



UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM
INFORMÁTICA

**Um Modelo de Organização de Recursos Didáticos
Baseado em Componentes para um
Ambiente de Ensino a Distância**

Míriam Nunes Carvalho

Orientador: Prof. Dr. Arturo Hernández-Domínguez

Campina Grande – PB

Julho de 2002



UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM
INFORMÁTICA

**Um Modelo de Organização de Recursos Didáticos
Baseado em Componentes para um
Ambiente de Ensino a Distância**

Míriam Nunes Carvalho

*Dissertação de mestrado submetida à Coordenação
do Curso de Pós-Graduação em Informática -
COPIN - da Universidade Federal de Campina
Grande, como parte dos requisitos necessários à
obtenção do grau de Mestre em Ciências(MSc).*

Orientador: Prof. Dr. Arturo Hernández-Domínguez

Área de Concentração: Ciência da Computação

Campina Grande – PB

Julho de 2002

CARVALHO, Míriam Nunes

C331M

Um Modelo de Organização de Recursos Didáticos Associados a um Domínio em Ambiente de Ensino a Distância

Míriam Nunes Carvalho – Campina Grande, PB, 2002.

Orientador: Prof. Dr. Arturo Hernández-Domínguez

Dissertação de Mestrado, Universidade Federal de Campina Grande, Centro de Ciências e Tecnologia, Coordenação de Pós Graduação em Informática, Campina Grande, Paraíba, julho de 2002.

1. Informática aplicada a Educação
2. Banco de Dados

CDU – 681.3.01:37

*O importante é continuar aprendendo, desfrutar os desafios e tolerar a
ambigüidade pois, em definitivo, não existem certezas.*

Marina Horner

A minha mãe, Rina,
eterna incentivadora e fã.
Ao meu pai Fernando (in memoriam).

AGRADECIMENTOS

“A satisfação está no esforço e não apenas na realização final.”

Gandhi

A elaboração de uma dissertação, de forma simultânea com o exercício de uma atividade profissional, é um grande desafio. Para vencê-lo é preciso contar com o apoio e incentivo de familiares e amigos. Por isto, agradeço a Deus, por tê-los colocado em meu caminho.

Ao professor Arturo, pela paciência e conselhos indispensáveis à conclusão deste trabalho.

A todos os professores do curso. Seus ensinamentos, sem dúvida, contribuíram muito para a confecção desta dissertação. A Marcus Sampaio, exemplo de professor, pelo respeito e carinho dedicado aos alunos. A Jacques, pela determinação em “acordar” o Mestre que existe em cada um de nós.

A George Leite, professor de Java, pela presteza em me ensinar Java e JSP, e pelas “dicas” durante a implementação do protótipo.

A todos os colegas que dividiram seu tempo comigo durante todo o mestrado, e em especial a Domingos e Débora, pelo companheirismo.

A toda a minha família, pela torcida, em especial a D. Rina, *my English teacher*. *Thanks, mom*. A meus irmãos Fernando e Maiesse, pelo exemplo.

Aos meus filhos, Lygia e Danilo, fontes de força e inspiração, por tentarem me mostrar, durante todo o tempo, que “nem só de mestrado vive o homem”.

A Adriana, por “segurar” a administração doméstica, principalmente na fase de conclusão deste trabalho.

De uma forma muito especial, a meu esposo, amigo, namorado e companheiro Ivo, por seu carinho e ajuda e, principalmente, sua paciência em aceitar, nos mais diversos tipos de situação, mais um “só depois do mestrado”.

SUMARIO

LISTAS.....	ix
RESUMO.....	xiv
1 INTRODUÇÃO	1
2 EDUCAÇÃO A DISTÂNCIA.....	5
2.1 CONCEITO DE EDUCAÇÃO A DISTÂNCIA.....	5
2.2 EDUCAÇÃO A DISTÂNCIA NO CONTEXTO DA INTERNET	7
2.3 AMBIENTES DE EDUCAÇÃO A DISTÂNCIA	8
2.3.1 O AMBIENTE WEBCT.....	10
2.3.2 O AMBIENTE LEARNING SPACE.....	12
2.3.3 O AMBIENTE TOP CLASS.....	13
2.3.4 O AMBIENTE AULANET.....	14
2.4 COMPARAÇÃO ENTRE OS AMBIENTES APRESENTADOS.....	15
2.5 O PROJETO ACVA	17
2.5.1 DESCRIÇÃO DO AMBIENTE	17
2.5.2 DESENVOLVIMENTO DO PROJETO NO CONTEXTO DA ACVA.....	19
2.6 CONCLUSÃO.....	20
3 MODELOS DE REPRESENTAÇÃO DE RECURSOS DIDÁTICOS	
ASSOCIADOS A UM DOMÍNIO.....	21
3.1 MODELOS DE ORGANIZAÇÃO DE MATERIAL DIDÁTICO	22
3.1.1 ÁRVORE CONCEITUAL.....	22
3.1.2 ONTOLOGIA.....	22
3.1.3 MAPA CONCEITUAL.....	25
3.2 COMPARAÇÃO DAS DIFERENTES ABORDAGENS E MODELOS DE	
REPRESENTAÇÃO DE MATERIAL DIDÁTICO	31
3.3 CONCLUSÃO.....	33
4 DESENVOLVIMENTO BASEADO EM COMPONENTES.....	34
4.1 COMPONENTES EM UML E JAVA.....	35
4.2 DIAGRAMAS DE COMPONENTES	36
4.3 DESENVOLVIMENTO BASEADO EM COMPONENTES.....	39

4.4	CONCLUSÃO.....	39
5	UM MODELO DE ORGANIZAÇÃO DE RECURSOS DIDÁTICOS.....	42
5.1	REPRESENTAÇÃO DOS RECURSOS DIDÁTICOS.....	42
5.2	UM EXEMPLO DO MODELO PROPOSTO	47
5.3	CONCLUSÃO	49
6	MODELAGEM DO SGC	50
6.1	LINGUAGEM E MÉTODO.....	50
6.2	REQUISITOS	50
6.2.1	<i>DESCRIÇÃO DO NEGÓCIO.....</i>	<i>51</i>
6.2.2	<i>ESQUEMA CONCEITUAL.....</i>	<i>52</i>
6.2.3	<i>GLOSSÁRIO DE TERMOS.....</i>	<i>53</i>
6.2.4	<i>REQUISITOS FUNCIONAIS.....</i>	<i>55</i>
6.2.5	<i>CASOS DE USO.....</i>	<i>56</i>
6.2.6	<i>CASOS DE USO – REFINAMENTO.....</i>	<i>60</i>
6.3	ESPECIFICAÇÃO DO SISTEMA.....	77
6.3.1	<i>CENÁRIOS E DIAGRAMAS DE SEQUÊNCIA.....</i>	<i>77</i>
6.4	ARQUITETURA GERAL DO SISTEMA	86
6.5	PROJETO INTERNO - COMPONENTES	88
6.5.1	<i>COMPONENTE – USUÁRIO.....</i>	<i>88</i>
6.5.2	<i>COMPONENTE – ONTOLOGIA.....</i>	<i>89</i>
6.5.3	<i>COMPONENTE – COMPONENTE DE APRENDIZAGEM.....</i>	<i>90</i>
6.6	CONCLUSÃO.....	91
7	IMPLEMENTAÇÃO DO SGC	94
7.1	FERRAMENTAS UTILIZADAS	94
7.2	OS COMPONENTES DO SGC.....	97
7.2.1	<i>FUNÇÕES DO ADMINISTRADOR.....</i>	<i>98</i>
7.2.2	<i>FUNÇÕES DO PROFESSOR.....</i>	<i>100</i>
7.2.3	<i>FUNÇÕES DO ALUNO.....</i>	<i>102</i>
7.3	O AMBIENTE SGC	103
7.3.1	<i>FUNÇÕES DO ADMINISTRADOR.....</i>	<i>104</i>
7.3.2	<i>FUNÇÕES DO PROFESSOR.....</i>	<i>106</i>

7.3.3	<i>FUNÇÕES DO ALUNO</i>	111
7.4	O SGC NO CONTEXTO DA ACVA	115
7.5	CONCLUSÃO.....	116
8	CONCLUSÃO.....	117
8.1	OBJETIVOS ALCANÇADOS.....	117
8.2	CONTRIBUIÇÕES	118
8.3	LIMITAÇÕES	118
8.4	TRABALHOS FUTUROS	119
9	REFERÊNCIAS BIBLIOGRÁFICAS	120
	ANEXO I.....	120
	ANEXO II.....	124
	ANEXO III	135

ACVA	-	Arquitetura de uma Classe Virtual Adaptativa
API	-	Application Programming Interface
CCTT	-	Center for Curriculum Transfer and Tecnology
EAD	-	Educação a Distância
FabriCar	-	Fábrica de Aatoria de Tutores Inteligentes Hiperímia para Cardiologia
GTS	-	Serviço de Treinamento de Grupos
HTML	-	Hypertest Markup Language
IBW	-	Introdução baseada na Web
JSP	-	Java Server Page
LES	-	Laboratorio de Engenharia de Software
OMG	-	Object Modeling Group
PUC-RJ	-	Pontifícia Universidade Católica do Rio de Janeiro
SGC	-	Sistema Gerenciador de Componentes de Aprendizagem
SIBT	-	Service Independent Training Building Block
SIT	-	Sistemas de Tutores Inteligentes
UML	-	Unified Modeling Language
VCTS	-	Serviço de Treinamento de Classe Virtual
WebCT	-	Web Course Tools
WWW	-	World Wide Web

FIGURA 1 – ROTEIRO DE CRIAÇÃO DE CURSO NO WEBCT	11
FIGURA 2 – ARQUITETURA GERAL PARA CADA SERVIÇO DE TREINAMENTO ADAPTADO	17
FIGURA 3 – CAMADAS DA ACVA	19
FIGURA 4 –SERVIÇOS FORNECIDOS PELA CAMADA SF_BÁSICOS ÀS OUTRAS CAMADAS DA ACVA.....	20
FIGURA 5 – ÁRVORE CONCEITUAL	22
FIGURA 6 – TIPOS DE ONTOLOGIA, DE ACORDO COM O NÍVEL DE DEPENDÊNCIA EM UMA TAREFA	23
FIGURA 7 – FABRICAR - ARQUITETURA DAS FERRAMENTAS DE AQUISIÇÃO DO CONHECIMENTO.	25
FIGURA 8 – EXEMPLO DE MAPA CONCEITUAL	28
FIGURA 9 –MAPA CONCEITUAL DESCREVENDO UM AMBIENTE DE EAD	29
FIGURA 10 – MAPA CONCEITUAL USADO PARA PLANEJAMENTO DE CURRÍCULO - MACRO	30
FIGURA 11 – MAPA CONCEITUAL USADO PARA CONCEITOS ESPECÍFICOS DO PROGRAMA INSTRUCIONAL – MICRO	31
FIGURA 12 – EXEMPLO DE CLASSE IMPLEMENTANDO INTERFACE	35
FIGURA 13 – EXEMPLO DE DIAGRAMA DE COMPONENTES	37
FIGURA 14A – NOTAÇÃO UTILIZADA NA REPRESENTAÇÃO DE UM COMPONENTE DO TIPO MATERIAL DIDÁTICO.....	43
FIGURA 14B – NOTAÇÃO UTILIZADA NA REPRESENTAÇÃO DE UM COMPONENTE DO TIPO AULA	43
FIGURA 14C – NOTAÇÃO UTILIZADA NA REPRESENTAÇÃO DE UM COMPONENTE DO TIPO CURSO	43
FIGURA 15 – MODELO DE REPRESENTAÇÃO E ORGANIZAÇÃO DOS COMPONENTES DE APRENDIZAGEM.....	44
FIGURA 16A – MODELO DE ONTOLOGIA DE APLICAÇÃO USADO NA DEFINIÇÃO DO TIPO DE MATERIAL DIDÁTICO.....	45
FIGURA 16B – MODELO DE ONTOLOGIA DE TAREFAS USADO NA DEFINIÇÃO DO ASSUNTO DAS AULAS.....	45
FIGURA 16C – MODELO DE ONTOLOGIA DE DOMÍNIO USADO NA DEFINIÇÃO DO DOMÍNIO DOS CURSOS.....	46
FIGURA 17A – INSTANCIAMENTO DA ONTOLOGIA DE DOMÍNIO PARA CURSO.....	46
FIGURA 17B – INSTANCIAMENTO DA ONTOLOGIA DE TAREFAS PARA AULAS	46

FIGURA 17C – INSTANCIACÃO DA ONTOLOGIA DA APLICAÇÃO PARA MATERIAL DIDÁTICO.....	46
FIGURA 18 – MODELO DE ORGANIZAÇÃO DE CURSO.....	47
FIGURA 19 A – EXEMPLO DE REUSO DO COMPONENTE MATERIAL DIDÁTICO NO MESMO CURSO.	48
FIGURA 19B – EXEMPLO DE REUSO DE COMPONENTES AULA E MATERIAL DIDÁTICO EM CURSOS DIFERENTES	49
FIGURA 20 – ESQUEMA CONCEITUAL DO SGC.....	53
FIGURA 21 – VISÃO GERAL DOS CASOS DE USO.....	57
FIGURA 22 – CASO DE USO MANTER CADASTRO DE USUÁRIOS.....	58
FIGURA 23 – CASO DE USO MANTER MATERIAL DIDÁTICO	58
FIGURA 24 – CASO DE USO MANTER AULA	59
FIGURA 25 – CASO DE USO MANTER CURSO.....	59
FIGURA 26 – CASO DE USO UTILIZAR CURSO	60
FIGURA 27 – CASO DE USO MANTER ONTOLOGIA	60
FIGURA 28 – CASO DE USO MANTER MATERIAL DIDÁTICO – REFINAMENTO	62
FIGURA 29 – CASO DE USO MANTER AULA – REFINAMENTO	64
FIGURA 30 – CASO DE USO MANTER CURSO – REFINAMENTO.....	68
FIGURA 31 – CASO DE USO UTILIZAR CURSO – REFINAMENTO.....	72
FIGURA 32 – CASO DE USO MANTER ONTOLOGIA – REFINAMENTO.....	74
FIGURA 33 – DIAGRAMA DE SEQÜÊNCIA – CADASTRAR USUÁRIO	78
FIGURA 34 – DIAGRAMA DE SEQÜÊNCIA – CRIAR MATERIAL.....	79
FIGURA 35 – DIAGRAMA DE SEQÜÊNCIA – CRIAR AULA	79
FIGURA 36 – DIAGRAMA DE SEQÜÊNCIA ALOCAR MATERIAL EM AULA	80
FIGURA 37 – DIAGRAMA DE SEQÜÊNCIA – ORGANIZAR MATERIAL DIDÁTICO DE UMA AULA ..	81
FIGURA 38 – DIAGRAMA DE SEQÜÊNCIA – CRIAR CURSO.....	82
FIGURA 39 – DIAGRAMA DE SEQÜÊNCIA – ALOCAR AULAS EM CURSO.....	83
FIGURA 40 – DIAGRAMA DE SEQÜÊNCIA – ORGANIZAR AULAS DE UM CURSO.....	84
FIGURA 41 – DIAGRAMA DE SEQÜÊNCIA – CRIAR DOMÍNIO	84
FIGURA 42 – DIAGRAMA DE SEQÜÊNCIA – CRIAR ASSUNTO	85
FIGURA 43 – DIAGRAMA DE SEQÜÊNCIA – UTILIZAR CURSO	86
FIGURA 44 – ARQUITETURA GERAL DO SISTEMA	88
FIGURA 45 – PROJETO INTERNO DO COMPONENTE USUÁRIO	89
FIGURA 46 – PROJETO INTERNO DO COMPONENTE ONTOLOGIA	90
FIGURA 47 – PROJETO INTERNO DO COMPONENTE COMPONENTE DE APRENDIZAGEM.....	91

FIGURA 48 – DIAGRAMA DE CLASSES PERTINENTES AO BANCO DE DADOS SQL SERVER	96
FIGURA 49 – DIAGRAMA DE COMPONENTES CRIAR ALUNO	99
FIGURA 50 – DIAGRAMA DE COMPONENTES CRIAR DOMÍNIO	100
FIGURA 51 – DIAGRAMA DE COMPONENTES CRIAR MATERIAL DIDÁTICO	101
FIGURA 52 – DIAGRAMA DE COMPONENTES ALOCAR MATERIAL EM AULA.	102
FIGURA 53 – DIAGRAMA DE COMPONENTES ASSISTIR CURSO.	103
FIGURA 54 – TELA PRINCIPAL DO SGC – SISTEMA GERENCIADOR DE COMPONENTES	104
FIGURA 55 – INTERFACE PARA CRIAÇÃO DE DOMÍNIOS	105
FIGURA 56 – INTERFACE PARA CRIAÇÃO DE ASSUNTOS	105
FIGURA 57 – INTERFACE PARA CRIAÇÃO DE ALUNOS	106
FIGURA 58 – INTERFACE PARA CRIAÇÃO DE MATERIAL DIDÁTICO	107
FIGURA 59 – INTERFACE PARA CRIAÇÃO DE AULAS	107
FIGURA 60 – INTERFACE PARA CRIAÇÃO DE CURSOS	108
FIGURA 61 – INTERFACE PARA ALOCAR MATERIAL EM AULA – ETAPA 1	109
FIGURA 62 – INTERFACE PARA ALOCAR MATERIAL A AULA – ETAPA 2	109
FIGURA 63 – INTERFACE PARA ALOCAR MATERIAL EM AULA – ETAPA 3	110
FIGURA 64 – INTERFACE PARA ORGANIZAR MATERIAL EM AULA – ETAPA 4	110
FIGURA 65 – INTERFACE PARA O ALUNO ESCOLHER UM DOMÍNIO	111
FIGURA 66 – INTERFACE PARA O ALUNO ESCOLHER UM CURSO	112
FIGURA 67 – INTERFACE PARA O ALUNO ESCOLHER UMA AULA	112
FIGURA 68 – INTERFACE PARA O ALUNO ESCOLHER O MATERIAL	113
FIGURA 69A – INTERFACE PARA EXIBIR MATERIAL PARA O ALUNO - CONCEITO	113
FIGURA 69B – INTERFACE PARA EXIBIR MATERIAL PARA O ALUNO – EXPLANAÇÃO	114
FIGURA 69C – INTERFACE PARA EXIBIR MATERIAL PARA O ALUNO - EXEMPLO	114
FIGURA 70 – ARQUITETURA SGC NO CONTEXTO DA ACVA.	115
FIGURA 71 – CATALYSIS - MODELO DE TRABALHO EM ESPIRAL	125
FIGURA 72 – CATALYSIS - PRINCIPAIS ATIVIDADES PARA UM TÍPICO SISTEMA DE NEGÓCIOS.....	126
FIGURA 73 – DIAGRAMA DE SEQÜÊNCIA – ALTERAR USUÁRIO	128
FIGURA 74 – DIAGRAMA DE SEQÜÊNCIA – EXCLUIR USUÁRIO	129
FIGURA 75 – DIAGRAMA DE SEQÜÊNCIA – ALTERAR MATERIAL	130
FIGURA 76 – DIAGRAMA DE SEQÜÊNCIA – EXCLUIR MATERIAL	131
FIGURA 77 – DIAGRAMA DE SEQÜÊNCIA – ALTERAR AULA	131
FIGURA 78 – DIAGRAMA DE SEQÜÊNCIA – EXCLUIR AULA	132

FIGURA 79 – DIAGRAMA DE SEQÜÊNCIA – EXCLUIR MATERIAL DE UMA AULA	133
FIGURA 80 – DIAGRAMA DE SEQÜÊNCIA – ALTERAR CURSO.....	134
FIGURA 81– DIAGRAMA DE SEQÜÊNCIA – EXCLUIR CURSO	134
FIGURA 82 – DIAGRAMA DE SEQÜÊNCIA – EXCLUIR AULA DE UM CURSO	135
FIGURA 83 – DIAGRAMA DE SEQÜÊNCIA – ALTERAR DOMÍNIO	136
FIGURA 84 – DIAGRAMA DE SEQÜÊNCIA – EXCLUIR DOMÍNIO	136
FIGURA 85 – DIAGRAMA DE SEQÜÊNCIA – ALTERAR ASSUNTO.....	137
FIGURA 86 – DIAGRAMA DE SEQÜÊNCIA – EXCLUIR ASSUNTO	138

TABELA 1: COMPARAÇÃO DE AMBIENTES DE APOIO À EDUCAÇÃO A DISTÂNCIA.....	16
TABELA 2 - COMPARAÇÃO ENTRE MODELOS DE ORGANIZAÇÃO DE MATERIAL DIDÁTICO.....	32
TABELA 3 – GLOSSÁRIO DO SGC	55
TABELA 4 – FUNCIONALIDADES DO SGC	56

RESUMO

Para atender a novas necessidades de aprendizagem contínua, diversas formas de ensino têm surgido, dentre as quais o Ensino a Distância. O advento das redes de computadores e da Internet facilitou a comunicação entre alunos e professores e incrementou a criação e distribuição de materiais didáticos. Apesar da existência de grande quantidade de ferramentas de autoria, a aplicação uma metodologia na organização e na apresentação do material didático ainda é uma tarefa pouco explorada.

O objetivo deste trabalho é definir um modelo para a criação, organização e reuso de material didático. Inicialmente são apresentados conceitos básicos, alguns ambientes de autoria de Ensino a Distância, alguns modelos pedagógicos de organização de material didático e uma comparação entre os aspectos desses modelos. Em seguida é proposto um modelo de organização de material, que associa os conceitos de Componentes, Ontologia e Mapas Conceituais. A fim de auxiliar a compreensão do objeto em estudo, são mostrados a modelagem e o protótipo de um sistema, baseados no modelo proposto.

ABSTRACT

In order to accomplish new needs for uninterrupted learning, various forms have been applied, including Distance Education. The coming of networks and of the Internet have increased communication between students and teachers as well as opportunities for creating and distributing didactic materials. In spite of the extensive quantities of softwares already known for the mentioned purpose, a methodology for organizing and presenting didactic material is a task yet quite unexplored.

The aim of this work is to define a model for creating, organizing and re-using didactic material. At first basic concepts, some known environments for Distance Education and models for organizing didactic material are presented as well as a comparison about some aspects of the mentioned models. Further, a model is presented for organizing didactic material having in mind concepts regarding Components, Ontology and Conceptual Maps. So that a better comprehension concerning the aim of this work can be achieved, the modeling and the prototype of a software – based on the proposed model – are presented.

1 INTRODUÇÃO

Costuma-se pensar em Educação como uma atividade associada aos primeiros anos do ser humano, que o prepara profissionalmente para toda a vida e se desenvolve entre as paredes de uma instituição. Entretanto, as novas formas de trabalho, cada vez mais ligadas ao campo intelectual, e a recente evolução nas mais diversas áreas de conhecimento têm gerado a necessidade de aprendizagem permanente, constituindo um constante desafio para alunos e profissionais (LUCENA, 2000, p.118-119). Aqueles que buscam manter-se atualizados pelos meios tradicionais de ensino podem vir a ter problemas como limitação de tempo e dificuldade de deslocamento físico.

Para atender a essas pessoas, diversas formas alternativas de ensino têm surgido, dentre as quais a Educação a Distância (EAD), que pode ser definida como “o ensino por meio de mídia impressa ou eletrônica para pessoas engajadas em um processo de aprendizado em tempo e local diferente do(s) instrutor(es) e dos outros aprendizes” (LUCENA, 2000, p.53). Dessa forma a EAD democratiza o saber, pois disponibiliza informação e educação, para quem encontra restrições de atendimento pelos meios tradicionais de ensino.

Segundo LoboNeto (1998), a EAD surgiu em 1728, quando *Caleb Philipps* anunciou em um jornal de Boston um curso de taquigrafia pelo serviço postal. Desde então, a EAD vem incorporando diversas tecnologias como o rádio, a televisão, o vídeo e, mais recentemente, o computador, promovendo, assim inovações no processo de aprendizagem. O advento das redes de computadores e da Internet facilitou não só a comunicação entre alunos e professores, como também a criação e distribuição de materiais instrucionais para os alunos, no ambiente *World Wide Web* (WWW). Um crescente número de ambientes educacionais está sendo desenvolvido por profissionais que utilizam a tecnologia hipermídia, motivados, principalmente, pelas suas características de representar informações em diversos tipos de mídia e pela facilidade de navegação entre os hiperdocumentos, mediante seus elos de ligação.

A análise desses ambientes educacionais existentes no mercado revela, contudo, a existência de alguns problemas comuns à maioria deles, relacionados com a organização e a apresentação do material didático, que merecem ser objeto de estudo. Incorporando os avanços tecnológicos, os materiais são ricos em imagens, animações, textos, vídeos, e *links* para outros materiais, mas não seguem algum tipo de metodologia pedagógica para seu desenvolvimento, organização e apresentação. Ora, de acordo com Lucena (2000, p.20) “a tecnologia educativa não pode estar desligada da Teoria da Educação, que envolve várias ciências”. Outro aspecto de destaque diz respeito à quantidade de reutilização do material didático existente. Verifica-se que a maioria dos *softwares* educacionais analisados não oferece facilidades para o reaproveitamento desse material, visto que, para ser reutilizado, ele precisa, antes de tudo, estar identificado e classificado dentro de um domínio didático que facilite a sua pesquisa; em outras palavras, inicialmente o material precisa ser encontrado e estar organizado de forma que sua reutilização seja fácil. Uma outra análise pode ser feita levando em consideração o nível de conhecimento que um ou um grupo de alunos já possui sobre determinado assunto. Na maioria dos ambientes educacionais o material didático é repassado a todos os alunos de uma única forma, sem considerar o nível de conhecimento de cada um sobre o assunto que está sendo apresentado.

