

FUNDAÇÃO DE ENSINO “EURÍPEDES SOARES DA ROCHA”  
CENTRO UNIVERSITÁRIO “EURÍPEDES DE MARÍLIA” - UNIVEM

**JULIANA PEREIRA DE SOUZA**

**ANÁLISE DE REQUISITOS E PROJETO ORIENTADO A OBJETO  
USANDO UML: Um Estudo de Caso**

MARÍLIA  
2005

JULIANA PEREIRA DE SOUZA

ANÁLISE DE REQUISITOS E PROJETO ORIENTADO A OBJETO  
USANDO UML: Um Estudo de Caso

Monografia apresentada ao Curso de Graduação em Ciência da Computação, do Centro Universitário Eurípides de Marília, mantido pela Fundação de Ensino Eurípides Soares da Rocha - Univem, como requisito parcial para a obtenção do título de Graduado em Ciência da Computação.

Orientadora: Prof. Dra. Ana Paula P. M. Peruzza  
CO-Orientador: Prof. Dr. Edmundo Sérgio Spoto

MARÍLIA  
2005

SOUZA, Juliana Pereira

Análise de Requisitos e Projeto Orientado a Objeto usando UML: Um estudo de Caso / Juliana Pereira de Souza; orientadora: Prof. Ana Paula P. M. Peruzza; co-orientador: Prof. Edmundo Sérgio Spoto, Marília, SP: [s.n.], 2005.

156 f.

Monografia (Graduação em Ciência da Computação) – Centro Universitário Eurípides de Marília – Fundação de Ensino Eurípides Soares da Rocha.

1. Engenharia de Software    2. UML  
3. Software Educacional de Química.

CDD: 005.1

*Dedico este trabalho  
ao meu pai Carlos Péricles Martins de Souza,  
a minha mãe Edina Pereira de Souza  
e a minha irmã Luciana Pereira de Souza  
pela dedicação, incentivo, paciência e sacrifícios pela minha formação.*

*Ao meu namorado Christiano Augusto Lopes,  
pelos ótimos momentos de convivência, amor, amizade,  
incentivo e compreensão pela minha ausência.*

*Pois nada teria sido feito sem o amor da minha família.*

## AGRADECIMENTOS

*A **Deus**, divino e criador,  
pela vida e por ter me proporcionado  
mais essa conquista.*

*A todos meus mestres que participaram dessa caminhada,  
em especial a minha orientadora Prof. Ana Paula Melchiori Peruzzo  
e meu Co-Orientador Prof. Edmundo Sérgio Spoto  
pela competência, dedicação, incentivo e amizade de ambos,  
não só da elaboração desse trabalho,  
como em toda essa jornada acadêmica.*

*Ao Prof. Paulo Marinelli,  
pela paciência e tempo dispensados  
em ajudar na compreensão da matéria de química.*

*Ao meu amigo Ederson Relvas,  
pelo incentivo e amizade.  
E a todos que direta ou indiretamente  
contribuíram no desenvolvimento deste trabalho.*

*“Somos o que fazemos,  
mas somos, principalmente,  
o que fazemos para mudar o que somos”.*

*(Eduardo Galeano)*

SOUZA, Juliana Pereira de. **Análise de Requisitos e Projeto Orientado a Objeto Usando UML: Um Estudo de Caso.** 2005. 156 f. Monografia (Graduação em Ciência da Computação) – Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2005.

## RESUMO

O trabalho descreve um estudo de caso de um Software Educacional de Química, no modelo construtivista, no qual o objetivo metodológico é auxiliar o aluno (usuário principal) analisar, apreender e compreender a respeito das ligações químicas existentes, e os compostos químicos que poderiam criar em um ambiente de laboratório, de forma que consiga a visualização, pois não poderiam ser feitas a olho nu. Foi desenvolvido de acordo com a lei de diretrizes e bases da Educação Nacional referente ao ensino médio. Além disso, relata-se um estudo científico e exploratório de abordagens da Linguagem de Modelagem Unificada (UML), conceituando a metodologia orientada a objetos (OO), podendo com a prática, ser modelados seus diagramas: Caso de Uso, Classe (lógico e físico), Objeto, Interação (Seqüência), Atividade, Estado e Implementação (Implantação e Componente) com requisitos bem definidos. Entretanto, enfocam-se as tarefas de Análise e Projeto, de acordo com o estudo de um Processo Unificado (RUP).

**Palavras-Chave:** Engenharia de Software. UML. Software Educacional de Química.

SOUZA, Juliana Pereira de. **Análise de Requisitos e Projeto Orientado a Objeto Usando UML: Um Estudo de Caso.** 2005. 156 f. Monografia (Graduação em Ciência da Computação) – Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2005.

## ABSTRACT

The essay describes the study of an Educational Chemistry Software in a constructive model whose methodological aim is to help the student, (main user) to analyse, learn and understand the existent chemical connections and the chemical compounds they could create in a laboratorial environment, so that the experiments could be visualized, for it could not be possible with a naked eye. It was developed according to the law of guidelines and bases of the National Education regarding the high school teaching. Besides, it states an explanatory and scientific study of the Unified Modeling Language (UML) approach, giving a concept to the Object Oriented (OO) methodology allowing it through practicing to model their diagrams: Case of Use, Class (Logical and physical). Object, Interaction, (Sequence). Activity, state of implementation (Implantation and Component) with very defined requirements. However, the Analysis and Project tasks are focused according to the study of a Unified Process (RUP).

**Keywords:** Software Engineering. UML. Educational Chemistry Software.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Exemplo de Classe.....	27
Figura 2 - Exemplo de Diagrama de Caso de Uso. ....	30
Figura 3 - Exemplo de Diagrama de Classes.....	32
Figura 4 - Exemplo de Agregação entre Classes.....	33
Figura 5 - Exemplo de Dependência entre classes. ....	34
Figura 6 - Exemplo de Diagrama de Seqüência. ....	37
Figura 7 - Exemplo de Máquina de Estado (FURLAN, 1998).....	39
Figura 8 - Exemplo de Máquina de Estado (FURLAN, 1998).....	40
Figura 9 - Exemplo de Diagrama de Atividade.....	43
Figura 10 - Exemplo de Componente.....	45
Figura 11 - Exemplo de Diagrama de Implantação (FURLAN, 1998). ....	46
Figura 12 - Distinção entre Elemento e de Substância Pura (CANTO, 2002). ....	53
Figura 13 - Tabela Periódica (REIS, 2005). ....	53
Figura 14 - Distribuição Eletrônica - Família 1 ou IA (REIS, 2005). ....	54
Figura 15 - Distribuição Eletrônica - Família 2 ou IIA (REIS, 2005).....	55
Figura 16 - Apresentação do subnível mais energético - Elementos Representativos (REIS, 2005).....	56
Figura 17 - Apresentação do subnível mais energético (escrito em azul) e do mais externo (escrito em preto) - Elementos de Transição (REIS, 2005).....	56
Figura 18 - Série lantanídios e Série actinídios (REIS, 2005).....	57
Figura 19 - Apresentação do subnível mais energético - Série lantanídios e Série actinídios (REIS, 2005).....	57
Figura 20 - Tipo de Substância Química (CANTO, 2002).....	58
Figura 21 - Fórmula de um Composto Iônico .....	61
Figura 22 - Exemplos de Compostos Iônicos.....	61
Figura 23 - Diagrama de Casos de Usos dos desenvolvedores do SEQ.....	68
Figura 24 - Fases RUP (Ciclo de Vida Adotado para SEQ) (TAPAS, 2005). ....	71

Figura 25 - Visão 1 - Laboratório Químico.....	77
Figura 26 - Visão 2 - Biblioteca. ....	78
Figura 27 - Visão 3 - Armário, com as gavetas e sem os elementos químicos. ....	78
Figura 28 - Porta automática .....	81
Figura 29 - Diagrama de Casos de Usos do SEQ.....	85
Figura 30 - Diagrama de Classe (lógico) Preliminar.....	87
Figura 31 - Diagrama de seqüência do caso de uso “Ligar Elemento Químico” .....	89
Figura 32 - Diagrama de Atividade de caso de uso “Ligar Elemento Químico” .....	91
Figura 33 - Diagrama de Classes do SEQ .....	92
Figura 34 - Diagrama de Objetos do SEQ.....	93
Figura 35 - Diagrama de Estados da classe Usuário .....	95
Figura 36 - Diagrama de Componente do SEQ.....	96
Figura 37 - Diagrama de Implantação do SEQ.....	97
Figura A1 - MER Lógico. ....	103
Figura A2 - Diagrama de Seqüência do caso de uso “Logar no Sistema”. ....	105
Figura A3 - Diagrama de Atividade do caso de uso “Logar no Sistema”. ....	105
Figura A4 - Diagrama de Seqüência do caso de uso “Autenticar Senha”.....	106
Figura A5 - Diagrama de Atividade do caso de uso “Autenticar Senha”.....	107
Figura A6 - Diagrama de Seqüência do caso de uso “Solicitar Senha”. ....	108
Figura A7 - Diagrama de Atividade do caso de uso “Solicitar Senha”. ....	108
Figura A8 - Diagrama de Seqüência do caso de uso “Cadastrar Usuário”.....	109
Figura A9 - Diagrama de Atividade dos casos de uso “Cadastrar Usuário”, “Validar Cadastro Aluno”, “Controlar o Sistema” e “Validar Cadastros”.....	110
Figura A10 - Diagrama de Seqüência do caso de uso “Validar Cadastro Aluno”. ....	111
Figura A11 - Diagrama de Seqüência do caso de uso “Controlar o Sistema”.....	112
Figura A12 - Diagrama de Seqüência do caso de uso “Validar Cadastros”.....	113
Figura A13 - Diagrama de Seqüência do caso de uso “Acessar Área Administrador”.....	114
Figura A14 - Diagrama de Atividade do caso de uso “Acessar Área Administrador”. ....	114

Figura A15 - Diagrama de Seqüência do caso de uso “Controlar Cadastro de Usuário”.....	115
Figura A16 - Diagrama de Atividade do caso de uso “Controlar Cadastro de Usuário”. .....	115
Figura A17 - Diagrama de Seqüência do caso de uso “Acessar Área Professor ” .....	116
Figura A18 - Diagrama de Atividade do caso de uso “Acessar Área Professor ” .....	116
Figura A19 - Diagrama de Seqüência do caso de uso “Cadastrar Composto Ativo”.....	117
Figura A20 - Diagrama de Atividade dos casos de uso “Cadastrar Composto Ativo” e “Validar Cadastro Composto Ativo.....	117
Figura A21 - Diagrama de Seqüência do caso de uso “Validar Cadastro Composto Ativo”. ..	118
Figura A22 - Diagrama de Seqüência do caso de uso “Controlar Cadastro de Composto” ...	119
Figura A23 - Diagrama de Atividade do caso de uso “Controlar Cadastro de Composto” ...	120
Figura A24 - Diagrama de Seqüência do caso de uso “Alterar Cadastro de Composto Inativo” .....	121
Figura A25 - Diagrama de Atividade do caso de uso “Alterar Cadastro de Composto Inativo” .....	121
Figura A26 - Diagrama de Seqüência do caso de uso “Cadastrar Arquivo PDF”.....	122
Figura A27 - Diagrama de Atividade dos casos de uso “Cadastrar Arquivo PDF” e “Validar Cadastro PDF” .....	122
Figura A28 - Diagrama de Seqüência do caso de uso “Validar Cadastro PDF” .....	123
Figura A29 - Diagrama de Seqüência do caso de uso “Controlar Cadastro de Arquivo”.....	124
Figura A30 - Diagrama de Atividade do caso de uso “Controlar Cadastro de Arquivo”.....	124
Figura A31 - Diagrama de Seqüência do caso de uso “Cadastrar Elemento Químico”.....	125
Figura A32 - Diagrama de Atividade dos casos de uso “Cadastrar Elemento Químico” e “Validar Cadastro Elemento” .....	125
Figura A33 - Diagrama de Seqüência do caso de uso “Validar Cadastro Elemento”.....	126
Figura A34 - Diagrama de Seqüência do caso de uso “Controlar Cadastro de Elemento”....	127
Figura A35 - Diagrama de Atividade do caso de uso “Controlar Cadastro de Elemento”....	128
Figura A36 - Diagrama de Seqüência do caso de uso “Acessar Área Aluno”.....	128
Figura A37 - Diagrama de Atividade do caso de uso “Acessar Área Aluno”.....	129
Figura A38 - Diagrama de Seqüência do caso de uso “Acessar Objetos Virtuais”.....	129

Figura A39 - Diagrama de Atividade do caso de uso “Acessar Objetos Virtuais”	130
Figura A40 - Diagrama de Seqüência do caso de uso “Gerar Animação e Escolher Arquivo”	130
Figura A41 - Diagrama de Atividade do caso de uso “Gerar Animação e Escolher Arquivo”	131
Figura A42 - Diagrama de Seqüência do caso de uso “Escolher Elemento Químico”	131
Figura A43 - Diagrama de Atividade do caso de uso “Escolher Elemento Químico”	132
Figura A44 - Diagrama de Seqüência do caso de uso “Excluir Escolha de Elemento”	133
Figura A45 - Diagrama de Atividade do caso de uso “Excluir Escolha de Elemento”	133
Figura A46 - Diagrama de Seqüência do caso de uso “Selecionar Elemento Químico”	134
Figura A47 - Diagrama de Atividade do caso de uso “Selecionar Elemento Químico”	134
Figura A48 - Diagrama de Seqüência do caso de uso “Ligar Elemento Químico”	135
Figura A49 - Diagrama de Atividade do caso de uso “Ligar Elemento Químico”	136
Figura A50 - Diagrama de Seqüência do caso de uso “Cortar Ligação”	136
Figura A51 - Diagrama de Atividade do caso de uso “Cortar Ligação”	137
Figura A52 - Diagrama de Seqüência do caso de uso “Excluir Composto (Inativo)”	137
Figura A53 - Diagrama de Atividade do caso de uso “Excluir Composto (Inativo)”	138
Figura A54 - Diagrama de Seqüência do caso de uso “Buscar Ligação Binária”	138
Figura A55 - Diagrama de Atividade do caso de uso “Buscar Ligação Binária”	139
Figura A56 - Diagrama de Seqüência do caso de uso “Acessar Ambientes”	139
Figura A57 - Diagrama de Atividade do caso de uso “Acessar Ambientes”	140
Figura A58 - Diagrama de Seqüência do caso de uso “Cadastrar Composto Inativo”	141
Figura A59 - Diagrama de Atividade dos casos de uso “Cadastrar Composto Inativo” e “Validar Cadastro Composto Ativo”	141
Figura A60 - Diagrama de Seqüência do caso de uso “Validar Cadastro Composto Inativo”	142
Figura A61 – MER Físico	143

## LISTA DE TABELAS

Tabela 1 - Exemplo de Objeto.....	25
Tabela 2 - Elementos do diagrama de caso de uso.....	29
Tabela 3 - Distribuição eletrônica dos gases nobres (CANTO, 2002).....	59
Tabela 4 - Comparação das substâncias Iônica, Molecular e Metálica.....	64
Tabela 5 - Identificação dos alunos e seus trabalhos.....	67
Tabela 6 - Lista de Atores do Diagrama de Casos de Usos.....	68
Tabela 7 - Lista de Atores do Diagrama de Casos de Usos do Projeto.....	83
Tabela A1 - Identificação e Comentário das Tabelas.....	144
Tabela A2 - Identificação e Comentário dos Atributos da Tabela ADMINISTRADOR_SISTEMA.....	145
Tabela A3 - Identificação e Comentário dos Atributos da Tabela ALUNO.....	145
Tabela A4 - Identificação e Comentário dos Atributos da Tabela BIBLIOTECA.....	145
Tabela A5 - Identificação e Comentário dos Atributos da Tabela COMPOSTO.....	145
Tabela A6 - Identificação e Comentário dos Atributos da Tabela COMPOSTO_ELEMENTO. .....	146
Tabela A7 - Identificação e Comentário dos Atributos da Tabela ELEMENTO_QUÍMICO. .....	146
Tabela A8 - Identificação e Comentário dos Atributos da Tabela PROFESSOR.....	147
Tabela A9 - Identificação e Comentário dos Atributos da Tabela USUÁRIO.....	147

## LISTA DE ABREVIATURAS E SIGLAS

DD	Dicionário de Dados
MER	Modelo Entidade-Relacionamento
OMG	Object Management Group
OMT	Técnica de Modelagem de Objetos
OO	Orientado a Objetos
OOAD	Object Oriented Analysis and Design
OOD	Projeto Orientado a Objetos
OOP	Programação Orientada a Objetos - Object Oriented Programming
OOSE	Engenharia de Software Orientada a Objetos
RUP	Rational Unified Process
SDEXPO	Software Development Conference & Expo
SEQ	Software Educacional de Química
SGBD	Sistema de Gerenciamento de Banco de Dados
UML	Linguagem de Modelagem Unificada - Unified Modeling Language
WTTML	Workshop on Integration and Transformation of UML Models

## SUMÁRIO

INTRODUÇÃO.....	16
CAPÍTULO 1 - LINGUAGEM DE MODELAGEM UNIFICADA (UML).....	20
1.1 História da UML.....	21
1.2 Conceitos de Orientação a Objetos.....	22
1.2.1 Conceitos Básicos.....	24
1.3 Diagramas da UML.....	27
1.3.1 Diagrama de Caso de Uso – Use Case.....	28
1.3.2 Diagrama de Classe.....	30
1.3.3 Diagramas de Interações.....	35
1.3.4 Diagrama de Estado e Atividade.....	37
1.3.5 Diagrama de Implementação.....	44
CAPÍTULO 2 - ANÁLISE E ESPECIFICAÇÃO DOS REQUISITOS.....	47
2.1 Objetivo do Software Educacional de Química.....	50
2.2 Informações Sobre o Sistema.....	51
2.2.1 Tabela Periódica.....	51
2.2.2 Ligações Químicas.....	57
2.3 Identificação do Problema.....	64
2.3.1 Sobre o Software.....	64
2.3.2 Necessidades do Sistema.....	66
2.4 Limites ao Uso do Software Educacional de Química.....	66
2.5 Descrição Geral.....	67
2.6 Ciclo de Vida - RUP.....	69
CAPÍTULO 3 - ESTUDO DE CASO.....	74
3.1 Descrição da Informação.....	74
3.1.1 Descrição Inicial da Interface Visual do Software.....	80
3.2 O Desenho do Diagrama de Casos de Uso.....	82
3.3 Modelo de Classe Preliminar.....	86
3.4 Desenhando Sistemas Dinâmicos.....	87
3.5 Desenhando Diagrama de Atividade.....	90
3.6 Desenhando Classes com Operações.....	91
3.7 Desenhando Diagrama de Estado.....	94
3.8 Desenhando Aspectos de Implementações.....	95
CONCLUSÕES.....	98
REFERÊNCIAS.....	100
APÊNDICE A - Modelo Entidade Relacionamento (Lógico).....	103

APÊNDICE B - Descrição dos Casos de Uso, Diagrama de Seqüência e de Atividade.....	104
APÊNDICE C - Modelo Entidade Relacionamento (Físico) .....	143
APÊNDICE D - Dicionário de Dados .....	144
APÊNDICE E - Relatórios de Revisões Técnicas.....	148

## INTRODUÇÃO

Atualmente, as metodologias de Engenharia de Software, como a programação e testes, são quantitativas, onde buscam alcançar a produtividade e a qualidade de softwares. No entanto, esta se relaciona diretamente com a qualidade dos processos e desenvolvimentos dos mesmos.

A qualidade e a produtividade do desenvolvimento de softwares deixaram de serem responsabilidades exclusivas dos dirigentes de uma organização, e passaram a ser de toda a estrutura do meio empresarial (camadas organizacionais), em qualquer nível (executivo, estratégico e tático). Todavia, para a sobrevivência desta organização, são executados processos definidos, com intuito de melhorias, até chegar ao aperfeiçoamento em cada uma das respectivas etapas.

Com as indústrias de informática vêm surgindo ambientes gráficos e interfaces ricas para o desenvolvimento de sistemas, buscando a diminuição de dificuldades para os usuários de softwares. Surgiram diversas metodologias que ofereciam direcionamentos para o futuro no ambiente computacional. A primeira metodologia, denominada técnica estruturada, enfoca a utilização de procedimentos e funções. Atualmente, muitos sistemas são desenvolvidos com base nesta técnica, e, vêm apresentando problemas de natureza grave, como: o sistema se torna instável, devido à dificuldade de gerar manutenção, deixando-o de má qualidade, pois os requisitos se modificam com frequência (principalmente para sistemas complexos) e o sistema cresce, devendo passar por feedback e manutenções.

As metodologias evoluíram e passaram a adotar notações diferentes. Nasceram então as metodologias orientadas a objetos (OO), enfocando-se classes e objetos. Para o meio empresarial, existe uma pergunta bastante significativa: Porque 100% das organizações não adotaram a tecnologia de objetos? Pela falta de pessoas treinadas, incertezas, ferramentas

imaturas, ou seja, para qualquer nova tecnologia adotada, há resistência em partes de seus usuários (FURLAN, 1998).

Entretanto, essa técnica ainda é bastante aceita na comunidade de informática, todavia, se caracteriza pelo desenvolvimento de sistema que abrange todos problemas e níveis de complexidade.

Para tal aceitação e sucesso, surgiram diversas metodologias sobre as mesmas notações OO. Essas metodologias cresceram de maneira desordenada em um curto espaço de tempo, dando origem a diversos métodos, cada qual com suas falhas e sem relacionamentos com as demais. Após esses surgimentos de métodos, seus usuários sentiram dificuldades de encontrar uma linguagem de modelagem, com objetivos de abrangerem todos os conceitos OO.

Os autores Grady Booch com a notação: Projeto Orientado a Objeto (OOD), James Rumbaugh com a notação: Técnica de Modelagem de Objetos (OMT) e Ivar Jacobson com a notação: Engenharia de Software Orientada Objetos (OOSE), motivados a criarem juntos uma linguagem de modelagem madura, aprimorando métodos já existentes e solucionando problemas que ninguém conseguiria até então. Mesmo porque, seus métodos estavam nessa mesma direção, com processos independentes e a unificação adquiria estabilidade no mercado OO. Através disto, surgia a Linguagem de Modelagem Unificada (UML).

O estudo enfocou a UML como sendo uma parte da metodologia responsável pela modelagem do sistema, disponibilizando ferramentas para criar e ler diagramas. A ordem e quais diagramas devem ser criados, dentro de um estudo de caso, é dada pela metodologia do ciclo de vida Rational Unified Process (RUP). Entretanto, será abordado um estudo de caso, de um Software Educacional de Química (SEQ), enfatizando as fases de Análise de Requisitos e Projeto.

O objetivo do estudo de caso foi desenvolver o SEQ no modelo construtivista, no qual o aluno (usuário principal) poderá analisar, apreender e compreender a respeito das ligações químicas existentes, e os compostos químicos que poderiam criar em um ambiente de laboratório, de forma que consiga a visualização, pois não poderiam ser feitas a olho nu.

Esta monografia possui três capítulos distintos, através dos quais, foi abordado o tema da maneira mais adequada possível.

No primeiro capítulo é apresentado de forma mais extensiva e minuciosa, o estudo científico e exploratório dos conceitos e definições sobre a UML, enfatizando a metodologia OO e termos como diagramas estruturais, comportamentais e arquiteturais que fazem parte da UML.

No segundo capítulo há a concepção de informações para o desenvolvimento do sistema, através de reuniões e revisões técnicas, como por exemplo, objetivos e necessidades do sistema. Também abordando o SEQ como sendo sistema especialista, porém, trazendo feedback e requisitos bem definidos, como o estudo da tabela periódica, ligações químicas e limites sobre o software educacional, atendendo a lei de diretrizes e bases da Educação Nacional para o ensino médio, enfocando a metodologia de ensino Construtivista. Este capítulo apresenta também o ciclo de vida adotado RUP.

E por fim, no terceiro capítulo elabora-se um estudo de caso, sendo agregado a uma análise detalhada sobre as funcionalidades do SEQ. Essa análise foi criada pelos desenvolvedores do sistema através de reuniões e conseqüentemente foram modelados os diagramas da UML. Porém, colocando em prática os conceitos do ciclo de vida RUP, implementando as tarefas de Análise e Projeto. Cabe salientar que, por se tratar de um trabalho acadêmico e de um tema imensamente complexo, o assunto não será esgotado, mas sim focado em alguns aspectos metodológicos (conceitos e práticas).

A pretensão é a de demonstrar a vasta extensão do tema, enfatizando a importância da modelagem de um sistema em OO, para qual, podemos visualizar e controlar seu desenvolvimento de maneira eficaz e detalhada, identificando e gerenciando riscos, estipulando e cumprindo prazos, dentro das estimativas de custo.

## CAPÍTULO 1 - LINGUAGEM DE MODELAGEM UNIFICADA (UML)

Para a concordância das definições conceituais apresentadas neste capítulo, foram utilizados alguns levantamentos bibliográficos: (LARMAN, 1999); (MATOS, 2002) e (DEITEL, 2003).

“A UNIFIED MODELING LANGUAGE (UML) é agora o esquema de representação gráfica mais amplamente utilizado para modelagem de sistemas OO. Ela certamente unificou os diversos esquemas de notação mais populares” (DEITEL, 2003).

A UML é uma linguagem de modelagem gráfica para descrever um projeto de software. O entendimento da UML é dado pela sua concepção de métodos, processo de análise, projeto e construção de software, criando visões do sistema que está em construção.

Decorre por três notações:

- OOD (Projeto Orientado a Objetos) – Grady Booch;
- OMT (Técnica de Modelagem de Objetos) – James Rumbaugh;
- OOSE (Engenharia de Software Orientada a Objetos) – Ivar Jacobson.

É notável a importância da criação de modelos para que bons sistemas obtenham qualidade, ou seja, dado um sistema, sua modelagem envolve diversas características e aspectos úteis na construção de seu desenvolvimento. Algumas vantagens importantes de utilizar a UML:

- A diminuição da complexidade no entendimento da construção de um sistema;
- Possibilidade de simular processos perigosos ou danosos em nível de sistema;
- A visualização dos problemas do sistema, antes que os aconteçam;
- Facilidade de testar o sistema;

- Facilidade de enxergar realidade por meio de uma abstração que pode ser facilmente entendida;
- Modelando o sistema em UML, não significa que a implementação será em OO;
- Utilização das técnicas OO adquirindo suas vantagens: reutilização de código, modelagem em um alto nível de abstração, etc...

Além desses aspectos, em seu estado atual, a UML define formalmente o formato estático, usando um metamodelo expressado em diagrama de classe, no qual é independente de linguagens de programação e processos de desenvolvimento. Todavia, aumenta as fabricações de ferramentas OO no mercado, sustentando todos os níveis de desenvolvimento de sistemas simples ou complexos em todas as fases de seu ciclo de vida.

Para cumprir seu objetivo, a UML permite que seus usuários modelem um sistema sob diferentes perspectivas. Cada uma destas perspectivas é uma abstração apresentada por diagramas criados a partir dos recursos oferecidos pela linguagem de modelagem.

É de grande importância ressaltar que OO, nos dias atuais, está começando a fazer parte de alguns sistemas comerciais: sistemas de WEB, etc., que até então, a técnica estruturada já havia tomado conta. Portanto, a UML é uma composição das boas características de outras notações que utilizam a OO, e está direcionada a desenvolvedores atuais, pois se caracterizam pela “facilidades de conceber, compreender, testar, validar, (arquitetar lógica e fisicamente) e ainda identificar todos os possíveis comportamentos do sistema” (MATOS, 2002).

## **1.1 História da UML**

A partir da década de 80, algumas organizações começaram a utilizar OOP (OBJECT ORIENTED PROGRAMMING), que permitiu aos projetistas implementar o OOD como um sistema que funcionasse. A partir dessas organizações, surgiu a necessidade de um processo

adequado para modelar a OOAD (OBJECT ORIENTED ANALYSIS AND DESIGN), ou seja, necessidades de desenvolver um processo que envolva analisar e projetar um sistema de um ponto de vista OO.

No final de 1980, havia um grande desenvolvimento de técnicas e ferramentas na OO. Em 1994 dois importantes autores, James Rumbaugh e Grady Booch imaginaram a fusão dos conceitos de seus métodos com a criação de um método unificado. Posteriormente juntava-se a eles Ivar Jacobson.

Com essa fusão implantada, em 1995 uma idéia do Grady Booch e James Rumbaugh, com a criação de um padrão, as indústrias participariam e opinariam, dando início a versão draft 0.8 de UML. A partir desta, as empresas de informática começaram a entender essa nova proposta que principalmente aumentaria a qualidade dos processos do software. A UML foi aprovada pelo OMG (OBJECT MANAGEMENT GROUP) em novembro de 1997.

Surgia então a primeira proposta completa da linguagem: versão 0.9 UML. Já em 2002, foi lançada a versão 1.4, na qual suas modificações são constantes em eventos e listas de discussões como: International Conference on UML, WITML (Workshop on Integration and Transformation of UML Models) e Sdexpo (Software Development Conference & Expo).

## **1.2 Conceitos de Orientação a Objetos**

A UML se caracteriza pela modelagem de sistemas concorrentes e distribuídos, reunindo vários recursos da OO. Entretanto, para abstrairmos os diagramas propostos pela UML, deve-se conhecer em primeiro lugar as técnicas OO, conseqüentemente sua história, entendendo também a diferença entre técnica estruturada e as vantagens de sua utilização.

Primeiramente, será apresentada uma breve avaliação entre as técnicas estruturadas e OO. DEITEL apresenta a primeira técnica como uma abordagem disciplinada para escrever programas, que são mais claros, fáceis de testar e depurar e mais fáceis de modificar do que programas não-estruturados, porém teve seu início de evolução na década de 1960. Foi

apresentada em 1971 a linguagem de programação Pascal por Nicklauss Wirth. Entre 1970 e o início de 1980 foi desenvolvida a linguagem de programação ADA sob o patrocínio do Departamento de Defesa dos Estados Unidos (DOD). Por fim, a linguagem de programação C, onde foram usadas (código legado de C) por mais de três décadas. Entretanto, centenas de linguagens estavam sendo utilizadas para produzir sistemas de controle e comando do DOD.

Atualmente, as técnicas estruturadas são, sem dúvida, as mais populares na comunidade de informática, em relação ao seu início. À medida que essas técnicas são utilizadas, encontram-se alguns problemas como: decomposição funcional inadequada em situações de sistemas complexos, principalmente para profissionais iniciantes. Todavia, esses sistemas ainda são difíceis de serem incrementados com novas funções, e as alterações em funções já existentes, muitas vezes, provocam sérios problemas em outras partes do software.

A OOP se iniciou no final da década de sessenta com a linguagem SIMULA-67 que apresentava conceitos e classes, rotinas e subclasses. Nos anos setenta, fazia parte importante na linguagem SMALLTALK, desenvolvida pela Xerox. Este paradigma só estreou na análise de sistemas no final da década de oitenta.

Orientação a Objetos é o maior avanço em software destes últimos anos, além de ser a forma mais natural de se analisar o mundo.

As palavras chaves de alguns autores sobre OOP são programação e abstração. “A proposta da orientação a objeto é permitir que os programadores organizem os programas da mesma forma que as nossas mentes enxergam os problemas: não como um conjunto de espaços de memória, mas como um conjunto de coisas que fazem parte do problema” (MATOS, 2002).

Uma de suas principais características é o modo de terem unificado os formalismos utilizados na análise, projeto e programação. A OO apresenta várias vantagens para sua

utilização, ou seja, apresenta vários benefícios no desenvolvimento e manutenção de software.

São elas:

- Facilidade para reutilização de código;
- Possibilidade do usuário (desenvolvedor) trabalhar em um nível mais elevado de abstração;
- Utilização de um único padrão conceitual durante todo o processo de criação de software;
- Adequação à arquitetura cliente/servidor;
- Facilidade de comunicação com os usuários e com outros profissionais de informática.
- Ciclo de vida mais longo para os sistemas;
- Desenvolvimento acelerado de sistemas;
- Construções de sistemas muito mais complexos, pela incorporação de funções prontas;
- Menor custo para desenvolvimento e manutenção de sistemas.

### 1.2.1 Conceitos Básicos

Mesmo para alguns programadores que utilizam OOP, para a eficiência e qualidade dos diagramas UML, é indispensável o entendimento dos conceitos da OO. Portanto, serão mostrados alguns tópicos importantes: Objeto, Estado, Comportamento, Identidade, Mensagem, Polimorfismo, Classe e Encapsulamento.

O primeiro conceito da OO é o próprio **objeto**. Um objeto é tudo aquilo que vemos no mundo real: aluno, pessoas, carros, computadores, etc. Em termos de abstração, os objetos não se separam, já na programação estruturada, são separados em dados e funções. Existem diversos conceitos a respeito de objetos. Uma definição aceita seria o fato de que possuem atributos e métodos comuns.

Atributos são características do objeto e métodos são operações ou funções oferecidas sobre aquilo que ele pode fazer. Por exemplo:

Tabela 1 - Exemplo de Objeto.

Aluno
Atributos: nome, idade, serie, telefone, etc.
Métodos: matricular, transferir, cancelar aula, etc.

Com essas características podem ser identificados diversos objetos semelhantes, como por exemplo, o objeto Christiano é um aluno, e a Luciana também.

O objeto é definido também por três características importantes: estado comportamento e identidade.

**O estado** é a situação na qual um objeto está em um determinado momento, por exemplo, um objeto animal pode estar no estado “dormindo” ou no estado “acordado”, podendo ser atribuído outros estados para esse objeto.

**O comportamento** é a maneira como um objeto reage ao seu ambiente em um determinado estado. São métodos e funções específicas do objeto. Normalmente, isto se dá mediante uma condição/ação (a ação é um método a ser executado).

**A identidade** nada mais é que a identificação do objeto, pois todo objeto é identificável, por exemplo, cada animal tem suas características, raça, cor, nome, portanto, cada um tem sua identidade e são distinguíveis entre si.

Conseqüentemente, temos o conceito de **mensagem**. “Os objetos se comunicam através de mensagens, isto é, um sinal enviado de um objeto a outro requisitando um serviço através da execução de uma operação” (FURLAN, 1998). Essa análise corresponde ao conceito de **Polimorfismo**, “cuja palavra é originária do grego (muitas formas). Tais formas se referem aos vários comportamentos que uma mesma operação pode assumir, assim como à

capacidade de uma variável referir-se a objetos diferentes que preenchem certas responsabilidades dependendo da mensagem que lhes é passada” (FURLAN, 1998). Porém pode se dizer que o conceito de polimorfismo permite uma única interface para múltiplas funções, ou seja, a habilidade de duas ou mais classes responderem à mesma solicitação, cada uma no seu modo.

Outro conceito importante é o de **Classe**. Todos os objetos são instâncias de uma classe. As propriedades de qualquer instância (objeto) são determinadas pela descrição da classe à qual pertence.

“Uma classe é uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica” (JACOBSON, 2000).

Classes são utilizadas para compor o vocabulário do sistema que está sendo desenvolvido através da abstração de objetos que constituem o domínio do problema. Seu maior benefício seria a reutilização de código, pois estariam invisíveis aos usuários externos e os programadores apenas precisariam saber o que faz determinada função, além de uma excelente organização de classes.

Em UML, uma classe é representada graficamente por um retângulo dividido em três partes. A primeira contém o nome da classe, que é a sua identificação e deve ser escolhida de forma a compreender todos os objetos de acordo a esta classe. A segunda contém o conjunto de atributos da classe e a última compreende respectivamente as operações da classe, de acordo com a Figura 1.

Portanto, uma classe é uma coleção de vários de objetos. Cada objeto é diferente entre si, no qual possuem um estado específico, ou seja, como uma pessoa é diferente da outra, independente da sua interface.

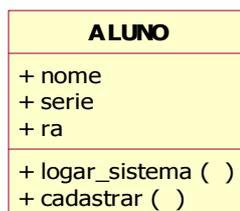


Figura 1 - Exemplo de Classe.

Uns dos principais benefícios de usar classes são os diversos recursos que a OO nos proporciona. Um deles é o **Encapsulamento**, que permite a segurança, na qual os objetos conheçam apenas seus dados, caso contrário, a comunicação entre os objetos é feita por métodos que possuem uma função conhecida como mensagem.

Encapsulamento nada mais é do que “a combinação de atributos e operações em uma classe” (FURLAN, 1998), ou seja, o empacotamento de dados e procedimentos dentro do objeto ou ocultamento de informações (oculta detalhes de implementações de um objeto). Caracteriza-se por dois módulos: o agrupamento do estado e operações do objeto, e ocultação dos dados.

Portanto, um objeto só permite o acesso a seus dados, mediante o acionamento de seus métodos, através de uma mensagem, para a qual, pode-se devolver uma resposta.

### 1.3 Diagramas da UML

A UML é apresentada pela notação definida pelos seus vários tipos de diagramas, que é a representação gráfica utilizada para se visualizar um sistema sob diferentes aspectos. A UML oferece nove diagramas e cinco níveis de abstração, efetuada pela OMG (MATOS, 2002).

Classificam-se também como Diagramas Estruturais (responsáveis por representar seu esqueleto e estrutura, relativamente estável e elementos do sistema, como atributos, entidades, regra de negócios, entre outros), Diagramas Comportamentais (responsáveis pela modelagem das interações com o sistema numa visão temporal) e Diagramas Arquiteturais

(que tratam o desenvolvimento de sistemas de grande porte em uma visão de pacotes, além de desenhar elementos de infra-estrutura de rede). Os principais diagramas da UML são apresentados nesta abordagem, em particular, o diagrama de caso de uso apresentado no estudo de caso do SEQ.

Em seus diagramas, de acordo com (BRAGA, 2001), existem elementos genéricos, que aparecem em qualquer um dos diagramas. Alguns deles devem ser citados:

- Anotação (Note): É um comentário inserido no diagrama;
- Pacotes (Packages): Agrupa elementos de modelagem, podendo conter classes, relacionamentos, classes abstratas, Packages, tipos, etc.
- Estereótipo (Stereotype): É um mecanismo de extensão introduzido pela UML, que permite que o usuário estenda a meta-modelo para suprir necessidades não encontradas entre os meta-elementos disponíveis, desde que a definição semântica da própria linguagem não seja ferida. Um estereótipo pode ser aplicado a classes, relacionamentos de dependência, atributos, operações, etc. Exemplos de estereótipos: <<extend> e <<include>>.

### **1.3.1 Diagrama de Caso de Uso – Use Case**

Classificado como Diagrama Comportamental e primeiro nível de abstração, o diagrama de caso de uso é importante para a modelagem de um sistema, por apresentar uma visão externa de como os elementos poderão ser utilizados dentro do contexto.

Diagramas de Caso de Uso são utilizados para identificar as regras de negócio e compreender o ponto de vista do usuário em relação ao sistema, entretanto, essa modelagem é utilizada para capturar necessidades de um novo sistema e desenvolver novas versões, adicionando novas funcionalidades, atores e casos.

Esse diagrama se caracteriza também pela técnica de modelar as funcionalidades do sistema, através de atores externos interagindo em casos de uso. Essas interações podem ser

do tipo de comunicação, onde um ator simplesmente se comunica com um caso de uso. Esses relacionamentos (interações) podem ser interligados através de associações e podem ter relacionamentos de generalização (atores se relacionando entre si) que descreve o conceito de herança estudada em diagrama de classes.

Além dos tipos de interações existentes para a modelagem do diagrama, que será implementada no Capítulo 3, usam-se também para o relacionamento os estereótipos “usa” (uses) e “extensão” (extends). O primeiro representa o comportamento comum entre dois ou mais casos de uso, ou seja, um determinado caso “usa” o outro, já o segundo representa o relacionamento opcional entre os casos de usos, por exemplo, um caso pode ou não utilizar o outro caso.

Portanto, um diagrama completo com todos casos de uso, respectivamente, com os atores, representa todas funcionalidades (situações) possíveis do sistema. Seus elementos são os próprios casos de uso, atores, sistema e interação existentes entre os dois primeiros elementos.

Tabela 2 - Elementos do diagrama de caso de uso

ELEMENTOS DO DIAGRAMA	DESCRIÇÃO
Caso de Uso	Descreve a seqüência de eventos que um ator faz no uso do sistema. São funcionalidades do sistema, ou seja, têm como objetivo decidir e fornecer uma descrição clara e consistente do que o sistema pode fazer. É iniciado por um ator e pode modelar diálogos entre atores, ou mesmo entre casos de uso.
Ator	Ator não é parte do sistema, tipicamente estimula (solicita) ações (eventos) do sistema e recebe reações. Cada ator pode participar de vários casos de uso, representando ações que os usuários do sistema podem desempenhar, podendo receber mensagens (informações) e interagir ativamente como sistema. Porém, pode ser um humano, máquina ou sistema.
Sistema	É o sistema a ser modelado.
Relacionamentos	Podem ser do tipo “comunicação”, “extensão” (extends) e “usa” (uses).

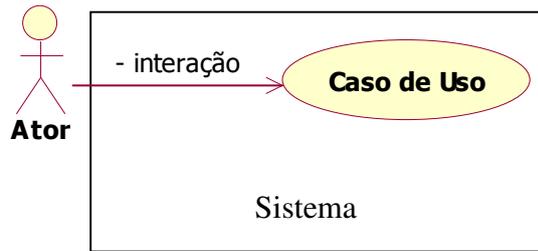


Figura 2 - Exemplo de Diagrama de Caso de Uso.

FURLAN (1998) descreve a modelagem de caso de uso como uma caixa-preta, fornecendo situações (casos de usos). Uma restrição bastante importante é o fato de que nesse momento de modelagem, não é necessário compreender como o sistema será implementado funcionalmente os casos, ou como ocorre o funcionamento interno do sistema.

### 1.3.2 Diagrama de Classe

Classificado como Diagrama Estrutural e segundo nível de abstração, o diagrama de classe trata-se de uma estrutura lógica estática, todavia, sempre válida para todo o ciclo de vida do sistema.

Esse diagrama é o mais utilizado em sistemas de modelagem OO, logo, é a essência da UML. É composto por um conjunto de classes relacionadas entre si.

“Uma classe captura a estrutura e o comportamento comum de um conjunto de objetos. É uma abstração de elementos do mundo real. Quando esses elementos existem no mundo real, são instâncias de classe e são referidos como objetos. Para cada classe que tem comportamento temporal significativo pode ser criado um diagrama de estados para descrever este comportamento” (FURLAN, 1998).

Um conceito chave para os relacionamentos entre as classes é a multiplicidade, que mostra a cardinalidade de uma associação, enfocando o modelo entidade relacionamento de banco de dados (veja Apêndices A e C).

A cardinalidade de um elemento mostra, por exemplo, quantos elementos químicos uma pessoa (objeto) pode pegar, ou seja, a multiplicidade especifica a cardinalidade que um conjunto pode assumir.

Na UML a multiplicidade é representada por: 1 representa exatamente 1, \* representa muitos (zero ou mais), 0..1 representa opcional (zero ou um) e m..n representa seqüência especificada. No qual a multiplicidade é um subconjunto infinito de inteiros e não negativos.

Já as visibilidades dos atributos e operações das classes são definidas como: + **visibilidade pública** (valor default), significa que todas as classes podem acessá-los, # **visibilidade protegida**, onde o atributo é acessado através de operações dentro da mesma classe e do pacote a ser definido referente a essa classe e – **visibilidade privada**, onde o atributo é privado a quaisquer outras classes, podendo ser acessado apenas por operações declaradas dentro da mesma classe. Entretanto, identificadores privados agem como identificadores públicos, exceto declarações locais e são desconhecidos e inacessíveis.

