



**FUNDAÇÃO FACULDADES LUIZ MENEGHEL**  
**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**Maria Virgínia Andreoti Mendes Bordin**

**ANÁLISE E DESENVOLVIMENTO DE UM PROJETO  
PILOTO ACADÊMICO UTILIZANDO UML E BUP**

# **FUNDAÇÃO FACULDADES LUIZ MENEGHEL**

Instituição de ensino superior vinculada à Universidade Estadual do Norte do Paraná, em acordo com a Lei Estadual nº 15.300, de 28 de setembro de 2006.

**Maria Virgínia Andreoti Mendes Bordin**

## **ANÁLISE E DESENVOLVIMENTO DE UM PROJETO PILOTO ACADÊMICO UTILIZANDO UML E BUP**

Trabalho de conclusão de curso submetido à Fundação Faculdades Luiz Meneghel como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. M.s Glauco Carlos Silva

**Maria Virgínia Andreoti Mendes Bordin**

**ANÁLISE E DESENVOLVIMENTO DE UM PROJETO  
PILOTO ACADÊMICO UTILIZANDO UML E BUP**

**COMISSÃO EXAMINADORA**

---

Prof.(a) M.s Glauco Carlos Silva

---

Prof.(a) M.s André Luiz Menoli

---

Prof.(a) Roberto Vedoato

Bandeirantes, 24 de novembro de 2006.

Aos meus pais, à minha família, aos meus amigos e a duas pessoas que foram muito importantes em minha vida e que já não se encontram mais presentes aqui comigo ao meu tio Rafael Andreoti Mendes e ao meu avô Wilson Ferreira Martins, mas tenho certeza que onde quer que eles estejam, estarão torcendo por mim neste momento e por toda minha vida. E a todas as pessoas que admiro muito e por ter sido... Companheiros de todas as horas.....

## AGRADECIMENTOS

Ao Professor Orientador, que me auxiliou em todas as etapas deste trabalho.

A minha família, pela confiança e motivação.

Aos meus amigos e colegas, pela força e pela vibração em relação à esta jornada.

Aos professores do curso, pois juntos trilhamos uma etapa importante de nossas vidas.

Ao meu primo Marcos Andreoti Mendes que me auxiliou nesta etapa.

A todos os Professores que, com boa intenção, colaboraram para a realização e finalização deste trabalho.

Dedico também este trabalho a Diretora da Escola Algodão Doce que me proporcionou todos os dados necessários para a realização desta pesquisa.

Aos amigos virtuais, que me ajudaram com dicas cruciais sobre *Delphi* e o *Ibexpert*.

***O Meu Muito Obrigada!!!***

**“Um bom software é capaz de satisfazer  
às necessidades de seus usuários  
e respectivos negócios”**

Grady Booch, James Rumbaugh e Ivar Jacobson.

BORDIN, Maria Virgínia Andreoti Mendes. **Análise e Desenvolvimento de um Projeto Piloto Acadêmico Utilizando UML e BUP**. 2006. 105 páginas. Dissertação de Graduação em Sistema de Informação pela UENP - Fundação Faculdades Luiz Meneghel – Bandeirantes – PR

## RESUMO

Este trabalho teve como objetivo, desenvolver um projeto piloto de um *software* acadêmico para a utilização da Escola Algodão Doce de Abatiá – PR. A proposta apresentada neste trabalho foi o desenvolvimento de uma aplicação destinada as funcionalidades básicas na qual é possível o controle de cadastros de alunos, notas, frequência, disciplina, turma, matrícula. Esta aplicação foi projetada utilizando-se a linguagem de modelagem UML (*Unified Modeling Language*) aplicáveis ao projeto, proporcionando um estudo sobre a técnica de documentação de *software* BUP (*Basic Unified Process*). Foi realizado o levantamento de requisitos junto com a diretora da instituição, em que foram identificados os dados necessários para a elaboração do sistema. Para a construção do projeto foi utilizada a ferramenta de desenvolvimento *Delphi*.

**Palavras Chaves:** Modelagem, Documentação e Desenvolvimento.

BORDIN, Maria Virgínia Andreoti Mendes. **Analysis and Development of an Project Aviator Academic Using UML and BUP**. 2006. 105 Pages. Graduate dissertation in System of Information for the UENP – Fundação Faculdades Luiz Meneghel - Pioneers - PR.

### **ABSTRACT**

The present work had as a goal to develop a pilot project of academic software that will be used by the school “Algodão Doce” from Abatiá – PR. The proposal presented by this work was a development of an application that was intended for the basic working whicls is possible the control of the register of students, grades, frequency, subject and matriculation. This application was planned using the UML (Unified Modeling Language) techniques applied to the project, which offered a study about the documentation technique from the software BUP (Basic Unified Process). It was carried out the list of requirement with the help of the institution’s principal, when were identified the mainly data wich were necessary to develop the system. To project this work it was used the development tool named Delphi.

**Key Words:** Modeling, Documentation and Development.



## LISTA DE ILUSTRAÇÕES

<b>Figura 1</b> - Estrutura das Fases.....	27
<b>Figura 2</b> - Definição da Arquitetura do <i>Software</i> .....	37
<b>Figura 3</b> - Definição de Desenvolvedor de <i>Software</i> .....	37
<b>Figura 4</b> - Definição de Plano de Gerente.....	37
<b>Figura 5</b> - Definição de Testador de <i>Software</i> .....	38
<b>Figura 6</b> – Definição de Papéis.....	38
<b>Figura 7</b> - Tela de Entrada do Sistema.....	71
<b>Figura 8</b> - Tela Inicial do Sistema Acadêmico.....	72
<b>Figura 9</b> - Tela Cadastro de Usuário.....	74
<b>Figura 10</b> - Interface Cadastro de Aluno.....	75
<b>Figura 11</b> - Interface Cadastro de Frequência.....	76

## LISTA DE TABELAS

<b>Tabela 1</b> - Levantamento dos Problemas e Impactos Ocorridos na Escola Algodão Doce.....	46
<b>Tabela 2</b> - Posição do Produto.....	47
<b>Tabela 3</b> - Representa a Descrição dos Envolvidos e Usuários.....	48
<b>Tabela 4</b> - Descrição das Necessidades e Características.....	50
<b>Tabela 5</b> - Representação de Gerente de Projeto.....	52
<b>Tabela 6</b> - Papéis e Responsabilidades dos Responsáveis pelo Projeto.....	53
<b>Tabela 7</b> - Estimativa do Projeto.....	54
<b>Tabela 8</b> - Lista de Riscos do Projeto.....	55
<b>Tabela 9</b> - Referente ao Plano de Iteração.....	69
<b>Tabela 10</b> - Nomeação das Tarefas e Nomeados.....	69

## LISTA DE QUADROS

<b>Quadro 1</b> – Blocos.....	39
<b>Quadro 2</b> – Exemplo de Entrega de Processo.....	39

## LISTA DE ABREVIATURAS E SIGLAS

BUP	<i>Basic Unified Process</i>
E.I.E.F	Ensino Fundamental e Ensino Infantil
NRE	Núcleo Regional de Ensino
RUP	<i>Rational Unified Process</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	<i>Structured Query Language</i>
UML	<i>Unified Modeling Language</i>
MER	Modelo de Entidade Relacionamento

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>17</b>
<b>2 PROBLEMATIZAÇÃO.....</b>	<b>19</b>
<b>3 JUSTIFICATIVA.....</b>	<b>20</b>
<b>4 OBJETIVOS.....</b>	<b>21</b>
4.1 Objetivo Geral.....	21
4.2 Objetivos Específicos.....	21
<b>5 MATERIAL E MÉTODOS.....</b>	<b>22</b>
5.1 Material.....	22
5.2 Metodologia.....	22
<b>6 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>24</b>
6.1 APRESENTAÇÃO DO PROJETO PILOTO.....	24
6.1.1 Características das Modelagens.....	24
6.2 CICLO DE VIDA DO <i>SOFTWARE</i> .....	25
6.2.1 Composição da UML.....	27
6.3 FASES DO DESENVOLVIMENTO DE UM SISTEMA.....	29
6.3.1 Análise de Requisitos.....	29

	13
6.3.2 Análise.....	30
6.3.3 <i>Design</i> (Projeto).....	30
6.3.4 Programação.....	31
6.3.5 Testes.....	31
6.4 <i>RATIONAL UNIFIED PROCESS</i> (RUP).....	32
6.5 <i>BASIC UNIFIED PROCESS</i> (BUP).....	34
6.5.1 Definição do BUP.....	34
6.5.1.1 Disciplinas.....	34
6.5.1.2 Papéis.....	35
6.5.1.3 Tarefas.....	36
6.5.1.4 Artefatos.....	36
6.5.1.5 Processo.....	38
6.6 PLANO DE VISÃO.....	40
6.6.1 Introdução.....	40
6.6.1.1 Definição de Plano de Visão.....	40
6.7 PLANO DE PROJETO.....	42
6.8 DESCRIÇÃO DA ADMINISTRAÇÃO DE REQUISITOS.....	42
6.9 DESCRIÇÃO DO PLANO DE ITERAÇÃO.....	43
<b>7 DESENVOLVIMENTO DO TRABALHO.....</b>	<b>44</b>
7.1 PLANO DE VISÃO.....	44
7.1.1 Introdução.....	44
7.1.2 Posicionamento.....	44

	14
7.1.2.1 Descrição do Problema.....	46
7.1.2.2 Definição de Posição do Produto.....	47
7.1.3 Descrições dos Envolvidos e dos Usuários.....	47
7.1.3.1 Resumo dos <i>Stakeholders</i> .....	48
7.1.4 Visão Geral do Produto.....	48
7.1.4.1 Perspectiva do Produto.....	48
7.1.4.2 Suposições e Dependências.....	49
7.1.4.3 Necessidades e Características.....	50
7.1.4.4 Alternativas e Competição.....	51
7.1.5 Requerimentos do Produto.....	51
7.2 PLANO DE PROJETO.....	52
7.2.1 Introdução.....	52
7.2.2 Organização do Projeto.....	52
7.2.2.1 Estrutura Organizacional.....	52
7.2.2.2 Interfaces Externas.....	53
7.2.2.3 Papéis e Responsabilidades.....	53
7.2.3 Programa de Projeto.....	54
7.2.4 Orçamento.....	55

	15
7.2.5 Lista de Risco.....	55
7.3 ADMINISTRAÇÃO DE REQUISITOS.....	56
7.3.1 Caso de Uso: Mantém Cadastro da Instituição.....	56
7.3.2 Caso de Uso: Mantém Cadastro de Funcionários.....	57
7.3.3 Caso de Uso: Mantém Cadastro de Professor.....	58
7.3.4 Caso de Uso: Mantém Usuário do Sistema.....	59
7.3.5 Caso de Uso: Mantém Cadastro do Aluno.....	60
7.3.6 Caso de Uso: Mantém Cadastro de Disciplina.....	61
7.3.7 Caso de Uso: Mantém Cadastro de Frequência.....	62
7.3.8 Caso de Uso: Mantém Cadastro de Nota.....	63
7.3.9 Caso de Uso: Mantém Cadastro de Turma.....	64
7.3.10 Caso de Uso: Realiza Matrícula.....	65
7.3.11 Caso de Uso: Mantém Cadastro de Avaliação.....	66
7.3.12 Caso de Uso: Relatório Parcial.....	67
7.3.13 Caso de Uso: Relatório Final.....	68
7.4 PLANO DE ITERAÇÃO.....	69
7.4.1 Introdução.....	69
7.4.2 Objetivo.....	69
7.5 IMPLEMENTAÇÃO DO SISTEMA ACADÊMICO.....	70
7.5.1 Apresentação do Protótipo Desenvolvido.....	70



7.5.1.1 Entrando no Sistema.....	73
<b>8 CONSIDERAÇÕES FINAIS.....</b>	<b>77</b>
<b>REFERÊNCIAS.....</b>	<b>78</b>
<b>BIBLIOGRAFIA.....</b>	<b>79</b>
<b>APÊNDICES.....</b>	<b>80</b>
<b>APÊNDICE A – Modelo Entidade Relacionamento.....</b>	<b>80</b>
<b>APÊNDICE B – Diagrama de Caso de Uso.....</b>	<b>81</b>
<b>APÊNDICE C - Diagrama de Classe.....</b>	<b>82</b>
<b>APÊNDICE D - Diagrama de Seqüência.....</b>	<b>83</b>
<b>APÊNDICE E - Diagrama de Estado.....</b>	<b>87</b>
<b>APÊNDICE F – <i>Scripts</i> Banco de Dados.....</b>	<b>88</b>
<b>ANEXOS.....</b>	<b>95</b>
<b>ANEXO A – Plano de Visão.....</b>	<b>95</b>
<b>ANEXO B – Plano de Iteração.....</b>	<b>99</b>
<b>ANEXO C – Plano de Projeto.....</b>	<b>100</b>
<b>ANEXO D – Descrição dos Casos de Uso.....</b>	<b>103</b>

## 1 INTRODUÇÃO

A maior preocupação dos países com a economia política da informação, abrange tanto os países evoluídos quanto aqueles que ainda estão em desenvolvimento (Cruz[ca.2004]). Estando o nosso país em desenvolvimento no campo da Informática já pode ser considerado desenvolvido. Embora esses avanços em algumas áreas ainda caminham lentamente. Nas escolas mesmo as particulares ainda se encontram carentes de um sistema informatizado. A idéia deste projeto surgiu do levantamento de questões de gerenciamento acadêmico de uma escola.

A montagem do projeto piloto tem como necessidade básica compreender as metas propostas pelo cliente para que este sistema possa ser aprimorado posteriormente, para tanto foram utilizados os conceitos do *Basic Unified Process* (BUP) que utiliza como notação a *Unified Modeling Language* (UML).

A UML é uma linguagem gráfica de desenvolvimento de *software* orientado a objeto que visa a representação da arquitetura de um sistema. Por conseqüência, este projeto piloto pretende comprovar a eficácia é o uso desta linguagem, com base no desenvolvimento do Sistema Acadêmico.

O processo do BUP têm o intuito de descrever as etapas do desenvolvimento do *software* de maneira concisa em que o desenvolvedor possa demonstrar as idéias propostas pelo cliente evidenciando, algumas etapas que estão propostas dentro dos conceitos existentes no *Rational Unified Process* (RUP), pois esta notação é uma versão mais resumida do RUP utilizada somente em projetos pequenos que constituem de 3 a 6 participantes.

O RUP é um processo de engenharia de *software*. Ele oferece uma abordagem baseada em disciplinas para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento. Sua meta é garantir a produção de *software* de alta qualidade que atenda às necessidades dos usuários dentro de um cronograma e de um orçamento previsível.

Para a elaboração deste projeto foi necessária uma coleta de informações por meio de entrevistas com a diretora da Instituição, visando a determinação das dificuldades da escola. As informações obtidas foram organizadas e especificadas através do processo BUP que engloba o Plano de Projeto, Plano de Visão, Administração de Requisitos e o Plano de Iteração para que o dimensionamento das informações dos dados levantados pudessem ser aplicados neste processo. Cada etapa do processo mencionado anteriormente será demonstrado no decorrer deste trabalho.

## 2 PROBLEMATIZAÇÃO

Segundo Cruz, [ca.2004] para estimular o desenvolvimento da sociedade, a informação tornou-se um fator muito importante, porém, há o fator político em que os países se classificam de acordo com a sua disposição e quantidade de informação que tem em seu poder.

Pode-se perceber que com o processo dos avanços tecnológicos, a busca por informação tornou-se alvo comum em toda a sociedade. Essas necessidades não atingem somente grandes corporações, mas também médias e pequenas empresas. Como é o caso da empresa em questão nesta proposta de projeto, a Escola Algodão Doce – Ensino Infantil e Ensino Fundamental (E.I.E.F) onde pôde-se perceber que há uma imensa defasagem no que se diz respeito à tecnologia. O intuito deste trabalho é propor um sistema que melhore o controle dos dados dos alunos na instituição regularizando os cadastros e buscando realizar uma melhoria no desempenho da automatização da escola em questão, melhorando os processos de controles de alunos, matrículas, registro de notas, faltas e principalmente diminuir o tempo de elaboração dos documentos realizados bimestralmente pela secretária através do livro de chamada.

### 3 JUSTIFICATIVA

Para que os problemas e dificuldades da Escola Algodão Doce – E.I.E.F fossem detectados foram realizadas entrevistas nas quais a diretora da instituição forneceu dados importantes para dimensionar este trabalho. Com base nestes dados observados, verificou-se que para melhorar o rendimento desta instituição teria como necessidade básica a implantação de um sistema informatizado. E, para analisar e modelar este sistema, pode-se evidenciar através da revisão literária que os conceitos de UML e BUP poderiam ser aplicados.

Inicialmente a proposta para o processo de desenvolvimento deste projeto foi através do uso do RUP, no entanto durante os estudos desta metodologia esta se revelou inadequada por ser utilizada em grandes projetos. Foi escolhido então o processo BUP, por ser uma metodologia mais simplificada e originada do RUP, aproveitando assim todo o estudo realizado anteriormente.

Para se construir este projeto inicialmente era preciso verificar as necessidades da escola e se realmente um sistema de informática ajudaria a resolve-los, para tanto foi realizada a análise de requisitos.

Segundo Pressman (2002) a análise de requisitos é o primeiro passo para a elaboração de um projeto, nele deverá conter a visão do *software*, e através da análise que o escopo do *software* será refinado. Pois através deste estudo poderá ser observada a análise do sistema atual, seus problemas e dificuldades que foram levantados com a elaboração desta pesquisa, e estão descritos na seção 7.1.2 -Definição de Visão, deste trabalho.

No decorrer do texto poderá ser observado que devido ao tempo escasso para o desenvolvimento deste trabalho conseguiu-se chegar até a fase de elaboração, por isso algumas etapas definidas pelo BUP foram suprimidas.

## 4 OBJETIVOS

### 4.1 Objetivo Geral

O objetivo geral do projeto foi realizar um estudo da linguagem de modelagem UML, e dos processos RUP e BUP, assim como as ferramentas que apóiam o uso dessas técnicas de modelagem para desenvolver um projeto piloto de um sistema de controle acadêmico para Escola Algodão Doce – E.I.E.F.

### 4.2 Objetivos Específicos

1 - Realizar um estudo e levantamento bibliográfico sobre UML, RUP e BUP;

2 - Analisar e modelar o sistema acadêmico para a escola utilizando diagramas de UML seguindo o processo BUP;

3 - Desenvolvimento do Projeto Piloto utilizando a ferramenta de desenvolvimento *Delphi*;

## 5 MATERIAL E MÉTODOS

### 5.1 Material

Nesta pesquisa foram empregados recursos materiais e metodológicos, tais como:

Periódicos;

Artigos;

Livros;

Microcomputador;

Ferramenta de modelagem;

Ferramenta de desenvolvimento;

### 5.2 Metodologia

A UML é uma linguagem de desenvolvimento de projetos sistemas orientados a objeto, comumente utilizados entre analistas e desenvolvedores de *software* que modelam seus projetos através da Engenharia de *Software*, seguindo estes preceitos primeiramente foi realizada a análise de requisitos, e de posse destes foi dado início ao trabalho de construção do sistema, modelando e

representando de forma diagramática os componentes do sistema seguindo o processo indicado pelo BUP.