Após a identificação dos problemas acima relacionados e tendo como enfoque a preparação do material didático, colocam-se algumas questões que nortearão este trabalho de dissertação:

- Como organizar, de forma clara e coerente, o conteúdo didático de um *software* educacional, de maneira a conduzir o aluno por meio de uma estratégia pedagógica?
- Como criar, organizar e divulgar o material didático de forma a ser identificado e reutilizado por outro autor que pretenda desenvolver um curso dentro de um mesmo domínio didático?
- Qual o tamanho ideal de um material didático, levando em conta o trabalho de criação e a facilidade de reutilizá-lo?
- Como oferecer ao aluno o material adequado às suas necessidades, de acordo com o nível de conhecimento que ele já possui sobre o assunto?

Procurando responder às questões acima, este trabalho tem como objetivo geral apresentar uma proposta de organização de recursos didáticos associados a um domínio de

conhecimento e em ambiente de ensino a distância. O modelo apresentado estará ancorado em uma metodologia pedagógica de organização de material didático e inserido no contexto da ACVA¹ – Arquitetura de uma Classe² Virtual Adaptativa (HERNÁNDEZ-DOMÍNGUEZ, 1995, 1997). Para melhor entendimento do modelo proposto, um protótipo de um sistema informatizado será apresentado.

O trabalho foi dividido em três partes. A primeira traz uma fundamentação teórica a respeito do tema “Educação a Distância”. Apresenta sua conceituação, alguns ambientes de autoria, um estudo comparativo da organização do material didático usado nesses ambientes, o projeto ACVA, algumas propostas de modelos para organização de material didático, ferramentas para a construção desses modelos e uma comparação entre eles. Finalmente apresenta os conceitos de componentes e desenvolvimento baseado em componentes e os resultados dessa revisão.

Na segunda parte propõe-se um modelo de organização de recursos didáticos, associados a um domínio, em ambiente de ensino a distância e baseado em componentes. Apresentam-se a análise e modelagem do sistema computacional do modelo proposto.

A terceira parte expõe o protótipo desse modelo. São apresentados o Sistema Gerenciador de Componentes(SGC), as ferramentas para a confecção do protótipo, o funcionamento interno do SGC, a implementação do modelo e uma proposta para utilização do SGC no contexto da ACVA.

Para finalizar, apresentam-se as conclusões obtidas a partir do estudo desenvolvido.

1. *Aspectos relacionados ao ACVA serão discutidos no capítulo 2, seção 2.5 desta dissertação.*
2. *Usa-se o termo Classe equivalente ao termo Aula.*

PARTE I

FUNDAMENTAÇÃO TEÓRICA

2 EDUCAÇÃO A DISTÂNCIA

A expressão EAD “cobre as diferentes formas de estudo em todos os níveis que não se encontram sob a contínua e imediata supervisão dos tutores, presentes com seus alunos na sala de aula, mas, não obstante, se beneficiam do planejamento, orientação e acompanhamento de uma organização tutorial” (HOLBERG apud LANDIM), mediante a combinação de tecnologias convencionais e modernas. A educação a distância pressupõe um processo educativo, sistemático e organizado, que exige não somente uma via dupla de comunicação, como também a instauração de um processo continuado, onde um ou até diversos meios de comunicação devem estar presentes na estratégia de ensino. A escolha dos meios de comunicação dá-se em razão do tipo de público alvo e dos custos operacionais versus a eficácia que se deseja atingir no processo educativo (ARETIO, 1994 apud LANDIM) ³.

2.1 CONCEITO DE EDUCAÇÃO A DISTÂNCIA

“Ensino a Distância é um sistema tecnológico de comunicação bidirecional, que pode ser massivo e que substitui a interação pessoal, na sala de aula, de professor e aluno, como meio preferencial de ensino, pela ação sistemática e conjunta de diversos recursos didáticos e pelo apoio de uma organização e tutoria que propiciam a aprendizagem independente e flexível dos alunos” (ARETIO, 1994 apud LANDIM).

3. ARETIO, García. *Educación a distancia hoy*. Madrid: UNED, 1994.

Aretio (apud LANDIM) assinala diversas características da EAD, dentre as quais se destacam:

- Separação professor/aluno – o professor transmite o conhecimento por meio do planejamento da instrução e acompanha o aluno como tutor, instigando-o a pesquisar.
- Aprendizagem independente e flexível – o tutor, além de transmitir conhecimentos, procura tornar o aluno capaz de aprender a aprender e aprender a fazer. O aluno dispõe de mais liberdade em relação ao tempo e ao ritmo de aprendizagem, além de mais estímulo.
- Comunicação bidirecional – a atividade educativa, como processo de comunicação, é bidirecional, com o conseqüente *feedback* entre o professor e o aluno.
- Comunicação massiva – as novas tecnologias nos meios de comunicação eliminam fronteiras de espaço e de tempo, facilitando a educação para um grande número de pessoas, dispersas geograficamente.
- Comunicação individualizada – as novas correntes educativas tratam o aluno individualmente, com suas motivações e necessidades, respeitando seu ritmo de aprendizagem. Dessa forma, pode haver um processo de formação personalizada do conteúdo.

Segundo Aretio (apud LANDIM), comparada com a educação convencional, a EAD oferece alguns benefícios: facilidade de espaço (onde estudar), tempo (quando estudar), ritmo (em que velocidade aprender) e possibilidade de formação permanente e pessoal, no trabalho ou em casa, visando ao aprimoramento cultural. Em contrapartida, são encontradas limitações: dificuldade de socialização; empobrecimento da troca direta de experiências, proporcionada pelas relações educativas e pessoais entre o professor e o aluno e entre alunos; o perigo da homogeneidade dos materiais didáticos, conjugado a poucas ocasiões de diálogo entre o professor e o aluno, além do fato de que o aluno deva possuir, para determinados cursos, um nível elevado de compreensão de textos, e deva saber utilizar recursos multimídia.

2.2 EDUCAÇÃO A DISTÂNCIA NO CONTEXTO DA INTERNET

A Instrução Baseada na *Web* (*IBW*) pode ser definida como um meio para a publicação do material de um curso, apresentação de tutoriais, aplicação de testes e comunicação com estudantes. A ênfase em comunicação será muito grande, e ela também compreende o uso da *Web* para apresentação de conferências multimídia de forma síncrona ou assíncrona (LUCENA, 2000, p.53-54).

Lucena (2000, p.54-55) caracteriza a *IBW*, em relação aos seus recursos, com base em três tipos de elementos:

- Conteúdo Multimídia – expande substancialmente a variedade de recursos que o aprendiz pode usar para participar do processo de aprendizado.
- Assincronismo – dá suporte ao aprendizado autoplanejado, por meio de agendas flexíveis.
- Obrigatoriedade de leitura – como as informações são apresentadas dinamicamente na tela sob forma de texto, áudio, arquivos de vídeo e animações, o aluno precisa lê-las e analisá-las.

É necessário considerar vários aspectos na utilização da *Web*. Para combater o isolamento do aluno, deve-se encorajá-lo a trabalhar em grupo; o curso não deve sobrecarregar o aluno de informações para evitar o fenômeno “perdido no hiperespaço”⁴; precisa-se tomar cuidado na formulação de temas para interação e discussão, pois os alunos têm velocidades diferentes de aprendizado; deve-se atentar para as dificuldades e o medo diante de novas tecnologias; o *site* deve ser atrativo e estar em constante atualização para estimular o aluno a visitá-lo (LUCENA, 2000, p. 55-56).

No ensino via *Web*, a forma de interatividade entre o professor e o aluno, que é completamente diferente da convencional, a começar pelos horários de atendimento, que podem ser feitos de forma síncrona ou assíncrona. Além disso, o professor deve preocupar-se com o material do curso, procurando apresentá-lo de forma atrativa para convencer o aluno a fazê-lo.

4. *Sensações de saturação e fadiga causadas pelo fluxo contínuo de informações quando o site do curso é preenchido com grande quantidade de informações, em diversos tipos de mídia.*

2.3 AMBIENTES DE EDUCAÇÃO A DISTÂNCIA

Na análise de ambientes voltados para educação a distância apoiados na Internet, deve-se primeiramente considerar as funcionalidades oferecidas aos principais usuários do sistema: professores, alunos e administradores do sistema. Para definir as principais funcionalidades de apoio a esses três grupos de usuários, este trabalho baseia-se em um trabalho desenvolvido pelo *CTTT* (apud SANTOS, 1999)⁵.

São funcionalidades de apoio ao professor:

Suporte ao planejamento do curso – ferramentas de apoio ao projeto inicial do curso, que fornecem esboço hierárquico ou de estrutura detalhada, por meio da definição de unidades, tópicos, descrições e ligações de referência para páginas dentro ou fora do ambiente de aplicação.

Suporte à apresentação do curso – ferramentas para o professor formatar e apresentar aos alunos os objetivos da aprendizagem e as informações gerais sobre o curso, tais como organização das aulas, listas de tópicos, critérios de avaliação, conteúdo do curso, tarefas a serem realizadas e questões que devem ser respondidas.

Suporte ao gerenciamento do curso – ferramentas que capacitam o professor a coletar informações relacionadas ao progresso dos alunos ou grupos na estrutura do curso, e podem eventualmente permitir ou vetar o acesso a determinados recursos educacionais.

Suporte à reestruturação rápida do curso – ferramentas que proporcionam facilidades para troca da estrutura do curso e suas atribuições. Isso pode ser feito por meio da importação de materiais do curso, movimentação de módulos ou revisão/adição de materiais.

Suporte à avaliação – ferramentas para auxiliar o processo de avaliação mediante envio e retorno de perguntas práticas, testes, exames, feitos de modo individual ou em grupo. Eventualmente, os sistemas podem proporcionar ferramentas que auxiliem o processo de elaboração de questões.

São funcionalidades de apoio ao aluno:

Suporte à pesquisa na *Web* – ferramentas para vista de documentos *HTML* (*Hypertext Markup Language*) que possuem ligações para qualquer lugar da *Web*.

5. *CTTT – Center for Curriculum Transfer and Technology: Innovation and Collaboration in Post-secondary Education. Disponível em < <http://www.c2t2.ca> >.*

Suporte à comunicação síncrona – ferramentas para troca de informações de forma *on-line*, como *chat*, vídeoteleconferência e espaços virtuais.

Suporte à comunicação assíncrona – ferramentas para troca de informações de forma não *on-line*, a exemplo de correio eletrônico e acesso a grupos de discussão.

Suporte à cooperação síncrona – ferramentas para o compartilhamento de uma base de dados, tais como suporte à co-autoria, co-anotação, *whiteboard* e *browsing* de grupo.

Suporte à cooperação assíncrona – ferramentas para troca de arquivos de forma não *on-line* para o armazenamento e recuperação das informações obtidas e geradas individualmente e em grupo, isto é, a capacidade de fazer *download* e *upload* (cópia de documentos para um servidor) de arquivos da *Web*.

Suporte à descrição dos participantes – ferramentas para conhecimento inicial do professor e dos colegas que trabalharão cooperativamente. Pode-se por exemplo, proporcionar um suporte a *homepages* para fotografias, experiências, interesses e informação para contato dos participantes.

Suporte à percepção das ações dos outros participantes – ferramentas para percepção do que os outros estão fazendo ou fizeram anteriormente. Diversos objetos como teleapontadores (para mostrar a posição dos *mouses* dos outros participantes no espaço de trabalho compartilhado) podem ser utilizados.

Suporte à tomada de decisão – ferramentas que auxiliam os alunos de um grupo na solução de problemas. Por exemplo: ferramentas para categorização das propostas a serem votadas, a votação propriamente dita e a contagem dos votos.

Suporte à coordenação das atividades em grupo – ferramentas para planejamento das atividades e divisão das tarefas a serem realizadas em grupo, para que este alcance seus objetivos de forma organizada e produtiva.

Suporte à auto-avaliação – ferramentas de perguntas práticas relacionadas ao conteúdo do curso. Sua finalidade é possibilitar o conhecimento de lacunas em relação ao domínio de estudo ou detectar imperfeições no estilo pessoal de aprendizado.

Suporte ao estudo efetivo – ferramentas que apóiam a prática de um estudo efetivo. Esse suporte se realiza por meio da anotação privada, da geração de um guia de estudo com tópicos selecionados pelo aluno, de minicursos sobre como estudar ou de ferramentas para revisão do domínio de estudo.

São funcionalidades de administração do sistema:

Suporte à autorização para acesso – ferramentas usadas para dar acesso e outros privilégios a certos grupos de usuários. Esses acessos podem ser especificados pelo administrador para que, mediante senhas, somente os elaboradores de curso tenham acesso à criação e às ferramentas de gerenciamento do curso, e apenas os alunos identificados como participantes do curso tenham acesso a ele.

Suporte à segurança dos dados – ferramentas usadas para evitar a perda dos dados. Para isso, o sistema pode possuir mecanismos de *backup* local.

2.3.1 O AMBIENTE WEBCT

DESCRIÇÃO DO AMBIENTE

Desenvolvido pelo grupo de Murraw W. Goldberg, da *University of British Columbia*, no Canadá, o *Web Course Tools (WebCT)* fornece um conjunto de ferramentas que facilita a criação de cursos educacionais baseados no ambiente *WWW*. A interação com o *WebCT* é baseada em documentos *HTML*, tanto para o estudante como para o autor, não sendo necessário criar versões especiais do *software* para diferentes plataformas. O *software* é processado a partir de um servidor central, podendo, portanto, ser acessado de qualquer local, onde um computador conectado à Internet esteja disponível. Possui grande variedade de ferramentas e características que podem ser adicionadas a um curso, como *chat*, trilha do progresso do aluno, organização de projeto em grupo, auto-avaliação do aluno, controle de acesso, ferramentas de navegação, correio eletrônico, geração de índice automático, calendário de curso, *homepages* dos alunos, pesquisas de conteúdo dos cursos, etc. Cada um dos cursos desenvolvidos no *WebCT* está organizado em torno de uma *homepage* principal que é o ponto de entrada do curso (*WebCT*).

Os usuários do *WebCT* estão organizados da seguinte forma (*WebCT*):

- **Administrador** – responsável pelo servidor *WebCT*; é quem autoriza a criação de novos cursos e de usuários *designers*, pessoas que se responsabilizam por cada curso.
- **Designer** – é o professor, na maioria dos casos. Responsável por um curso, cria suas próprias páginas, monta-o, publica-o na Internet, envia-o para o servidor *WebCT*; também cadastra os alunos e lhes atribui notas. Pode, ainda, criar um usuário especial, chamado “monitor”.
- **Monitor** – pessoa que vai ajudar os professores a administrar os alunos e suas notas. Também pode criar, alterar ou excluir alunos do curso.
- **Aluno** – o usuário que irá seguir o curso criado. Cada curso possui um grupo de alunos que foram cadastrados por um *designer* ou por um monitor.

A INTERFACE PARA AUTORIA DE CURSOS *WEBCT*

A interface que o *WebCT* oferece para autoria de cursos está dividida em 5 passos principais (Figura 1).

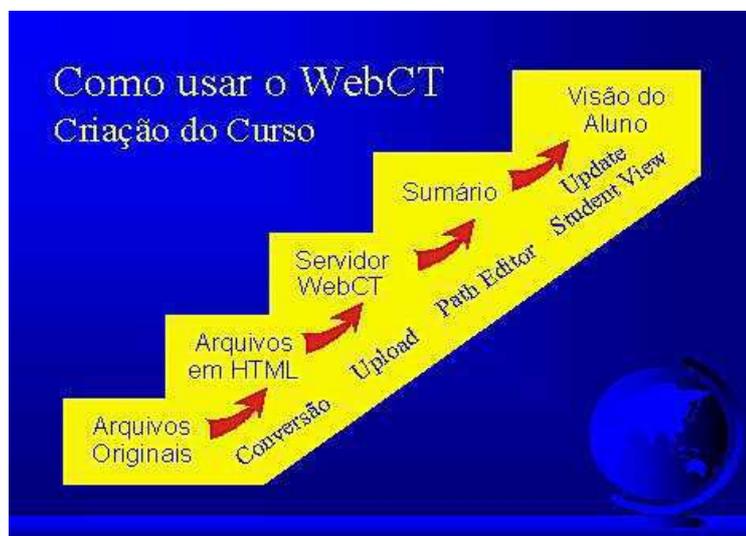


Figura 1 – Roteiro de criação de curso no *WebCT* ⁶

No primeiro passo, “Arquivos Originais”, o autor organiza seu material, que pode estar em qualquer formato — arquivos do tipo *Microsoft Word*, *PowerPoint*, figuras, desenhos, planilhas, etc. No segundo passo, “Conversão”, os arquivos produzidos são convertidos para o formato *HTML*. No terceiro passo “*Upload*”, o autor começa realmente a usar o *WebCT* em três etapas: configuração das páginas iniciais do curso, compactação dos arquivos e *upload* dos arquivos propriamente ditos. No quarto passo, “*Path Editor*”, o autor organiza, com base na definição de um sumário, os arquivos existentes na pasta do curso para criar a visão do aluno. O quinto e último passo é o “*Update Student View*”, em que o professor conclui a formatação da visão que o aluno terá do curso (*WEBCT*).

6. Fonte: <http://jassy.redealuno.usp.br:8900/menusup.html>

2.3.2 O AMBIENTE LEARNING SPACE

DESCRIÇÃO DO AMBIENTE

A *Lotus Development Corporation*, subsidiária da *IBM Corporation* criou o *Lotus Learning Space*, um sistema de apoio à EAD, desenvolvido sobre o ambiente de *groupware Lotus Notes*. Possui cinco bases de dados interconectadas, que proporcionam um ambiente para desenvolvimento e entrega de cursos em sala de aula, estruturados da seguinte forma (SANTOS; 1998; LEARNING SPACE):

- *Schedule* (Agenda) – Consiste em um módulo central para que os participantes naveguem pelos materiais do curso, de acordo com o projeto instrucional e a estrutura do curso criada pelo professor. Com esse módulo, os estudantes podem conhecer os objetivos do aprendizado, as tarefas que devem ser realizadas, os prazos marcados para navegação nos materiais do curso, as perguntas que devem ser respondidas, etc. A agenda pode ser desenvolvida e organizada por dias, semanas ou meses, e também por módulos para instrução autodirigida.
- *MediaCenter* (Centro de Mídia) – O professor ou projetista do curso cria o centro de mídia, a base de conhecimento, que possui todo o conteúdo relacionado ao curso.
- *CourseRoom* (Sala de Curso) – Consiste em um ambiente interativo para que os alunos tenham discussões privadas e públicas entre si e com o professor, para compartilhamento de informações e execução de trabalhos em grupo.
- *Profiles* (Descrição dos Participantes) – Esse módulo consiste em uma coleção de descrições dos participantes, alunos e professores, que inclui informação para contato, fotografias, experiências e interesses. A descrição está baseada em *homepages* criadas pelos alunos e professores com informações sobre si mesmos.
- *Assessment Manager* (Gerenciador de Avaliação) – Esse módulo possui uma ferramenta de avaliação que possibilita ao professor enviar perguntas e receber respostas dos alunos de forma privada. As perguntas são colocadas na agenda e enviadas por correio eletrônico para os alunos, que as enviam de volta junto com a resposta acessível somente ao professor.

A INTERFACE PARA AUTORIA DO *LEARNING SPACE*

O centro de mídia, criado professor/projetista do curso possui todo o conteúdo relacionado ao curso e inclui o acesso a fontes externas tal como a *WWW* e outros repositórios de recursos educacionais. A informação pode tomar diversas formas: texto, *vídeoclips*, gráficos, planilhas eletrônicas, simulações, treinamento baseado em computador, etc. Nesse contexto é possível criar taxonomias de palavras-chave para categorização e busca de informação(SANTOS, 1998; *LEARNING SPACE*).

2.3.3 O AMBIENTE TOP CLASS

DESCRIÇÃO DO AMBIENTE

O *TopClass*, sistema de apoio à EAD baseado em rede e desenvolvido pela empresa *WBT Systems*, consiste em uma combinação de ferramentas de aprendizado colaborativo, ferramentas de entrega e de gerenciamento de conteúdo e pessoas. São exemplos dessas ferramentas: e-mail interno, lista de discussão para cada turma criada, quadro de avisos, segurança de acesso, relatórios e estatísticas, pesquisa de conteúdos de cursos. (*TOPCLASS*).

A INTERFACE PARA AUTORIA DO CURSO TOP CLASS

O *TopClass* possui uma ferramenta própria de autoria, o *TopClass Publisher*, que oferece condições para estruturação de cursos por meio das seguintes tarefas (*TOPCLASS*):

- Inclusão de recursos *Web* – usando-se a ferramenta *TC Publisher*, qualquer recurso encontrado na Internet pode ser introduzido nos cursos.
- Definição de testes e questões dissertativas – A ferramenta possui sete tipos de testes do tipo múltipla escolha, verdadeiro ou falso, etc., que podem ou não ser configurados para autocorreção.
- Estruturação modular de conteúdos – A ferramenta permite a criação de uma estrutura modular para o curso, formada por pastas e subpastas que armazenam o conteúdo. Isso permite que o curso seja liberado paulatinamente para os participantes.

2.3.4 O AMBIENTE AULANET

DESCRIÇÃO DO AMBIENTE

Desenvolvido no Laboratório de Engenharia de *Software* do Departamento de Informática da Pontifícia Universidade Católica do Rio de Janeiro (PUC-RJ), o *AulaNet* é um ambiente de aprendizado cooperativo baseado na *Web*, que oferece ferramentas para administração, criação, manutenção e assistência a cursos a distância (SAPIENS, 2000).

O *AulaNet* considera os seguintes atores envolvidos no processo de criação/aprendizado (LUCENA, 2000, p.123):

- **Administrador** – É o facilitador da integração professor/curso/aluno; trata de questões de natureza eminentemente operacional: inscrição de professores, admissão de alunos em cursos, etc.
- **Aluno** – É o usuário final do curso; representa o público-alvo, o destinatário do curso.
- **Professor** – É o criador do curso; participa de todas as etapas, desde a descrição inicial até a entrada dos conteúdos. Poderá ou não ser o responsável pela aplicação do curso, e contará ou não com o auxílio de um professor co-autor.

O *AulaNet* oferece mecanismos de cooperação, coordenação e comunicação, utilizados para troca de informações entre professores e alunos, mediante ferramentas de correio eletrônico, listas de discussão, *newsgroup*, *chat* e videoconferência. Também permite o acompanhamento do curso, fornecendo ferramentas para o planejamento de tarefas (a exemplo de agenda e quadro de aviso) e para a avaliação do aprendizado (provas, trabalhos e exercícios). Como mecanismos de cooperação oferece transparências, apresentação gravada, texto de aula, livro-texto, bibliografia, demonstração, co-autoria de professor e co-autoria de aluno (LUCENA, 2000, p. 130-142).

A INTERFACE PARA AUTORIA DE CURSOS DO AULANET

O ambiente *AulaNet* permite a confecção de cursos com diferentes ênfases — *workshops*, cursos não estruturados, planos de aula, transparências, textos e recursos de

animação e apresentações, etc. A ferramenta suporta os seguintes tipos de arquivos: *HTML, PDF, PPT, Shockware, Applets Java, Gif/jpg e RM.*

Sobre o conteúdo didático, Lucena enfatiza que:

A solução proposta pelo AulaNet baseia-se na separação entre a autoria do conteúdo e o esforço de programação necessário para implementar a navegação. Dessa forma, o AulaNet cria um ambiente para interação, mas não interfere na autoria do conteúdo didático, que é feito off-line – portanto, fora do ambiente (LUCENA, 2000, p.126).

Assim, o professor pode aproveitar a maior parte dos conteúdos didáticos já existentes, e transferi-los para o servidor *AulaNet*, durante a preparação da aula. Para a exibição do conteúdo didático da aula Lucena (2000, p.146) enfatiza que “o *AulaNet* deliberadamente não oferece nenhum tipo de sincronização entre os diferentes conteúdos, pois entendemos que o aprendiz deve ter o controle do processo”.

2.4 COMPARAÇÃO ENTRE OS AMBIENTES APRESENTADOS

Para se ter uma visão geral dos ambientes apresentados, construiu-se a tabela seguinte, que procura evidenciar as diferenças entre eles. Foram aplicados como itens de comparação as funcionalidades para alunos, professores e administradores, definidas no trabalho desenvolvido pelo *CCTT*, apresentado na seção 2.3 .

