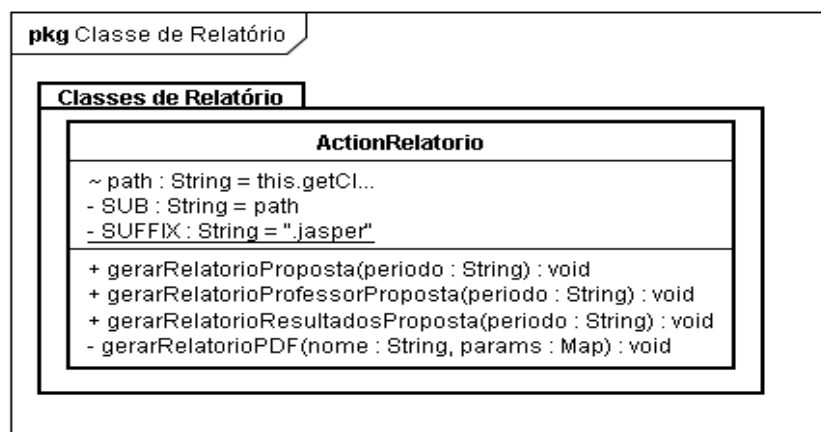


Figura 18 - Classe de Processamento



#### 4.2.5. Classe de Relatório

Essa figura mostra a classe que faz a criação do relatório do sistema.



**Figura 20 - Classe de Relatório**

#### 4.2.6. Classes – Da Arquitetura do Sistema

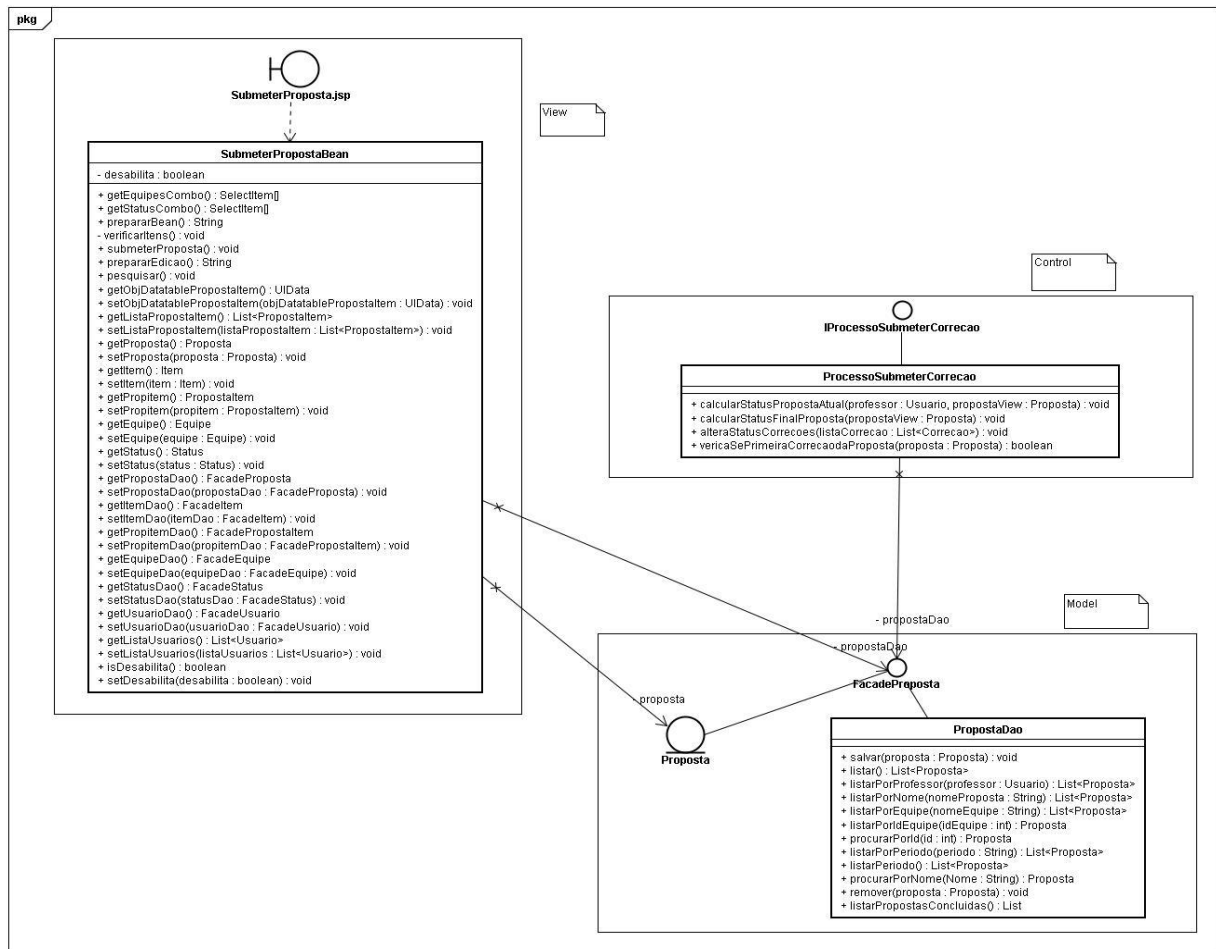


Figura 21 - Arquitetura do Sistema

## 4.2.7. Sequência Cadastrar Proposta

Este diagrama foi impresso em formato A3 para melhor visualização

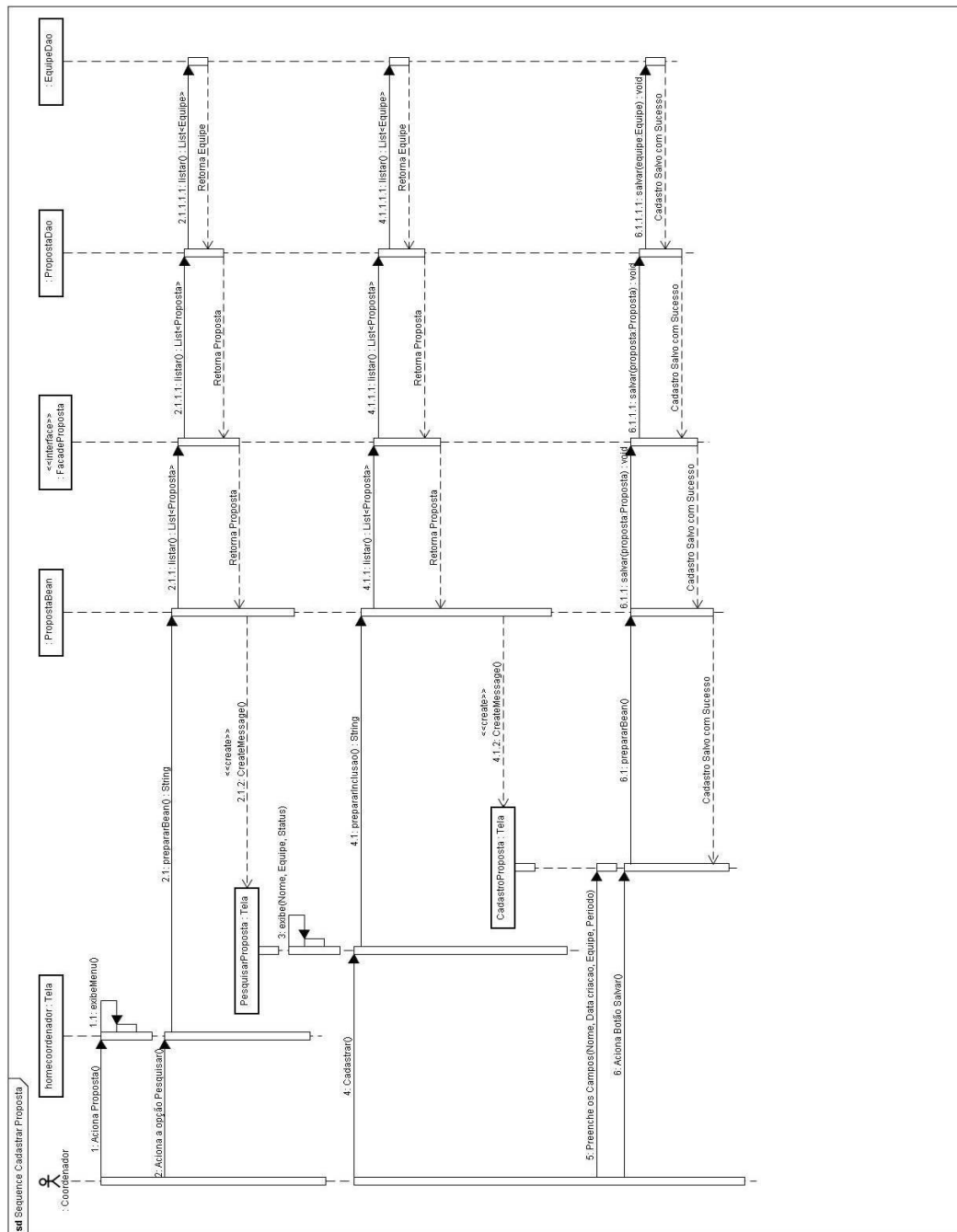


Figura 22 - Diagrama de Sequência Cadastrar Proposta (Projeto)