Podemos definir também a navegabilidade de uma classe, onde “Uma instância de uma classe pode negar a instância de outra classe e vice-versa” (FURLAN, 1998). É representada por uma seta aberta, indicando a navegação em direção à classe.

Neste diagrama, classes que possuem características em comum podem ser adicionadas em pacotes separados, com o objetivo de diminuir a complexidade do sistema e implementação, ou seja, para sistemas complexos, classes, além de ser definidas dentro de pacotes, são agrupadas para melhor entendimento em nível de modelistas.

O diagrama de classes é formado por quatro tipos básicos de relacionamentos: Generalização/ Especificação, Agregação, Associação e Dependência.

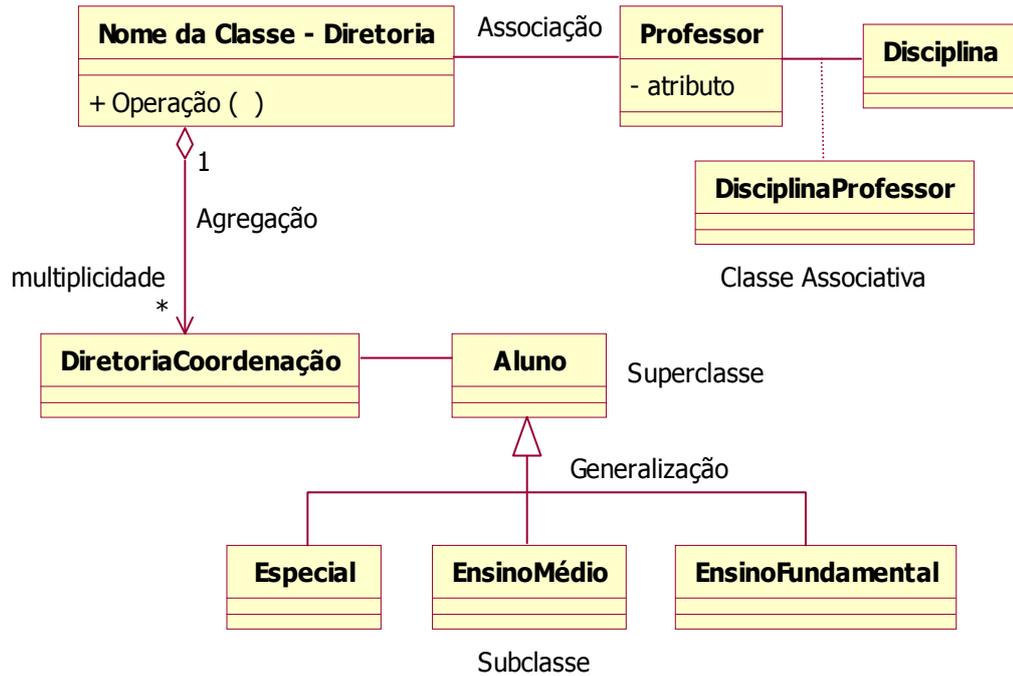


Figura 3 - Exemplo de Diagrama de Classes.

- **Generalização Especificação**

Uma generalização é um relacionamento hierárquico, no qual indica um relacionamento entre a classe de maior nível denominada superclasse e a outra denominada subclasse.

É enfatizada pelo conceito de herança, onde se distingue pela característica “de reutilização de atributos e operações definidos em classes gerais por classes mais específicas, podendo ser usada para expressar tanto generalização quanto associação” (FURLAN, 1998).

Portanto, as subclasses (classes mais especializadas) conhecidas também por classe filha ou derivada, herdam atributos, operações e associações da superclasse (também conhecida por classe pai ou classe base) e agregam atributos, operações particulares a subclasse que se referem.

Graficamente é representada por uma seta em branco apontando sempre a superclasse, sendo uma linha contínua, conforme Figura 3.

- **Agregação**

Um objeto é composto “em partes” de outro em relação de todo/parte e um objeto gerado da classe “todo” tem um objeto gerado da classe “parte”.

Graficamente é representada por uma linha contínua com um losango em uma das extremidades, sendo que esta, conecta-se com a classe “todo”, conforme Figura 4.

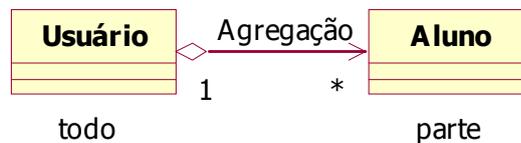


Figura 4 - Exemplo de Agregação entre Classes.

O objeto parte (Aluno) é um atributo do objeto todo (Usuário), porém, o objeto todo (Usuário) não pode ser destruído antes do objeto parte (Aluno), da mesma forma que o objeto parte (Aluno) não é criado a menos que o objeto todo (Usuário) seja criado, ao qual estão agregados.

- **Associação**

Associação ocorre quando uma ou duas classes apresentam interdependência, ou seja, quando uma instância de uma se associa ou unifica com uma ou mais instâncias de outra. “Na UML, uma associação é definida como um relacionamento que descreve um conjunto de vínculos, onde vínculo é definido como uma conexão semântica entre tuplas de objetos” (FURLAN, 1998).

As classes envolvidas se caracterizam em: unária quando há apenas uma classe relacionando consigo própria, binária quando há duas classes se relacionando com forma direta, e n-ária onde há associação entre três ou mais classes.

Quando a multiplicidade de associação for do tipo 0..1 – 0..1 ou 0..1 - \* ou \* - \* dão origem a uma nova classe denominada classes de associação.

Graficamente a associação entre duas classes é representada por uma linha contínua e a classe criada pela associação é mostrada com uma linha tracejada conectada na linha contínua e é mostrada na Figura 3.

- **Dependência**

É um relacionamento entre duas ou mais classes onde uma classe depende da outra. É representada graficamente por uma seta aberta, com linha tracejada, conforme é mostrada na Figura 5.



Figura 5 - Exemplo de Dependência entre classes.

A classe aluno é dependente de algum serviço da classe professor, mas não tem uma dependência estrutural interna com esse professor.

### 1.3.2.1 Diagrama de Objetos

Existe também o Diagrama de Objetos, também classificado como Estrutural e segundo nível de abstração, esse não é tão importante quanto o diagrama de classes, mas pode ser uma chave para exemplificar os diagrama de classes complexas.

Diagrama de Objetos é um diagrama de classes instanciado, ou seja, é uma variação do diagrama de classes e utiliza a mesma notação, além de mostrar um número de instâncias de classes, em vez de classes reais.

Suas principais características são: desenho do sistema em determinado instante, entendimento e validação do sistema, desenham os objetos e as ligações entre eles e não descreve o sistema como um todo.

### **1.3.3 Diagramas de Interações**

Diagramas de Interações estão classificados em Diagramas Comportamentais de terceiro nível de abstração, todavia, se caracteriza pela ferramenta de constatação ou validação usando diversas trocas de mensagens. Representar interações é compreender e modelar as entidades dinâmicas e saber como diversas operações permitirá que o que se deseja do sistema realmente seja executado. É composto por Diagrama de Seqüência e Diagrama de Colaboração.

Ambos os diagramas se comunicam ou interagem em um conjunto de objetos através de troca de mensagens. Essas são alguns tipos de operações ou tarefas pré-definidas.

O principal propósito para modelagem do diagrama de interação é a necessidade de visualizar o comportamento de vários objetos de um único caso de uso, a partir das mensagens que são passadas entre eles. Entretanto, para especificar essa interação, para ambos diagramas (seqüência ou colaboração), é necessário definir uma descrição de caso de uso estabelecendo os objetos que interagem e seus relacionamentos.

Portanto, “diagramas de interações são aplicados para mostrar a realização de casos de uso, e as possíveis seqüências de interação são especificadas em uma descrição única que contém condicionais (ramificações ou sinais condicionais) ou através de descrições múltiplas, cada uma detalhando um caminho particular pelos possíveis caminhos de realização” (FURLAN, 1998).

### 1.3.3.1 Diagrama de Seqüência

Diagrama de Seqüência é um Diagrama de Interação responsável por modelar aspectos dinâmicos para a fase de análise e, enfatiza o comportamento de objetos, incluindo suas operações, interações, colaborações e histórias de estado, dando ênfase à ordenação temporal das mensagens e operações trocadas por eles.

Para uma ótima modelagem sob diferentes níveis de abstração, é preciso de uma análise das descrições do diagrama de casos de usos. Essa análise é de suma importância e representará os passos, mensagens, descrições algorítmicas, interações e operações dos casos de uso.

Após essa análise, é necessário avaliar quais classes podem fazer parte desse caso de uso, podendo acontecer mudanças nos diagramas anteriores assim modelados, devido a alguns erros. A modelagem de um diagrama de seqüência depende no mínimo de uma idéia inicial de um diagrama de classes, pois para cada descrição de caso de uso, será escolhida uma classe que seja responsável por fazer a tarefa responsável à descrição.

Desta forma, para uma melhor visualização de fluxo de controle ao longo do tempo, o diagrama de seqüência organiza-se colocando os objetos como linhas verticais, correspondentes na parte superior do diagrama, ao longo do eixo horizontal; e as mensagens são representadas por setas horizontais, onde a ponta da seta identifica o objeto alvo. E as seqüências das mensagens são lidas de cima para baixo.

O Diagrama de Seqüência também se caracteriza pelas linhas de vida dos objetos que são linhas verticais tracejadas, as quais representam o tempo de existência do objeto, e um retângulo alto e estreito, que mostra o nome do objeto e dois pontos seguidos o nome da classe. De acordo com a Figura 6.

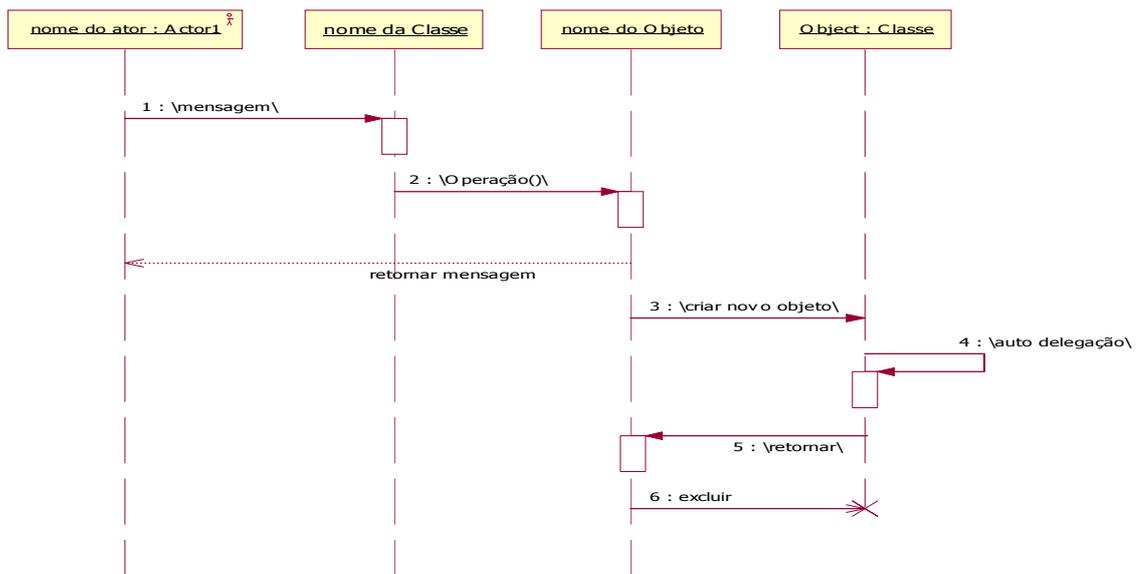


Figura 6 - Exemplo de Diagrama de Seqüência.

### 1.3.3.2 Diagrama de Colaboração

No estudo de caso do SEQ, foi utilizado o diagrama de seqüência por apresentar maior ênfase nos quesitos de casos de usos complexos, tornando mais fácil de ser visualizada a ordem na qual as mensagens acontecem (facilidade de interpretação do diagrama). Todavia, não foi enfatizado o diagrama de colaboração para este projeto.

“Diagrama de colaboração mostra o contexto completo de uma interação, inclusive os objetos e seus relacionamentos pertinentes a uma interação particular, sendo freqüentemente melhores para propósitos de desenho. Um diagrama de colaboração é um contexto, um gráfico de objetos e vínculos com fluxos de mensagens anexados” (FURLAN, 1998).

### 1.3.4 Diagrama de Estado e Atividade

Em se tratar de modelagem dinâmica, a UML é bastante vasta. Além dos Diagramas de Interações, os Diagramas de Estados e Atividades também possuem aspectos dinâmicos, entretanto, se caracterizam por serem Diagramas Comportamentais e pelo 4º nível de

abstração. Atividades e estados se completam e ambos partem de um mesmo princípio: máquinas de estados.

Uma máquina de estados representa a seqüência de estados que um objeto assume durante o seu tempo de vida, em resposta a eventos. Desta forma, uma máquina de estados modela o comportamento de um objeto individual.

Um dos objetivos principais de uma máquina de estado é estudar certos tipos de lógica que envolve transições possíveis entre diferentes estados, porém, pode ser analisada como uma caixa-preta que recebe um número finito de entradas (estímulos) e está disposta a tratá-las oferecendo resposta.

Os elementos de uma máquina de estado são: estados, saídas, transições, um estado inicial e um estado final, utilizando-se conceitos de grafos e autômatos.

O diagrama de estado possui conceitos distintos sobre o diagrama de atividades. As principais diferenças são que diagrama de estado enfoca no comportamento das instâncias, ou seja, ao longo de sua vida, são avaliadas as possíveis seqüências de ações. Já o diagrama de atividades é uma continuação ao conceito de máquinas de estados, avaliando melhor as condições pelas quais as instâncias chegam a determinadas decisões.

Outra diferença bastante visível é o fato de que em uma atividade, quando ocorre a ação, significa que será a finalização. Não há ações internas como ocorre em um estado, ou seja, uma atividade refere-se a qualquer coisa que está em andamento de execução, porém, não finalizado.

Quando uma atividade termina alguma ação ocorre. Já em estado refere-se a qualquer coisa que já foi atingido e o interesse seria o que ocorre quando ele é alcançado.

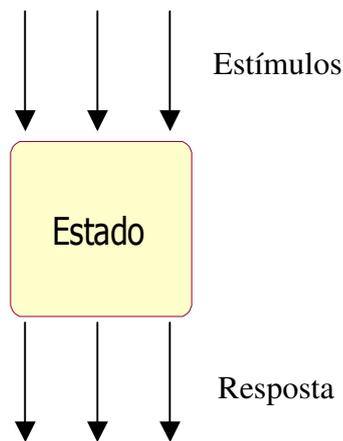


Figura 7 - Exemplo de Máquina de Estado (FURLAN, 1998).

### 1.3.4.1 Diagrama de Estado

Em um objeto encontra-se estados nos quais implicam na ordem em que as operações são executadas, portanto, é necessário analisarmos o ciclo de vida de instâncias de uma classe, uma vez que, apenas as ações oferecem esse recurso.

Diagrama de Estado de um sistema, conhecido também por diagrama de gráfico de estados, define todos os possíveis estados encontrados para sistemas grandes e complexos. Para evitar uma explosão de estados, a UML propõe uma maneira individualizada para cada classe, porém cada classe do sistema poderá conter um diagrama de estados.

O objetivo deste diagrama é modelar fluxos de controle de um estado para outro através de máquinas de estados, ou seja, modelar situações em que encontramos durante o ciclo de vida de instâncias de uma classe.

Podemos caracterizar um estado como sendo a condição ou situação onde uma instância se encontra e um evento como uma ocorrência significativa, quando o ocorre, uma instância deixará um estado e chegará ao outro, relacionando-se com uma transição. Essa, por sua vez, é um relacionamento entre dois estados onde uma instância no primeiro estado realiza determinadas ações e passa para o segundo estado quando um evento é acionado e os requisitos são satisfeitos.

Os elementos do diagrama de estados podem ser visto a seguir:

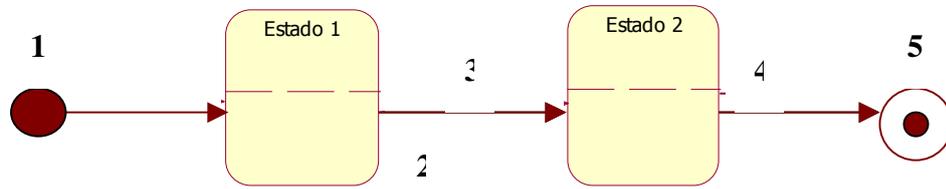


Figura 8 - Exemplo de Máquina de Estado (FURLAN, 1998).

1. Estado inicial: É representado por um círculo fechado em negrito. O objeto está à espera de algum evento que o transporta para o próximo estado.
2. Estado: O nome é representado por verbos no gerúndio e quando ocorrem ações e eventos, sua representação passa a possuir uma linha divisória para que esses possam ser definidos.
3. Transição: Pode ser uma expressão.
4. Chegada de um novo estado, não precisa provir de um único estado anterior.
5. Estado final: É representado por dois círculos: o menor fechado em negrito e o maior aberto sobre o menor. Este estado também não precisa ser atingido de um único estado de origem.

Quando estivermos modelando um estado e percebermos uma complexidade significativa, a ponto de se esconder às transições e ações ao entendimento do diagrama, ou a ponto dos estados iniciais e finais não estiverem bem definidos, será preciso a modelagem de um estado composto.

Mesmo não sendo obrigatório, para os problemas no desenvolvimento de um diagrama de estado apresentados neste contexto, um estado composto é fundamental. Este é

chamado também de estado aninhado, entretanto, são conjunções de vários subestados ou agrupamento de estados que colaboram para uma única ação.

Finalizando, uma característica importante de um estado composto é o fato de apresentar um estado inicial, pois é a forma de enxergarmos o que realmente é executado quando se entra neste estado.

#### **1.3.4.2 Diagrama de Atividade**

“O diagrama de atividade permite escolher a ordem pela qual as coisas devem ser feitas” (FURLAN, 1998), ou seja, é responsável pela modelagem do fluxo de controle entre as atividades ou fluxo seqüencial das atividades geradas por uma operação do sistema ou por um agente externo.

É importante ressaltar seu foco de modelagem, que permite manipular processos paralelos, utilizando os procedimentos de paralelismo que é de suma importância para a modelagem de negócio.

Outra observação é o fato de ser semelhante a um fluxograma, podendo até ser visto como fluxograma estendido, pelo entendimento de fluxo das operações, decisões e operações. Mas o diagrama de atividade obtém regras de seqüência para ser seguida, portanto, apresentam diferenças entre fluxograma e diagrama de atividade. Fluxogramas são limitados a processos seqüências, já o diagrama de atividade, trabalha com o paralelismo.

O diagrama de Atividade é composto por três notações: estados de atividades, estados de ação e transições, além do estado final e inicial apresentado na Figura 9.

O estados de atividades referem-se a uma execução normal de uma atividade com um evento, ativando uma transição de saída, entretanto, podem ser interrompidos e decompostos para serem representados mais detalhadamente.

Os estados de ação não podem ser interrompidos ou decompostos e representam um estado do sistema sob a execução de uma ação. Todavia, um estado de ação é um estado de atividade que não pode mais ser decomposto.

As transições são utilizadas para modelar um fluxo de controle de um estado (ação ou atividade) para o outro estado, ou seja, assim que o primeiro estado estiver concluído o fluxo de controle passará para o outro estado, obedecendo ou não a condição determinada. As transições podem ser: Seqüência simples e Ramificação.

A seqüência simples se caracteriza por uma linha com uma direção, pela qual liga dois estados representando um fluxo de controle entre eles. Já a ramificação é representada por um losango que define um símbolo de decisão, ou seja, caminhos alternativos que o modelista toma decisões. Pode ser uma transição de entrada e duas ou mais de saídas, obedecendo à ordem em que uma transição de saída pode ser tomada por vez.

Quando se tratar de fluxos concorrentes, serão necessárias as barras de sincronização, onde sua função é representar momentos em que ocorre uma bifurcação ou união de fluxos de controle. É representada por uma reta escura (horizontal ou vertical).

A bifurcação define um fluxo de controle paralelo e independente, é representada por uma única transição de entrada e duas ou mais de saída. A união define a sincronização dos fluxos de controle de entrada, entretanto, é representada por duas ou mais transições de entrada e uma única transição de saída.

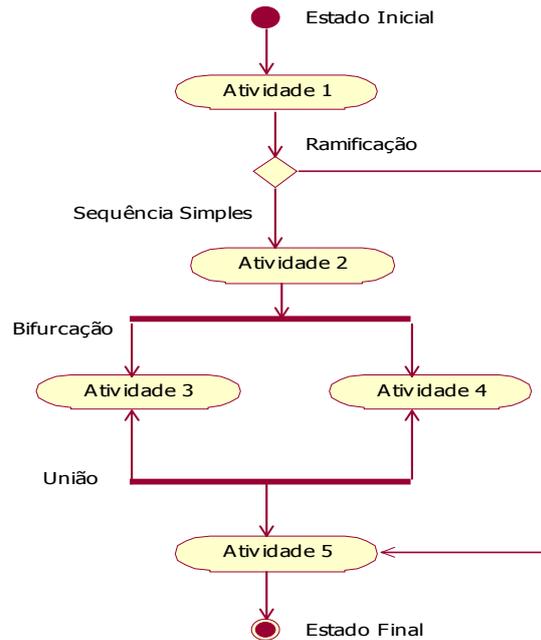


Figura 9 - Exemplo de Diagrama de Atividade.

Portanto, o objetivo de um modelista (analista ou projetista) atuar em um diagrama de atividade de acordo com (FURLAN, 1998), além de mostrar como se escreve um comportamento paralelo ou mostrar como interagem comportamentos em vários casos de usos, seria a necessidade de capturar alguns propósitos:

- Funcionamento interno de um objeto;
- Trabalho (ações) que será desempenhado quando uma operação é executada;
- Mostrar como um processo de negócio funciona em termos de atores, fluxos de trabalho, organização e objetos;
- Mostrar como uma instância de caso de uso pode ser realizada em termos de ações e mudanças de estado de objetos;
- Mostrar como um conjunto de ações relacionadas pode ser executado como afetará objetos ao redor.

Para concluir esses propósitos, o modelista poderá focar uma classe, uma implementação de operação ou um caso de uso. Para esse último, primeiramente precisará

fazer uma análise dos casos de usos, para compreender quais ações precisam ser realizadas e quais são as dependências comportamentais. Também uma análise aos fluxos entre os caso de usos, onde o diagrama de atividade representa essas interações com o propósito de compreender esses comportamentos.

### **1.3.5 Diagrama de Implementação**

Os Diagramas de Implementações se caracterizam pelo 5º nível de abstração e por serem Diagramas Arquiteturais. Eles são de suma importância para os analistas e programadores modelarem, pois possuem grandes características como o reuso de código e projeto. Todavia, a execução de um reuso de código vem se originando desde do primeiro objeto criado até as mais complexas unidades de código (MATOS, 2002).

Esses diagramas são denominados de duas formas:

- Diagrama de componentes: Caracteriza-se pela apresentação da estrutura do código fonte;
- Diagrama de implantação: Caracteriza-se pela apresentação da estrutura do sistema de run-time.

#### **1.3.5.1 Diagrama de Componente**

“Um diagrama de Componente é um gráfico de componentes conectados por relacionamentos de dependência onde também podem ser associados componentes a outros por retenção física que representa relacionamentos de composição. Exibe as organizações e dependências entre componentes com o propósito de modelar a visão de implementação dos módulos de software executáveis com identidade e interface bem-definida de um sistema e seus relacionamentos” (FURLAN, 1998).

A grande vantagem de ser modelado é a facilidade do reuso de código, com idéia básica de que componentes de software sejam projetados e implementados de forma que possam ser reusados em muitos sistemas diferentes (BRAGA, 2001).

Este diagrama se caracteriza por ser de alto nível, porém, pode apresentar as dependências entre componentes de softwares: componentes executáveis (programas .exe); componentes de código binário ( bibliotecas e interfaces .dll em VB ou .class em Java); componentes e código (.h em C++ ou .Java em Java); e documentação, classes, tabelas de banco de dados e arquivos complementares.

De acordo com (MATOS, 2002), a organização de um sistema é dependente da solução adotada, ou seja, a linguagem de programação usada sempre refletirá neste diagrama. Todavia, se destina a apresentar uma forma de tipo, caso o sistema necessite de apresentar a instância desses componentes, são necessários à modelagem do diagrama de implantação.

As dependências dos relacionamentos entre componentes e interfaces que são assinaturas de métodos, são representadas com setas tracejadas de componentes para interfaces em outros componentes. Ressaltamos também a possibilidade de usar estereótipos das dependências, esses são específicos da linguagem e ajuda ao modelista compreender e documentar melhor o diagrama e também a possibilidade de se referir ao tipo do elemento (fonte, binário, executável, etc...).

De acordo com a Figura 10, um componente é uma unidade de código e é desenhado como um retângulo maior e dois menores em seu lado esquerdo. Porém seu nome e tipo são apresentados dentro, em cima ou abaixo dele.

Nome-do-componente ":" tipo-de-componente



Figura 10 - Exemplo de Componente.

### 1.3.5.2 Diagrama de Implantação

Diagrama de implantação é frequentemente modelado em sistemas de tempo reais, porém, além do desenvolvimento do software, é necessário o desenvolvimento dos componentes de hardware.

Esses diagramas, de acordo com o (FURLAN, 1998), se caracterizam pelo fato de apresentarem um gráfico de nós conectados por associações indicando um caminho de comunicação. A associação pode ter um estereótipo para indicar a natureza do caminho.

Na UML 'nós' são objetos físicos de run-time (veja Figura 11), ou seja, são elementos de hardware em que componentes de software possam ser executados. Estes componentes podem conter instâncias, que por sua vez, podem conter classes, bibliotecas ou executáveis.

Portanto, a modelagem deste diagrama para o SEQ não foi interessante pelo fato desse último não necessitar de recursos computacionais em tempo real e também por apresentar apenas um nó, o próprio PC, onde o sistema é instalado e executado. Mas é bastante vantajoso para sistemas onde necessitam de proteção de dados, portabilidade de implementação, acesso a dispositivos ou sistemas cliente-servidor, distribuídos ou que utilizam sensores.

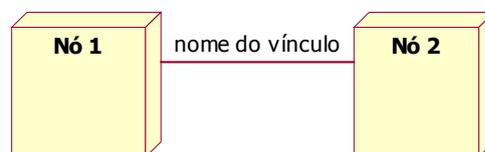


Figura 11 - Exemplo de Diagrama de Implantação (FURLAN, 1998).

## **CAPÍTULO 2 - ANÁLISE E ESPECIFICAÇÃO DOS REQUISITOS**

Neste capítulo foram identificados e gerenciados os processos do SEQ dentro dos conceitos da Engenharia de Software, enfatizando a análise de requisitos. Esta bem desenvolvida, ou seja, com a concepção e compreensão dos requisitos do software é fundamental para um bem-sucedido desenvolvimento do mesmo. Esta análise, de acordo com (PRESSMAN, 1995), proporciona aos desenvolvedores e clientes os critérios para avaliar a qualidade logo que o software for construído.

Focamos o papel de Analista de Sistema, no qual se destaca pela facilidade de compreender todo ambiente de desenvolvimento do software, como: compreender conceitos abstratos; reorganizá-los em divisões lógicas e sintetizar “soluções” baseadas em cada divisão; entender o ambiente usuário/clientes; etc.

O propósito deste capítulo é esboçar a análise de forma objetiva e simples, descrevendo e criando no contexto do SEQ todos seus requisitos, de acordo com um estudo aprofundado. Após sua elaboração, conseqüentemente foi executada, gerando um estudo de caso apresentado no Capítulo 3.

O ciclo de vida foi definido nesta etapa, onde a análise e especificação de requisitos representam a modelagem de negócios, apresentado na seção 2.6.

Foram apresentados também todos os requisitos do sistema, porém, uma análise detalhada sobre as necessidades que o SEQ apresenta em relação a todos os usuários.

Os requisitos que se referem à fase de concepção são de suma importância para uma boa qualidade de software, porém se distingue por quatro vantagens fundamentais:

- Um requisito bem controlado reduz o risco de violação de prazo, pois a identificação de erros no início do desenvolvimento, permite que os desenvolvedores administrem o tempo;

- Um requisito bem administrado ajuda a encontrar e controlar erros de modelagem, podendo assim reutilizá-lo;
- Um requisito bem controlado ajuda os desenvolvedores a adiantar o seu trabalho, ou seja, maior desempenho em menor tempo;
- Um requisito bem administrado contém reuniões e revisões técnicas fechadas (formais), nos quais podemos estudar e modelar requisitos ao mesmo tempo em que os revisamos, e conseqüentemente corrigir erros, garantindo também a qualidade futura do software.

O SEQ classifica-se por ser sistema especialista, portanto, necessitou de informações e experiências de um especialista da área, que neste caso, foi um professor de química, com a finalidade de chegarmos a uma base e regras de informações sobre conceitos de química e diretrizes de um software educacional. Para uma melhor qualidade interagimos também com outros possíveis usuários do SEQ. Para isso é importante a necessidade de reuniões e revisões técnicas, essa última podendo armazenar as informações obtidas para um reuso de análise, projeto, implementação e teste do software.

Pressmam (1995) descreve reuniões e revisões como sendo uma técnica simples e direta de interação entre o entrevistado, que deve ser um especialista do conhecimento. Participam também o cliente ou usuário e o entrevistador, que é o engenheiro de requisitos (analista, projetista ou líder da revisão), buscando revelar conceitos, objetivos, e a organização e planejamento do que deve ser revisado (domínio do problema), além de buscar soluções ou projeções de soluções. Seus maiores objetivos são: instituir a idéia do trabalho em equipe; descobrir erros de função, lógica ou implementação em qualquer ciclo de vida do software; verificar se o sistema atende a seus requisitos; garantir que o sistema tenha sido

representado de acordo com padrões predefinidos; obter um software que seja desenvolvido uniformemente e tornar os projetos administráveis.

Neste contexto, foram utilizadas entrevistas abertas (informais), nas quais as necessidades e o conhecimento do domínio de aplicação são discutidos abertamente, sem um conjunto pré-definido de perguntas e fechadas (formais), destacando-se pela preparação prévia de perguntas feitas pelo entrevistador, além de sua formalização como: restrições de quantidade de pessoas envolvidas na revisão; preparação antecipada das perguntas, não devendo exigir mais que duas horas; a duração da revisão deve ser inferior a duas horas; e diversas outras técnicas como a walkthrough. As três principais entrevistas fechadas (reuniões e reunião da revisão) serão apresentadas em Apêndice E e os resultados obtidos (alterações dos problemas) são apresentados neste capítulo.

Em UML o diagrama chave para essa etapa de concepção do sistema é o caso de uso, onde os requisitos funcionais são expressos nos próprios casos de usos. Para os requisitos mais refinados, foram modelados os diagramas de seqüência e atividade, entretanto, o estudo desta fase só se encerrou quando representados quase todos os requisitos funcionais do sistema, de maneira que os futuros usuários e desenvolvedores possam compreender, todavia, os requisitos ainda não foram implementados, só modelados.

Contudo, esta fase retrata o modelo de casos de uso, que por sua vez, é a mais adequada ferramenta utilizada para organizar os requisitos em uma forma fácil de gerenciar. Para os usuários do SEQ, esses devem entendê-lo e executá-lo em uma forma consistente e não redundante. Para os desenvolvedores, esses podem analisar o estudo de caso, que se encontram no Capítulo 3, para que possam desenvolver as etapas seguintes: implementação e teste.

Este capítulo apenas enfocará o esboço da Análise de Requisitos, os diagramas relacionados e ele, serão apresentados no Capítulo 3 em Estudo de Caso.

## 2.1 Objetivo do Software Educacional de Química

Este trabalho tem como objetivo principal a proposição de um processo de desenvolvimento para apoiar a elaboração de um software educacional em Realidade Virtual. Especificado em conceitos de Química do ensino médio, baseia-se nos aspectos educacionais do modelo Construtivista, de forma que o aluno consiga a visualização de ligações químicas que não poderiam ser visualizadas a olho nu.

Este software possui diversos aspectos computacionais, neste caso, o metodológico, explorando o processo de aprendizagem, cujo objetivo é fazer com que os alunos analisem a respeito das ligações químicas existentes e os compostos que poderiam criar em um ambiente de laboratório.

De acordo com a metodologia construtivista, este software ajudará o aluno a um aprendizado ativo, ou seja, a interação com os objetos e ambientes do software, desenvolverá uma lógica mental e por fim o seu aprendizado. Foram ampliadas diversas etapas onde o aluno poderá criar o seu próprio composto (de acordo com as ligações químicas), tendo uma avaliação se este composto estará certo ou errado, podendo assim futuramente o professor analisá-lo e corrigi-lo. Os usuários finais do sistema foram definidos como: Aluno, Professor e Administrador do Sistema. Este último foi criado com a finalidade de controle do sistema.

As novas tecnologias prometem uma mudança qualitativa na forma que pode ocorrer a aprendizagem. Após a finalização da implementação do SEQ, será utilizada a inclusão digital, em colégios reais, variando estratégias para o desenvolvimento do mesmo. Porém, uma dessas estratégias, é o fato de que a realidade virtual oferece ao aluno a flexibilidade, personalização, interatividade e qualidade no ensino, além de admitir a concentração nas habilidades e prática do mesmo e não na memorização das tarefas.

O SEQ em seu término permitirá aos alunos selecionar e desenvolver suas próprias estratégias, todavia, eles serão encorajados a buscar novos domínios de conhecimento.

## 2.2 Informações Sobre o Sistema

Como referência, as reuniões e revisões técnicas (ver Apêndice E), são os marcos iniciais para podermos enfatizar este subcapítulo, pois representam assuntos das funcionalidades do SEQ. Para chegarmos a esses objetivos e compreensão, estudamos e apresentamos os conceitos da tabela periódica e ligações químicas.

Para os conceitos e definições utilizadas neste subcapítulo empregamos dois levantamentos bibliográficos: (CANTO, 2002); (REIS, 2005).

### 2.2.1 Tabela Periódica

Para abranger o estudo de química referente à tabela periódica, que foi utilizado para o desenvolvimento deste SEQ, primeiramente conceituamos resumidamente os átomos, as moléculas, os elementos químicos e as substâncias puras.

A palavra **Átomos** vem do grego e significa “indivisível”. Entre 1803 e 1808, o cientista inglês John Dalton propôs uma teoria para explicar as leis enunciadas por Lavoisier e Proust. Essa teoria foi conhecida como Atômica de Dalton que serão apresentados alguns tópicos seguir (CANTO, 2002).

- Toda as substâncias são formadas por átomos.
- Os átomos de um mesmo elemento químico são iguais em todas as suas características (por exemplo, tamanho e massa).
- Os átomos dos diferentes elementos químicos são diferentes entre si;
- As substâncias simples são formadas por átomos de um mesmo elemento químico;
- As substâncias compostas (também chamadas compostos químicos, ou compostos) são formadas por átomos de dois ou mais elementos químicos diferentes, que combinam sempre numa mesma proporção;

- Para o ensino médio, os átomos são considerados esferas rígidas indivisíveis que não podem ser criados e nem destruídos;
- Nas reações químicas os átomos se recombinaem.

**As moléculas** são as menores unidades que apresentam a composição característica de uma substância. Elas são formadas pela união de dois ou mais átomos. Por exemplo:

A fórmula  $O_2$  representa a substância cujas moléculas são formadas por dois átomos do elemento químico oxigênio.

A fórmula  $O_3$  representa a substância cujas moléculas são formadas por três átomos do elemento químico oxigênio.

A fórmula  $H_2O$  representa a substância cujas moléculas são formadas por dois átomos do elemento químicos hidrogênio e um átomo do elemento químico oxigênio.

**O Elemento Químico** é um conjunto de átomos que possuem características semelhantes, ou seja, tem o mesmo número atômico. Cada um deles tem um nome e um símbolo diferente.

Outro conceito importante é a **substância pura** onde se caracteriza por simples e composta. As substâncias simples são formadas por átomos de apenas um elemento e as substâncias compostas, por átomos de dois ou mais elementos, conforme salienta a Figura 12.

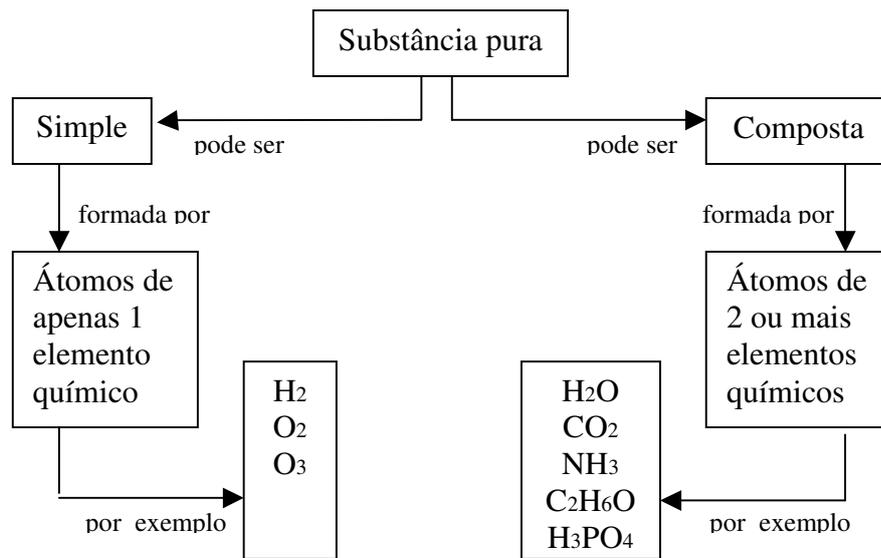


Figura 12 - Distinção entre Elemento e de Substância Pura (CANTO, 2002).

Para fazerem as ligações químicas, todos usuários do sistema irão utilizar como base à tabela periódica.

H																	He
Li	Be											B	C	N	O	F	Ne
Na	Mg											Al	Si	P	S	Cl	Ar
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe
Cs	Ba	*	Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn
Fr	Ra	**	Rf	Db	Sg	Bh	Hs	Mt	Uun	Uuu	Uub						
		*	La	Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu
		**	Ac	Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	No	Lr

Figura 13 - Tabela Periódica (REIS, 2005).

A tabela periódica é composta por 114 elementos químicos, dispostos em 18 colunas que se agrupam em uma família ou grupo. As famílias são numeradas sequencialmente da

esquerda para a direita. Os elementos são dispostos, um em cada quadrado, em ordem crescente de número atômico. Cada quadrado contém o símbolo e nome do elemento, número atômico e massa atômica.

A primeira família 1 ou IA, é as dos metais alcalinos. Seu nome decorre do fato de se formar substância alcalinas quando alguns dos elementos que a compõe reagem com a água (lítio, sódio e potássio).

OBS: O Hidrogênio está posicionado na família 1 devido a sua distribuição eletrônica. Mesmo estando no topo, ele não apresenta características dos elementos dessa família, nem de qualquer outra, por isso não faz parte de nenhuma delas.

A distribuição eletrônica por subníveis de energia de cada elemento da família 1, possui na ultima camada eletrônica, apenas 1 elétron no orbital s.

1	H	$1s^1$
	Li	$1s^2 2s^1$
	Na	$1s^2 2s^2 2p^6 3s^1$
	K	$1s^2 2s^2 2p^6 3s^2 3p^6 4s^1$
	Rb	$1s^2 2s^2 2p^6 3s^2 3p^6 4s^2 3d^{10} 4p^6 5s^1$
	Cs	$1s^2 2s^2 2p^6 3s^2 3p^6 4s^2 3d^{10} 4p^6 5s^2 4d^{10} 5p^6 6s^1$
	Fr	$1s^2 2s^2 2p^6 3s^2 3p^6 4s^2 3d^{10} 4p^6 5s^2 4d^{10} 5p^6 6s^2 4f^{14} 5d^{10} 6p^6 7s^1$

Figura 14 - Distribuição Eletrônica - Família 1 ou IA (REIS, 2005).

A família 2 ou IIA, é as dos metais alcalinos-terrosos. O termo alcalino se deve aos seus principais elementos encontrar-se na forma de óxido que, ao reagirem com a água produz substâncias alcalinas. Já o termo terroso (derivado de terra), designa qualquer substância insolúvel em água e estável a altas temperaturas, geralmente óxidos.

Sobre a distribuição eletrônica por subníveis de energia de cada elemento, todos eles possuem, na última camada eletrônica, 2 elétrons no orbital s.

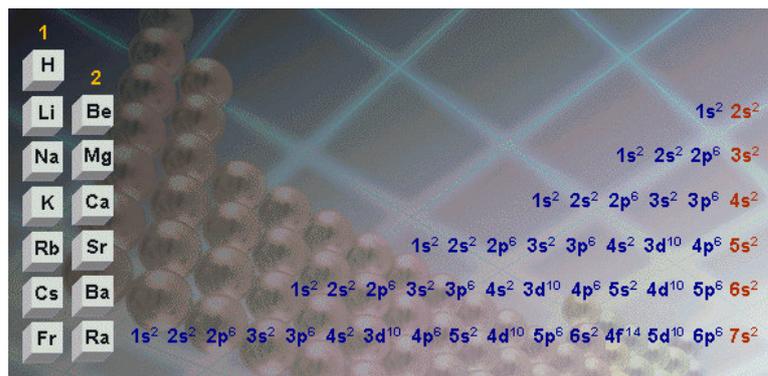


Figura 15 - Distribuição Eletrônica - Família 2 ou IIA (REIS, 2005).

As demais famílias de elementos químicos terão o mesmo tipo de regularidade em relação ao último subnível de suas distribuições eletrônicas, ou seja: A família 13, IIIA ou família do boro; a família 14, IVA ou família do carbono e a família 15, VA ou família do nitrogênio.

A família 16, VIA ou família dos calcogênios, cujo nome significa “formadores de cobre” e a família 17, VIIA ou família halogênios, onde seu nome significa “formadores de sais”.

A família 18, 0 (zero) ou família dos gases nobres, cujo nome reflete a pequena tendência desses elementos em particular de reações químicas.

Essas oito famílias apresentam um comportamento químico relativamente de menor complexidade que os demais e são denominados de elementos representativos e diferem do ponto de vista das distribuições eletrônicas.