## 6 FUNDAMENTAÇÃO TEÓRICA

### 6.1 APRESENTAÇÃO DO PROJETO PILOTO

No decorrer desta seção serão apresentados os estudos realizados sobre as seguintes abordagens de processos de desenvolvimento de sistemas: *Unified Modeling Language (UML)*, *Rational Unified Process (RUP)* e o *Basic Unified Process (BUP)*.

#### 6.1.1 Características das Modelagens

Com base no que afirmam Booch, Rumbaugh, Jacobson (2000) o BUP e o RUP estão embasados na UML, pode-se considerar que a importância do uso das duas modelagens se distingue através do processo de ciclo de vida, que tem como meta a entrega do projeto analisado de maneira eficiente e previsível, no qual o produto elaborado atende as necessidades do cliente.

Considera-se que a UML é amplamente independente, pois torna-se possível que esta linguagem seja utilizada de acordo com os recursos que possam ser oferecidos através dela. O RUP consiste na abordagem do ciclo de vida de um *software*, permitindo a criação de um projeto de alta qualidade, o qual deverá atender as necessidades dos usuários finais e que estejam dentro do planejamento e orçamento previsível. O RUP proporciona algumas das melhores práticas atuais de desenvolvimento de *software*, desta forma ele poderá estar interconectado a uma ampla variedade de projetos. Na questão de gerenciamento proporciona um processo disciplinado o qual deverão ser atribuídas tarefas e responsabilidades no

desenvolvimento de um *software* (BOOCH, RUMBAUGH, JACOBSON, 2000).

As atividades do RUP enfatizam a criação de manutenção e modelos que são especificados através da UML, a sua utilização proporciona representações semanticamente ricas do sistema que está sendo desenvolvido.

O BUP é uma versão mais simplificada do RUP que visa a elaboração de projetos de pequenas empresas que constituem por uma equipe que contenha no mínimo 3 a 6 pessoas. Alguns processos do RUP foram suprimidos do BUP, pois seria inviável a utilização destes processos devido ao tempo de execução previsto pelo desenvolvimento de pequenos *Software*, é utilizando as definições do BUP (o tempo de execução varia de 3 a 6 meses).

## 6.2 CICLO DE VIDA DO SOFTWARE

Existem algumas abordagens utilizadas no ciclo de vida do *software* que tem como objetivo enfatizar a importância da modelagem do sistema acompanhando todas as fases de elaboração. Ou seja, um processo para explorar a UML deverá ter as seguintes características (Booch, Rumbaugh, Jacobson, 2000):

- **Orientado a caso de uso** tem como definição à utilização dos principais artefatos para o estabelecimento do comportamento, para a validação e verificação da arquitetura do *software*, no qual serão realizadas a fase de testes e a interação entre os participantes do projeto.
- **Centrado na Arquitetura** quer dizer que a arquitetura do sistema é importante para a construção, o gerenciamento e a evolução do sistema em desenvolvimento.
- **O Processo Iterativo** é aquele que tem como

gerenciamento a seqüência de versões executáveis do processo. O processo incremental se desenvolve a cada integração, pois a cada *software* elaborado poderá ter novas versões para o seu desenvolvimento e aprimoramentos em relação aos demais processos existentes.

Conforme mostra as quatro fases do RUP pôde-se observar que a UML está relacionada com o RUP através do ciclo de vida do projeto e poderá ser observado logo abaixo as quatro fases de elaboração do projeto como ilustra a figura 1:

➤ **Concepção:-** a fase em que a idéia inicial do projeto é elaborada.

➤ **Elaboração:-** Através desta fase será elaborada a visão do produto e sua arquitetura, proporcionando a visão da análise de requisitos funcionais e não funcionais que conterão no sistema.

➤ **Construção:-** Esta fase vai ser a etapa em que o processo chega a uma arquitetura *baseline*, pois aqui são analisadas as necessidades que o sistema possui, solucionando os riscos que o sistema poderá conter.

➤ **Transição:-** nesta fase o *software* chega até as mãos dos possíveis usuários. Este *software* ainda poderá sofrer alterações que no decorrer do estudo não foram observadas.

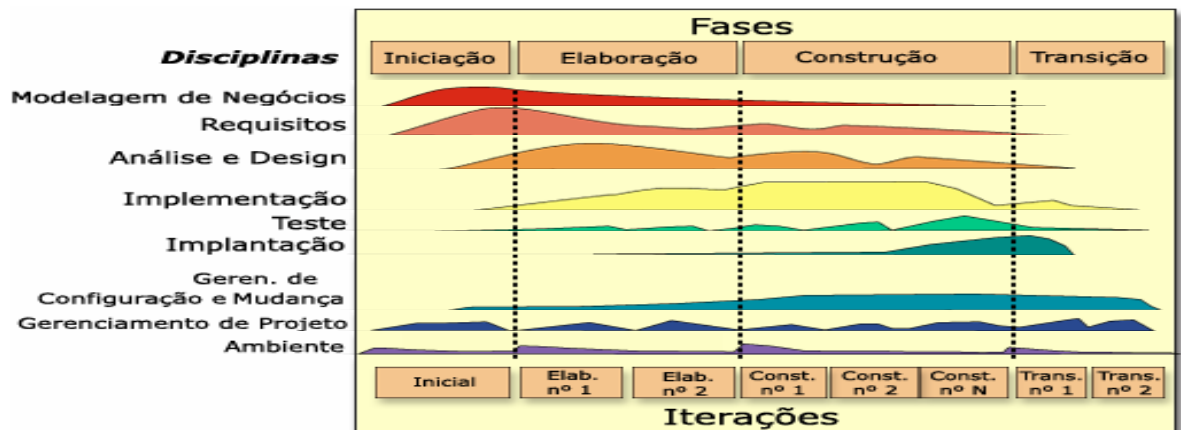


Figura 1 - Estrutura das Fases. Fonte: KRUCHTEN, P. Introdução ao RUP Rational Unified Process. Rio de Janeiro: editora Ciência Moderna, 2003.

### 6.2.1 Composição da UML

Com base nos autores Booch, Rumbaugh, Jacobson (2000) a UML é uma linguagem gráfica que visa a arquitetura de um sistema, proporcionando ao projeto processos que satisfaçam as necessidades do cliente.

Podendo definir basicamente os nove diagramas que a UML contém de acordo com a versão 2.0.

**Diagramas de Casos de Usos:** têm como papel principal a modelagem do sistema ou de um subsistema no qual se visualiza um conjunto de casos de usos e atores e seus relacionamentos que são capazes de visualizar, especificar e documentar o comportamento de um elemento.

**Diagramas de Classes:** visualiza as diferentes classes e como elas se relacionam. Estes diagramas também são importantes não só apenas para a visualização, mas também para a especificação e a documentação. Este modelo de diagrama procura demonstrar os suportes dos requisitos funcionais aos seus usuários finais.

**Diagrama de Objetos:** mostra-se um conjunto de elementos e seus relacionamentos, pois este diagrama direciona o projeto a uma visão estática ou uma visão estática do processo de um sistema, através dos diagramas de classes, considerando casos reais ou protótipos.

**Diagrama de Gráfico de Estado:** este diagrama é relacionado à uma máquina de estados, onde suas transições, eventos e atividades são elaboradas. Este diagrama é importante para a criação da modelagem do comportamento de uma interface, classe ou colaboração.

**Diagrama de Seqüência:** diagrama que visa à interação, onde as ordenações temporais das mensagens são instâncias nomeadas ou anônimas de classes, podendo também representar instâncias relacionadas a outros itens como colaborações, componentes e nós. Este diagrama tem como objetivo ilustrar a visão dinâmica de um sistema.

**Diagrama de Colaboração** tem como finalidade nos demonstrar as interações que ocorrem entre os objetos e sua organização, no qual esses diagramas enviam e recebem mensagens, tendo como um conjunto de objetos e suas conexões que ocorrem através destas mensagens.

**Diagrama de Atividade** auxilia na criação do fluxo em que essas atividades são transmitidas para outros sistemas. Sendo assim, estas atividades nos demonstram um fluxo seqüencial ou ramificado de uma atividade para outra atividade no qual os objetos que existem nelas sofrem alterações.

**Diagrama de Componente** este diagrama visualiza um conjunto de componentes e seus relacionamentos, contudo está relacionado junto a este diagrama de componentes o diagrama de funcionamento.

**Diagrama de Implantação:** é um conjunto de nós e seus relacionamentos, em que os diagramas de funcionamento estarão relacionados aos diagramas de componentes, pois tipicamente um nó contém um ou mais componentes.

## 6.3 FASES DO DESENVOLVIMENTO DE UM SISTEMA

### 6.3.1 Análise de Requisitos

O processo de identificação dos requisitos tem como necessidade à elaboração de uma entrevista para que se possa obter o máximo de informações possíveis para a construção de um projeto. E, para que o desenvolvedor tenha uma compreensão adequada do sistema, é necessário que a análise de requisitos seja feita de acordo com que o cliente deseja, obedecendo todas as suas restrições para que o cliente e o desenvolvedor tenham um ótimo relacionamento (PFLEEGER, 2004).

O mesmo autor comenta que a Identificação dos requisitos é um processo essencial na análise de requisitos, pois, através desta etapa poderá ser identificado o problema que o cliente requereu para uma determinada solução, que viabiliza uma análise de alto desempenho pelo analista do sistema para que ele possa solucionar o problema proposto pelo cliente e este deverá estar de acordo com o pedido solicitado.

Para entender melhor a necessidade do cliente os requisitos são classificados em dois tipos de documentos: a definição dos requisitos, que através dela poderá ser definido um conjunto completo de tudo o que o cliente espera do sistema, cujo documento de visualização do sistema é escrito em comum acordo entre cliente e desenvolvedor, e a especificação dos requisitos, que é escrita pelos analistas em termos técnicos apropriados para o desenvolvimento do sistema (PFLEEGER, 2004).

Os requisitos têm como intuito descrever o comportamento do sistema, os quais são classificados de duas maneiras: Os requisitos funcionais que através deles poderão ser identificada a interação que ocorre entre o sistema e o ambiente em que será projetado o sistema. E os requisitos não funcionais que

especificam as restrições que conterão no sistema permitindo limitar as opções para criar uma nova solução para o sistema (PFLEEGER, 2004).

### **6.3.2 Análise**

Nesta fase procura-se identificar as possíveis abstrações (classes e objetos) e os mecanismos existentes no problema a ser resolvido (*apud* Larman, 2000). As possíveis classes dos Diagramas de Classes existentes demonstram as descrições que auxiliam na elaboração dos Casos de Uso, “podendo” aquelas também estarem interconectados umas nas outras através de possíveis relacionamentos. Afinal, nesta fase será modelado somente aquelas classes que fazem parte do domínio principal do problema, ou seja, “classes técnicas mais detalhadas não estarão neste diagrama” (ALMEIDA, DAROLT, 2001).

### **6.3.3 Design (Projeto)**

A fase do projeto é a criação de uma representação de um determinado domínio de um problema que através dele será processado os resultados obtidos na fase de análise. “O Projeto cria uma representação do domínio do problema do mundo com a elaboração do projeto obtemos detalhadamente as especificações para dar início a fase de programação” (ALMEIDA, DAROLT, 2001).

### 6.3.4 Programação

Para que nesta fase se possa ter um bom desempenho, é necessário, que o projeto tenha sido bem elaborado, afinal as classes são convertidas da “fase do projeto para o código da linguagem Orientada a Objeto escolhida” (*Apud Furlan, 1998*). Nesta fase há uma possível complexibilidade de conversão dependendo da capacidade e dificuldade da linguagem escolhida tornará o projeto fácil ou difícil de ser realizado (*ALMEIDA, DAROLT, 2001*).

### 6.3.5 Testes

A fase de teste tem como objetivo verificar se existe algum erro no programa, procurando testar todas as rotinas e processos de maneira detalhada, na averiguação da integração dos processos e a aceitação, sendo que as rotinas deverão ser testadas através dos programadores na qual eles irão demonstrar os seus conhecimentos nas possíveis classes e grupo de classes que envolvam uma rotina, “sabendo desta forma onde (sic) estão os pontos mais críticos que podem causar falhas”. O teste de integração é elaborado segundo as classes e seus componentes de integração no qual estes poderão verificar se as suas classes estão realmente colaborando uma com as outras conforme o que foi especificado no modelo. Nos testes de aceitação será verificado se o sistema está de acordo com o que foi especificado no Diagramas de Caso de Uso. Porém este sistema poderá ser testado pelo usuário final e o mesmo verificar se os resultados obtidos estão de acordo com as intenções expostas no início do projeto (*ALMEIDA, DAROLT, 2001*).



## 6.4 RATIONAL UNIFIED PROCESS (RUP)

Segundo o autor Kruchten, (2003) o RUP tem por finalidade demonstrar para o usuário a usabilidade do sistema e seus casos essenciais. Através deste estudo poderá ser evidenciada a importância da análise de requisitos que deverá ser demonstrada através de um caso de uso de um ator interagindo com o sistema que terá seu próprio desempenho e o seu papel.

Algumas das melhores práticas de desenvolvimento de *software* vão ser citadas a seguir:

**Gerenciamento de Requisitos** é definido como uma condição da capacidade de gerenciar e definir as variáveis de um sistema. Os principais itens para o gerenciamento de atributos incluem o tipo de requisitos para cada rastreabilidade.

**Usar Arquiteturas Baseadas em Componente** são os grupos de código coesos, na forma de código fonte ou executável, com interfaces bem definidas e comportamentos que fornecem forte encapsulamento do conteúdo e são, portanto, substituíveis. As arquiteturas baseadas em componentes tendem a reduzir o tamanho efetivo e a complexidade da solução e, portanto, são mais resistentes e flexíveis.

**Modelar Visualmente o Software** um modelo de sistema descreve a realidade e a perspectiva de um determinado *Software* que será construído. É importante modelar porque ajuda a ter uma visão mais ampla do projeto, também auxilia na especificação, construção, documentação e na estrutura e comportamento da arquitetura do sistema.

**Verificar Continuamente Qualidade do Software** a localização e a solução dos problemas de *software* ficam de 100 a 1.000 vezes mais caros, se realizados após a implantação. A verificação e o gerenciamento da qualidade

durante o ciclo de vida do projeto são essenciais para atingir os objetivos corretos no momento certo.

**Controlar mudanças no *Software*** oferece várias soluções para as causas originais de problemas de desenvolvimento de *software*:

- O fluxo de trabalho da mudança de requisitos é definido e pode ser repetido;
- As solicitações de mudança facilitam a comunicação clara;
- Os espaços de trabalho isolados reduzem a interferência entre membros da equipe que trabalham em paralelo;
- As estatísticas de taxa de mudanças fornecem métricas satisfatórias para avaliar objetivamente o status do projeto;
- Os espaços de trabalho contêm todos os artefatos, o que facilita a consistência;
- A propagação da mudança pode ser avaliada e controlada;
- As mudanças podem ser mantidas em um sistema robusto e personalizável.

## **6.5 BASIC UNIFIED PROCESS (BUP)**

### **6.5.1 Definição do BUP**

Balduino [20-?] comenta o fato de o BUP ser organizado de duas formas diferentes, através do método e processo. Método é a dimensão dos elementos que possuem papéis, tarefas, artefatos, orientação e estes estão definidos conforme a aplicação do Projeto. E a dimensão do Processo é o elemento aplicado a um comportamento dos métodos que através deles poderão ser criados diferentes tipos de projetos.

#### **6.5.1.1 Disciplinas**

O BUP é visualizado através dos conteúdos das seguintes disciplinas: requisitos, arquitetura, desenvolvimento, teste, administração de projeto e administração de mudança. Sendo estas disciplinas reorganizadas em pacotes permitindo que o conteúdo desejado possa sofrer alterações.

Existem outras disciplinas que poderão ser aplicadas, não necessariamente pela equipe de projeto, como: a área de preocupação, negócio, ambiente, administração de exigências avançada, administração de configuração.

A análise e o *design* não são observados através de uma disciplina separada, mas sim absorvido por disciplinas que contenham arquitetura e desenvolvimento. O conteúdo abstraído pelo arquiteto serve para elaborar a análise quando ela é identificada por abstrações fundamentais na formação das camadas,

usando novamente as soluções existentes e seus padrões.

### 6.5.1.2 Papéis

As possíveis habilidades essenciais que serão denominadas através das representações dos papéis que o BUP oferece são:

- **Analista:** é o responsável para documentar o projeto a ser desenvolvido. Em projetos pequenos, até mesmo um representante do cliente poderá fazer o papel do analista.
- **Arquiteto:** é o responsável para criação da arquitetura do *software* este deverá tomar as decisões de acordo com as restrições do sistema.
- **Desenvolvedor:** tem como objetivo criar a solução do projeto, testes de unidades e integração de componentes.
- **Testador:** é o responsável por testar o sistema e averiguar se as perspectivas do sistema esta sendo atingida como foi definida pelo cliente.
- **Gerente de Projeto:** administra o projeto para que os planos possam ser seguidos.
- **Papéis:** este irá representar as pessoas que executaram realizaram as tarefas para a execução do *software*, submetendo as mudanças, participando das reuniões e revisando as tarefas.

### **6.5.1.3 Tarefas**

As tarefas consideradas pelo BUP são tipicamente utilizadas por projetos pequenos. Algumas dessas tarefas formam diretrizes, e são referenciadas através de tarefas mais simples.

Algumas das tarefas do RUP foram transformadas em passos menores e incluídas dentro de uma tarefa principal executada pelo mesmo papel. De acordo com o projeto pode ser definido qual o passo executado ou não, dependendo do que ele necessita.