#### 4.2.8. Sequência Pesquisar Proposta

Este diagrama foi impresso em formato A3 para melhor visualização

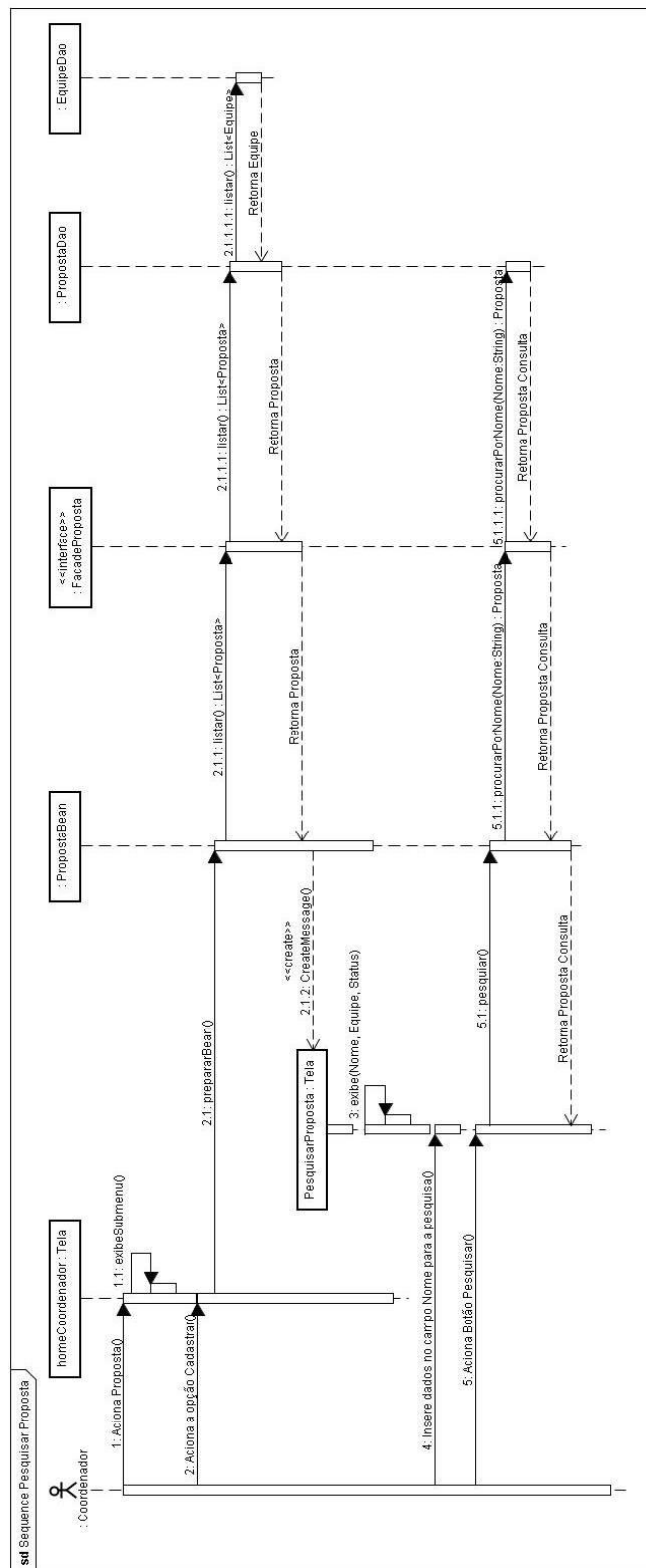


Figura 23 - Diagrama de Sequência Pesquisar Proposta (Projeto)

## 4.2.9. Sequência Editar Proposta

Este diagrama foi impresso em formato A3 para melhor visualização

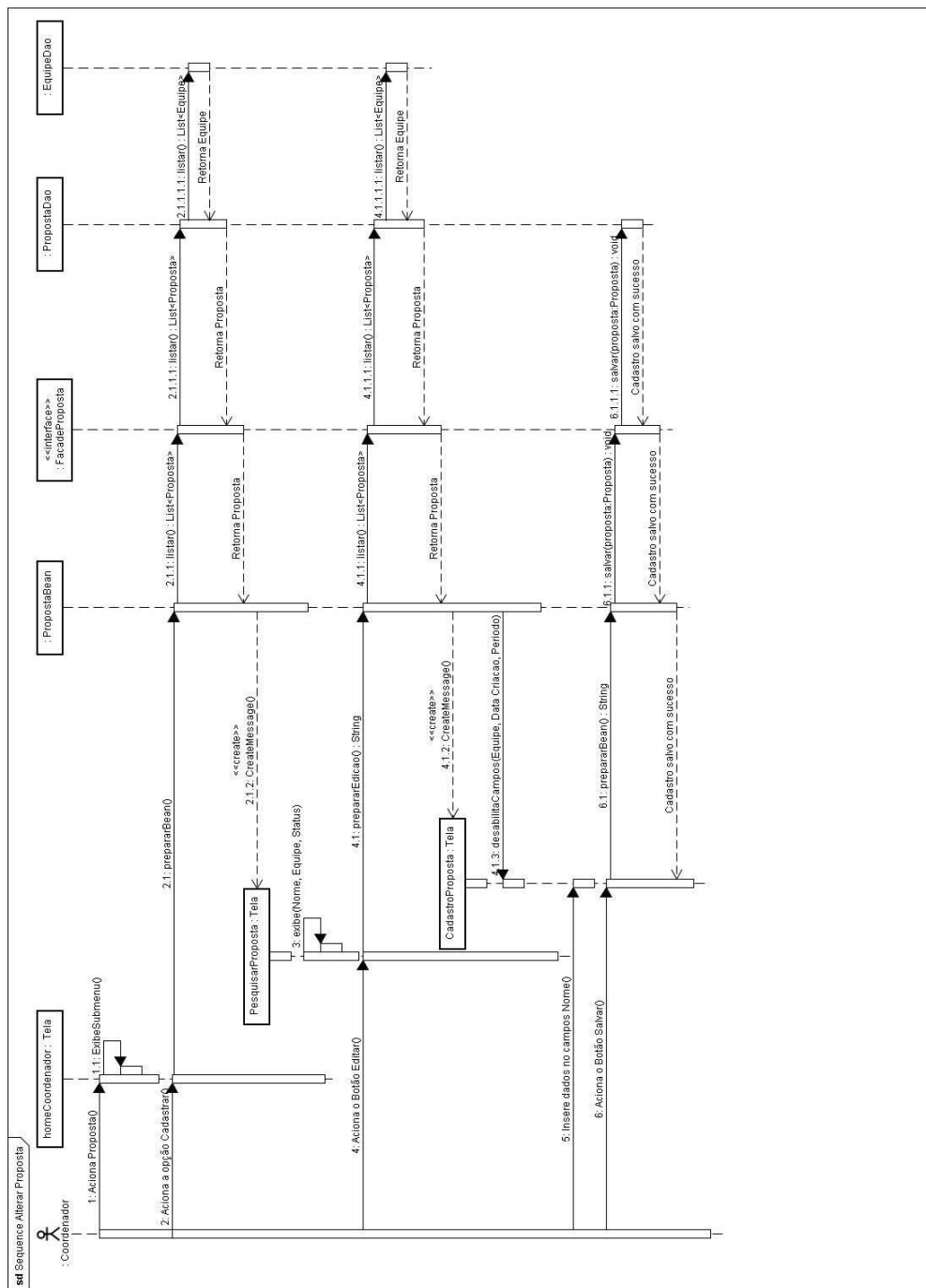


Figura 24 - Diagrama de Sequência Editar Proposta (Projeto)

#### 4.2.10. Seqüência Alocar Proposta

Este diagrama foi impresso em formato A3 para melhor visualização

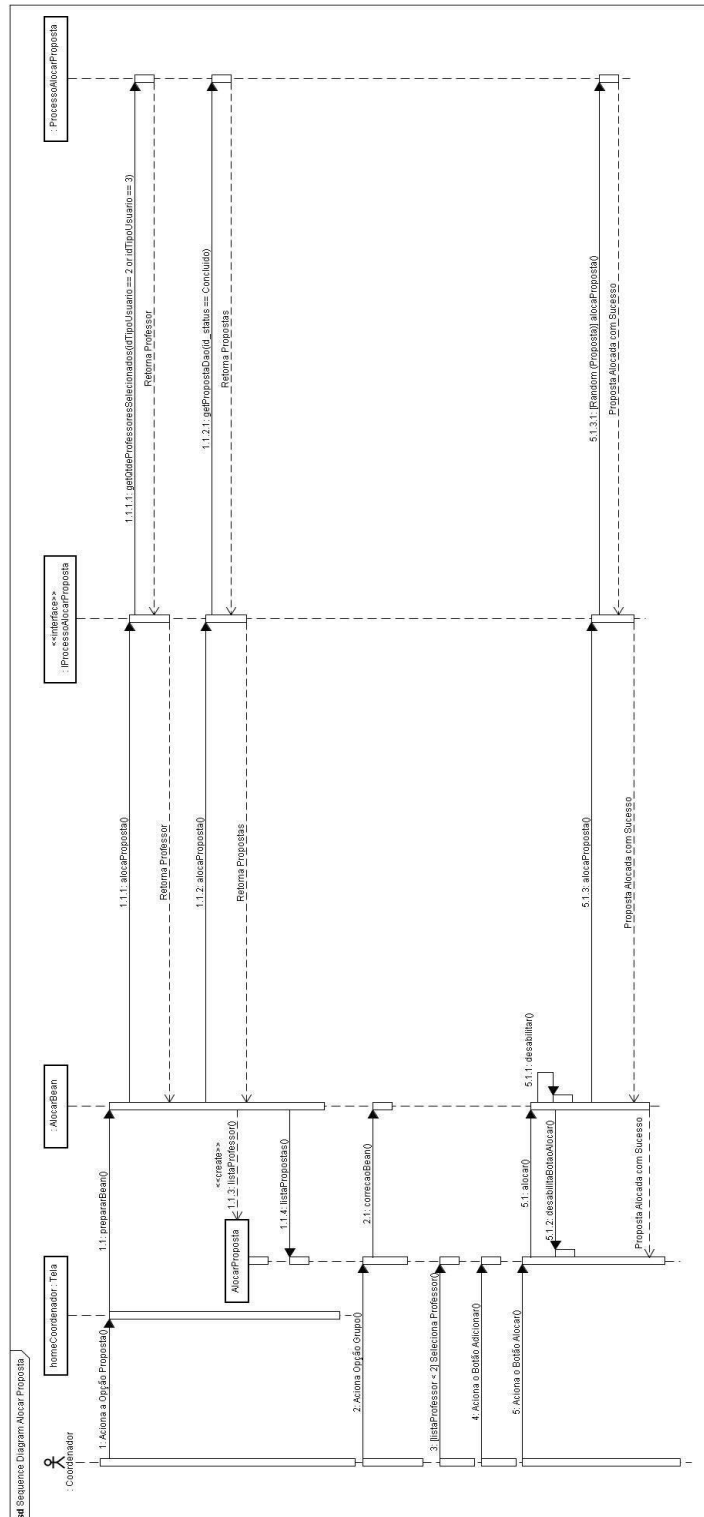


Figura 25 - Diagrama de Seqüência Alocar Proposta (Projeto)

#### 4.2.11. Seqüência Submeter Correção

Este diagrama foi impresso no formato A3 para melhor visualização.