1	2															18	
1s <sup>1</sup>																1s <sup>1</sup>	
2s <sup>1</sup>	2s <sup>2</sup>											2s <sup>2</sup> 2p <sup>1</sup>	2s <sup>2</sup> 2p <sup>2</sup>	2s <sup>2</sup> 2p <sup>3</sup>	2s <sup>2</sup> 2p <sup>4</sup>	2s <sup>2</sup> 2p <sup>5</sup>	2s <sup>2</sup> 2p <sup>6</sup>
3s <sup>1</sup>	3s <sup>2</sup>											3s <sup>2</sup> 2p <sup>1</sup>	3s <sup>2</sup> 2p <sup>2</sup>	3s <sup>2</sup> 2p <sup>3</sup>	3s <sup>2</sup> 2p <sup>4</sup>	3s <sup>2</sup> 2p <sup>5</sup>	3s <sup>2</sup> 2p <sup>6</sup>
4s <sup>1</sup>	4s <sup>2</sup>	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	4s <sup>2</sup> 4p <sup>1</sup>	4s <sup>2</sup> 4p <sup>2</sup>	4s <sup>2</sup> 4p <sup>3</sup>	4s <sup>2</sup> 4p <sup>4</sup>	4s <sup>2</sup> 4p <sup>5</sup>	4s <sup>2</sup> 4p <sup>6</sup>
5s <sup>1</sup>	5s <sup>2</sup>	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	5s <sup>2</sup> 5p <sup>1</sup>	5s <sup>2</sup> 5p <sup>2</sup>	5s <sup>2</sup> 5p <sup>3</sup>	5s <sup>2</sup> 5p <sup>4</sup>	5s <sup>2</sup> 5p <sup>5</sup>	5s <sup>2</sup> 5p <sup>6</sup>
6s <sup>1</sup>	6s <sup>2</sup>	*	Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	6s <sup>2</sup> 6p <sup>1</sup>	6s <sup>2</sup> 6p <sup>2</sup>	6s <sup>2</sup> 6p <sup>3</sup>	6s <sup>2</sup> 6p <sup>4</sup>	6s <sup>2</sup> 6p <sup>5</sup>	6s <sup>2</sup> 6p <sup>6</sup>
7s <sup>1</sup>	7s <sup>2</sup>	**	Rf	Db	Sg	Bh	Hs	Mt	Uun	Uuu	Uub						

Figura 16 - Apresentação do subnível mais energético - Elementos Representativos (REIS, 2005).

As demais famílias contêm os elementos cuja distribuição eletrônica difere na penúltima camada, sendo por isso chamados de elementos de transição. Com algumas exceções, eles seguem o mesmo padrão, observado para os elementos representativos, quanto à distribuição eletrônica. São as seguintes famílias: Família 3, ou IIIB, família 4 ou IVB, família 5 ou VB, família 6 ou VIB, família 7 ou VIIB, família 8 ou VIIIB, família 9 ou VIIIB, família 10 ou VIIIB, família 11 ou IB e a família 12 ou IIB.

Nem todos os elementos seguem a distribuição eletrônica prevista para suas famílias devido a um favorecimento energético que torna suas configurações elétrons mais estáveis que aquelas previstas teoricamente. São esses com o retorno dos quadradinhos em vermelho.

1	2																18
1s <sup>1</sup>																	1s <sup>1</sup>
2s <sup>1</sup>	2s <sup>2</sup>											2s <sup>2</sup> 2p <sup>1</sup>	2s <sup>2</sup> 2p <sup>2</sup>	2s <sup>2</sup> 2p <sup>3</sup>	2s <sup>2</sup> 2p <sup>4</sup>	2s <sup>2</sup> 2p <sup>5</sup>	2s <sup>2</sup> 2p <sup>6</sup>
3s <sup>1</sup>	3s <sup>2</sup>											3s <sup>2</sup> 2p <sup>1</sup>	3s <sup>2</sup> 2p <sup>2</sup>	3s <sup>2</sup> 2p <sup>3</sup>	3s <sup>2</sup> 2p <sup>4</sup>	3s <sup>2</sup> 2p <sup>5</sup>	3s <sup>2</sup> 2p <sup>6</sup>
4s <sup>1</sup>	4s <sup>2</sup>	3d <sup>1</sup> 4s <sup>2</sup>	3d <sup>2</sup> 4s <sup>2</sup>	3d <sup>3</sup> 4s <sup>2</sup>	3d <sup>5</sup> 4s <sup>1</sup>	3d <sup>5</sup> 4s <sup>2</sup>	3d <sup>6</sup> 4s <sup>2</sup>	3d <sup>7</sup> 4s <sup>2</sup>	3d <sup>8</sup> 4s <sup>2</sup>	3d <sup>9</sup> 4s <sup>1</sup>	3d <sup>10</sup> 4s <sup>2</sup>	4s <sup>2</sup> 4p <sup>1</sup>	4s <sup>2</sup> 4p <sup>2</sup>	4s <sup>2</sup> 4p <sup>3</sup>	4s <sup>2</sup> 4p <sup>4</sup>	4s <sup>2</sup> 4p <sup>5</sup>	4s <sup>2</sup> 4p <sup>6</sup>
5s <sup>1</sup>	5s <sup>2</sup>	4d <sup>1</sup> 5s <sup>2</sup>	4d <sup>2</sup> 5s <sup>2</sup>	4d <sup>4</sup> 5s <sup>1</sup>	4d <sup>5</sup> 5s <sup>1</sup>	4d <sup>5</sup> 5s <sup>2</sup>	4d <sup>7</sup> 5s <sup>1</sup>	4d <sup>8</sup> 5s <sup>1</sup>	4d <sup>10</sup> 5s <sup>1</sup>	4d <sup>10</sup> 5s <sup>2</sup>	5s <sup>2</sup> 5p <sup>1</sup>	5s <sup>2</sup> 5p <sup>2</sup>	5s <sup>2</sup> 5p <sup>3</sup>	5s <sup>2</sup> 5p <sup>4</sup>	5s <sup>2</sup> 5p <sup>5</sup>	5s <sup>2</sup> 5p <sup>6</sup>	
6s <sup>1</sup>	6s <sup>2</sup>	5d <sup>1</sup> 6s <sup>2</sup>	5d <sup>2</sup> 6s <sup>2</sup>	5d <sup>3</sup> 6s <sup>2</sup>	5d <sup>4</sup> 6s <sup>2</sup>	5d <sup>5</sup> 6s <sup>2</sup>	5d <sup>6</sup> 6s <sup>2</sup>	5d <sup>7</sup> 6s <sup>2</sup>	5d <sup>8</sup> 6s <sup>1</sup>	5d <sup>10</sup> 6s <sup>1</sup>	5d <sup>10</sup> 6s <sup>2</sup>	6s <sup>2</sup> 6p <sup>1</sup>	6s <sup>2</sup> 6p <sup>2</sup>	6s <sup>2</sup> 6p <sup>3</sup>	6s <sup>2</sup> 6p <sup>4</sup>	6s <sup>2</sup> 6p <sup>5</sup>	6s <sup>2</sup> 6p <sup>6</sup>
7s <sup>1</sup>	7s <sup>2</sup>	6d <sup>1</sup> 7s <sup>2</sup>	6d <sup>2</sup> 7s <sup>2</sup>	6d <sup>3</sup> 7s <sup>2</sup>	6d <sup>4</sup> 7s <sup>2</sup>	6d <sup>5</sup> 7s <sup>2</sup>	6d <sup>6</sup> 7s <sup>2</sup>	6d <sup>7</sup> 7s <sup>2</sup>	6d <sup>8</sup> 7s <sup>1</sup>	6d <sup>10</sup> 7s <sup>1</sup>	6d <sup>10</sup> 7s <sup>2</sup>						

Figura 17 - Apresentação do subnível mais energético (escrito em azul) e do mais externo (escrito em preto) - Elementos de Transição (REIS, 2005).

Existe ainda um grupo que, por suas características, são separados do resto da tabela formando duas séries de elementos, a série lantanídeos e actinídeos.

La	Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu
Ac	Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	No	Lr

Figura 18 - Série lantanídeos e Série actinídeos (REIS, 2005).

Essas séries de elementos também apresentam regularidades em suas distribuições eletrônicas, porém em camadas mais internas da eletrosfera, sendo por isso chamados de elementos de transição interna.

$4f^2 6s^2$	$4f^3 6s^2$	$4f^4 6s^2$	$4f^5 6s^2$	$4f^6 6s^2$	$4f^7 6s^2$	$4f^7 5d^1 6s^2$	$4f^9 6s^2$	$4f^{10} 6s^2$	$4f^{11} 6s^2$	$4f^{12} 6s^2$	$4f^{13} 6s^2$	$4f^{14} 6s^2$	$4f^{14} 5d^1 6s^2$
$6d^2 7s^2$	$5f^2 6d^1 7s^2$	$5f^3 6d^1 7s^2$	$5f^7 6d^1 7s^2$	$5f^6 7s^2$	$5f^7 7s^2$	$5f^4 6d^1 7s^2$	$5f^9 7s^2$	$5f^{10} 7s^2$	$5f^{11} 7s^2$	$5f^{12} 7s^2$	$5f^{13} 7s^2$	$5f^{14} 7s^2$	$5f^{14} 6d^1 7s^2$

Figura 19 - Apresentação do subnível mais energético - Série lantanídeos e Série actinídeos (REIS, 2005).

### 2.2.2 Ligações Químicas

Com base nas propriedades das substâncias simples formadas pelos elementos químicos é possível classificar esses elementos em três grandes grupos: os metais, os não metais e os semimetais.

Analisando a tabela periódica de acordo com a Figura 13, a cor rosa representa os elementos do tipo metais, a cor verde os não-metais, a cor azul os não-metais (gases nobres) e a cor amarela os semimetais. O hidrogênio é classificado por não-metais.

Com a atualização do estudo da química, foram renomeados os semimetais da seguinte forma: os elementos germânio, antimônio e polônio passarão a ser metais e os elementos boro, silício, arsênio e telúrio serão os não metais.

Foram utilizados três tipos de ligações interatômicas: Ligação iônica, Ligação covalente e ligação metálica:

- **A ligação iônica** tem sua composição de elementos metal e não metal onde são geradas substâncias iônicas ou composto iônico (não formados por moléculas).
- **A ligação covalente** tem sua composição apenas de elementos do tipo não metal onde são geradas substâncias moleculares ou composto molecular.
- **A ligação metálica** tem sua composição apenas de elementos do tipo metal onde são geradas substâncias metálicas ou composto metálico (não formados por moléculas).

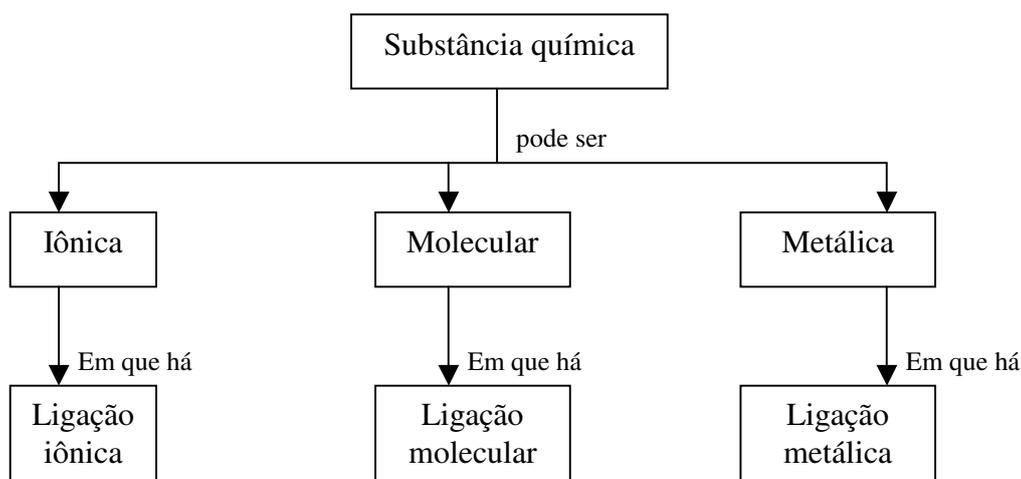


Figura 20 - Tipo de Substância Química (CANTO, 2002).

Para abrangermos melhor os tipos de ligações, em seguida estão as definições de cada ligações e suas características usadas para o desenvolvimento deste SEQ:

- **Os Gases Nobres:** São caracterizados por seis elementos nas quais existem átomos não-combinados. Essas substâncias são hélio, neônio, argônio, criptônio, xenônio e radônio (He, Ne, Ar, Kr, Xe e Rn), estão na família 18 ou 0 (zero). Estes gases estão presentes em pequenas quantidades na atmosfera terrestre, são formados por átomos não-combinados e sabe-se que até hoje não foi descoberta uma substância natural na qual os gases nobres estejam combinados entre si ou com átomos de outros elementos.
- **A Regra de Octeto:** Será de fácil interpretação se for analisada a distribuição eletrônica dos gases nobres.

Tabela 3 - Distribuição eletrônica dos gases nobres (CANTO, 2002).

Gás Nobre	K	L	M	N	O	P
Hélio	2					
Neônio	2	8				
Argônio	2	8	8			
Criptônio	2	8	18	8		
Xenônio	2	8	18	18	8	
Radônio	2	8	18	32	18	8

A eletrosfera é a parte mais externa dos átomos e o núcleo é muito pequeno, sendo assim, é ela que atua na combinação dos átomos. E já que os gases nobres não tendem a se combinar, sua eletrosfera permite aos átomos estabilizarem-se.

Diferente dos outros elementos químicos os gases nobres apresentam a última camada contendo oito elétrons, com exceção do hélio que contém a última camada contendo dois. Portanto possuir oito elétrons na última camada (ou dois, caso seja a camada K) faz com que o átomo fique estável. Os átomos não-estáveis se unem uns aos outros a fim de adquirir essa configuração de estabilidade.

OBS: A configuração eletrônica com a camada da valência completa é chamada configuração estável. Os átomos dos gases nobres são os únicos que já têm a camada da valência completa.

Contudo os átomos dos elementos ligam-se uns aos outros na tentativa de completar a camada da valência de seus átomos. Isso pode ser conseguido de diversas maneiras, dando origem a diversos tipos de ligações químicas.

- **A camada de valência:** É a camada mais externa do átomo de um elemento químico, é envolvida diretamente no estabelecimento de união com outros átomos. Os elétrons dessa camada são chamados de elétrons na camada de valência.
- **Ligação Iônica:** Formados pelos elementos metais e não-metias, união de átomos é a atração eletrostática entre íons de cargas opostas num retículo cristalino. Esses íons formam-se pela transferência de elétrons dos átomos de um elemento para os átomos de outro elemento.

Para se formar uma ligação iônica, é necessário que os átomos de um dos elementos tenham tendência a ceder elétrons e os átomos do outro elemento tenham tendência a receber elétrons, ou seja, quando os átomos de dois elementos A e B têm ambos tendência a ceder ou a receber elétrons, não pode se formar uma ligação iônica entre eles.

Os átomos com tendência a ceder elétrons apresentam um, dois ou três elétrons na camada da valência; são todos átomos de metais, com exceção dos átomos de H e He. Os átomos com tendência a receber elétrons apresentam quatro, cinco, seis e sete elétrons na camada da valência; são os átomos dos não-metals e do H, ou seja,

metais têm facilidade para perder elétron (formando cátions) e os não-metais, facilidade para recebê-los (formando ânions).

Para escrever a fórmula de um composto iônico, é necessário descobrir qual a carga do íon estável formado pelo elemento e, a seguir, utilizar a seguinte regra prática, que se fundamenta no fato de que, num composto, a carga elétrica total positiva deve ser igual à negativa, resultando carga total nula.

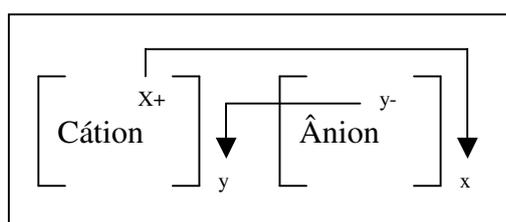


Figura 21 - Fórmula de um Composto Iônico

O cátion é escrito à esquerda e o ânion, à direita.

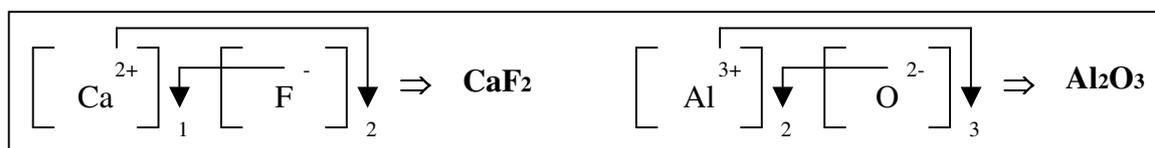


Figura 22 - Exemplos de Compostos Iônicos

Portanto, uma ligação iônica forma-se entre um metal e um não-metal ou entre um metal e o Hidrogênio (H). Os elétrons são transferidos dos átomos dos metais para os dos não-metais ou do H.

Os átomos dos metais, cedendo elétrons, transformam-se em íons positivos ou cátions, e os átomos dos não-metais ou do H, recebendo elétrons, transforma-se em íons negativos ou ânions.

- **Hidrogênio não é Metal:** O Hidrogênio apresenta apenas um elétron. Para ficar com eletrosfera de gás nobre (igual à do hélio, que possui dois elétrons), ele precisa receber 1 elétron. Assim, em ligação iônica, o hidrogênio apresenta carga negativa, ou seja, quando ligado o metal ( $H^-$ ) será chamado de hidreto. Portanto, do ponto de vista de ligações químicas, chamaremos o Hidrogênio como sendo não-metal, assim apresenta tendência para receber elétron para se estabilizar, de acordo com a regra de duplete.
- **Ligação Covalente:** É um par de elétrons compartilhado por dois átomos, sendo um elétron de cada átomo participante da ligação. Nesta ligação, nenhum dos átomos envolvidos transformou-se em íon, ou seja, nenhum deles perdeu ou recebeu elétrons.

Devido ao compartilhamento, todos passaram a ter, em suas eletrosferas, número de elétrons igual ao dos gases nobres. Os grupos de átomos unidos por ligação covalente são denominados moléculas. Três maneiras distintas de representar uma molécula são a fórmula moléculas ( $H_2$ ,  $O_2$ , etc.), a fórmula eletrônica (com o símbolo e bolinhas pretas que representa os elétrons na camada de valência e a fórmula estrutural, na qual cada par de elétrons compartilhado é representado por meio de traço, esta última iremos utilizar para ser representada em Realidade Virtual. Tanto as substâncias simples quanto à composta podem fazer parte desta ligação).

Foi importante para o desenvolvimento do SEQ, a fórmula estrutural, teremos: **Ligação simples:** é uma ligação covalente entre dois átomos ( $A-B$ ). **Ligação dupla:** são duas ligações covalentes entre dois átomos ( $A=B$ ). E **Ligação tripla** são três ligações covalentes entre dois átomos ( $A\equiv B$ ).

- Algumas exceções à regra do octeto:
  - a) Berílio (Be) com 4 elétrons na camada de valência e boro (B) com 6 elétrons na camada de valência, aparecem em algumas moléculas com o octeto incompleto. Mesmo pertencerem a grupo nos quais outros elementos tendem a formar cátions em ligações iônicas, esses dois elementos fazem ligações covalentes.
  - b) Compostos de enxofre e de fósforo, nos quais aparecem com mais de 8 elétrons na camada de valência e é relativamente comum encontrar o fósforo com dez elétrons na camada de valência e o enxofre com doze.
  - c) Compostos NO, NO<sub>2</sub>, ClO<sub>2</sub>, em cujas moléculas aparecem é tomos com número ímpar de elétrons na camada de valência. Esses três compostos apresentam alta tendência a reagir quimicamente, transformando-se em compostos mais estáveis.
- **Ligação Metálica:** As substâncias formadas exclusivamente por átomos de elementos metálicos são denominadas substâncias metálicas, são constituídas pelos elétrons livres que ficam entre os cátions dos metais (modelo do gás eletrônico ou do mar de elétrons). Os metais são constituídos por seus cátions mergulhados em um mar de elétrons.

A ligação metálica explica a condutividade elétrica, a maleabilidade, a ductilidade e outras propriedades dos metais. Entretanto conduzem a eletricidades no estado sólido e no líquido. Na sua maioria, apresentam altos pontos de fusão e ebulição.

- **Ligação Dativa:** É um par de elétrons compartilhado por dois átomos, no qual os dois elétrons são fornecidos apenas por um dos átomos participantes da ligação. Forma-se quando um dos átomos já tem o seu octeto completo e o outro ainda não.
- **Comparando os três tipos de substâncias:**

Tabela 4 - Comparação das substâncias Iônica, Molecular e Metálica.

Substância	Pontos de fusão e de ebulição	Estado físico nas condições ambientes	Conduz corrente Estado sólido?	Elétrica no estado líquido?	Características Complementares
<b>Iônica</b>	Altos	Sólido	Não	Sim	Constituídas por íons (cátions e ânions)
<b>Molecular</b>	Baixos	Sólido, líquido Ou gasoso	Não	Não	Constituídas por moléculas
<b>Metálica</b>	Altos	Sólido	Sim	Sim	Constituído por íons de elementos do tipo metal disposto em retículos cristalino metálicos (modelo do mar de elétrons).

## 2.3 Identificação do Problema

Este tópico foi de extrema importância para o desenvolvimento deste trabalho, pois apresenta todas as dificuldades e necessidades que o SEQ precisa, além do contexto adquirido em revisões técnicas, sobre um estudo de requisitos. Porém, serão apresentados tópicos fundamentais para os desenvolvedores do SEQ.

### 2.3.1 Sobre o Software

- As configurações mínimas para a execução do SEQ inicialmente foram definidas como: Sistema Operacional Windows XP – Pentium IV de 1.4 GHz com 128 de memória Ram e um HD de 20 GB;

- Os desenvolvedores precisarão utilizar algumas estratégias Didáticas sobre a ilustração e animação em Realidade Virtual que possibilitará o aluno a melhor visualização das ligações químicas, podendo desta forma, fixar melhor sua atenção;
- O Modelo Educacional é uma Abordagem Construtivista, entretanto, o SEQ irá auxiliar o aluno no aprendizado e nas soluções dos problemas;
- Este Software está especificamente voltado para alunos do ensino médio (1º a 3º colegial), mas não terá restrições para outros usuários com pouco conhecimento sobre a matéria de química. Mesmo porque para manuseá-lo precisa-se de conhecimentos de ligações químicas, se o aluno quiser formar compostos; ou pouco conhecimento se o aluno quiser testar ou tentar formar algum. Pois o SEQ fornecerá uma opção para gerar o composto automaticamente. Precisa-se também do mínimo conhecimento de informática, pois o software será de fácil manuseio, com diversas ilustrações explicativas;
- Futuramente, com a prática dos alunos, o software teria um grande impacto em suas aprendizagens, proporcionando possibilidades de aprender, pois terão a segurança de realizar as mesmas funções que o software educativo oferece em várias situações didáticas. Os alunos poderão ver os resultados das execuções em animação, comparando suas expectativas originais aos elementos escolhidos e obtendo informações para aonde poderiam utilizar esses elementos ou compostos no cotidiano de suas vidas;
- Serão desenvolvidos vários comandos para as interações com os alunos além da percepção com o ambiente, tanto nos seus aspectos físicos quanto gráficos;
- Um dos principais problemas encontrados seria a quantidade de informações coletadas sobre os conceitos de química, como ela deve ser representada funcionalmente e as técnicas e ferramentas disponíveis para facilitar a coleta de informações.

### 2.3.2 Necessidades do Sistema

- Uma interface visual, pois deverá apresentar os assuntos de forma eficiente e atraente, facilitando assim o aprendizado;
- Relatório de descrição para cada composto, mostrando sua utilidade no meio ambiente;
- O software deverá oferecer um bom nível de interação com o aluno e um planejamento didático;
- Oferecer diversos LINKS e arquivos PDFS, para consultas sobre a matéria de química;
- Apresentar uma lógica de estratégia de escolha dos elementos químicos de forma que permita o aluno a acompanhar esta execução;
- Mostrar todos os estados possíveis de interação com o aluno;
- Apresentar uma boa referência de Ligações Químicas;
- O SEQ precisará conter tópicos de ajuda e bibliografias para o entendimento de química e do manuseio do software como um todo.

### 2.4 Limites ao Uso do Software Educacional de Química

De acordo com as normas de química, o objetivo principal deste software é fazer as ligações dos elementos com coerência, onde o aluno poderá escolher os elementos e as ligações desejadas. Não tendo qualquer tipo de restrições, pois poderá ao término da ligação gravar seu composto para que futuramente o professor o corrija. Caso contrário, se assim desejado, terão limites com as regras das ligações, onde os alunos poderão ver onde estão errando.

Devido à complexidade das ligações químicas (iônica, covalente e metálica), em alguns casos poderão não ocorrer, pelo fato do SEQ ser implementado para o ensino médio. Os melhores tipos de elementos para serem gerados o composto químico, devido a essa

restrição são os metais e não-metais, ou seja, para ligação covalente: ligação entre os não-metais; para ligação metálica: todos são permitidos; e para ligação iônica: a família 1A e 2A com não metal.

Podemos destacar também, a delimitação sobre a usabilidade do software nas escolas, pois o uso desse Software (construtivista) pode provocar uma mudança profunda na abordagem do trabalho no ensino. Uma mudança que coloca ênfase na aprendizagem, colocando no ensino a construção do conhecimento e não na instrução.

## 2.5 Descrição Geral

O desenvolvimento deste software foi composto por quatro módulos: Engenharia de Software, Implementação Funcional, Implementação de Interface e Teste do Software Educacional. Cada módulo foi desenvolvido por um aluno da graduação, de acordo com a tabela a seguir. Onde:

Tabela 5 - Identificação dos alunos e seus trabalhos.

<b>Nomes</b>	<b>Módulos</b>	<b>Títulos dos Projetos</b>
Caio Corrêa da Silva	Implementação de Interface	Ferramenta Didática para Auxílio no Ensino de Química do Ensino Fundamenta – Implementação de Interface.
Éderson Relvas	Teste do Software Educacional	Um Instrumento Para Avaliação de Produto de Software Voltado à Área de Química.
Juliana P. Souza	Engenharia do Software	Análise de Requisitos e Projeto Orientado a Objeto usando UML: Um Estudo de Caso
Roberto Y. Shirtuki	Implementação Funcional	Ferramenta Didática para Auxílio no Ensino de Química do Ensino Fundamental – Implementação Funcional.

Para o melhor entendimento e visualização, as etapas do desenvolvimento foram modeladas no diagrama de Casos de Usos, de acordo com a Figura 23. Apesar de também fazer parte do grupo de diagramas utilizados nesta Análise de Requisitos, nesta seção ele apresenta apenas como uma ferramenta gráfica de entendimento, por tanto, não possui função

adicional de desenvolvimento da aplicação do SEQ. Veja descrição destes casos de uso na seção 2.6 em Ciclo de Vida.

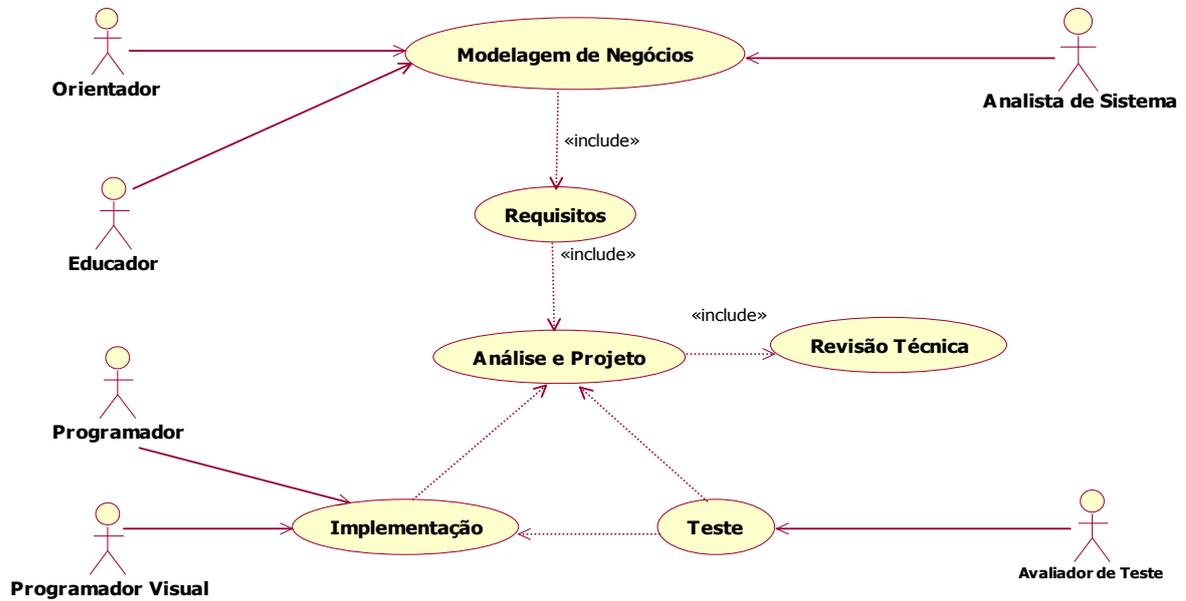


Figura 23 - Diagrama de Casos de Usos dos desenvolvedores do SEQ.

A partir deste diagrama de casos de usos, podemos identificar os atores e suas descrições apresentadas na Tabela 6.

Tabela 6 - Lista de Atores do Diagrama de Casos de Usos.

Atores	Descrições
Analista de Sistema	Desenvolvedor responsável na modelagem lógica do software. Módulo - Engenharia de Software.
Programador	Desenvolvedor responsável pela implementação funcional do software. Módulo - Implementação Funcional.
Programador Visual	Desenvolvedor da manipulação da interface em Realidade Virtual. Módulo - Implementação de Interface.
Avaliador de Teste	Desenvolvedor responsável pela implantação da Avaliação do software. Módulo - Teste do Software Educacional.
Educador	Professor da disciplina de Química, que passará referências para os desenvolvedores e possivelmente utilizará o software como suporte as aulas.
Orientador	Professores das disciplinas de Realidade Virtual e Engenharia de Software que passaram para os desenvolvedores todas as orientações necessárias.

## 2.6 Ciclo de Vida - RUP

As palavras chaves para que desenvolvedores de software cheguem a uma boa qualidade de software são: processo de software, pois se trata de um formalismo que é adquirido por um método, ou seja, é o estudo do processo em seu desenvolvimento. Porém para chegar a um bom nível de qualidade de software, não basta que os desenvolvedores conheçam apenas as aplicações UML.

Existem ciclos de vida fundamentados em ciclos seqüenciais, geralmente utilizados para técnicas estruturadas (conhecido como Tradicional – Cascata) (PRESMAN, 1995): Análise e Especificação de Requisitos, Projeto, Implementação, Implantação e Teste. Essas atividades bem definidas são totalmente dependentes de processos, onde uma etapa depende da outra, podendo trazer vários problemas para os programadores, principalmente para os usuários do sistema que estão aguardando seu término.

De acordo com estas notações, foi utilizado o ciclo de vida baseado em processos OO, onde processos são interativos, independentes e permitem que os resultados parciais sejam avaliados e seus riscos constantemente verificados. Além do que, a UML se trata de uma linguagem independente de processos, onde temos vários requisitos e validações em todo o desenvolvimento do sistema, devido à complexidade dos conceitos e funcionalidades da química.

Portanto, a técnica adotada para ciclo de Vida OO foi a Rational Unified Process (RUP), pois sua distinção é excelente para a modelagem UML.

RUP foi uma proposta feita por BOOCH, RAUMBAUGH e JACOBSON de um processo da Engenharia de Software, que fornece uma abordagem disciplinada para assumir tarefas e responsabilidades dentro de uma organização de desenvolvimento.

Seu objetivo é produzir, dentro de prazos e orçamentos previsíveis, software de alta qualidade que satisfaça às necessidades de seus usuários finais. São utilizadas as melhores

práticas no desenvolvimento moderno de software, permitindo assim, seu uso em diversos projetos e organizações. Dentre as práticas abordadas pelo RUP, existem duas denominada gerência de riscos e modelagem da arquitetura do software (Matos, 2002).

Este processo inicializa-se pela validação dos casos de usos, para garantir que o software se aproprie à organização. E organiza-se por fases e interações que representam o início e a nítida conclusão de uma atividade em Engenharia de Software. (MATOS, 2002) e (FURLAN, 1998) as descrevem como: Concepção, Elaboração, Construção e Transição.

- Fase de Concepção (Inception): Nesta fase os requisitos do sistema são identificados e validados, e também um protótipo pode ser gerado, com o intuito de discussão e constatação dos requisitos. É definida a maioria dos casos de uso, mas são descritos apenas os mais significativos. São estabelecidos os casos de negócios, com objetivo de gerar critério de sucesso, taxa de risco, estimativa de recursos, etc.
- Fase de Elaboração (Elaboration): Esta fase identifica e avalia os riscos, podendo gerar um plano de projeto e uma arquitetura inicial do sistema. Esta fase de elaboração faz uma análise mais detalhada do sistema. Suas metas são: “analisar o domínio de problema, estabelecer uma arquitetura com fundação sólida, desenvolver uma plano de projeto e minimizar elementos de risco potenciais ao projeto” (FURLAN, 1998).
- Fase de Construção (Construction): É a fase similar ao ciclo de vida espiral, onde o resultado de uma fase pode ser constantemente reavaliado até que todos os riscos sejam minimizados. Durante esta fase o produto é desenvolvido de forma interativa e incremental, para que seja pronto à transição para comunidade usuária. Pela suas validações e feedback, esta fase se caracteriza também pela formatação de novos casos de uso refinados, verificando a implementação e o teste do software.
- Fase de Transição (Transition): Neta fase o software torna-se disponível para a comunidade usuária. Entretanto, o sistema pode precisar de ajustes, corrigindo

problemas não detectados ou concluir algumas características adicionais. Para a UML, o uso desta fase enfoca-se pela documentação e reavaliação do modelo proposto.

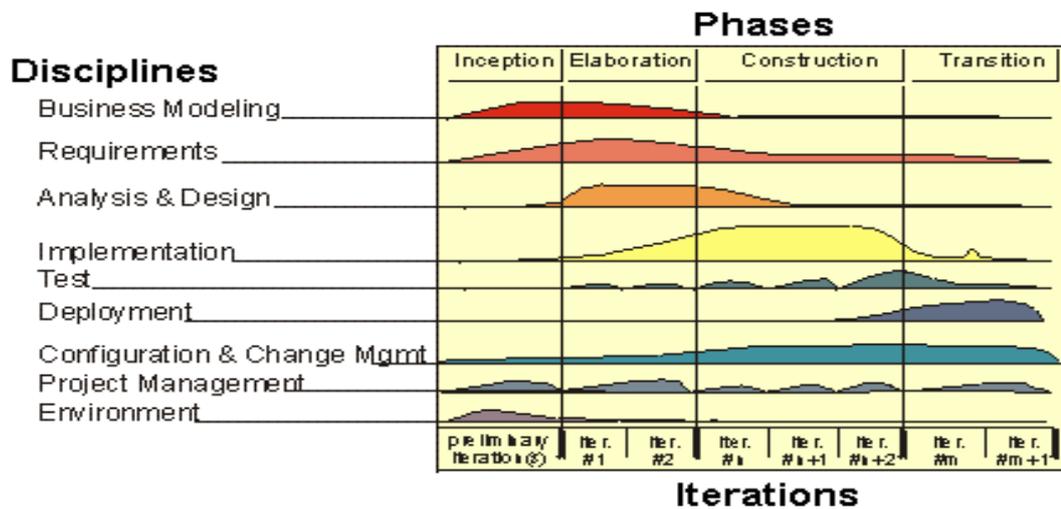


Figura 24 - Fases RUP (Ciclo de Vida Adotado para SEQ) (TAPAS, 2005).

De acordo com a Figura 24 as fases (concepção, elaboração, construção e transição), possuem distintas tarefas: Modelagem do Negócio, Requisitos, Análise e Projeto, Implementação e Testes. Essas são tratadas sobre o diagrama (não funcional) dos casos de usos do sistema, esses estão descritos nas tarefas:

- **Modelagem de Negócio:** Esta tarefa trata-se de avaliação e levantamentos de requisitos, ou seja, abstrações de necessidades sobre o sistema, permitindo que os desenvolvedores e usuários concordem com a mesma descrição. Nesta tarefa como na seguinte, dão ênfase a uma modelagem de negócios (objetivos, regras, recursos, ações) e o que o sistema deve desempenhar (confiabilidade, ambiente técnico, produtos e linguagens). Em UML o diagrama de caso de uso é o principal, mas pode-se desenvolver um diagrama de classe de nível alto. Esta fase foi descrita e modelada no contexto deste trabalho e desenvolvida pela aluna Juliana.

- Requisitos: Esta tarefa é constantemente verificada. Os requisitos são extremamente importantes, até o momento de implantação. Entretanto, faz uma continuidade a fase anterior. Esta fase também foi descrita e modelada no contexto deste trabalho e desenvolvida pela aluna Juliana.
- Análise e Projeto: Conhecida também por Análise e Desenho, esta tarefa inicializa-se quando o sistema estiver sido verificado, neste período, o ambiente e os usuários passam ser definido. O objetivo desta fase é mostrar como o sistema será materializado na fase de implementação, buscando construir um sistema estruturado, de acordo com as tarefas e funções modeladas nos casos de usos. “O propósito básico do modelo de análise é prover uma compreensão do sistema e facilitar a comunicação entre técnicos e usuários – não é voltado a soluções técnicas ou detalhes de código ou programas, mas a um entendimento dos requerimentos e dos casos de uso reais do sistema em desenvolvimento. Os diagramas da UML mais empregados durante a fase de análise são: caso de uso, classe, seqüência e estado” (FURLAN, 1998).

Entretanto, o modelo de Projeto ou Desenho serve como abstração do código fonte, atuando como esboço de sua estrutura. “As atividades de desenho são centradas ao redor da noção da arquitetura, representada por várias visões estruturais que, em essência, são abstrações ou simplificações do desenho total no qual características importantes são tornadas mais visíveis deixando detalhe a parte” (FURLAN, 1998). Não só a análise, mas o projeto (desenho), bem elaborado, ajuda o sistema a obter a qualidade assegurada. Esta fase também foi descrita e modelada no contexto deste trabalho, pela aluna Juliana, essa utilizou como ferramentas para o desenvolvimento destas três tarefas anteriores: Computer Associates Erwin 4.0 (COMPUTER, 20005) e Rational XDE Professional Java Platform Edition (UNIFIED, 2005).

- **Implementação:** Conhecida também como Construção, essa tarefa mostra a independência de processos ou interações, pois junto com análise, os programadores podem construir o primeiro protótipo do sistema. Esta tarefa inicializa-se no momento em que as classes precisam ser implementadas, tomando como base dos diagramas desenvolvidos até agora. Seu objetivo é gerar código fonte, de acordo com os diagramas, esses se forem bem elaborados, a codificação será facilitada. Nesta tarefa poucos diagramas UML são desenvolvidos, na verdade, esses já foram desenhados e validados na fase anterior. Essa fase foi desenvolvida pelos alunos: Roberto em implementação funcional, esse utilizará a linguagem Java e JDBC e Caio em implementação de interface, esse utilizará as ferramentas JAVA e 3D STUDIO.
- **Testes:** Assim que sistema chegue a ser término, essa tarefa é executada. O objetivo desta etapa é identificar erros no código através de um número de casos de teste que avaliem diferentes aspectos de cada módulo sendo testado, garantindo que a entrada definida produza resultados que concordem com os esperados. Este teste não será desenvolvido neste trabalho como um todo, mas sim, uma avaliação sobre o produto (SEQ) final, com o intuito testar seu funcionamento em relação às diretrizes de SEQ focando o ensino médio. Essa aplicação foi feita através de perguntas com seus usuários. Esta fase será desenvolvida pelo aluno Éderson Relvas.

Por este SEQ ter sido desenvolvido por quatro alunos referente a quatro módulos, para atingirmos o sucesso de independência de processos, foi preciso seguir à risca a proposta que o ambiente RUP nos traz. Nos quais Analistas e Programadores utilizariam uma base de informações compartilhada ou distribuída, podendo utilizar algumas ferramentas de apoio, como: todo suíte de produtos Rational, Microsoft Project, Microsoft Word, etc...

## **CAPÍTULO 3 - ESTUDO DE CASO**

O estudo de caso apresentado neste capítulo exercitou os conceitos de OO, UML, banco de dados relacional para apoio do desenvolvimento do diagrama de classes (lógico) e análise de requisitos vistos até agora, tendo principal fonte de informação os objetivos e descrição da informação do SEQ.

Foi utilizado o desenvolvimento do processo de software baseado em UML - RUP, nos quais serão apresentadas as tarefas de análise de requisitos e projetos dentre as fases concepção, elaboração, construção e transição. As ferramentas de apoio para o desenvolvimento do estudo de casos foram:

- RATIONAL XDE PROFESSIONAL para modelagem dos diagramas UML (UNIFIED, 2005);
- COMPUTER ASSOCIATES ERWIN 4.0 para modelagem do banco de dados (Modelo Entidade Relacionamento) que esta em Apêndices A e C (ASSOCIATES, 2005).

### **3.1 Descrição da Informação**

Empregamos nesta descrição funcional o termo “visão” no ambiente virtual para uma melhor interpretação. As Figuras apresentadas neste subcapítulo foram cedidas pelo desenvolvedor de interfaces Caio Corrêa da Silva.

O software terá três tipos de usuários denominados: Administrador do Sistema, Professor e Aluno.

O Administrador do Sistema poderá acessar todas as visões do software, controlar o seu cadastro e o do Professor, esse poderá acessar todas as visões, menos se cadastrar automaticamente, ou seja, precisará do aval do Administrador. Já o Aluno, poderá acessar apenas o laboratório (Visão 1), assim conseqüentemente todas as visões que estarão agregada

a ele (laboratório), com restrição a área do Professor e conseqüentemente a do Administrador. Entretanto, quando neste contexto estiver citado Usuário, refere-se a todos eles, sem restrições.

O Usuário ao iniciar o software, visualizará uma interface onde se caracteriza por um ambiente de entrada (bem-vindo), encontrará um cabide com um jaleco pendurado, possibilitando o usuário a escolher quatro opções: “Entrar no Sistema”, “Cadastrar Usuário”, “Esqueci Minha Senha” e “Sair do Sistema”.

Se o Usuário escolher a opção para “Entrar no sistema” (login), ele terá que inserir seu apelido e senha, caso seja cadastrado, se não, o Usuário poderá se cadastrar escolhendo a opção cadastrar usuário. Neste momento, abrirá uma interface onde constarão os campos para serem preenchidos como: tipo usuário (se for Administrador, Professor ou Aluno), nome, e-mail, data de nascimento, apelido e senha. Caso o usuário escolher por Administrador, constará os campos CPF, RG e Função Administrativa (esse não é obrigatório). O sistema não o cadastrará automaticamente, portanto, será enviado um e-mail para o Administrador do Sistema antecessor validar o cadastro deste usuário, conseqüentemente cadastrando-o.

Caso o Usuário escolher Professor, constará o campo CPF e CRQ (Conselho Regional de Química) esse não é obrigatório. O sistema também não fará o cadastro automaticamente, será enviado também um e-mail para o Administrador do Sistema antecessor, para que possa validar o cadastro, posteriormente cadastrando-o.

Caso o Usuário escolher Aluno, constará sua série e RA (número acadêmico). Automaticamente o cadastro será realizado.

A opção “esqueci minha senha”, ao ser selecionada, terá um campo para ser escrito o e-mail do usuário e nome, assim esse automaticamente será enviado para o Usuário contendo sua senha e apelido.

Após o login:

Se o usuário for o Administrador do Sistema, em seu menu inicial (área restrita), poderá escolher cinco opções, para cada uma terá uma opção para voltar ao menu inicial:

1. Validar, Efetuar e Controlar o Cadastro do Usuário (Professor ou Administrador);
2. Entrar no Laboratório (Visão 1);
3. Cadastrar Arquivos PDF para a Biblioteca;
4. Cadastrar Elementos Químicos para que o Usuário possa utilizar na visão do armário e mesa;
5. Cadastrar Compostos Ativos podendo cadastrar novo ou buscar cadastro de composto inativo que o Aluno criou e alterando-o para composto ativo, esses terão os campos básicos (nome do composto, elementos utilizados, tipo de ligação, etc...).