### **6.5.1.4 Artefatos**

Abaixo serão definidos os artefatos que contêm o BUP. O artefato possui diretrizes que representam as informações capturadas dos artefatos existentes, planilha eletrônica, banco de dados. Permitindo assim selecionar os projetos no nível apropriado de acordo com seus artefatos. As fases do BUP serão demonstradas logo abaixo:

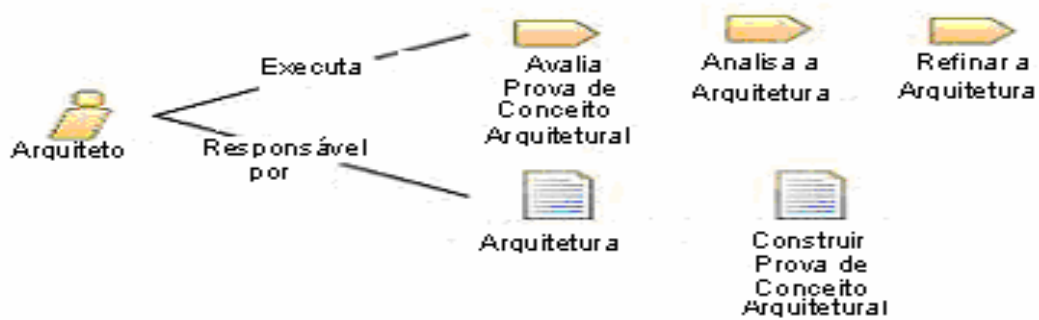


Figura 2: Definição de Arquitetura de Software. Fonte: BALDUINO, R. Basic Unified Process: A Process for Small and Agile Projects, [20-?]. Disponível em: <<http://www.eclipse.org/proposals/Beacon/>>. Acesso em 12/09/2006 hs.



Figura 3: Definição de Desenvolvedor de Software. Fonte: BALDUINO, R. Basic Unified Process: A Process for Small and Agile Projects, [20-?]. Disponível em: <<http://www.eclipse.org/proposals/Beacon/>>. Acesso em 12/09/2006 hs.



Figura 4: Definição de Plano de Gerente. Fonte: BALDUINO, R. Basic Unified Process: A Process for Small and Agile Projects, [20-?]. Disponível em: <<http://www.eclipse.org/proposals/Beacon/>>. Acesso em 12/09/2006 hs.



Figura 5: Definição de Testador de Software. Fonte: BALDUINO, R. Basic Unified Process: A Process for Small and Agile Projects, [20-?]. Disponível em: <<http://www.eclipse.org/proposals/Beacon/>>. Acesso em 12/09/2006 hs.

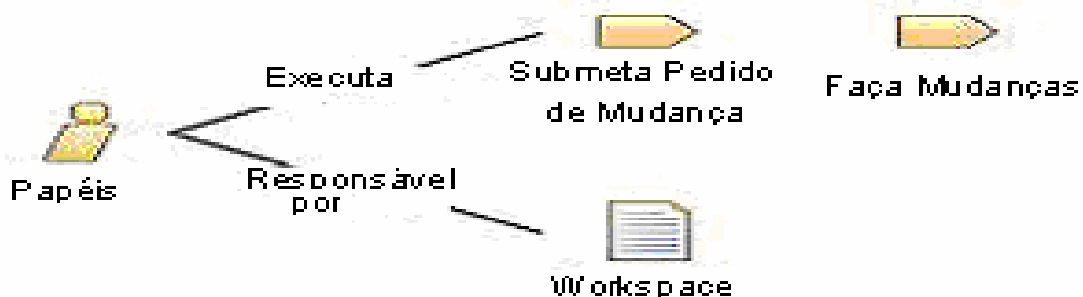


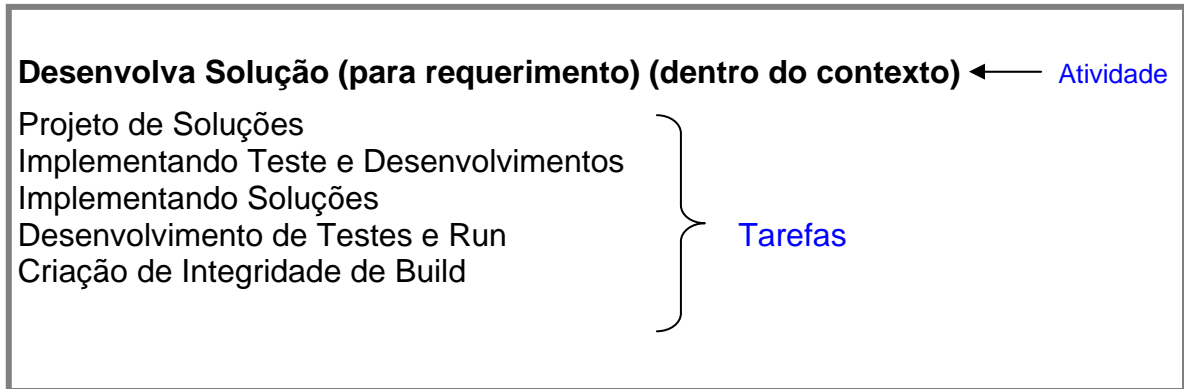
Figura 6 – Definição de Papéis. Fonte: BALDUINO, R. Basic Unified Process: A Process for Small and Agile Projects, [20-?]. Disponível em: <<http://www.eclipse.org/proposals/Beacon/>>. Acesso em 12/09/2006 hs.

### 6.5.1.5 Processo

O processo é um método que pode ser reutilizado e criado conforme a sua aplicação. O conteúdo do processo descreve passo-a-passo como são alcançadas as metas de desenvolvimento do projeto.

Os elementos dos processos são relacionados em sucessões personalizadas para cada projeto. Elementos do método são organizados em partes reutilizáveis do processo chamado de capacidade de padrões. Esses padrões são as atividades que reorganizam as tarefas e os grupos no qual ele será aplicado. Os

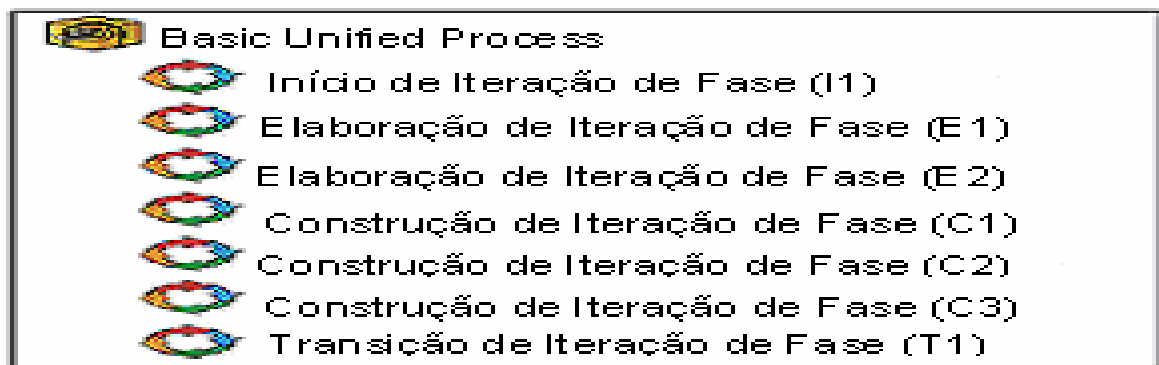
Padrões devem focalizar as áreas em particular como: o projeto, definição da arquitetura e assim por diante. Estes blocos são definidos para criar padrões de processo.



Fonte: BALDUINO, R. Basic Unified Process: A Process for Small and Agile Projects, [20-?]. Disponível em: <<http://www.eclipse.org/proposals/Beacon/>>. Acesso em 12/09/2006 hs.

Quadro 1: Blocos

O Gerente de projeto usa este padrão para executar o seu planejamento baseado nas metas alcançadas. O contexto será especificado quando for desenvolvido ou não. Então a responsabilidade deste será descrever e elaborar os testes de unidade. E para que possa ser evitado o plano detalhado, o gerente do projeto elabora um plano de incremento para que as tarefas possuam um guia de como executar o trabalho.



Fonte: BALDUINO, R. Basic Unified Process: A Process for Small and Agile Projects, [20-?]. Disponível em: <<http://www.eclipse.org/proposals/Beacon/>>. Acesso em 12/09/2006 hs.

Quadro 2: Exemplo de Entrega de Processo.



## **6.6 PLANO DE VISÃO**

### **6.6.1 Introdução**

A análise deste sistema foi realizada através da perspectiva da solicitação do cliente perante as suas necessidades que são descritas dentro do processo do BUP. A Instituição de ensino tem como objetivo melhorar o processo de elaboração da documentação da escola em que os funcionários e professores possam desempenhar seus papéis, independentes dos transtornos ocorridos atualmente por ter que fazer o mesmo serviço várias vezes. Para alcançar esse objetivo, a Instituição conta com uma equipe de funcionários centrada no desenvolvimento do processo de execução destes documentos.

#### **6.6.1.1 Definição de Plano de Visão**

A visão é definida através das pessoas envolvidas com o sistema, e o produto a ser desenvolvido, abrangendo as necessidades e características do sistema. De acordo com a definição do Plano de Visão este deverá conter as descrições dos requisitos no qual poderá nos proporcionar uma visão mais detalhada do produto em si.

Para que o documento de visão possa ter um bom entendimento deve-se seguir as seguintes características:-

1 Estabelecer um acordo sobre o problema que está sendo resolvido.

2 Identificar as pessoas que estão envolvidas diretamente com o sistema.

3 Definir os limites do Sistema.

4 Identificar as restrições que o sistema deverá conter.

5 Relatar sobre o problema que está sendo exposto.

6 Definir os recursos utilizados no sistema.

Para que se tenha uma dimensão mais clara do Plano de Visão poderá ser observada uma breve definição dos itens que está sendo utilizado para a montagem do projeto.

**Posicionamento:** é a definição da análise de requisitos para que o analista do sistema possa compreender o que está sendo proposto pelo cliente.

**Descrição do problema:** é a descrição sobre o problema que está sendo resolvido.

**Sentença de Posição do Produto:** é a posição em que o produto pretende estar assumindo no mercado.

**Descrições dos Envolvidos é dos Usuários:** é uma maneira de estar representando às reais necessidades do usuário e identificando os envolvidos que vão interagir com o sistema. Pois deverá também ser especificado o envolvido e uma breve descrição de suas responsabilidades.

**Ambiente do Usuário:** é a definição do ambiente físico, as pessoas envolvidas, tempo de duração das tarefas elaboradas no ambiente de trabalho e o tipo de plataforma de sistema a ser utilizado.

**Visão Geral do Produto:** fornece para o analista uma visão mais detalhada dos recursos utilizados para a elaboração do projeto. Viabilizando as suas perspectivas e suposições de dependências do produto. Através destas poderá também ser definidos as necessidades e características, Alternativas e Competição e os Recursos Utilizados pelo Produto.

## 6.7 PLANO DE PROJETO

O Planejamento de Projeto é de fundamental importância para o gerente administrar o tempo, pessoas, equipamentos e estimar com uma margem maior de tempo o período necessário para execução do projeto, pensar em futuros problemas que pode enfrentar.

## 6.8 DESCRIÇÃO DA ADMINISTRAÇÃO DE REQUISITOS

O caso de uso apresentado para este projeto trata-se das funcionalidades que conterà no sistema acadêmico, ou seja, o Caso de Uso gerado é uma visão geral do sistema acadêmico proposto para a Escola Algodão Doce que está disponível para mais detalhes no apêndice B.

Os casos de uso gerados para este diagrama são os seguintes:-  
Mantém Cadastro de Instituição, Mantém Cadastro de Funcionário, Mantém Usuário

do Sistema, Mantém Cadastro do Aluno, Mantém Cadastro de Disciplina, Mantém Cadastro de Frequência, Mantém Cadastro de Nota, Mantém Cadastro de Turma, Realiza Matrícula, Mantém Cadastro de Avaliação, Realiza Relatório Parcial e Realiza Relatório Final. E os atores gerados são os seguintes: Administrador, Professor e Secretária.

## **6.9 DESCRIÇÃO DO PLANO DE ITERAÇÃO**

O Plano de Iteração tem como objetivo validar os requisitos do sistema é descrever os principais casos de uso para que a viabilidade e a arquitetura do *software* possa ser testada.

## 7 DESENVOLVIMENTO DO TRABALHO

No decorrer desta seção serão descritas algumas etapas de aplicação do BUP. Devido a este ser um projeto piloto e com apenas um integrante alguns processos foram suprimidos.

### 7.1 PLANO DE VISÃO

#### 7.1.1 Introdução

#### 7.1.2 Posicionamento

Através da Análise do sistema atual da Escola Algodão Doce pode-se perceber as dificuldades que os funcionários enfrentam para elaborar relatórios e controlar e organizar os diversos registros de forma manual. Os controles realizados manualmente são:

**Cadastro dos Alunos:** é feito pela secretária o qual o pai, mãe ou responsável irá informar todos os dados pessoais do aluno que serão transcritos posteriormente nas fichas.

**Livro de Chamada:** cabe ao professor fazer o registro das notas, freqüência dos alunos e a escrituração dos conteúdos para que os registros escolares obedeçam ao prazo estipulado.

**Cadastro Nota:** é feito bimestralmente pela professora de cada disciplina através do livro de chamada. E tem como objetivo verificar o rendimento escolar do aluno.

**Resultado Final:** é feito pela secretária a cada final de bimestre no livro de chamada onde estarão registradas as informações dos alunos durante os 4 bimestres. E, ao final do ano, deverão ser somadas estas 4 notas que o aluno obteve correspondente à matéria especificada. A finalidade do Relatório Final é demonstrar a situação do aluno no ano letivo se ele foi Aprovado ou Reprovado.

**Freqüência:** serão registradas no livro de chamada pela professora, na qual verifica-se a cada bimestre quantas faltas o aluno teve e, ao final junto com o relatório de notas vai informar se o aluno não excedeu o número máximo de faltas.

**Conceito:** é registrado o parecer da professora referente ao resultado final mais as faltas do aluno durante os 4 bimestres e, em seguida, passado para a secretária preencher a ficha individual do aluno que contém as informações que estão registradas no livro de chamada como nota e freqüência. Tem como objetivo demonstrar o rendimento escolar do aluno a cada bimestre e anualmente.

**Matrícula:** Cabe a secretária preencher a ficha de matrícula de cada aluno no início do ano letivo informando a série que o aluno irá freqüentar, sendo que esta ficha só será preenchida caso o aluno estudou na escola durante o ano anterior. Caso o aluno não estudou nesta escola, é necessário fazer o cadastro do aluno no qual conterà todos os dados do aluno citados anteriormente. A ficha de matrícula e matrícula tem como objetivo ter o máximo de informações possíveis do aluno para que, quando necessário, esteja disponível.

**Desistência:** é quando o aluno para de freqüentar a escola. O professor deverá informar a secretária se o aluno está freqüentando as aulas ou não, porém cabe a secretária também observar a situação do aluno na escola.

**Relatório:** a cada final de ano a secretária tem a função de fazer o

relatório final que o Núcleo Regional de Ensino (NRE) solicita. Neste relatório constará a situação do aluno durante o ano letivo. Para que o relatório possa ser feito, é necessário olhar pasta por pasta para verificar os dados dos alunos. No relatório final verifica-se a turma, as notas durante os quatros bimestres, a situação do aluno, se ele foi aprovado ou reprovado, média final de cada aluno por disciplina, a qual deverá ser maior que 6.0 para que o aluno seja aprovado e não poderá ter freqüência menor que 75% pois se tiver, ele vai ser reprovado por falta.

### 7.1.2.1 Descrição do Problema

**Tabela 1 - Levantamento dos Problemas e Impactos Ocorridos na Escola Algodão Doce.**

<b>O problema de</b>	Os principais Problemas e dificuldades que afetam os funcionários da escola são: - Atraso na entrega de relatórios necessários internamente ou solicitados por outra instituição como o Núcleo Regional de Educação (NER). - Todos os cadastros e processos que a escola deve realizar como obrigação ou por controle são realizados de forma manual.
<b>Afeta</b>	- Diretora - Secretária - Professores
<b>Cujo impacto é</b>	- O atraso na entrega de relatórios causa morosidade e perda de informações, causando conflitos entre funcionários e a perda de credibilidade da escola por outras instituições. - Os cadastros sendo realizados de forma manual acarretam morosidade nos processos internos aumentado o custo operacional e também apresentando uma fragilidade em relação a segurança da informação.
<b>Uma boa solução seria</b>	- A informatização dos processos de cadastro e extração de relatórios poderia agilizar o fluxo de informação e como conseqüência diminuir os custos operacionais.

### 7.1.2.2 Definição de Posição do Produto

**Tabela 2 - Posição do Produto**

<b>Para</b>	Escola Algodão Doce
<b>Quem</b>	Secretária, Diretora e Professores
<b>Sistema Acadêmico</b>	Projeto Piloto de um sistema de gestão acadêmico para pequenas escolas
<b>Que</b>	Disponibiliza os dados dos alunos para facilitar o gerenciamento dos cadastros e a extração dos relatórios e consultas.
<b>Ao contrário</b>	Ter que fazer estes cadastros, relatórios e controles por meio manual ou controlar através do sistema disponibilizado pelo NRE que possui características muito complexas para pequenas escolas.
<b>Nosso produto</b>	No sistema será registrada as informações por meio informatizado, estas informações poderão ser formatadas rapidamente e disponibilizadas através de relatórios de maneira bem simplificada.

### 7.1.3 Descrições dos Envolvidos e dos Usuários

Para que se tenha uma visão mais detalhada sobre os respectivos envolvidos do sistema poderá ser observado qual e a posição de cada envolvido e usuário deste sistema.

A Diretora tem a funcionalidade de dar as coordenadas para os seus funcionários.

A função do Professor e avaliar o aluno e ter o registro das notas e faltas em dia para que a Secretária possa ter em mãos quando necessitar.

A função da Secretária e manter o cadastro dos alunos em dia para que quando necessário possa estar disponível.



### 7.1.3.1 Resumo dos *Stakeholders*

**Tabela 3 - Representa a Descrição dos Envolvidos e Usuários**

<b>Nome</b>	<b>Descrição</b>	<b>Responsabilidades</b>
Patrícia Pitoli Yamagami	Secretária/ Diretora	Responsável pelo preenchimento dos cadastros e elaboração dos documentos dos alunos solicitados pelo NRE.
Joyce Eli Jofre Maria Delfina Borges Anacleto Maria José Pedroso	Professores	São responsáveis pelo registro das notas, freqüências e elaborar o resultado final do aluno durante os bimestres registrando essas informações através do livro de chamada.

### 7.1.4 Visão Geral do Produto

O sistema roda em ambiente gráfico *Windows* e funciona em 1 computador disposto em 1 sala que será usado pela Diretora, Secretária e Professores da Escola.

#### 7.1.4.1 Perspectiva do Produto

O sistema implantado nesta instituição foi criado de acordo com as reais necessidades dos envolvidos, pois esta escola não possuía qualquer tipo de recursos computacionais, e devido a falta de cultura em informática o sistema proposto é a melhor alternativa por ser mais simplificado do que o sistema disponibilizado pelo NRE.

#### **7.1.4.2 Suposições e Dependências**

Poderá ocasionar que com o decorrer do tempo os requisitos do sistema poderão sofrer algumas alterações propostas pelos membros da instituição ou de forma externa como requisitos legais, e conforme o impacto destes o sistema sofrerá alterações.