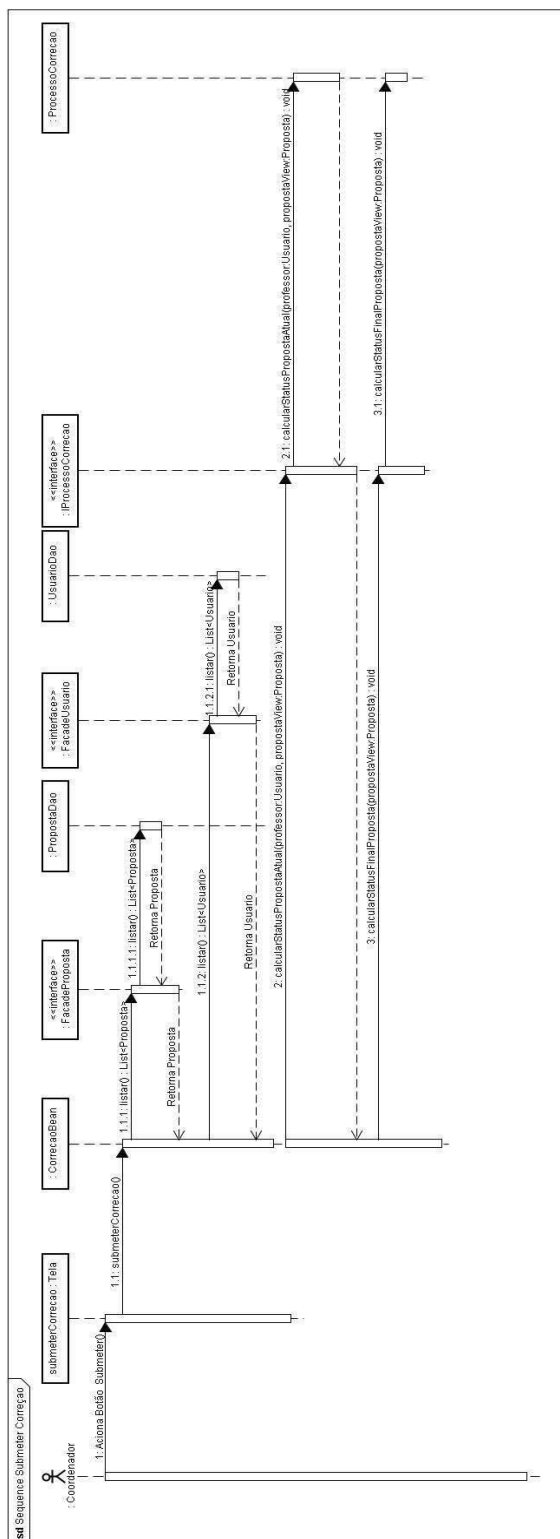
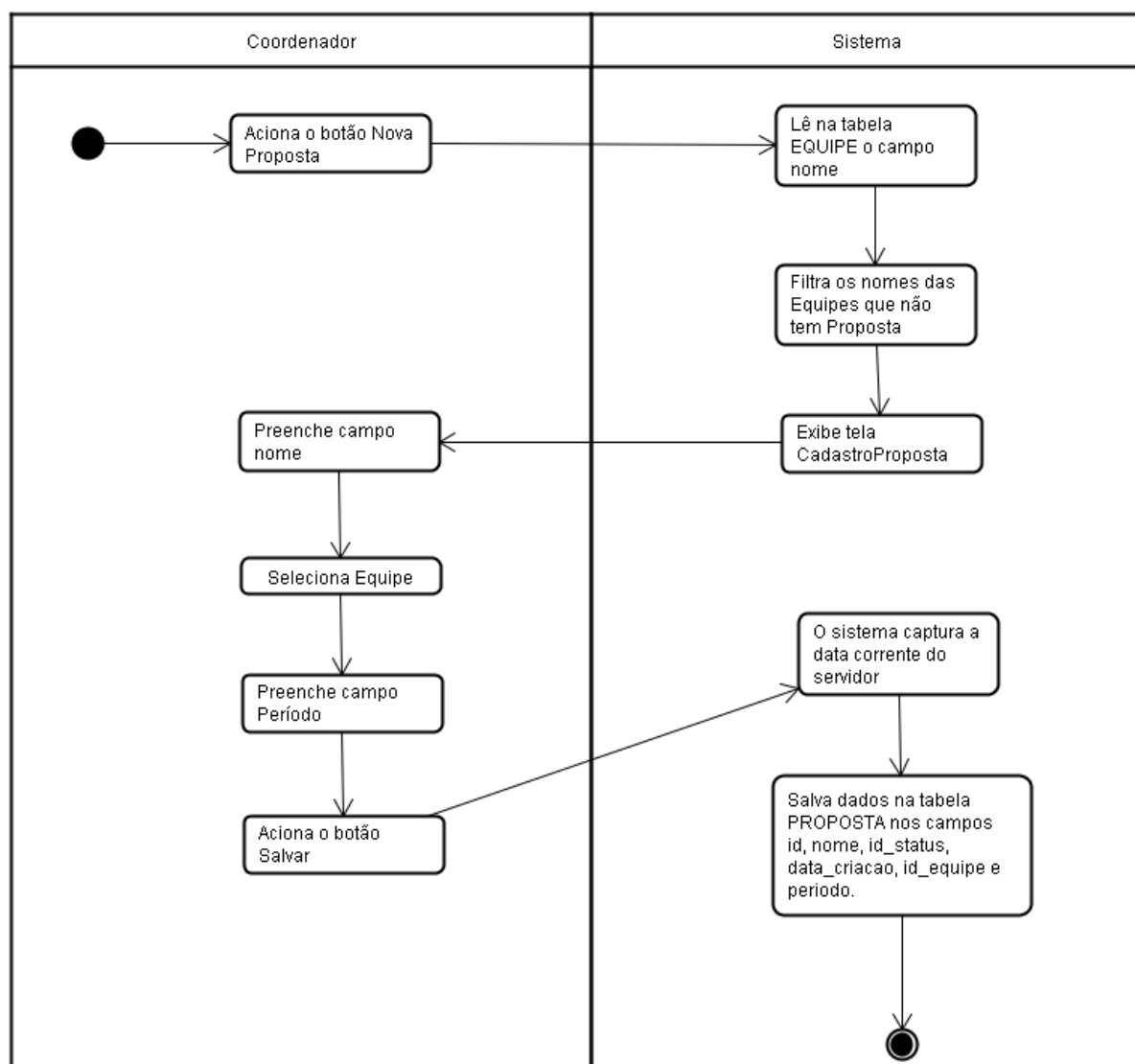


Figura 26 - Diagrama de Seqüência Submeter Correção (Projeto)

**Figura 27 - Diagrama de Seqüência do Relatório de Porcentagem (Projeto)**

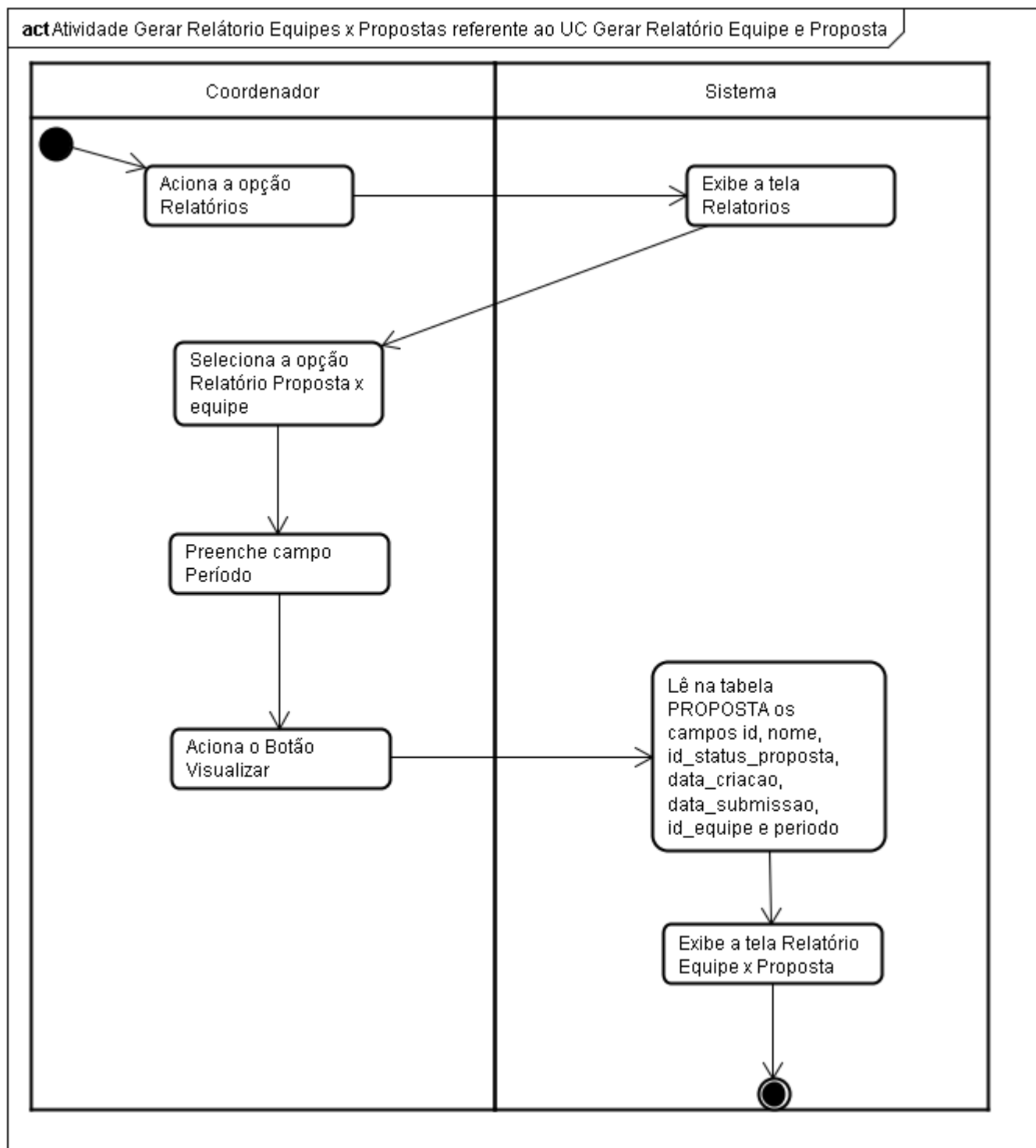


#### 4.2.14. Atividade – Cadastrar Proposta



**Figura 29 - Diagrama de Atividade Cadastrar Proposta (Projeto)**

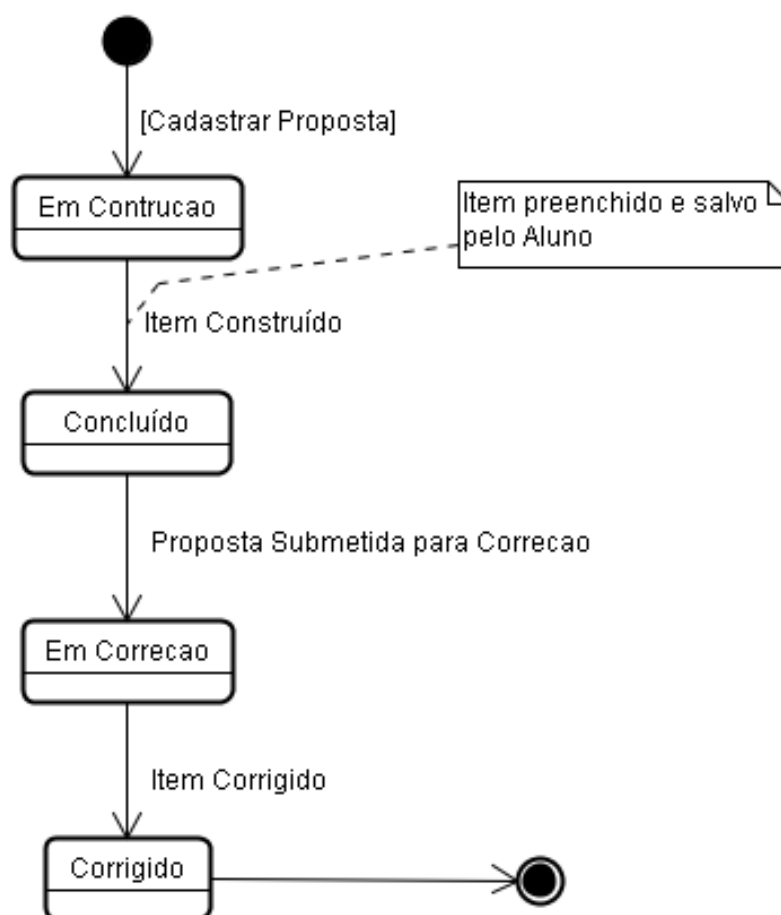
#### 4.2.15. Atividade Gerar Relatório de Equipe e Propostas