Se o usuário for Professor, em seu menu inicial (área restrita), poderá escolher quatro opções, também para cada uma terá uma opção de voltar ao menu inicial:

1. Entrar no Laboratório (Visão 1);
2. Cadastrar Arquivos PDF para a Biblioteca (Visão 2);
3. Cadastrar Elementos Químicos para que o Usuário possa utilizar na visão do armário (Visão 3) e mesa (Visão 4);
4. Cadastrar Compostos Ativos podendo cadastrar novo ou buscar cadastro de composto inativo que o Aluno criou e alterando-o para composto ativo, esses terão os campos básicos (nome do composto, elementos utilizados, tipo de ligação, etc...).

Se o usuário for Aluno, terá apenas uma opção:

1. Entrar no Laboratório (Visão 1). Esse é o ambiente principal e apresenta três visões (biblioteca, armário e a mesa).

Portanto, o Laboratório chamado de Visão 1, será o próprio laboratório e terá todas as possíveis características de um laboratório químico virtual. De início, terá a sombra de uma pessoa em uma suposta porta, como se, a câmera para visualização estivesse focada nela. Para gerar cada uma das visões, o usuário terá que interagir com o ambiente, com o mouse terá de clicar em cima dos objetos. Referente a Figura 25.

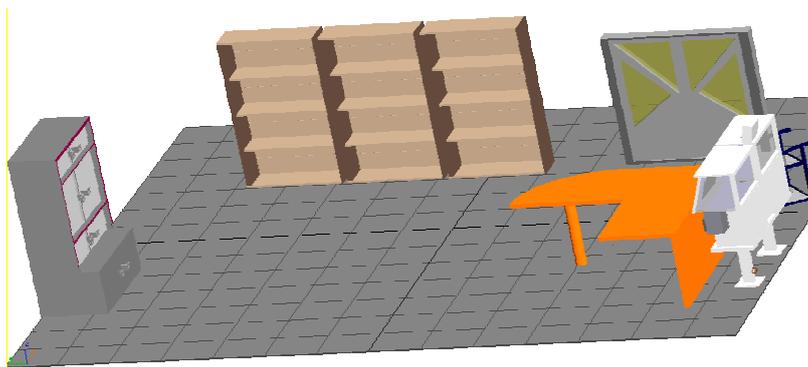


Figura 25 - Visão 1 - Laboratório Químico.

A Biblioteca chamada de segunda visão (Visão 2), constará diversas bibliografias sobre a química, inicialmente estará em um canto do lado esquerdo do laboratório e terá uma estante com as portas fechadas. Assim que o usuário a selecionar e gerar a visão, a animação ocorrerá, a porta se abrirá e encontrarão livros e disquetes. O usuário poderá escolher com o mouse o livro e/ou disquete que deseja, gerando arquivos PDFs e/ou LINKs em HTML. Estes arquivos serão consultas para o usuário sobre a matéria de química. Terá uma opção para voltar a primeira visão (laboratório).



Figura 26 - Visão 2 - Biblioteca.

O Armário denominado como sendo a terceira visão (Visão 3), estará perto da parede de fundo e desenvolvido em cima da tabela periódica. Ele terá diversas gavetas animadas, cada gaveta é representada por um elemento químico com propriedades físicas e químicas. Com o mouse o usuário poderá escolher a cada click quantos elementos deseja pegar ou excluir, automaticamente os elementos irão por um tubo, animando até a parede direita, que será a Mesa (quarta visão). Também terá a opção para o usuário voltar para a primeira visão (laboratório) e ir direto para a quarta visão.

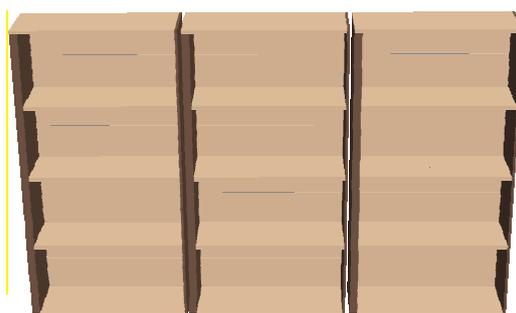


Figura 27 - Visão 3 - Armário, com as gavetas e sem os elementos químicos.

A quarta visão (Visão 4) será a Mesa, possuirá vários objetos característicos de laboratório, onde um deste, será uma máquina animada virtual que armazenará os elementos vindos do armário (pela escolha do usuário). Nesta fase o usuário poderá selecionar com o

mouse os elementos desejados, ele sairá da máquina, passará por um tubo e cairá na mesa. O usuário poderá voltar à terceira visão (armário) para escolher mais elementos. Além disso, esta visão irá conter a opção para o usuário voltar à primeira visão (laboratório).

Nesta quarta visão, haverá algumas funções como: O usuário poderá excluir os elementos ou o composto pronto, selecionando com o mouse o lixo (que estará em cima da mesa) e o elemento ou composto a ser excluído, o lixo poderá ser habilitado ajudando o usuário a excluir mais de um elemento de uma só vez. Haverá a opção de Ligação Automática, onde se habilitada, selecionando os elementos, se ligarão automaticamente (se existir o composto).

Terá também três opções para a ligação dos elementos que serão: ligação iônica, ligação covalente e ligação metálica, onde se habilitada, gera a ligação manualmente de elementos possíveis, caso contrário haverá uma animação de repulsão mostrando para o usuário que essa ligação não pode ser feita. Caso o usuário deseje fazer múltiplas ligações, ele a fará por fases, ligação por ligação (binária) até chegar em sua fase final. Terá uma base de dados mostrando as possíveis ligações existentes para unir com a outra.

A opção de Cortar, graficamente será um “desenho de tesoura” se habilitada, cortará a ligação feita. Quando o composto estiver gerado, e for inativo para o software, o usuário poderá salvá-lo, respondendo um breve cadastro de composto inativo (nomes dos elementos utilizados, etc...) podendo posteriormente o professor ou administrador acessá-lo e analisá-lo, salvando-o como composto ativo.

Depois que o composto for salvo como composto inativo, pelo Aluno, se quiser, poderá alterá-lo até que o professor o corrija e passe de inativo a ativo. Também conterà uma opção para um breve relatório, que constará a descrição do composto gerado, essa opção só será realizada quando as transformações de elementos para compostos estiverem prontas, será denominado “Sobre o Composto”.

Esta visão terá alguns tubos de ensaios em cima da mesa, ligado a máquina para quando esta estiver cheia de elementos e com dificuldades para serem visualizados, eles (elementos) ficarão armazenados nesses tubos, com o intuito de melhor visualização.

A cada transformação de elementos para compostos, esperaremos reações consistentes e capazes de mostrar para o usuário todo o conceito de como fazer ligações químicas, usando conteúdos importantes como: Regra do octeto, ligação iônica, ligação covalente, ligação metálica e ligação dativa.

Neste contexto retrata-se o detalhamento inicial do SEQ como um todo, todavia, é importante ressaltar alguns pontos que somente foram minuciado na descrição dos casos de usos ou no diagrama de seqüência.

### **3.1.1 Descrição Inicial da Interface Visual do Software**

A descrição da interface foi discutida em uma das reuniões técnicas pelos desenvolvedores (Juliana Pereira de Souza e Caio Corrêa da Silva), foram encontrados os seguintes quesitos:

- Todos os objetos funcionais do laboratório ao passar o mouse irão ter uma animação de brilho (mudança de cor);
- O Laboratório (Visão 1) terá um som ambiente de ar-condicionado podendo ser habilitado, para que o usuário possa ter a opção de on/off;
- Na parede de fundo do laboratório, terá uma porta automática transparente, que ao ser selecionada com o mouse, executará uma animação abrindo e jogando um chuvisco de jato com a funcionalidade de desinfetador;

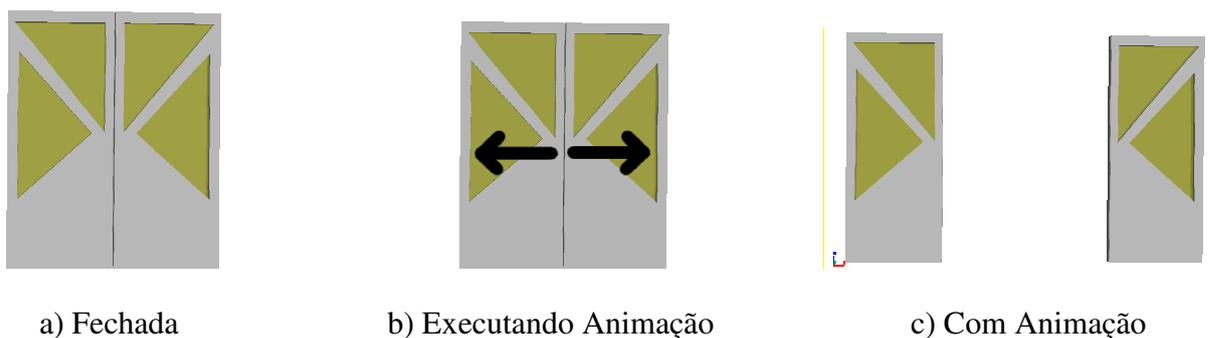


Figura 28 - Porta automática

- Além de um tópico de ajuda que terá na entrada do software, no laboratório será colocado um objeto virtual, em um canto da sala, onde no momento em que ele for selecionado pelo usuário, abrirá o link de ajuda sobre o manuseio do software;
- Usou-se um software, desenvolvido por alunos de ciência da computação, formados em 2004, em que trata de uma ferramenta de visualização tridimensional dos átomos, apresentados na tabela periódica dos elementos denominada Quico (Ferramenta didática para o ensino de química). Em uma das paredes, estará em um quadro referente à tabela periódica, o aluno o selecionará abrindo o link para o Quico. Portanto, o aluno terá mais uma opção de busca para o entendimento dos componentes de química e a visualização em 3D dos átomos (GIOVANINI, 2004).
- O ambiente virtual será um laboratório de química, onde constarão suas características, como: mesa, acessórios específicos, luz clara, ou seja, todos recursos necessários para que o usuário possa focar toda sua concentração.
- As interfaces das ligações químicas (compostos) vão ser estruturadas, com cores, formatos, animações, características de acordo com as suas normas químicas, onde os usuários poderão analisar cada composto criado perfeitamente o que não poderia ser feita com o ensino aluno-professor.

- No suposto armário, cada gaveta estará descrita o símbolo, número, nome, e massa atômica dos elementos químicos, como se fossem a própria tabela periódica.

### **3.2 O Desenho do Diagrama de Casos de Uso**

Após a narrativa do sistema, será apresentada uma solução modelando casos de uso, com intuito de representar uma solução técnica possível. Porém, mesmo sabendo que o SEQ refere-se a várias outras possibilidades de soluções, somente com a implementação funcional (prática), podemos analisar a solução proposta e verificar se uma é melhor que outra. Entretanto, o objetivo deste estudo de caso, foi produzir o SEQ com os melhores resultados, melhor proposta de modelagem e garantindo sua qualidade.

Neste contexto, foram levantados atores e suas responsabilidades para a modelagem do diagrama de caso. Todavia, a missão desta fase foi analisar e determinar todos os casos de usos que descrevem o SEQ, definindo como será utilizado por cada usuário, através de suas funcionalidades. O SEQ por ser um sistema complexo abrangendo realidade virtual, terá diversos ambientes, nos quais para cada usuário, terá uma área específica de manuseio. Entretanto, foram definidos os atores como sendo humanos: Administrador\_Sistema, Aluno e Professor, e sistemas (ambientes definidos): Mesa, Armário, Laboratório e Biblioteca. Portanto, a análise de requisitos permitiu a definição desses atores, conforme a Tabela 7.

Tabela 7 - Lista de Atores do Diagrama de Casos de Usos do Projeto.

<b>Atores</b>	<b>Descrições</b>
Usuário	Usuário do sistema ainda não identificado poderá ser Aluno ou Professor ou Administrador do Sistema.
Aluno	Usuário classificado, o software é destinado a ele aprender, praticando as ligações químicas existentes. Poderá acessar apenas o Laboratório (Visão 1).
Professor	Usuário classificado, poderá acessar todos ambientes virtuais, menos se cadastrar e cadastrar o Administrador do Sistema.
Administrador do Sistema	Usuário classificado, apenas usuário que for administrador, poderá se cadastrar e cadastrar o professor. Acessará todos os ambientes virtuais do software.
Laboratório	É o ambiente virtual principal. Terá características de um laboratório químico virtual. Poderá ser direcionado a todos ambientes do sistema e acessar diversas funcionalidades.
Biblioteca	É um ambiente virtual. Será uma estante com porta animada, livros e disquetes e poderá voltar ao laboratório.
Armário	É um ambiente Virtual. Será desenvolvido em cima da tabela Periódica, onde terá diversas gavetas, cada qual, um elemento químico. Poderá ser direcionado a mesa ou voltar ao laboratório.
Mesa	É um Ambiente Virtual. Terá suas características específicas como: máquina virtual animada e objetos químicos. Poderá ser direcionado ao armário ou voltar ao laboratório.

Após estas abordagens, o desenho passa por outra análise, onde define seus relacionamentos e estereótipos. A ferramenta Rational usa o estereótipo <<include>> em vez de <<use>>, por exemplo, o caso de uso “Cadastrar Usuário” usa obrigatoriamente o caso “Validar Cadastro Aluno”. Entretanto, o caso “Escolher Elemento Químico” estende o caso “Excluir Escolha de Elemento”, ou seja, o primeiro caso é estendido ao segundo, não obrigatoriamente.

Existe também o estereótipo <<access>>, por exemplo, o ator Usuário acessa o caso “Acessar Objetos Virtuais” e o <<dependency>>, representa apenas por uma linha pontilhada e com uma seta aberta, todavia, não vem acompanhado pelo nome, por exemplo, os casos de uso “Acessar Área Aluno”, “Acessar Área Professor” e “Acessar Área Administrador” dependem do caso de Uso “Autenticar Senha”.

No diagrama de caso de uso foram apresentados os relacionamentos de generalização (associação) entre os atores: Usuário, Aluno; Usuário, Professor e Usuário, Administrador\_Sistema, onde declara que os atores Aluno, Professor e Administrador\_Sistema são um Usuário, obedecendo a uma hierarquia e conceito de herança. Foram apresentadas também as associações simples entre ator e caso de uso e agregações entre atores, por exemplo, os atores Biblioteca, Mesa e Armário são parte do Laboratório, apresentando o conceito de todo-parte.

O diagrama de caso, apresentado na Figura 29, por ser caracterizado por diagrama chave da Análise de Requisitos, foi detalhadamente refinado em relação à descrição da informação. Pois o esforço empenhado, sobre tal refinamento, evita os erros de projeto através da prévia identificação de riscos. Entretanto, a causa de erros encontrados em projetos, da se a riscos críticos encontrados tardes de mais, esses encontrados principalmente na fase de teste, conseqüentemente o sistema ficará de baixo nível.

Foram modeladas todas as funcionalidades do SEQ, ou seja, trinta e três casos de usos para oito atores. Para cada caso de uso serão mostradas suas descrições (Ver Apêndice B). Por serem muitos extensos, representaremos para os diagramas de seqüência e atividade, o caso de uso número vinte e sete, denominado “Ligar Elemento Químico” e os restantes estão em Apêndice B.

A partir deste diagrama de caso de uso, que foi o primeiro passo para a fase de análise de requisitos, e através de seu refinamento, o SEQ chegou ao objetivo deste trabalho: a representação dos diagramas UML com base na teoria da OO consistentes e principalmente com qualidade assegurada.

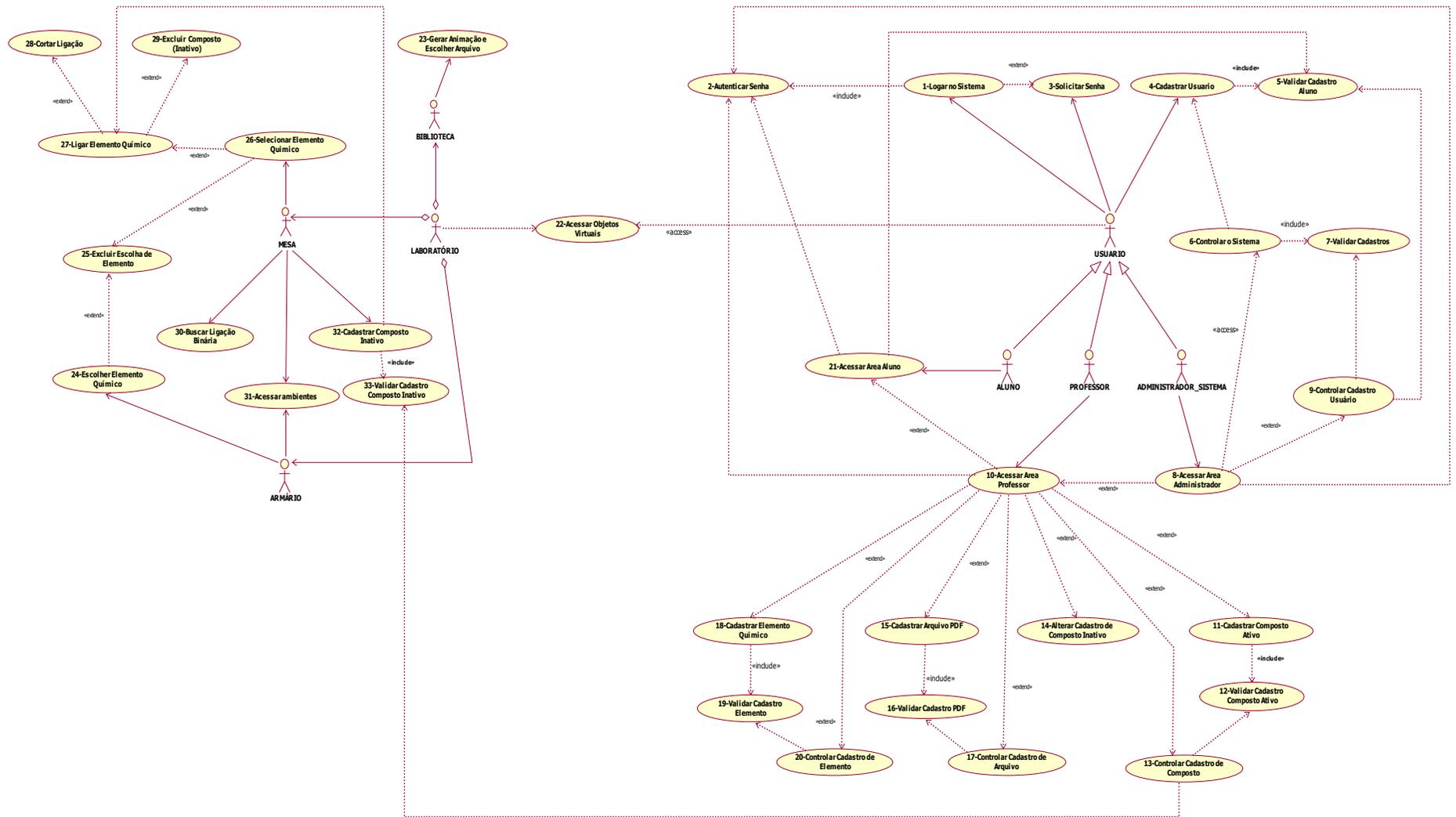


Figura 29 - Diagrama de Casos de Usos do SEQ

### 3.3 Modelo de Classe Preliminar

“Após casos de uso terem sido criados, uma análise dos objetos que permitem cada uma das tarefas documentadas serem executadas (ou seja, terem um proprietário) é a próxima etapa” (MATOS, 2002).

Primeiramente apresentamos um modelo de classe preliminar (modelo lógico) mais simples possível, onde o objetivo desta etapa do processo foi identificar classes, relacionamentos, atributos, de acordo com um estudo do diagrama de casos de usos modelado no item anterior.

Para essa modelagem, de acordo com o (FURLAN, 1998), os relacionamentos das classes, ainda não sofreram quaisquer questionamentos conceituais profundos. Entretanto, além do diagrama de casos de uso, foram feitos alguns levantamentos e validações em relação aos possíveis usuários (Professor, Aluno e Administrador do Sistema). Com intuito de identificar os conceitos de relacionamentos, cardinalidade e chaves primárias do modelo. Todavia, este processo é imprescindível para a próxima etapa de modelagem de interação, pois através dessa idéia inicial de classes, podemos pensar em operações e mensagens correspondente às classes.

Portanto, o diagrama de classe lógico passou a ser composto pelas classes preliminares: Usuário, Aluno, Professor, Administrador\_Sistema, Composto, Elemento\_Químico, Composto\_Elemento e Biblioteca.

Os relacionamentos propostos foram esclarecidos no Diagrama de Classes final. Porém foi apresentado como modo de apoio ao Banco de Dados Relacional, o Modelo de Entidade e Relacionamento (MER) desenvolvido pela ferramenta Erwin. Este modelo é apresentado no Apêndice A.

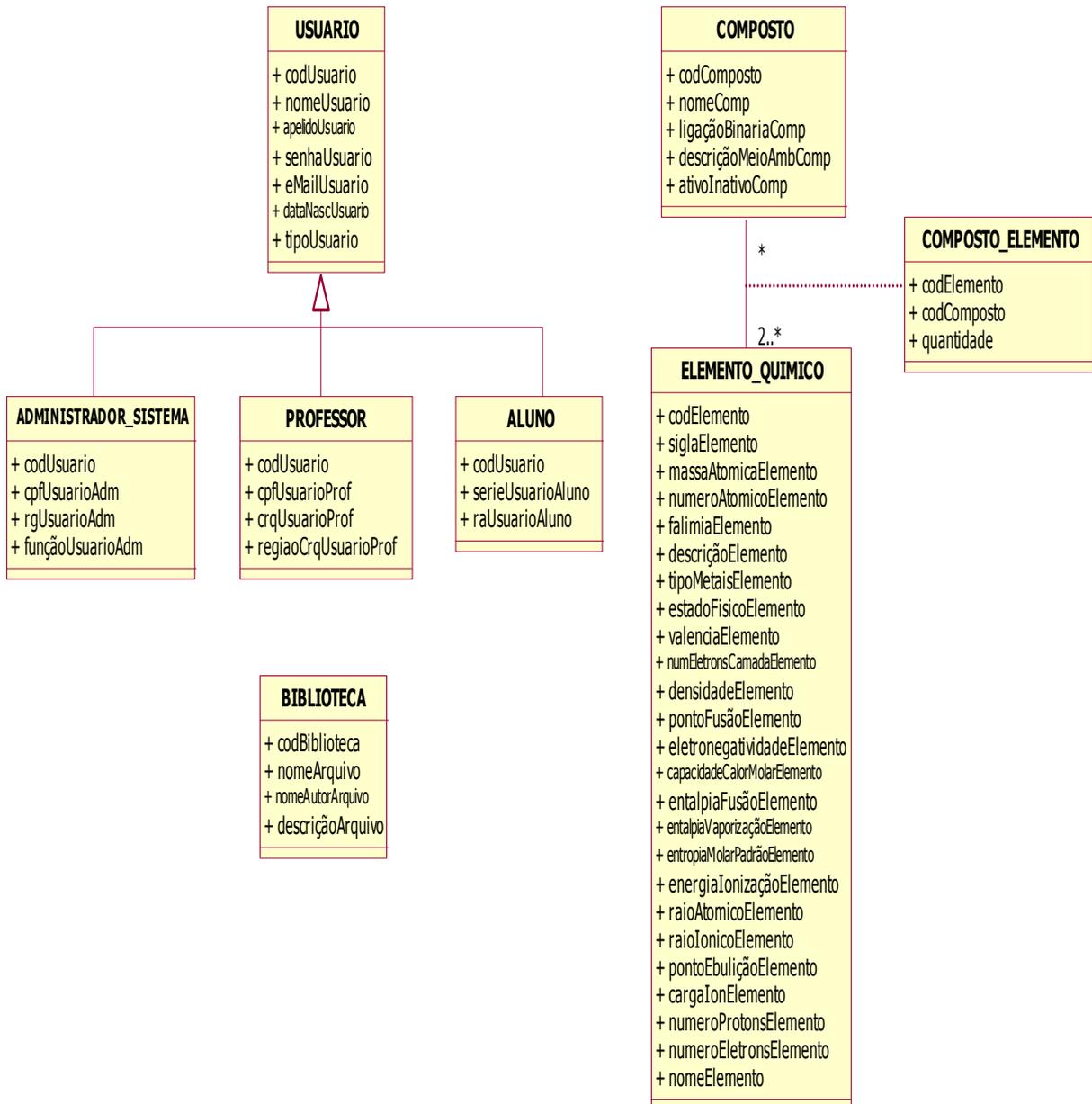


Figura 30 - Diagrama de Classe (lógico) Preliminar

### 3.4 Desenhando Sistemas Dinâmicos

Após da construção do diagrama de classe preliminar, foi possível a criação de diagramas de interação e atividade, todavia, o objetivo desta fase consistiu em modelar diagramas de seqüência para cada caso de uso. Entretanto, podendo estabelecer todas operações e mensagens das classes do sistema, com o propósito de modelar o comportamento

dos usuários e atores, diante das interfaces gráficas do sistema e como estas interagem com o banco de dados.

Pelo SEQ se caracterizar por um software especialista, entre diagrama de seqüência e colaboração, o primeiro se enquadra melhor, além de mostrar seqüência explícita de mensagens e ter uma melhor abrangência em relação à dimensão de tempo e espaço. Por isso, este fora modelado.

Este diagrama representa o comportamento do usuário, caracterizado por um ator, diante das interfaces gráficas do sistema e como este sistema interage com as informações armazenadas em resposta às requisições do usuário.

Serão mostrados todos diagramas de Seqüência do SEQ. O caso de uso apresentado é o “Ligar Elemento Químico”, pelo fato de proporcionar características incomuns para outros sistemas e fundamentais para o SEQ. Todavia, precisamos ter como ponto de partida, um domínio total da descrição desse caso de uso, ou seja, validações das tarefas intermediárias que compõem o caso.

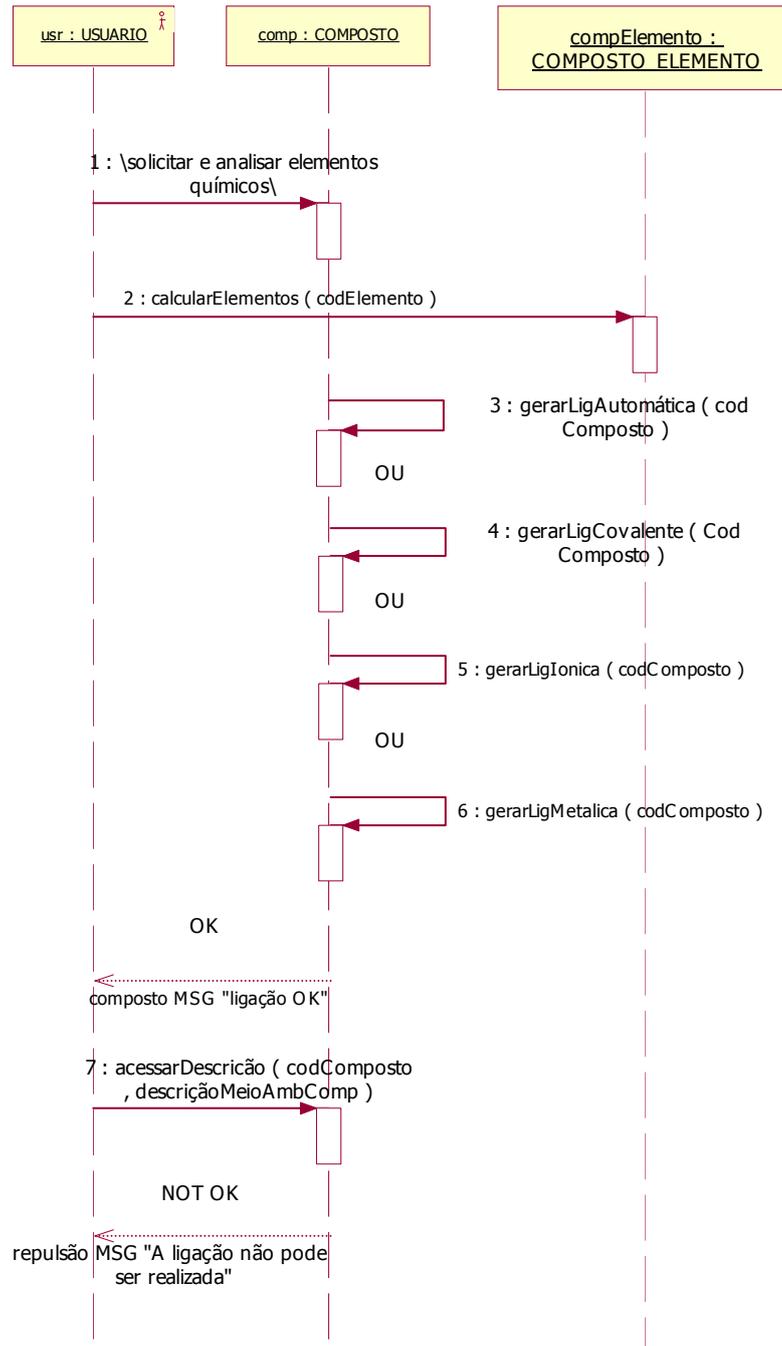


Figura 31 - Diagrama de seqüência do caso de uso “Ligar Elemento Químico”

MATOS (2002) descreve esta fase de modelagem de diagrama de seqüência, colaborando com alguns aspectos fundamentais:

- Colabora com a identificação de métodos das classes;

- Colabora com a visualização do nível de detalhamento que foi dado aos casos de usos, podendo analisá-lo e corrigi-lo, principalmente os estereótipos (extends e include);
- Colabora com o comportamento das mensagens.

Portanto, este diagrama representa o comportamento do sistema, dando ênfase no aspecto temporal. Os demais trinta e dois diagramas estarão representados no Apêndice B.

### **3.5 Desenhando Diagrama de Atividade**

O diagrama de atividade descreve o ciclo de vida de um objeto de um sistema ou uma operação. Para o SEQ foram modeladas as atividades referentes aos casos de usos do sistema. Para alguns diagramas apresentados em Apêndice B, foram modelados mais de um caso em apenas um diagrama, com o objetivo de diminuir a simplicidade de alguns casos se fossem modelados sozinhos. Por exemplo, os de números 4, 5, 6 e 7: “Cadastrar Usuário”, “Validar Cadastro Aluno”, “Controlar o Sistema” e “Validar Cadastro”.

A principal característica desta fase foi o fato de apresentar o comportamento do objeto em relação às atividades. No exemplo da Figura 32, será mostrada claramente a seqüência proposta para o Usuário, podendo solicitar e analisar elementos químicos. Entretanto, foram modeladas em vários pontos, as possibilidades de condições entre gerar ligações químicas. Ressaltamos também, o paralelismo em outros casos, em Apêndice B.

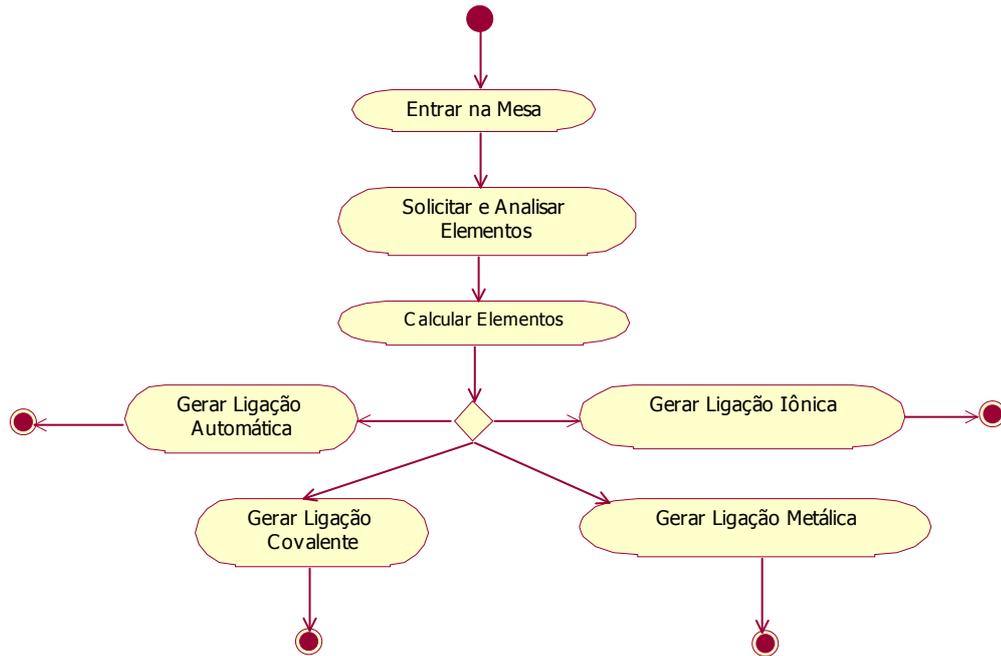


Figura 32 - Diagrama de Atividade de caso de uso “Ligar Elemento Químico”

### 3.6 Desenhando Classes com Operações

Esta fase de projeto consiste na complementação do diagrama de classes, resultante das fases anteriores. Colocando em prática alguns conceitos como: abstração, ocultação de informações, integridade e modularidade, dentre outros estudos OO, como herança, operação, atributos. Todavia, alcançamos as operações através de mensagens recebidas pelas classes nos diagramas de seqüência de casos de usos.

Objetivando também a representação do modelo físico do banco de dados relacional do sistema. Além da criação deste diagrama, esta fase de projeto também incrementou a criação de um diagrama de objetos, com o intuito de instanciar os objetos, porém, representando como eles se relacionam em determinado ponto no tempo de execução do sistema, conseqüentemente, validando os atributos e comportamento das classes.

São consideradas oito classes: Administrador\_Sistema, Professor e Aluno classificadas como subclasses (classes especializadas). Essas herdam atributos, operações e

associações da classe Usuário, classificada como superclasse. Este relacionamento denomina-se generalização, dando ênfase aos conceitos de herança.

Temos também as classes Biblioteca, Composto que se relacionam com a classe Elemento\_Químico. A multiplicidade deste relacionamento de associação é de muitos-para-muitos (\* - 2..\* ). Porém, foi criada uma nova classe “Composto\_Elemento”, classificada como classe de associação.

O diagrama de classes físico, com sua devida normalização, ou seja, sua integridade e estabilidade, e o diagrama de objetos são mostrados nas Figuras 33 e 34, respectivamente.

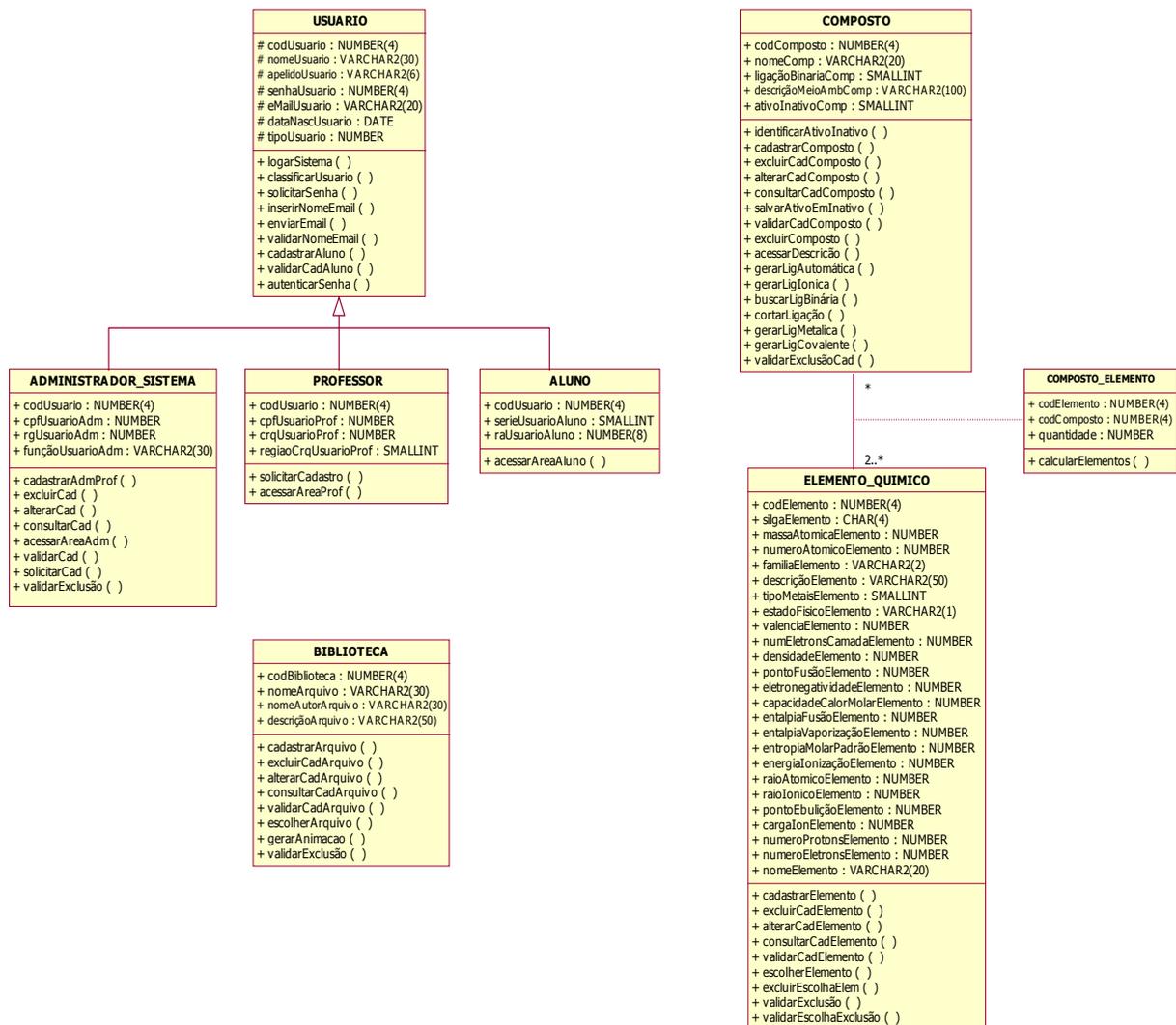


Figura 33 - Diagrama de Classes do SEQ

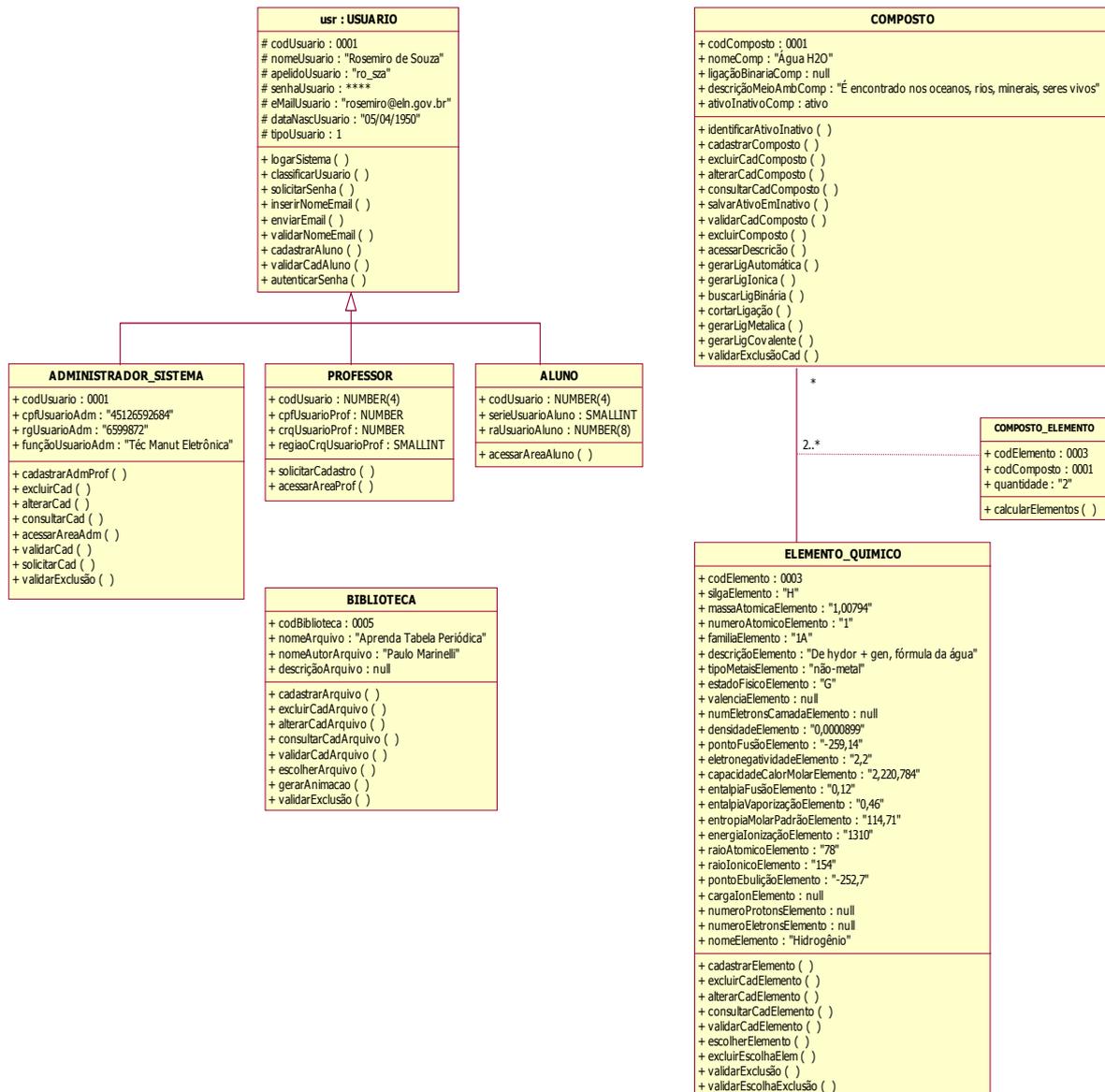


Figura 34 - Diagrama de Objetos do SEQ

O SEQ como visto no capítulo anterior, será implementado em linguagem Java, ou seja, linguagem de POO. Entretanto, uma das vantagens da UML é seu alto nível de abstração aos conceitos OO, porém, não obrigando qualquer programador a utilizar essa notação a programação.

Tendo em vista esses quesitos, foi elaborado o MER físico, originando o dicionário de dados (DD), de acordo com os Apêndices C e D. Com o intuito de ajudar futuramente algum programador que queira implementar este software em qualquer outra ou até nova

linguagem de programação. A partir deste contexto, os objetos apresentados na Figura 34 podem ser visto pelos programadores como registros, ou seja, linhas de tabelas do banco de dados.

Portanto, para esses modelos MER e DD foram utilizados a ferramenta Erwin, e os tipos de dados apresentados tanto para o MER quanto para o diagrama de classes, foram usados o gerenciador de banco de dados (SGBD) ORACLE 8.x, possibilitando mais segurança e confiabilidade nas informações das atividades desenvolvidas, tanto para usuários internos quanto externos. O SGBD escolhido declara tipo de dados conceituais, porém, não se preocupando se oferecerá recursos para serem implementados em OPP.