Funcionalidades de apoio ao professor	WebCT	Learning Space	TopClass	AulaNet
Suporte ao planejamento do curso	Não	Não	Não	Não
Suporte à apresentação do curso	Sim	Sim	Sim	Sim
Suporte ao gerenciamento do curso	Não	Não	Sim	Não
Suporte à reestruturação rápida do curso	Não	Não	Não	Não
Suporte à avaliação	Sim	Sim	Sim	Sim
Funcionalidades de apoio ao aluno				
Suporte à pesquisa na <i>Web</i>	Sim	Não	Sim	Sim
Suporte à comunicação síncrona	Sim	Sim	Sim	Sim
Suporte à comunicação assíncrona	Sim	Sim	Sim	Sim
Suporte à cooperação síncrona	Não	Sim	Não	Sim

Funcionalidades de apoio ao professor	WebCT	Learning Space	TopClass	AulaNet
Suporte à cooperação assíncrona	Não	Sim	Não	Sim
Suporte à percepção de outros participantes	Não	Sim	Não	Não
Suporte à tomada de decisões	Não	Não	Sim	Não
Suporte à coordenação das atividades em grupo	Sim	Sim	Sim	Sim
Suporte à auto-avaliação	Sim	Não	Sim	Não
Suporte ao estudo efetivo	Não	Não	Não	Não
Funcionalidades de Administração	WebCT	Learning Space	TopClass	AulaNet
Suporte à autorização para acesso	Sim	Sim	Sim	Sim
Suporte à segurança de dados	Sim	Sim	Sim	Sim

Tabela 1: Comparação de ambientes de apoio à Educação a Distância

Observando essa tabela, pode-se verificar uma tendência comum aos ambientes nos itens que se referem a disponibilizar ferramentas tecnológicas de apoio aos participantes como *chats* e correio eletrônico. Em relação às funcionalidades de apoio aos professores, verifica-se que a maioria dos ambientes estudados optou por deixar a cargo do professor/autor do curso a organização do material didático, não apresentando qualquer proposta metodológica educacional para a preparação desse material. Geralmente ele é preparado *off-line* pelo autor do curso e só depois relacionado ao tópico pertinente. Alguns usam avaliações como forma de controle e liberação de módulos para os participantes, enquanto outros deixam ao encargo do aluno a forma de estudar e percorrer os módulos do curso.

Outra observação a ser feita diz respeito às necessidades individuais do aprendiz. Nos ambientes educacionais analisados, não existe a opção de avaliação prévia do nível de conhecimento do aluno para a montagem do curso de forma a oferecer-lhe uma aprendizagem diferenciada e adaptada a seu nível de conhecimento. Esse é um aspecto relevante quanto ao uso dos recursos tecnológicos porque se contrapõe às propostas educacionais inovadoras.

Em relação ao item reestruturação rápida do curso, observa-se que, na maioria dos ambientes avaliados, ela não pode ser considerada rápida e prática, em consequência da falta de um modelo de organização do material didático. Novas propostas de desenvolvimento de cursos defendem um modelo de criação de material didático baseado em componentes, que,

definidos como parte de um todo, podem ser montados e reaproveitados. No capítulo 3 será apresentado um estudo sobre desenvolvimento baseado em componentes.

2.5 O PROJETO ACVA

2.5.1 DESCRIÇÃO DO AMBIENTE

A ACVA, conforme Hernández-Domínguez (1995, 1997), é um modelo para o desenvolvimento de um ambiente de ensino e aprendizagem suportado por um sistema de comunicação e controlado por computador, para transmitir o conhecimento a grupos virtuais de estudantes. Propõe o controle sobre a comunicação e sobre algumas atividades de ensino. O professor adquire as funções de orientar o aluno, controlar sua progressão dentro de um grupo e distribuir recursos. A proposta está embasada pedagogicamente no método de ensino cooperativo e nos princípios dos STIs – Sistemas de Tutores Inteligentes, em que as funções de ensino são distribuídas considerando a participação de professores, sistemas, e recursos (Figura 2).

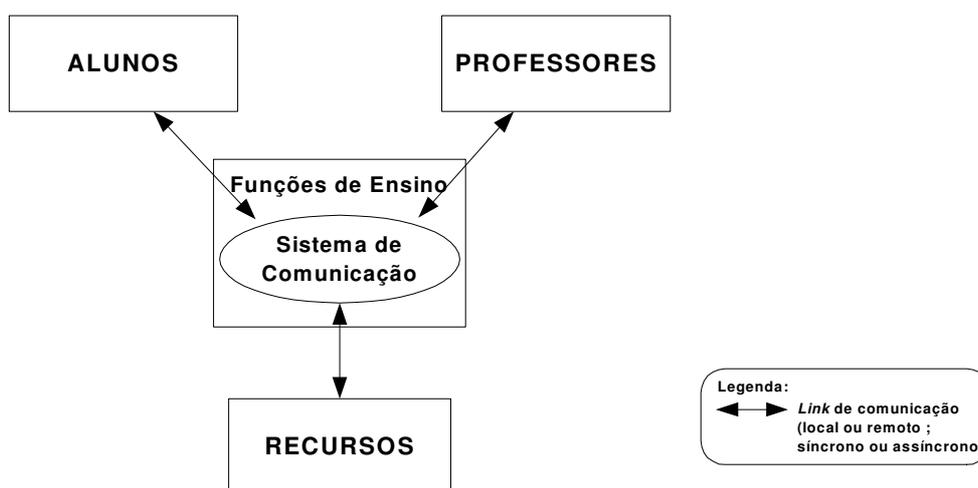


Figura 2 – Arquitetura geral para cada serviço de treinamento adaptado ⁷

7. Fonte: (SILVA, 2001)

A ACVA é composta de aulas virtuais que reúnem diversos grupos, e cada grupo é formado por alunos de diferentes níveis de conhecimento.

Tal arquitetura é dita adaptativa porque permite e controla a mobilidade dos estudantes entre os diversos grupos heterogêneos ou mesmo dentro desses grupos (intergrupo ou intragrupo). Essa mobilidade ocorre conforme as necessidades educacionais dos alunos, detectadas a partir dos seus comportamentos durante uma sessão de ensino/treinamento⁸ (SILVA, 2000, p.33)

Quatro níveis ou camadas compõem a arquitetura da ACVA (Figura 3), a saber (HERNÁNDEZ-DOMÍNGUEZ, 1995 apud SILVA, 2000):

- **Nível SF_CV** – A camada de Serviços de Formação de Classe Virtual gerencia uma sessão de ensino/treinamento de uma aula virtual, composta de um conjunto de grupos remotos. Oferece assistência pedagógica remota, controla os objetivos pedagógicos a atingir e controla a mobilidade lógica dos alunos nos grupos de aprendizagem, que representa a progressão ou regressão do aluno em relação ao grupo.

- **Nível SF_Grupo** – A camada de Serviços de Formação de Grupos oferece estrutura para controle didático de grupos de estudantes, ou seja, representa um sistema tutor de um grupo. Assim, cada grupo de estudantes é associado a um nível de conhecimento particular (exemplo: iniciante, intermediário e avançado). Caso um aluno não esteja adaptado ao nível de conhecimento do seu grupo, ele é transferido de grupo.

- **Nível SF_Básico** – A camada de Serviços de Formação Básicos representa um controlador que tem acesso ao servidor de recursos de objetos didáticos. Esses recursos são componentes funcionais, independentes e reusáveis, representados na Figura 3 como SIBT (Service Independent Training Building Block)⁹ ou blocos de construção. Serão usados pelo controlador de grupo (tutor), representado no nível SF_Grupo.

- **Nível Suporte** – A camada de Comunicação e Informação oferece estrutura para armazenamento das informações e controle da comunicação entre alunos (intergrupos ou intragrupos) e entre os alunos e o orientador/professor (comunicação individual ou em grupo).

8. Na ACVA, um curso é composto por um conjunto de sessões de ensino/treinamento, que, nesta dissertação, serão chamadas de **Aulas**.
9. Neste trabalho, os SIBTs serão chamados de **Componentes de Aprendizagem** ou apenas **Componentes**.

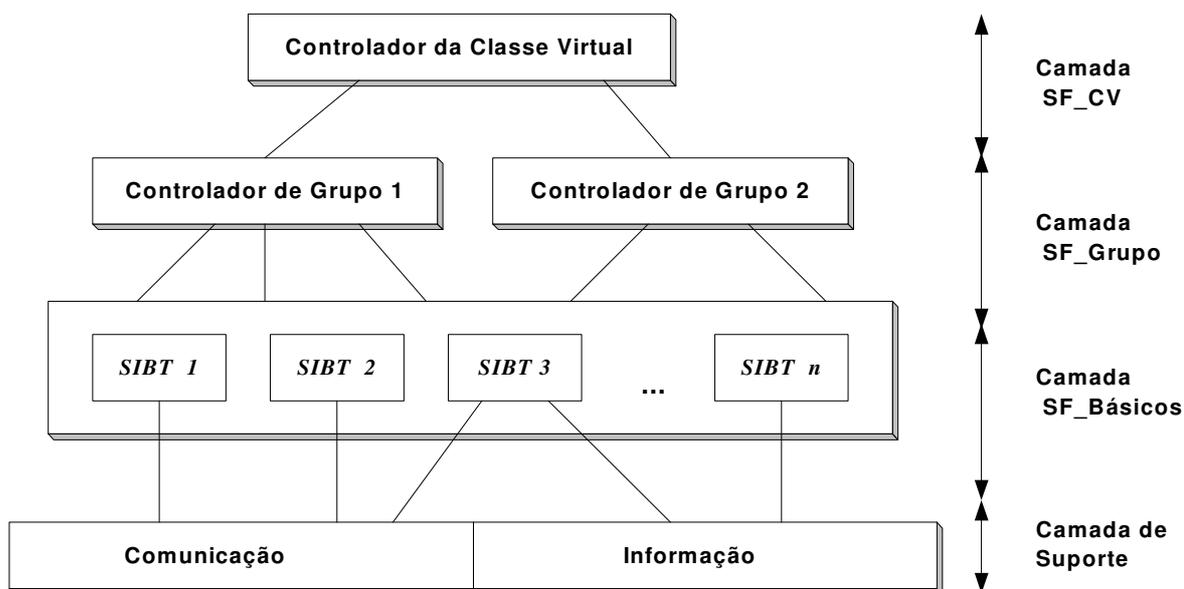


Figura 3 – Camadas da ACVA ¹⁰

2.5.2 DESENVOLVIMENTO DO PROJETO NO CONTEXTO DA ACVA

Pode-se considerar cada camada da ACVA um subsistema, representado por um conjunto de elementos (HERNÁNDEZ-DOMINGUEZ, 1995 apud SILVA, 2000, p. 39). A interação desses subsistemas permitirá o funcionamento da ACVA, dotada de todas as funções propostas.

Este trabalho se integra ao projeto com o objetivo de oferecer uma solução para as camadas de Serviços de Formação Básicos e de Suporte, mediante proposta de um modelo para o desenvolvimento, organização, armazenamento e apresentação de material didático, a ser utilizado via Internet.

A Figura 4 apresenta um diagrama de classes que mostra como a camada de Serviços de Formação Básicos fornecerá serviços para as outras camadas da arquitetura. O coordenador dos Componentes de Aprendizagem deve controlar a interação com os alunos, professores,

autores e coordenadores dos GTS (Serviço de Treinamento de Grupos) e VCTS (Serviço de Treinamento de Classe Virtual).

10. Fonte: (HERNÁNDEZ-DOMINGUEZ, 1995 apud SILVA, 2000, p.37).

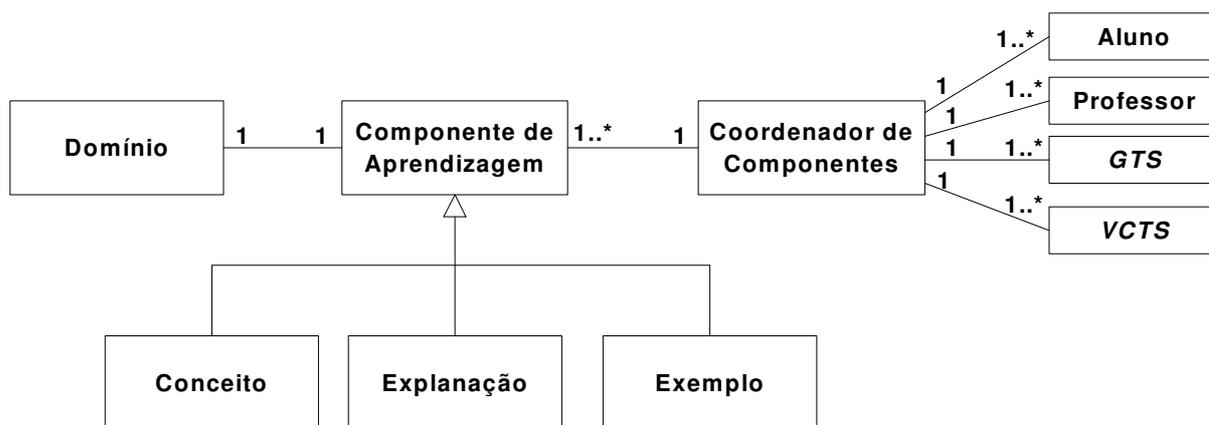


Figura 4 –Serviços fornecidos pela camada SF_Básicos às outras camadas da ACVA

2.6 CONCLUSÃO

Embora a utilização dos recursos técnicos do microcomputador e da Internet seja, hoje, parte fundamental da educação, a distância ou presencial, deve-se evitar que “o meio imponha a metodologia e crie restrições para a instrução”(LUCENA, 2000, p. 55). Por isso, aspectos pedagógicos, organizacionais e instrucionais precisam ser observados.

No capítulo que segue, diversos modelos de representação de recursos didáticos são abordados e comparados, colocando-se em foco o que deles se espera para a definição do modelo objeto deste trabalho.

3 MODELOS DE REPRESENTAÇÃO DE RECURSOS DIDÁTICOS ASSOCIADOS A UM DOMÍNIO

O grande desenvolvimento da Informática, nos últimos anos, apresenta-se como um fator importante a ser explorado em termos de disponibilização e de consumo de materiais didáticos no ambiente *Web*. Nesse contexto, é crescente o número de aplicações, ferramentas e ambientes para suporte à produção e publicação de material didático, via Internet.

No entanto, quando se trabalha com recursos do tipo hipermídia na elaboração de um domínio, tem-se dificuldade na organização do conteúdo, pela própria natureza dos hiperdocumentos, que são ricamente inter-relacionados, permitindo uma navegação sem restrições por meio dos *links*, que podem ser utilizados para associar as informações. De acordo com Ludwig (1997), do ponto de vista didático não é muito conveniente permitir uma navegação sem restrições por todos os elos do hiperdocumento, uma vez que isso pode ocasionar problemas tais como desorientação e sobrecarga cognitiva.

Segundo Giraffa¹¹ (1999 apud Cabral, 2001, p.3), “um ambiente educacional, ao ser projetado, traz consigo o conjunto de crenças sobre Educação que o projetista possui e professa. Assim sendo, acredita-se que um curso a distância pode ser mais qualificado se questões metodológicas forem consideradas”. Aceita essa colocação, procurou-se incrementar qualidade pedagógica ao modelo objeto deste trabalho, tornando-se necessário o estudo de propostas pedagógicas para a modelagem e a organização de recursos didáticos.

11. Girafa, L.M.M. *Uma Arquitetura de Tutor Utilizando Estados Mentais*, 1999. Tese (Doutorado em Ciência da Computação) – PGCC, Universidade Federal do Rio Grande do Sul, Porto Alegre.

3.1 MODELOS DE ORGANIZAÇÃO DE MATERIAL DIDÁTICO

3.1.1 ÁRVORE CONCEITUAL

Conforme representado na figura 5,

Um domínio do conhecimento consiste num conjunto de conceitos, temas, definições, problemas, soluções e referências que têm aplicações comuns distintas do processo educacional. Uma árvore conceitual é a representação estruturada de um domínio do conhecimento, em forma de diagrama hierárquico, possibilitando a visualização gráfica deste domínio e dos relacionamentos entre seus módulos conceituais. (REZENDE, 2000, p.2).

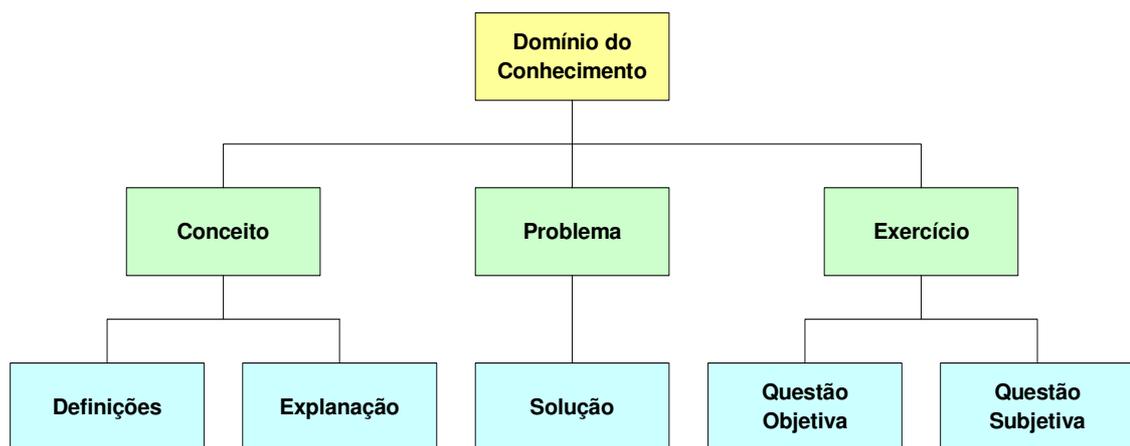


Figura 5 – Árvore Conceitual

3.1.2 ONTOLOGIA

Ontologia é um documento que define formalmente as relações entre os termos, dentro de um determinado contexto. Uma ontologia considera a impossibilidade de se representar o mundo em todos os detalhes, por isso procura restringir a atenção a um pequeno número de conceitos que sejam significativos o suficiente para interpretar o mundo e fornecer a representação adequada a uma certa tarefa. Como consequência, pode-se representar o conhecimento mediante a elaboração de um conjunto de conceitos existentes num certo domínio, e a explicação das relações que podem surgir entre eles.

No sentido filosófico, Ontologia é um sistema específico de categorias que reflete uma visão específica do mundo. No contexto de sistemas baseados em conhecimento, ontologia é definida como um vocabulário específico para descrever uma certa realidade, de acordo com o sentido intencional das palavras usando para isso um conjunto de premissas (GUARINO, 1998, p.2). Em outras palavras, trata-se de um vocabulário comum, que será usado na definição de objetos e outras entidades, e na relação e troca de mensagens entre eles. Empregadas nesse contexto, as ontologias permitem o compartilhamento e a reutilização desse vocabulário comum e uma classificação mais precisa de aplicações de domínio. Guarino (1998, p.7-8), define quatro tipos de ontologias, conforme seus níveis de generalização em relação a uma tarefa.(Figura 6):

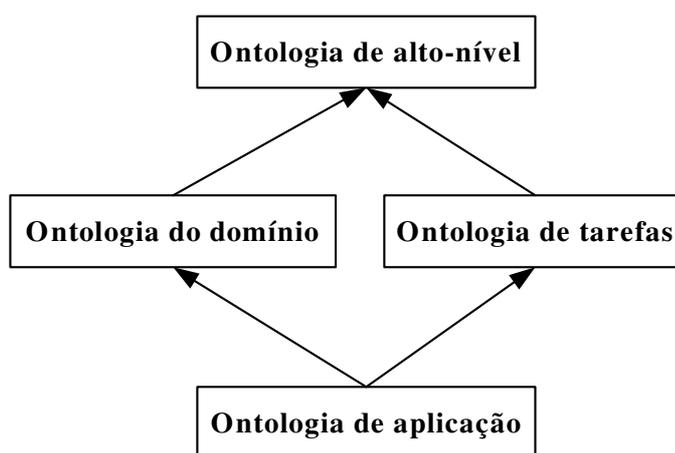


Figura 6 – Tipos de ontologia, de acordo com o nível de dependência em uma tarefa¹²

- **Ontologia de alto nível** – Descreve conceitos gerais como espaço, tempo, eventos, ações, etc., que independem de um problema ou domínio em particular, podendo ser usados por um grande número de grupos de usuários.

- **Ontologia de domínio e de tarefas** – Descreve, respectivamente, o vocabulário relativo a um domínio genérico, como Medicina ou automóveis, ou uma tarefa ou atividade genérica, como diagnóstico ou vendas, especializando os termos definidos na ontologia de alto nível.

12. Fonte: (GUARINO, 1998, p.7)

- **Ontologia de aplicação** – Descreve conceitos de um domínio ou de uma tarefa em particular, podendo ser também uma especialização de ambos. Esses conceitos sempre correspondem a funções do domínio, quando processam uma determinada atividade.

Pode-se compor uma ontologia mediante uma taxonomia e um conjunto de regras de inferência. Uma taxonomia define as classes de objetos e as relações entre eles e, quando corretamente definida, preserva o significado específico dos termos e expressões de um domínio do conhecimento. Para se criar uma taxonomia em uma área do conhecimento é preciso definir os conceitos mais usados na área e a relação entre eles. Essa relação pode ser representada de diversas formas, como, por exemplo, uma tabela ou uma árvore hierárquica (SANTOS, 2001, p.1). O trabalho desenvolvido pelo CCTT – *Center for Curriculum Transfer and Technology*, do Canadá, apresentado na seção 2.3, é um exemplo de ontologia para definir um conjunto de funcionalidades necessárias aos principais usuários do sistema de aprendizagem.

O projeto CardioEducar (ROCHA, 2000), um ambiente educacional, oferece suporte para o ensino de Cardiologia, via Internet, com a tecnologia *Web*, e utiliza a ontologia para a definição do domínio do conhecimento na preparação do material didático, valendo-se do *software* de autoria FabriCar (Fábrica de Autoria de Tutores Inteligentes Hipermedia para Cardiologia). Esse *software* utiliza dois tipos de ontologia: de domínio e de tarefa. A ontologia de domínio é usada para expressar a conceituação específica para o domínio, referindo-se ao vocabulário da Cardiologia. As ontologias de tarefas descrevem uma tarefa ou atividade genérica (diagnóstico, planejamento, monitoração, etc.).

A arquitetura do *software* (Figura 7) é composta de três níveis: o nível do conhecimento, que representa a ontologia de domínio da Cardiologia, o nível da modelagem da tarefa, particular de cada ferramenta, e o nível operacional, em que os casos utilizados no processo são armazenados em uma base de dados.

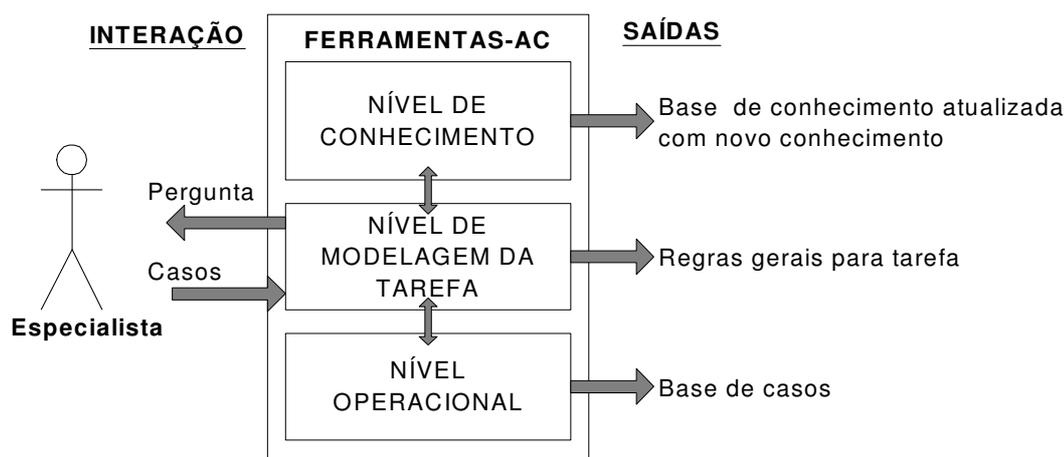


Figura 7 – FabriCar - Arquitetura das ferramentas de aquisição do conhecimento.¹³

3.1.3 MAPA CONCEITUAL

APRENDIZAGEM SIGNIFICATIVA

O uso de Mapas Conceituais para organização de material didático, proposto por Novak, baseia-se na teoria sobre Aprendizagem Significativa, de David Ausubel (1963, 1968, 1986 apud NOVAK, 1998, 2001)¹⁴. A idéia fundamental de Ausubel é que a aprendizagem ocorre quando uma nova informação se relaciona com um aspecto relevante da estrutura de conhecimento do aprendiz, ou seja, quando essa nova informação se ancora em conceitos ou proposições relevantes preexistentes na estrutura cognitiva do indivíduo (NOVAK, 2001. p.2).

Esse pesquisador faz distinção entre aprendizagem significativa e aprendizagem mecânica. A aprendizagem mecânica consiste na simples memorização da informação, sem ênfase em um possível relacionamento entre os conceitos aprendidos. Em contrapartida, a aprendizagem significativa resulta da ancoragem de novas informações em conceitos relevantes preexistentes na estrutura cognitiva do indivíduo (NOVAK, 2001).

13. Fonte: (ROCHA, 2000, p. 6).

14. AUSUBEL, D.P. *The Psychology of Meaningful Verbal Learning*. New York: Grune and Stratton, 1963.
 AUSUBEL, D.P. *Educational Psychology: A Cognitive View*. New York: Holt, Rinehart and Winston, 1968.

AUSUBEL, D.P. *et al. Educational Psychology: A Cognitive View*. 2nd ed. New York: Warbel & Peck, 1986.

A aprendizagem significativa requer três condições: (AUSUBEL apud NOVAK, 2001, p.2-3)

- O material deve ser conceitualmente claro e potencialmente significativo.
- O aprendiz deve possuir conhecimentos anteriores relevantes. Isto requer cuidados específicos na construção de padrões conceituais.
- O aluno deve estar motivado para a aprendizagem significativa ao invés da memorização de uma aprendizagem mecânica.