### 7.1.4.3 Necessidades e Características

**Tabela 4 - Descrição das Necessidades e Características**

<b>Necessidades</b>	<b>Prioridade</b>	<b>Características</b>	<b>Liberação Planejada</b>	<b>Soluções Propostas</b>
Este Projeto Piloto tem como necessidade Manter os seguintes Cadastros: da Instituição, Funcionário, Professor, Usuário do Sistema, Aluno, Disciplina, Frequência, Nota, Turma, Avaliação e deverá Realizar Matrícula, Relatório Final e Relatório Parcial.	Neste item serão citados os casos de usos de maior importância: Cadastro de Aluno, Cadastro de Disciplina, Cadastro de Frequência, Cadastro de Nota.	<ul style="list-style-type: none"> <li>- Mantém Usuário do Sistema: terá como definição fazer o cadastramento dos funcionários que terão acesso ao sistema.</li> <li>- Mantém Cadastro da Instituição: este terá como função básica realizar o cadastro da Instituição que adquiriu este Sistema.</li> <li>- Mantém Cadastro de Funcionário: irá realizar um novo cadastro de funcionário, cajo este seja um Professor através deste poderá ser elaborado o Cadastro de Professor.</li> <li>- Cadastro de Aluno serve para inserir o cadastro de um novo Aluno. Após ter sido realizado este cadastro estará disponível a Matrícula do Aluno.</li> <li>- Cadastro de Disciplina servirá para fazer o cadastro das disciplina exigidas pelo NRE na carga Horária dos Alunos.</li> <li>- Cadastro de Frequência terá como funcionalidade inserir a quantidade de presença e faltas em uma determinada disciplina.</li> <li>- Cadastro de Avaliação: terá como objetivo registrar as avaliações bimestrais. Após ter sido registrado estas poderá então fazer o cadastro das notas dos alunos.</li> <li>- Relatório Parcial: será elaborado bimestralmente conforme a Situação do Aluno. Já o Relatório Final será realizado a partir do Relatório Parcial no qual será informado a situação do Aluno durante o ano todo.</li> </ul>	- Deverá ser elaborada de acordo com as necessidades do cliente.	Através dos Casos de Uso levantados e a montagem do Projeto Piloto pode-se averiguar que com o registro dos dados dos alunos disponibilizados tornará mais fácil a elaboração dos documentos solicitados pelo NRE.

#### **7.1.4.4 Alternativas e Competição**

O NRE oferece para todas as escolas um Sistema Acadêmico gratuito com todas as funcionalidades. Mas a Escola em questão é uma instituição com uma média de 50 alunos anuais. A diretora desta escola não adquiriu a esta tecnologia por se tratar de um sistema complexo que não seria viável a esta instituição. Pois o sistema é utilizado por instituições que contêm em média mais de 300 alunos.

#### **7.1.5 Requerimentos do Produto**

- Hardware 256 MB RAM;
- Velocidade do Processador: mínimo de 500 mhz
- Espaço em disco de 1 GB recomendável;
- Sistema Operacional: Windows 2000, XP ou superior.
- Banco de Dados utilizado Interbase 6.0

## 7.2 PLANO DE PROJETO

### 7.2.1 Introdução

### 7.2.2 Organização do Projeto

#### 7.2.2.1 Estrutura Organizacional

O gerente deste sistema teve como finalidade comandar a tramitação da construção deste *software* para que este projeto revelasse uma boa qualidade.

Com base no Plano de Projeto o gerente deve estipular em quanto tempo o *software* ficará pronto e prever as estimativas dos riscos do produto. O projeto foi desenvolvido por apenas uma pessoa com papel fundamental para a criação deste *software* cuja elaboração do mesmo compõe a documentação do sistema e o desenvolvido de um projeto piloto de um *software* acadêmico.

**Tabela 5 - Representação de Gerente de Projeto.**

Gerente do Projeto Maria Virgínia
--------------------------------------

### 7.2.2.2 Interfaces Externas

Através das pesquisas elaboradas evidenciam-se os tipos de documentos solicitados e expedidos que caracterizam a situação do aluno. O NRE solicita os seguintes documentos para a escola Cadastro dos Alunos, Boletim Anual e Bimestral. Também são enviados pela escola os boletins dos alunos para os seus pais ou responsável.

### 7.2.2.3 Papéis e Responsabilidades

**Tabela 6 - Papéis e Responsabilidades dos Responsáveis pelo Projeto.**

<b>Pessoas</b>	<b>Papéis</b>
Maria Virgínia Andreotti Mendes Bordin	Analista Arquiteto Desenvolvedor Testador Gerente de Projeto

### 7.2.3 Programa de Projeto

**Tabela7 - Estimativa do Projeto**

<b>Fases</b>	<b>Iteração</b>	<b>Primeiro Objetivo (riscos / trabalhos)</b>	<b>Programa Início / Fim</b>	<b>Estimativa de esforço (dias por pessoas)</b>
Iniciação	0	-Levantamento das Necessidades -Análise de Requisitos -Modelagem	Início: 01/03 Fim: 31/03 Início: 01/04 Fim: 29/04 Início: 02/05 Fim: 31/07	Estas fases serão realizadas por apenas 1 pessoa
Elaboração	1	- Criar o protótipo do Sistema Acadêmico	Início: 01/10 Fim: 31/10	Esta fase será realizada por apenas 1 pessoa
Desenvolvimento	1	-Construção do módulo de cadastro da instituição	Início: 01/10 Fim: 03/10	Esta fase será realizada por apenas 1 pessoa
Desenvolvimento	1	- Construção do módulo de cadastro de funcionário	Início: 01/10 Fim: 03/10	Esta fase será realizada por apenas 1 pessoa
Desenvolvimento	1	- Construção do módulo de cadastro de professor	Início: 04/10 Fim: 05/10	Esta fase será realizada por apenas 1 pessoa
Desenvolvimento	1	- Construção do módulo de usuário do sistema	Início: 06/10 Fim: 08/10	Esta fase será realizada por apenas 1 pessoa
Desenvolvimento	1	- Construção do módulo de cadastro do aluno	Início: 09/10 Fim: 10/10	Esta fase será realizada por apenas 1 pessoa
Desenvolvimento	1	- Construção do módulo de cadastro de disciplina	Início: 11/10 Fim: 12/10	Esta fase será realizada por apenas 1 pessoa
Desenvolvimento	1	- Construção do módulo de cadastro de frequência	Início: 13/10 Fim: 15/10	Esta fase será realizada por apenas 1 pessoa
Desenvolvimento	1	- Construção do módulo de cadastro de nota	Início: 16/10 Fim: 18/10	Esta fase será realizada por apenas 1 pessoa
Desenvolvimento	1	- Construção do módulo de realiza matrícula	Início: 19/10 Fim: 21/10	Esta fase será realizada por apenas 1 pessoa
Desenvolvimento	1	- Construção do módulo de cadastro de avaliação	Início: 25/10 Fim: 28/10	Esta fase será realizada por apenas 1 pessoa
Desenvolvimento	1	- Construção do módulo de relatório parcial	Início: 29/10 Fim: 31/10	Esta fase será realizada por apenas 1 pessoa
Desenvolvimento	1	- Construção do módulo de relatório final	Início: 29/10 Fim: 31/10	Esta fase será realizada por apenas 1 pessoa
Construção	-----	-----	-----	-----
Transição	-----	-----	-----	-----

### 7.2.4 Orçamento

Não houve gastos significantes para o desenvolvimento deste *software*.

### 7.2.5 Lista de Risco

**Tabela 8 - Lista de Riscos do Projeto.**

<b>Descrição</b>	<b>Tipo</b>	<b>Probabilidade</b>	<b>Impacto</b>	<b>Importância</b>
Tecnologia	Projeto	Baixa	Catastrófico	- Futuramente o Banco de dados <i>Interbase</i> correrá o risco de não poder processar tantas transações por segundo, como se espera, devido ao aumento de matrículas realizadas pela secretária.
Pessoal	Projeto	Alta	Catastrófico	- Treinamento e tempo necessário não estão disponíveis.
Requisitos	Projeto e Produto	Moderada	Sérios	- Uma análise de requisitos mal feita faz com que haja algumas funções inadequadas no sistema, fazendo com que se gaste muito com suporte técnico = manutenção.
Estimativas	Projeto/ Produto	Alta	Sérios	- O tempo requerido para o desenvolvimento do <i>software</i> foi maior que o esperado.
Atrasos na Especificação	Projeto/ Produto	Moderada	Sérios	- As especificações da interface essenciais não estão disponíveis dentro do prazo



## 7.3 ADMINISTRAÇÃO DE REQUISITOS

### 7.3.1 Caso de Uso: Mantêm Cadastro da Instituição

**Descrição:** caso de uso que permite ao Administrador, desde que logado corretamente, a manutenção do cadastro da Instituição.

**Atores:** Administrador.

**Fluxo Básico de Eventos:-** para atender os possíveis casos de uso foram necessários os cenários que serão demonstrados logo abaixo:

**1) Incluir:** O Administrador poderá realizar a inserção do cadastro da Instituição.

**2) Alterar:** O Administrador pode fazer uma busca filtrada pelo nome da Instituição, resultando em uma tela que apresentará os dados atuais da Instituição. Possibilitando em seguida a alteração do dado desejado.

**Fluxo Alternativo:** para o funcionamento de cadastro da Instituição:

Se houver algum dado que não foi processado pelo sistema, automaticamente este emitirá uma mensagem informando o erro.

Se na verificação de consultar Instituição não for encontrado nenhum dado especificado pelo usuário, o sistema deverá emitir uma mensagem de que não existe aquele dado.

### 7.3.2 Caso de Uso: Mantêm Cadastro de Funcionários

**Descrição:** Caso de uso que permite ao usuário, desde que logado corretamente, a manutenção do cadastro dos funcionários que interagem com o sistema.

**Ator:** Administrador.

**Fluxo Básico de Evento:-** para atender os possíveis casos de uso foram necessários os cenários que serão demonstrados logo abaixo:

**Incluir:** O usuário que efetuará o cadastro dos funcionários será o administrador e este, será possibilitado de realizar a inserção de uma conta de um novo funcionário para a Instituição.

**1) Alterar:** O usuário pode fazer uma busca filtrada por nome do funcionário, resultando em uma tela que apresentará os dados atuais do funcionário. Possibilitando em seguida a alteração do dado desejado.

**2) Desativar:** O usuário pode fazer uma busca filtrada por nome, resultando em uma tela que apresentará os dados atuais do funcionário, possibilitando em seguida a desativação do mesmo do sistema.

**3) Consultar:** Nesta área usuário logado como Administrador pode realizar uma busca filtrada de todos os funcionários registrados na base de dados.

**Fluxo Alternativo:** para o funcionamento do cadastro de Funcionário:

Se houver algum dado que não foi processado pelo sistema, automaticamente este emitirá uma mensagem informando o erro.

Se na verificação de consultar funcionário não for encontrado nenhum utilizando os dados informados ao sistema este deverá emitir uma

mensagem de que não existe aquele funcionário.

Se houver algum erro na exclusão dos dados, emitir uma mensagem informando que os dados não foram excluídos.

### 7.3.3 Caso de Uso: Mantêm Cadastro de Professor

**Descrição:** Caso de uso que permite ao usuário, desde que logado corretamente, a manutenção do cadastro do professor que interagem com o sistema.

**Ator:** Administrador.

**Fluxo Normal de Eventos:-** para atender os possíveis casos de uso foram necessários os cenários que serão demonstrados logo abaixo:

**1) Incluir:** O usuário que efetuará o cadastro dos professores poderá, incluir um novo cadastro de professor no sistema.

**2) Alterar:** O usuário pode fazer uma busca filtrada por nome do professor, resultando em uma tela que apresentará os dados atuais do mesmo. Possibilitando em seguida a alteração do dado desejado.

**3) Desativar:** O usuário pode fazer uma busca filtrada por nome, resultando em uma tela que apresentará os dados atuais do professor, possibilitando em seguida a desativação do mesmo do sistema.

**4) Consultar:** Nesta área o usuário logado como Administrador pode realizar uma busca filtrada de todos os professores registrados na base de dados.

**Fluxo Alternativo:** para o funcionamento de cadastro de professor:

Se houver algum dado que não foi processado pelo sistema este automaticamente emitirá uma mensagem informando o erro.

Se na verificação de consultar professor não for encontrado nenhum utilizando os dados informados ao sistema deverá emitir uma mensagem de que não

existe aquele professor cadastrado.

Se houver algum erro na exclusão dos dados, emitir uma mensagem informando que os dados não foram excluídos.

#### **7.3.4 Caso de Uso: Mantém Usuário do Sistema**

**Descrição:** Caso de uso que permite ao usuário, desde que logado corretamente, a manutenção dos usuários que interagem com o sistema.

**Ator:** Administrador.

**Fluxo Normal Eventos:-** para atender os possíveis casos de uso foram necessários os cenários que serão demonstrados logo abaixo:

**1) Incluir:** O usuário pode atuar no sistema como administradores de outros usuários, possibilitando realizar a inserção de uma conta de login para um novo usuário.

**2) Alterar:** O usuário pode fazer uma busca filtrada por nome, resultando em uma tela que apresentará os dados atuais do usuário. Possibilitando em seguida a alteração do dado.

**3) Excluir:** O usuário pode fazer uma busca filtrada por nome, resultando em uma tela que apresentará os dados atuais do usuário, possibilitando em seguida a desativação do usuário do sistema.

**4) Consultar:** Nesta área usuário logado como Administrador pode realizar uma busca filtrada de todos os usuários registrados na base de dados.

**Fluxo Alternativo:** para o funcionamento de Mantém usuário do sistema:

Se houver algum dado que não foi processado pelo sistema, automaticamente este emitirá uma mensagem informando o erro.

Se na verificação de consultar usuário do sistema não for encontrado nenhum usuário, utilizando os seguintes dados o sistema deverá emitir uma mensagem de que não existe aquele usuário.

Se houver algum erro na exclusão dos dados, emitir uma mensagem informando que os dados não foram excluídos.

### 7.3.5 Caso de Uso: Mantém Cadastro do Aluno

**Descrição:** este caso de uso tem como objetivo cadastrar os alunos. O caso de uso inicia-se quando o funcionário escolhe a opção cadastro de aluno.

**Atores:** Secretária e Administrador.

**Fluxo Normal de eventos:** para atender os possíveis casos de uso foram necessários os cenários que serão demonstrados logo abaixo:

**1) Incluir:** O ator escolhe a opção cadastro de aluno, em seguida informa os dados do aluno. O sistema irá armazenar os dados.

**2) Alterar:** O ator escolhe o dado do aluno para fazer a alteração, em seguida o sistema irá verificar a existência do dado do aluno no SGBD (Sistema de Gerenciamento de Dados), sendo disponibilizado os dados do aluno o ator informará ao sistema a alteração desejada. E em seguida o sistema irá armazenar os dados.

**3) Excluir:** O ator escolhe o dado do aluno que deseja excluir, o sistema irá verificar a existência, caso seja verdadeira então o sistema excluirá o dado do aluno;

**4) Consultar:** O ator escolhe a opção consultar aluno, o sistema irá verificar a existência, assim que possível os dados do aluno estarão disponíveis para

o Ator;

**5) Salvar:** O ator entrará com os dados do aluno no SGBD, e em seguida o sistema armazenará;

**Fluxo Alternativo:** para o funcionamento do cadastro dos alunos:

Se houver algum dado que não foi processado pelo sistema então automaticamente este emitirá uma mensagem informando o erro.

Se na verificação de consultar aluno não for encontrado nenhum aluno, o sistema deverá emitir uma mensagem de que não existe aquele aluno cadastrado.

### 7.3.6 Caso de Uso: Mantém Cadastro de Disciplina

**Descrição:** A secretária e o administrador poderão cadastrar as disciplinas relacionadas a cada turma.

**Atores:** Secretária e Administrador.

**Fluxo normal de eventos:** para atender os possíveis casos de uso foram necessários os cenários que serão demonstrados logo abaixo.

**1) Incluir:** O ator escolhe a opção cadastro de disciplina, é informa os dados da mesma. Em seguida o sistema armazena os dados no SGBD.

**2) Alterar:** O ator escolhe a disciplina para fazer a alteração, e em seguida o sistema verifica a existência, assim que o SGBD disponibilizar os dados da disciplina o ator irá informar ao sistema a alteração desejada, sendo possível o sistema irá armazenar os dados;

**3) Excluir:** O ator escolhe a Disciplina que deseja excluir, o sistema irá verificar a existência e assim que for disponibilizado o sistema irá excluir a disciplina do SGBD;

**4) Consultar:** O ator escolhe a opção consultar disciplina, o sistema irá verificar a existência da disciplina no SGBD, e em seguida o mesmo apresenta os dados da disciplina para a secretária.

**5) Salvar:** O ator entrará com os dados da disciplina, e o sistema irá armazenar os dados da disciplina no SGBD.

**Fluxos Alternativos:** para o funcionamento de cadastro de disciplina:

Se houver algum dado que não foi processado pelo sistema, automaticamente este emitirá uma mensagem informando o erro.

Se na verificação de consultar disciplina não for encontrada nenhuma, o sistema deverá emitir uma mensagem de que não existe aquela disciplina cadastrada.

Se houver algum erro na exclusão dos dados, emitir uma mensagem informando que os dados não foram excluídos.

### **7.3.7 Caso de Uso: Mantém Cadastro de Frequência**

**Descrição:** este caso de uso tem como objetivo fazer cadastramento de frequência do aluno. O caso de uso inicia-se quando a Secretária escolhe a opção cadastrar frequência.

**Atores:** Secretária e Administrador.

**Fluxo Normal de eventos:** para atender os possíveis casos de uso foram necessários os cenários que serão demonstrados logo abaixo:-

**1) Incluir:** O ator escolhe a opção cadastro de frequência e irá informar os dados do aluno, o sistema irá armazenar estes dados no SGBD.

**2) Alterar:** O ator escolhe o dado do aluno para fazer a alteração, o sistema irá verificar a existência, assim que possível o ator informará ao sistema a alteração desejada em seguida estes dados serão armazenados no SGBD;

**3) Excluir:** O ator escolhe o dado do aluno que deseja excluir, o sistema irá verificar a existência, e em seguida o dado do aluno será excluído do SGBD;

**Fluxos Alternativos:** para o funcionamento de cadastro de frequência:

Se houver algum dado que não foi processado pelo sistema, automaticamente este emitirá uma mensagem informando o erro.

Se houver algum erro na exclusão dos dados, emitir uma mensagem informando que os dados não foram excluídos.

### 7.3.8 Caso de Uso: Mantém Cadastro de Nota

**Descrição:** este caso de uso tem como objetivo fazer o cadastro das notas dos alunos de acordo com a disciplina. O caso de uso inicia-se quando a Secretária escolhe a opção cadastrar nota.