### **3.7 Desenhando Diagrama de Estado**

De acordo com (MATOS, 2002), a análise dos aspectos dinâmico de um estudo de caso, pode ser útil na especificação do comportamento temporal de uma classe. Esta constatação pode ser feita se uma classe não for tratada como um elemento estático, mas sim um elemento com várias transições desde a concepção.

Após os diagramas já modelados, analisamos estados, onde foi desenhado o ciclo de vida de objetos do sistema. Neste contexto, utilizamos a classe Usuário, representando máquinas de estados, com objetivo de focar situações e comportamentos em que o Usuário entra no sistema, até o modo em que ele se classifica como Aluno ou Professor ou Administrador de Sistema, conseqüentemente cada um, acessando sua área restrita. De acordo com a Figura 35.

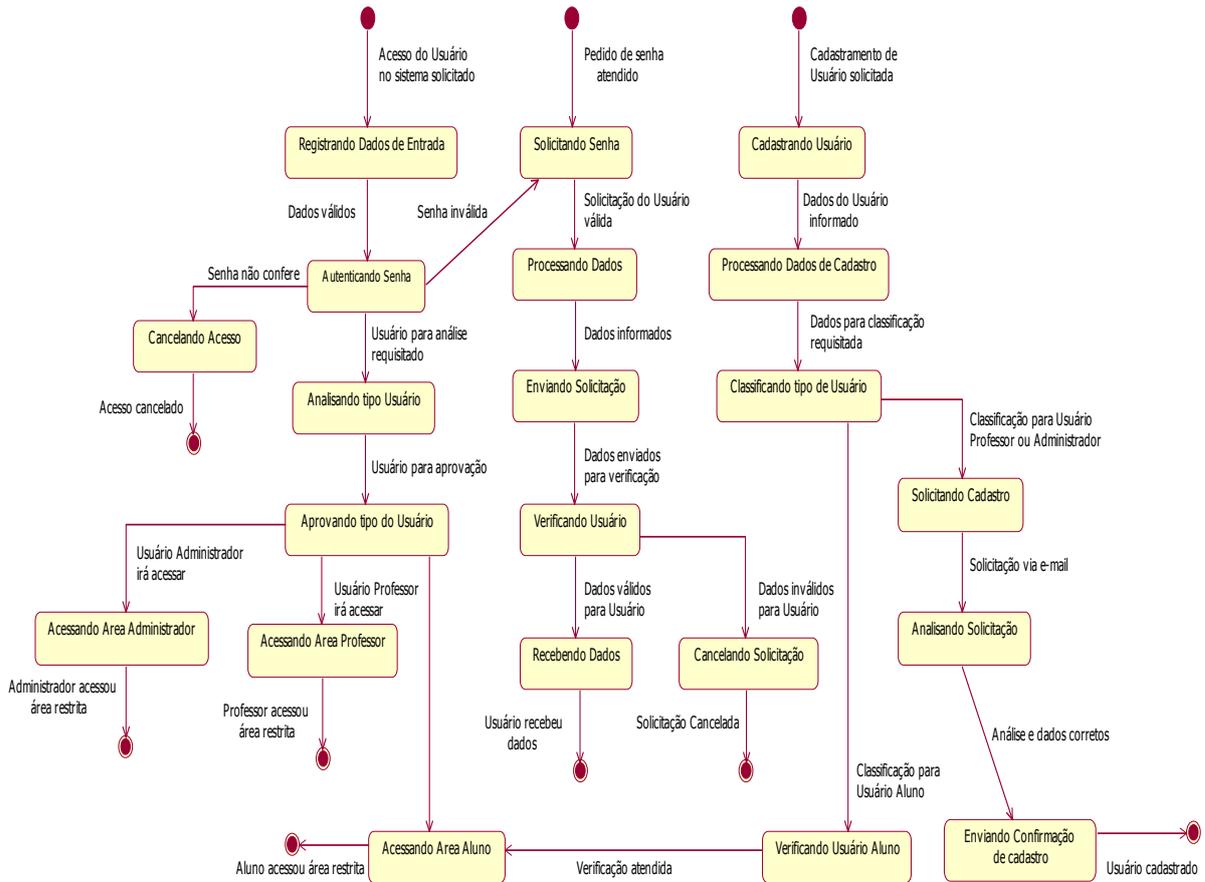


Figura 35 - Diagrama de Estados da classe Usuário

### 3.8 Desenhando Aspectos de Implementações

Nessa fase de implementação, o sistema passa por uma análise arquitetural, onde necessita do desenvolvimento dos diagramas de componente, mostrando a estrutura do código fonte e ajudando a organizar melhor essa estrutura, conseqüentemente, pelo fato da criação de componentes, ajuda também na reutilização de código. E o diagrama de Implantação, para mostrar a organização do hardware e a ligação do software aos dispositivos físicos.

O diagrama de Componente foi desenhado a partir de uma junção das classes, onde representam as dependências das bibliotecas (estereótipos) (Usuário.dll, Arquivo.dll e Química.dll) com as mesmas. Todos os componentes possuem interfaces de assinatura de métodos requerida. Cada uma dessas bibliotecas apresenta interfaces fornecidas, ou seja, elas

forneem algum mecanismo para que o arquivo SEQ.exe precise dependê-las, de acordo com a Figura 36.

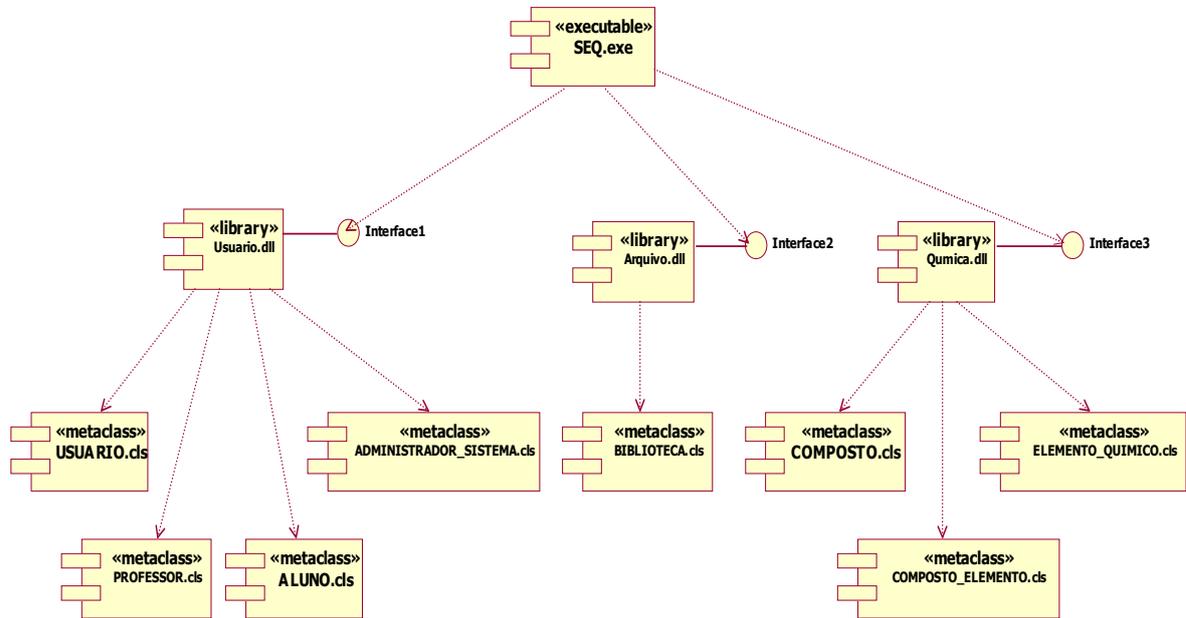


Figura 36 - Diagrama de Componente do SEQ

Já o diagrama de implantação, pelo fato de ser software educacional, não possui estereótipos, como fax, server, modem, entre outros, como acontece na maioria dos sistemas empresariais. Por isso, foi desenhado um pequeno diagrama, com o intuito de representar apenas dois nós (estereótipos): O PC que seria o computador onde o SEQ será instalado, e a Impressora, pois os usuários conseguirão imprimir os arquivos PDFs cadastrados. Esse se relaciona com a seta de associação para o PC, com multiplicidade de 1 (PC) para \* (muitas – Impressoras). Entretanto, o diagrama exibiu também que o componente SEQ.exe pode executar neste nó (PC), usando entre eles o relacionamento de dependência com o estereótipo <<suporta>>. Porém, de acordo com a Figura 37, o tipo de nó PC suporta o tipo de componente SEQ.exe e esse depende do nó para ser executado.

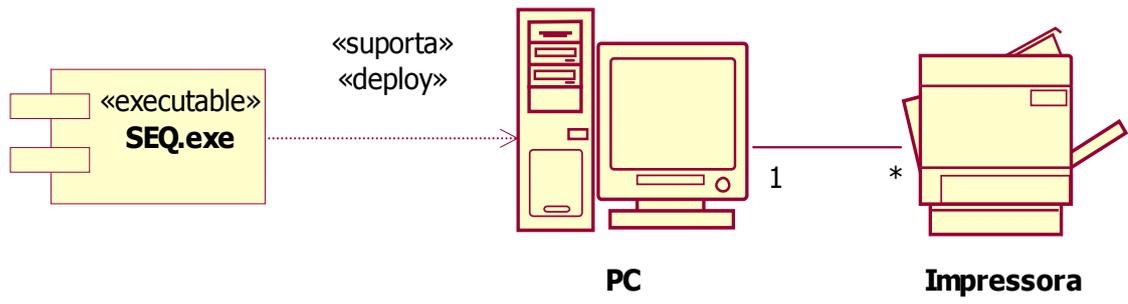


Figura 37 - Diagrama de Implantação do SEQ

## CONCLUSÕES

As abordagens da Engenharia de Software utilizadas sobre a UML, agregando o Ciclo de Vida RUP, foram suficientes para o desenvolvimento do Estudo de Caso do SEQ, podendo apresentar os níveis de abstrações de seu problema.

Porém, fica clara a importância das etapas de Análise de Requisitos e Projeto para desenvolver um sistema de qualidade assegurada, podemos assim visualizar e controlar seu processo de desenvolvimento de maneira eficaz e detalhada, identificando e gerenciando riscos, estipulando e cumprindo prazos, dentro das estimativas de custo e principalmente de forma padronizada.

É necessário destacar também a importância da criação de modelos de requisitos, estes envolvem diversas características e aspectos fundamentais na construção de seu desenvolvimento. Para isso a UML abrange não só o SEQ, mas todos sistemas em desenvolvimento, independente de sua linguagem de programação, pois seus diagramas compreendem diversos níveis de abstração, tanto estrutural, comportamental, quanto arquitetural.

Entretanto, no estudo de caso, obtivemos facilidades de enxergar por meio de abstração, que pode ser entendida por qualquer programador. Todavia, destaco a vantagem da OO para a implementação do SEQ, dando continuidade para essa técnica, possibilitando também o melhor aproveitamento dos recursos da Metodologia Unificada e outras vantagens, como por exemplo, a facilidade de teste e reutilização de código.

Particularmente, ressalto a extensão no que diz respeito à Engenharia de Software. Atualmente, ainda se vê programadores ou outros profissionais da computação, considerarem essa área, como sendo irrelevante para o desenvolvimento de um software. Porém, com este trabalho, apresento além de sua importância, alguns mecanismos para planejar e gerenciar os processos de desenvolvimento do SEQ.

Neste contexto, a Análise e Projetos OO são sem dúvidas primordiais e indispensáveis, para que o software tenha qualidade, portanto, satisfazendo seus usuários. Por esse motivo, na busca por melhorias no processo de desenvolvimento do SEQ, uma sugestão para trabalho futuro, seria a implementação de uma técnica de teste no paradigma OO, com as quais pode-se gerenciar este processo e detectar falhas de Projeto, aplicadas neste trabalho ou na implementação funcional (código fonte).

Em virtude das modelagens dos diagramas deste respectivo estudo passarem por constantes validações, ressalto também, uma alteração no caso de uso “Validar Cadastro Aluno”, para uma continuidade de trabalho futuro. Pois, seria interessante que o usuário “Aluno” não se cadastrasse automaticamente e sim, solicitasse um aval para o usuário “Professor”, assim, seguiria uma restrição de hierarquia entre “Administrador de Sistema, Professor e Aluno”. Ainda mais que, para a metodologia de software educacional do ensino médio de química, é interessante para inclusão digital nas escolas, a identificação do aluno para o manuseio do software.

## REFERÊNCIAS

ANQUETIL, Nicolas. **Desenvolvimento de Software Orientado a Objetos**. Disponível em: <<http://www.ucb.br/ucbtic/mgcti/paginapessoalprof/Nicolas/Disciplinas/UML/>>. Acesso em: 02 maio 2005.

BRAGA, Rosana Teresinha Vaccare; GERMANO, Fernão Stella Rodrigues; MASIERO, Paulo César; MALDONADO, José Carlos. **Introdução aos Padrões de Software**. São Carlos, 2001, Disponível em: <<http://sugarloafplop2005.icmc.usp.br/NotasDidaticasPadroes.pdf>>. Acesso em: 14 set. 2005.

CAMPOS, Fernanda C. A; ROCHA, Ana Regina C. da. **Design Instrucional e Construtivismo: Em Busca de Modelos para o Desenvolvimento de Software**. Disponível em: <<http://www.niee.ufrgs.br/ribie98/TRABALHOS/250M.PDF>>. Acesso em: 10 març. 2005.

CANTO, Eduardo Leite do; PERUZZO, Francisco Miragaia. **Tito & Canto Química na abordagem do cotidiano**. 2. ed. São Paulo: Moderna, 2002. 584 p.

CASTRO, Glauco Cabral Monteiro de; Aguiar Teresa Cristina de. **Engenharia de Software no Desenvolvimento de Software Educacional Hiperfídia**. XXV conferencia Latinoamericana de Informática. El Marco de Clei, Paraguay, 1999. Disponível em: <<http://lsm.dei.uc.pt/ribie/docfiles/txt200332519457Engenharia%20de%20Software%20no%20Desenvolvimento%20de%20Software%20Educacional%20Hiperfídia.pdf>>. Acesso em: 21 março 2005.

COMPUTER Associates. Disponível em: <<http://www.ca.com/>>. Acesso em: 20 out. 2005.

DEITEL, H. M; DEITEL, P. J. **Java como Programar**. 4. ed. São Paulo: Makron Books, 2003.

ESMIN, Ahmed Ali Abdalla. **Modelando com UML: Unified Modeling Language**. Disponível em: <<http://www.dcc.ufla.br/infocomp/artigos/v1.1/tutorialUML.pdf>>. Acesso em: 01 agos. 2005.

FURLAN, J. D. **Modelagem em Objetos Através da UML: The Unified Modeling Language**. São Paulo: Makron Books, 1998. 329 p.

GIOVANINI, Fernando Lopes; NOVO, Karine Rodrigues; PARDO, Marília; MIYAZATO, Renata Mayumi. **Ferramenta Didática para Ensino de Química: Módulo Tabela Periódica**. 2004. 76f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2004.

JACOBSON, Ivar; BOOCH, Grady; RUMBAUGH, James. **UML Guia do Usuário**. Rio de Janeiro: Campus, 2000. 473 p.

LARMAN, Craig. **Utilizando UML e Padrões**. São Paulo: Bookman, 1999.

**LIGAÇÃO Química**. Disponível em: <[www.quiprocura.net/ligacao/](http://www.quiprocura.net/ligacao/)>. Acesso em: 06 jul. 2005.

MATOS, Alexandre Veloso de. **Unified Modeling Language Prático e Descomplicado**. 1.ed. São Paulo: Érica, 2002. 187 p.

**MODELAGEM Visual de Aplicações de Software**. Disponível em: <[http://www.microsoft.com/brasil/msdn/tecnologias/vsnet/visualstudio\\_modelagem.aspx](http://www.microsoft.com/brasil/msdn/tecnologias/vsnet/visualstudio_modelagem.aspx)>. Acesso em: 25 abr. 2005.

O PROCESSO Unificado de Desenvolvimento de Software aplicado a Projetos Pequenos. Disponível em: <<http://www.inf.pucrs.br/~eperes/upsmall.pdf>>. Acesso em: 10 març. 2005.

PRESSMAN, R. S. **Engenharia de Software**. São Paulo: Makron Books, 1995. 1056 p.

REIS, Ronaldo. **Tabela Periódica Multimídia. Moderna Multimídia: Macromédia**, 2005. 1CD-ROM.

SOUZA, Juliana Pereira. Análise de Requisitos e Projeto OO para Software Educacional de Química. In: VII SEMINÁRIO DE INICIAÇÃO CIENTÍFICA, 2005, Marília, **Resumos...** Marília: Fundação de Ensino Eurípides Soares da Rocha, 2005. p. 2.

\_\_\_\_\_. Análise de Requisitos e Projeto Orientado a Objeto Usando UML: Um Estudo de Caso. In: 13º SIMPÓSIO INTERNACIONAL DE INICIAÇÃO CIENTÍFICA DA USP, 2005, São Carlos, **Resumos...** São Carlos: Universidade de São Paulo, 2005. p. 2.

TABELA Periódica. Disponível em:

<<http://www.christus.com.br/infochristus/tabperiodica/tprincipal.htm>>. Acesso em: 06 jul. 2005.

TAPAS Development Method. Disponível em:

<<http://www.opentapas.org/docs/method.html>>. Acesso em: 02 jul. 2005.

TREINAMENTOS Conceituais. Disponível em:

<<http://www.squadra.com.br/website/servicos/treinamento/b17.html>>. Acesso em: 15 ago. 2005.

UML Tutorial: Disponível em: <<http://uml.tutorials.treime.com/>>. Acesso em: 11 abr. 2005.

UNIFIED Modeling Language. Disponível em: <<http://www.rational.com/uml>>. Acesso em: 23 maio 2005.

VALE, Christiano. **UML para Java**. Disponível em:

<[http://www.guj.com.br/content/articles/uml/guj\\_uml\\_java.pdf](http://www.guj.com.br/content/articles/uml/guj_uml_java.pdf)>. Acesso em: 01 ago. 2005.

## APÊNDICE A - Modelo Entidade Relacionamento (Lógico)

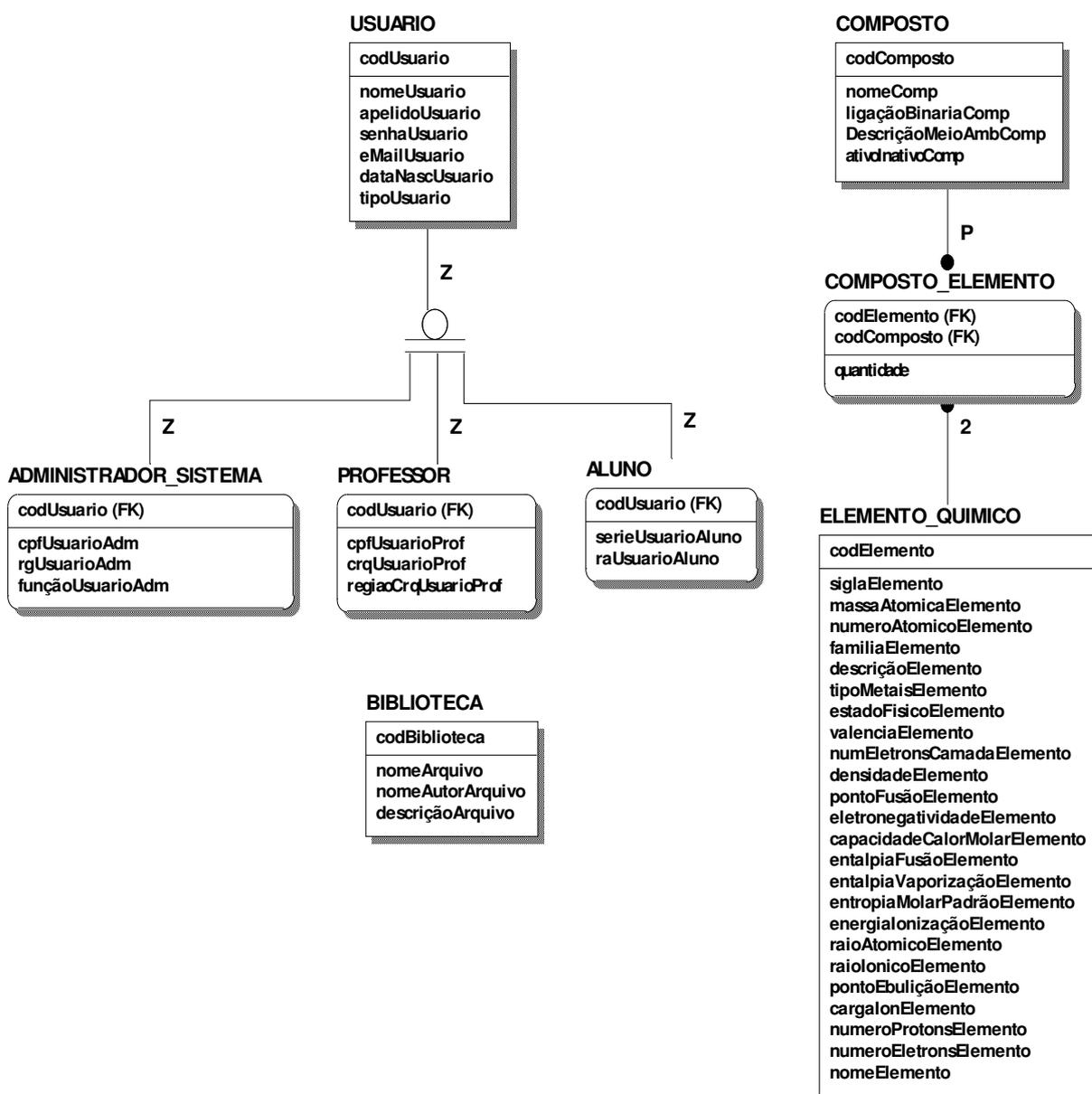


Figura A1 - MER Lógico.

Este diagrama refere-se ao modelo de entidade-relacionamento do SEQ. Apresenta a mesma estrutura de tabelas e dados do diagrama de classes preliminar, e para este trabalho, seu objetivo é de ajudar futuros programadores desenvolverem a implementação funcional do sistema. Foi modelado pela ferramenta COMPUTER ASSOCIATES ERWIN 4.0.

## APÊNDICE B - Descrição dos Casos de Uso, Diagrama de Sequência e de Atividade

Nesta descrição apresenta-se o termo Usuário, quando este não tiver restrições de acessos de ambientes, ou seja, Usuário poderá ser Aluno, Professor ou Administrador de Sistema. Serão apresentados para cada descrição os diagramas de sequência e atividade referente ao caso de uso.

Para ajudar os desenvolvedores a interpretar os casos de uso, esses devem ser classificados como:

- **Caso de Uso primário:** representa processo principal.
- **Caso de Uso secundário:** representa processo menos importante ou raro.
- **Caso de Uso opcional:** representa processo que pode não ser considerado.

### 1) Nome do Caso de Uso: **Logar no Sistema**

Atores que Participam: Usuário, Aluno, Professor e Administrador do Sistema.

Quem inicia o caso: Usuário.

Tipo: Primário.

Descrição do caso de Uso: Ao iniciar o software o usuário terá três opções: logar no sistema (caso seja cadastrado), se cadastrar ou solicitar a senha (caso tenha esquecido).

Se escolher “Logar no Sistema”, terá que inserir seu apelido e senha. Esse próprio caso irá validar se os campos foram preenchidos corretamente. Se os campos estiverem corretos (verdadeiro), o sistema automaticamente passará por uma autenticação de senha, de acordo com o caso de uso “Autenticar Senha”, caso contrário, o sistema apresentará uma mensagem dizendo que os campos foram preenchidos incorretamente e o usuário poderá ficar tentando inserir seu apelido e senha, ou acessar a opção Esqueci Minha Senha, que após o fracasso do login, aparecerá automaticamente, de acordo com o caso de uso “Solicitar Senha”.

Se escolher se cadastrar, o caso de uso referente é: “Cadastrar Usuário” ou se a escolha for esqueci senha, o caso de uso referente é: “Solicitar Senha”.

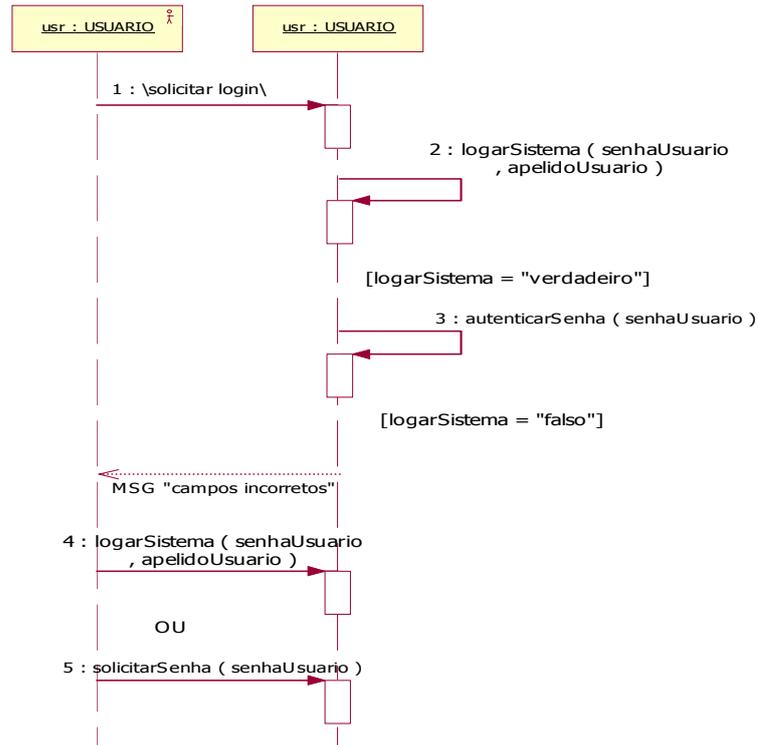


Figura A2 - Diagrama de Sequência do caso de uso “Logar no Sistema”.

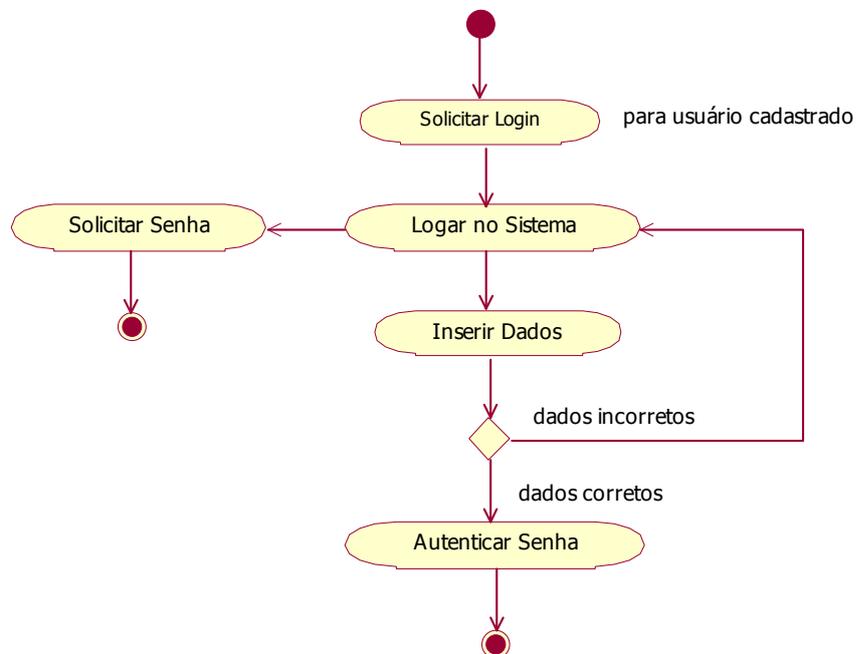


Figura A3 - Diagrama de Atividade do caso de uso “Logar no Sistema”.

## 2) Nome do Caso de Uso: **Autenticar Senha**

Atores que Participam: Usuário, Aluno, Professor e Administrador do Sistema.

Quem inicia o caso: Usuário.

Tipo: Primário.

Descrição do caso de Uso: Após a validação dos campos (de acordo com o caso de uso “Logar no Sistema”), automaticamente o sistema passará por uma Autenticação. Essa é uma validação da senha referente ao apelido.

Se a senha estiver correta, o usuário entrará em sua área restrita, referente ao seu tipo (Professor, Aluno ou Administrador). Caso for falsa, o sistema apresentará uma mensagem de senha incorreta e poderá ficar tentando logar.

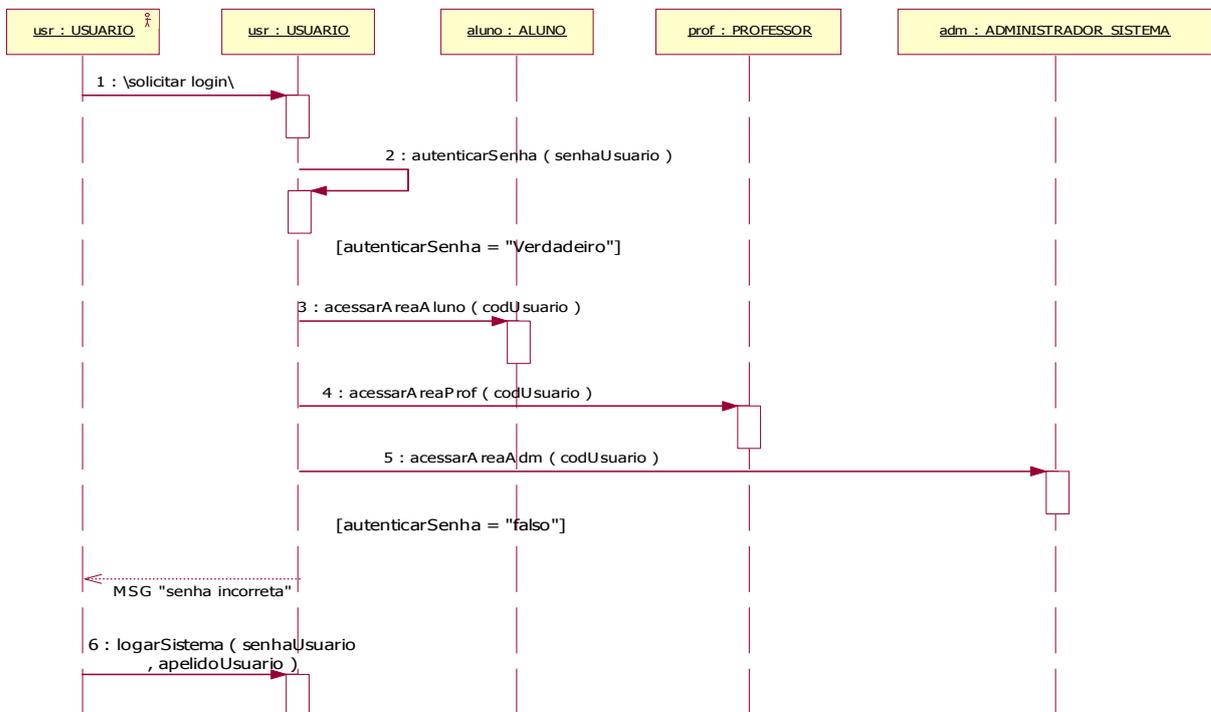


Figura A4 - Diagrama de Seqüência do caso de uso “Autenticar Senha”.

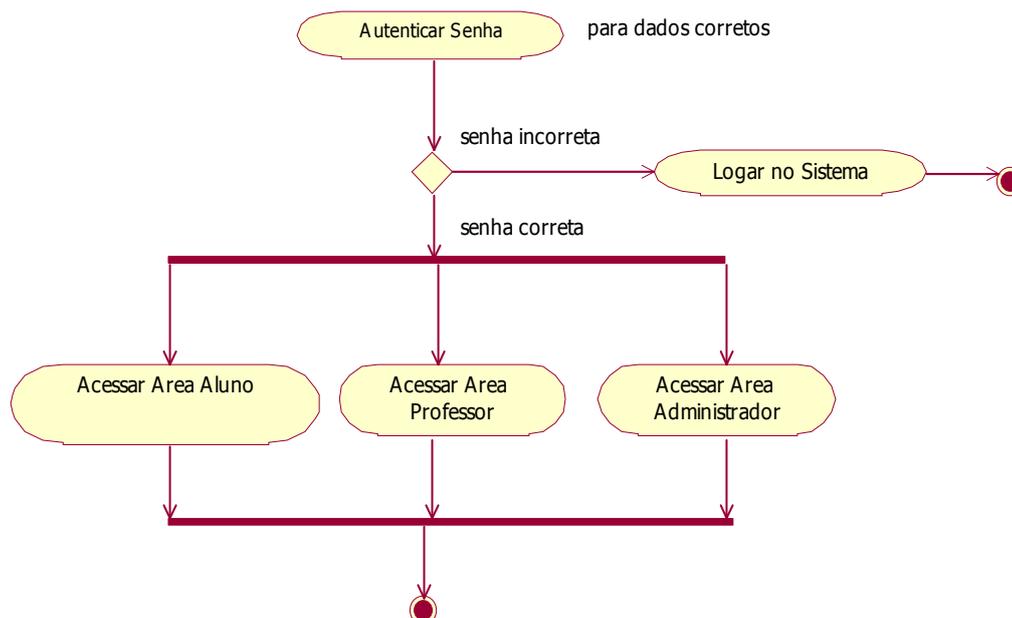


Figura A5 - Diagrama de Atividade do caso de uso "Autenticar Senha".

### 3) Caso de Uso: **Solicitar Senha**

Atores que Participam: Usuário, Aluno, Professor e Administrador do Sistema.

Quem inicia o caso: Usuário.

Tipo: Secundário.

Descrição: Quando o usuário for logar no sistema e tiver esquecido sua senha, ele poderá solicitá-la, selecionando uma opção denominada "esqueci minha senha", onde ao ser selecionada, terá um campo para inserir seu nome e e-mail, assim, automaticamente será validado e conseqüentemente será enviado por e-mail sua senha e apelido.

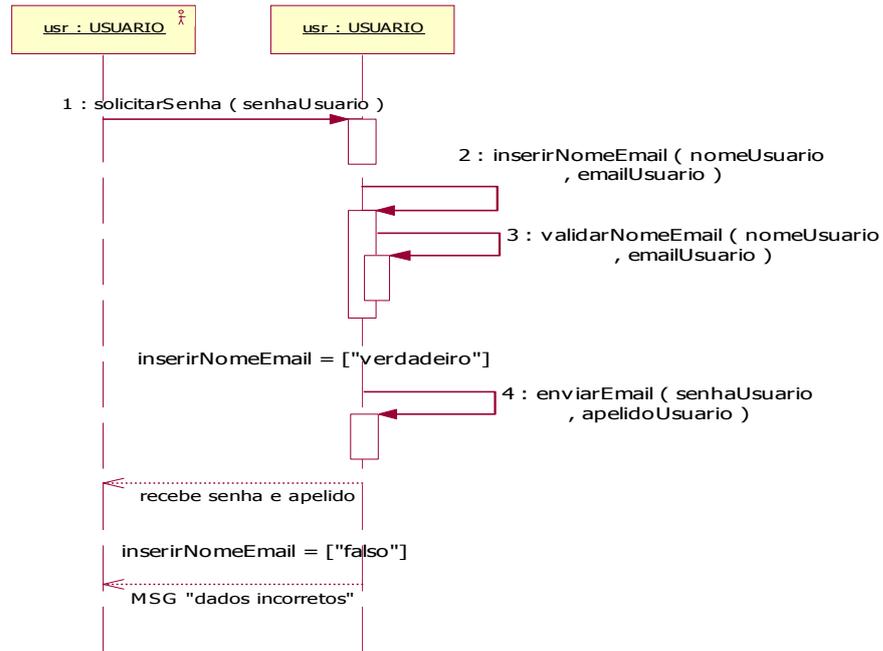


Figura A6 - Diagrama de Seqüência do caso de uso “Solicitar Senha”.

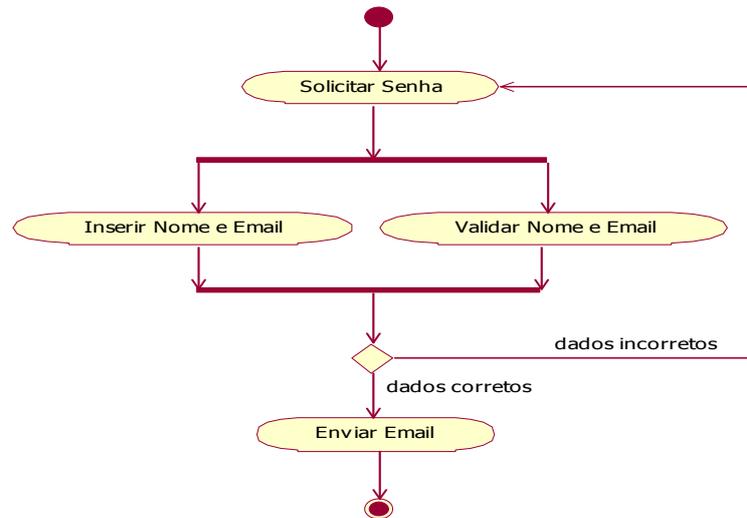


Figura A7 - Diagrama de Atividade do caso de uso “Solicitar Senha”.

#### 4) Caso de Uso: **Cadastrar Usuário**

Atores que Participam: Usuário, Aluno, Professor e Administrador do Sistema.

Quem inicia o caso: Usuário.

Tipo: Primário.

Descrição: O usuário, ao entrar no software, se não for cadastrado, poderá solicitá-lo. Automaticamente abrirá um ambiente onde constarão os campos para serem preenchidos

como nome, e-mail, data de nascimento, apelido e senha. Conterá um campo aonde o usuário irá se classificar se é Aluno, Professor ou Administrador, para cada tipo de Usuário terão campos referentes a eles.

Caso o Usuário for Aluno, seu cadastro será feito automaticamente. Durante e após será feito uma validação de cadastro, de acordo com o caso de uso “Validar Cadastro Aluno”, caso contrário será enviado por e-mail, uma solicitação para o Administrador (antecessor) cadastrar, de acordo com o caso de uso “Controle do Sistema”.

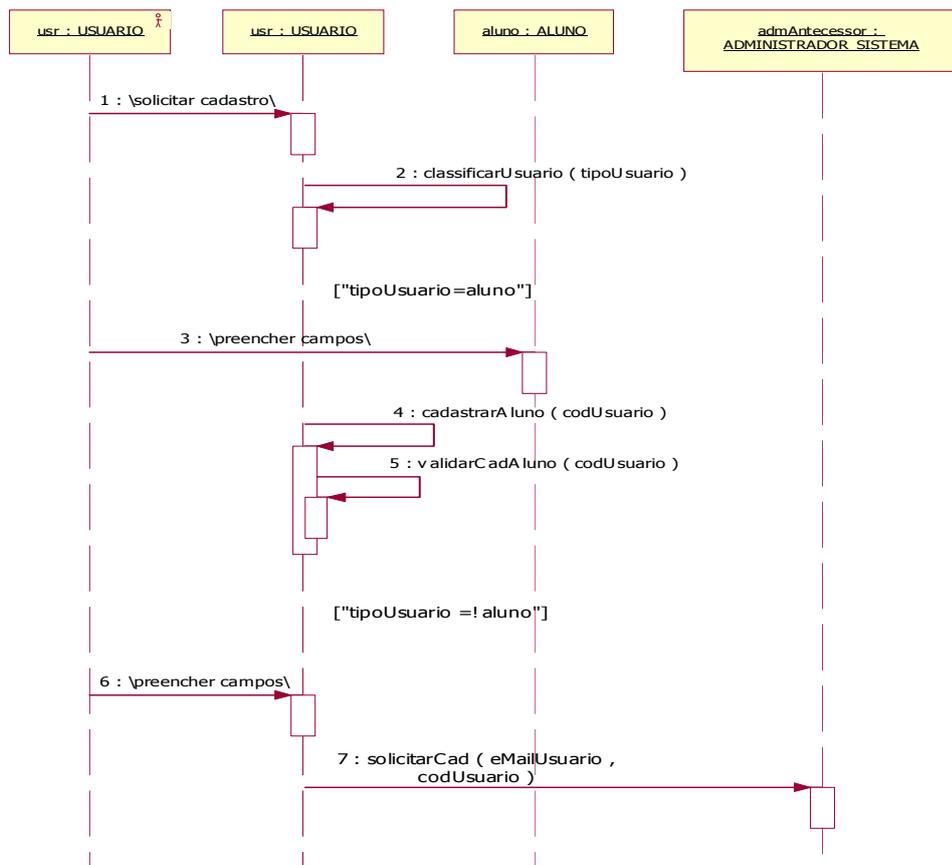


Figura A8 - Diagrama de Sequência do caso de uso “Cadastrar Usuário”.

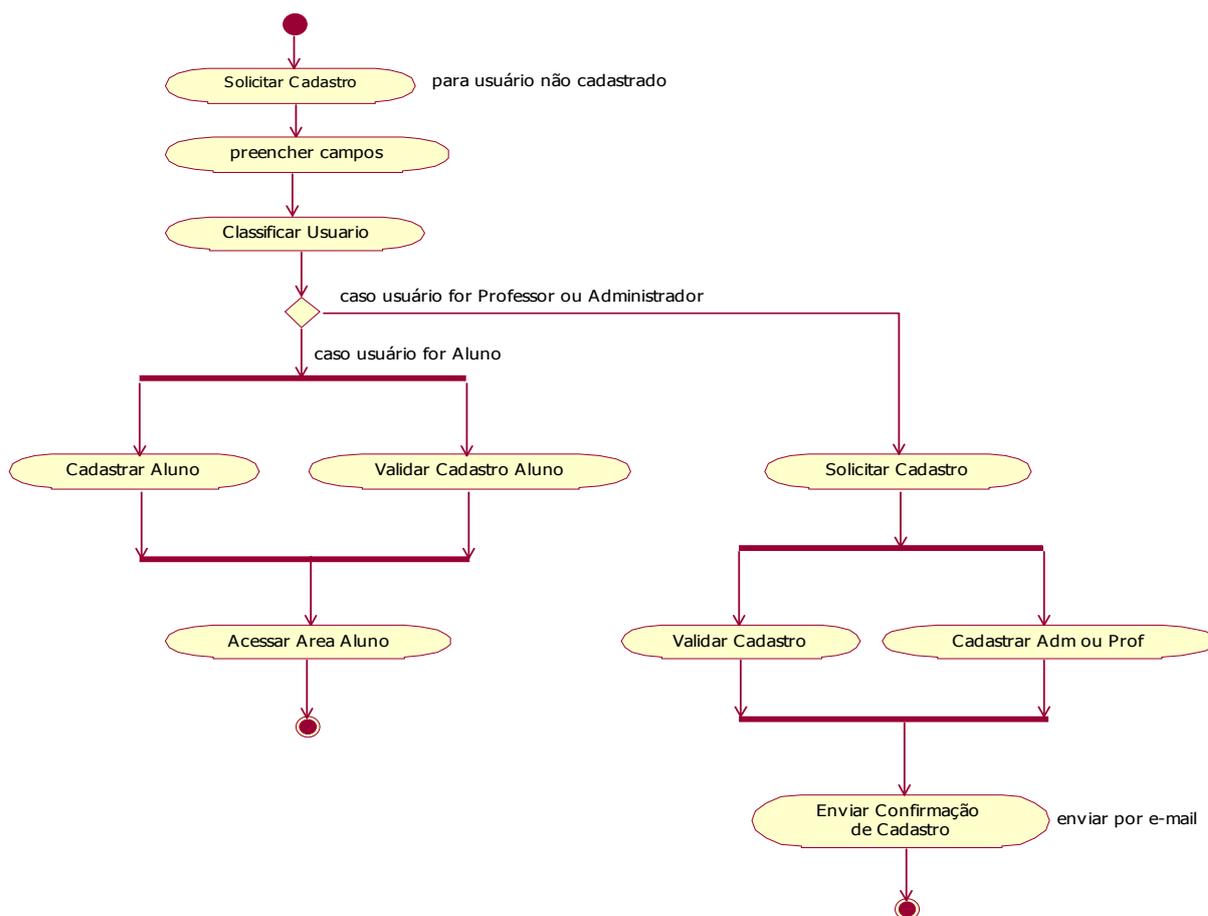


Figura A9 - Diagrama de Atividade dos casos de uso “Cadastrar Usuário”, “Validar Cadastro Aluno”, “Controlar o Sistema” e “Validar Cadastros”.