No âmbito dessa teoria, a preparação do material didático é feita a partir da identificação dos conceitos gerais do domínio. Em seguida, define-se o conjunto de informações que deve constar no material que será elaborado. Essas informações devem estar ligadas por uma estrutura que inclua desde os conceitos mais abrangentes até os mais inclusivos. Dessa forma, “pode-se usar recursos de hipermídia, com fins educacionais, desde que o aprendiz seja conduzido pelos textos seguindo uma estrutura hierarquizada, estabelecida no programa, para que ele não se perca na navegação” (Cabral, 2001, p.68).

MAPAS CONCEITUAIS

Com base na teoria de Ausubel, Joseph D. Novak, da Cornell University, desenvolveu um trabalho de representação do conhecimento por meio de Mapas Conceituais, uma ferramenta para a organização e a representação do conhecimento. Mapas conceituais descrevem visualmente o relacionamento entre idéias, dentro de um domínio do conhecimento. Seu formato visual oferece um ganho na percepção desse domínio. Para isso, os conceitos são incluídos em círculos ou retângulos e os relacionamentos (proposições) entre eles são indicados por uma linha conectando dois conceitos.

Define-se conceito, em mapas conceituais, como uma regularidade percebida em eventos ou objetos. Eventos podem ser acontecimentos, enquanto objetos podem ser abstrações ou coisas reais do ambiente. Nomes, símbolos e sinais da linguagem denotam, tipicamente, conceitos. Usam-se esses conceitos quando se observam ou se interpretam eventos e objetos. Em suma, conceitos ajudam o indivíduo a descrever e a explicar o mundo como ele é. Proposições são afirmações sobre algum objeto ou evento do universo, naturais ou construídos. Contêm dois ou mais conceitos conectados com outras palavras para formar

uma afirmativa coerente, às vezes chamadas de unidades semânticas ou unidades de significado (NOVAK, 2001).

Outra característica importante dos mapas conceituais é a inclusão de *cross-links*, que são relacionamentos (proposições) entre conceitos em domínios diferentes de um mapa conceitual. *Cross-links* ajudam o indivíduo a perceber como alguns domínios do conhecimento, repensados nos mapas, mantêm relações entre si. Na criação de novos conhecimentos, freqüentemente representam saltos criativos realizados pelo organizador do conhecimento. (NOVAK, 2001, p.2)

Para se entender melhor o que são mapas conceituais, deve-se definir de forma detalhada o que seja a subsunção e os organizadores avançados, constantes na teoria de Ausubel. A palavra “subsumir” vem do latim, onde “*sumere*” significa “tomar, acolher, aceitar”. A subsunção é uma operação que se caracteriza pela classificação, inclusão, incorporação de algo em uma categoria ou em um princípio mais geral (LEITE, 1999, p.2).

“Na teoria da Aprendizagem Significativa, os conceitos subsunçores são conceitos mais gerais e já estáveis, que figuram na estrutura cognitiva de um indivíduo e se prestam à ancoragem (inclusão) de novos conceitos. Para que a ancoragem de novos conceitos seja assumida como uma aprendizagem significativa, o indivíduo deve ter presente em sua estrutura cognitiva não somente os conceitos subsunçores necessários, mas possuí-los num nível adequado àquele processo. Os organizadores avançados são conteúdos introdutórios caracterizados por serem claros e estáveis, relevantes e inclusivos do conteúdo que será oferecido à aprendizagem, e tem o objetivo de “revitalizar” os conceitos subsunçores em termos de aprofundamento e ampliação da abrangência” (MOREIRA E MASINI, 1982 apud LEITE, 1999, p.2)¹⁵.

Mapas conceituais podem ser usados com vários propósitos (PLOTNICK, 1997, p.1):

- Gerar novas idéias;
- Desenhar estruturas complexas como textos longos e *web sites* grandes;
- Transmitir idéias complexas;
- Ajudar na aprendizagem, integrando novos e velhos conhecimentos;
- Avaliar o entendimento ou diagnosticar equívocos.

15. MOREIRA et al. *Aprendizagem Significativa: A teoria de David Ausubel*. São Paulo: Moraes, 1982.

A Figura 8 é apresenta um exemplo de mapa conceitual. Nele se descrevem o conceito de mapas conceituais, sua estrutura e características.

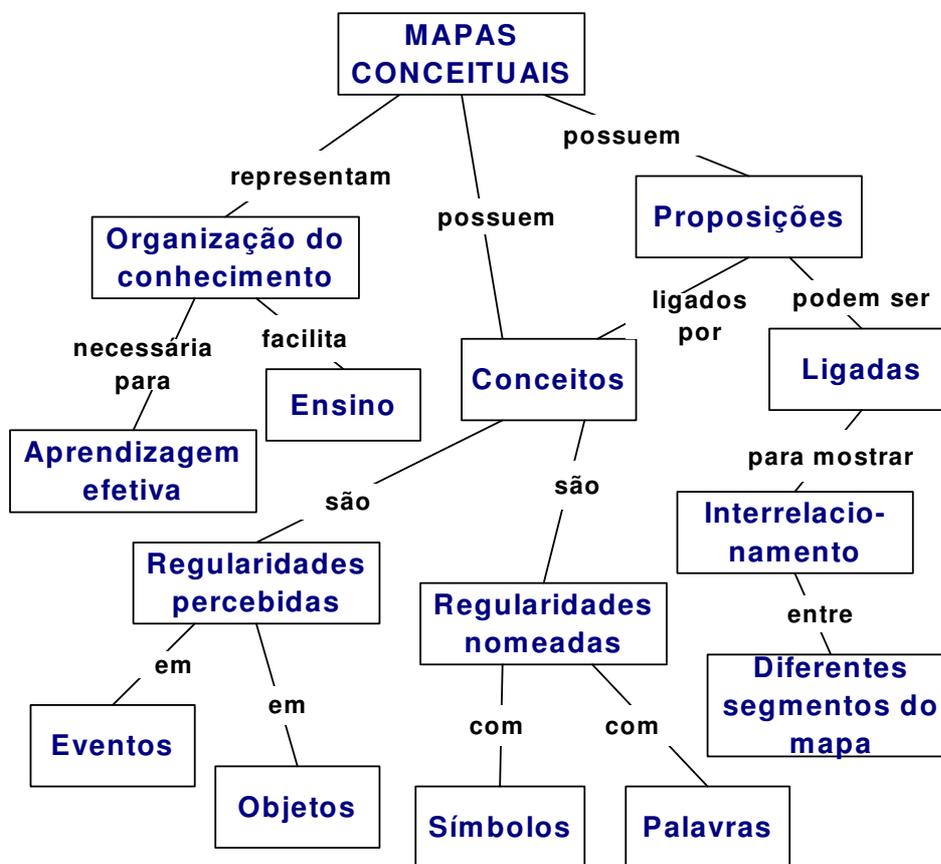


Figura 8 – Exemplo de mapa conceitual¹⁶

Mapas conceituais podem ser usados por alunos e professores. Quando construídos por estudantes, eles os auxiliam na atividade de entender o relacionamento entre conceitos, e ao professor, na tarefa de avaliar se o aluno entendeu o assunto estudado. Quando construído por professores, os mapas ajudam os estudantes a identificar e explorar a natureza do conhecimento dentro de um domínio. Assim, poderão, com mais facilidade, conhecer novos conceitos, relacioná-los a outros já conhecidos e descobrir novos relacionamentos entre conceitos, necessários para o entendimento. Mapas Conceituais também podem ajudar um grupo de professores na construção de um domínio de conhecimento, de forma colaborativa (Figura 9).

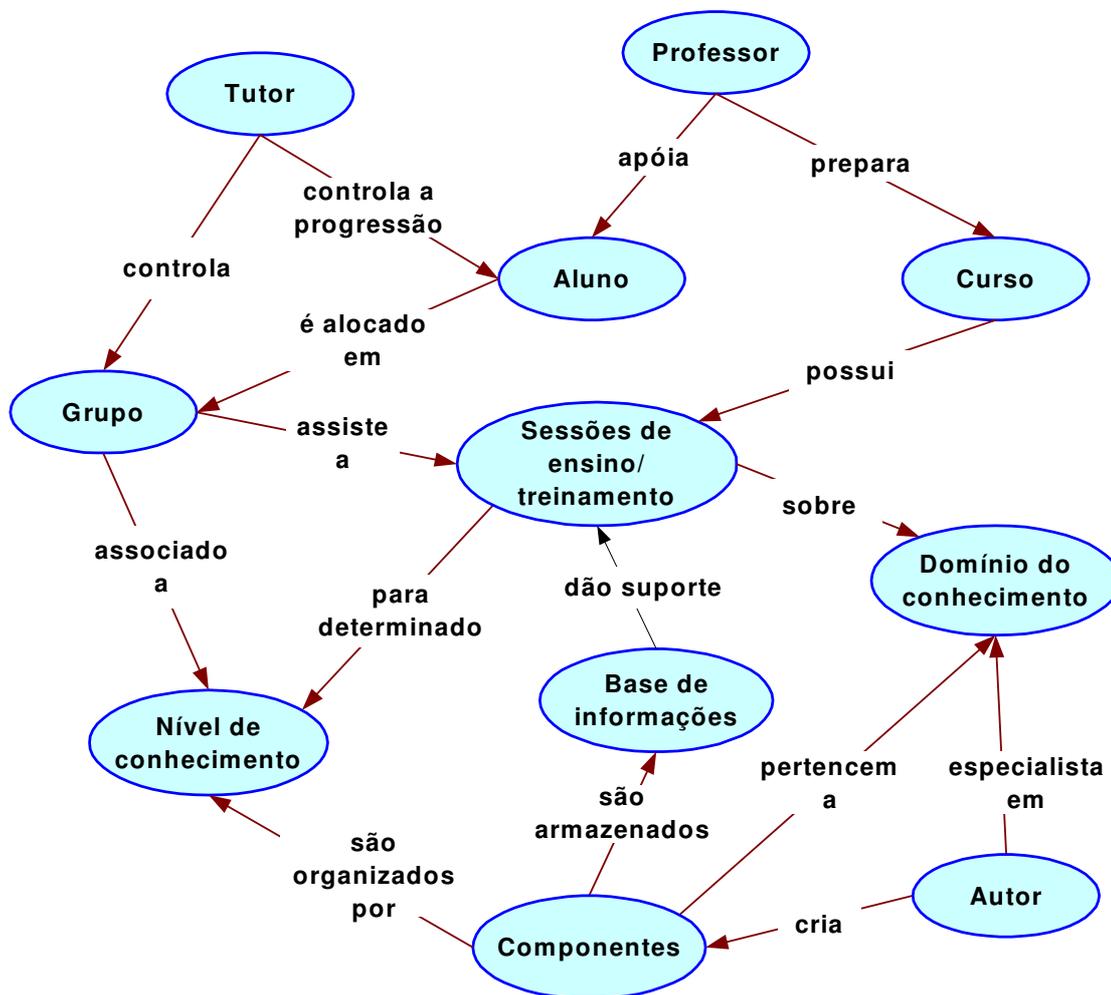


Figura 9 –Mapa conceitual descrevendo um ambiente de EAD

Novak (2001) explica que, na aprendizagem para a construção de mapas conceituais é importante começar com um domínio do conhecimento que seja familiar para a pessoa que está construindo o mapa, e apresenta um roteiro para construí-los:

- Escolha o assunto e identifique o contexto, limitando o domínio;
- Identifique os conceitos-chave aplicáveis a esse domínio;
- Liste e classifique os conceitos do mais geral (inclusivo) para o mais específico.

Ainda que a estrutura seja provisória, isso ajudará na construção;

- Construa um mapa preliminar;

- Procure *cross-links*, que são elos entre domínios diferentes do conhecimento e podem ajudar a ilustrar como esses domínios estão relacionados;
- Revise o mapa várias vezes até chegar a uma versão final.

UTILIZAÇÃO DE MAPAS CONCEITUAIS NA CONSTRUÇÃO DE UM MODELO DE ORGANIZAÇÃO DE MATERIAL DIDÁTICO

Mapas conceituais podem ser usados de forma integrada para planejamento de currículos e material instrucional (NOVAK, 2001). Para o planejamento curricular (Figura 10), precisa-se construir um mapa conceitual macro (global), mostrando as idéias principais que se planeja apresentar durante o curso. Para segmentos mais específicos do programa instrucional, constroem-se micromapas conceituais (Figura 11).

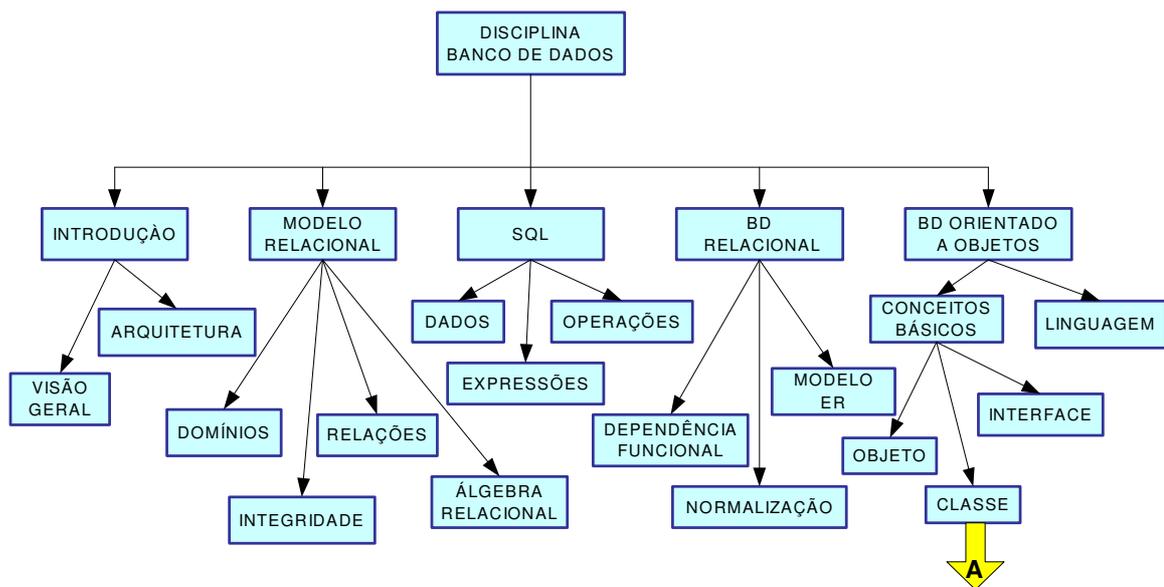


Figura 10 – Mapa conceitual usado para planejamento de currículo - macro

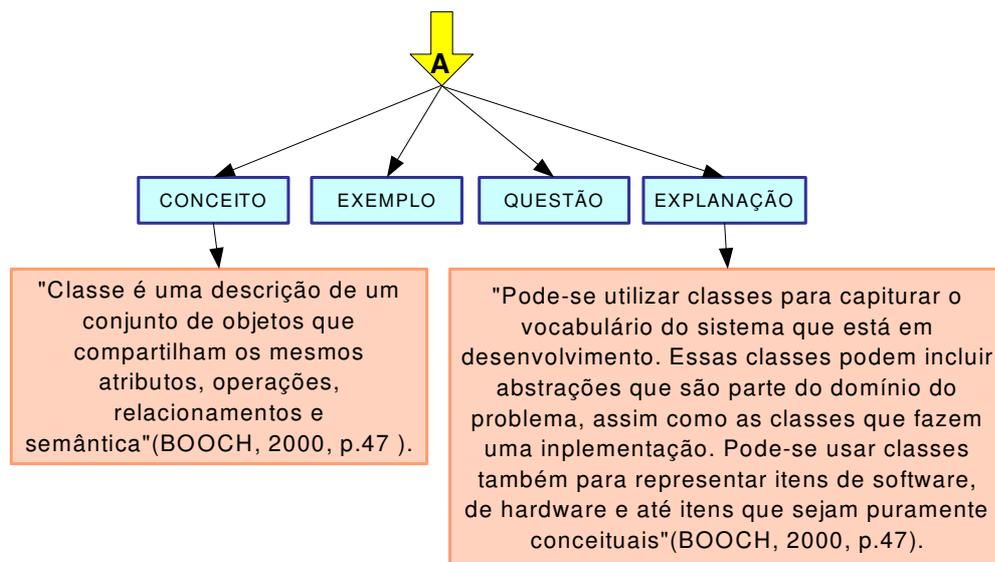


Figura 11 – Mapa conceitual usado para conceitos específicos do programa instrucional – micro

Existem diversas ferramentas no mercado utilizadas para a construção de mapas conceituais, como por exemplo (CABRAL, 2001, p.41-60): *CMAP Tools*, *Inspiration* e *Decision Explorer*¹⁷.

3.2 COMPARAÇÃO DAS DIFERENTES ABORDAGENS E MODELOS DE REPRESENTAÇÃO DE MATERIAL DIDÁTICO

Para tornar mais clara a comparação entre os modelos, elaborou-se a tabela 2, onde eles são analisados com base nas questões colocadas na introdução deste trabalho, com enfoque na modelagem e na preparação de material didático.

QUESTÕES	ÁRVORE CONCEITUAL	ONTOLOGIA	MAPA CONCEITUAL
Proporciona uma organização do material?	Sim	Sim	Sim
Permite considerar o nível de conhecimento que o aprendiz possui?	Sim	Sim	Sim
Facilita a divulgação e a identificação do material	Não	Sim	Não

didático?			
Proporciona a criação de material em um tamanho que facilite a reutilização?	Não	Não	Não

Tabela 2 - Comparação entre modelos de organização de material didático

Como se pode observar, a partir dessa tabela, a organização do material didático, realizada conforme os modelos de árvore conceitual, ontologia e mapa conceitual atendem parcialmente à proposta inicial deste trabalho.

17. .CMAP Tools (<http://www.uwf.com>)

Inspiration (<http://www.inspiration.com>)

Decision Explorer (<http://www.scotnet.co.uk/banxia/demain.html>)

Árvore conceitual é a representação estruturada do domínio do conhecimento, em forma de diagrama hierárquico e mapa conceitual é a representação gráfica por meio de diagramas ligados por palavras e representa uma estrutura que vai desde os conceitos mais abrangentes até os menos inclusivos. Pode ser utilizado para auxiliar na montagem e ordenação dos conteúdos de ensino. Essa ordenação pode ser feita inclusive de forma hierárquica. A comparação desses dois modelos permite concluir que o mapa conceitual pode ser utilizado para representar uma árvore conceitual, que, nesse caso, pode ser vista como um subconjunto dos possíveis modelos representados pelos mapas conceituais.

Quanto à ontologia, foi definida como um vocabulário específico para a descrição de uma certa realidade, (objetos e outras entidades), mediante um conjunto de premissas e conforme o sentido intencional das palavras no vocabulário. A ontologia também é usada na relação e na troca de mensagens entre os elementos dessa realidade. As ontologias de domínio e de tarefas descrevem, respectivamente, o vocabulário relativo a um domínio genérico e uma tarefa ou atividade genérica. Usadas neste contexto, as ontologias permitem que o vocabulário comum seja compartilhado e reutilizado, além de fornecer uma classificação mais precisa de aplicações de domínio.

3.3 CONCLUSÃO

Na definição de um modelo para organização de material didático, os mapas conceituais podem ser utilizados para determinar o “caminho” por onde os alunos deverão seguir, dentro de um curso, de forma a evitar a sobrecarga cognitiva e a desorientação, enquanto a ontologia pode ser usada de forma eficiente para definir o domínio do conhecimento, estimulando a divulgação, o compartilhamento e reuso do material. Essas duas abordagens pedagógicas, usadas associativamente, propiciam a criação de um ambiente de aprendizagem organizado de forma clara e coerente, onde o aluno poderá ser conduzido em seu aprendizado. Porém, para se definir o tamanho ideal do material didático que facilite a sua reutilização, será adotado o conceito de componente.

O capítulo que segue apresenta o conceito de componente e de ambientes de desenvolvimento baseado em componentes, além da proposta para o modelo objeto deste trabalho, que tem como suporte triplo: mapa conceitual, ontologia e componente.

4 DESENVOLVIMENTO BASEADO EM COMPONENTES

O conceito mais geral de componentes é que eles são partes reusáveis e podem ser adaptadas, mas não modificadas, podendo-se compor com outros componentes para formar alguma coisa maior (D'SOUZA, 1998, p.386). Encontram-se exemplos de componentes em diversas áreas tecnológicas, desde o projeto de automóveis e eletrodomésticos até a parte de *hardware* de um novo computador. Atualmente, nenhuma indústria constrói um novo modelo de automóvel a partir do zero. Na verdade, uma nova família deles é projetada e, a partir daí, são feitas combinações de um conjunto básico de componentes e sendo acrescentados alguns outros, construídos especialmente para aquele modelo. Alguns componentes são usados inclusive por marcas diferentes de carros (D'SOUZA, 1998, p.28).

Na área de Informática, há muito tempo projetos de *hardware* vêm sendo construídos com componentes unificados. Em relação ao *software*, esse modelo começou a ser adotado recentemente. Hoje, processadores de textos podem “conversar” com planilhas eletrônicas e *software* de gráficos podem acessar as informações em banco de dados. Cada um deles trabalha de forma independente, com sua própria interface usuária, mas todos oferecem uma forma de interagir com outros *softwares*. Na área de desenvolvimento de sistemas, muitas equipes estão desenvolvendo componentes específicos para suas áreas de aplicação e montando bibliotecas de componentes.

Um componente pode ser utilizado para construir códigos executáveis, códigos-fonte, projetos, especificações, testes, documentações, etc. O uso de um componente e sua interação posterior com outro só será possível mediante uma documentação explícita de suas interfaces. Considera-se um bom componente aquele que define abstrações, com interfaces bem definidas, o que torna possível a fácil substituição dos mais antigos por outros mais novos. (BOOCH, 2000, p.341). Um componente, portanto, sempre é parte de um sistema, raramente existindo sozinho. Ele colabora com outros dentro do contexto da arquitetura, ou da

tecnologia que pretende utilizá-lo. Além disso, poderá ser reutilizado em muitos sistemas, ou seja, um sistema em um nível de abstração poderá ser um componente em um nível de abstração mais alto (BOOCH, 2000, p.347).

4.1 COMPONENTES EM UML E JAVA

Do ponto de vista técnico, a *UML (Unified Modeling Language)* define componente como a parte física e substituível de um sistema ao qual se adapta e para o qual fornece a realização de um conjunto de interfaces. As interfaces, portanto, constituem a ponte entre os modelos lógico e físico. Por exemplo, se alguém especificar uma interface para uma classe em um modelo lógico, essa mesma interface será executada por algum componente físico. Ele representa o pacote físico de elementos lógicos, a exemplo de classes, interfaces e colaborações. Os componentes vivem no mundo material dos bits, logo, são importantes blocos de construção, para a modelagem de aspectos físicos de um sistema, tais como programas executáveis, bibliotecas, tabelas, arquivos e documentos (BOOCH, 2000, p.343-346). A Figura 12 ilustra um código em Java mostrando que uma interface é uma construção distinta de uma classe e uma classe pode implementar várias interfaces.

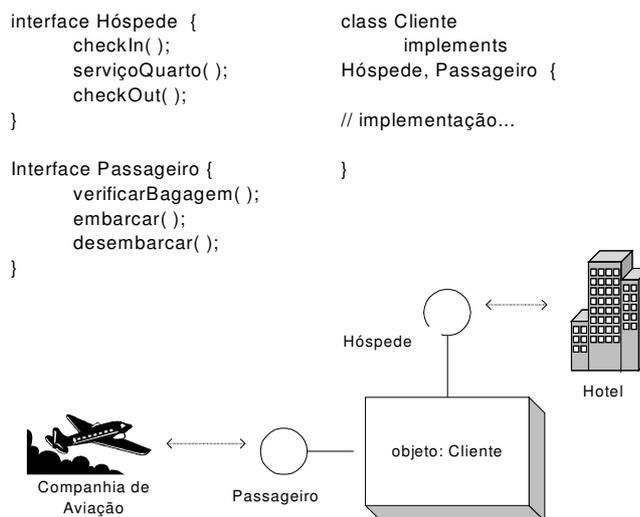


Figura 12 – Exemplo de classe implementando interface ¹⁸

18. Fonte:(D'SOUZA, 2001, p.10)

Em Java, os componentes são semelhantes às classes em muitos aspectos. Ambos podem realizar um conjunto de interfaces, podem participar de um relacionamento de dependência, generalização e associação, podem ser aninhados, admitem instâncias e podem ser participantes de interações. Entretanto, existem diferenças significativas entre componentes e classes (BOOCH, 2000, p.344):

- As classes representam abstrações lógicas; os componentes representam coisas físicas que vivem no mundo dos *bits*. Em resumo, os componentes podem viver em nós, as classes não.
- Os componentes representam o pacote físico de componentes lógicos e se encontram em um nível diferente de abstração. Em particular, um componente é uma implementação física de um conjunto de outros elementos lógicos, como as classes e as colaborações.
- As classes podem ter diretamente atributos e operações. Em geral, os componentes somente têm operações que são alcançadas por meio de suas interfaces.