**Atores:** Secretária e Administrador.

**Fluxo Normal de eventos:** para atender os possíveis casos de uso



foram necessários os cenários que serão demonstrados logo abaixo:-

**1) Incluir:** O ator escolhe a opção cadastro de nota, este deverá informar os dados do aluno, em seguida o sistema irá armazenar os dados no SGBD.

**2) Alterar:** O ator escolhe o dado do aluno para fazer a alteração, o sistema verifica a existência do dado do aluno no SGBD, em seguida o sistema é informado sobre a alteração desejada, após ter concluído a alteração o sistema irá armazenar o dado no SGBD;

**3) Excluir:** O ator escolhe o dado do aluno que deseja excluir, o sistema irá verificar a sua existência, em seguida o dado do aluno será excluído do SGBD;

**Fluxos Alternativos:** para o funcionamento de cadastramento de nota:

Se houver algum dado que não foi processado pelo sistema, automaticamente este emitirá uma mensagem informando o erro.

Se houver algum erro na exclusão dos dados, emitir uma mensagem informando que os dados não foram excluídos.

### **7.3.9 Caso de Uso: Mantêm Cadastro de Turma**

**Descrição:** A secretária e o administrador poderão cadastrar uma nova turma para uma determinada série do sistema.

**Atores:** Secretária e Administrador.

**Fluxo Normal de eventos:** para atender os possíveis casos de uso

foram necessários os cenários que serão demonstrados logo abaixo:

**1) Incluir:** O ator escolhe a opção cadastro de Turma, informa os dados da Turma para o SGBD e em seguida o sistema irá incluir os dados no SGBD.

**2) Alterar:** O ator escolhe o cadastro de turma para fazer a alteração, o sistema verifica a existência da Turma em seguida o ator informa ao sistema a alteração desejada e o mesmo armazena os dados no SGBD;

**3) Excluir:** O ator escolhe a Turma que deseja excluir, o sistema irá verificar a existência da Turma no SGBD e em seguida será excluído a turma desejada;

**Fluxos Alternativos:** para o funcionamento de Cadastro de Turma:

Se houver algum dado que não foi processado pelo sistema, automaticamente este emitirá uma mensagem informando o erro.

Se na verificação de consultar turma não for encontrada nenhuma turma, o sistema deverá emitir uma mensagem de que não existe aquela turma cadastrada.

Se houver algum erro na exclusão dos dados, emitir uma mensagem informando que os dados não foram excluídos.

### **7.3.10 Caso de Uso: Realiza Matrícula.**

**Descrição:** este caso de uso tem como objetivo fazer o cadastro dos alunos. O caso de uso inicia-se quando o funcionário escolhe a opção Realiza Matrícula.

**Atores:** Secretária e Administrador.

**Fluxo Normal de eventos:** para atender os possíveis casos de uso foram necessários os cenários que serão demonstrados logo abaixo:

**1) Incluir:** O ator escolhe a opção Incluir Matrícula, e informa os dados do aluno, o sistema irá armazenar os dados no SGBD.

**2) Excluir:** O ator escolhe o dado do aluno que deseja excluir e em seguida o sistema verifica a existência do dado do aluno no SGBD, se possível o sistema exclui o dado do SGBD.

**Fluxos Alternativos:** para o funcionamento de matrícula:

Se houver algum dado que não foi processado pelo sistema, automaticamente este emitirá uma mensagem informando o erro.

Se houver algum erro na exclusão dos dados, emitir uma mensagem informando que os dados não foram excluídos.

### 7.3.11 Caso de Uso: Mantém Cadastro de Avaliação

**Descrição:** este caso de uso tem como objetivo cadastrar as avaliações dos alunos.

**Atores:** Administrador e Secretária.

**Fluxo Normal de eventos:** para atender os possíveis casos de uso foram necessários os cenários que serão demonstrados logo abaixo:-

**1) Incluir:** O ator escolhe a opção cadastro de avaliação, informa os dados do aluno e em seguida o sistema irá armazenar os dados no SGBD.

**2) Alterar:** O ator escolhe o cadastro de avaliação para fazer a

alteração, o sistema irá verificar a existência da avaliação no SGBD e em seguida o sistema será informado sobre a alteração desejada e o mesmo irá armazenar os dados no SGBD;

**3) Excluir:** O ator escolhe a Avaliação que deseja excluir, o sistema verifica a existência da Avaliação no SGBD e em seguida o sistema irá excluir a Avaliação desejada do SGBD;

**Fluxos Alternativos:** para o funcionamento de Cadastro de Avaliação.

Se houver algum dado que não foi processado pelo sistema, automaticamente este emitirá uma mensagem informando o erro.

Se na verificação de consultar avaliação não for encontrada nenhuma avaliação, o sistema deverá emitir uma mensagem de que não existe aquela avaliação cadastrada.

Se houver algum erro na exclusão dos dados, emitir uma mensagem informando que os dados não foram excluídos.

### 7.3.12 Caso de Uso: Relatório Parcial

**Descrição:** este caso de uso tem como objetivo demonstrar o andamento do aluno por Bimestre.

**Atores:** Administrador, Secretária e Professor.

**Fluxo Normal de eventos:** para atender este caso de uso foi necessário o cenário que será demonstrado logo abaixo:-

**1) Consultar:** O ator escolhe a opção consultar aluno, o sistema

verifica a existência no SGBD, e em seguida o sistema irá apresentar o dado do aluno para o Ator;

**Fluxos Alternativos:** para o funcionamento de consulta relatório parcial do aluno e necessário que:

Se na verificação de dados do aluno não for encontrado nenhum aluno, o sistema deverá emitir uma mensagem de que não existe aquele aluno cadastrado.

### 7.3.13 Caso de Uso: Relatório Final

**Descrição:** este caso de uso tem como objetivo demonstrar o Andamento do aluno durante os quatro Bimestre.

**Atores:** Administrador, Secretária e Professor.

**Fluxo Normal de eventos:** para atender este caso de uso foi necessário o cenário que será demonstrado logo abaixo:-

**1) Consultar:** O ator escolhe a opção consultar aluno, o sistema verifica a existência no SGBD, em seguida o sistema apresenta os dados do aluno para o Ator;

**Fluxos Alternativos:** para o funcionamento de consulta relatório final do aluno é necessário que:

Se na verificação de consulta de dados do aluno, não for encontrado o aluno desejado, o sistema deverá emitir uma mensagem de que não existe aquele aluno cadastrado.

## 7.4 PLANO DE ITERAÇÃO

### 7.4.1 Introdução

Este plano de iteração de elaboração terá como intuito analisar e concluir os principais requisitos e casos de uso do sistema, para que possa ser desenvolvido o Projeto Piloto de arquitetura, assim poderá ser testado a viabilidade e o desempenho da arquitetura. Durante esta iteração, o *design* do sistema e o plano de execução do projeto serão refinados.

**Tabela 9 - Referente ao Plano de Iteração**

Marco	Data
Início da Iteração	01/10/2006
Fim da Iteração	31/10/2006

### 7.4.2 Objetivo

O objetivo desta iteração é desenvolver um projeto piloto de arquitetura, e fazer a modelagem do sistema e validar o Sistema Acadêmico para a Escola Algodão Doce – E.I.E.F.

**Tabela 10 - Nomeação das Tarefas e Nomeados**

Tarefa	Nomeado
Analisar os requisitos e desenvolver a arquitetura do Projeto Piloto.	Maria Virgínia Andreotti Mendes Bordin

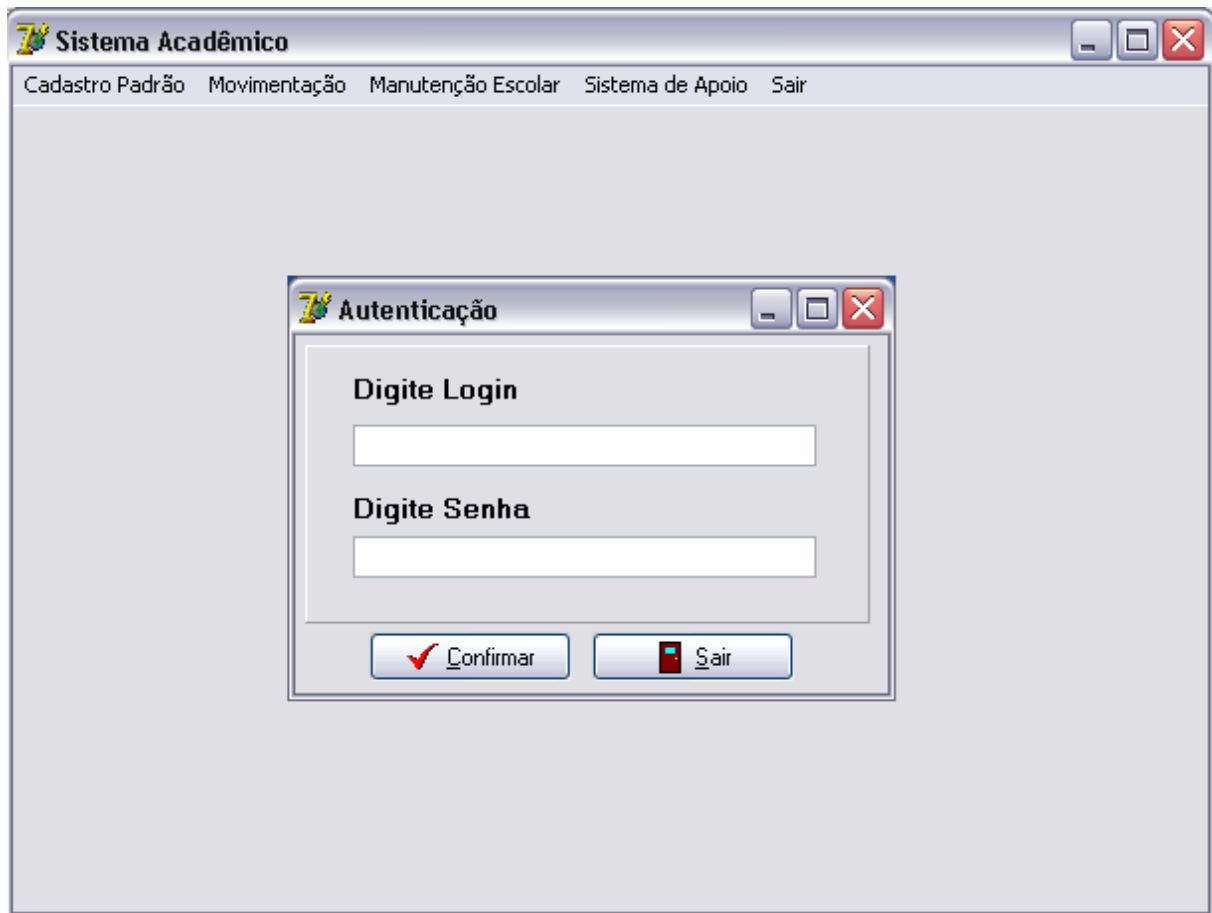
## **7.5 IMPLEMENTAÇÃO DO SISTEMA ACADÊMICO**

Como primeiro passo para o desenvolvimento do Projeto Piloto, constatou-se os principais problemas e dificuldades citadas anteriormente neste trabalho, cujas informações obtidas tiveram como importância a definição dos aspectos do Projeto.

Levando-se em conta os processos identificados e representados através da UML e BUP conforme os apêndices A,B,C,D,E,F e a seção 7.1 respectivamente, foi elaborado também o Modelo de Entidade e Relacionamento (MER) que foi implantado no banco de dados do projeto piloto. Posteriormente foi iniciado o processo de desenvolvimento do projeto piloto que a seguir é descrito através das telas que são apresentadas para o usuário.

### **7.5.1 Apresentação do Projeto Piloto Desenvolvido**

A seguir, nesta seção, serão mostradas algumas das funcionalidades do Projeto Piloto para a Escola Algodão Doce, que poderão ser observadas a partir do *layout* das telas do sistema.

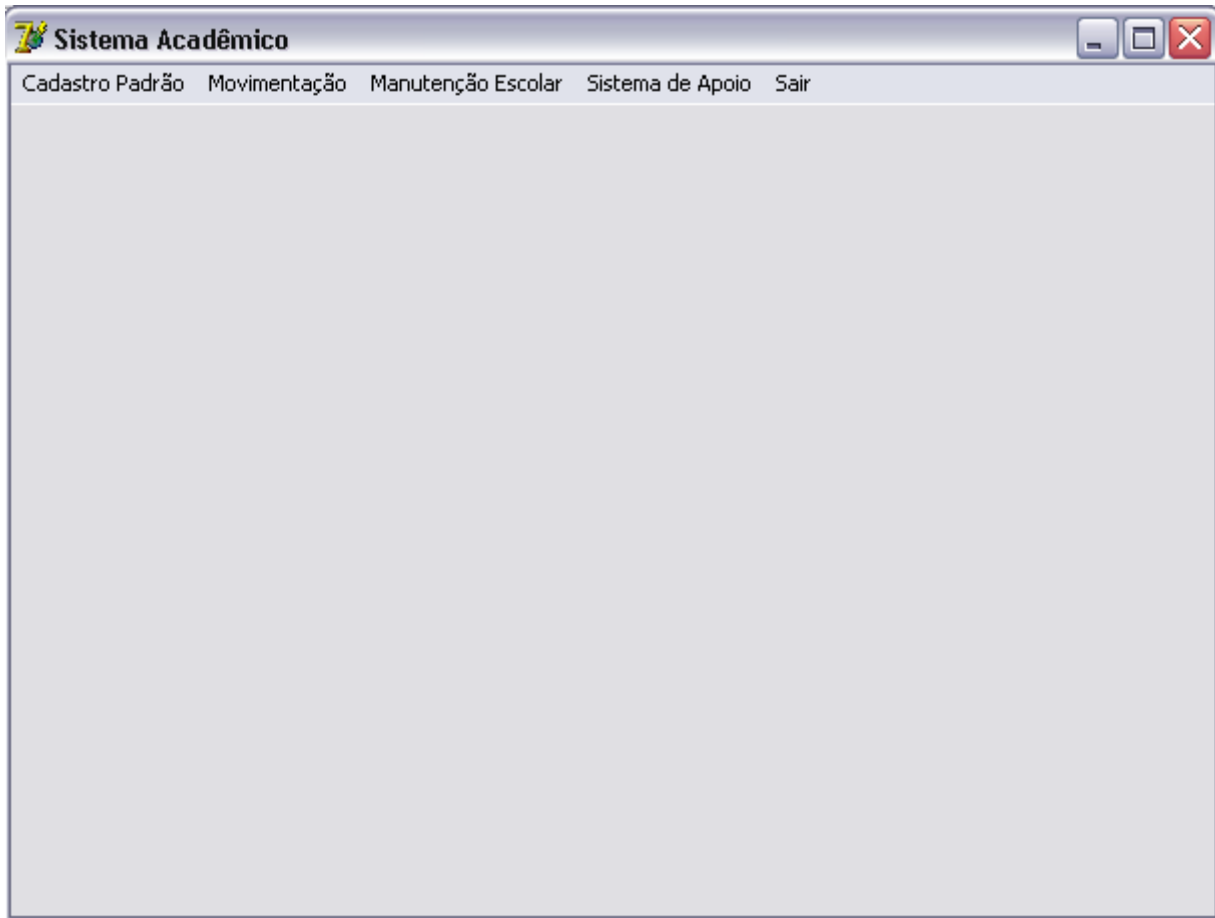


**Figura 7: Tela de Entrada do Sistema.**

Para que o usuário possa efetivar a sua entrada no sistema com sucesso, são disponibilizadas as seguintes opções o botão confirmar e sair. Caso o usuário escolha o botão confirmar, o login e senha serão validados e se o usuário for um administrador do sistema este será identificado.

Ao entrar no sistema a interface representada na figura 8 que corresponde com a tela principal do sistema será apresentada.





**Figura 8: Tela Inicial do Sistema Acadêmico.**

A seguir será demonstradas as opções da tela inicial do sistema:

- A primeira opção é o cadastro padrão, este menu está dividido em:
  - Registro do Aluno que fornece um submenu com as seguintes informações Cadastro de Aluno, Cadastro de Matrícula.
  - Registro de Usuário que também fornece um submenu com as seguintes informações Cadastro de Funcionário e Cadastro de Professor.
  - Cadastro da Instituição.
- A segunda opção é a movimentação, este menu está dividido em:

- Cadastro de Disciplina
- Cadastro de Turma
- A terceira opção é a Manutenção Escolar, este menu está dividido em:
  - Registro Escolar que fornece um submenu com as seguintes informações Cadastro de Avaliação, Cadastro de Frequência, Cadastro de Nota.
  - Cadastro Disciplina/Turma
- A quarta opção é a Sistema de Apoio, este menu está dividido em:
  - Cadastro de Usuário.
- A quinta opção é o comando Sair este irá finalizar a aplicação.

#### **7.5.1.1 Entrando no Sistema**

Abaixo serão demonstradas algumas interfaces do projeto piloto desenvolvido com o intuito de demonstrar as funcionalidades existentes em cada uma delas.

Para que o *Layout* do Cadastro de Usuário esteja disponível, é necessário que o Administrador do sistema esteja conectado. Assim poderá ser efetivado um novo cadastro de usuário para o sistema. Como será demonstrado na figura 9.

The image shows a software window titled "Usuário" with a standard Windows-style title bar. The window contains the following elements:

- Form fields: "Login:" (text input), "Codigo Funcionário:" (dropdown menu), "Senha" (password input), "Confirmar Senha:" (password input), and "ADM" (checkbox).
- Table: A table with three columns: "LOGIN", "FUNCIONÁRIO", and "ADM". The table is currently empty.
- Buttons: A row of three buttons: "Novo" (with a plus icon), "Excluir" (with a trash icon), and "Editar" (with a pencil icon). A second row of three buttons: "Salvar" (with a floppy disk icon), "Cancelar" (with a red X icon), and "Sair" (with a red X icon).
- Search: A text input field followed by a "Pesquisar" button.

**Figura 9: Tela Cadastro de Usuário.**

Após a habilitação do cadastro de usuário o administrador efetivará o cadastramento do novo usuário do sistema. Sendo este um funcionário já cadastrado no sistema. E para a validação do cadastro deverá ser utilizado as seguintes opções que estarão disponíveis na tela de cadastro.

Na figura 10 é apresentado o Cadastro de Alunos e são demonstradas todas as informações a serem trabalhadas.

**Figura 10: Interface Cadastro de Aluno.**

Na tela de cadastro de alunos os dados estão divididos por seções sendo disponibilizadas conforme a sua necessidade, em primeiro lugar quando o funcionário da instituição for efetuar o cadastro do aluno será disponibilizado primeiramente os dados pessoais onde contêm os dados relevantes da vida pessoal de cada um como nome, endereço, telefone.