**Figura 30 - Diagrama de Atividade Gerar Relatório de Equipe X Proposta (Projeto)**



#### 4.2.16. Estado – Status do Item



**Figura 31 - Diagrama de Estados do Status do Item (Projeto)**

#### 4.2.17. Estado – Status da Proposta

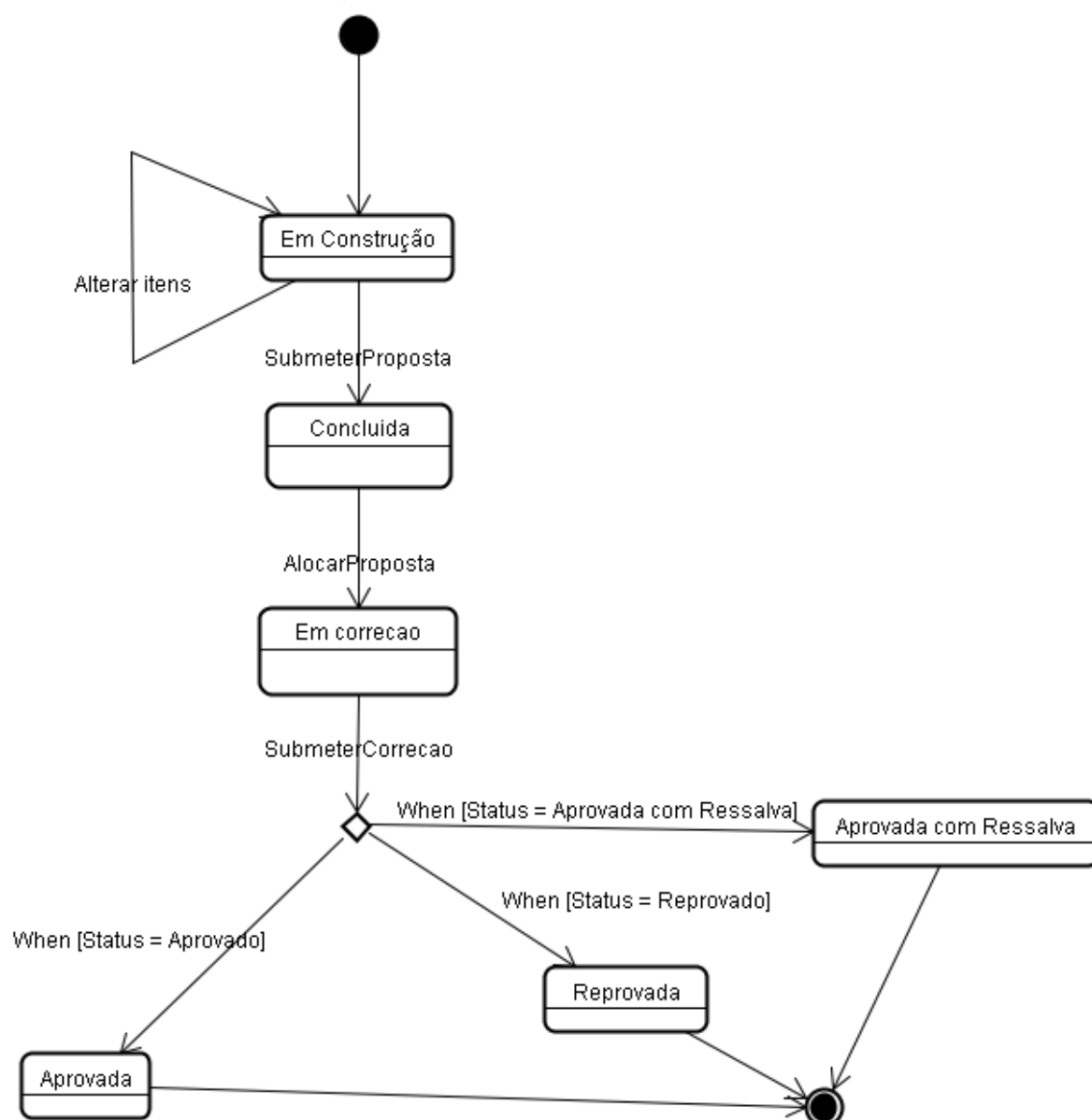


Figura 32 - Diagrama de Estados do Status da Proposta (Projeto)



### 4.3. PROJETO DE BANCO DE DADOS

#### 4.3.1. Dicionário de Dados

**Tabela:** Artefatos

**Objetivo:** Armazenar dados referentes à tabela artefatos.

Nome Coluna	Tipo	Tam.	NULL	PK	FK	Tab. Origem	Descrição Domínio
Id	Integer	-	N	S	-	-	Atributo único que define a artefato
nome	varchar	50	N	-	-	-	Nome do artefato.

Tabela: Equipe

**Objetivo:** Armazenar dados referentes à tabela equipe.

Nome Coluna	Tipo	Tam.	NULL	PK	FK	Tab. Origem	Descrição Domínio
Id	Integer	-	N	S	-	-	Atributo único que define a equipe
nome	varchar	50	N	-	-	-	Nome da equipe.

Tabela: Item

**Objetivo:** Armazenar dados referentes à tabela item.

Nome Coluna	Tipo	Tam.	NULL	PK	FK	Tab. Origem	Descrição Domínio
Id	Integer	-	N	S	-	-	Atributo único que define o item
nome	varchar	30	-	-	-	-	Nome do item.

**Tabela:** Resposta

**Objetivo:** Armazenar dados referentes à tabela resposta da correção dos professores para os itens da proposta.

<b>Nome Coluna</b>	<b>Tipo</b>	<b>Tam.</b>	<b>NULL</b>	<b>PK</b>	<b>FK</b>	<b>Tab. Origem</b>	<b>Descrição Domínio</b>
Id	Integer	-	N	S	-	-	Atributo único que define a resposta
descricao	varchar	10	-	-	-	-	Descrição referente à resposta.

**Tabela:** Status

**Objetivo:** Armazenar dados referentes à tabela status para os itens da proposta e para a proposta.

<b>Nome Coluna</b>	<b>Tipo</b>	<b>Tam.</b>	<b>NULL</b>	<b>PK</b>	<b>FK</b>	<b>Tab. Origem</b>	<b>Descrição Domínio</b>
Id	Integer	-	N	S	-	-	Atributo único que define o Status
nome	varchar	30	-	-	-	-	Nome do Status

**Tabela:** Pergunta

**Objetivo:** Armazenar dados referentes à tabela pergunta para avaliação dos itens da proposta e da proposta.

<b>Nome Coluna</b>	<b>Tipo</b>	<b>Tam.</b>	<b>NULL</b>	<b>PK</b>	<b>FK</b>	<b>Tab. Origem</b>	<b>Descrição Domínio</b>
Id	Integer	-	N	S	-	-	Atributo único que define a Pergunta
descricao	varchar	255	-	-	-	-	Descrição referente à Pergunta.
Id_item	integer	-	-	-	S	Item	Identificador do Item para a pergunta.

Tabela: Tipo\_usuario

Objetivo: Armazenar dados referentes à tabela Tipo\_usuario do sistema.

<b>Nome Coluna</b>	<b>Tipo</b>	<b>Tam.</b>	<b>NULL</b>	<b>PK</b>	<b>FK</b>	<b>Tab. Origem</b>	<b>Descrição Domínio</b>
Id	Integer	-	N	S	-	-	Atributo único que define o Tipo_usuario
nome	varchar	20	N	-	-	-	Nome do Usuário.

Tabela: Usuário

Objetivo: Armazenar dados referentes à tabela Usuario.

<b>Nome Coluna</b>	<b>Tipo</b>	<b>Tam.</b>	<b>NULL</b>	<b>PK</b>	<b>FK</b>	<b>Tab. Origem</b>	<b>Descrição Domínio</b>
matricula	varchar	8	N	S	-	-	Atributo único que define o usuário e o dá poderes de acesso ao sistema.
Nome	varchar	80	N	-	-	-	Nome do Usuário.
Senha	varchar	32	N	-	-	-	Senha do Usuário

							de acesso ao sistema.
Id_equipe	integer	-	-	-	S	Equipe	Identificador da Equipe do Usuário.
Id_tipo	Integer	-	N	-	S	Tipo_Usuario	Identificador do Tipo do Usuário.

Tabela: tipo\_funcao

Objetivo: Armazenar dados referentes à tabela tipo\_funcao para descrição do tipo de funcionalidade da proposta.

Nome Coluna	Tipo	Tam.	NULL	PK	FK	Tab. Origem	Descrição Domínio
Id	TINYINT	4	N	S	-	-	Atributo único que define a tipo_funcao
Nome	varchar	80	N	-	-	-	Descrição referente à tipo_funcao.
Abreviatura	Char	1	N	-	-	-	Atributo que define uma abreviação para o tipo de funcionalidade.

Tabela: lista\_funcao

Objetivo: Armazenar dados referentes à tabela lista\_funcao.