Os componentes podem ser de três tipos (BOOCH, 2000, p. 347-348):

Componentes de implementação – Formam um sistema executável, como as bibliotecas dinâmicas e os executáveis. Abrangem modelos clássicos de objetos, como *Enterprise Java Beans*, até projetos envolvendo páginas dinâmicas da *Web*, tabelas de bancos de dados e programas executáveis.

- **Componentes do produto do trabalho** – São o resíduo do processo de desenvolvimento, formado por arquivos de código-fonte e arquivos de dados a partir dos quais se criam os componentes de implementação. São produtos de trabalho de desenvolvimento, utilizados para a criação do sistema executável.
- **Componentes de execução** – São criados como uma consequência de um sistema em execução; por exemplo, um objeto COM+, que é instanciado a partir de uma DLL.

4.2 DIAGRAMAS DE COMPONENTES

Os diagramas de componentes são essencialmente diagramas de classes que focalizam os componentes de um sistema e são empregados para a modelagem da visão estática de

implementação de um sistema, provendo uma visão física do modelo, por meio da representação do conjunto de componentes e seus relacionamentos (Figura 13). Esses diagramas também mostram o comportamento externo ou visível dos componentes, mediante suas interfaces. Apesar de as interfaces pertencerem de fato à visão lógica, elas podem ser representadas, tanto em diagramas de classes quanto em diagramas de componentes. Graficamente, o diagrama de componentes é uma coleção de vértices e arcos que modela itens físicos que residem em um nó, a exemplo de executáveis, bibliotecas, tabelas, arquivos e documentos (BOOCH, 2000, P. 387-388).

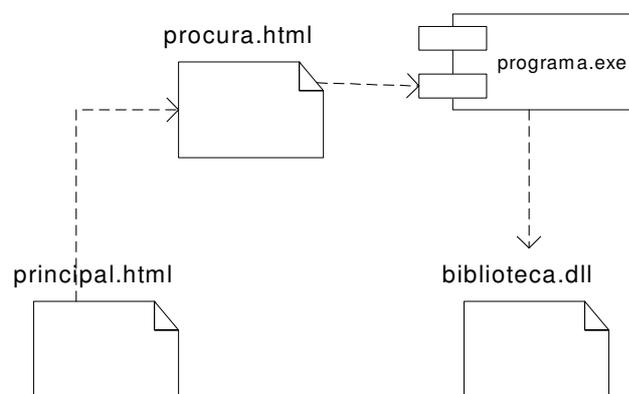


Figura 13 – Exemplo de diagrama de componentes ¹⁹

Os diagramas de componentes costumam conter (BOOCH, 2000, p. 389)

- Pacotes de componentes – Representam agrupamentos de componentes logicamente relacionados, ou pedaços principais de um sistema.. Permitem dividir o modelo físico do sistema.
- Componentes – Um componente representa um módulo de *software* (código-fonte, código binário, executável, DLL, etc.) com uma interface bem definida. A interface de um componente é representada por um ou alguns elementos conectados, que o componente provê. Componentes também podem ser usados para mostrar a dependência entre módulos de *software*.

19. Fonte: (BOOCH, 2000, p.388)

- Interfaces – Uma interface especifica as operações externamente visíveis de um componente ou de uma classe. Ela especifica só uma parte limitada do comportamento de uma classe ou de um componente. As interfaces pertencem à visão lógica do sistema, mas podem ser representadas também em diagramas de componentes.

- Dependências – Uma dependência é uma relação entre dois elementos do modelo; a mudança de um elemento afetará o outro.

Pode-se criar um ou mais diagramas de componentes para descrever pacotes de componentes e componentes em uma visão de alto nível, ou para descrever os conteúdos de cada pacote de componentes.

Uma especificação de pacote de componente permite ao programador exibir e modificar as propriedades de um pacote de componentes. Semelhantemente, uma especificação de componente e uma especificação de classe permitem exibir e modificar as propriedades de um componente e uma interface, respectivamente. As informações nessas especificações são apresentadas textualmente; algumas também podem ser exibidas dentro dos ícones que representam os pacotes de componentes.

Os diagramas de componentes podem ser usados, basicamente, para fazer a modelagem de (BOOCH, 2000, p.390):

- Código-fonte – Os códigos-fonte são armazenados em arquivos e podem ser representados como produto do trabalho de desenvolvimento;

- Versões executáveis – Servem para modelar e documentar os componentes que serão entregues;

- Banco de dados físicos – Esse esquema funciona como uma API (*Application Programming Interface*) para informações consistentes sobre o modo como as informações estão armazenadas no banco de dados;

- Sistemas adaptáveis – São utilizados em conjunto com diagramas de comportamento para representar sistemas mais dinâmicos.

4.3 DESENVOLVIMENTO BASEADO EM COMPONENTES

O desenvolvimento baseado em componentes é

uma abordagem para desenvolvimento de *softwares*, na qual todos os artefatos – do código executável até a especificação da interface, arquitetura e modelo de negócios; de aplicações completas e sistemas até partes pequenas – podem ser construídos, juntando-se e adaptando-se os componentes existentes em uma variedade de configurações[...]. [Nesse contexto, pode-se expandir o conceito de componente como] uma implementação coerente de um pacote de *software*, que pode ser desenvolvida e entregue independentemente; deve possuir interfaces especificadas de forma explícita, tanto para os serviços que oferece quanto para os serviços que espera dos outros componentes e pode ser composto de outros componentes. (D'SOUZA, 1998, p.385-387).

Assim, componente é um pacote de software que inclui (D'SOUZA, 1998, p.387):

- Uma lista das interfaces oferecidas - Contém a especificação dessas interfaces; as quais freqüentemente são importadas por outros pacotes.
- Uma lista de interfaces necessárias – É sempre constituída de pacotes separados, apenas com interfaces de contato.
- Especificação externa – É a especificação do comportamento externo oferecido e requerido; relata todas as interfaces em um modelo compartilhado.
- O código executável – Se construído de acordo com uma arquitetura satisfatória e consistente, pode ser acoplado ao código de outros componentes por intermédio de suas interfaces.
- Código de validação – Pode ajudar a decidir se a conexão proposta entre os componentes está OK.
- O projeto – Inclui todos os documentos e códigos-fonte desenvolvidos para satisfazer a especificação. Pode ser omitido para o cliente.

4.4 CONCLUSÃO

A primeira parte deste trabalho procurou sistematizar as informações dispersas em diversas fontes de forma a possibilitar a definição de um modelo de organização de material

didático, que será usado na definição do protótipo da ferramenta de autoria de cursos, mediante seleção e montagem de componentes de aprendizagem.

Segue-se a segunda parte do trabalho, que apresenta uma proposta de um modelo para organização de recursos didáticos associados a um domínio, em ambiente de ensino a distância e baseado em componentes, mapas conceituais e ontologia.

PARTE II

UM MODELO DE ORGANIZAÇÃO DE RECURSOS DIDÁTICOS ASSOCIADOS A UM DOMÍNIO, EM AMBIENTE DE ENSINO A DISTÂNCIA E BASEADO EM COMPONENTES

5 UM MODELO DE ORGANIZAÇÃO DE RECURSOS DIDÁTICOS

A presente proposta de modelo de organização de recursos didáticos associados a um domínio, em ambiente de ensino a distância, baseia-se em componentes de aprendizagem, mapas conceituais e ontologia. A escolha dessa base mostra-se vantajosa tanto para o professor/tutor quanto para o aprendiz, pois permite construir e utilizar o material didático de forma organizada e individualizada.

Conforme se viu, componentes são partes pequenas e reusáveis que devem funcionar de forma independente, mas podem ser combinadas para formar um componente maior. Mapas conceituais podem representar eficientemente conhecimentos e orientar a ação didática do professor. Ontologia permite a criação de um vocabulário comum a ser usado na definição de objetos e outras entidades, e na relação e troca de mensagens entre eles. Portanto pode-se usar eficientemente a ontologia na definição do domínio. Esses três conceitos, usados em conjunto, são fundamentais na elaboração do ambiente de aprendizagem proposto, onde o conjunto de informações do material a ser elaborado será dividido em partes, de forma organizada e coerente.

5.1 REPRESENTAÇÃO DOS RECURSOS DIDÁTICOS

Para a definição do tamanho do recurso didático utilizou-se o conceito de componente, pois, conforme visto anteriormente, a divisão do material instrucional em partes pequenas, estimula o reuso e proporciona a um sistema tutor ou a um professor organizar o material de forma individualizada, levando em consideração o nível de conhecimento que o aprendiz já possui sobre o assunto, além de facilitar seu desenvolvimento e sua armazenagem, em banco

de dados persistentes, para posterior recuperação e reuso, proporcionando a organização rápida e eficiente de uma aula ou de um novo curso.

Assim foram definidos três tamanhos de recursos didáticos, denominados de Componentes de Aprendizagem: Material Didático, Aula e Curso. Dessa forma, um autor, ao definir seu curso, poderá selecionar componentes em uma biblioteca de componentes de aprendizagem que poderão ser combinados para a preparação de aulas e cursos direcionados para grupos de alunos que possuam o mesmo nível de conhecimento. O material didático é o componente de aprendizagem de menor granularidade. Cada Material Didático trata de um determinado assunto podendo ser dos tipos conceito, explanação, exemplo, questão, exercício e etc. O componente de aprendizagem denominado de Aula de tamanho médio, representa um conjunto de materiais Didáticos de um mesmo assunto. Ao conjunto de aulas que reúnem assuntos pertencentes ao mesmo domínio do conhecimento, foi definido o componente Curso que é o de maior granularidade no modelo proposto. A figura 14 apresenta a notação que será utilizada na representação dos componentes de aprendizagem Material Didático, Aula e Curso na definição do modelo.



Figura 14a – Notação utilizada na representação de um componente do tipo material didático



Figura 14b – Notação utilizada na representação de um componente do tipo Aula

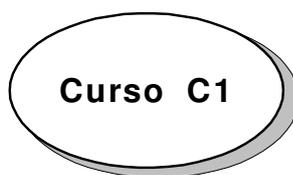


Figura 14c – Notação utilizada na representação de um componente do tipo Curso

Na definição da organização e da representação dos componentes de aprendizagem as idéias preconizadas pelos criadores dos mapas conceituais foram utilizadas, pois, conforme demonstrado no capítulo 3, mapa conceitual é uma ferramenta utilizada para a organização e a representação do conhecimento e descrevem visualmente o relacionamento entre idéias, dentro de um domínio do conhecimento. Assim, a figura 15 define o modelo de representação e organização dos componentes Material, Aula e Curso e o relacionamento entre eles.

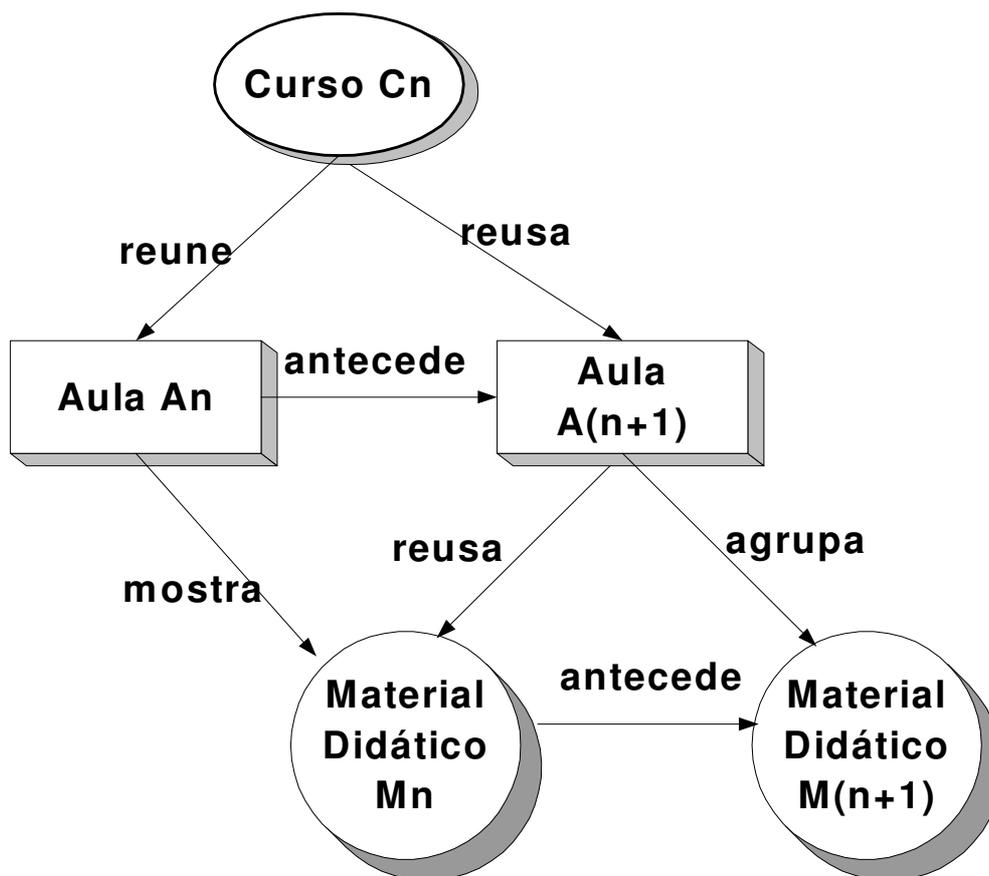


Figura 15 – Modelo de representação e organização dos Componentes de Aprendizagem

Conforme a figura acima, um curso reúne várias aulas de assuntos diversos, pertencentes ao mesmo domínio de conhecimento, podendo também reutilizar aulas de outros cursos. As aulas estão organizadas segundo uma ordem de apresentação, possibilitando ao tutor controlar a disponibilidade do material instrucional para o aluno ou grupo de alunos. Elas mostram materiais didáticos de diversos tipos (conceito, explanação, exemplo e etc.). Os materiais são dispostos conforme a conveniência do tutor e também podem ser reutilizados em outras aulas que tratem do mesmo assunto.

Para o material ser reutilizado e compartilhado ele precisa estar definido dentro de um padrão, de forma a permitir uma recuperação rápida e eficiente. Assim foram utilizados os conceitos de ontologia no contexto de sistemas baseados em conhecimento, onde ontologia é definida como um vocabulário comum, a ser usado na definição de objetos e outras entidades, e na relação e troca de mensagens entre eles, permitindo o compartilhamento e a reutilização desse vocabulário comum e uma classificação mais precisa de aplicações de domínio.

O componente Material Didático é definido por uma ontologia de aplicação (Figura 16a), que define como vocabulário comum a todos os materiais, um nome para designar a essência do material; a descrição, que determina a que ele se destina; o tipo que pode ser conceito, descrição, exemplo, exercício etc. e o assunto sobre o qual o material trata.

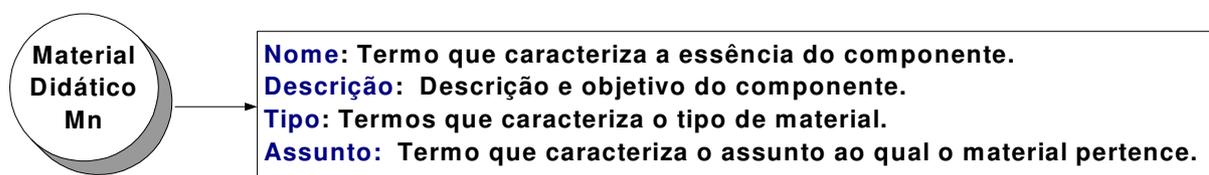


Figura16a – Modelo de ontologia de aplicação usado na definição do tipo de material didático

O nível da aula é representado pela ontologia de tarefas (Figura 16b). Assim todas as aulas possuem um nome que designa o assunto ao qual a aula pertence; uma descrição, que determina seu objetivo e a definição do domínio do conhecimento no qual a aula será utilizada.

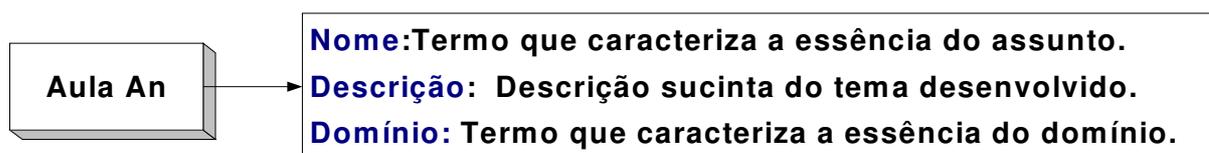


Figura16b – Modelo de ontologia de tarefas usado na definição do assunto das aulas

O nível do curso é representado pela ontologia de domínio (Figura 16c) que determina o domínio do conhecimento ao qual o curso pertence, uma descrição e objetivos do curso e um sinônimo do termo definido no domínio do conhecimento.

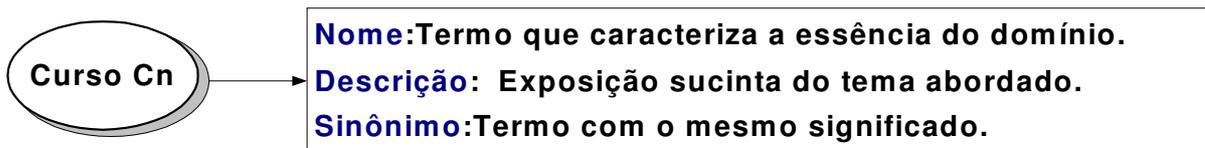


Figura 16c – Modelo de ontologia de domínio usado na definição do domínio dos cursos

As Figuras 17a, b e c mostram respectivamente uma instanciação da ontologia para curso, aula e material didático.

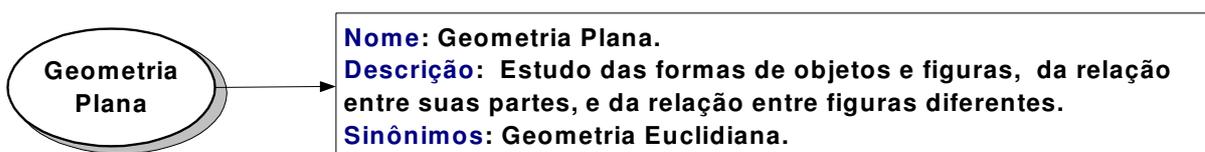


Figura 17a – Instanciação da ontologia de domínio para curso

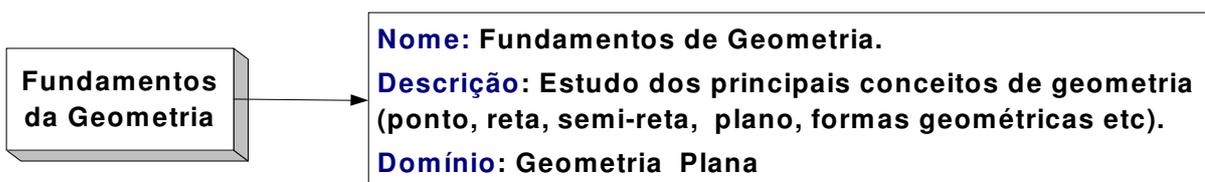


Figura 17b – Instanciação da ontologia de tarefas para aulas

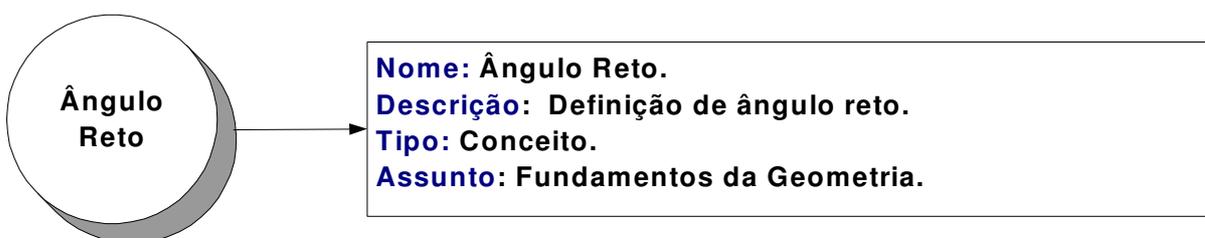


Figura 17c – Instanciação da ontologia da aplicação para material didático

A Figura 18 apresenta o modelo proposto para organização de um curso em três planos. No primeiro são definidos os materiais didáticos de diversos tipos que farão parte de aulas de vários assuntos de um mesmo domínio do conhecimento, pertencentes ao segundo nível do modelo. As aulas serão utilizadas na montagem de cursos sobre determinado domínio do conhecimento. Durante a montagem da aula, poderão ser reaproveitados os materiais

didáticos de diversos autores, que já existem na biblioteca de componentes e que poderão ser pesquisados por meio dos termos comuns utilizados na definição de cada material.

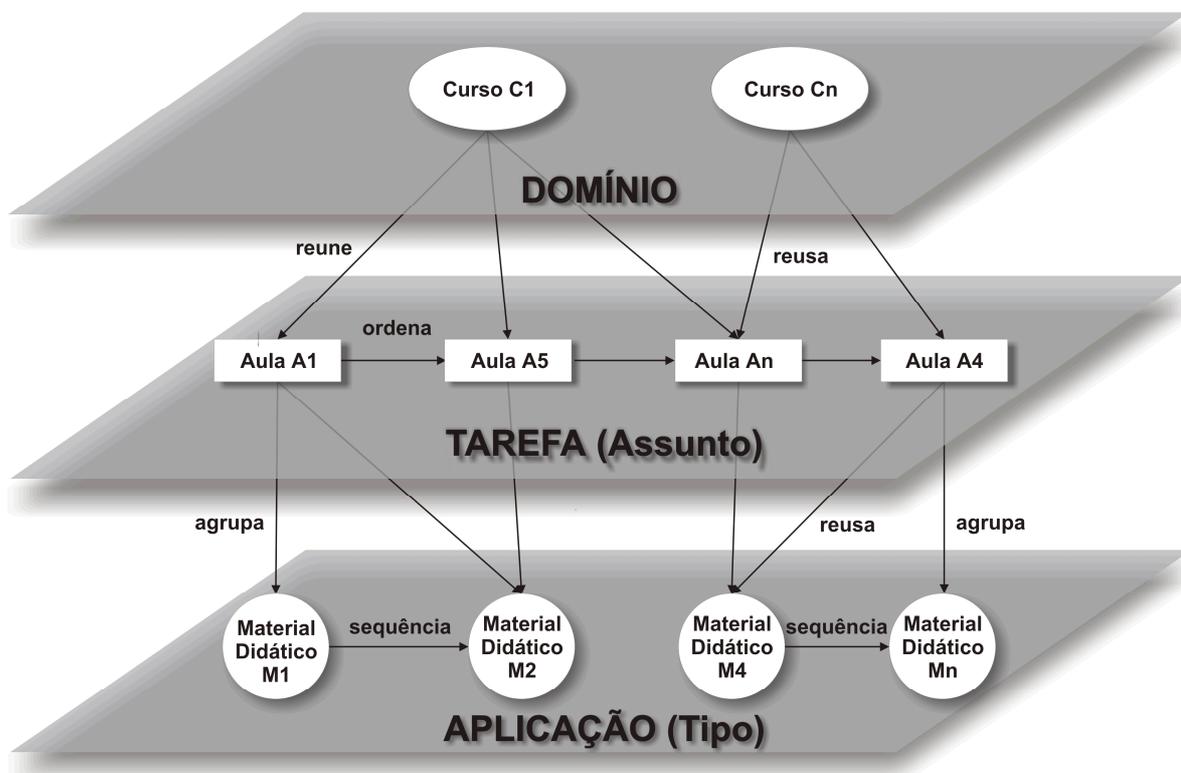


Figura 18 – Modelo de organização de curso

5.2 UM EXEMPLO DO MODELO PROPOSTO

Para tornar mais claro o modelo proposto, e demonstrar o uso dos conceitos de mapas conceituais, ontologia e componentes, apresenta-se, a seguir, um exemplo de curso. A Figura 19a mostra a estruturação de um curso de Geometria e o reuso de componentes no nível de material didático, dentro de um mesmo curso. A Figura 19b ilustra um curso de Geometria Descritiva e o reuso de componentes nos níveis de aula e material didático, em cursos diferentes. A informação do assunto nas ontologias de tarefas e aplicação serve como indicação para agilizar a procura e facilitar o reuso, mas não restringe o uso, ou seja, um material didático (MD 3: Plano) criado para a aula de Fundamentos da Geometria pode ser reutilizado em uma aula sobre Planos e uma aula criada para um curso de Geometria Plana

(Aula 1: Fundamentos da Geometria) pode ser reutilizada em um curso de Geometria Descritiva.