A seguir é demonstrada a interface de Cadastro de Freqüência representado na figura 11:

The image shows a software window titled "Cadastro de Frequência". The window contains a form with the following fields:

- Nome:** A text input field with a dropdown arrow on the right.
- Disciplina:** A dropdown menu.
- Turma:** A dropdown menu.
- Data:** A text input field.
- Presença:** A text input field.

Below the form is a table with the following columns:

ALUNO	TURMA	DISCIPLINA	DATA	PRESENCIA

At the bottom of the window, there are several buttons:

- Novo:** Represented by a document icon.
- Excluir:** Represented by a trash can icon.
- Editar:** Represented by a pencil icon.
- Salvar:** Represented by a floppy disk icon.
- Cancelar:** Represented by a red circle with a slash icon.
- Sair:** Represented by a red square icon.
- Pesquisar:** A button with a magnifying glass icon.

There is also a search input field located below the buttons.

**Figura 11: Interface Cadastro de Frequência.**

Para a finalização desta especificação serão demonstradas as funcionalidades de cada botão apresentado nas telas citadas anteriormente. O botão Novo é responsável pela inserção de um novo cadastro. Já o botão Excluir caso queira deletar algum dado e só selecionar. E caso algum cadastro necessite de alteração selecione o botão Editar. Após os dados terem sido modificados ou alterados o usuário deverá Salvar os dados inseridos. Se caso o usuário não deseja utilizar nenhuma opção de inserção ou edição este deverá Cancelar a operação. Estará disponível também o comando de Imprimir Relatório para algumas interfaces nas quais estarão disponíveis os dados impressos do cadastro solicitado. Ou, perante a necessidade do usuário, o mesmo poderá apenas Pesquisar uma determinada informação.

## 8 CONSIDERAÇÕES FINAIS

Através deste Projeto pôde-se visualizar os problemas que os funcionários da Escola Algodão Doce enfrentam com o prazo da entrega dos documentos dos alunos. Este projeto teve como meta solucionar estes problemas utilizando o processo do BUP e a linguagem UML.

A utilização destas notações foram de grande importância, não somente para o desenvolvimento deste trabalho, mas também para ampliação do conhecimento sobre eles.

Durante a construção deste trabalho verificou-se a melhoria dos processos utilizados na Escola Algodão Doce fazendo com que as tarefas se tornassem mais rápidas e eficientes, e a aplicação demonstrou uma possibilidade de transformar os registros e processamentos das informações em atividades menos árduas e, conseqüentemente, mais produtivas.

De acordo com os artefatos citados no decorrer do texto foi possível perceber que o tempo de execução de cada tarefa varia conforme a estimativa de execução do projeto.

A finalização do sistema terá como intuito a sua melhoria de modo a aperfeiçoá-lo de acordo com as necessidades dos usuários e implementando alguns avanços em relação às interfaces, sendo estas duas propostas para trabalhos futuros.

## REFERÊNCIAS

ALMEIDA, Alexandre; DAROLT, Reginaldo. **Pesquisa e Desenvolvimento em UML**. Araranguá, 2001. Monografia (Especialização em Ciência da Computação) – Universidade do Sul de Santa Catarina, Santa Catarina.

**Basic Unified Process**, Version 0.1. (c) Copyright Eclipse contributors and others 2004, 2006. All Rights Reserved.

BALDUINO, Ricardo. **Basic Unified Process: A Process for Small and Agile Projects**, [20-?]. Disponível em: <<http://www.eclipse.org/proposals/beacon/>> . Acesso em 12/09/2006 às 23:17 hrs.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML Guia do Usuário**. Rio de Janeiro: Editora Elseiver, 2000.

CRUZ, Antonio Anastácio da. **Evolução Tecnológica em Sistemas de Informação**, [ca. 2004]. Disponível em: < <http://www.bibli.fae.unicamp.br/etd/revlit01v5n1.pdf>> . Acesso em 15/05/2006 às 13:00 hrs.

KRUCHTEN, Philippe. **Introdução ao RUP Rational Unified Process**. Rio de Janeiro: Editora Ciência Moderna, 2003.

PFLEEGER, Shari Lawrence. **Engenharia de Software: teoria e prática**. 2<sup>o</sup> Edição. São Paulo: Editora Pearson. Prentice Hall, 2004.

PRESSMAN, Roger S. **Engenharia de Software**. 5<sup>o</sup> Edição. Rio de Janeiro: Editora McGraw. Hill do Brasil, 2002.

## BIBLIOGRAFIA

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 10520**: apresentação de citações em documentos. Rio de Janeiro, ago. 2002b.

\_\_\_\_\_. **NBR 14724**: Informação e Documentação – Trabalhos Acadêmicos - Apresentação. Rio de Janeiro, ago. 2002c.

\_\_\_\_\_. **NBR 6023**: Informação e Documentação – Referências - Elaboração. Rio de Janeiro, ago. 2002a.

DIAS, Adilson de Souza. **Delphi Bancos de Dados com SQL**. Rio de Janeiro: Editora Ciência Moderna Ltda.,2000.

GANE, Chris; SARSON, Trish. **Análise Estruturada de Sistema**. Rio de Janeiro: Editora Livros Técnicos e Científicos S. A, 2002.

MANZANO, José Augusto N.G.; MENDES, Sandro Vicca Mendes. **Estudo Dirigido Delphi 7**. São Paulo: Editora Érica Ltda., 2004.

MULLER, Mary Stela; CORNELSEN, Julce Mary. **Teses, Dissertações e Monografias**. 5º Edição. Londrina: Editora da Universidade Estadual de Londrina, 2003.

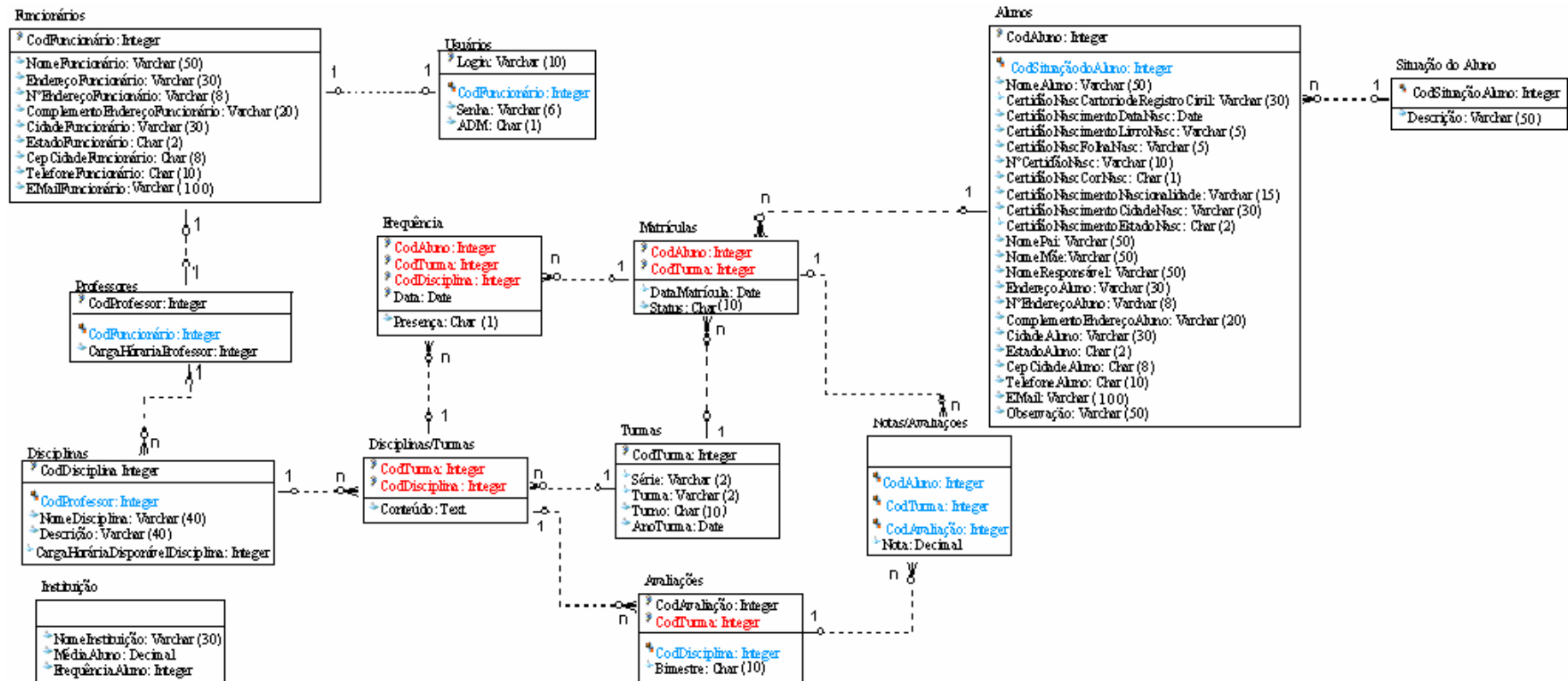
OBERG, Roger; PROBASCO, Leslee; ERICSSON, Maria. **Applying Requirements Management with Use Cases**. Rational Software White Paper. Rational U.K., Rational Canada, TP505, 2000.

SPENCE, Ian; PROBASCO, Leslee. **Traceability Strategies for Managing Requirements with Use Cases**. Rational Software White Paper. Rational U.K., Rational Canada, TP166, 2000.

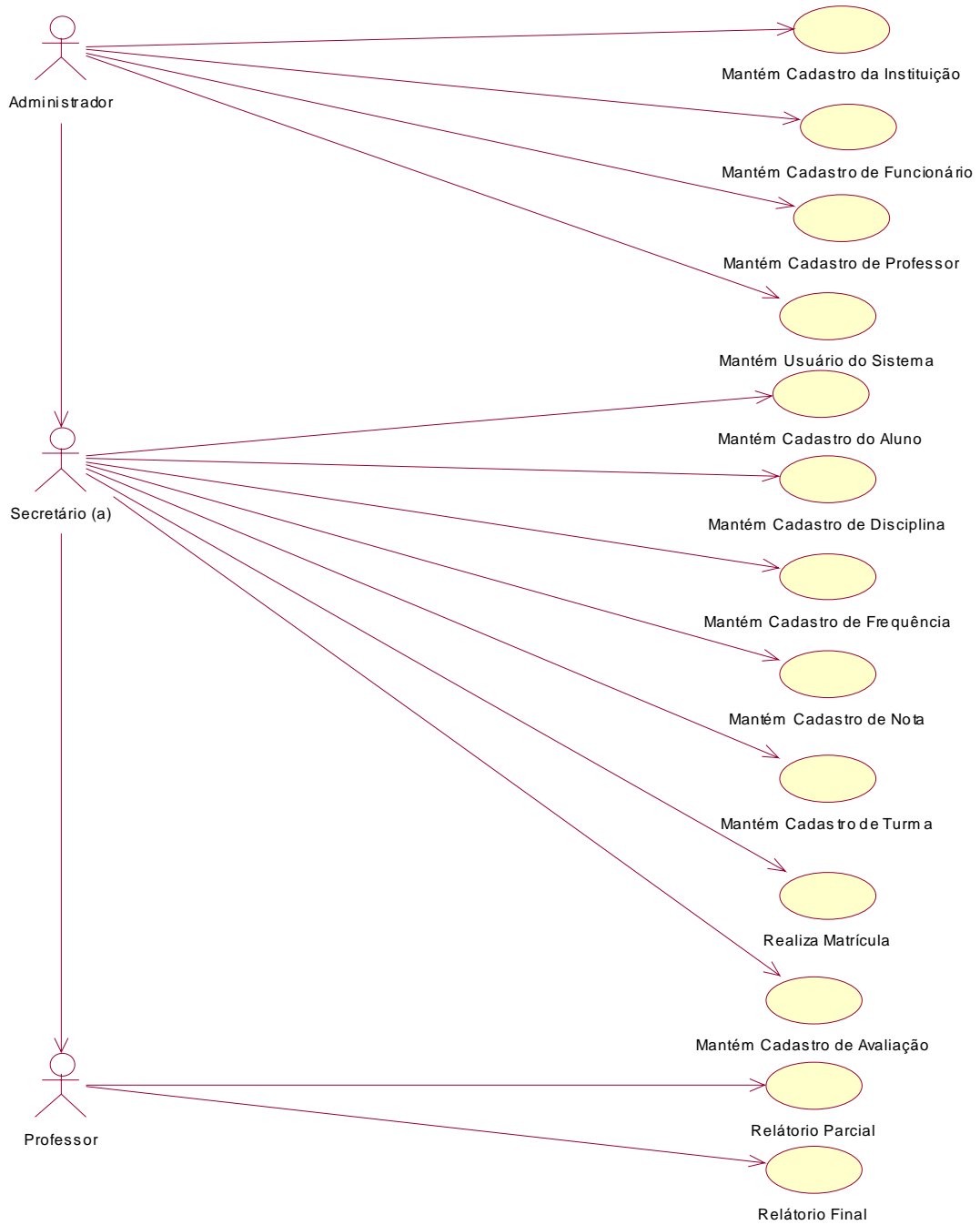


## APÊNDICES

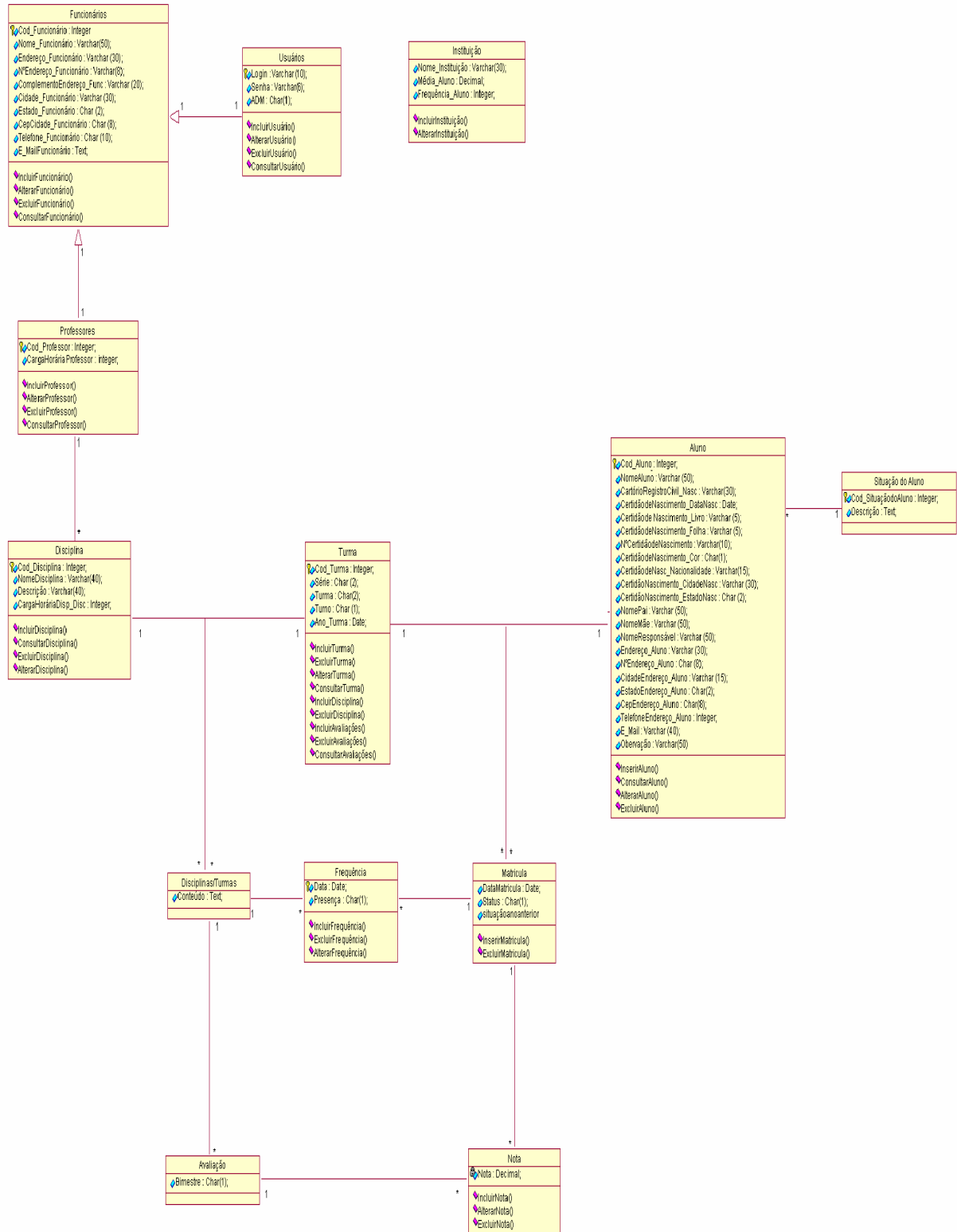
### APÊNDICE A – MODELO ENTIDADE RELACIONAMENTO