### 5) Caso de Uso: **Validar Cadastro Aluno**

Atores que Participam: Usuário, Aluno, Professor e Administrador do Sistema.

Quem inicia o caso: Aluno.

Tipo: Primário.

Descrição: Após a identificação do Usuário, se for Aluno, o sistema automaticamente passará por uma validação, para confirmar se os campos estão todos preenchidos e consistentes. Conseqüentemente o Aluno terá uma confirmação de cadastro e poderá entrar em sua área, de acordo com o caso de uso “Acessar Área Aluno”.

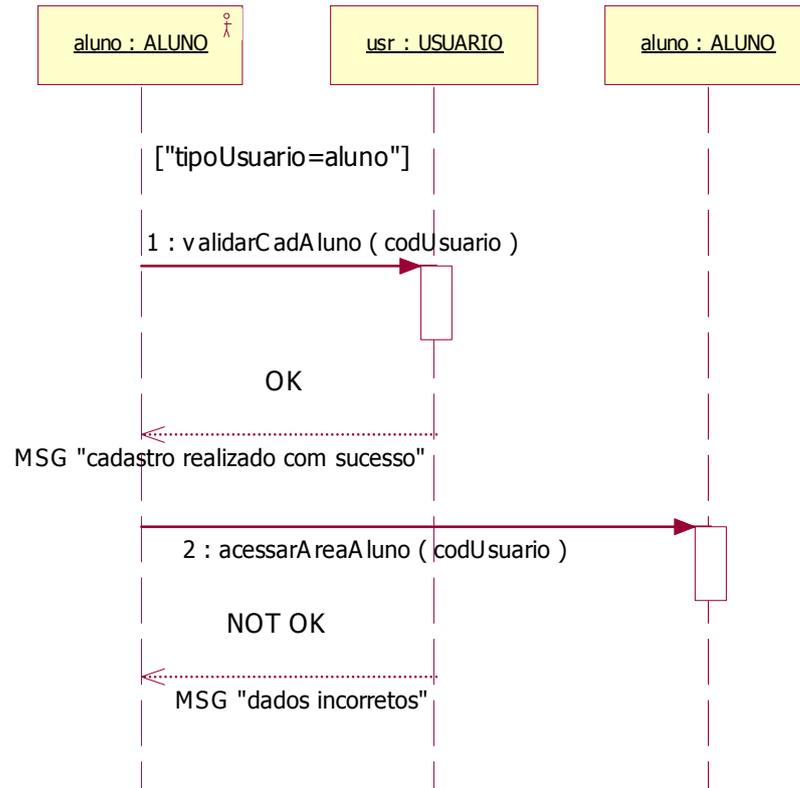


Figura A10 - Diagrama de Seqüência do caso de uso “Validar Cadastro Aluno”.

#### 6) Caso de Uso: **Controlar o Sistema**

Atores que Participam: Administrador do Sistema e Professor.

Quem inicia o caso: Administrador do Sistema ou Professor.

Tipo: Primário.

Descrição: Após a solicitação de cadastro e identificação do Usuário, se for Administrador ou Professor, eles solicitarão o cadastro, onde o sistema enviará um e-mail para o Administrador antecessor (administrador antecessor é o ultimo cadastrado) para que esse possa validar seu cadastro, de acordo com o caso de uso “Validar Cadastros”.

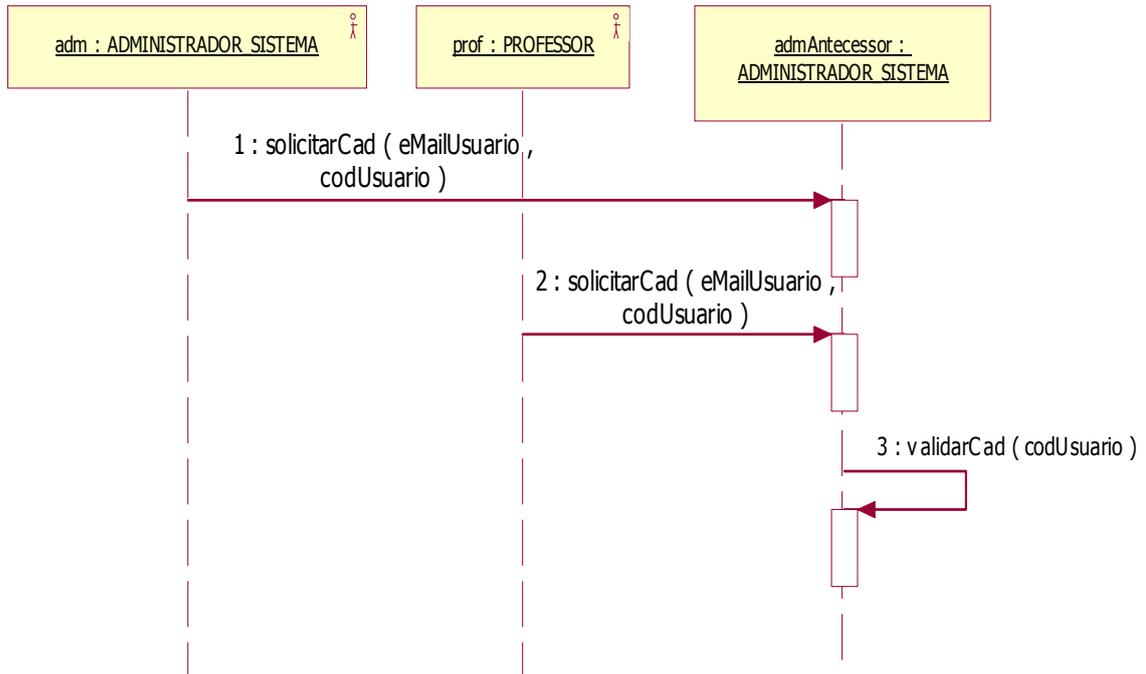


Figura A11 - Diagrama de Seqüência do caso de uso “Controlar o Sistema”.

#### 7) Caso de Uso: **Validar Cadastros**

Atores que Participam: Professor e Administrador do Sistema.

Quem inicia o caso: Administrador do Sistema.

Tipo: Primário.

Descrição: O Administrador antecessor validará a autenticação dos campos desses usuários (Professor ou Administrador), conseqüentemente os cadastrando e enviando um e-mail de confirmação para esses usuários que o solicitou mostrando sua senha e apelido.

Portanto, os cadastros do Professor e Administrador não serão feitos automaticamente.

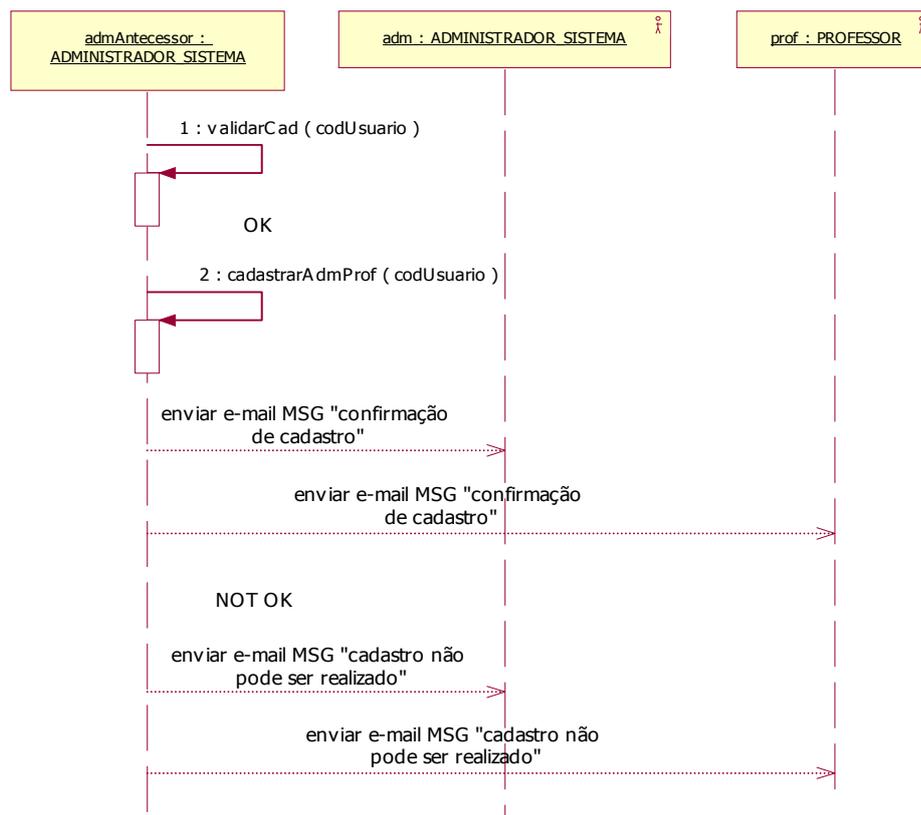


Figura A12 - Diagrama de Sequência do caso de uso “Validar Cadastros”

#### 8) Caso de Uso: **Acessar Área Administrador**

Atores que Participam: Administrador do Sistema.

Quem inicia o caso: Administrador do Sistema.

Tipo: Primário

Descrição: Após a Autenticação de Senha, de acordo com o caso de uso “Autenticar Senha” o Administrador acessará a sua área restrita. Essa área é o ambiente em que o Administrador visualizara as funcionalidades que poderá executar no SEQ.

Ele poderá acessar a Área do Professor, conseqüentemente a Área do Aluno, Controlar o Sistema e Controlar o Cadastro de usuário.

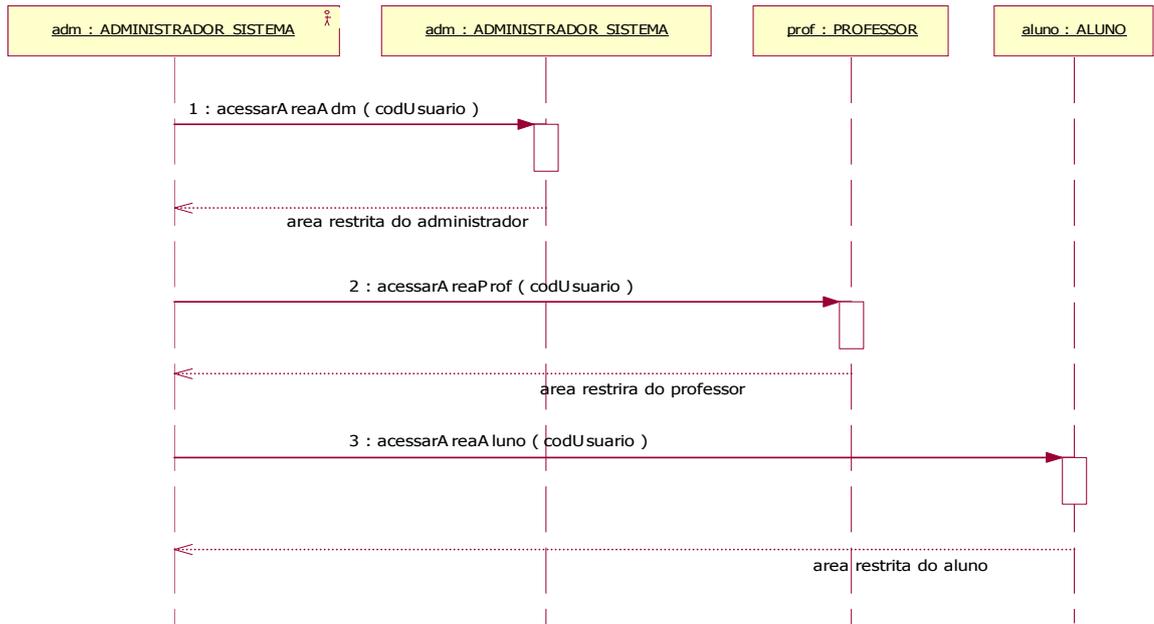


Figura A13 - Diagrama de Seqüência do caso de uso “Acessar Área Administrador”.

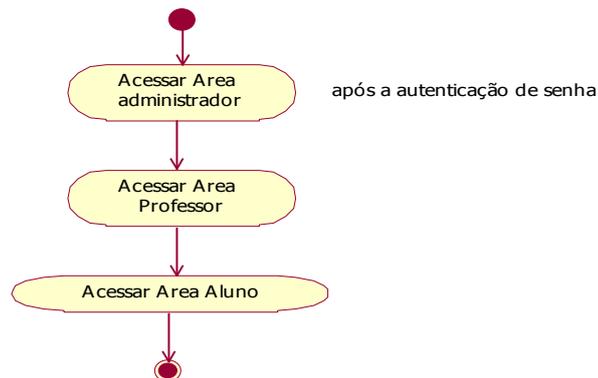


Figura A14 - Diagrama de Atividade do caso de uso “Acessar Área Administrador”.

### 9) Caso de Uso: **Controlar Cadastro de Usuário**

Atores que Participam: Administrador do Sistema.

Quem inicia o caso: Administrador do Sistema.

Tipo: Primário.

Descrição: Após as validações dos cadastros dos Usuários (Professor, Aluno ou Administrador), o Administrador poderá excluir, alterar ou consultar os cadastrados feitos.

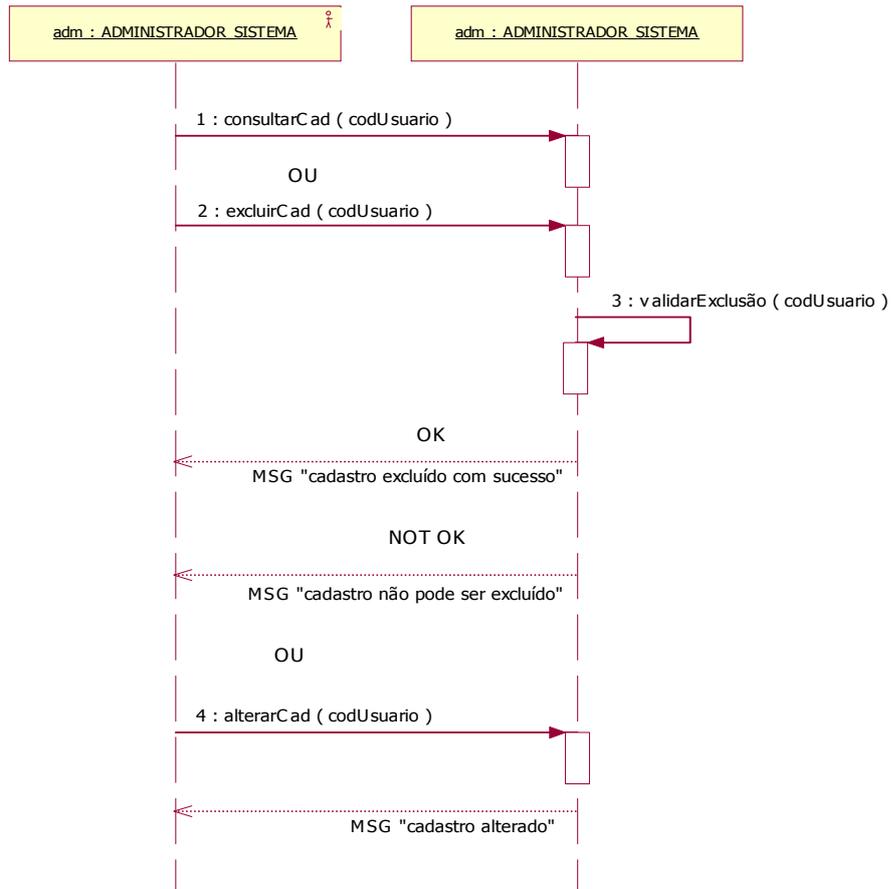


Figura A15 - Diagrama de Seqüência do caso de uso “Controlar Cadastro de Usuário”.

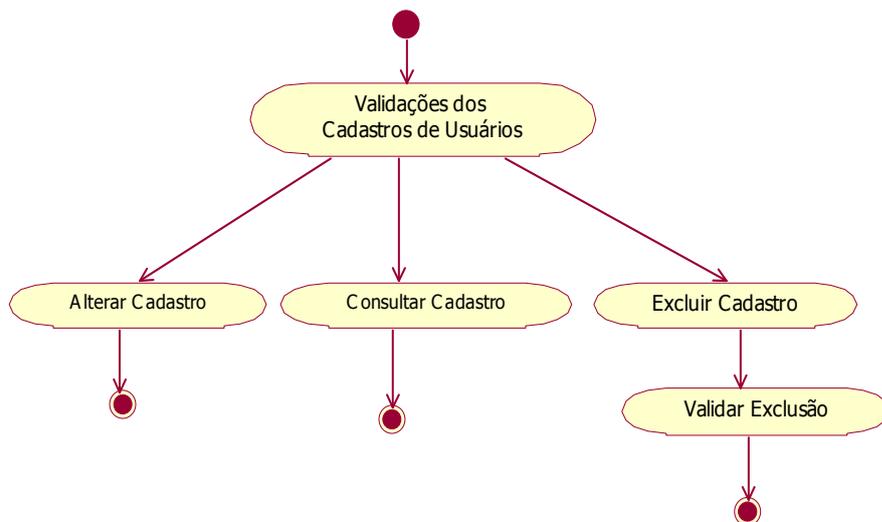


Figura A16 - Diagrama de Atividade do caso de uso “Controlar Cadastro de Usuário”.

### 10) Caso de Uso: **Acessar Área Professor**

Atores que Participam: Professor.

Quem inicia o caso: Professor.

Tipo: Primário

Descrição: Após a Autenticação de Senha, de acordo com o caso de uso “Autenticar Senha” o Professor acessará a sua área restrita. Essa área é o ambiente em que o Professor visualizará as funcionalidades que poderá executar no SEQ.

Além de poder acessar suas funcionalidades particulares, ele poderá também acessar a área do Aluno, de acordo com o caso de uso “Acessar Área Aluno”.

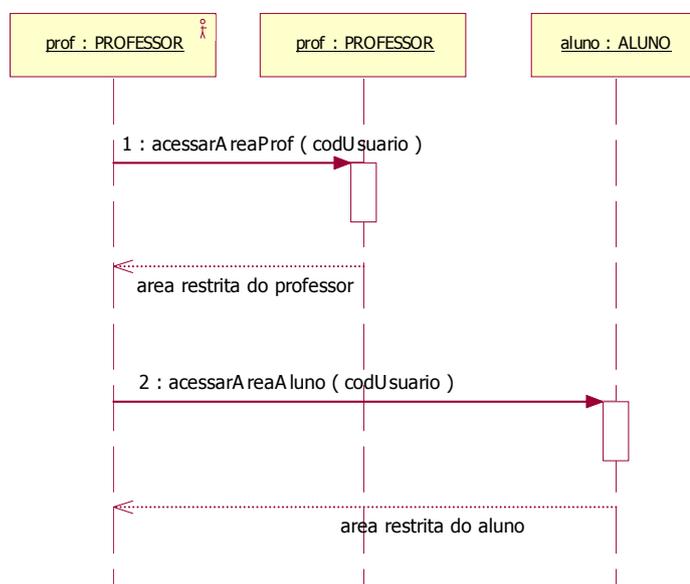


Figura A17 - Diagrama de Seqüência do caso de uso “Acessar Área Professor”

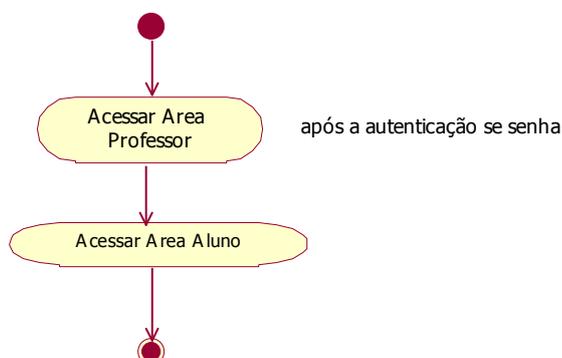


Figura A18 - Diagrama de Atividade do caso de uso “Acessar Área Professor”

### 11) Caso de Uso: **Cadastrar Composto Ativo**

Atores que Participam: Professor e Administrador do Sistema.

Quem inicia o caso: Professor ou Administrador do Sistema.

Tipo: Primário.

Descrição: O Professor ou Administrador poderão cadastrar os compostos que serão usados para que os usuários possam desenvolver as ligações químicas. No cadastro conterá os dados básicos referentes ao composto (nome, descrição, etc...), além das funcionalidades: salvar, excluir, alterar e consultar compostos ativos, referente ao caso de uso “Controlar Cadastro de Composto”. Durante e após a solicitação do cadastro, o sistema passará por uma validação, de acordo com o caso de uso “Validar Cadastro Composto Ativo”.

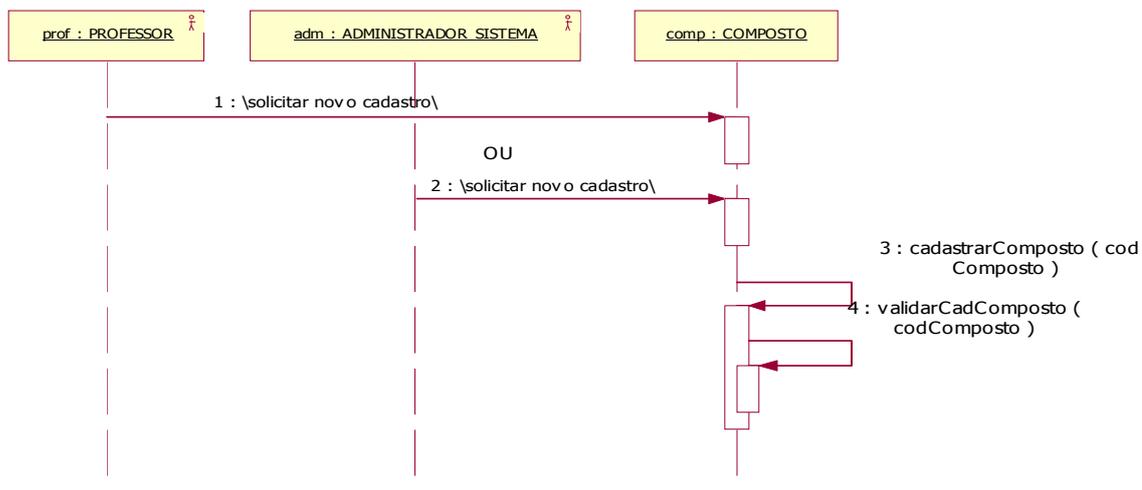


Figura A19 - Diagrama de Seqüência do caso de uso “Cadastrar Composto Ativo”.



Figura A20 - Diagrama de Atividade dos casos de uso “Cadastrar Composto Ativo” e “Validar Cadastro Composto Ativo”.

### 12) Caso de Uso: **Validar Cadastro Composto Ativo**

Atores que Participam: Professor e Administrador do Sistema.

Quem inicia o caso: Professor ou Administrador do Sistema.

Tipo: Primário.

Descrição: Após a solicitação de cadastro de composto ativo, o sistema passará por uma validação, essa usará propriedades da ligação iônica, ligação covalente e ligação metálica, confirmando se o composto a ser cadastrado esta nas normas da química.

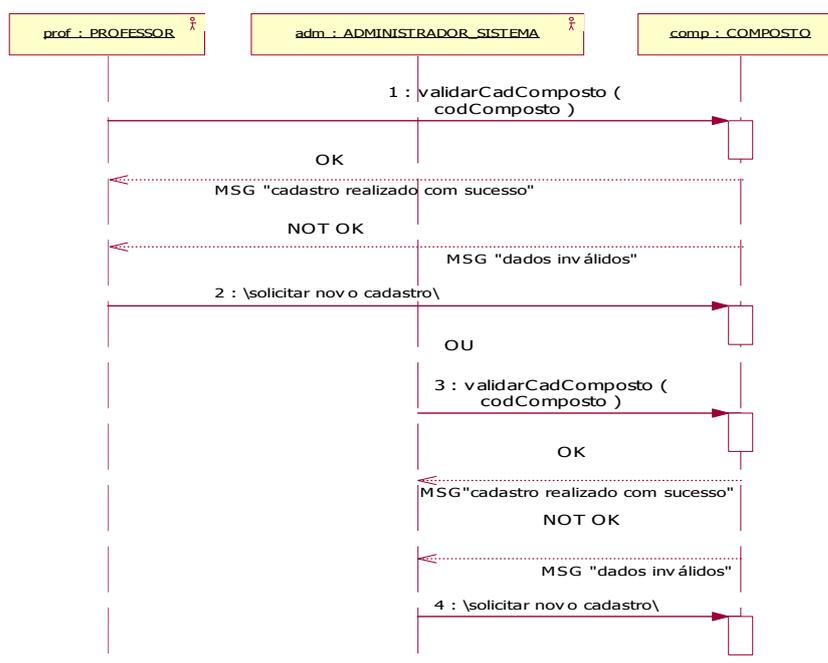


Figura A21 - Diagrama de Seqüência do caso de uso “Validar Cadastro Composto Ativo”.

### 13) Caso de Uso: **Controlar Cadastro de Composto**

Atores que Participam: Usuário, Professor e Administrador do Sistema.

Quem inicia o caso: Aluno ou Professor ou Administrador do Sistema.

Tipo: Primário.

Descrição: Após a validação do cadastro de composto ativo e inativo, se o composto for ativo, o Professor ou Administrador poderão excluí-lo, alterá-lo ou consultá-lo o composto cadastrado, caso contrário, se o composto for inativo, o Aluno (em sua área restrita) poderá excluí-lo, alterá-lo e consultá-lo o composto Inativo cadastrado.

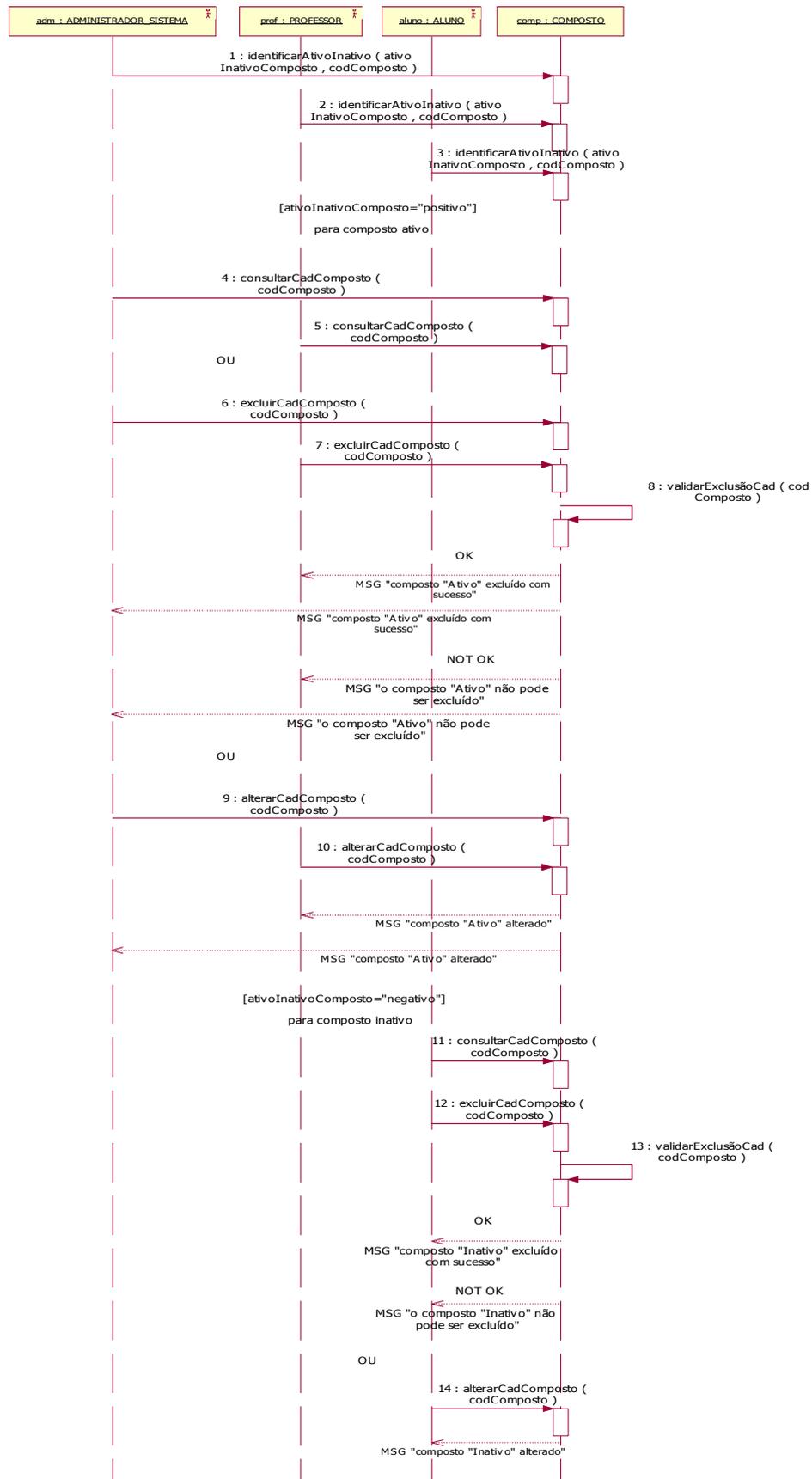


Figura A22 - Diagrama de Sequência do caso de uso “Controlar Cadastro de Composto”

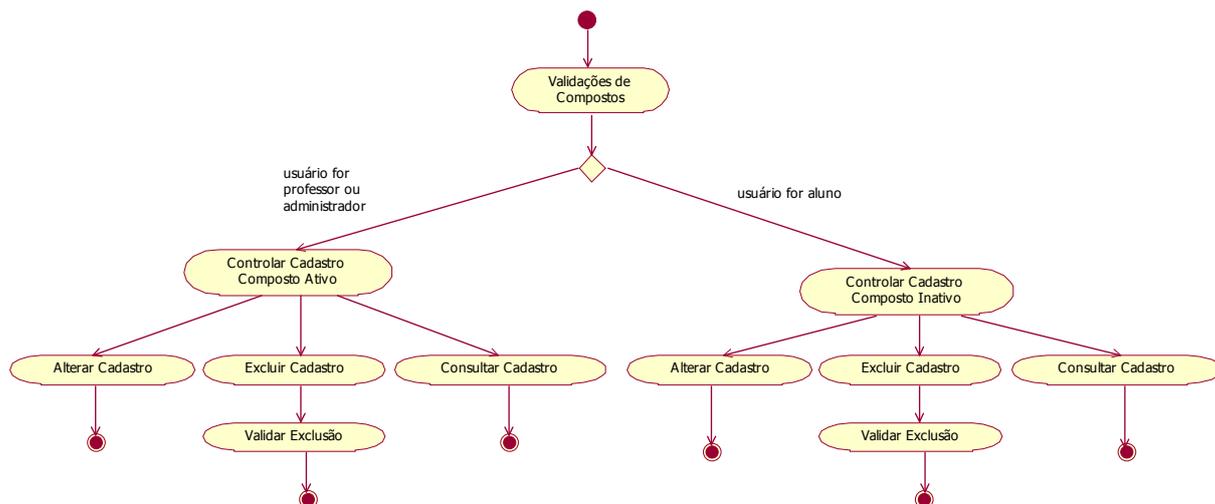


Figura A23 - Diagrama de Atividade do caso de uso “Controlar Cadastro de Composto”

#### 14) Caso de Uso: **Alterar Cadastro de Composto Inativo**

Atores que Participam: Professor e Administrador do Sistema.

Quem inicia o caso: Professor ou Administrador do Sistema.

Tipo: Primário.

Descrição: O professor ou Administrador poderão acessar o cadastro do composto inativo que os usuários criaram no ambiente laboratório, podendo posteriormente corrigi-lo e salvá-lo, alterando sua base de dados para composto ativo.

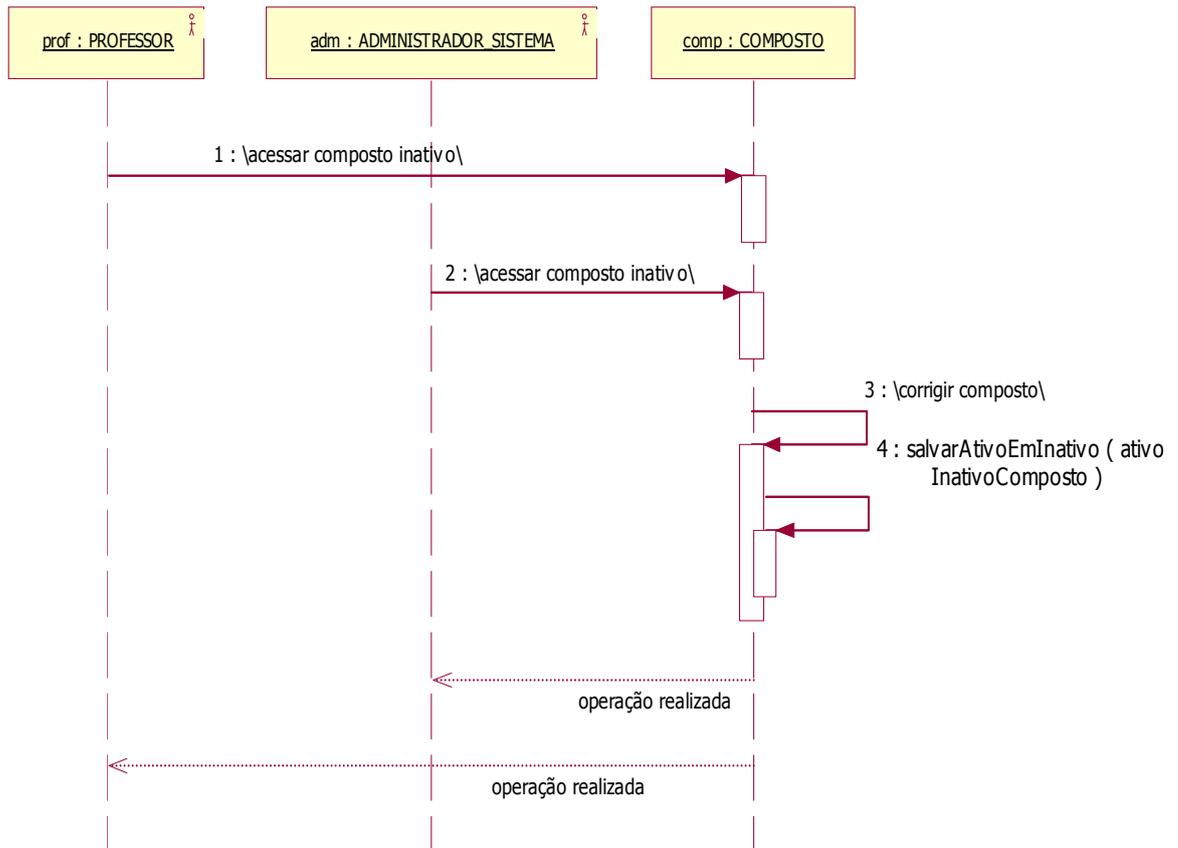


Figura A24 - Diagrama de Sequência do caso de uso “Alterar Cadastro de Composto Inativo”

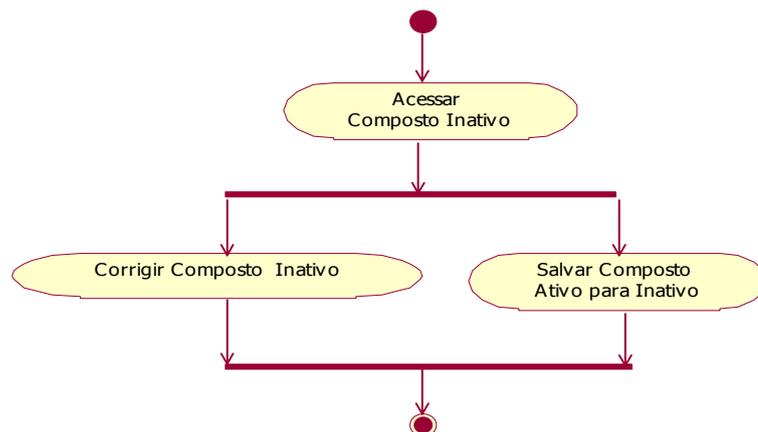


Figura A25 - Diagrama de Atividade do caso de uso “Alterar Cadastro de Composto Inativo”

### 15) Caso de Uso: **Cadastrar Arquivo PDF**

Atores que Participam: Professor e Administrador do Sistema.

Quem inicia o caso: Professor ou Administrador do Sistema.

Tipo: Secundário.

Descrição: O Professor ou Administrador poderão cadastrar diversos arquivos PDFs em uma base de dados, para que o Usuário possa acessar na visão biblioteca. Essa base de dados terá os seguintes atributos: Nome, Nome do Autor e Descrição do Arquivo.

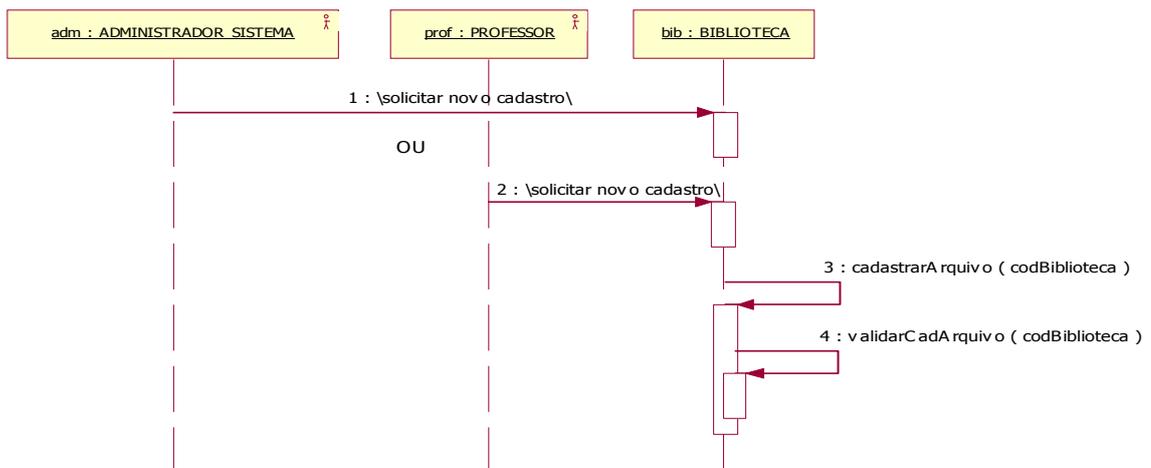


Figura A26 - Diagrama de Seqüência do caso de uso "Cadastrar Arquivo PDF"

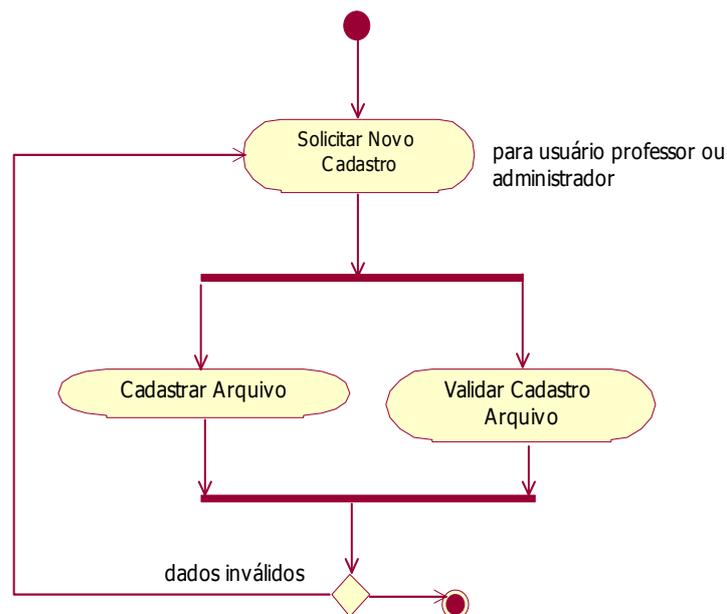


Figura A27 - Diagrama de Atividade dos casos de uso "Cadastrar Arquivo PDF" e "Validar Cadastro PDF".

### 16) Caso de Uso: **Validar Cadastro PDF**

Atores que Participam: Professor e Administrador do Sistema.

Quem inicia o caso: Professor ou Administrador do Sistema.

Tipo: Secundário.

Descrição: Durante e Após o Cadastro de PDFs, haverá uma validação para conferir se os atributos estão preenchidos corretamente.

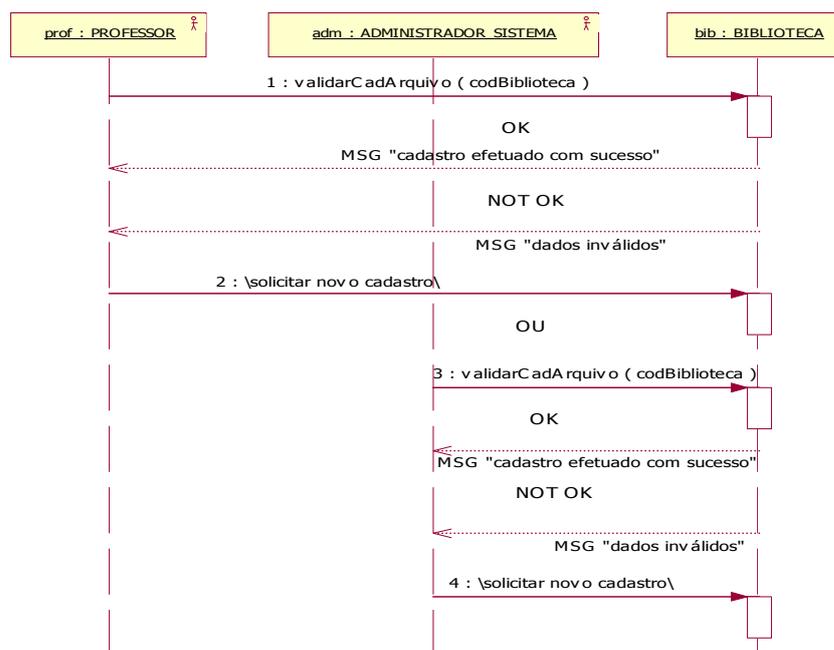


Figura A28 - Diagrama de Seqüência do caso de uso “Validar Cadastro PDF”

### 17) Caso de Uso: **Controlar Cadastro de Arquivo**

Atores que Participam: Professor e Administrador do Sistema.

Quem inicia o caso: Professor ou Administrador do Sistema.

Tipo: Secundário.

Descrição: Após a validação do cadastro de Arquivo, o Administrador ou Professor poderá excluí-lo, alterá-lo ou consultá-lo.