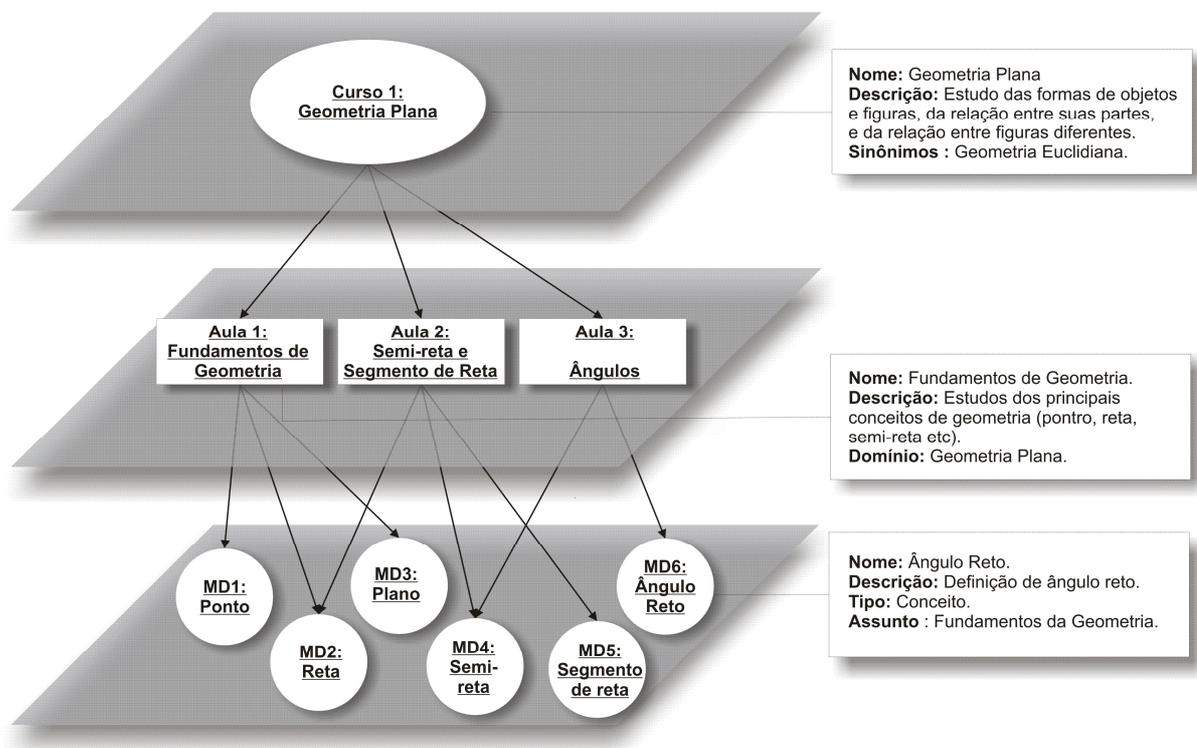


Figura 19 a – Exemplo de reuso do componente material didático no mesmo curso

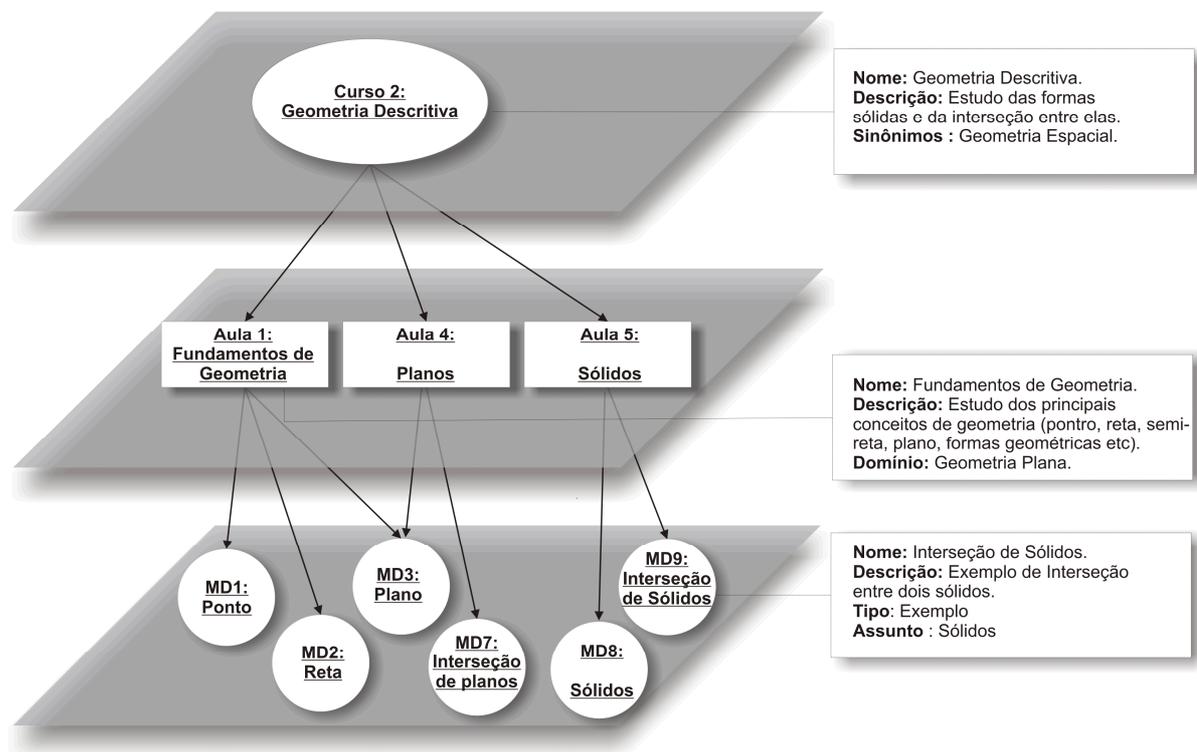


Figura 19b – Exemplo de reuso de componentes aula e material didático em cursos diferentes

5.3 CONCLUSÃO

Essa proposta de modelo para representação de material didático utiliza, de forma associada, os conceitos de componentes, ontologia e mapas conceituais, para proporcionar um ambiente com facilidades de criação, organização, identificação e reuso de material didático.

O próximo capítulo versará sobre a modelagem de um sistema baseado no modelo que se acabou de definir. Serão apresentados a linguagem, o método, os requisitos e especificações do Sistema gerenciador de Componentes (SGC), sua arquitetura e o modelo de seus componentes.

6 MODELAGEM DO SGC

6.1 LINGUAGEM E MÉTODO

Para a modelagem do SGC – Sistema Gerenciador de Componentes, a *UML (Unified Modeling Language)* e o método *Catalysis* são utilizados.

“*UML* é uma linguagem padrão, para a elaboração da estrutura de um projeto de software, podendo ser empregada para a visualização, a especificação, a construção e a documentação de artefatos que façam uso de sistemas complexos de software” (BOOCH, 2000, p.13).

Catalysis é uma metodologia baseada em padrões para o desenvolvimento sistemático de sistemas baseados em objetos e componentes. Foi desenvolvida, originalmente, por Desmond D’Souza e Alan Wills e usa notações baseadas no padrão *UML*. *Catalysis* oferece um método completo que possibilita especificar, documentar, implementar e testar um sistema desde a especificação do negócio até o código-fonte (D’SOUZA, 1998).

A metodologia *Catalysis* encontra-se detalhada no Anexo I.

6.2 REQUISITOS

Segundo D’Souza (1998, p.518), as atividades de levantamento dos requisitos são necessárias para a contextualização e o entendimento do problema e devem refletir a visão do cliente, servindo de ligação entre o *software* e o mundo real. A fim de permitir uma clara definição do SGC, serão desenvolvidos como atividades de levantamento de requisitos a descrição do negócio, o esquema conceitual, o glossário, os requisitos funcionais e os casos de uso do SGC.

6.2.1 DESCRIÇÃO DO NEGÓCIO

A descrição do negócio é uma atividade do sistema em que o cliente descreve a sua visão do problema e os objetivos esperados da sua solução.

O objetivo do SGC é oferecer aos seus usuários um ambiente de preparação e uso de material didático para cursos de EAD. Esse material será preparado por meio da criação de componentes de aprendizagem, que poderão ser de três tamanhos: pequenos, médios e grandes. A organização dos componentes atenderá a uma seqüência lógica, baseada nos conceitos definidos em mapas conceituais, onde conceitos mais amplos servem de ancoragem para conceitos mais específicos. A documentação e posterior recuperação dos componentes será feita mediante a ontologia definida no capítulo anterior.

Ao menor componente de aprendizagem dar-se-á o nome de material didático. Cada material didático tratará de um assunto e poderá ser dos seguintes tipos: conceito, explanação e exemplo. Ele poderá ser representado por informações encontradas na Internet, documentos, apresentações, planilhas, páginas *Web*, etc., ou seja, poderá ser preparado em uma ferramenta com a qual o autor já possui familiaridade sendo, depois, *linkada* ao controle do material didático que se deseja criar. Um material didático só poderá ser alterado por seu autor, porém poderá ser reutilizado por qualquer professor.

Ao conjunto de materiais didáticos dar-se-á o nome de aula ou componente médio. Uma aula aborda um determinado assunto pertencente a um domínio do conhecimento. Portanto, todos os materiais didáticos pertencentes àquela aula tratarão do mesmo assunto. Nesse contexto, uma aula representa uma unidade lógica de conhecimento sobre determinado assunto. Assim como os materiais didáticos, uma aula só poderá ser alterada por seu autor, mas qualquer professor poderá utilizá-la.

Ao conjunto de aulas dá-se o nome de curso ou componente grande. Um curso é preparado com o objetivo de atender a um grupo de alunos agregados por nível de conhecimento — básico, intermediário ou avançado, a depender do conhecimento que possui sobre o domínio a ser abordado. Um curso está relacionado a um determinado domínio de conhecimento, que, por sua vez, agrega vários assuntos correlatos. Domínios e assuntos serão definidos por especialistas, de acordo com o modelo de ontologia apresentado no capítulo anterior. O administrador do sistema é responsável por manter atualizadas as ontologias de domínio e assunto no sistema, e o cadastro de usuários do sistema. A ligação de um assunto a

um determinado domínio do conhecimento serve para orientar o professor na pesquisa do material didático, facilitando o reuso do material pelo professor que está montando o curso.

Autores e professores poderão consultar o material existente, conforme as seguintes premissas:

- Um curso poderá ser consultado por domínio do conhecimento;
- Uma aula poderá ser consultada por assunto;
- Um material didático poderá ser consultado por assunto.

Para controlar o andamento do curso, cada componente terá controle de acessibilidade, ou seja, um dispositivo que indicará se determinado grupo de alunos poderá ou não ter acesso ao material.

Acessarão o sistema quatro tipos de usuários:

- **O administrador** – Responsável por cadastrar usuários e manter as ontologias de domínio e assunto, definidas pelos autores;
- **O autor** – Especialista em determinado domínio do conhecimento, responsável por criar componentes de aprendizagem do tipo material didático;
- **O professor** – Responsável por montar aulas e cursos. Um autor pode desempenhar o papel de professor;
- **O aluno/grupo de alunos** – Usuário que acessará o sistema para participar de cursos e assistir a aulas.

6.2.2 ESQUEMA CONCEITUAL

O esquema conceitual representa os aspectos importantes do sistema, capturados a partir da descrição do negócio, mediante levantamento dos objetos que participam do sistema, análise das funções de cada objeto e identificação das interações entre os objetos. A Figura 20 ilustra o esquema conceitual do SGC.

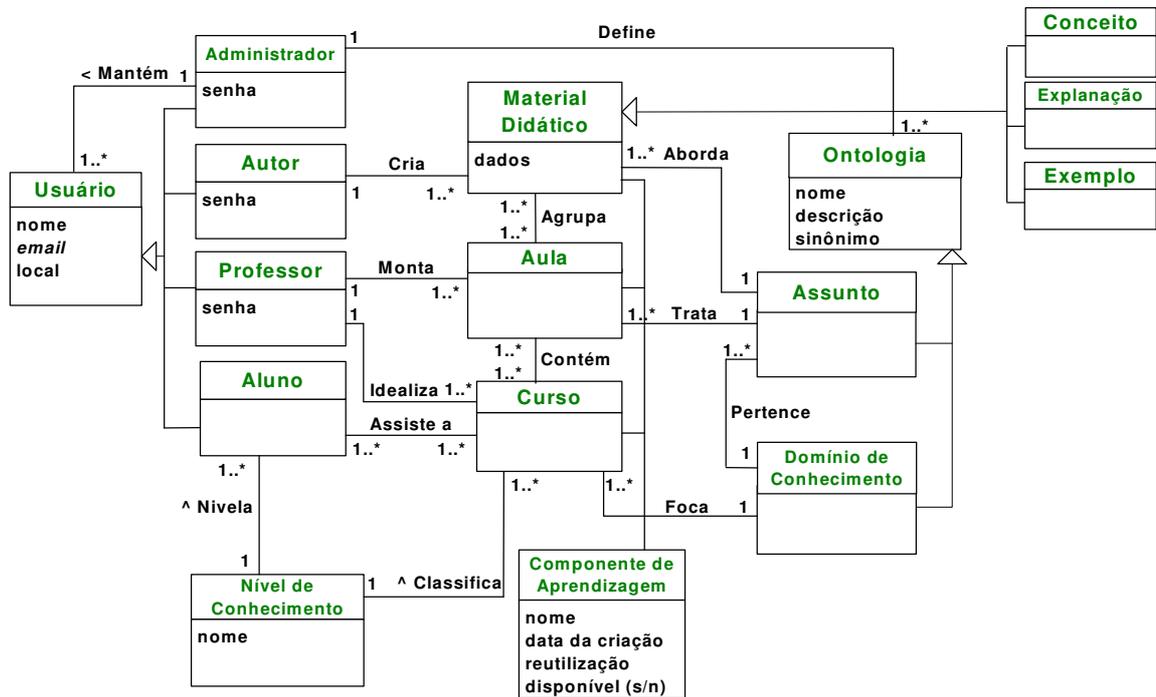


Figura 20 – Esquema conceitual do SGC

6.2.3 GLOSSÁRIO DE TERMOS

Um glossário é um documento simples, que lista e define os termos que são importantes para o desenvolvimento da aplicação, de forma a unificar o entendimento (LARMAN, 2000, p.139). A tabela 3 apresenta os termos relevantes e suas definições, utilizados na modelagem do SGC.

TERMO	DEFINIÇÃO
Administrador	Usuário responsável pelas tarefas operacionais do sistema, como manutenção do cadastro de usuários e manutenção das tabelas de domínio do conhecimento e assunto.
Aluno	Usuário com direito a participar de cursos.
Assunto	Objeto de que trata um componente do tipo aula ou material didático.

TERMO	DEFINIÇÃO
Aula	Componente médio, identifica uma parte de um curso relacionado a um domínio do conhecimento. É composta por diversos materiais didáticos de um mesmo assunto.
Autor	Participante responsável pela criação dos materiais didáticos. Um autor tem direito a criar novos materiais, porém só pode alterar um componente de sua autoria. Tem acesso a todas as funções do professor.
Biblioteca de Componentes	Repositório de componentes ou materiais didáticos.
Componentes de Aprendizagem	Utilizados para disponibilizar recursos didáticos para as aulas. Podem ser de três tamanhos: pequeno ou material didático, médio ou aula, e grande ou curso. São criados por autores e só podem ser excluídos por quem os criou.
Curso	É uma unidade completa. Formado por um conjunto de aulas relacionadas ao mesmo domínio do conhecimento, é preparado considerando o nível de conhecimento do aluno/grupo de alunos.
Domínio do Conhecimento	Conjunto de informações comuns a respeito de um assunto complexo.
Grupo de Alunos	Conjunto de alunos agrupados pelo mesmo nível de conhecimento sobre determinado domínio.
Material Didático	Menor componente de aprendizagem, pode ser do tipo conceito, explanação e exemplo.
Nível de Conhecimento	Mede o progresso de um aprendiz ou de um grupo de aprendizes. Classifica o conhecimento de um aluno em básico, intermediário e avançado.
Professor	Participante responsável pela montagem de aulas e cursos, mediante seleção de materiais didáticos criados pelos autores. Pode alterar ou excluir aulas e cursos de sua autoria.
Usuários	Pessoas que acessam o sistema. Estão classificados em administrador, aluno, professor e autor.

Tabela 3 – Glossário do SGC

6.2.4 REQUISITOS FUNCIONAIS

Os requisitos funcionais do sistema definem as necessidades ou desejos para um produto (LARMAN, 2000, 60). A Tabela 5 descreve as principais funcionalidades do SGC.

REF.	FUNCIONALIDADE	CATEGORIA
R01	Manter cadastro de autores que podem criar componentes didáticos.	Evidente.
R02	Manter cadastro de professores que podem montar aulas e cursos.	Evidente.
R03	Manter cadastro de alunos que podem consultar cursos.	Evidente.
R04	Manter biblioteca de componentes didáticos.	Evidente.
R05	Selecionar material didático por assunto, autor e tipo de recurso didático.	Oculto.
R06	Exibir material didático por assunto, autor e tipo de recurso didático.	Evidente.
R07	Manter biblioteca de aulas.	Evidente.
R08	Selecionar aulas existentes por assunto e/ou autor.	Oculto.
R09	Exibir aulas selecionadas por assunto e/ou autor.	Evidente.
R10	Identificar materiais didáticos que compõem uma aula.	Evidente.
R11	Montar seqüência de materiais didáticos pertencentes a uma aula.	Evidente.
R12	Manter curso para determinado domínio e nível de conhecimento.	Evidente.
R13	Identificar aulas que compõem um curso.	Evidente.
R14	Montar seqüência de aulas que compõem um curso.	Evidente.
R15	Identificar cursos existentes por domínio, nível de conhecimento e autor.	Oculto.
R16	Exibir curso solicitado.	Evidente.
R17	Efetuar <i>login</i> do usuário por meio de identificação e senha.	Evidente.
R18	Prover um mecanismo de armazenamento consistente.	Oculto.

R19	Manter ontologias de domínio e assunto.	Evidente.
-----	---	-----------

Tabela 4 – Funcionalidades do SGC

6.2.5 CASOS DE USO

Um caso de uso representa a interação entre o sistema e um ator para a realização de uma ação. Um ator pode ser uma pessoa, um sistema existente, um recurso, etc. que interage diretamente com o sistema (D’SOUZA, 2000, p.165).

Neste trabalho, os casos de uso se apresentam em três níveis de refinamento. Inicialmente se oferece uma visão geral deles e dos atores do SGC (Figura 21), representando o nível mais alto de compreensão do sistema. A seguir os casos de uso são apresentados mediante uma descrição sucinta das ações e da representação gráfica correspondente. As seções 6.2.5.1 a 6.2.5.6 fornecem a descrição detalhada de cada um deles.

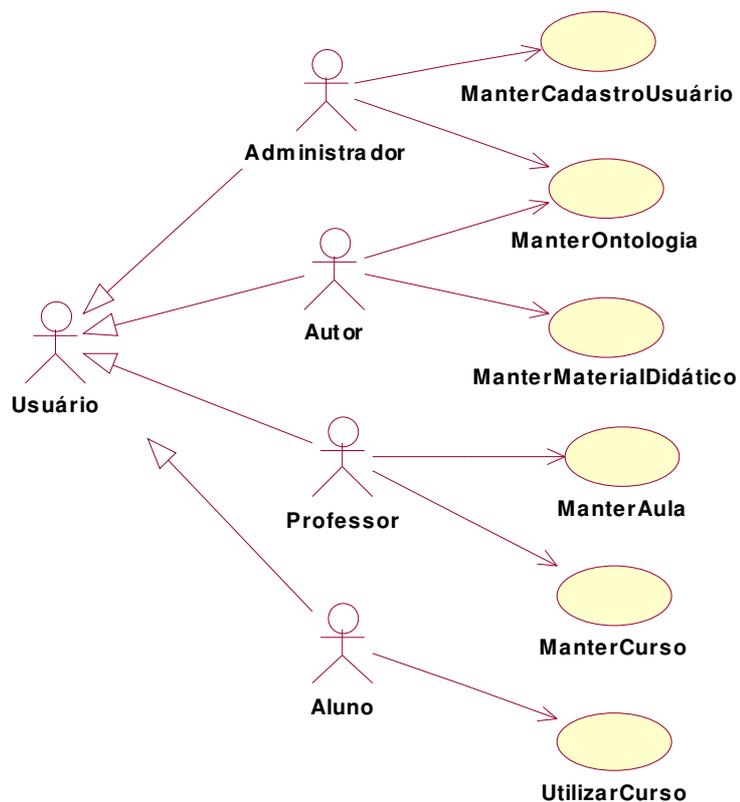


Figura 21 – Visão geral dos casos de uso

CASO DE USO 1 – MANTER CADASTRO DE USUÁRIOS

Atores:	Usuários (autor, professor, aluno e administrador).
Descrição:	O administrador é responsável por manter o cadastro de usuários. O autor tem permissão para criar, alterar e excluir materiais didáticos. O professor tem permissão para montar aulas e cursos, e o aluno tem permissão para consultar cursos (Figura 22).

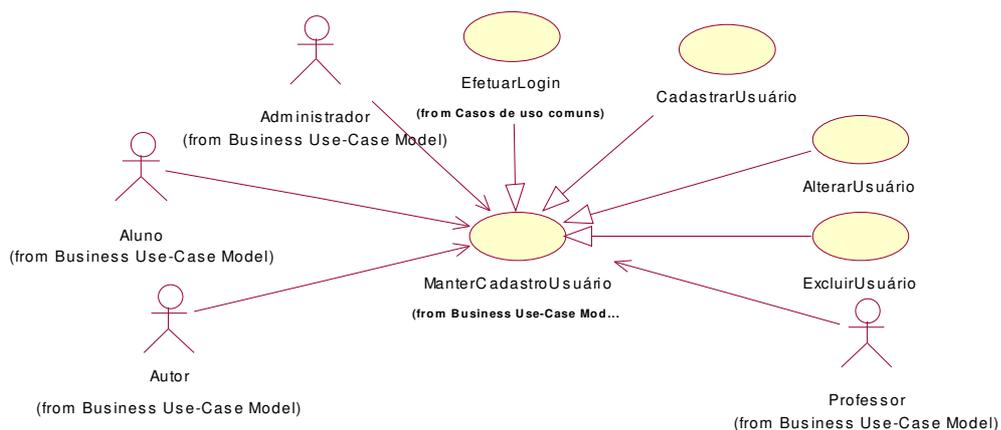


Figura 22 – Caso de uso Manter Cadastro de Usuários

CASO DE USO 2 – MANTER MATERIAL DIDÁTICO

Ator:	Autor.
Descrição:	Manter biblioteca de materiais didáticos classificados por assunto (Figura 23).

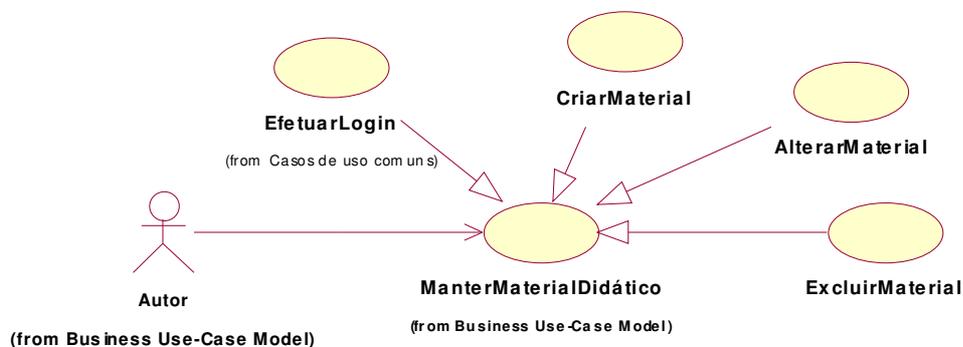


Figura 23 – Caso de uso Manter Material Didático

CASO DE USO 3 – MANTER AULA

Ator:	Professor.
Descrição:	Manter cadastro de aula para determinado assunto. Especificar os materiais didáticos que irão compor a aula. Montar seqüência dos materiais que compõem a aula (Figura 24).

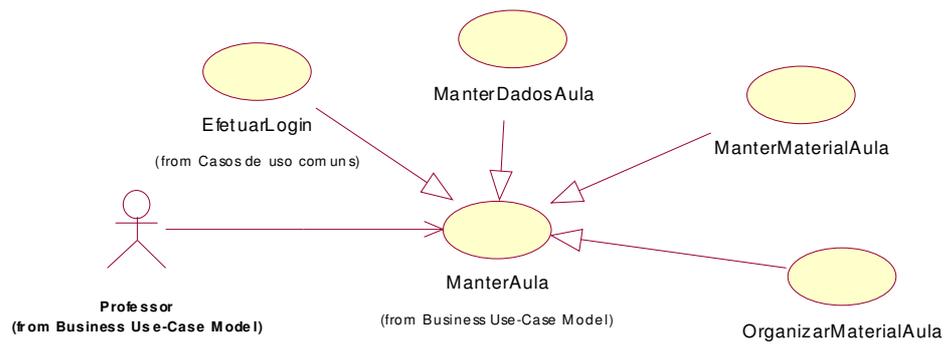


Figura 24 – Caso de uso Manter Aula

CASO DE USO 4 – MANTER CURSO

Ator:	Professor.
Descrição:	Manter curso para determinado domínio e nível de conhecimento, selecionar aulas que farão parte do curso e organizar a seqüência de aulas (Figura 25).

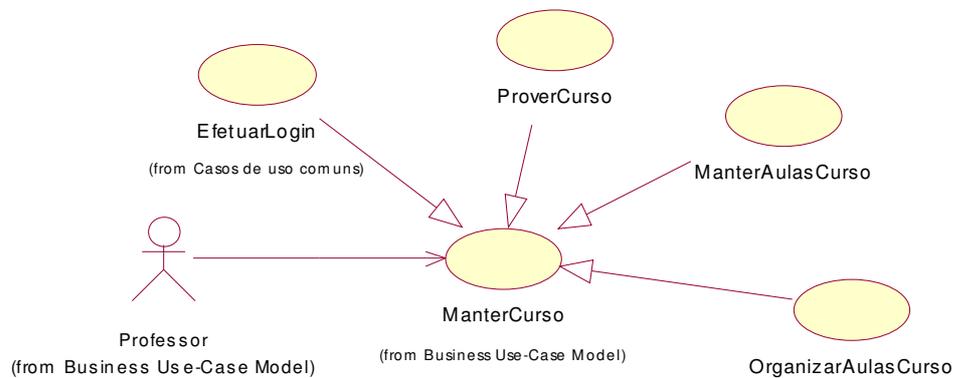


Figura 25 – Caso de uso Manter Curso

CASO DE USO 5 – UTILIZAR CURSO

Ator:	Aluno.
Descrição:	Apresentar cursos existentes sobre determinado domínio.(Figura 26).