### APÊNDICE B – DIAGRAMA DE CASO DE USO



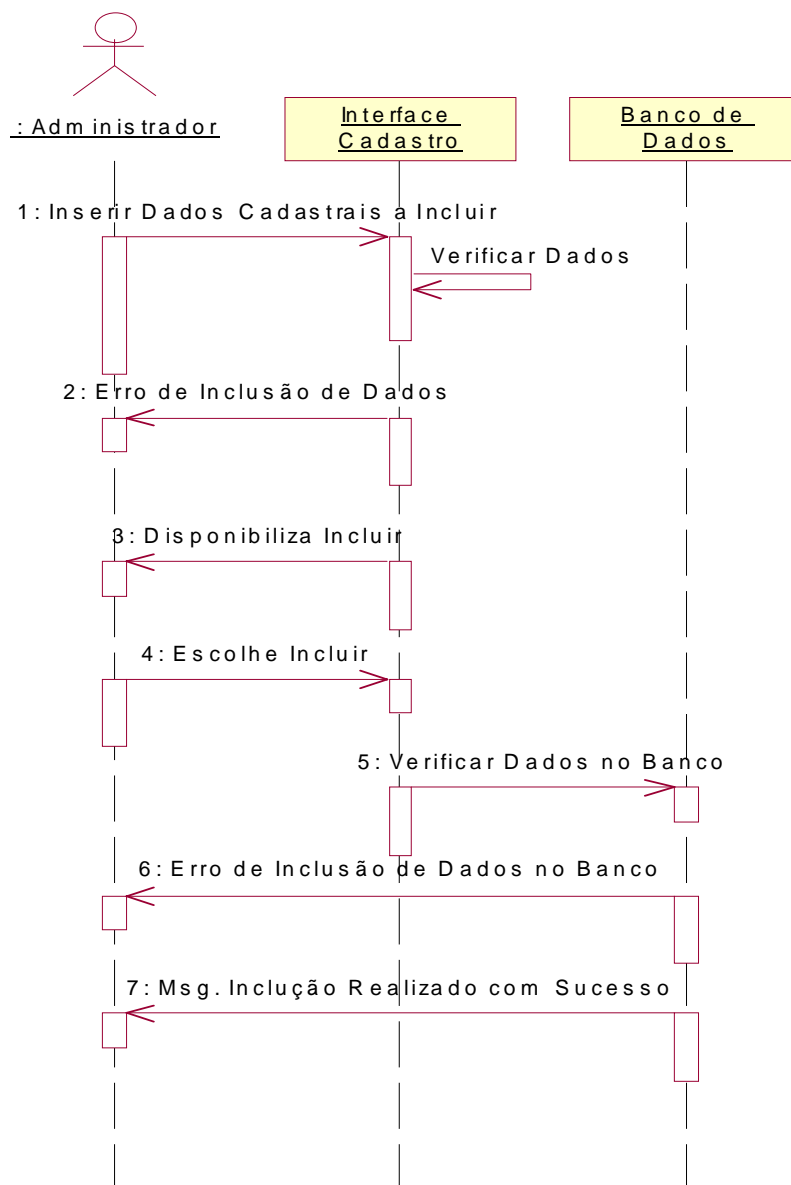
## APÊNDICE C - DIAGRAMA DE CLASSE



## APÊNDICE D - DIAGRAMA DE SEQÜÊNCIA

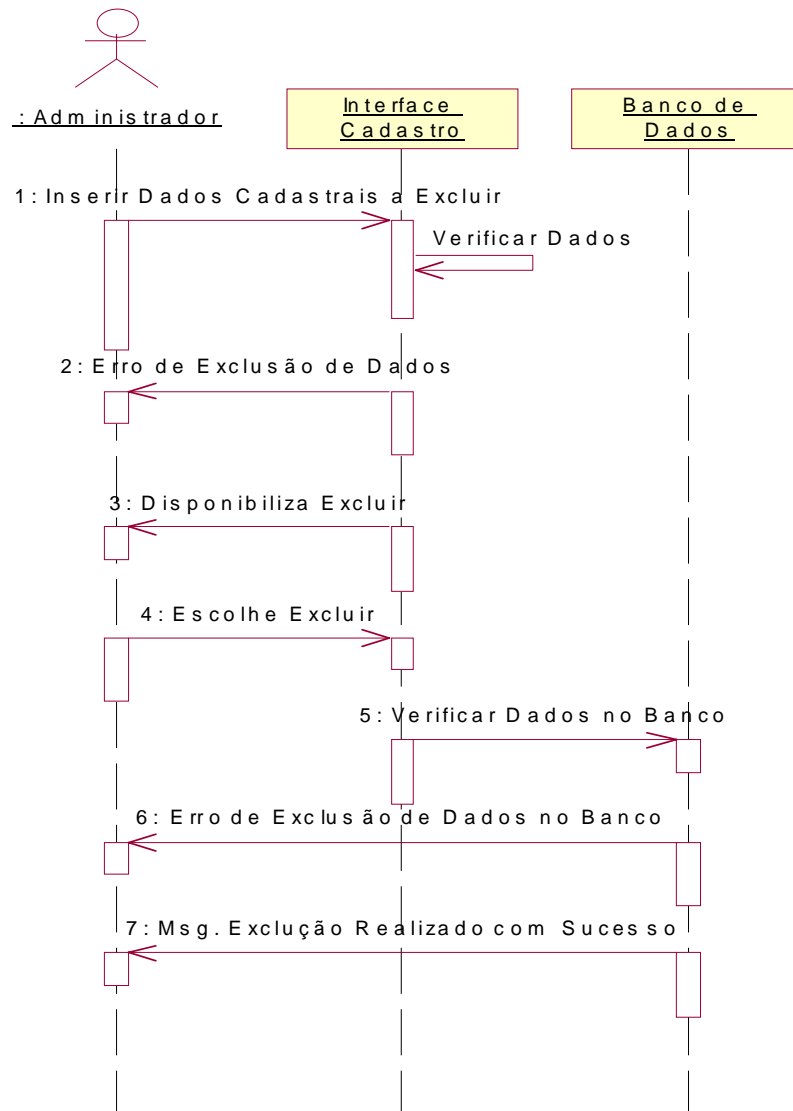
❖ **Caso de Uso: Mantém Cadastros Instituição, Usuários, Funcionários, Professores, Disciplina, Turma, Aluno, Frequência, Matrícula e Nota.**

- **Cenário: Incluir**



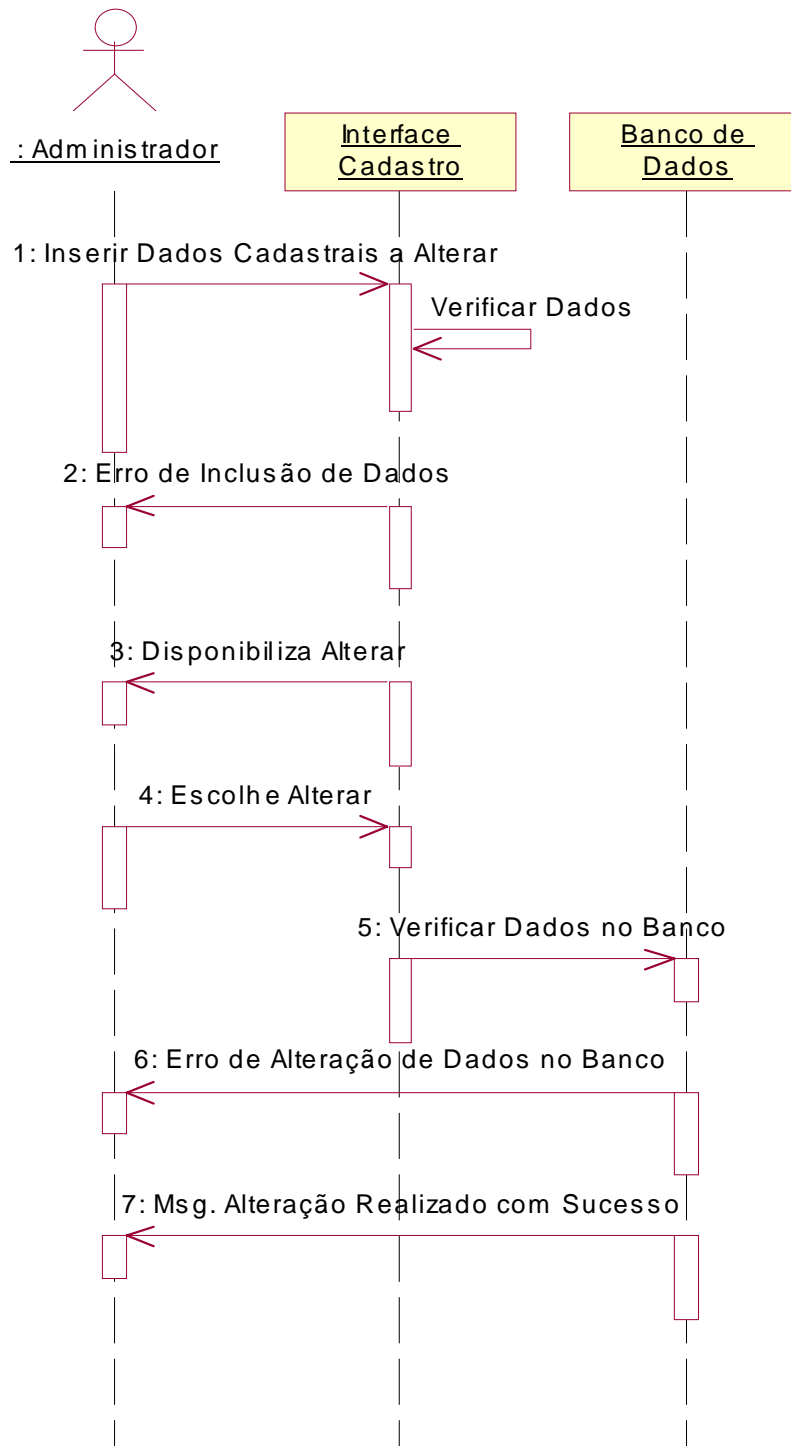
❖ **Caso de Uso: Mantém Cadastros Usuários, Funcionários, Professores, Disciplina, Turma, Aluno, Frequência, Matrícula e Nota.**

• **Cenário: Excluir**



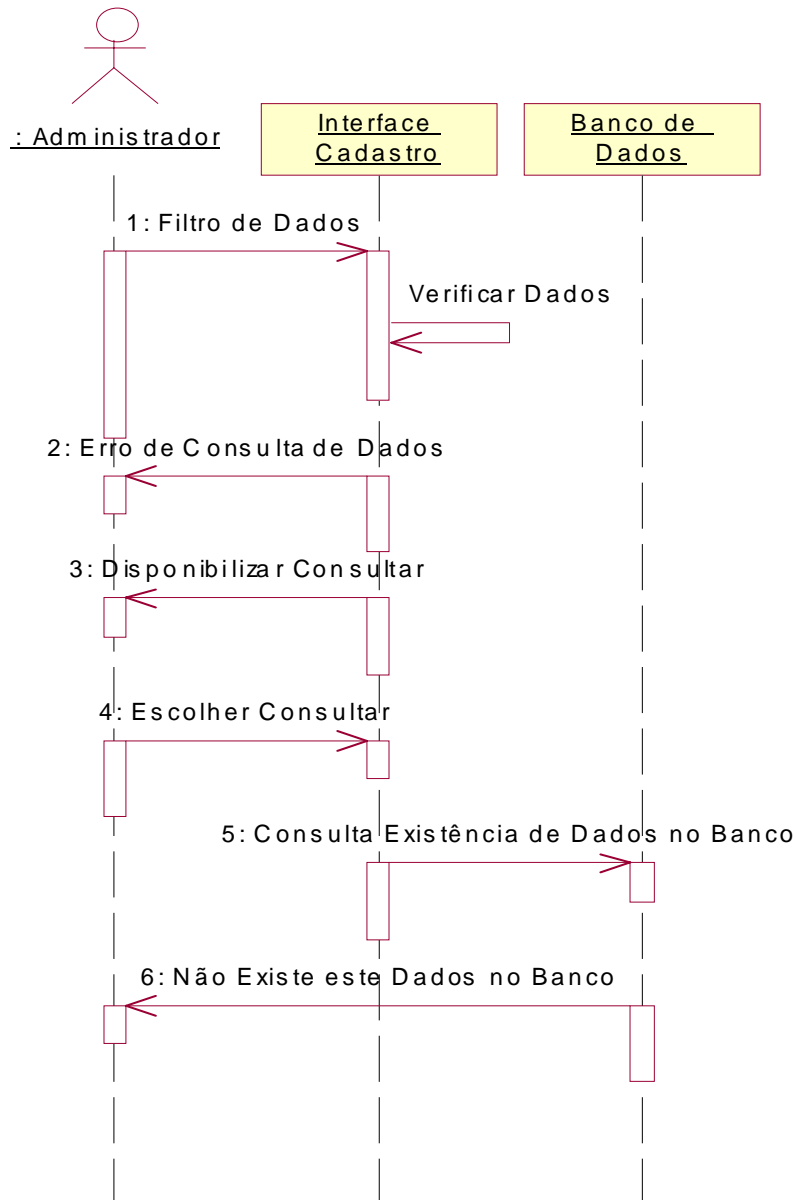
❖ **Caso de Uso: Mantém Cadastro Usuários, Funcionários, Professores, Disciplina, Turma, Aluno, Frequência e Nota.**

• **Cenário: Alterar**



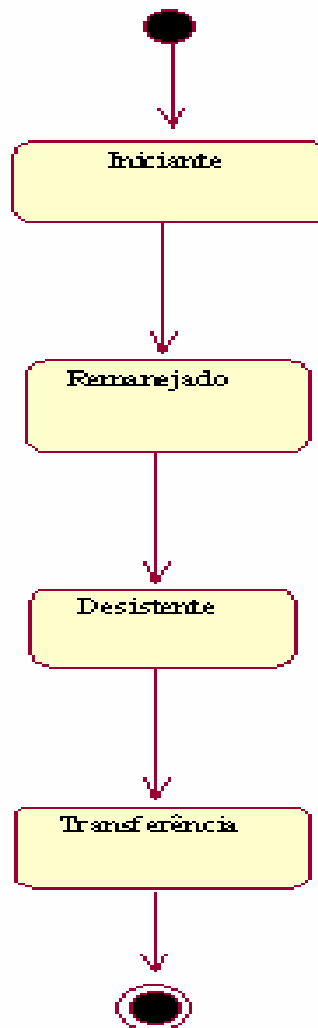
❖ **Caso de Uso: Mantém Cadastros Usuários, Funcionários, Professores, Disciplina, Turma e Aluno.**

• **Cenário: Consultar**



## APÊNDICE E - DIAGRAMA DE ESTADO

### ❖ Cenário: Situação do Aluno





## APÊNDICE F – *SCRIPTS* BANCO DE DADOS

```

SET NAMES NONE;
CREATE GENERATOR ALUNOSGEN;
CREATE TABLE CAD_ALUNOS (
  COD_ALUNO          INTEGER NOT NULL,
  COD_SITUACAODOALUNO  INTEGER NOT NULL,
  NOME_ALUNO         VARCHAR(50) CHARACTER SET ASCII NOT NULL,
  CARTORIODEREGISTRO_NASC VARCHAR(30),
  CERTIDAONASC_DATANASC  DATE NOT NULL,
  CERTIDAONASC_LIVRONASC VARCHAR(5),
  CERTIDAONASC_FOLHANASC VARCHAR(5),
  N_CERTIDNASCIMENTO   VARCHAR(10),
  CERTNASC_CORNASC     VARCHAR(10),
  CERTNASC_NASCIONALIDADE VARCHAR(15),
  CERTDNASC_CIDADENASC VARCHAR(30),
  CERTIDAONASC_ESTADONASC CHAR(2),
  NOME_PAI           VARCHAR(50) CHARACTER SET ASCII NOT NULL,
  NOME_MAE           VARCHAR(50) CHARACTER SET ASCII NOT NULL,
  NOME_RESPONSAVEL   VARCHAR(50) CHARACTER SET ASCII NOT NULL,
  ENDERECO_ALUNO     VARCHAR(30),
  N_ENDERECO_ALUNO   CHAR(8),
  CIDADEENDERECO_ALUNO VARCHAR(15),
  ESTADOENDERECO_ALUNO CHAR(2),
  CEPENDERECO_ALUNO  CHAR(8),
  TELEFONEENDERECO_ALUNO CHAR(10),
  COMPLEMENTO_ENDERECO VARCHAR(30),
  OBSERVACAO        VARCHAR(50),
  EMAIL              DMTEXTO /* DMTEXTO = VARCHAR(100) */);
ALTER TABLE CAD_ALUNOS ADD CONSTRAINT PK_CAD_ALUNOS PRIMARY KEY
(COD_ALUNO);
ALTER TABLE CAD_ALUNOS ADD CONSTRAINT FK_CAD_ALUNOS FOREIGN KEY
(COD_SITUACAODOALUNO) REFERENCES CAD_SITUACAODOALUNO
(COD_SITUACAODOALUNO);
SET TERM ^ ;
/* Trigger: ALUNOSTRG */
CREATE TRIGGER ALUNOSTRG FOR CAD_ALUNOS
ACTIVE BEFORE INSERT POSITION 0
AS
begin
  new.cod_aluno = gen_id( alunosgen, 1);
end
SET TERM ; ^

```

```

SET NAMES NONE;
CREATE GENERATOR CODAVALIACAOPEN;

CREATE TABLE CAD_AVALIACOES (
  COD_AVALIACAO INTEGER NOT NULL,
  COD_TURMA     INTEGER NOT NULL,
  COD_DISCIPLINA INTEGER NOT NULL,
  BIMESTRE     CHAR(10) CHARACTER SET ASCII NOT NULL);
/* Check constraints definition */
ALTER TABLE CAD_AVALIACOES ADD CONSTRAINT CHK_CAD_AVALIACOES check
(Bimestre in ('1º Bimestre', '2º Bimestre', '3º Bimestre', '4º Bimestre'));
ALTER TABLE CAD_AVALIACOES ADD CONSTRAINT PK_CAD_AVALIACOES
PRIMARY KEY (COD_AVALIACAO, COD_TURMA);
ALTER TABLE CAD_AVALIACOES ADD CONSTRAINT FK_CAD_AVALIACOES FOREIGN
KEY (COD_TURMA, COD_DISCIPLINA) REFERENCES CAD_DISCIPLINAS_TURMAS
(COD_TURMA, COD_DISCIPLINA);
SET TERM ^;
/* Trigger: CODAVALIACAOTRG */
CREATE TRIGGER CODAVALIACAOTRG FOR CAD_AVALIACOES
ACTIVE BEFORE INSERT POSITION 0
AS
begin
  new.cod_avaliacao = gen_id (codavaliacaopen,1);
end
^
SET TERM ; ^

```

```

SET NAMES NONE;
CREATE GENERATOR CODDISCIPLINAGEN;
CREATE TABLE CAD_DISCIPLINA (
  COD_DISCIPLINA     INTEGER NOT NULL,
  COD_PROFESSOR      INTEGER NOT NULL,
  NOME_DISCIPLINA    VARCHAR(40) CHARACTER SET ASCII NOT NULL,
  DESCRICAO          DMTEXTO NOT NULL /* DMTEXTO = VARCHAR(100) */,
  CARGA_HORAR_DISP_DISC INTEGER NOT NULL);
ALTER TABLE CAD_DISCIPLINA ADD CONSTRAINT PK_CAD_DISCIPLINA PRIMARY
KEY (COD_DISCIPLINA);
ALTER TABLE CAD_DISCIPLINA ADD CONSTRAINT FK_CAD_DISCIPLINA FOREIGN
KEY (COD_PROFESSOR) REFERENCES CAD_PROFESSORES
(COD_PROFESSORES);
SET TERM ^;
/* Trigger: CODDISCIPLINA */
CREATE TRIGGER CODDISCIPLINA FOR CAD_DISCIPLINA
ACTIVE BEFORE INSERT POSITION 0
AS
begin
  new.cod_disciplina = gen_id (coddisciplinagen,1);
end
^
SET TERM ; ^