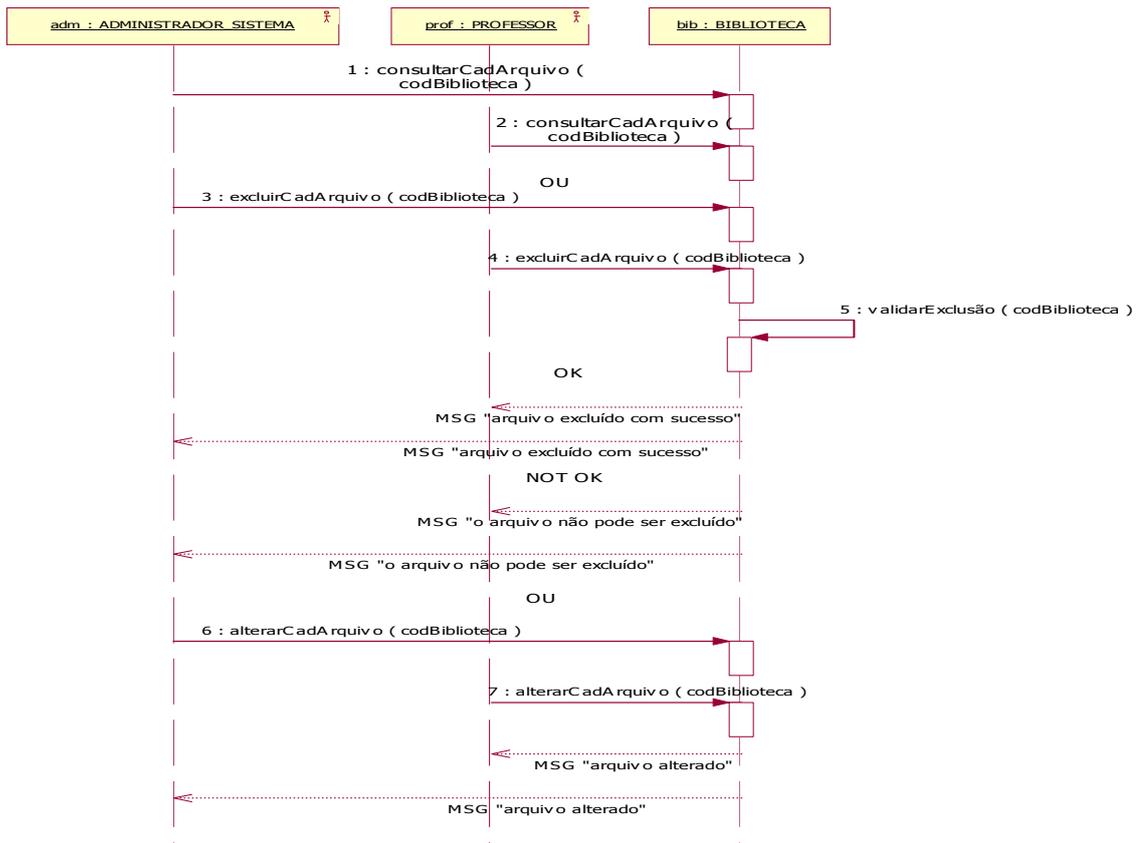


Figura A29 - Diagrama de Sequência do caso de uso “Controlar Cadastro de Arquivo”.

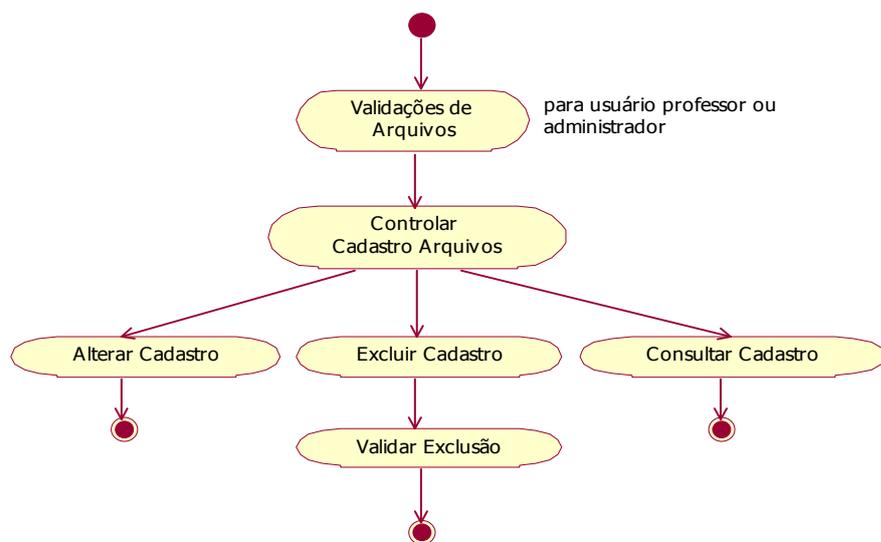


Figura A30 - Diagrama de Atividade do caso de uso “Controlar Cadastro de Arquivo”.

### 18) Caso de Uso: **Cadastrar Elemento Químico**

Atores que Participam: Professor e Administrador do Sistema.

Quem inicia o caso: Professor ou Administrador do Sistema.

Tipo: Primário.

Descrição: O Professor ou Administrador, poderão cadastrar os Elementos Químicos, preenchendo um questionário de atributos referentes a ele. Durante e após o cadastro, haverá uma validação de acordo com o caso de uso “Validar Cadastro Elementos”. Este será usado (no armário e na mesa), quando os usuários o escolherem para gerar o composto.

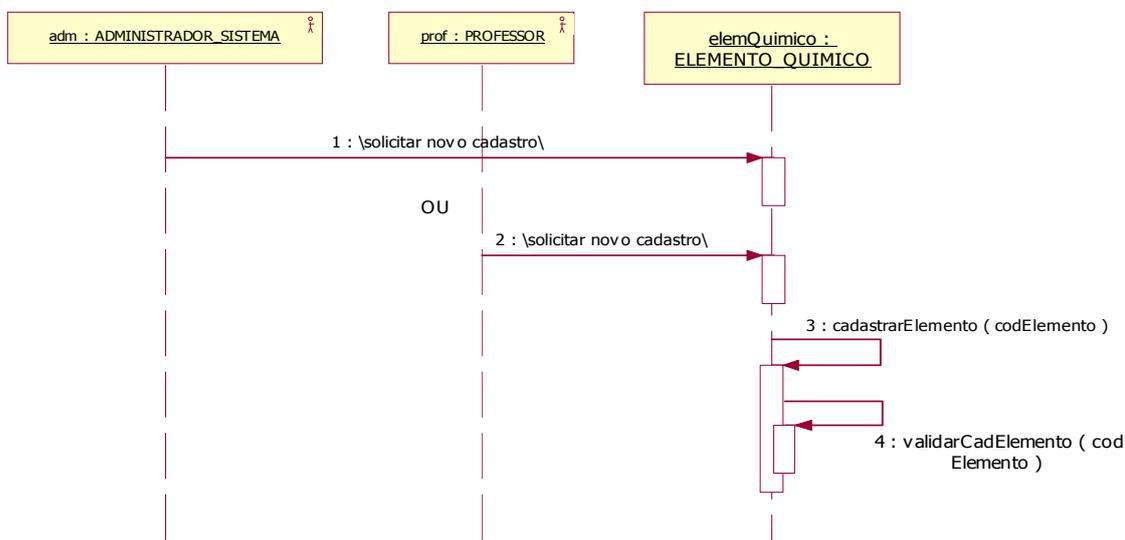


Figura A31 - Diagrama de Seqüência do caso de uso “Cadastrar Elemento Químico”.

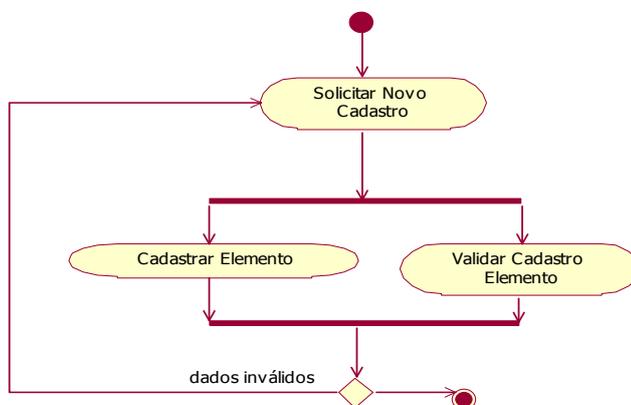


Figura A32 - Diagrama de Atividade dos casos de uso “Cadastrar Elemento Químico” e “Validar Cadastro Elemento”.

### 19) Caso de Uso: **Validar Cadastro Elemento**

Atores que Participam: Professor e Administrador do Sistema.

Quem inicia o caso: Professor ou Administrador do Sistema.

Tipo: Primário.

Descrição: Após o Professor ou Administrador inserirem os campos pedidos pelo cadastro de elemento químico e solicitar o cadastro, o sistema passara por uma validação para verificar se os campos foram preenchidos corretamente.

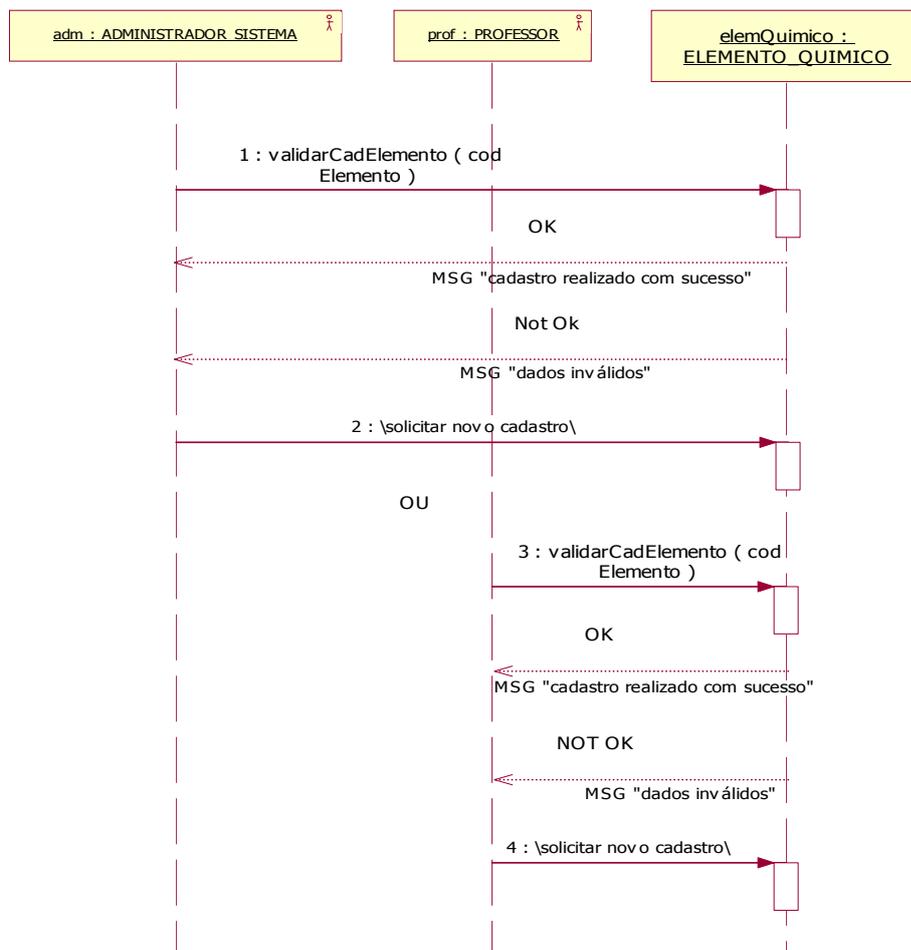


Figura A33 - Diagrama de Seqüência do caso de uso “Validar Cadastro Elemento”.

### 20) Caso de Uso: **Controlar Cadastro de Elemento**

Atores que Participam: Professor e Administrador do Sistema.

Quem inicia o caso: Professor ou Administrador do Sistema.

Tipo: Primário.

Descrição: Após a validação do cadastro de Elemento, o Administrador ou Professor poderá excluí-lo, alterá-lo ou consultá-lo.

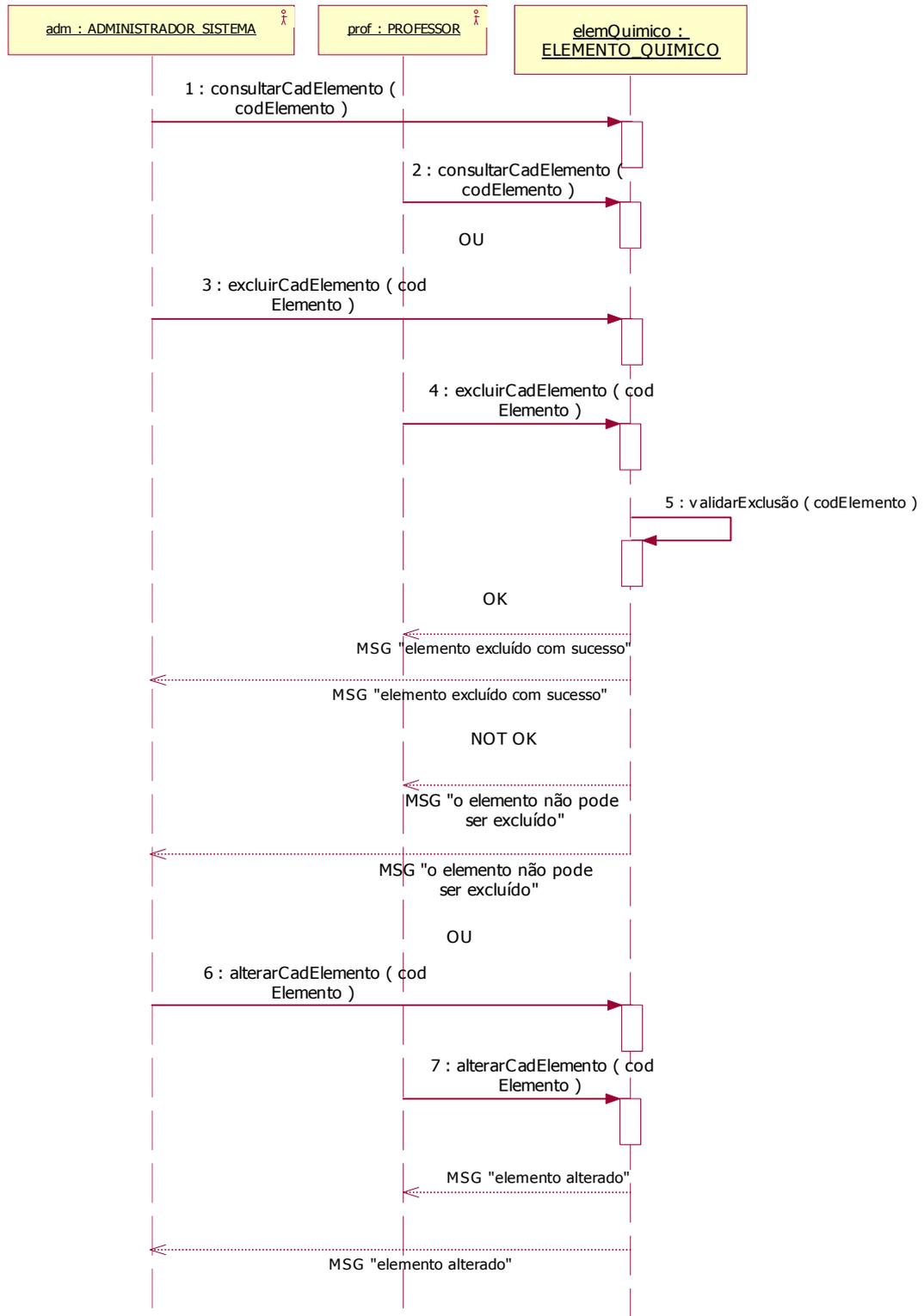


Figura A34 - Diagrama de Seqüência do caso de uso “Controlar Cadastro de Elemento”.

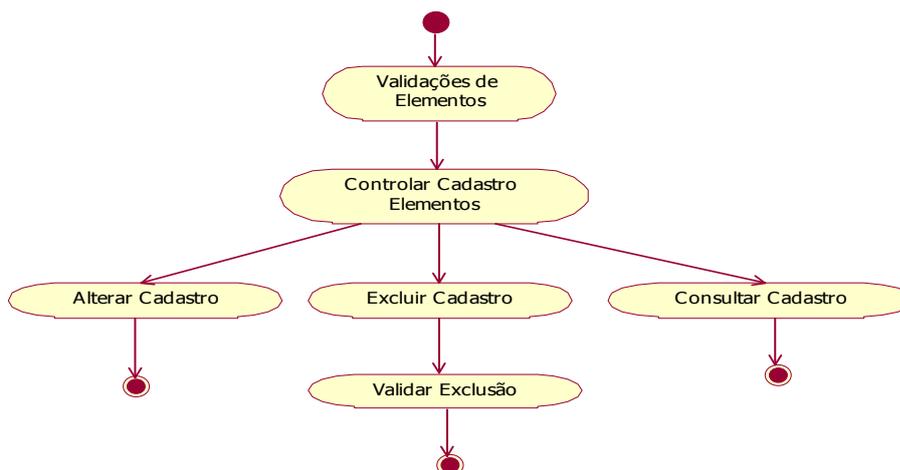


Figura A35 - Diagrama de Atividade do caso de uso “Controlar Cadastro de Elemento”.

### 21) Caso de Uso: **Acessar Área Aluno**

Atores que Participam: Usuário, Aluno.

Quem inicia o caso: Aluno.

Tipo: Primário

Descrição: Após a Autenticação de Senha, de acordo com o caso de uso “Autenticar Senha” ou após a confirmação do cadastro, de acordo com o caso de uso “Validar Cadastro Aluno”, o Aluno acessará a sua área, que será o Laboratório Químico. Todos os usuários poderão ter acesso a essa área.

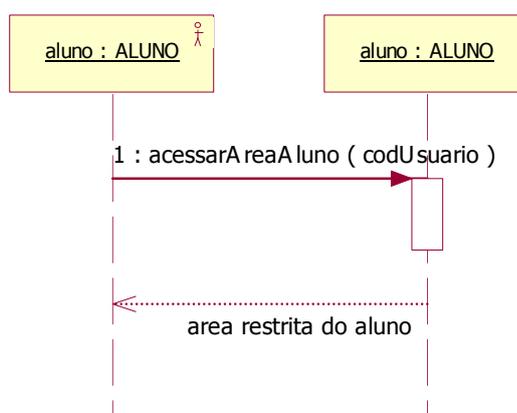


Figura A36 - Diagrama de Seqüência do caso de uso “Acessar Área Aluno”.

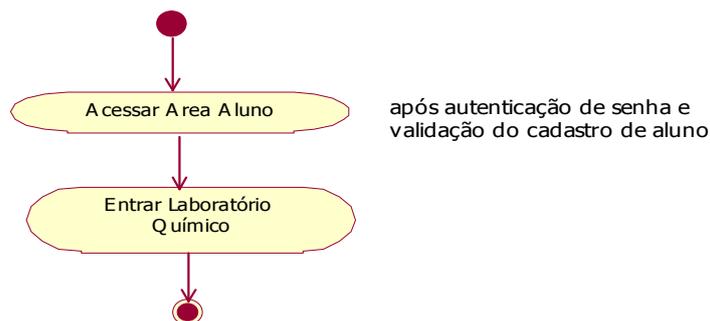


Figura A37 - Diagrama de Atividade do caso de uso “Acessar Área Aluno”.

## 22) Caso de Uso: **Acessar Objetos Virtuais**

Atores que Participam: Usuário, Aluno, Professor, Administrador do Sistema, Laboratório, Biblioteca, Mesa e Armário.

Quem inicia o caso: Usuário.

Tipo: Primário.

Descrição: Qualquer tipo de Usuário poderá acessar os objetos virtuais, cada qual com sua área restrita. Esses estarão dentro do Laboratório Químico. Ex. os próprios ambientes virtuais (biblioteca, armário e mesa), ar-condicionado, porta automática, tópico de ajuda, etc...

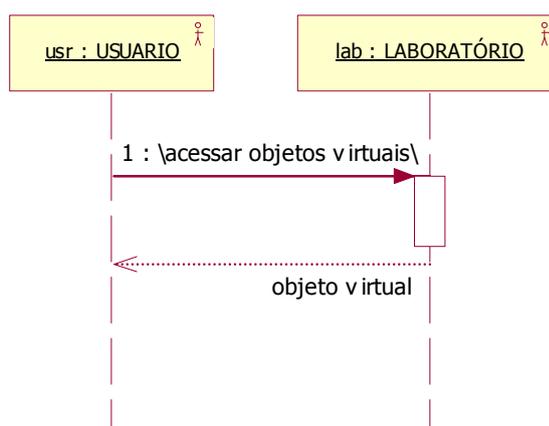


Figura A38 - Diagrama de Seqüência do caso de uso “Acessar Objetos Virtuais”.

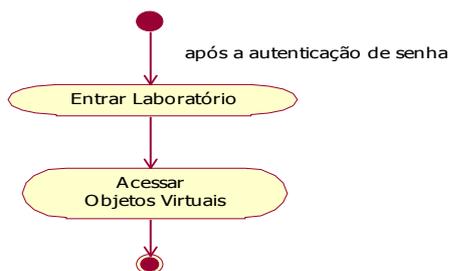


Figura A39 - Diagrama de Atividade do caso de uso “Acessar Objetos Virtuais”.

### 23) Caso de Uso: **Gerar Animação e Escolher Arquivo**

Atores que Participam: Usuário, Aluno, Professor, Administrador do Sistema, Laboratório, Biblioteca, Mesa e Armário.

Quem inicia o caso: Usuário.

Tipo: Opcional.

Descrição: O ambiente (ator) Biblioteca terá características de uma estante com as portas fechadas, assim que o usuário selecionar a visão, a animação ocorrerá, a porta se abrirá e encontrarão livros e disquetes.

Conseqüentemente o Usuário poderá escolher com o mouse o livro e/ou disquetes que desejam, gerando arquivos PDFs e/ou LINKs em HTML. Estes arquivos serão consultas para sobre a matéria e química.

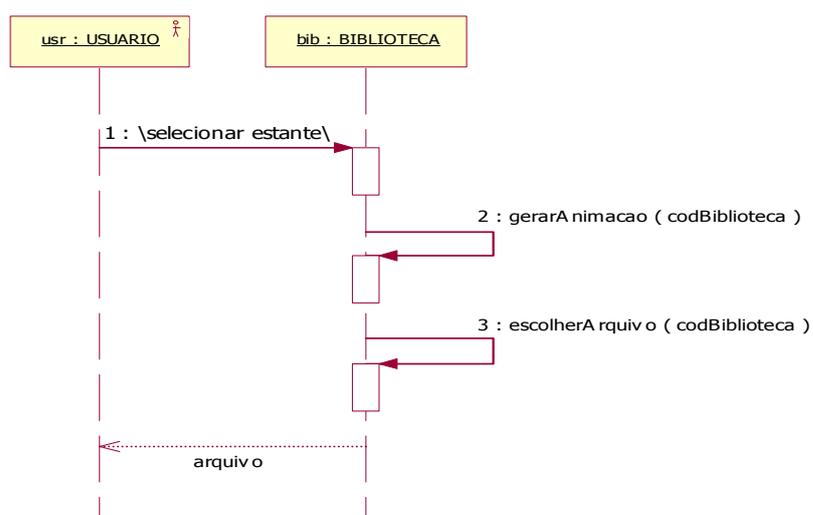


Figura A40 - Diagrama de Seqüência do caso de uso “Gerar Animação e Escolher Arquivo”.

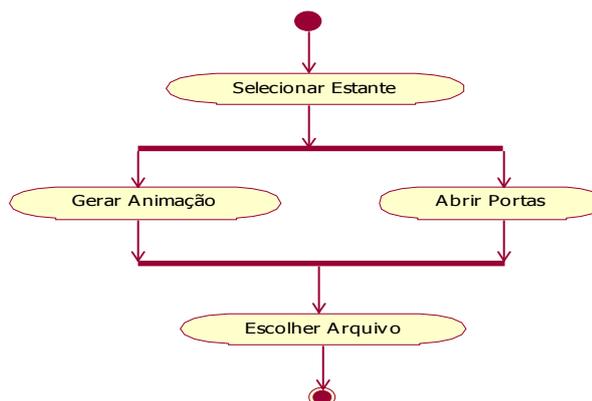


Figura A41 - Diagrama de Atividade do caso de uso “Gerar Animação e Escolher Arquivo”.

#### 24) Caso de Uso: **Escolher Elemento Químico**

Atores que Participam: Usuário, Aluno, Professor, Administrador do Sistema, Laboratório, Biblioteca, Mesa e Armário.

Quem inicia o caso: Usuário.

Tipo: Primário.

Descrição: O Usuário poderá na visão (ator) do Armário, escolher a cada click com o mouse quantos elementos deseja pegar. As restrições serão impostas pelos tipos de dados definidos pela implementação funcional. Terá 114 opções de escolha podendo mesclar e repetir, assim podendo iniciar o objetivo principal do software.

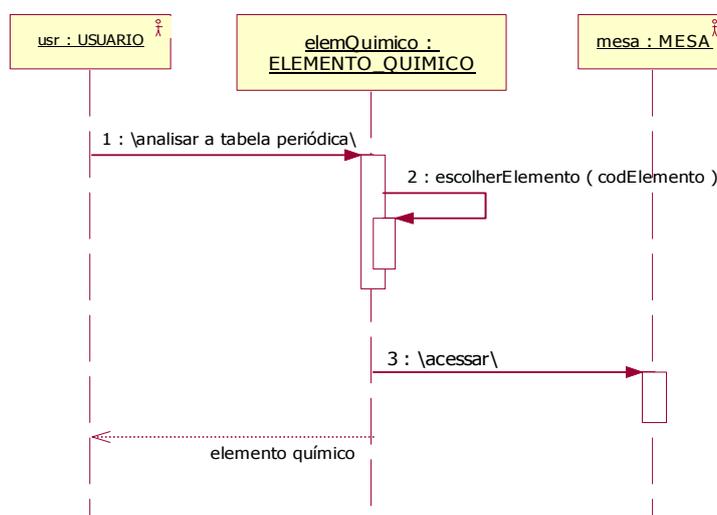


Figura A42 - Diagrama de Seqüência do caso de uso “Escolher Elemento Químico”.

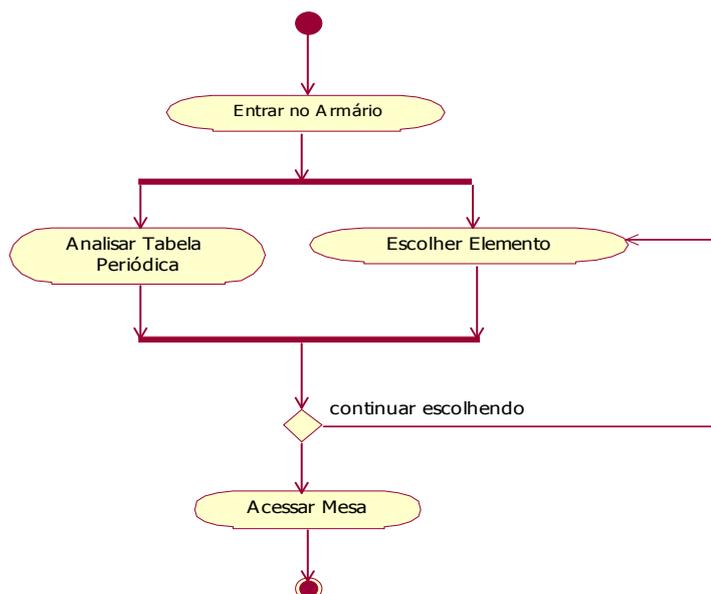


Figura A43 - Diagrama de Atividade do caso de uso “Escolher Elemento Químico”.

## 25) Caso de Uso: **Excluir Escolha de Elemento**

Atores que Participam: Usuário, Aluno, Professor, Administrador do Sistema, Laboratório, Biblioteca, Mesa e Armário.

Quem inicia o caso: Usuário.

Tipo: Primário.

Descrição: Após a escolha do elemento, de acordo com o caso de uso “Escolher Elemento Químico”, se o Usuário não quiser mais o elemento escolhido, poderá excluí-lo, selecionando com o mouse e excluindo com a opção de acordo. Entretanto, será feita uma validação de exclusão, confirmando se aquele elemento pode ser excluído.

Essa validação funcionará da seguinte maneira: Se o Usuário confirmar a exclusão, o sistema analisará se o elemento a ser excluído está sendo usado por algum composto que o próprio Usuário criou naquele momento (esse composto não é cadastrado, foi apenas criado naquele momento).

A exclusão também poderá ser feita após as seleções dos elementos, de acordo com os caso de uso “Selecionar Elemento Químico”, o Usuário poderá excluí-lo, selecionando com o mouse o lixo (objeto virtual) e a opção a ser excluída, o lixo poderá ser habilitado ajudando a excluir mais de uma opção de uma só vez.

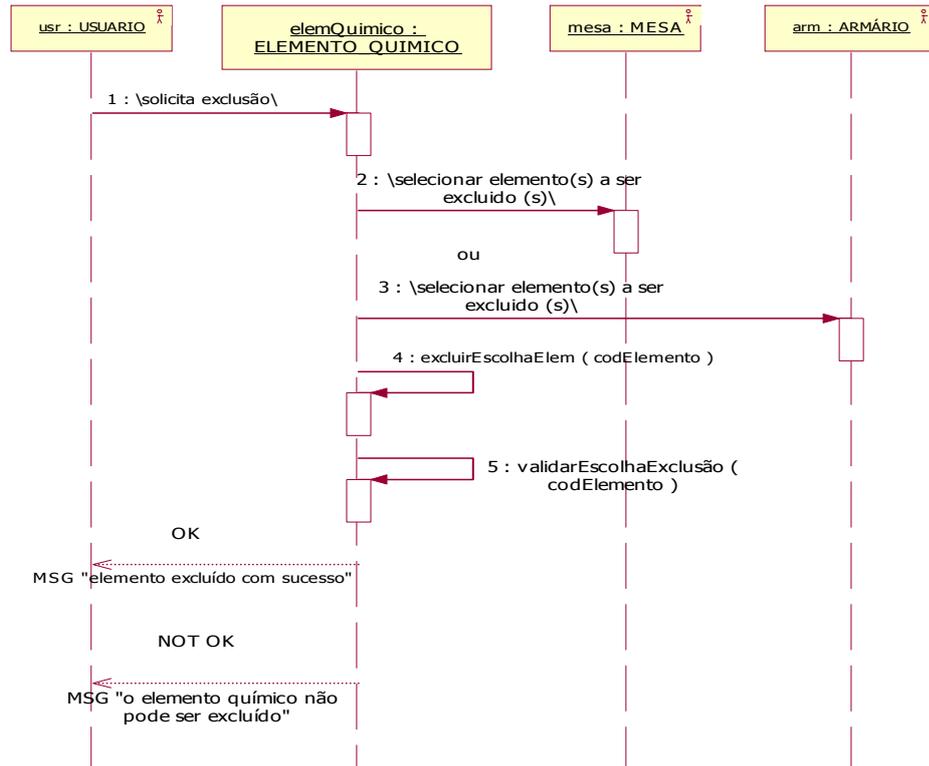


Figura A44 - Diagrama de Sequência do caso de uso “Excluir Escolha de Elemento”.

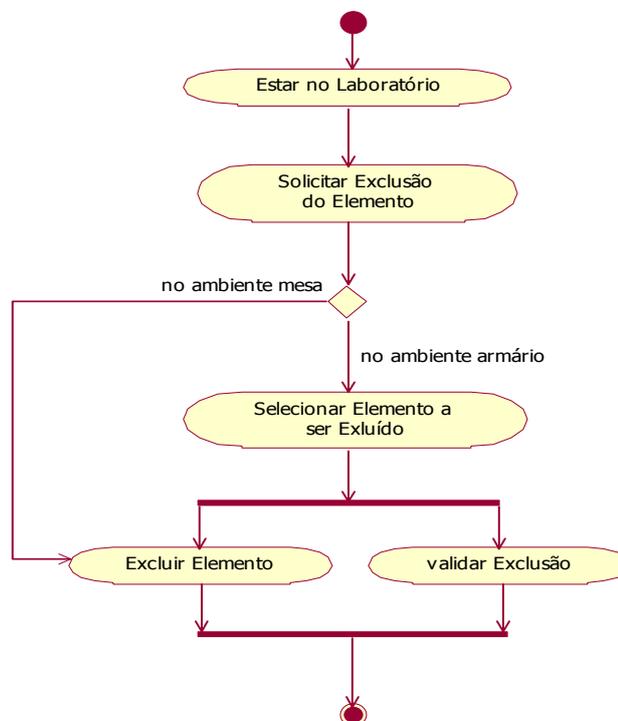


Figura A45 - Diagrama de Atividade do caso de uso “Excluir Escolha de Elemento”.

## 26) Caso de Uso: **Selecionar Elemento Químico**

Atores que Participam: Usuário, Aluno, Professor, Administrador do Sistema, Laboratório, Biblioteca, Mesa e Armário.

Quem inicia o caso: Usuário.

Tipo: Primário.

Descrição: O Usuário poderá selecionar elementos pré-escolhidos, onde estarão vindos do armário, ou seja, dos elementos selecionados, ele poderá separar os que vão compor o composto.

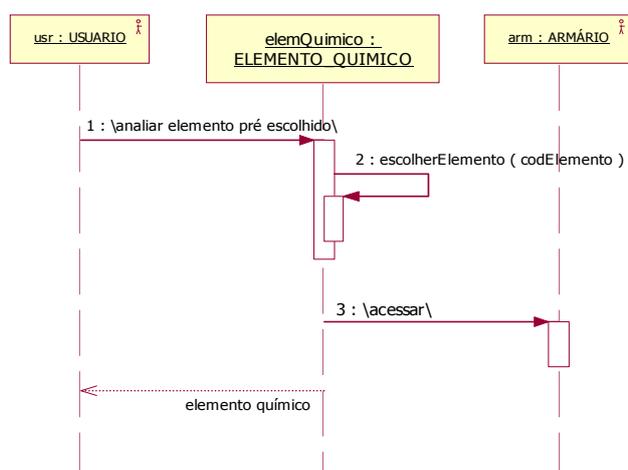


Figura A46 - Diagrama de Seqüência do caso de uso “Selecionar Elemento Químico”.

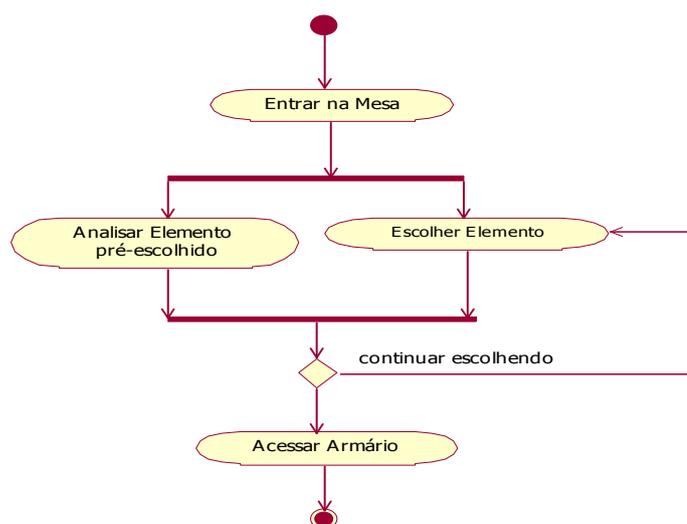


Figura A47 - Diagrama de Atividade do caso de uso “Selecionar Elemento Químico”.

## 27) Caso de Uso: **Ligar Elemento Químico**

Atores que Participam: Usuário, Aluno, Professor, Administrador do Sistema, Laboratório, Biblioteca, Mesa e Armário.

Quem inicia o caso: Usuário.

Tipo: Primário

Descrição: Assim que os elementos químicos forem selecionados, de acordo com o caso de uso “Selecionar Elemento Químico”, o Usuário terá quatro opções de ligações de elementos. A primeira será a ligação automática, onde se habilitada, selecionando os elementos, se ligarão automaticamente (se existir o composto). As outras ligações serão: ligação iônica, ligação covalente e ligação metálica, onde se habilitada, gera a ligação manualmente de elementos possíveis, caso contrário haverá uma animação de repulsão mostrando para os usuários que a ligação não pode ser feita, ou seja, não pode ser gerada, de acordo com suas normas químicas.

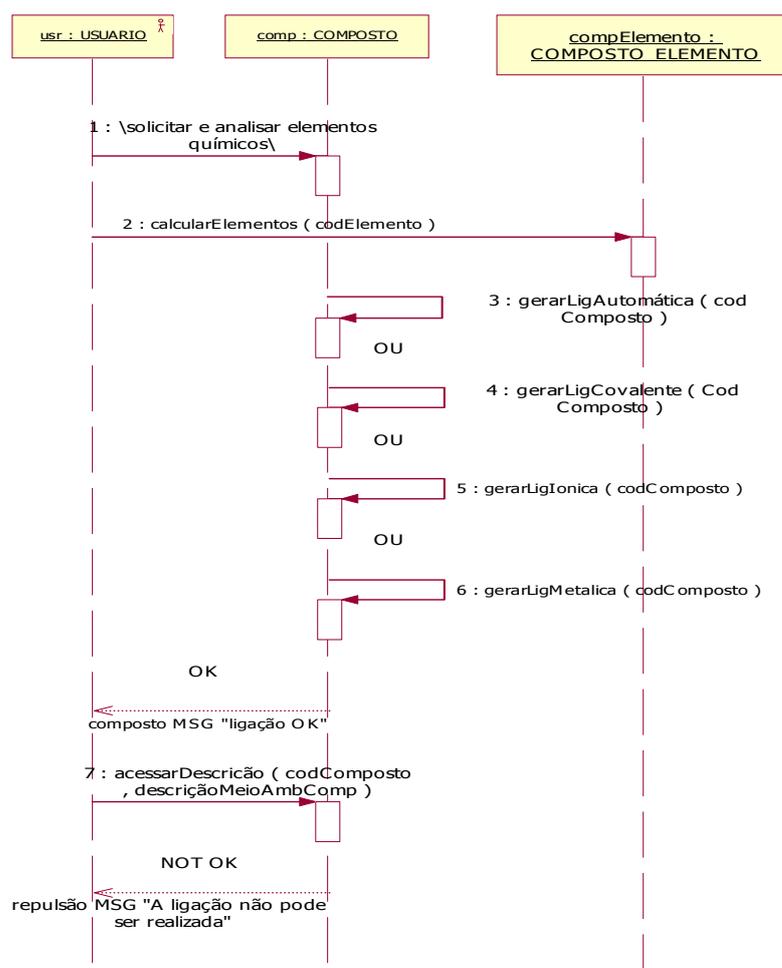


Figura A48 - Diagrama de Seqüência do caso de uso “Ligar Elemento Químico”.

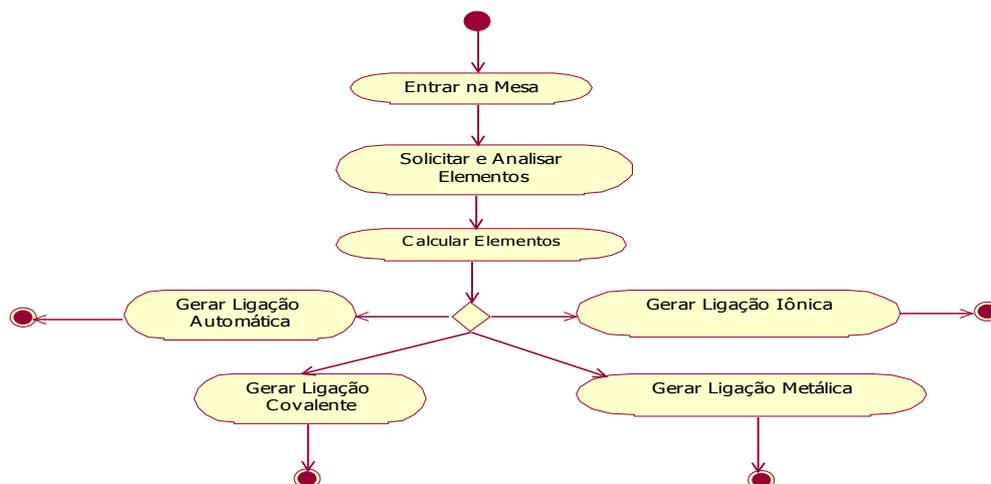


Figura A49 - Diagrama de Atividade do caso de uso "Ligar Elemento Químico"

## 28) Caso de Uso: **Cortar Ligação**

Atores que Participam: Usuário, Aluno, Professor, Administrador do Sistema, Laboratório, Biblioteca, Mesa e Armário.

Quem inicia o caso: Usuário

Tipo: Secundário

Descrição: Após a escolha do elemento químico e sua ligação, de acordo com os casos de usos "Selecionar Elemento Químico" e "Ligar Elemento Químico", o usuário poderá "cortar" (cancelar) as ligações se a opção estiver habilitada (uma ligação de cada vez).

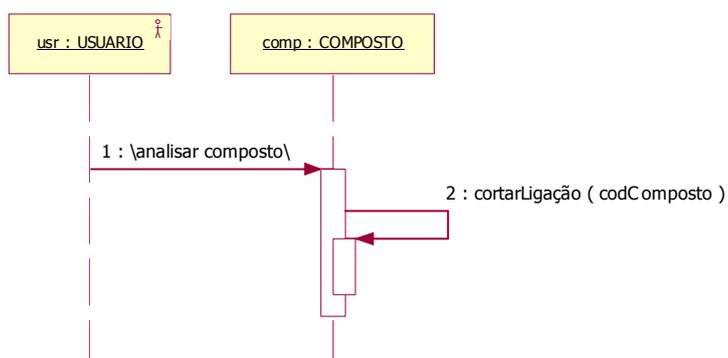


Figura A50 - Diagrama de Sequência do caso de uso "Cortar Ligação"

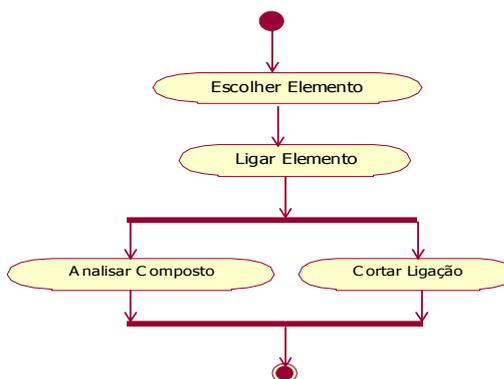


Figura A51 - Diagrama de Atividade do caso de uso “Cortar Ligação”

### 29) Caso de Uso: **Excluir Composto (Inativo)**

Atores que Participam: Usuário, Aluno, Professor, Administrador do Sistema, Laboratório, Biblioteca, Mesa e Armário.

Quem inicia o caso: Usuário.

Tipo: Secundário

Descrição: Após as seleções dos elementos ou da geração de compostos, de acordo com os casos de usos “Selecionar Elemento Químico” e “Ligar Elemento Químico”, o Usuário poderá excluí-los, selecionando com o mouse o lixo (objeto virtual) e o composto(s). O lixo poderá ser habilitado ajudando excluir mais de uma opção de uma só vez. Automaticamente será feita uma validação de exclusão, confirmando se o composto deverá realmente ser excluído.