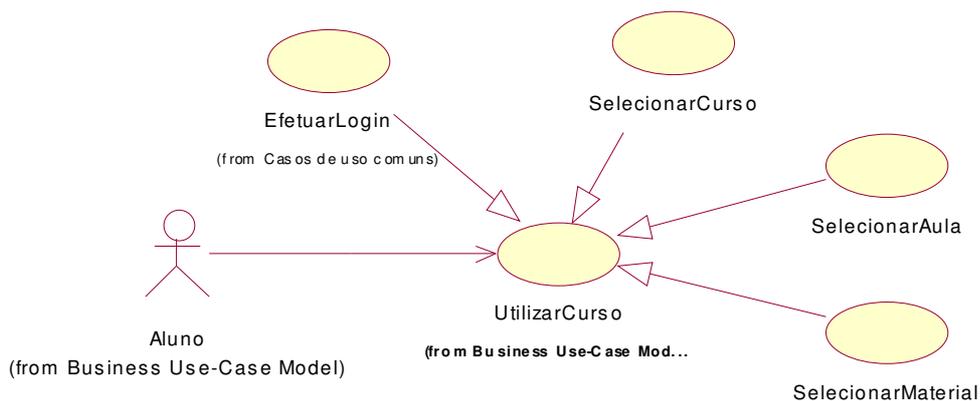


Figura 26 – Caso de uso Utilizar Curso

CASO DE USO 6 – MANTER ONTOLOGIA

Atores:	Administrador e Autor.
Descrição:	Manter ontologias de domínio do conhecimento e assunto (Figura 27).

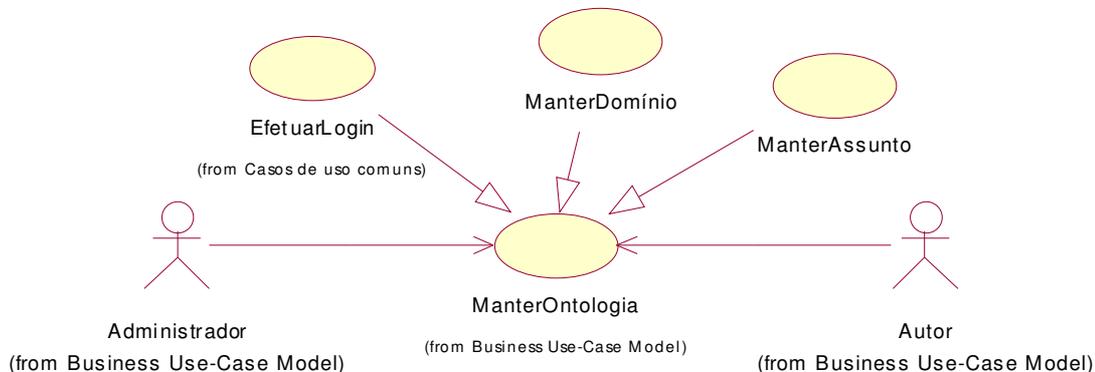


Figura 27 – Caso de uso Manter Ontologia

6.2.6 CASOS DE USO – REFINAMENTO

Uma das características da metodologia *Catalysis* é o suporte a vários níveis de descrição do mesmo fenômeno, baseados no conceito de refinamento. Para uma abordagem

dirigida a casos de uso, oferece uma separação clara entre a especificação do comportamento externo do sistema e a implementação (D'SOUZA, 2000). As seções 5.2.6.1 a 5.2.6.6 expõem uma abordagem mais detalhada dos casos de uso do sistema, apresentados anteriormente.

CASO DE USO 1 – MANTER CADASTRO DE USUÁRIOS

- Atores:** Usuários (autor, professor, aluno e administrador).
- Propósito:** Manter o cadastro de usuários do sistema.
- Visão Geral:** O administrador do sistema recebe solicitação de um usuário para incluí-lo no sistema, excluí-lo ou alterar seus dados cadastrais. Caso essa pessoa se cadastre como: a) professor, terá permissão para utilizar os materiais didáticos, sem alterá-los, e para montar aulas e cursos; b) autor, terá permissão para criar componentes didáticos, além de poder assumir as funções de professor; c) aluno, terá direito a assistir aos cursos.
- Funcionalidades:** R01 – Manter cadastro de autores que podem criar componentes didáticos.
R02 – Manter cadastro de professores que podem criar aulas e cursos.
R03 – Manter cadastro de alunos que podem consultar cursos.
R17 – Efetuar *login* do usuário por meio de identificação e senha.
R18 – Prover um mecanismo de armazenamento consistente.

CASO DE USO 2 – MANTER MATERIAL DIDÁTICO

- Seção:** **Principal.**
- Atores:** Autor.
- Propósito:** Fornecer ao autor interface amigável para manutenção de materiais didáticos.
- Visão Geral:** Manter biblioteca de materiais didáticos. Criar, alterar e excluir materiais didáticos.
- Funcionalidades:** R04 – Manter biblioteca de materiais didáticos.
R05 – Selecionar materiais didáticos por assunto, autor e tipo de recurso didático.

R06 – Exibir materiais didáticos por assunto, autor e tipo de recurso didático.

R18 – Prover um mecanismo de armazenamento consistente.

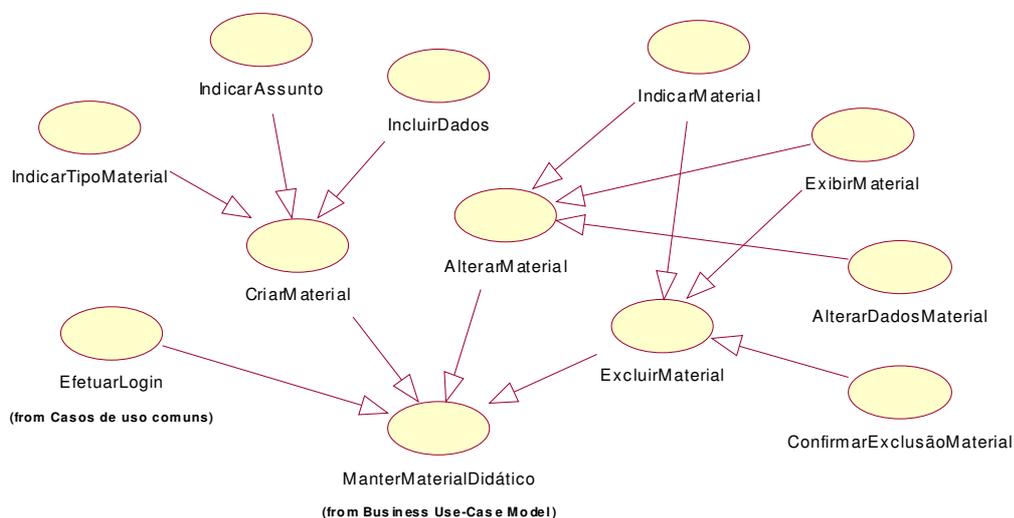


Figura 28 – Caso de uso Manter Material Didático – Refinamento

Seqüência típica de eventos	Manter Material Didático
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. Um usuário acessa o sistema para fazer manutenção de materiais didáticos. O usuário informa sua identificação e senha.	2. O sistema verifica se a identificação e a senha estão corretas.
3. O autor informa se pretende criar, alterar ou excluir um material didático: a) para criar, ver seção Criar Material Didático; b) para alterar, ver seção Alterar Material Didático; c) para excluir, ver seção Excluir Material Didático.	4. O sistema registra e informa o sucesso da operação.

Seqüência alternativa 2 – O usuário não tem permissão para criar materiais didáticos. O sistema informa ao usuário e finaliza a operação.

Seção	Criar Material Didático
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>

	1. O sistema solicita informações sobre material didático.
2. O autor informa nome, assunto, domínio, tipo de recurso, tipo de material e endereço de referência (servidor, web, hard disk(HD) e outros).	

Seção	Alterar Material Didático
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O autor informa material didático a ser alterado.	2. O sistema exibe os dados do material didático a ser alterado.
3. O autor altera os dados conforme sua necessidade.	

Seqüência alternativa 2 – O sistema não encontra o material solicitado, informa ao usuário e finaliza a operação.

Seção	Excluir Material Didático
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O autor informa material didático a ser excluído.	2. O sistema exibe os dados e solicita confirmação.
3. O autor confirma a exclusão.	

Seqüência alternativa 1 – O sistema não encontra o material solicitado, informa ao usuário e finaliza a operação.

Seqüência alternativa 3 – O autor não confirma a exclusão. O sistema finaliza a operação.

Seqüência alternativa 3 – O sistema informa que o material a ser excluído pertence a uma aula e não efetua a operação.

CASO DE USO 3 – MANTER AULAS

Seção: Principal.

Ator: Professor.

Propósito: Fornecer ao professor interface amigável para manutenção de aulas

Visão Geral: Manter biblioteca de aulas para determinado assunto. Especificar os materiais didáticos que irão compor a aula. Montar seqüência dos materiais que compõem a aula.

Funcionalidades: R07 – Manter biblioteca de aulas.

R08 – Selecionar aulas existentes por assunto e/ou autor.

R09 – Exibir aulas selecionadas por assunto e ou autor.

R10 – Identificar materiais didáticos que compõem uma aula.

R11 – Montar seqüência de materiais didáticos pertencentes a uma aula.

R17 – Efetuar *login* do usuário mediante identificação e senha.

R18 – Prover um mecanismo de armazenamento consistente.

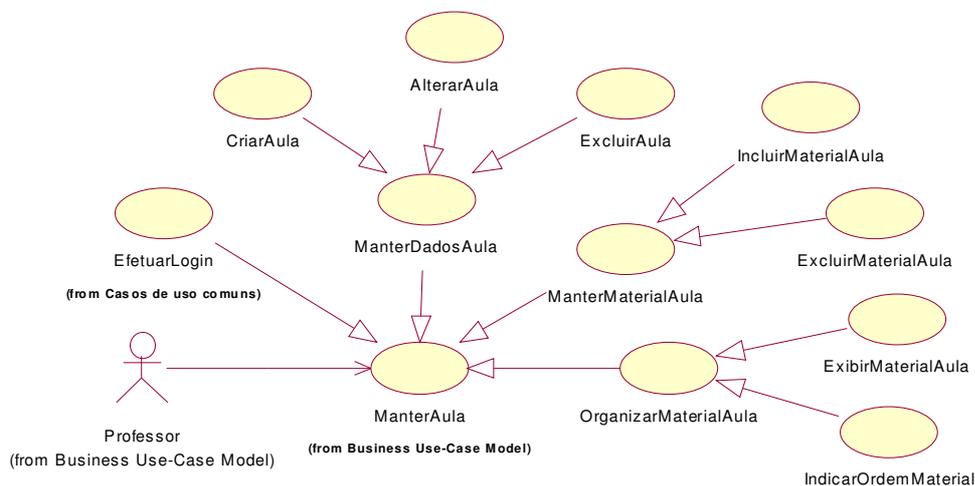


Figura 29 – Caso de uso Manter Aula – Refinamento

Seqüência típica de eventos	Manter Aulas
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. Um usuário acessa o sistema para fazer manutenção de aulas. O usuário informa sua identificação e senha.	2. O sistema verifica se a identificação e a senha estão corretas.
3. O professor informa qual a operação pretendida: a) para manter dados cadastrais da aula, ver seção Manter Dados Aula; b) para especificar os materiais didáticos que compõem uma aula, ver seção Manter Material Aula; c) para organizar seqüencialmente o material, ver seção Organizar Material Aula.	4. O sistema registra e informa o sucesso da operação.

Seqüência alternativa 2 – O usuário não tem permissão para manter aulas. O sistema informa ao usuário e finaliza a operação.

Seção	Manter Dados Aula
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O professor informa se pretende criar, alterar ou excluir uma aula: a) para criar, ver seção Criar Aula; b) para alterar, ver seção Alterar Aula; c) Para excluir, ver seção Excluir Aula.	

Seção	Criar Aula
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
	1. O sistema solicita informações da aula.
2. O autor informa o assunto e o domínio da aula a ser criada.	3. O sistema cadastra a aula e informa o sucesso da operação.

Seção	Alterar Aula
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O autor informa a aula a ser alterada.	2. O sistema exibe os dados.
3. O autor altera os dados conforme sua necessidade.	

Seqüência alternativa 2 – O sistema não encontra a aula solicitada, informa ao usuário e finaliza a operação.

Seção	Excluir Aula
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O autor informa a aula a ser excluída.	2. O sistema exibe os dados e solicita confirmação.
3. O autor confirma a exclusão	

Seqüência alternativa 2 – O sistema não encontra a aula solicitada, informa ao usuário e finaliza a operação.

Seqüência alternativa 3 – O usuário não confirma a exclusão. O sistema finaliza a operação.

Seção	Manter Material Aula
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O professor solicita ao sistema a aula que pretende trabalhar.	2. O sistema recupera e exibe os dados solicitados.
3. O professor informa se pretende incluir ou excluir componentes didáticos: a) para incluir novos materiais didáticos em aula, ver seção Incluir Material Aula; b) para excluir materiais já existentes, ver seção Excluir Material Aula.	

Seqüência alternativa 2 – O sistema não encontra a aula solicitada, informa ao usuário e finaliza a operação.

Seção	Incluir Material Aula
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O professor informa o tipo de recurso didático do material que pretende inserir.	2. O sistema recupera e exibe os materiais didáticos solicitados.
3. O professor seleciona o material escolhido.	4. O sistema associa o material à aula e informa o sucesso da operação.

Seqüência alternativa 2 – O sistema não encontra o material do tipo solicitado, informa ao usuário e finaliza a operação.

Seção	Excluir Material Aula
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
	2. O sistema recupera e exibe os materiais didáticos existentes em aula.
3. O professor seleciona os materiais que deseja excluir.	4. O sistema desassocia os materiais da aula e informa o sucesso da operação.

Seção	Organizar Material Aula
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O professor informa a aula que terá seus materiais didáticos organizados.	1. O sistema recupera e exibe os materiais didáticos existentes em aula.
2. O professor informa a ordem lógica dos materiais em aula.	3. O sistema atualiza a ordem dos materiais e informa o sucesso da operação.

CASO DE USO 4 – MANTER CURSO

Seção: **Principal.**

Ator: Professor.

Propósito: Fornecer ao professor interface amigável para manutenção de cursos.

Visão Geral: Manter cursos para determinado domínio. Especificar as aulas que irão compor o curso. Montar seqüência das aulas que compõem o curso. Definir para que nível de conhecimento dos alunos o curso se aplica.

- Funcionalidades: R12 – Manter curso para determinado domínio e nível de conhecimento.
R13 – Identificar aulas que compõem um curso.
R14 – Montar seqüência de aulas que compõem um curso.
R17 – Efetuar *login* do usuário através de identificação e senha.
R18 – Prover um mecanismo de armazenamento consistente.

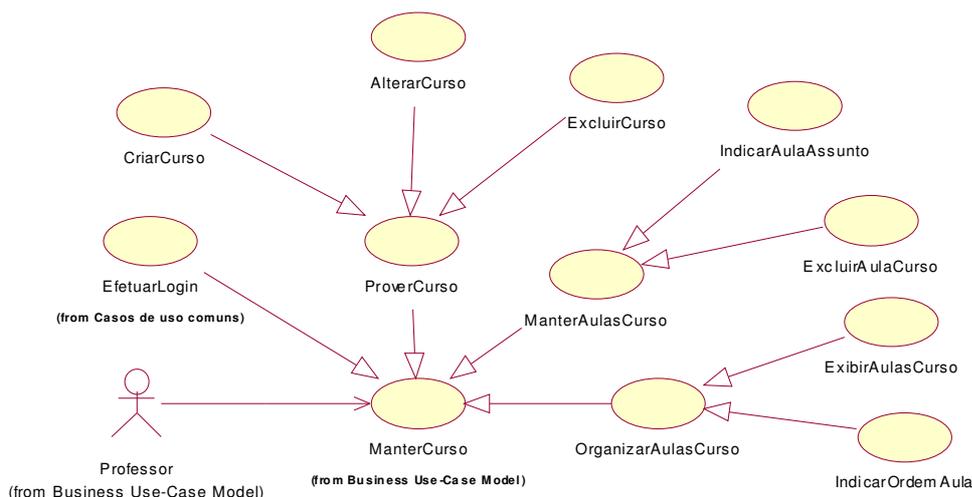


Figura 30 – Caso de uso Manter Curso – Refinamento

Seqüência típica de eventos	Manter Cursos
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. Um usuário acessa o sistema para fazer manutenção de cursos. O usuário informa sua identificação e senha.	2. O sistema verifica se a identificação e senha estão corretas.
3. O professor informa qual a operação pretendida: a) para manter dados cadastrais do curso, ver seção Prover Curso; b) para especificar as aulas que compõem um curso, ver seção Manter Aulas Curso; c) para organizar seqüencialmente as aulas de um curso, ver seção Organizar Aulas Curso.	4. O sistema registra e informa o sucesso da operação.

Seqüência alternativa 2 – O usuário não tem permissão para manter cursos. O sistema informa ao usuário e finaliza a operação.

Seção	Prover Curso
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O professor informa se pretende criar, alterar ou excluir um curso: a) para criar, ver seção Criar Curso; b) para alterar, ver seção Alterar Curso; d) para excluir, ver seção Excluir Curso.	

Seção	Criar Curso
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
	1. O sistema solicita informações para criação do curso.
2. O autor informa nome, domínio e nível de conhecimento do curso a ser criado.	3. O sistema cadastra o curso e informa o sucesso da operação.

Seção	Alterar Curso
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O autor informa o curso a ser alterado.	2. O sistema exibe os dados
3. O autor altera os dados do curso.	

Seqüência alternativa 2 – O sistema não encontra o curso solicitado, informa ao usuário e finaliza a operação.

Seção	Excluir Curso
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O autor informa o curso a ser excluído.	2. O sistema exibe os dados e solicita confirmação.
3. O autor confirma a exclusão	

Seqüência alternativa 2 – O sistema não encontra o curso solicitado, informa ao usuário e finaliza a operação.

Seqüência alternativa 3 – O usuário não confirma a exclusão. O sistema finaliza a operação.

Seção	Manter Aulas Curso
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O professor solicita ao sistema o curso que pretende trabalhar.	2. O sistema recupera e exibe os dados solicitados.
3. O professor informa se pretende incluir ou excluir aulas: a) para incluir novas aulas em curso, ver seção Indicar Aula Assunto; b) para excluir aulas já existentes, ver seção Excluir Aula Curso.	

Seqüência alternativa 2 – O sistema não encontra o curso solicitado, informa ao usuário e finaliza a operação.

Seção	Indicar Aula Assunto
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O professor informa assunto da aula que pretende inserir.	2. O sistema recupera e exibe as aulas solicitadas.
3. O professor seleciona a aula escolhida.	4. O sistema associa a aula ao curso e informa o sucesso da operação.

Seqüência alternativa 2 – O sistema não encontra nenhuma aula do assunto solicitado, informa ao usuário e finaliza a operação.

Seção	Excluir Aula Curso
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
	1. O sistema recupera e exibe as aulas existentes no curso.
2. O professor seleciona a(s) aula(s) que deseja excluir.	3. O sistema desassocia a(s) aulas(s) do curso e informa o sucesso da operação.

Seção	Organizar Aulas Curso
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O professor indica o nome do curso que pretende organizar.	2. O sistema recupera e exibe as aulas existentes no curso.
3. O professor informa a ordem lógica das aulas no curso.	4. O sistema atualiza a ordem das aulas e informa o sucesso da operação.

CASO DE USO 5 – UTILIZAR CURSOS

Seção: Principal.

Ator: Aluno.

Propósito: Fornecer ao aluno interface amigável para estudar sobre determinado domínio.

Visão Geral: Prover autores, professores e alunos de cursos sobre determinado domínio. Os cursos poderão ser utilizados por um ou um grupo de alunos com o mesmo nível de conhecimento sobre o domínio a ser estudado.

Funcionalidades: R15 – Identificar cursos existentes por domínio, nível de conhecimento e autor.

R16 – Exibir curso solicitado.

R17 – Efetuar *login* do usuário através de identificação e senha.

R18 – Prover um mecanismo de armazenamento consistente.

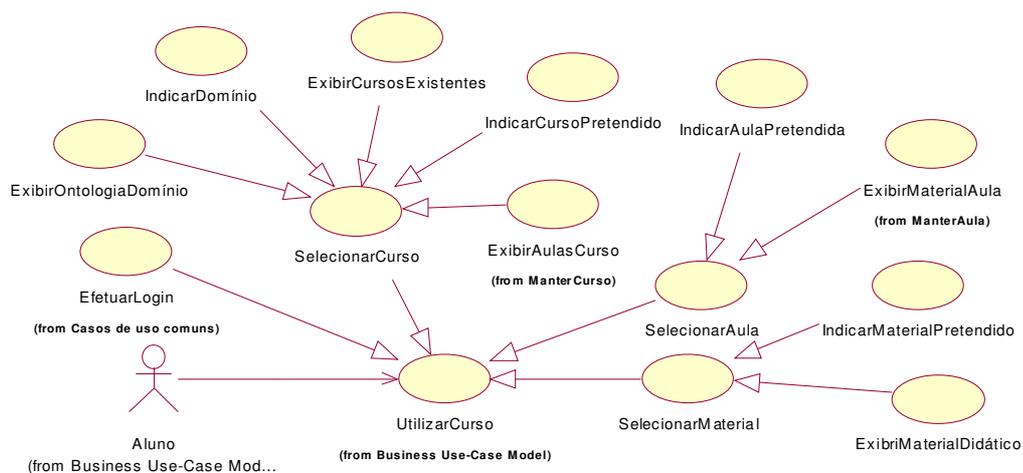


Figura 31 – Caso de uso Utilizar Curso – Refinamento

Seqüência típica de eventos	Utilizar Curso
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. Um usuário acessa o sistema para assistir a um curso: a) para selecionar curso de um determinado domínio e nível de conhecimento, ver seção Selecionar Curso; b) para selecionar aula do curso escolhido, ver seção Selecionar Aula; c) para selecionar material de uma aula selecionada, ver seção Selecionar Material.	2. O sistema finaliza a operação.

Seção	Selecionar Curso
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
	1. O sistema exibe ontologia dos domínios existentes.
2. O professor/aluno informa o domínio e nível de conhecimento que deseja consultar.	3 O sistema recupera e exibe os cursos pertinentes.
4. O professor/aluno seleciona o curso.	5. O sistema recupera e exibe as aulas pertencentes ao curso selecionado.

Seqüência alternativa 2 – O sistema não encontra nenhum curso para o domínio e o nível de conhecimento solicitados e informa ao usuário.

Seção	Selecionar Aula
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O professor/aluno informa a aula que deseja consultar.	2 O sistema recupera e exibe a lista de materiais pertencentes à aula escolhida.

Seção	Selecionar Material
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O professor/aluno informa o material que deseja consultar	2 O sistema recupera e exibe o material didático escolhido.

CASO DE USO 6 – MANTER ONTOLOGIA

Seção:	Principal.
Atores:	Autor e administrador.
Propósito:	Manter ontologias de domínios e assuntos.
Visão Geral:	Prover autores e professores de uma ontologia de domínios e assuntos a respeito dos quais pretendam desenvolver cursos, de forma organizada, para evitar a duplicação de conceitos.
Funcionalidades:	R19 – Manter ontologias de domínio do conhecimento e assuntos. R17 – Efetuar <i>login</i> do usuário mediante identificação e senha. R18 – Prover um mecanismo de armazenamento consistente.

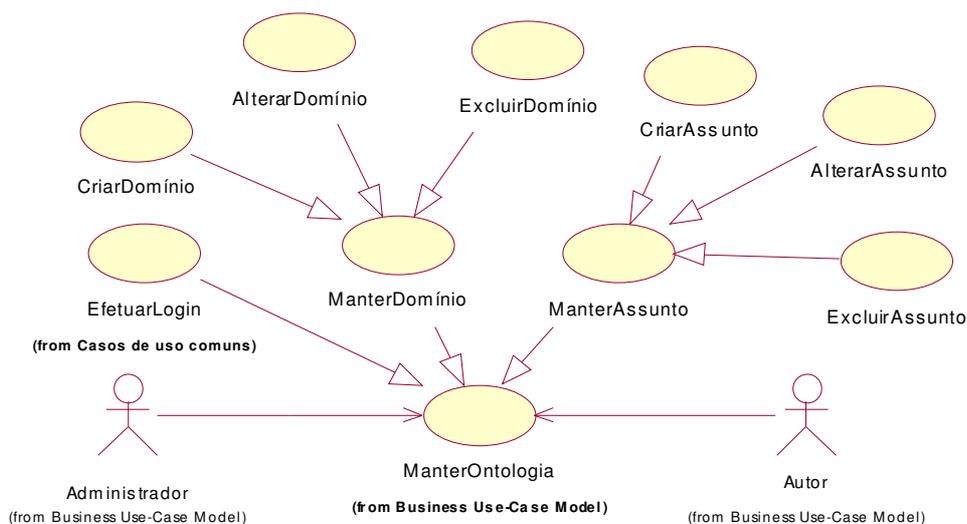


Figura 32 – Caso de uso Manter Ontologia – Refinamento

Seqüência típica de eventos	Manter Ontologia
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. Autores, especialistas em determinado domínio do conhecimento, solicitam ao administrador a manutenção das antologias de domínio e de assuntos em questão: a) para manter ontologia do domínio, ver seção Manter Domínio; b) para manter ontologia do assunto, ver seção Manter Assunto.	2. O sistema finaliza a operação.