Nome Coluna	Tipo	Tam.	NULL	PK	FK	Tab. Origem	Descrição Domínio
Id_proposta	integer	-	N	S	S	Proposta_item	Atributo composto que define o item lista_funcao.
Numero_sequencia	Integer	-	N	S	-	-	Atributo composto que define o item lista_função

Id_item	Integer	-	N	S	S	Proposta_item	Atributo composto que define o item lista_função
Id_tipo_funcao	tinyint	4	N	S	-	Tipo_Funcao	Identificador do Tipo de Função da lista de funções.
UC	varchar	100	S	-	-	-	Nome do Caso de Uso.
descricao	text	-	S	-	-	-	Descrição da funcionalidade.

Tabela: cronograma

Objetivo: Armazenar dados referentes à tabela cronograma.

<b>Nome Coluna</b>	<b>Tipo</b>	<b>Tam.</b>	<b>NULL</b>	<b>PK</b>	<b>FK</b>	<b>Tab. Origem</b>	<b>Descrição Domínio</b>
Id_proposta	integer	-	N	S	S	Proposta_item	Atributo composto que define o item cronograma.
Id_item	integer	-	N	S	-	Proposta_item	Atributo composto que define o item cronograma.
Numero_sequencia	Integer	-	N	S	-	-	Atributo composto que define uma sequência no cronograma
Data_inicial	Date	-	S	-	-	-	Data de inicio das atividades na proposta.
Data_final	Date	-	S	-	-	-	Data final das atividades na proposta.
id_artefatos	Integer	-	N	-	S	Artefatos	Atributo que define o artefato do Cronograma



Tabela: Proposta

Objetivo: Armazenar dados referentes à tabela Proposta.

Nome Coluna	Tipo	Tam	NUL L	PK	FK	Tab. Origem	Descrição Domínio
id	integer	-	N	S	-	-	Atributo único que define a Proposta.
nome	varchar	100	N	-	-	-	Nome da Proposta.
Id_status_proposta	integer	-	N	-	S	Status	Identificador do status da proposta.
Data_criacao	Varchar	10	N	-	-	-	Data da criação da Proposta.
Data_submissao	Varchar	10	N	-	-	-	Data da submissão da Proposta.
periodo	Varchar	8	N	-	-	-	Identificador para o Período em que a proposta foi criada.
Id_equipe	integer	-	N	-	S	Equipe	Identificador da Equipe que construirá a proposta.

Tabela: Proposta\_item

Objetivo: Armazenar dados referentes à tabela Propostas\_Item, tabela resultante da relação entre a tabela de proposta e dos itens.

Nome Coluna	Tipo	Tam.	NULL	PK	FK	Tab. Origem	Descrição Domínio
Id_proposta	integer	-	N	S	S	Proposta	Atributo Composto que define a Proposta_Item.
Id_item	integer	-	N	S	S	Item	Atributo Composto que define a Proposta_Item.
Conteudo_Item	text	-	S	-	-	-	Armazena o conteúdo do item da proposta.
Id_status_criacao	integer	-	S	-	S	Status	Identificador do status do item da

							proposta.
--	--	--	--	--	--	--	-----------

Tabela: Correção

Objetivo: Armazenar dados referentes à tabela Correção.

Nome Coluna	Tipo	Tam .	NUL L	PK	FK	Tab. Origem	Descrição Domínio
Id_proposta	integer	-	N	S	S	proposta_item	Atributo Composto que define a correcao.
Id_item	integer	-	N	S	S	proposta_item	Atributo Composto que define a correcao.
Matricula_professor	varchar	8	N	S	S	Usuario	Atributo Composto que define a correcao.
Id_pergunta	integer	-	N	S	S	Pergunta	Atributo Composto que define a correcao.
observacao	text	-	-	-	-	-	Observações referentes as respostas da correcao.
Data_correcao	Date	-	-	-	-	-	Data da correção da correcao.
resposta	integer	-	-	-	S	Resposta	Identificador para a resposta dada à correcao.
Id_status_correcao	integer	-	N	-	S	Status	Identificador para o status dado à correcao

#### 4.4. MAPEAMENTO DE DADOS

PK (#) – PRIMARY KEY(CHAVE PRIMÁRIA).

FK (\*) – FOREIGN KEY(CHAVE ESTRANGEIRA).

## **TABELAS**

ARTEFATOS – (id#, nome)

PK – id.

EQUIPE – (id#, nome)

PK – id.

ITEM – (id#, nome)

PK – id.

RESPOSTA – (id#, descricao)

PK – id.

STATUS – (id#, nome)

PK – id.

PERGUNTA – (id#, descrição, id\_item)

PK – id.

FK – id\_item, referente à tabela ITEM.

TIPO\_USUARIO – (id#, nome)

PK – id.

USUARIO – (matricula#, nome, senha, email, id\_tipo\*, id\_eqp\*)

PK – matricula.

FK – id\_tipo, referente à tabela TIPO\_USUARIO.

FK – id\_eqp, referente à tabela EQUIPE.

TIPO\_FUNCAO – (id#, nome, abreviatura)

PK – id.

LISTA\_FUNCAO – (id\_proposta##, numero\_sequencia##, id\_item##, id\_tipo\_funcao, UC, descricao)

PK – id\_proposta, numero\_sequencia, id\_item.

FK – id\_item, referente à tabela PROPOSTA\_ITEM.

FK – id\_proposta, referente à tabela PROPOSTA\_ITEM.

CRONOGRAMA – (id\_proposta##, id\_item##, numero\_sequencia##, data\_inicial, data\_final, id\_artefato)

PK – id\_proposta, id\_item, numero\_sequencia.

FK – id\_item, referente à tabela PROPOSTA\_ITEM.

FK – id\_proposta, referente à tabela PROPOSTA\_ITEM.

FK – id\_artefato, referente à tabela ARTEFATOS.

PROPOSTA – (id#, nome, id\_status\_proposta\*, data\_criacao, data\_submissao, período, id\_equipe\*)

PK – pro\_nome.

FK – id\_status\_proposta, referente à tabela STATUS.

FK – id\_equipe, referente à tabela EQUIPE.

PROPOSTA\_ITEM – (id\_proposta##, id\_item##, conteúdo\_item, id\_status\_criacao)

PK – id\_proposta, id\_item.

FK – id\_proposta, referente à tabela PROPOSTA.

FK – id\_item, referente à tabela ITEM.

CORRECAO – (id\_proposta##, id\_item##, matricula\_professor##, id\_pergunta, observações, data\_correcao, resposta, id\_status\_correcao)

PK – id\_proposta, id\_item, matricula\_professor, id\_pergunta.

FK – id\_proposta, referente à tabela PROPOSTA\_ITEM.

FK – id\_item, referente à tabela PROPOSTA\_ITEM.

FK – matricula\_professor, referente à tabela USUARIO.

FK – id\_pergunta, referente à tabela PERGUNTA.

FK – resposta referente à tabela RESPOSTA.

FK – id\_status\_correcao, referente à tabela STATUS.

## PROJETO DE INTERFACE

### 4.4.1. Interfaces do Tipo Entrada de Dados – Registrar Dados

Não aplicável.

#### 4.4.1.1. Interface do Tipo Manter

##### 4.4.1.1.1. Interface Cadastro Proposta (CadastroProposta).

The screenshot displays the GerPro system interface, titled "Sistema de Geração e Apoio a Correção de PESw" and "CADASTRO DE PROPOSTAS". The interface is divided into a left sidebar menu and a main content area.

**Left Sidebar Menu:**

- Equipe** (expandable):
  - 11 Pesquisar
  - 11 Cadastrar
- Propostas** (expandable):
  - 11 Alocar
  - 11 Cadastrar
- Usuários** (expandable):
  - 11 Alocar
  - 11 Pesquisar
  - 11 Cadastrar
- Relatórios** (expandable)
- Correção (Modo Professor)** (expandable):
  - 11 Lista de Propostas
  - 11 Missão do Produto
  - 11 Lista de Função
  - 11 Justificativa
  - 11 Metodologia
  - 11 Cronograma
  - 11 Avaliação Geral
  - 11 Submeter Correção
- Alterar Senha** (expandable)
- Sair** (expandable)

**Main Content Area: Dados do Cadastro**

Nome\*:

Equipe\*:

Período\*:

**Footer:** GerPro - Sistema de Geração e Apoio a Correção de PESw © Copyright

Figura 34 - Tela de Cadastro de Proposta

## 4.4.1.1.1.1. Relacionamentos com outras interfaces

Tipo de Componente	Nome do Componente	Tipo de Componente	Nome do Componente	Evento	Resposta
Menu	Equipe	Hiperlink	Cadastrar	Clicar	Exibe a tela PesquisarEquipe
Menu	Propostas	Hiperlink	Alocar	Clicar	Exibe a tela AlocarProposta
		Hiperlink	Cadastrar	Clicar	Exibe a tela PesquisarProposta
Menu	Usuários	Hiperlink	Cadastrar	Clicar	Exibe a tela PesquisarUsuario
Hiperlink	Relatórios			Clicar	Exibe a tela Relatório.
Menu	Correção (modo Professor)	Hiperlink	Lista de Propostas	Clicar	Exibe a tela ListarPropostas
		Hiperlink	Missão Produto	Clicar	Exibe a tela CorrigirMissao
		Hiperlink	Lista de Finções	Clicar	Exibe a tela CorrigirListaFuncoes
		Hiperlink	Justificativa	Clicar	Exibe a tela CorrigirJustificativa
		Hiperlink	Metodologia	Clicar	Exibe a tela CorrigirMetodologia
		Hiperlink	Cronograma	Clicar	Exibe a tela CorrigirCronograma
Submenu	Avaliação Geral	Hiperlink	Avaliação geral-1	Clicar	Exibir a tela CorrigirAvalicaoogeral1
		Hiperlink	Avaliação Geral-2	Clicar	Exibe a tela CorrigirAvaliacaoGeral2
		Hiperlink	Avaliação geral-3	Clicar	Exibe a tela CorrigirAvaliacaoGeral3
Hiperlink	Alterar Senha			Clicar	Exibe a tela AlterarSenha
Hiperlink	Sair			Clicar	Exibe a tela Login

## 4.4.1.1.1.2. Controle de Botões – Funcionalidade/Visibilidade

Comando	Componente	Funcionalidade	Visibilidade	Observação
Salvar	Botão	Salva os dados.	Sempre habilitado	Após acionar botão O sistema exibe a mensagem: Parabéns Cadastro Realizado com sucesso. Se botão acionado, mas deixar de preencher algum campo obrigatório, o sistema exibe a mensagem: Campos obrigatório preenchimento.