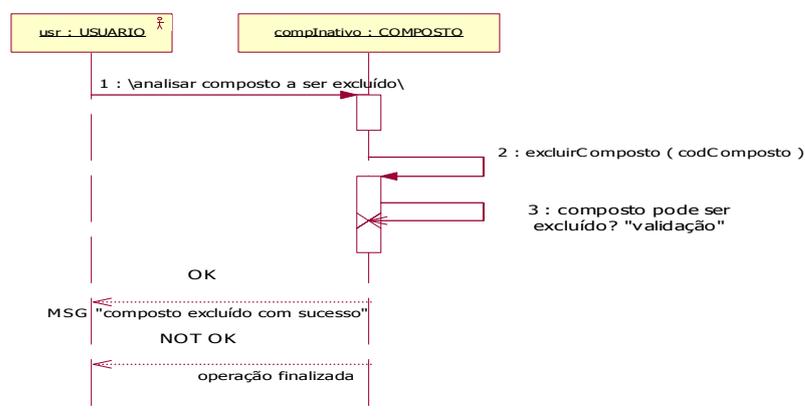


Figura A52 - Diagrama de Seqüência do caso de uso “Excluir Composto (Inativo)”.

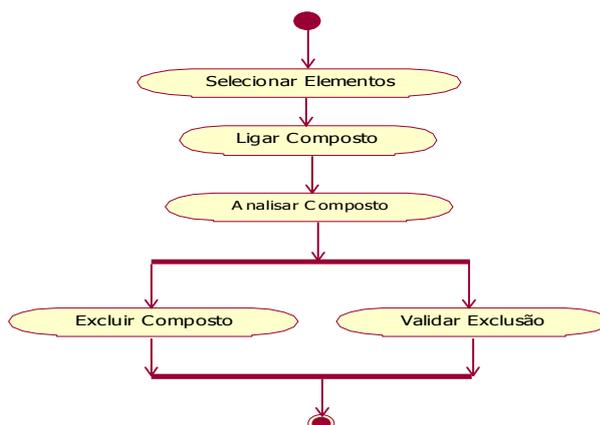


Figura A53 - Diagrama de Atividade do caso de uso “Excluir Composto (Inativo)”.

### 30) Caso de Uso: **Buscar Ligação Binária**

Atores que Participam: Usuário, Aluno, Professor, Administrador do Sistema, Laboratório, Biblioteca, Mesa e Armário.

Quem inicia o caso: Usuário.

Tipo: Secundário

Descrição: Caso o Usuário deseje fazer múltiplas ligações (binárias) até obter o composto, poderá consultar uma lista de tabela (denominada ligação binária) contendo as possíveis ligações para unir com a outra, assim ajudando-o nessa função. Essa tabela poderá ser solicitada.

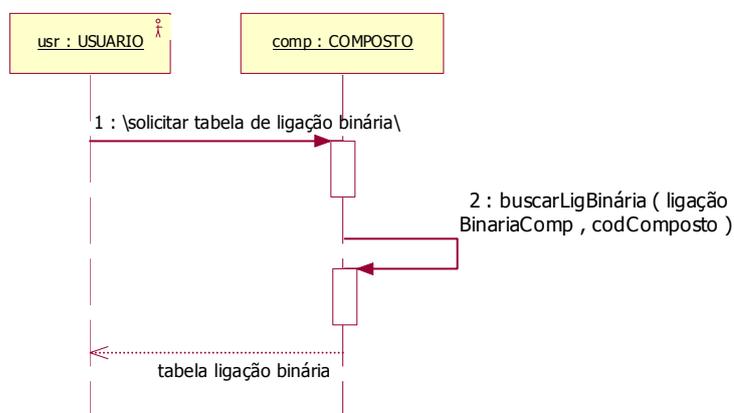


Figura A54 - Diagrama de Seqüência do caso de uso “Buscar Ligação Binária”.

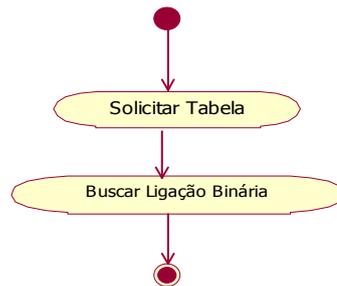


Figura A55 - Diagrama de Atividade do caso de uso “Buscar Ligação Binária”.

### 31) Caso de Uso: **Acessar Ambientes**

Atores que Participam: Usuário, Aluno, Professor, Administrador do Sistema, Laboratório, Biblioteca, Mesa e Armário.

Quem inicia o caso: Armário ou Mesa.

Tipo: Primário

Descrição: Meio em que o Usuário possa estar no ambiente Armário e acessar o Ambiente Mesa, ou vice-versa.

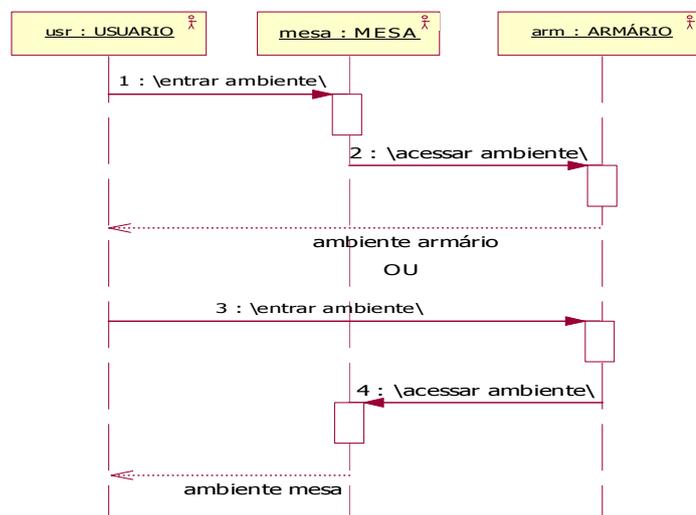


Figura A56 - Diagrama de Sequência do caso de uso “Acessar Ambientes”

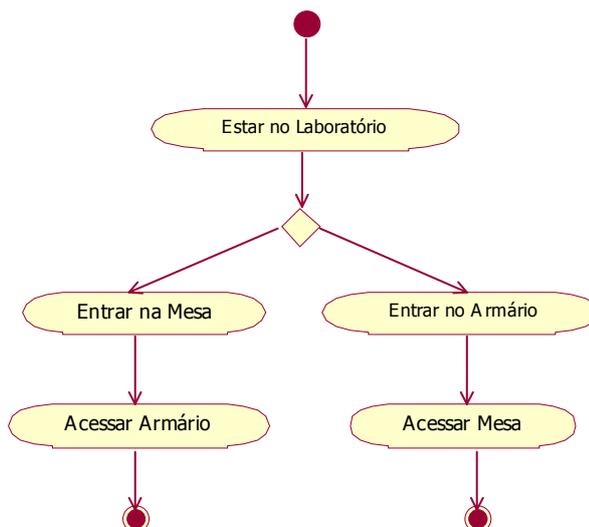


Figura A57 - Diagrama de Atividade do caso de uso “Acessar Ambientes”

### 32) Caso de Uso: **Cadastrar Composto Inativo**

Atores que Participam: Usuário, Aluno, Professor, Administrador do Sistema, Laboratório, Biblioteca, Mesa e Armário.

Quem inicia o caso: Usuário.

Tipo: Primário

Descrição: O Usuário poderá salvar o composto criado no laboratório químico em uma base de dados, como composto inativo. Automaticamente o sistema passará por uma validação, de acordo com o caso de uso “Validar Cadastro Composto Ativo”, podendo posteriormente ser acessada, excluída, alterada e corrigida pelo próprio Aluno em sua área restrita, de acordo com o caso de uso “Controlar Cadastro de Composto”. Este composto poderá também ser alterado diretamente de composto inativo para compostos ativo, de acordo com o caso de uso “Alterar Cadastro de Composto Inativo”.

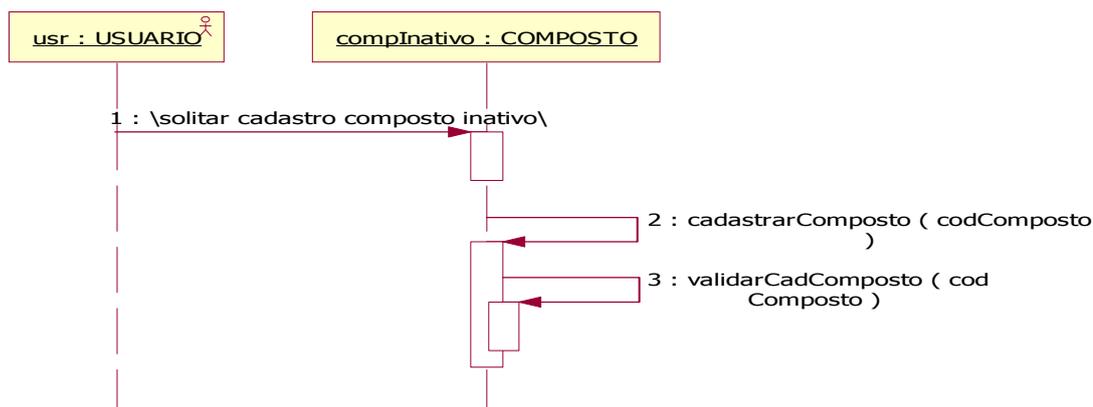


Figura A58 - Diagrama de Sequência do caso de uso “Cadastrar Composto Inativo”

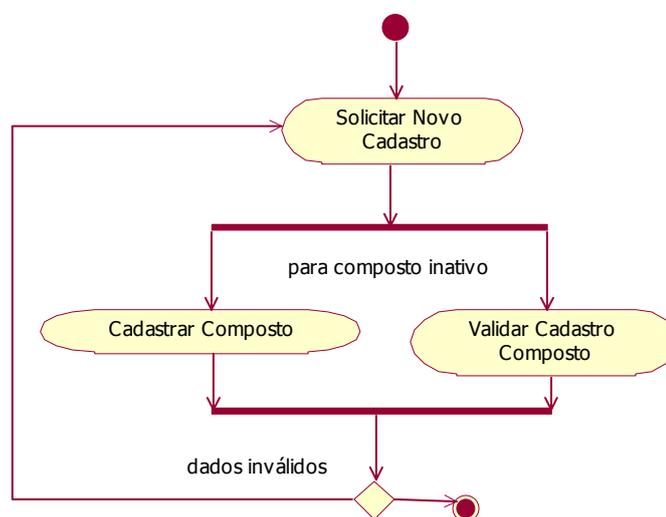


Figura A59 - Diagrama de Atividade dos casos de uso “Cadastrar Composto Inativo” e “Validar Cadastro Composto Ativo”.

### 33) Caso de Uso: **Validar Cadastro Composto Inativo**

Atores que Participam: Usuário, Aluno, Professor, Administrador do Sistema, Laboratório, Biblioteca, Mesa e Armário.

Quem inicia o caso: Usuário.

Tipo: Primário

Descrição: Após o cadastro de compostos inativos, terá uma confirmação para que todos os campos tenham sido preenchidos.

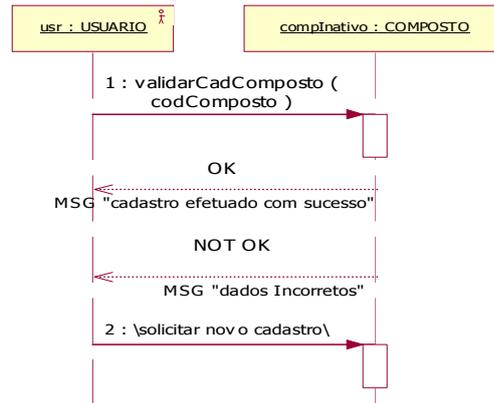


Figura A60 - Diagrama de Seqüência do caso de uso “Validar Cadastro Composto Inativo”.

## APÊNDICE C - Modelo Entidade Relacionamento (Físico)

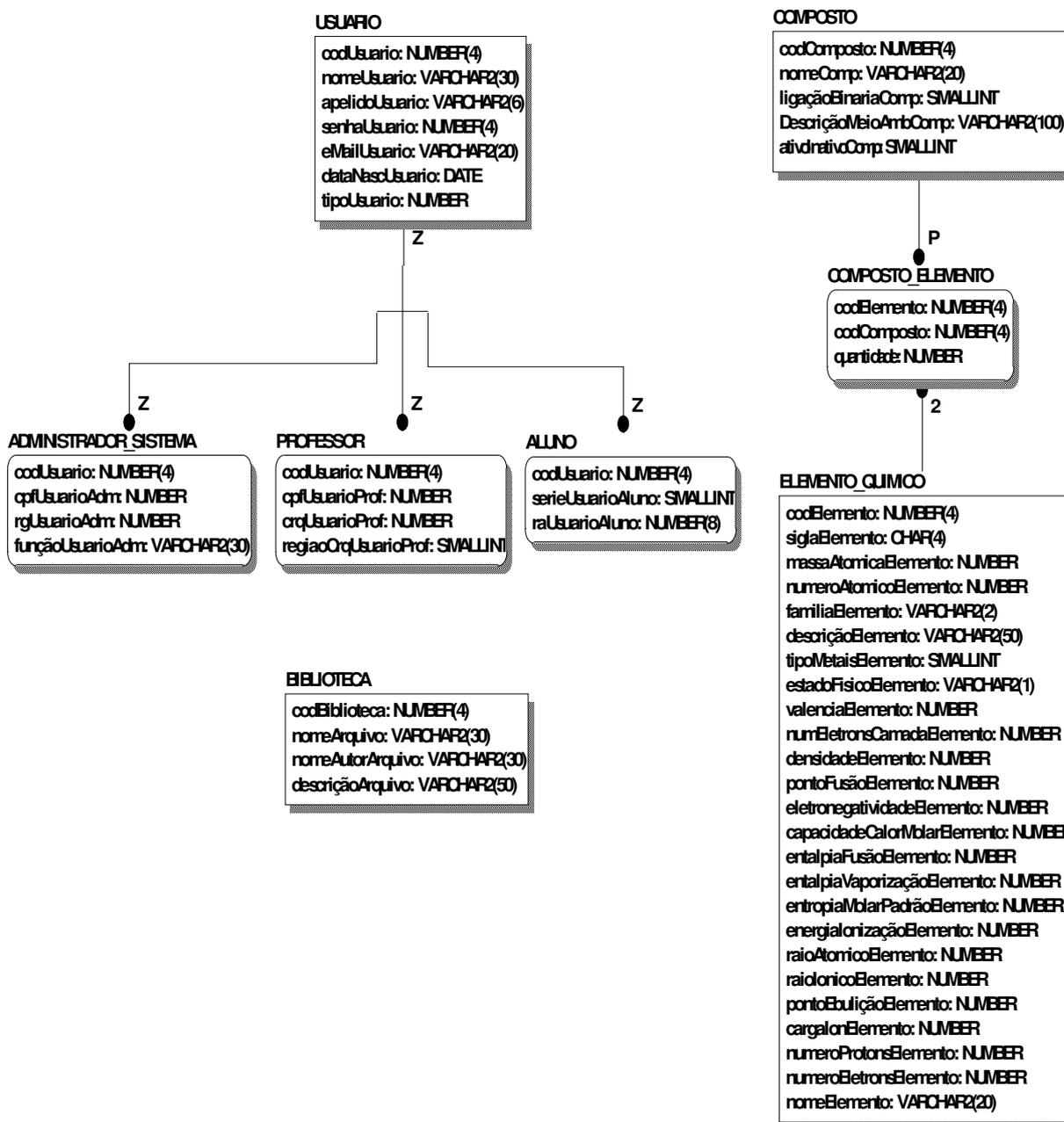


Figura A61 – MER Físico.

Este diagrama apresenta a mesma estrutura de tabelas e tipos de dados do diagrama de classes, e a partir de seu término, foi gerado um dicionário de dados apresentado em Apêndice D, mostrando todas descrições das tabelas e dados do SEQ. Foi modelado pela ferramenta COMPUTER ASSOCIATES ERWIN 4.0.

## APÊNDICE D - Dicionário de Dados

Tabela A1 - Identificação e Comentário das Tabelas.

Table	
Name	Comment
<b>ADMINISTRADOR_SISTEMA</b>	Essa tabela é uma generalização da tabela USUARIO e tem como função armazenar os dados do Administrador do Sistema. Por ser uma generalização, terá todos os atributos da tabela USUARIO além dos restritos próprios de sua tabela "ADMINISTRADOR_SISTEMA". O Administrador poderá controlar o cadastro do Professor e poderá acessar todos os ambientes do Software Educacional de Química Terá o Cod_Usuario para uma possível identificação e consulta. Para um melhor entendimento de sua funcionalidade, veja diagramas UML. Relaciona-se com a tabela: - USUARIO
<b>ALUNO</b>	Essa tabela é uma generalização da tabela USUARIO e tem como função armazenar os dados do Aluno. Por ser uma generalização, terá todos os atributos da tabela USUARIO além dos restritos próprios de sua tabela "ALUNO". O Aluno poderá acessar apenas ao Laboratório (Visão 1) e conseqüentemente os ambientes que estarão dentro dele, além de se cadastrar automaticamente. Terá o Cod_Usuario para uma possível identificação. Para um melhor entendimento de sua funcionalidade, veja diagramas UML. Relaciona-se com a tabela: - USUARIO
<b>BIBLIOTECA</b>	Essa tabela tem como função de armazenar os dados cadastrais dos arquivos PDFS, onde será usado na Biblioteca (Visão 2). Terá como chave primária o Cod_Biblioteca, garantindo uma possível consulta e identificação. Apenas o Professor e Administrador do Sistema (em suas áreas restritas) poderão ter acesso a esse cadastro, e todos os usuários poderão acessa-los dentro do Laboratório (Visão 1) em Biblioteca. Para um melhor entendimento de sua funcionalidade, veja diagramas UML.
<b>COMPOSTO</b>	Essa tabela tem como função armazenar os dados cadastrais do Composto. Esse poderá ser considerado como "Ativo" pelo Usuário Professor e Administrador do Sistema e "Inativo" pelo Usuário Aluno. Futuramente os usuários Professor e Administrador poderão acessar os compostos inativos, podendo alterá-lo (corrigindo e salvando) como compostos ativos. Para um melhor entendimento de sua funcionalidade, veja diagramas UML. Relaciona-se com a tabela: - COMPOSTO_ELEMENTO
<b>COMPOSTO_ELEMENTO</b>	Relaciona-se com as tabelas COMPOSTO e ELEMENTO, por seu relacionamento ser muitos-para-muitos, essa é gerada automaticamente. Tem como chave primária os atributos: Cod_Composto e Cod_Elemento.
<b>ELEMENTO_QUIMICO</b>	Essa tabela tem como função armazenar os dados cadastrais dos Elementos Químicos. Somente os usuários Professor e Administrador poderão cadastrar (em suas áreas restritas). Os Elementos serão escolhidos pelos usuários (no ambiente laboratório) para gerar o composto, nas Visões 3 e 4 (armário e mesa). Para um melhor entendimento de sua funcionalidade, veja diagramas UML. Relaciona-se com a tabela: - COMPOSTO_ELEMENTO
<b>PROFESSOR</b>	Essa tabela é uma generalização da tabela USUARIO e tem como função armazenar os dados do Professor. Por ser uma generalização, terá todos os atributos da tabela USUARIO além dos restritos próprios de sua tabela "PROFESSOR". O Professor poderá acessar todos os ambientes virtuais menos controlar o seu cadastro e do Administrador do Sistema. Terá o Cod_Usuario para uma possível identificação. Para um melhor entendimento de sua funcionalidade, veja diagramas UML. Relaciona-se com as tabelas: - USUARIO - BIBLIOTECA_USUARIO.
<b>USUARIO</b>	Essa tabela tem como função armazenar os dados do Usuário cadastrados e se caracteriza por ser generalização, portanto, os usuários poderão ser de 3 tipos: Administrador do Sistema ou Professor ou Aluno. Ao iniciar o software o usuário poderá logar, inserindo sua senha e nome caso seja

Table	
Name	Comment
	<p>cadastrado, caso contrário, poderá se cadastrar.</p> <p>Para cada tipo de usuário terá um CodUsuario, garantindo uma possível identificação, além do seu login.</p> <p>Para um melhor entendimento de sua funcionalidade, veja diagramas UML.</p> <p>Relaciona-se com as tabelas:</p> <ul style="list-style-type: none"> <li>- ADMINISTRADOR_SISTEMA (generalização)</li> <li>- PROFESSOR (generalização)</li> <li>- ALUNO (generalização)</li> </ul>

Tabela A2 - Identificação e Comentário dos Atributos da Tabela ADMINISTRADOR\_SISTEMA.

Column(s) of "ADMINISTRADOR_SISTEMA" Table		
Name	Datatype	Comment
CodUsuario	NUMBER(4)	Chave primária da tabela.
cpfUsuarioAdm	NUMBER	Recebe o n° do CPF do usuário Administrador do Sistema, no formato XXXXXXXXXXXX numérico sem ponto e sem hífen.
rgUsuarioAdm	NUMBER	Recebe o n° do RG do usuário Administrador do Sistema, no formato XXXXXXXXXXXX numérico sem ponto e sem hífen.
funçãoUsuarioAdm	VARCHAR2(30)	Recebe o nome da função onde o Administrador do Sistema atua.

Tabela A3 - Identificação e Comentário dos Atributos da Tabela ALUNO.

Column(s) of "ALUNO" Table		
Name	Datatype	Comment
CodUsuario	NUMBER(4)	Chave primária da tabela.
RaUsuarioAluno	NUMBER(8)	Recebe o RA (número acadêmico) do usuário Aluno.
SerieUsuarioAluno	SMALLINT	Recebe a série do Aluno.

Tabela A4 - Identificação e Comentário dos Atributos da Tabela BIBLIOTECA.

Column(s) of "BIBLIOTECA" Table		
Name	Datatype	Comment
CodBiblioteca	NUMBER(4)	Chave primária da tabela.
NomeAutorArquivo	VARCHAR2(30)	Recebe os nomes dos autores do arquivo PDF.
NomeArquivo	VARCHAR2(30)	Recebe o nome do arquivo PDF.
DescriçãoArquivo	VARCHAR2(50)	Recebe algumas observações ou descrições do arquivo PDF.

Tabela A5 - Identificação e Comentário dos Atributos da Tabela COMPOSTO.

Column(s) of "COMPOSTO" Table		
Name	Datatype	Comment
CodComposto	NUMBER(4)	Chave primária da tabela.
NomeComp	VARCHAR2(20)	Recebe o Nome do composto gerado.
DescriçãoMeioAmbComp	VARCHAR2(100)	<p>Descrição sobre o composto gerado, onde conterà todas as informações como:</p> <p>Onde podemos usar o composto no meio ambiente.</p>
LigaçãoBinariaComp	SMALLINT	<p>Se habilitada, o atributo será do tipo "Binária".</p> <p>Esse atributo servirá quando o usuário quiser gerar um composto onde esse terá duas ou mais tipos de ligações. Porém poderá consultar uma tabela de ligação binária, para juntar-se com outra ligação.</p> <p>Em nível de Programação, quando o usuário for cadastrar o</p>

Column(s) of "COMPOSTO" Table		
Name	Datatype	Comment
		composto no ambiente de laboratório, essa opção será desativada, pois apenas o usuário professor e administrador poderão habilitar esse atributo em suas áreas restritas. Portanto na área restrita do aluno, terá apenas a consulta desta tabela, se esse campo estiver sido habilitado pelos outros usuários (professor ou administrador).
AtivoInativoComp	SMALLINT	Atributo referente ao tipo de composto no sistema se é ativo ou inativo. Ao nível de programação, essa opção será habilitada se os usuários que forem cadastrar o composto forem professor ou Administrador (em suas áreas restritas), caso contrário, é automaticamente desativada.

Tabela A6 - Identificação e Comentário dos Atributos da Tabela COMPOSTO\_ELEMENTO.

Column(s) of "COMPOSTO_ELEMENTO" Table		
Name	Datatype	Comment
CodComposto	NUMBER(4)	Chave primária da tabela.
CodElemento	NUMBER(4)	Chave primária da tabela.
Quantidade	NUMBER	Recebe a quantidade de elementos necessários para gerar um composto.

Tabela A7 - Identificação e Comentário dos Atributos da Tabela ELEMENTO\_QUÍMICO.

Column(s) of "ELEMENTO_QUIMICO" Table		
Name	Datatype	Comment
CodElemento	NUMBER(4)	Chave primária da tabela.
SiglaElemento	CHAR(4)	Recebe a sigla científica de do elemento químico. Ex: H.
massaAtomicaElemento	NUMBER	Recebe a massa atômica científica de do elemento químico. Ex: (H): 1,0.
numeroAtomicoElemento	NUMBER	Recebe o número atômico científico do elemento químico. Ex: (H): 1.
familiaElemento	VARCHAR2(2)	Recebe a família do elemento. 1A = metais alcalinos 2A = metais alcalino-terroso 6A = calcogênios 7A = halogênios 0 = gases nobres
descricaoElemento	VARCHAR2(50)	Recebe as observações e descrições do elemento. Ex: A palavra Hidrogênio deriva de hydor e gen, ou que forma água. Foi descoberto em 1766. Apesar do Sol e das demais estrelas serem hidrogênio quase puro, ele é raramente encontrado livre na Terra.
tipoMetaisElemento	SMALLINT	Recebe o tipo do elemento. O elemento químico pode ser do tipo metais ou não metais.
estadoFisicoElemento	VARCHAR2(1)	Recebe o estado físico de elemento químico. Se é Sólido = S Líquido = L ou Gasoso = G. Ex: (H): Gasoso.
valenciaElemento	NUMBER	Recebe a valência científica do elemento químico.
numEletronsCamadaElemento	NUMBER	Recebe a quantidade de números de elétrons da última camada.
densidadeElemento	NUMBER	Recebe a densidade do elemento químico. Ex: (H) Densidade: 0,089 g/cm <sup>3</sup> .
pontoFusãoElemento	NUMBER	Recebe o ponto de fusão do elemento químico. Ex: (H) Ponto de Fusão: -259°C.
eletronegatividadeElemento	NUMBER	Recebe a Eletronegatividade do elemento químico. Ex: (H) Eletronegatividade: 2,2
capacidadeCalorMolarElemento	NUMBER	Recebe a Capacidade de Calor Molar do elemento químico. Ex: (H) Calor Molar: 2,220,784 J K <sup>-1</sup> mol <sup>-1</sup> .
entalpiaFusãoElemento	NUMBER	Recebe a Entalpia de Fusão do elemento químico. Ex: (H) Entalpia de Fusão: 0,12 kJ mol <sup>-1</sup> .

Column(s) of "ELEMENTO_QUIMICO" Table		
Name	Datatype	Comment
entalpiaVaporizaçãoElemento	NUMBER	Recebe a Entalpia de Vaporização do elemento químico. Ex: (H) Entalpia de Vaporização: 0,46 kJ mol-1.
entropiaMolarPadrãoElemento	NUMBER	Recebe a Entropia Molar Padrão do elemento químico. Ex: (H) Entropia Molar Padrão: 114,71 J K-1mol-1.
energiaIonizaçãoElemento	NUMBER	Recebe a Energia de Ionização do elemento químico. Ex: (H) Energia de Ionização: (1) 1310 kJ mol-1.
raioAtômicoElemento	NUMBER	Recebe o Raio Atômico do elemento químico. Ex: (H) Raio Atômico: 78 pm.
raioIônicoElemento	NUMBER	Recebe o Raio Iônico do elemento químico. Ex: (H) Raio Iônico: 154 pm.
pontoEbuliçãoElemento	NUMBER	Recebe o Ponto de Ebulição do elemento químico. Ex: (H) Ponto de Ebulição: -253°C.
cargaIonElemento	NUMBER	Recebe a carga de íon do elemento químico.
numeroEletronsElemento	NUMBER	Recebe o número de elétrons do elemento químico.
numeroProtonsElemento	NUMBER	Recebe o número de prótons do elemento químico.
nomeElemento	VARCHAR2(20)	Recebe o nome científico de do elemento químico. Ex: Hidrogênio.

Tabela A8 - Identificação e Comentário dos Atributos da Tabela PROFESSOR.

Column(s) of "PROFESSOR" Table		
Name	Datatype	Comment
<b>codUsuario</b>	<b>NUMBER(4)</b>	<b>Chave primária da tabela.</b>
crqUsuarioProf	NUMBER	Recebe o nº do CRQ (Conselho Regional de Química) do usuário Professor, no formato XXXXXXXX numerico sem ponto e sem ifem.
cpfUsuarioProf	NUMBER	Recebe o nº do CPF do usuário Professor, no formato XXXXXXXXXXXX numerico sem ponto e sem ifem.
regiaoCrqUsuarioProf	SMALLINT	Recebe a região do CRQ (Conselho Regional de Química) do usuário Professor, exemplo: 3º Região.

Tabela A9 - Identificação e Comentário dos Atributos da Tabela USUÁRIO.

Column(s) of "USUARIO" Table		
Name	Datatype	Comment
<b>codUsuario</b>	<b>NUMBER(4)</b>	<b>Chave primária da tabela.</b>
nomeUsuario	VARCHAR2(30)	Recebe o Nome do Usuário.
apelidoUsuario	VARCHAR2(6)	Recebe o apelido para login.
senhaUsuario	NUMBER(4)	Recebe a senha para login.
eMailUsuario	VARCHAR2(20)	Recebe o e-mail do usuário.
dataNascUsuario	DATE	Recebe a data do nascimento do usuário, no formato DD/MM/AAAA
TipoUsuario	NUMBER	Recebe o tipo de usuário: Se é Administrador do Sistema = 1 ou Professor = 2 ou Aluno = 3.

## APÊNDICE E - Relatórios de Revisões Técnicas

### Relatório Sumário de Revisão Técnica

#### Identificação da Revisão:

Projeto: Software Educacional de Química (SEQ)    Número da Revisão: 01  
 Data: 07/03/2005    Local: Univem, sala 7    Hora: 14h

#### Identificação do Produto:

Material revisado: Alguns tópicos da Análise de Requisitos  
 Produtor: Juliana Pereira de Souza (líder)  
 Descrição Breve: Alguns tópicos primordiais para o processo de desenvolvimento do SEQ e futuro desenvolvimento do Estudo de Caso.

#### Material Revisado:

- Identificação do Problema;
- Necessidades do Sistema.

#### Equipe de Revisão:

Nome	Assinatura
1. <u>Juliana Pereira de Souza (líder)</u>	_____
2. <u>Ederson Relvas (secretário)</u>	_____
3. <u>Ana Paula P. M. P. (orientadora)</u>	_____

#### Avaliação do Projeto:

Aceito: como está ( ) com pequenas modificações (X)  
 Não Aceito: revisão significativa ( ) pequena revisão ( )  
 Revisão Não Completada: ( )

#### Material Suplementar em Anexo:

Lista de questões ( ) Materiais produzidos anotados ( )  
 Outros (X) CHECK LIST

Número da Revisão: 01

Data da Revisão: 07/03/2005

Líder da Revisão: Juliana Pereira de Souza

Secretário: Ederson Relvas

CHECK LIST:

1. Quantos tipos de usuários o SEQ (Software Educacional de Química) terá?
2. Qual a função (missão) de cada usuário?
3. Até que ponto os usuários poderão interagir com as atividades realizadas no SEQ?
4. De que maneira o SEQ explorará o conteúdo sendo um software educacional construtivista?
5. Quais estratégias didáticas que serão utilizadas?
6. Que tipo de argumento o SEQ tratará predominantemente?
7. Quais são as ferramentas necessárias para seu desenvolvimento (modelagem de diagramas UML, implementação funcional de interface)?
8. Quais são os recursos de informática necessários para que o SEQ seja implantado (configurações)?
9. O SEQ precisará ser interligado em rede? Qual?
10. Qual o principal objetivo para informatizar um SEQ?
11. Quais os impactos provocados pela informatização de um SEQ?
12. Quais as contribuições do SEQ terá em relação aos objetivos didáticos?
13. Quais são as principais dificuldades para implementar ligações químicas?
14. Como será o controle do usuário? Precisar de login?
15. O usuário Aluno poderá salvar seu composto? Mesmo que incorreto?
16. Como será o controle de compostos ativos e inativos?
17. Os usuários serão cadastrados?
18. Os elementos químicos serão cadastrados? Por qual tipo de usuário?
19. Quem controlará o cadastro de professor?
20. Quais elementos químicos que podem ser ligados?
21. Quais são os principais problemas encontrados até agora? E as necessidades?

## Relatório Sumário de Revisão Técnica

### Identificação da Revisão:

Projeto: Software Educacional de Química (SEQ) Número da Revisão: 02  
 Data: 24/03/2005 Local: Univem, sala 7 Hora: 14h

### Identificação do Produto:

Material revisado: Conceitos de Química  
 Produtor: Juliana Pereira de Souza (líder)  
 Descrição Breve: Conceitos de todos requisitos de química do ensino médio, necessário para o desenvolvimento do software.

### Material Revisado:

- Tabela Periódica;
- Ligações Químicas.

### Equipe de Revisão:

Nome	Assinatura
1. <u>Juliana Pereira de Souza (líder)</u>	_____
2. <u>Caio Correa (secretário)</u>	_____
3. <u>Paulo Marinelli (Prof. química)</u>	_____
4. <u>Ana Paula P. M. P. (orientadora)</u>	_____

### Avaliação do Projeto:

Aceito: como está ( ) com pequenas modificações (X)  
 Não Aceito: revisão significativa ( ) pequena revisão ( )  
 Revisão Não Completada: ( )

### Material Suplementar em Anexo:

Lista de questões (X) Materiais produzidos anotados ( )  
 Outros ( ) \_\_\_\_\_

Número da Revisão: 02

Data da Revisão: 24/03/2005

Líder da Revisão: Juliana Pereira de Souza

Secretário: Caio Correa

Lista de Questões (Problemas):

1. O conceito de átomo tem que ser bem entendido pelo produtor (a), pois um átomo não é um elemento químico, todavia não é uma molécula. Entretanto, não pode se referir a um elemento químico dizendo um átomo.
2. Os usuários do SEQ (Software Educacional de Química) poderão escolher dos 114 qualquer um elemento da tabela periódica para gerar a ligação química, mas algumas ligações não poderão ser concluídas, pois alguns elementos químicos são vistos pelo ensino superior e não pelo ensino médio que é o foco deste software educacional. Portanto essa restrição deve ser analisada.
3. A equipe de revisão recomenda uma modificação sobre a classificação dos elementos da tabela periódica, esses classificados como: metais, não-metais, semimetais e não-metais. Devem ser analisados pela tabela mais atual, ou seja, tirando a divisão de semimetais.
4. Devem ser utilizados os principais tipos de ligações químicas: Ligação Iônica, Ligação Covalente e Ligação Metálica.

## Relatório Sumário de Revisão Técnica

### Identificação da Revisão:

Projeto: Software Educacional de Química      Número da Revisão: 03  
 Data: 05/08/2005      Local: Univem, sala 7      Hora: 14h

### Identificação do Produto:

Material revisado: Análise de Requisitos  
 Produtor: Juliana Pereira de Souza (líder)  
 Descrição Breve: Todos os Requisitos do Sistema, para o Estudo de Caso do Diagrama de Use Case.

### Material Revisado:

- Avaliar a Descrição do diagrama de caso de uso, dentro do CHECK LIST adotado.

### Equipe de Revisão:

Nome	Assinatura
1. <u>Juliana Pereira de Souza (líder)</u>	_____
2. <u>Ederson Relvas (secretário)</u>	_____
3. <u>Ana Paula P. M. P. (orientadora)</u>	_____

### Avaliação do Projeto:

Aceito: como está ( ) com pequenas modificações ( )  
 Não Aceito: revisão significativa (X) pequena revisão ( )  
 Revisão Não Completada: ( )

### Material Suplementar em Anexo:

Lista de questões (X) Materiais produzidos anotados ( )  
 Outros (X) CHECK POINTS e CHECK LIST

Número da Revisão: 03

Data da Revisão: 05/08/2005

Líder da Revisão: Juliana Pereira de Souza

Secretário: Éderson Relvas

**CHECK POINTS:**

1. Avaliar o Requisito “Logar no Sistema”;
2. Avaliar o Requisito “Solicitar Senha”;
3. Avaliar o Requisito “Cadastrar Usuário”;
4. Avaliar o Requisito “Validar Cadastro Aluno”;
5. Avaliar o Requisito “Controle do Sistema”;
6. Avaliar o Requisito “Cadastrar Arquivo PDF”;
7. Avaliar o Requisito “Validar Cadastro PDF”;
8. Avaliar o Requisito “Cadastrar Elemento Químico”;
9. Avaliar o Requisito “Validar Cadastro Elemento”;
10. Avaliar o Requisito “Cadastrar Composto Ativo”;
11. Avaliar o Requisito “Validar Cadastro Composto Ativo”;
12. Avaliar o Requisito “Alterar Cadastro de Composto Inativo”;
13. Avaliar o Requisito “Acessar Objetos Virtuais”;
14. Avaliar o Requisito “Gerar Animação e Escolher Arquivo”;
15. Avaliar o Requisito “Escolher Elemento Químico”;
16. Avaliar o Requisito “Excluir Escolha de Elemento”;
17. Avaliar o Requisito “Selecionar Elemento Químico”;
18. Avaliar o Requisito “Ligar Elemento Químico”;
19. Avaliar o Requisito “Cortar Ligação”;
20. Avaliar o Requisito “Excluir Composto (Inativo)”;
21. Avaliar o Requisito “Buscar Ligação Binária”;
22. Avaliar o Requisito “Cadastrar Composto Inativo”;
23. Avaliar o Requisito “Validar Cadastro Composto Inativo”;
24. Avaliar o Requisito “Acessar Ambientes”.

Número da Revisão: 03

Data da Revisão: 05/08/2005

Líder da Revisão: Juliana Pereira de Souza

Secretário: Éderson Relvas

**CHECKLIST:**

1. A análise do domínio da informação está completa?  
OBS: Espera-se com este item verificar se a Análise de Requisitos possui todas as funcionalidades necessárias ou se existe alguma funcionalidade que pode ser incluída.
2. A análise do domínio da informação está consistente?  
OBS: Espera-se com este item verificar se as informações da Análise estão consistentes, ou seja, não existe nenhuma funcionalidade ou requisito ambíguo.
3. A análise do domínio da informação está correta?  
OBS: Espera-se com este item verificar se as informações estão corretas, não havendo nenhuma informação que não faça parte do domínio das necessidades encontradas.
4. O particionamento do problema está completo?  
OBS: O objetivo deste item é verificar se o problema está corretamente dividido em subproblemas.
5. Todos os requisitos podem ser mapeados para o nível de sistema?  
OBS: Neste item esperamos saber se todos os requisitos encontrados serão realmente implementados no sistema.
6. Os requisitos correspondem ao modelo do mundo real ou foi maquiado?  
OBS: Os requisitos devem corresponder ao modelo do mundo real.
7. Cada requisito corresponde a uma única funcionalidade do sistema?  
OBS: Cada requisito deve corresponder a uma única funcionalidade do sistema.
8. As descrições dos requisitos estão escritas de forma clara sem deixar dúvidas sobre a função que eles devem desempenhar no sistema?  
OBS: O requisito deve descrever claramente sua função no sistema.
9. Serão necessários protótipos para o perfeito entendimento dos requisitos?  
OBS: Caso necessários devem ser construídos os protótipos para cada requisito.
10. Os requisitos estão prontos para serem mapeados para o projeto preliminar?  
OBS: As transições dos requisitos para o projeto preliminar devem ser o mais automático possível.
11. Os requisitos atendem ao desempenho esperado?  
OBS: Os requisitos devem ser compatíveis com o desempenho esperado do sistema.
12. Os critérios de validações estão completamente especificados?  
OBS: Os critérios de validações devem estar completamente especificados.

Número da Revisão: 03

Data da Revisão: 06/05/2005

Líder da Revisão: Juliana Pereira de Souza

Secretário: Éderson Relvas

Lista de Questões (Problemas):

N ° Problema	Nº Check Points	Descrição do Problema (dentre Check List)
1	1	Falta incluir o requisito para autenticar senha, pois o Usuário, após logar no sistema, sua senha e apelido precisa passar por uma autenticação, para que possa entrar no sistema. Se essa Autenticação for positiva, cada usuário poderá entrar em sua área restrita (se for cadastrado).
2	2	OK, Nenhuma Observação.
3	3	OK, Nenhuma Observação.
4	4	OK, Nenhuma Observação.
5	5	O Administrador antecessor poderá controlar o cadastro do Administrador e Professor, porém, precisará validar os campos desses usuários.
6	6	O caso "Acessar Área Professor" se relaciona com esse caso. Falta o relacionamento.
7	7	Após a validação do cadastro, o Administrador ou Professor poderá excluí-los, altera-los ou consultá-los os arquivos cadastrados. Porém falta um requisito para esse controle.
8	8	O caso "Acessar Área Professor" se relaciona com esse caso. Falta o relacionamento.
9	9	Após a validação do cadastro, o Administrador ou Professor poderá excluí-los, altera-los ou consultá-los os elementos cadastrados. Porém falta um requisito para esse controle.
10	10	O caso "Acessar Área Professor" se relaciona com esse caso. Falta o relacionamento.
11	11	Após a validação do cadastro, o Administrador ou Professor poderá excluí-los, altera-los ou consultá-los os composto ativo. Porém falta um requisito para esse controle.
12	12	O caso "Acessar Área Professor" se relaciona com esse caso. Falta o relacionamento.
13	13	OK, Nenhuma Observação.
14	14	OK, Nenhuma Observação.
15	15	OK, Nenhuma Observação.
16	16	OK, Nenhuma Observação.
17	17	OK, Nenhuma Observação.
18	18	OK, Nenhuma Observação.
19	19	OK, Nenhuma Observação.
20	20	OK, Nenhuma Observação.
21	21	OK, Nenhuma Observação.
22	22	OK, Nenhuma Observação.
23	23	OK, Nenhuma Observação.
24	24	OK, Nenhuma Observação.

N ° Problema	Solução (Providências a serem tomadas)
1	Incluir os casos de usos "Autenticar Senha", "Acessar Área Aluno", "Acessar Área Professor" e "Acessar Área Administrador". Esses três últimos casos citados se relacionam com a seta de dependência com o caso de uso "Autenticar

	Senha”. E também, o caso “Acessar Área Administrador” <<extend>> o caso “Acessar Área Professor” que por sua vez <<extend>> o caso “Acessar Área Aluno”.
5	Incluir o caso de uso “Validar Cadastros”
6	Incluir o relacionamento: “Acessar Área Professor” <<extend>> “Cadastrar Arquivo PDF”
7	Incluir o caso de uso “Controlar Cadastro de Arquivo”
8	Incluir o relacionamento: “Acessar Área Professor” <<extend>> “Cadastrar Elemento Químico”
9	Incluir o caso de uso “Controlar Cadastro de Elemento”
10	Incluir o relacionamento: “Acessar Área Professor” <<extend>> “Cadastrar Composto Ativo”
11	Incluir o caso de uso “Controlar Cadastro de Composto”. Este depende do caso “Validar Cadastro Composto Inativo”.
12	Incluir o relacionamento: “Acessar Área Professor” <<extend>> “Alterar Cadastro de Composto Inativo”
OBS:	Incluir o caso de uso: “Controlar Cadastro Usuário”, dependendo do caso “Validar Cadastros”.