```

```

SET NAMES NONE;
CREATE GENERATOR ALUNOSGEN;
CREATE TABLE CAD_ALUNOS (
  COD_ALUNO          INTEGER NOT NULL,
  COD_SITUACAODOALUNO  INTEGER NOT NULL,
  NOME_ALUNO         VARCHAR(50) CHARACTER SET ASCII NOT NULL,
  CARTORIODEREGISTRO_NASC VARCHAR(30),
  CERTIDAONASC_DATANASC DATE NOT NULL,
  CERTIDAONASC_LIVRONASC VARCHAR(5),
  CERTIDAONASC_FOLHANASC VARCHAR(5),
  N_CERTIDNASCIMENTO  VARCHAR(10),
  CERTNASC_CORNASC    VARCHAR(10),
  CERTNASC_NASCIONALIDADE VARCHAR(15),
  CERTDNASC_CIDADENASC VARCHAR(30),
  CERTIDAONASC_ESTADONASC CHAR(2),
  NOME_PAI           VARCHAR(50) CHARACTER SET ASCII NOT NULL,
  NOME_MAE           VARCHAR(50) CHARACTER SET ASCII NOT NULL,
  NOME_RESPONSAVEL   VARCHAR(50) CHARACTER SET ASCII NOT NULL,
  ENDERECO_ALUNO     VARCHAR(30),
  N_ENDERECO_ALUNO   CHAR(8),
  CIDADEENDERECO_ALUNO VARCHAR(15),
  ESTADOENDERECO_ALUNO CHAR(2),
  CEPENDERECO_ALUNO  CHAR(8),
  TELEFONEENDERECO_ALUNO CHAR(10),
  COMPLEMENTO_ENDERECO VARCHAR(30),
  OBSERVACAO         VARCHAR(50),
  EMAIL              DMTEXTO /* DMTEXTO = VARCHAR(100) */);
ALTER TABLE CAD_ALUNOS ADD CONSTRAINT PK_CAD_ALUNOS PRIMARY KEY
(COD_ALUNO);
ALTER TABLE CAD_ALUNOS ADD CONSTRAINT FK_CAD_ALUNOS FOREIGN KEY
(COD_SITUACAODOALUNO) REFERENCES CAD_SITUACAODOALUNO
(COD_SITUACAODOALUNO);
SET TERM ^;
/* Trigger: ALUNOSTRG */
CREATE TRIGGER ALUNOSTRG FOR CAD_ALUNOS
ACTIVE BEFORE INSERT POSITION 0
AS
begin
  new.cod_aluno = gen_id( alunosgen, 1);
end
SET TERM ; ^

```

```

SET NAMES NONE;
CREATE TABLE CAD_DISCIPLINAS_TURMAS (
  COD_TURMA          INTEGER NOT NULL,
  COD_DISCIPLINA     INTEGER NOT NULL,
  CONTEUDO           DMTEXTO /* DMTEXTO = VARCHAR(100) */);
ALTER TABLE CAD_DISCIPLINAS_TURMAS ADD CONSTRAINT
PK_CAD_DISCIPLINAS_TURMAS PRIMARY KEY (COD_TURMA, COD_DISCIPLINA);
ALTER TABLE CAD_DISCIPLINAS_TURMAS ADD CONSTRAINT
FK_CAD_DISCIPLINAS_TURMAS FOREIGN KEY (COD_DISCIPLINA) REFERENCES
CAD_DISCIPLINA (COD_DISCIPLINA);
ALTER TABLE CAD_DISCIPLINAS_TURMAS ADD CONSTRAINT
FK_CAD_DISCIPLINA_TURMAS FOREIGN KEY (COD_TURMA) REFERENCES
CAD_TURMA (COD_TURMA);

```

```

SET NAMES NONE;
CREATE TABLE CAD_FREQUENCIA (
  COD_ALUNO    INTEGER NOT NULL,
  COD_TURMA    INTEGER NOT NULL,
  COD_DISCIPLINA INTEGER NOT NULL,
  DATA        DATE NOT NULL,
  PRESENCA     CHAR(1) CHARACTER SET ASCII NOT NULL);
/* Check constraints definition */
ALTER TABLE CAD_FREQUENCIA ADD CONSTRAINT CHK_CAD_FREQUENCIA check
(Presenca in ('Sim', 'Nao'));
ALTER TABLE CAD_FREQUENCIA ADD CONSTRAINT PK_CAD_FREQUENCIA
PRIMARY KEY (COD_ALUNO, COD_TURMA, COD_DISCIPLINA, DATA);
ALTER TABLE CAD_FREQUENCIA ADD CONSTRAINT FK_CAD_FREQUENCIA
FOREIGN KEY (COD_ALUNO, COD_TURMA) REFERENCES CAD_MATRICULA
(COD_ALUNO, COD_TURMAS);
ALTER TABLE CAD_FREQUENCIA ADD CONSTRAINT FK_CAD_FREQUENCIA1
FOREIGN KEY (COD_TURMA, COD_DISCIPLINA) REFERENCES
CAD_DISCIPLINAS_TURMAS (COD_TURMA, COD_DISCIPLINA);

```

```

SET NAMES NONE;
CREATE GENERATOR CODFUNCIONARIOGEN;

CREATE TABLE CAD_FUNCIONARIOS (
  COD_FUNCIONARIO    INTEGER NOT NULL,
  NOME_FUNCIONARIO   VARCHAR(50) NOT NULL,
  ENDERECO_FUNCIONARIO VARCHAR(30),
  N_ENDERECO_FUNCIONARIO VARCHAR(8),
  COMPLEMENTOEND_FUNC VARCHAR(20),
  CIDADE_FUNCIONARIO  VARCHAR(30),
  ESTADO_FUNCIONARIO  CHAR(2),
  CEP_CIDADE_FUNCIONARIO CHAR(8),
  TELEFONE_FUNCIONARIO CHAR(10),
  EMAIL_FUNCIONARIO   DMTEXT /* DMTEXT = VARCHAR(100) */);
ALTER TABLE CAD_FUNCIONARIOS ADD CONSTRAINT PK_CAD_FUNCIONARIOS
PRIMARY KEY (COD_FUNCIONARIO);
SET TERM ^;
/* Trigger: CODFUNCIONARIO */
CREATE TRIGGER CODFUNCIONARIO FOR CAD_FUNCIONARIOS
ACTIVE BEFORE INSERT POSITION 0
AS
begin
  new.cod_funcionario = gen_id (codfuncionariogen,1);
end
^
SET TERM ; ^

```

```

SET NAMES NONE;
CREATE TABLE CAD_INSTITUICAO (
  NOME_INSTITUICAO VARCHAR(30) CHARACTER SET ASCII NOT NULL,
  MEDIA_ALUNO      DECIMAL(8,2) NOT NULL,
  FREQUENCIA_ALUNO INTEGER NOT NULL);

```

```

SET NAMES NONE;
CREATE TABLE CAD_MATRICULA (
  COD_ALUNO    INTEGER NOT NULL,
  COD_TURMAS    INTEGER NOT NULL,
  DATA_MATRICULA DATE NOT NULL,
  STATUS        CHAR(1) CHARACTER SET ASCII NOT NULL);
/* Check constraints definition */
ALTER TABLE CAD_MATRICULA ADD CONSTRAINT CHK_CAD_MATRICULA check

```

```
(Status in ('Aprovado', 'Reprovado', 'Cursando'));
ALTER TABLE CAD_MATRICULA ADD CONSTRAINT PK_CAD_MATRICULA PRIMARY
KEY (COD_ALUNO, COD_TURMAS);
ALTER TABLE CAD_MATRICULA ADD CONSTRAINT FK_CAD_MATRICULA FOREIGN
KEY (COD_ALUNO) REFERENCES CAD_ALUNOS (COD_ALUNO);
ALTER TABLE CAD_MATRICULA ADD CONSTRAINT FK_CAD_MATRICULA1 FOREIGN
KEY (COD_TURMAS) REFERENCES CAD_TURMA (COD_TURMA);
```

```
SET NAMES NONE;
CREATE TABLE CAD_NOTAS_AVALIACOES (
  COD_ALUNO   INTEGER NOT NULL,
  COD_TURMA   INTEGER NOT NULL,
  COD_AVALIACAO INTEGER NOT NULL,
  NOTA        NUMERIC(15,2) NOT NULL);
ALTER TABLE CAD_NOTAS_AVALIACOES ADD CONSTRAINT
FK_CAD_NOTAS_AVALIACOES FOREIGN KEY (COD_AVALIACAO, COD_TURMA)
REFERENCES CAD_AVALIACOES (COD_AVALIACAO, COD_TURMA);
ALTER TABLE CAD_NOTAS_AVALIACOES ADD CONSTRAINT
FK_CAD_NOTAS_AVALIACOES1 FOREIGN KEY (COD_ALUNO, COD_TURMA)
REFERENCES CAD_MATRICULA (COD_ALUNO, COD_TURMAS);
```

```
SET NAMES NONE;
CREATE GENERATOR CODPROFESSORGEN;

CREATE TABLE CAD_PROFESSORES (
  COD_PROFESSORES   INTEGER NOT NULL,
  COD_FUNCIONARIO   INTEGER NOT NULL,
  CARGAHORARIA_PROFESSOR INTEGER NOT NULL);
ALTER TABLE CAD_PROFESSORES ADD CONSTRAINT PK_CAD_PROFESSORES
PRIMARY KEY (COD_PROFESSORES);
ALTER TABLE CAD_PROFESSORES ADD CONSTRAINT FK_CAD_PROFESSORES
FOREIGN KEY (COD_FUNCIONARIO) REFERENCES CAD_FUNCIONARIOS
(COD_FUNCIONARIO);
SET TERM ^;
```

```
/* Trigger: CODPROFESSORTRG */
CREATE TRIGGER CODPROFESSORTRG FOR CAD_PROFESSORES
ACTIVE BEFORE INSERT POSITION 0
AS
begin
  new.cod_professores = gen_id (codprofessorgen,1);
end
^
SET TERM ; ^
```

```

SET NAMES NONE;
CREATE GENERATOR SITUACAOALUNOGEN;
CREATE TABLE CAD_SITUACAOALUNO (
  COD_SITUACAOALUNO INTEGER NOT NULL,
  DESCRICAO          DMDESCRICAO NOT NULL /* DMDESCRICAO = VARCHAR(50)
*/);
ALTER TABLE CAD_SITUACAOALUNO ADD CONSTRAINT
PK_CAD_SITUACAOALUNO PRIMARY KEY (COD_SITUACAOALUNO);
SET TERM ^ ;
/* Trigger: SITUACAOALUNOTRG */
CREATE TRIGGER SITUACAOALUNOTRG FOR CAD_SITUACAOALUNO
ACTIVE BEFORE INSERT POSITION 0
AS
begin
  new.cod_situacaoaluno = gen_id (situacaoalunogen,1);
end
^
SET TERM ; ^

```

```

SET NAMES NONE;
CREATE GENERATOR TURMAGEN;
CREATE TABLE CAD_TURMA (
  COD_TURMA INTEGER NOT NULL,
  SERIE CHAR(2) CHARACTER SET ASCII NOT NULL,
  TURMA CHAR(2) CHARACTER SET ASCII NOT NULL,
  TURNO VARCHAR(20) CHARACTER SET ASCII NOT NULL,
  ANO_TURMA INTEGER NOT NULL);
ALTER TABLE CAD_TURMA ADD CONSTRAINT PK_CAD_TURMA PRIMARY KEY
(COD_TURMA);
SET TERM ^ ;
/* Trigger: TURMATRG */
CREATE TRIGGER TURMATRG FOR CAD_TURMA
ACTIVE BEFORE INSERT POSITION 0
AS
begin
  new.cod_turma= gen_id(turmagen,1);
end
^
SET TERM ; ^

```

```

SET NAMES NONE;
CREATE GENERATOR LOGINEN;
CREATE TABLE CAD_USUARIOS (
  LOGIN VARCHAR(10) CHARACTER SET ASCII NOT NULL,
  COD_FUNCIONARIO INTEGER NOT NULL,
  SENHA VARCHAR(6) CHARACTER SET ASCII NOT NULL,
  ADM CHAR(1) CHARACTER SET ASCII NOT NULL);
/* Check constraints definition */
ALTER TABLE CAD_USUARIOS ADD CONSTRAINT CHK_CAD_USUARIOS check (ADM
in ('S', 'N'));
ALTER TABLE CAD_USUARIOS ADD CONSTRAINT PK_CAD_USUARIOS PRIMARY
KEY (LOGIN);
ALTER TABLE CAD_USUARIOS ADD CONSTRAINT FK_CAD_USUARIOS FOREIGN
KEY (COD_FUNCIONARIO) REFERENCES CAD_FUNCIONARIOS
(COD_FUNCIONARIO);
SET TERM ^ ;
/* Trigger: LOGINTRG */
CREATE TRIGGER LOGINTRG FOR CAD_USUARIOS
ACTIVE BEFORE INSERT POSITION 0
AS

```

```
begin
  new.login = gen_id (logingen,1);
end
^
SET TERM ; ^
```

## ANEXOS

### ANEXO A – PLANO DE VISÃO

<Project Name>

#### Vision

##### 1. Introduction

##### 2. Positioning

###### 2.1 Problem Statement

[Provide a statement summarizing the problem being solved by this project. The following format may be used:]

The problem of	[describe the problem]
affects	[the stakeholders affected by the problem]
the impact of which is	[what is the impact of the problem?]
a successful solution would be	[list some key benefits of a successful solution]

###### 2.2 Product Position Statement

[Provide an overall statement summarizing, at the highest level, the unique position the product intends to fill in the marketplace. The following format may be used:]

For	[target customer]
Who	[statement of the need or opportunity]
The (product name)	is a [product category]
That	[statement of key benefit; that is, the compelling reason to buy]
Unlike	[primary competitive alternative]
Our product	[statement of primary differentiation]



[A product position statement communicates the intent of the application and the importance of the project to all concerned personnel.]

### 3. Stakeholder Descriptions

#### 3.1 Stakeholder Summary

Name	Description	Responsibilities
[Name the stakeholder type.]	[Briefly describe the stakeholder.]	[Summarize the stakeholder's key responsibilities with regard to the system being developed; that is, their interest as a stakeholder. For example, this stakeholder: ensures that the system will be maintainable ensures that there will be a market demand for the product's features monitors the project's progress approves funding and so forth]

#### 3.2 User Environment

[Detail the working environment of the target user. Here are some suggestions:

Number of people involved in completing the task? Is this changing?

How long is a task cycle? Amount of time spent in each activity? Is this changing?

Any unique environmental constraints: mobile, outdoors, in-flight, and so on?

Which system platforms are in use today? Future platforms?

What other applications are in use? Does your application need to integrate with them?

This is where extracts from the Business Model could be included to outline the task and roles involved, and so on.]

## 4. Product Overview

### 4.1 Product Perspective

[This subsection of the **Vision** document puts the product in perspective to other related products and the user's environment. If the product is independent and totally self-contained, state it here. If the product is a component of a larger system, then this subsection needs to relate how these systems interact and needs to identify the relevant interfaces between the systems. One easy way to display the major components of the larger system, interconnections, and external interfaces is with a block diagram.]

### 4.2 Assumptions and Dependencies

[List each factor that affects the features stated in the **Vision** document. List assumptions that, if changed, will alter the **Vision** document. For example, an assumption may state that a specific operating system will be available for the hardware designated for the software product. If the operating system is not available, the **Vision** document will need to change.]

### 4.3 Needs and Features

[Avoid design. Keep feature descriptions at a general level. Focus on capabilities needed and why (not how) they should be implemented.]

Need	Priority	Features	Planned Release

#### **4.4 Alternatives and Competition**

[Identify alternatives the stakeholder perceives as available. These can include buying a competitor's product, building a homegrown solution, or simply maintaining the status quo. List any known competitive choices that exist or may become available. Include the major strengths and weaknesses of each competitor as perceived by the stakeholder or end user.]

#### **5. Other Product Requirements**

[At a high level, list applicable standards, hardware, or platform requirements; performance requirements; and environmental requirements.]

Define the quality ranges for performance, robustness, fault tolerance, usability, and similar characteristics that are not captured in the Feature Set.

Note any design constraints, external constraints, or other dependencies.

Define any specific documentation requirements, including user manuals, online help, installation, labeling, and packaging requirements.

Define the priority of these other product requirements. Include, if useful, attributes such as stability, benefit, effort, and risk.]

## ANEXO B – PLANO DE ITERAÇÃO

<Project Name>

### Iteration Plan

#### 1. Key Milestones

[Detailed diagrams showing timelines, intermediate milestones, when testing starts, beta version, demos and so on for the iteration.]

Milestone	Date
Iteration Start	
Iteration Stop	

#### 2. Iteration Objectives

[Objectives/tasks assigned as part of this iteration and assignment to persons responsible]

[Objectives may include creating or refining specific artifacts, addressing risks, or implementing specific requirements, or performing supporting tasks. Some example objectives are listed below]

Objective/Task	Assigned to
Implement Use Case: Register for Course, Basic flow, Alternatve 1, Alternate 2	Fred
Complete Vision	Jill
Detail UC3: Publish Calendar	John
Test all developed requirements	Lance
Create plan for next iteration	Jill

## ANEXO C – PLANO DE PROJETO

<Project Name>

### Project Plan

#### 1. Introduction

#### 2. Project Organization

##### 2.1 Organizational Structure

##### 2.2 External Interfaces

[Describe how the project interfaces with external groups. For each external group, identify the internal and external contact names. This should include responsibilities related to deployment and acceptance of the product.]

##### 2.3 Roles and Responsibilities

[Identify the people or organizational units that have a role on the project, and what role(s) they play.]

Person	Role

### 3. Project Schedule

[Diagrams or tables showing target dates for completion of iterations and phases, release points, demos, and other milestones.] For example:

<b>Phase</b>	<b>Iteration</b>	<b>Primary Objective (risks/work addressed)</b>	<b>Scheduled Start/Stop</b>	<b>Effort Estimate (person days)</b>
Inception	I1			
Elaboration	E1			
Construction	C1			
	C2			
Transition	T1			
	T2			

### 4. Project Resourcing

[Identify the numbers and type of staff required here, including any special skills or experience, scheduled by project phase or iteration.

Describe how you will approach finding and acquiring the staff needed for the project.

List any special training project team members will require, with target dates for when this training should be completed.]

### 5. Budget

[Describe the size of the budget, how it is allocated, and how it will be monitored.]

## 6. Risk List

[Describe and revisit periodically the risks associated to this project.]

<b>Description</b>	<b>Type</b> (direct/indirect)	<b>Probability</b> (H/M/L)	<b>Impact</b> (H/M/L)	<b>Magnitude</b> (H/M/L)

## ANEXO D – DESCRIÇÃO DOS CASOS DE USO

<Project Name>

**Use-Case:** <use-case name>

### **Brief Description**

<brief description of use-case>

### **Actors Brief Descriptions**

<brief description of actors interacting with use case>

### **Preconditions**

<pre-condition 1>

### Basic Flow of Events

1. <basic flow steps>

2...

3.The use case ends.

### **Alternative Flows**

<alternate flow 1>



**Subflows**

<subflow 1>

**Key Scenarios**

<scenario 1>

**Post-conditions**

<post-condition 1>

**Extension Points**

<extension point 1>

**Special Requirements**

<supplementary requirement 1>

**Additional Information**