## 4.4.2. Interface do Tipo Processamento

### 4.4.2.1. Interface Alocar Propostas (AlocarProposta)



**Figura 35 - Tela de Alocar Proposta**

#### 4.4.2.1.1. Relacionamentos com outras interfaces

Tipo de Componente	Nome do Componente	Tipo de Componente	Nome do Componente	Evento	Resposta
Menu	Equipe	Hiperlink	Cadastrar	Clicar	Exibe a tela PesquisarEquipe
Menu	Propostas	Hiperlink	Alocar	Clicar	Exibe a tela AlocarProposta
		Hiperlink	Cadastrar	Clicar	Exibe a tela PesquisarProposta
Menu	Usuários	Hiperlink	Cadastrar	Clicar	Exibe a tela PesquisarUsuario
Hiperlink	Relatórios			Clicar	Exibe a tela Relatório.
Menu	Correção (modo Professor)	Hiperlink	Lista de Propostas	Clicar	Exibe a tela ListarPropostas
		Hiperlink	Missão Produto	Clicar	Exibe a tela CorrigirMissao

		Hiperlink	Lista de Funções	Clicar	Exibe a tela CorrigirListaFuncoes
		Hiperlink	Justificativa	Clicar	Exibe a tela CorrigirJustificativa
		Hiperlink	Metodologia	Clicar	Exibe a tela CorrigirMetodologia
		Hiperlink	Cronograma	Clicar	Exibe a tela CorrigirCronograma
Submenu	Avaliação Geral	Hiperlink	Avaliação geral-1	Clicar	Exiber a tela CorrigirAvalicaogeral1
		Hiperlink	Avaliação Geral-2	Clicar	Exibe a tela CorrigirAvaliacaoGeral2
		Hiperlink	Avaliação geral-3	Clicar	Exibe a tela CorrigirAvaliacaoGeral3
Hiperlink	Alterar Senha			Clicar	Exibe a tela AlterarSenha
Hiperlink	Sair			Clicar	Exibe a tela Login

#### 4.4.2.1.2. Controle de Botões – Funcionalidade / Visibilidade

Comando	Componente	Funcionalidade	Visibilidade	Observação
Grupo	Radio	Define tipo de alocação	Sempre habilitado	Não aplicável.
Individual	Radio	Define tipo de alocação	Sempre habilitado	Não aplicável
Adicionar	Botão	Define Professor que irá participar do processo de correção.	Habilitado após usuário selecionar professor	Não aplicável.
Remover	Botão	Remove apenas o professor da lista de seleção.	Habilitado após usuário selecionar professor	Não aplicável.
Remover Todos	Botão	Remove todos os professores da lista de alocados.	Habilitado após usuário adicionar algum professor para lista de alocados.	Não aplicável.
Alocar	Botão	Aloca as proposta dinamicamente para os professores que estão na lista de alocados.	Desabilitado se não existir propostas para serem alocadas	Se acionado antes de selecionar professor o sistema exibe a mensagem: Selecione o professor.



#### 4.4.2.2. Interface Submeter Correção (SubmeterCorrecao).



**Sistema de Geração e Apoio a Correção de PESw**

**SUBMETER CORREÇÃO**

**Proposta** ⬆

- ⌵ Lista de Propostas
- ⌵ Missão do Produto
- ⌵ Lista de Função
- ⌵ Justificativa
- ⌵ Metodologia
- ⌵ Cronograma
- **Avaliação Geral**
- ⌵ Submeter Correção

**Sair** ▶

Pergunta	Resposta	Status
Missão do produto de acordo com a abrangência do sistema?	Sim	Corrigido
Lista de Funções bem definidas, demonstrando, no mínimo, 7 (sete) requisitos do sistema (entrada, processamento e saída). Sendo pelo menos 3 de processamento?	Não	Corrigido
Justificativa clara, com texto bem organizado e, enfatizando a relevância e a problemática que implica na necessidade do desenvolvimento do Sistema?	Sim	Corrigido
Metodologia coerente com os objetivos definidos e adequada abrangência do tema?	Sim	Corrigido
O cronograma apresenta datas coerentes?	Sim	Corrigido
Apresentação e estilo (bem organizado, claro, correção gramatical e ortográfica), coesão e coerência textual?	Sim	Corrigido
Trabalho de acordo com os padrões de escrita definidos?	Sim	Corrigido
Sistema pertencente ao grupo de trabalhos aceitos?	Sim	Corrigido

GerPro - Sistema de Geração e Apoio a Correção de PESw © Copyright

**Figura 36 - Tela de Submeter Correção**

## 4.4.2.2.1. Relacionamentos com outras interfaces

Tipo de Componente	Nome do Componente	Tipo Componente	Nome do Componente	Evento	Resposta
Menu	Propostas	Hiperlink	Lista de Propostas	Clicar	Exibe a tela ListarPropostas
		Hiperlink	Missão do Produto	Clicar	Exibe a tela CorrigirMissao
		Hiperlink	Lista de Função	Clicar	Exibe a tela CorrigirListaFuncoes
		Hiperlink	Justificativa	Clicar	Exibe a tela CorrigirJustificativa
		Hiperlink	Metodologia	Clicar	Exibe a tela CorrigirMetodologia
		Hiperlink	Cronograma	Clicar	Exibe a tela CorrigirCronograma
Menu	Avaliação Geral	Hiperlink	Avaliação Geral-1	Clicar	Exibe a tela CorrigirAvaliacaoGeralper-gunta1
		Hiperlink	Avaliação Geral-2	Clicar	Exibe a tela CorrigirAvaliacaoGeralper-gunta2
		Hiperlink	Avaliação Geral-3	Clicar	Exibe a tela CorrigirAvaliacaoGeralper-gunta3

## 4.4.2.2.2. Controle de Botões – Funcionalidade / Visibilidade

Comando	Componente	Funcionalidade	Visibilidade	Observação
Submeter	Botão	Executar o calculo do status da proposta(Aprovado, Aprovado com Ressalvas ou Reprovado.	Sempre habilitado	Se houver algum item com status diferente de corrigido, o sistema exibe a mensagem: ##### Se todos os itens estiverem com status igual a corrigido o sistema exibe a mensagem: #####

### 4.4.3. Interface do Tipo Busca Dados.

#### 4.4.3.1. Interface Pesquisar Propostas (Pesquisa).

**GerPro**  
Sistema de Geração e Apoio a Correção de PESw

**PESQUISAR PROPOSTAS**

**Pesquisa**

Equipe

Proposta	Equipe	Status	Opções
p1	m2r	Aprovado com ressalvas	<input type="button" value="Editar"/>
p2	1	Aprovado com ressalvas	<input type="button" value="Editar"/>
p3	2	Reprovado	<input type="button" value="Editar"/>
p4	3	Aprovado	<input type="button" value="Editar"/>
p5	4	Aprovado	<input type="button" value="Editar"/>

1 2

GerPro - Sistema de Geração e Apoio a Correção de PESw © Copyright

**Menu Lateral:**

- Equipe**
  - Pesquisar
  - Cadastrar
- Propostas**
  - Alocar
  - Cadastrar
- Usuários**
  - Alocar
  - Pesquisar
  - Cadastrar
- Relatórios**
- Correção (Modo Professor)**
  - Lista de Propostas
  - Missão do Produto
  - Lista de Função
  - Justificativa
  - Metodologia
  - Cronograma
  - Avaliação Geral
  - Submeter Correção
- Alterar Senha**
- Sair**

Figura 37 - Tela de Pesquisar Proposta

## 4.4.3.1.1. Relacionamentos com outras interfaces

Tipo de Componente	Nome do Componente	Tipo de Componente	Nome do Componente	Evento	Resposta
Menu	Equipe	Hiperlink	Cadastrar	Clicar	Exibe a tela PesquisarEquipe
Menu	Propostas	Hiperlink	Alocar	Clicar	Exibe a tela AlocarProposta
		Hiperlink	Cadastrar	Clicar	Exibe a tela PesquisarProposta
Menu	Usuários	Hiperlink	Cadastrar	Clicar	Exibe a tela PesquisarUsuario
Hiperlink	Relatórios			Clicar	Exibe a tela Relatório.
Menu	Correção (modo Professor)	Hiperlink	Lista de Propostas	Clicar	Exibe a tela ListarPropostas
		Hiperlink	Missão Produto	Clicar	Exibe a tela CorrigirMissao
		Hiperlink	Lista de Funções	Clicar	Exibe a tela CorrigirListaFuncoes
		Hiperlink	Justificativa	Clicar	Exibe a tela CorrigirJustificativa
		Hiperlink	Metodologia	Clicar	Exibe a tela CorrigirMetodologia
		Hiperlink	Cronograma	Clicar	Exibe a tela CorrigirCronograma
Submenu	Avaliação Geral	Hiperlink	Avaliação geral-1	Clicar	Exibe a tela CorrigirAvaliacaogeral1
		Hiperlink	Avaliação Geral-2	Clicar	Exibe a tela CorrigirAvaliacaoGera2
		Hiperlink	Avaliação geral-3	Clicar	Exibe a tela CorrigirAvaliacaoGera3
Hiperlink	Alterar Senha			Clicar	Exibe a tela AlterarSenha
Hiperlink	Sair			Clicar	Exibe a tela login
Botão	Nova Proposta			Clicar	Exibe a tela CadastroPropsta
Botão	Editar			Clicar	Exibe a tela AlterarProposta

## 4.4.3.1.2. Controle de Botões – Funcionalidade / Visibilidade.

Comando	Componente	Funcionalidade	Visibilidade	Observação
Pesquisar	Botão	Executa uma pesquisa de acordo com a entrada de dados feita pelo usuário.	Sempre habilitado.	Não aplicável.
Nova Proposta	Botão	Exibe a tela CadastrarProposta.	Sempre habilitado.	Não aplicável
Editar	Botão	Exibe a tela de AlterarProposta.	Sempre habilitado, mas visível somente se existir proposta cadastrada.	Não aplicável.

#### 4.4.4. Interfaces do Tipo Saída de Dados – Relatórios.

##### 4.4.4.1. Interface Relatórios (Relatório)

The screenshot displays the GerPro system interface. At the top, a blue banner features the 'GerPro' logo and the text 'Sistema de Geração e Apoio a Correção de PESw'. Below this, the title 'RELATÓRIOS' is centered. On the left, a vertical menu lists options: 'Equipe', 'Propostas', 'Usuários', 'Relatórios', 'Correção (Modo Professor)', 'Alterar Senha', and 'Sair'. The 'Relatórios' option is selected. The main content area contains three radio buttons for report types: 'Relatório Equipe X Proposta' (selected), 'Relatório Professor X Proposta', and 'Relatório Resultados Proposta'. Below these is a 'Período:' label followed by a dropdown menu showing '2009/01'. A 'Visualizar' button is positioned at the bottom left of the report selection area. The footer of the interface reads 'GerPro - Sistema de Geração e Apoio a Correção de PESw © Copyright'.

**Figura 38 - Tela para Chamada de Relatórios**

## 4.4.4.1.1. Relacionamentos com outras interfaces

Tipo de Componente	Nome do Componente	Tipo de Componente	Nome do Componente	Evento	Resposta
Menu	Equipe	Hiperlink	Cadastrar	Clicar	Exibe a tela PesquisarEquipe
Menu	Propostas	Hiperlink	Alocar	Clicar	Exibe a tela AlocarProposta
		Hiperlink	Cadastrar	Clicar	Exibe a tela PesquisarProposta
Menu	Usuários	Hiperlink	Cadastrar	Clicar	Exibe a tela PesquisarUsuario
Hiperlink	Relatórios			Clicar	Exibe a tela Relatório.
Menu	Correção (modo Professor)	Hiperlink	Lista de Propostas	Clicar	Exibe a tela ListarPropostas
		Hiperlink	Missão Produto	Clicar	Exibe a tela CorrigirMissao
		Hiperlink	Lista de Funções	Clicar	Exibe a tela CorrigirListaFuncoes
		Hiperlink	Justificativa	Clicar	Exibe a tela CorrigirJustificativa
		Hiperlink	Metodologia	Clicar	Exibe a tela CorrigirMetodologia
		Hiperlink	Cronograma	Clicar	Exibe a tela CorrigirCronograma
Submenu	Avaliação Geral	Hiperlink	Avaliação geral-1	Clicar	Exibir a tela CorrigirAvaliacaogeral1
		Hiperlink	Avaliação Geral-2	Clicar	Exibe a tela CorrigirAvaliacaoGera2
		Hiperlink	Avaliação geral-3	Clicar	Exibe a tela CorrigirAvaliacaoGera3
Hiperlink	Alterar Senha			Clicar	Exibe a tela AlterarSenha
Hiperlink	Sair			Clicar	Exibe a tela Login

## 4.4.4.1.2. Controle de Botões – Funcionalidade / Visibilidade

Comando	Componente	Funcionalidade	Visibilidade	Observação
Relatório Equipe x Proposta	Radio	Define o tipo de relatório	Sempre habilitado.	Não aplicável
Relatório Professor x Proposta	Radio	Seleciona	Sempre habilitado	Não aplicável
Relatório Resultados Proposta	Radio	Seleciona	Sempre habilitado	Não aplicável
Visualizar	Botão	Exibe a tela de relatório de acordo com o tipo de relatório escolhido.	Sempre habilitado	Não aplicável

#### 4.4.5. Interfaces do Tipo Visualização de Relatórios

##### 4.4.5.1. Interface Relatório : RelatorioEquipeProposta

GerPro		Relatório Equipe x Proposta	
Sistema de Geração e Apoio a Correção de PESw			
Equipe		1	
Proposta	p2		
Data Criação	09/06/2009	Data Submissão	
Período	2009/01	Status	Aprovado com ressalvas
Equipe		2	
Proposta	p3		
Data Criação	09/06/2009	Data Submissão	
Período	2009/01	Status	Reprovado
Equipe		3	
Proposta	p4		
Data Criação	09/06/2009	Data Submissão	
Período	2009/01	Status	Aprovado
Equipe		4	
Proposta	p5		
Data Criação	09/06/2009	Data Submissão	
Período	2009/01	Status	Aprovado
Equipe		5	
Proposta	SisPonto		
Data Criação	27/06/2009	Data Submissão	
Período	2009/01	Status	Em construção
Equipe		m2r	
Proposta	p1		
Data Criação	09/06/2009	Data Submissão	
Período	2009/01	Status	Aprovado com ressalvas
Equipe		M3R	

27/06/09 11:45

Pág 1 de 2

Figura 39 - Tela de Relatório

#### 4.4.5.1.1. Relacionamentos com outras interfaces

Não Aplicável.

#### 4.4.5.1.2. Controle de Botões – Funcionalidade / Visibilidade

Comando	Componente	Funcionalidade	Visibilidade	Observação
Imprimir	Ícone	Enviar arquivo para dispositivo de impressão.	Sempre Habilitado	Não aplicável
Salvar	Ícone	Salva dados em local desejado	Sempre Habilitado	Não aplicável

#### 4.4.5.1.3. Interfaces de Entrada que seguirão o mesmo padrão

	Tela	Características Particulares
1	Relatório Professor x Equipe	Não Aplicável
2	Relatório Resultados Proposta	Não Aplicável

### 4.5. ASPECTOS ERGONÔMICOS

Os botões de comando serão localizados na parte inferior da tela, abaixo do formulário que será preenchido e, em algumas telas, estão disponíveis acima do grid utilizado para pesquisa.

Todas a telas terão um banner com nome GerPro de cor azul escuro, o fundo das telas será de cor azul claro, botões em tom azul claro, cor da fonte preta. No lado esquerdo das telas será localizado um menu com um conjunto de hiperlinks estarão disponíveis de acordo com o nível de acesso do usuário, que servirá para a navegabilidade das telas.





As páginas serão identificadas pelos títulos compostos por uma caixa de texto com nome da página em questão. A cor de fundo desta caixa será diferente da cor de fundo do restante da página, porém, neste projeto não há diferenças devido às restrições no software de prototipagem das telas.

Na visualização de relatórios, cada item que será mostrado terá uma barra de títulos de cor azul e o logo abaixo o conteúdo referente ao título que serão separados por barra de título com seu referido nome de acordo com o tipo de relatório escolhido.

### 4.5.1. Comunicabilidade

#### 4.5.1.1. Convenções utilizadas

Símbolo	Objetivo	Funcionalidade	Aspectos Ergonômicos	Exemplo
*	Indicar os campos obrigatórios	Este símbolo tem característica meramente de sinalização.	Localizado no final do nome do campo	Nome*
	Indicar os hiperlink não expandidos no momento	Este símbolo tem característica meramente de sinalização.	Localizado no início dos hiperlinks do menu de navegação	+ Cadastros
	Indicar os hiperlink atualmente expandidos	Este símbolo tem característica meramente de sinalização.	Localizado no início dos hiperlinks do menu de navegação	- Cadastros

#### 4.5.1.2. Mensagens

Tipo	Objetivo	Funcionalidade	Aspectos Ergonômicos	Exemplo
Erro	Comunicar falha na comunicação entre usuário e o sistema.	Tornar a interação entre o usuário e sistema clara e amigável.	A mensagem será exibida na própria tela que gerou o erro. A cor da fonte será vermelho e será exibida logo abaixo do título e acima do corpo da página.	Ver a tela login.
Alerta	Comunicar falha na comunicação entre usuário e o sistema.	Deve ser exibido após usuário acionar o comando salvar.	A Mensagem será exibida na própria tela que gerou o erro. A cor da fonte será vermelho e será exibida logo abaixo do	Ver tela AlocarPropostas

			título e acima do corpo da pagina.	
Confirmação	Comunicar que a operação foi realizada.	Deve ser exibido após o usuário acionar um comando que realize uma operação.	A Mensagem será exibida na própria tela que gerou o erro. A cor da fonte será vermelho e será exibida logo abaixo do título e acima do corpo da pagina.	Ver tela CadastroProposta

#### 4.5.1.3. Tela login: Mensagem de erro.



The screenshot shows the GerPro login interface. At the top, the logo 'GerPro' is displayed in white on a dark blue background, followed by the text 'Sistema de Geração e Apoio a Correção de PESw'. Below this, a prompt asks the user to enter their login and password. A red error message, '• Usuário ou senha inválidos', is displayed above the login fields. The fields are labeled 'Matricula\*' and 'Senha\*', with the first field containing the number '1'. An 'Entrar' button is located below the fields. At the bottom of the page, a footer contains the text 'GerPro - Sistema de Geração e Apoio a Correção de PESw © Copyright'.

**Figura 40 - Tela de Login Mensagem de Erro**

## 4.5.1.4. Tela AlocarProposta: Mensagem de alerta



Figura 41 - Tela de Alocar Proposta

## 4.5.1.5. Tela CadastroProposta: Mensagem de confirmação

The screenshot displays the GerPro system interface. At the top, the logo 'GerPro' is shown next to the text 'Sistema de Geração e Apoio a Correção de PESw'. Below this, the title 'CADASTRO DE PROPOSTAS' is centered. A success message '• Cadastro salvo com sucesso' is displayed in blue. On the left, a vertical menu contains options: 'Equipe', 'Propostas', 'Usuários', 'Relatórios', 'Correção (Modo Professor)', 'Alterar Senha', and 'Sair'. The main area is titled 'Dados do Cadastro' and contains three input fields: 'Nome\*', 'Equipe\*' (with a dropdown menu showing 'Windows'), and 'Período\*'. A 'Salvar' button is located below these fields. The footer of the page reads 'GerPro - Sistema de Geração e Apoio a Correção de PESw © Copyright'.

GerPro

Sistema de Geração e Apoio a Correção de PESw

**CADASTRO DE PROPOSTAS**

• Cadastro salvo com sucesso

**Dados do Cadastro**

Nome\*:

Equipe\*:

Período\*:

Salvar

GerPro - Sistema de Geração e Apoio a Correção de PESw © Copyright

Figura 42 - Tela de Cadastro de Propostas

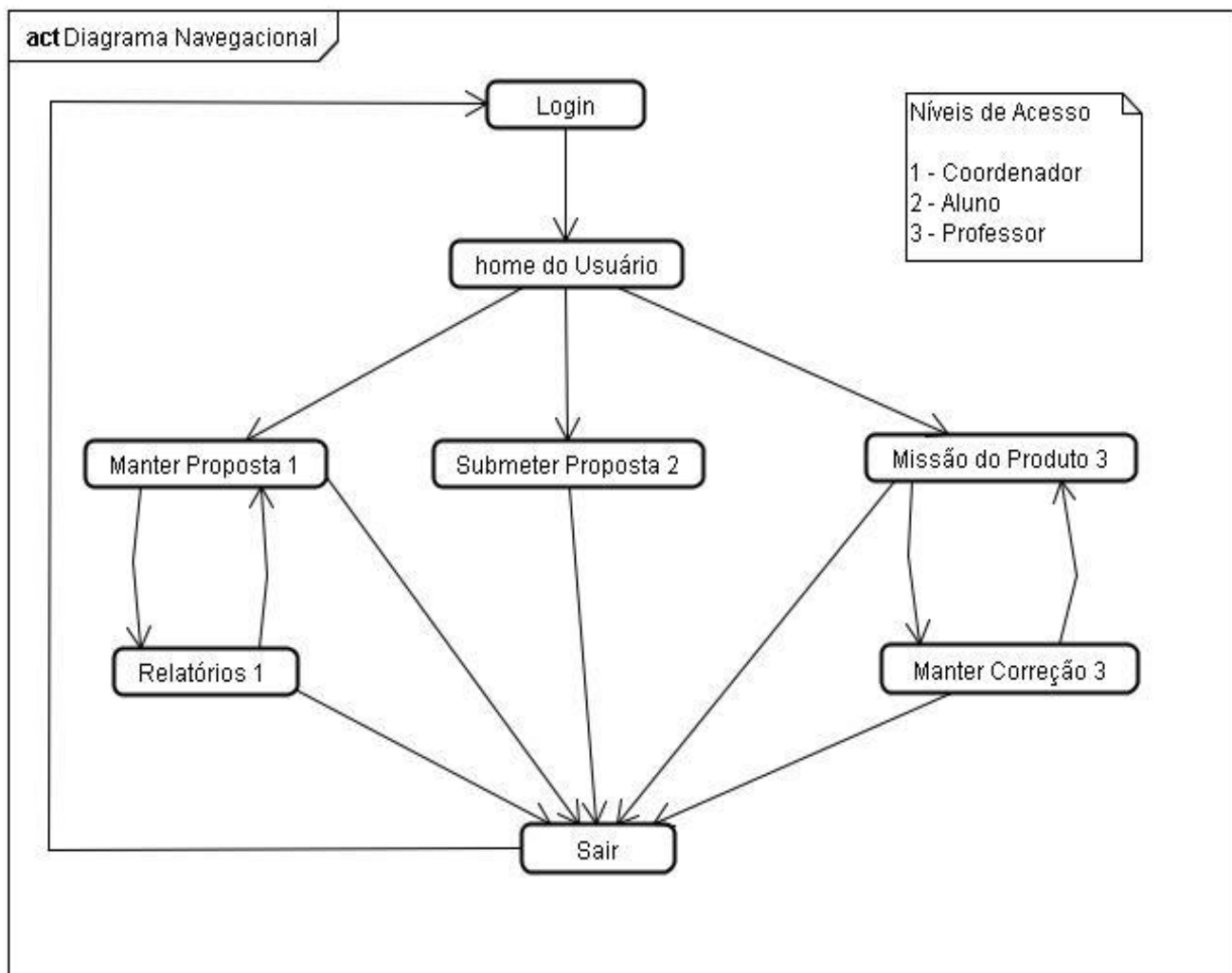
#### 4.6. PROJETO NAVEGACIONAL

Níveis de acesso:

Nível 1: Coordenador

Nível 2: Professor

Nível 3: Aluno



**Figura 43 - Projeto Navegacional**

## 5. CONSIDERAÇÕES FINAIS

O Sistema GerPro tem como objetivo agilizar o processo de geração das propostas de TCC desenvolvidas pelos alunos, bem como auxiliar os professores nas correções das mesmas.

Neste projeto, pretendeu-se aprimorar os conhecimentos adquiridos em sala de aula, pondo em práticas técnicas para levantamento de requisitos, análise e a documentação dos mesmos, tentando-se assim absorver o máximo de conhecimento do sistema e do processo de desenvolvimento de software em seu início.

O processo de desenvolvimento serviu como orientação para que a produção dos documentos fosse realizada de forma concisa e sem maiores custos de tempo e esforço. Dessa forma, a documentação do sistema GerPro, mesmo passando por várias modificações ao longo do processo, foi realizada a contento. Também contribuíram para este objetivo, o trabalho em equipe e a escolha das ferramentas e frameworks mais apropriados para a modelagem.

O GerPro apresentou-se satisfatório na execução de suas funções. O processo de testes unitários nos ajudou a manter o foco do que era realmente necessário produzir. A utilização de *design patterns* foi de extrema importância para que problemas de modularização, alto-acoplamento, persistência de dados, entre outros, fossem solucionados de forma rápida e clara.

Assim, conseguimos alcançar de modo satisfatório nossas metas. A produção deste aplicativo impôs que determinação, esforço e trabalho em equipe fossem indispensáveis em cada momento deste período acadêmico.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

[BEZ07A] – BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. Padrão Facade. Elsevier Editora Ltda. Rio de Janeiro - RJ. Pag. 278;

[BEZ07B] – BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. Padrão DAO. Elsevier Editora Ltda. – Rio de Janeiro - RJ. pag. 353;

[CAR09] – CARNEIRO, Davi Luan <http://www.javafree.org/artigo/871452/Introducao-ao-pattern-DAO.html> / Padrão Dao - acessado dia 16/06/09 – 11h43min;

[COU02] – COUNTINHO, Flávio, <http://www.guj.com.br/article.show.logic?id=> / Apache Tomcat. acessado dia 15/06/2009 às 11h25min;

[DEI05A] – DEITEL, H.M. Java : como programar / H. M. Deitel, P. J.Deitel. UML tradução Edson Furmankiewicz. revisão técnica Fábio Luchini. – 6. Ed. – São Paulo. pag. 17;

[DEI05B] – DEITEL, H.M. Java : como programar. H. M. Deitel, P. J.Deitel. Internet e World Wide Web. tradução Edson Furmankiewicz. revisão técnica Fábio Luchini. – 6. Ed. – São Paulo. pag. 5;



[DEI05C] – DEITEL, H.M. Java : como programar. H. M. Deitel, P. J. Deitel. Internet e World Wide Web. tradução Edson Furmankiewicz. revisão técnica Fábio Luchini. – 6. Ed. – São Paulo. pag. 5;

[DEI05D] – DEITEL, H.M. Java : como programar / H. M. Deitel, P. J. Deitel. JAVA tradução Edson Furmankiewicz; revisão técnica Fábio Luchini. – 6. Ed. – São Paulo. Pág. 7;

[EN05A] – ELMASRI, Ramez e Shamkant B. Navathe; Sistemas de banco de dados / Banco de Dados - revisor técnico Luis Ricardo de Figueiredo. – São Paulo : Pearson Addison / Wesley, 2005. Pág. 5;

[EN05B] – ELMASRI, Ramez e Shamkant B. Navathe; Sistemas de banco de dados / O que é Banco de Dados - revisor técnico Luis Ricardo de Figueiredo. – São Paulo : Pearson Addison / Wesley, 2005. Pág. 5;

[FIL01] - FILHO, Wilson de Pádua Paula. Engenharia de Software – Fundamentos, Métodos e Padrões. Praxis. LTC - 2001. Rio de Janeiro - RJ. Pag. 278;

[GEA02] GEARY, David; [http://translate.google.com.br/translate?hl=pt-BR&langpair=en|pt&u=http://www.javaworld.com/javaworld/jw-11-2002/jw-1129-jsf.html&prev=/translate\\_s%3Fhl%3Dpt-BR%26q%3Dartigos%2Bsobre%2Bjsf%26sl%3Dpt%26tl%3Den](http://translate.google.com.br/translate?hl=pt-BR&langpair=en|pt&u=http://www.javaworld.com/javaworld/jw-11-2002/jw-1129-jsf.html&prev=/translate_s%3Fhl%3Dpt-BR%26q%3Dartigos%2Bsobre%2Bjsf%26sl%3Dpt%26tl%3Den) / Artigos sobre JSF – acessado dia 17/06/2009 às 13h22min;

[GON07] - GONÇALVES, Edson - *Desenvolvendo Aplicações Web com JSP Servlet, JavaServer Faces, Hibernate, EJB 3 Persistence e Ajax /MVC* - Rio de Janeiro - 2007: Editora Ciência Moderna Ltda;

[GUE08] – GUEDES, Gilleanes T. A. *UML Uma Abordagem Prática*. UML .Editora Novatec Editora Ltda. 3ª. Edição. São Paulo - SP - 2008. Pag. 18 e 27;

[HEM07A] – HERMRAJANI, Anil - *Desenvolvimento Ágil em Java com Spring, Hibernate e Eclipse*; Hibernate, Apache Tomcat, Eclipse. São Paulo Pearson Prentice Hall, 2007;

[HEM07B] – HERMRAJANI, Anil - *Desenvolvimento Ágil em Java com Spring, Hibernate e Eclipse*; Hibernate, Apache Tomcat, Ferramenta Eclipse. São Paulo Pearson Prentice Hall, 2007;

[INF08] - <http://www.infoblogs.com.br/view.action?contentId=34104&JUDE-System-Design-Tool-Modelagem-UML-MER-e-outros.html> / O que é Jude Design - acessado dia 17/06/2009 às 16h12min;

[INF09] - <http://www.infowester.com/versoesjava.php> / Artigos JEE - acessado dia 17/06/2009 às 12h:04min;

[JAV09A] - <http://www.javafree.org/artigo/871498/Tutorial-Java-O-que-e-Java.html> - Artigos sobre JAVA - acessado dia 16/06/2009 às 12h22min;

[JAV09B] - <http://www.javafree.org/artigo/871497/Tutorial-Java-3-Orientacao-a-Objetos.html> Orientação À Objetos - acessado dia 16/06/2009 às 14h20min;

[JAV09C] - <http://www.javafree.org/artigo/3154/Tutorial-de-IREPORT.html> IREPORT - acessado dia 16/06/2009 às 21h19min;

[LOC09] - <http://tecblog.locaweb.com.br/2008/10/27/evolucao-de-versionamento-de-codigo-git/> Versionamento – acessado dia 14/06/2009 às 15h33min;

[MAR09A] - [http://www.macoratti.net/vbn\\_mvc.htm](http://www.macoratti.net/vbn_mvc.htm) / PADRÕES DE PROJETO - acessado dia 15/06/2009 16h37min;

[MAR09B] - [http://www.macoratti.net/vbn\\_mvc.htm](http://www.macoratti.net/vbn_mvc.htm) / MVC - acessado dia 11/02/2009 11h37min;

[MAR09C] - [http://www.macoratti.net/vbn\\_mvc.htm](http://www.macoratti.net/vbn_mvc.htm) / FACADE - acessado dia 15/06/2009 16h46min;

[MYS09A] - <http://dev.mysql.com/doc/refman/4.1/pt/documenters-translators.html> MySQL. acessado dia 16/06/2009 10h:08min;

[MYS09B] - <http://dev.mysql.com/doc/refman/4.1/pt/documenters-translators.html> MySQL. acessado dia 16/06/2009 10h:18min;

[OLI02] – OLIVEIRA, Celso Henrique Poderoso de. SQL .Editora Novatec Editora Ltda. São Paulo - SP - 2002.Pag. 18;

[PER08] – PERREIRA, Nice Brito Machado

[http://anacarol.blog.br/old/aulas/artigos\\_uteis/modelo\\_visualizacao\\_controle.pdf](http://anacarol.blog.br/old/aulas/artigos_uteis/modelo_visualizacao_controle.pdf) MVC e

Smalltalk - acessado dia 12/02/2008 às 12h37min;

# ANEXOS