Seção	Manter Domínio
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O administrador acessa o sistema para manter ontologia do domínio: a) para criar novo domínio, ver seção Criar Domínio; b) para alterar ontologia do domínio, ver seção Alterar Domínio; c) para excluir domínio, ver seção Excluir Domínio.	

Seção	Criar Domínio
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
	1. O sistema solicita informações para criação de domínio.
2. O administrador informa nome, descrição, sinônimo, autor e data de criação do domínio.	3. O sistema cadastra o domínio e informa o sucesso da operação.

Seção	Alterar Domínio
<u>Ação do Ator</u>	<u>Resposta do Sistema.</u>
1. O autor informa o domínio a ser alterado.	2. O sistema exibe os dados.
3. O autor altera os dados conforme sua necessidade.	

Seqüência alternativa 2 – O sistema não encontra o domínio solicitado, informa ao usuário e finaliza a operação.

Seção	Excluir Domínio
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O administrador informa o domínio a ser excluído.	2. O sistema exibe os dados e solicita confirmação.
3. O administrador confirma a exclusão	

Seqüência alternativa 2 – O sistema não encontra o domínio solicitado, informa ao administrador e finaliza a operação.

Seção	Manter Assunto
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O administrador acessa o sistema para manter ontologia de assuntos: a) para criar novo assunto, ver seção Criar Assunto; b) para alterar assunto,	

ver seção Alterar Assunto; c) para excluir assunto, ver seção Excluir Assunto.	
--	--

Seção	Criar Assunto
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
	1. O sistema solicita informações para criação de assunto.
2. O administrador informa nome, descrição, sinônimo, autor e data de criação do assunto e domínio a que pertence o assunto.	3. O sistema cadastra o assunto e informa o sucesso da operação.

Seção	Alterar Assunto
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O autor informa o assunto a ser alterado.	2. O sistema exibe os dados.
3. O autor altera os dados conforme sua necessidade.	

Seqüência alternativa 2 – O sistema não encontra o assunto solicitado, informa ao usuário e finaliza a operação.

Seção	Excluir Assunto
<u>Ação do Ator</u>	<u>Resposta do Sistema</u>
1. O administrador informa o assunto a ser excluído.	2. O sistema exibe os dados e solicita confirmação.
3. O administrador confirma a exclusão.	

Seqüência alternativa 2 – O sistema não encontra o assunto solicitado, informa ao administrador e finaliza a operação.

6.3 ESPECIFICAÇÃO DO SISTEMA

A especificação do sistema descreve o comportamento externo dos objetos, dentro de um contexto, servindo-se de cenários e modelos de especificação como, por exemplo os diagramas de interação (D'SOUZA, 1998, p.519). Um diagrama de interação consiste em um conjunto de objetos e seus relacionamentos, incluindo as mensagens que podem ser trocadas entre eles. Um diagrama de seqüência é um diagrama de interação, cuja ênfase está na ordem temporal das mensagens. O diagrama de colaboração também é um diagrama de interação, cuja ênfase está na organização estrutural dos objetos, que enviam e recebem mensagens. O diagrama de seqüência e o de colaboração são isomórficos, ou seja, um tipo de diagrama pode ser transformado no outro tipo (BOOCH, 2000, p.241-247).

A especificação do SGC será definida pela combinação da narrativa de cenários com diagramas de seqüência. Os diagramas de seqüência e narrativas que não serão implementados no contexto do protótipo do SGC estarão definidos no Anexo II. Os diagramas de colaboração estarão contidos no Anexo III.

6.3.1 CENÁRIOS E DIAGRAMAS DE SEQUÊNCIA

CONTEXTO – MANTER CADASTRO DE USUÁRIOS

CENÁRIO – CADASTRAR USUÁRIO

O administrador do sistema é solicitado por um usuário para fazer um novo cadastramento no cadastro de usuários. O administrador efetua *login* e informa os dados cadastrais do novo usuário (Figura 33).

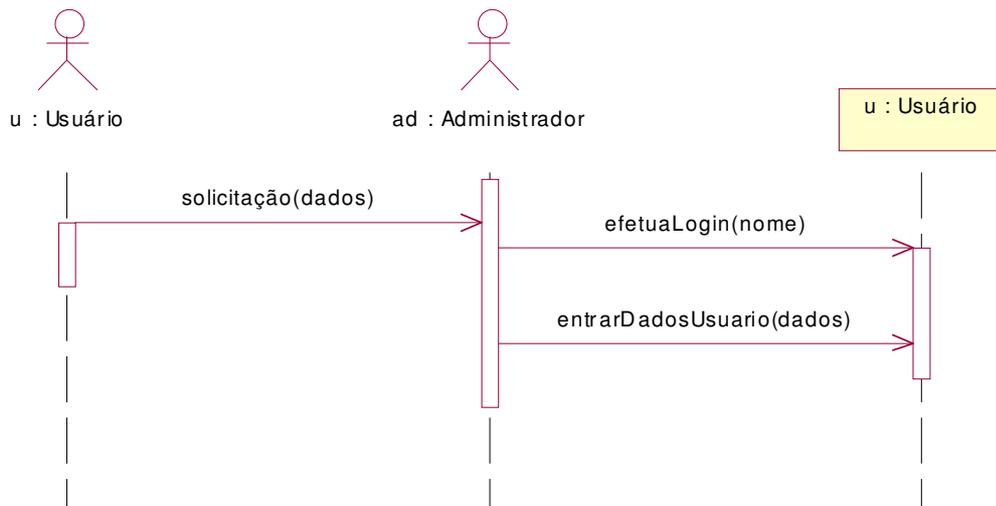


Figura 33 – Diagrama de Seqüência – Cadastrar Usuário

CONTEXTO – MANTER MATERIAL DIDÁTICO

CENÁRIO – CRIAR MATERIAL DIDÁTICO

Um autor acessa o sistema para criar um material didático e informa nome e senha. O sistema informa tipos de materiais existentes e tipos de assunto. O autor informa o tipo de material, o assunto e os dados do material. O sistema cria um novo material (Figura 34).

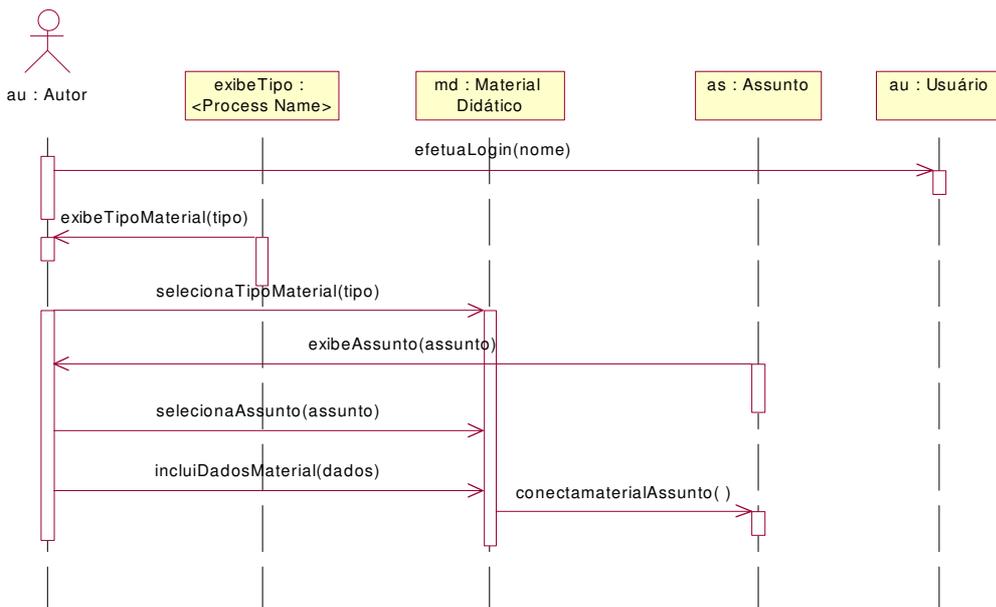


Figura 34 – Diagrama de Seqüência – Criar Material

CONTEXTO – MANTER AULA

CENÁRIO – CRIAR AULA

Um professor acessa o sistema para criar uma aula, efetua *login* e informa o assunto e os dados cadastrais de uma aula (Figura 35).

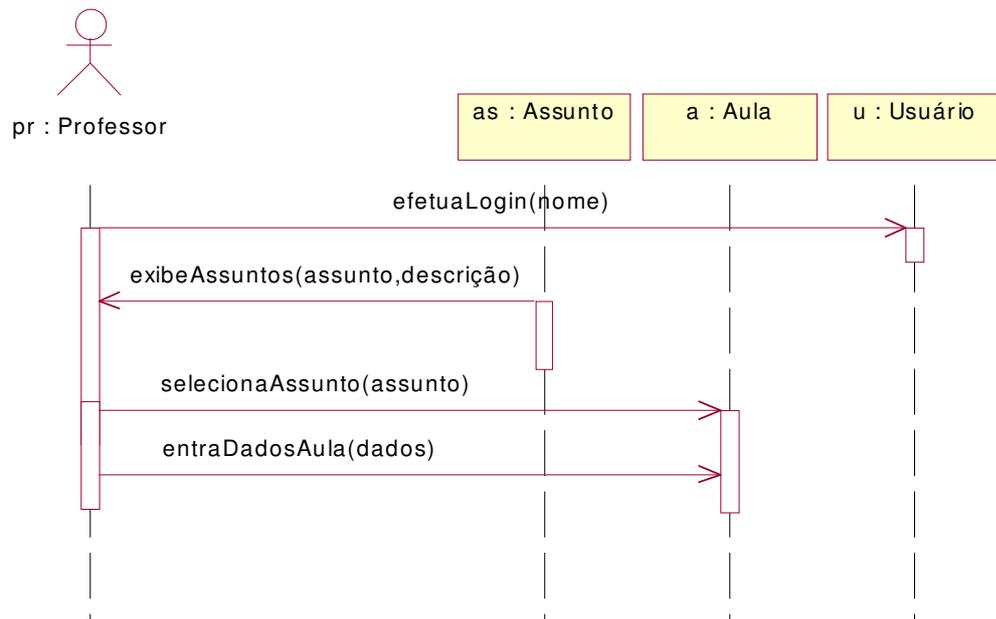


Figura 35 – Diagrama de Seqüência – Criar Aula

CENÁRIO – ALOCAR MATERIAL DIDÁTICO EM AULA

Contexto – Um professor acessa o sistema para agregar material didático a uma aula; efetua *login* e informa a aula a ser trabalhada. O sistema exibe os dados cadastrais da aula e a lista de material didático existente sobre o assunto da aula. O professor seleciona o(s) assunto(s) que incluirá em sua aula. O sistema associa o material indicado à aula (Figura 36).

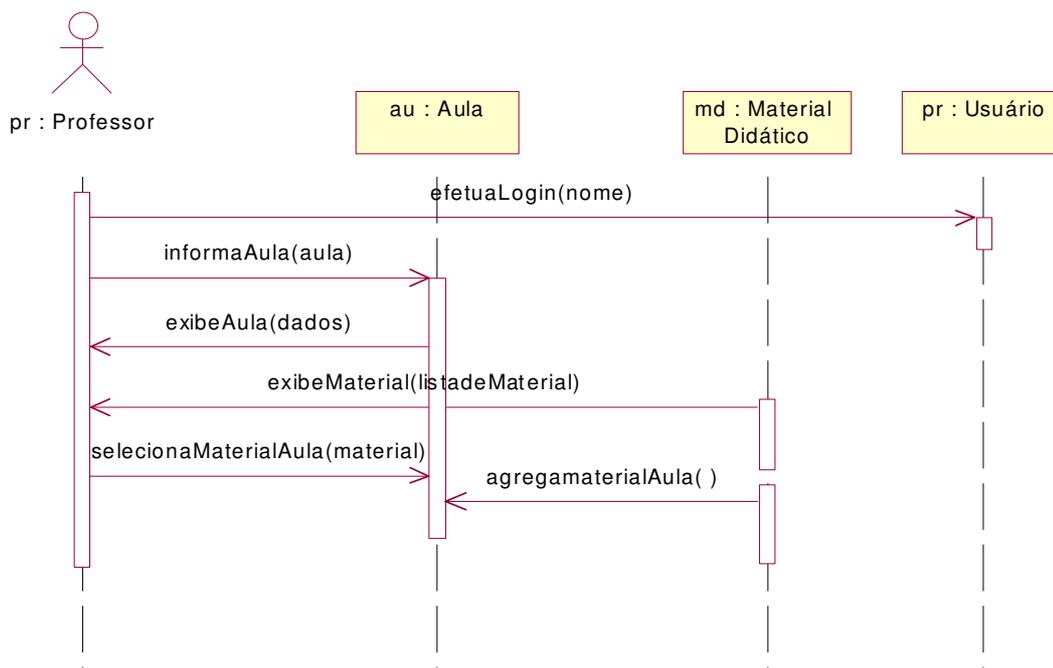


Figura 36 – Diagrama de Seqüência Alocar Material em Aula

CENÁRIO – ORGANIZAR MATERIAL DIDÁTICO DE UMA AULA

Contexto – Um professor acessa o sistema para organizar o material didático de uma aula, efetua *login* e informa a aula a ser trabalhada. O sistema exibe os dados cadastrais da aula e a lista de material didático já alocado nela. O professor informa a ordem de apresentação do material didático da aula (Figura 37).

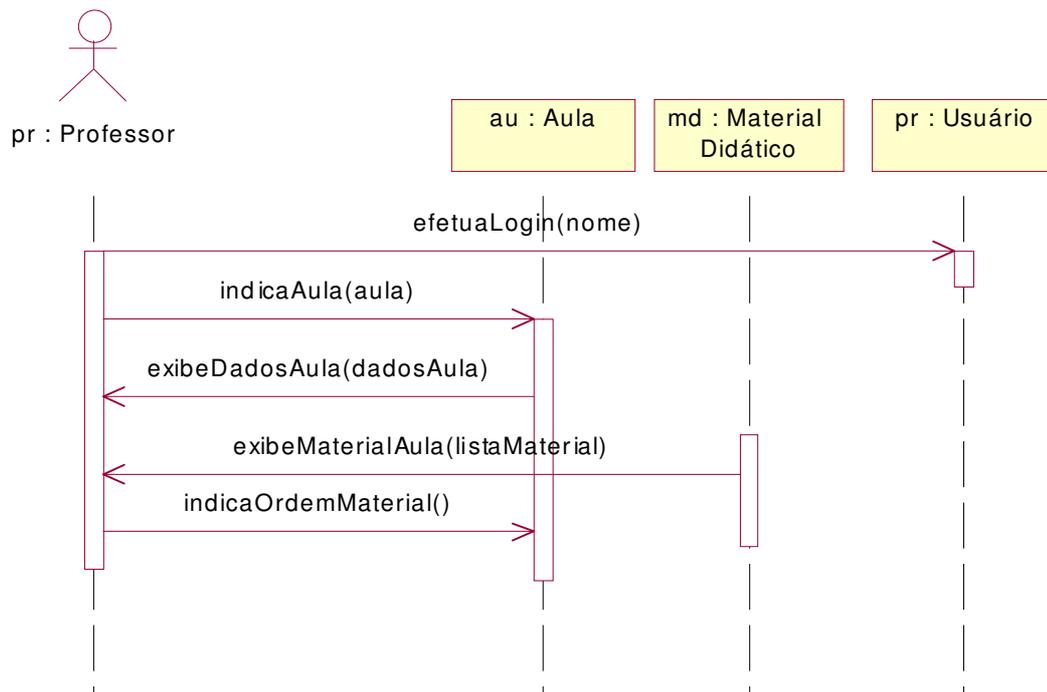


Figura 37 – Diagrama de Seqüência – Organizar Material Didático de uma Aula

CONTEXTO – MANTER CURSO

CENÁRIO – CRIAR CURSO

Um professor acessa o sistema para criar um curso e informa o domínio de conhecimento, o nível de conhecimento e os dados cadastrais do curso (Figura 38).

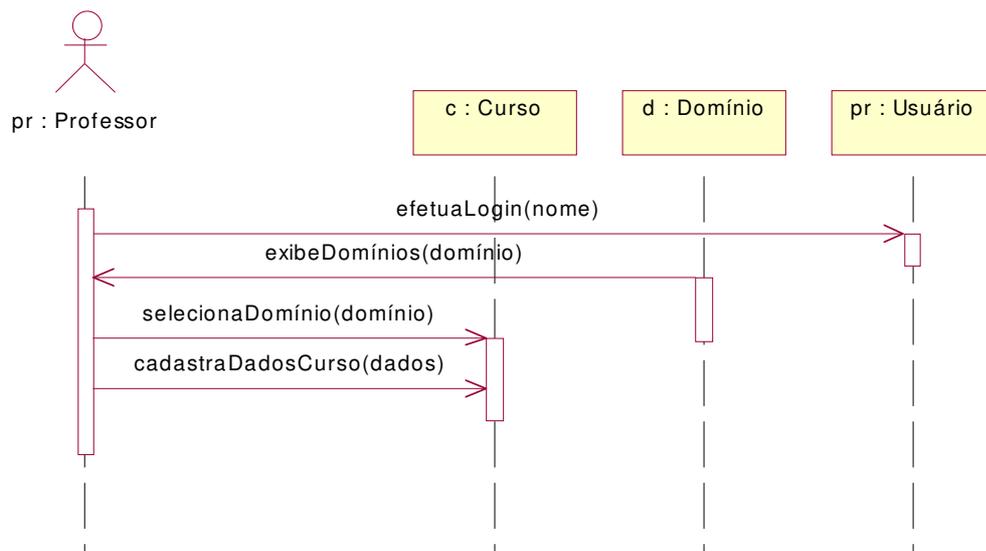


Figura 38 – Diagrama de Seqüência – Criar Curso

CENÁRIO – ALOCAR AULAS EM CURSO

Contexto – Um professor acessa o sistema para indicar aulas de um curso. Efetua *login* e informa o curso a ser alterado. O sistema exibe os dados. O professor confirma a operação (Figura 39).

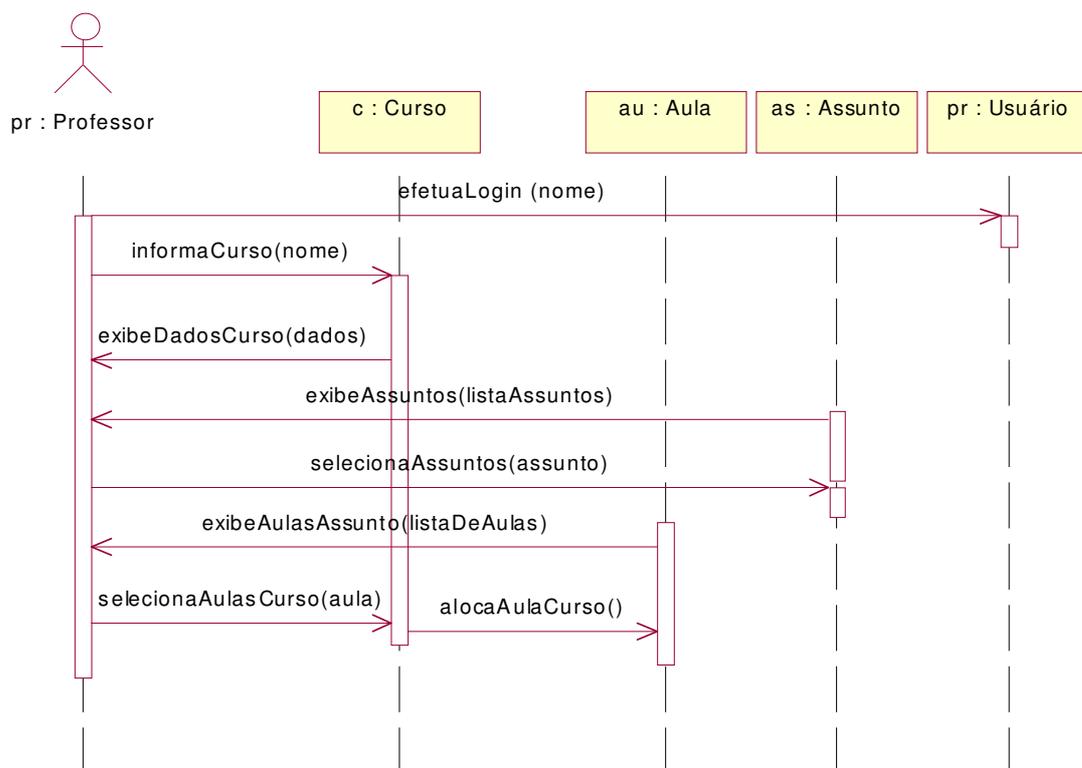


Figura 39 – Diagrama de Seqüência – Alocar Aulas em Curso

CENÁRIO – ORGANIZAR AULAS DE UM CURSO

Contexto – Um professor acessa o sistema para organizar as aulas de um curso, efetua *login* e informa o curso a ser trabalhado. O sistema exibe a lista de aulas alocadas no curso. O professor informa a ordem de apresentação das aulas do curso (Figura 40).

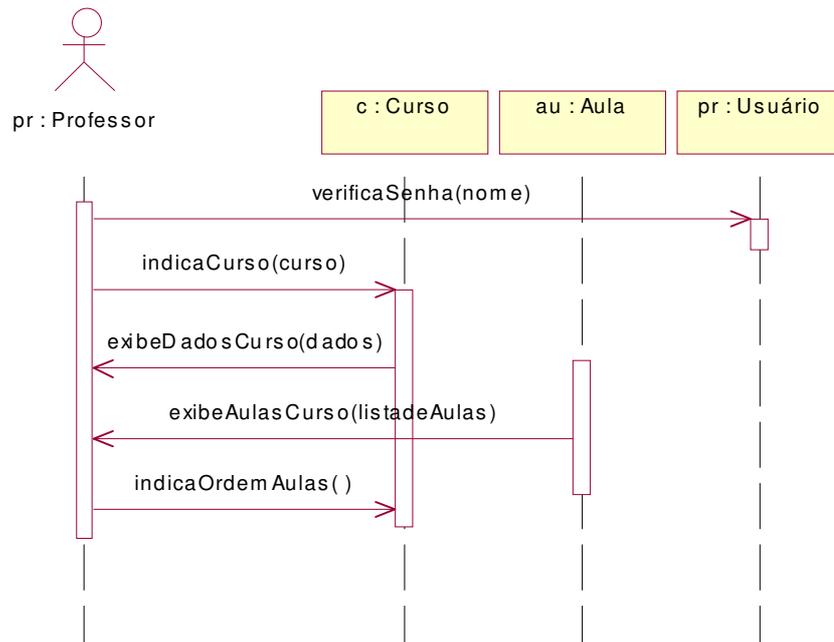


Figura 40 – Diagrama de Seqüência – Organizar Aulas de um Curso

CONTEXTO – MANTER ONTOLOGIA

CENÁRIO – CRIAR DOMÍNIO

O administrador acessa o sistema e entra com os dados do novo domínio do conhecimento (Figura 41).

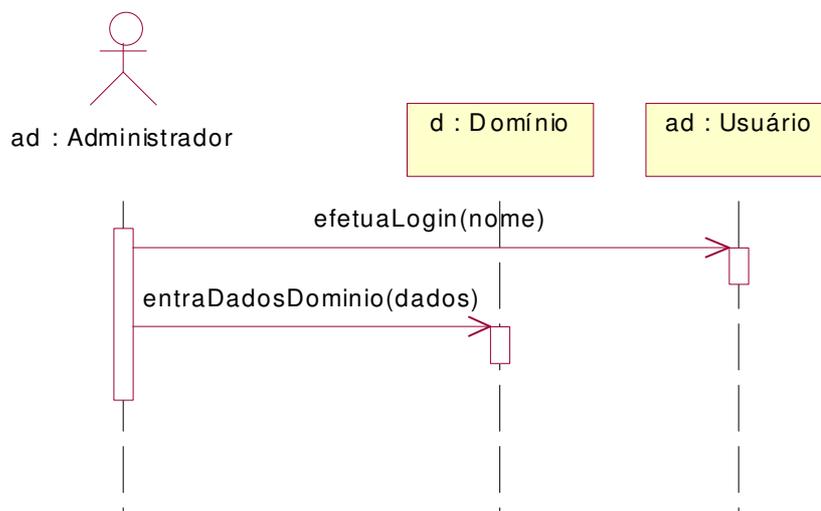


Figura 41 – Diagrama de Seqüência – Criar Domínio

CENÁRIO – CRIAR ASSUNTO

Contexto – O administrador acessa o sistema para criar novo assunto. O sistema exibe os domínios existentes. O administrador seleciona o domínio e informa os dados do assunto a ser criado (Figura 42).

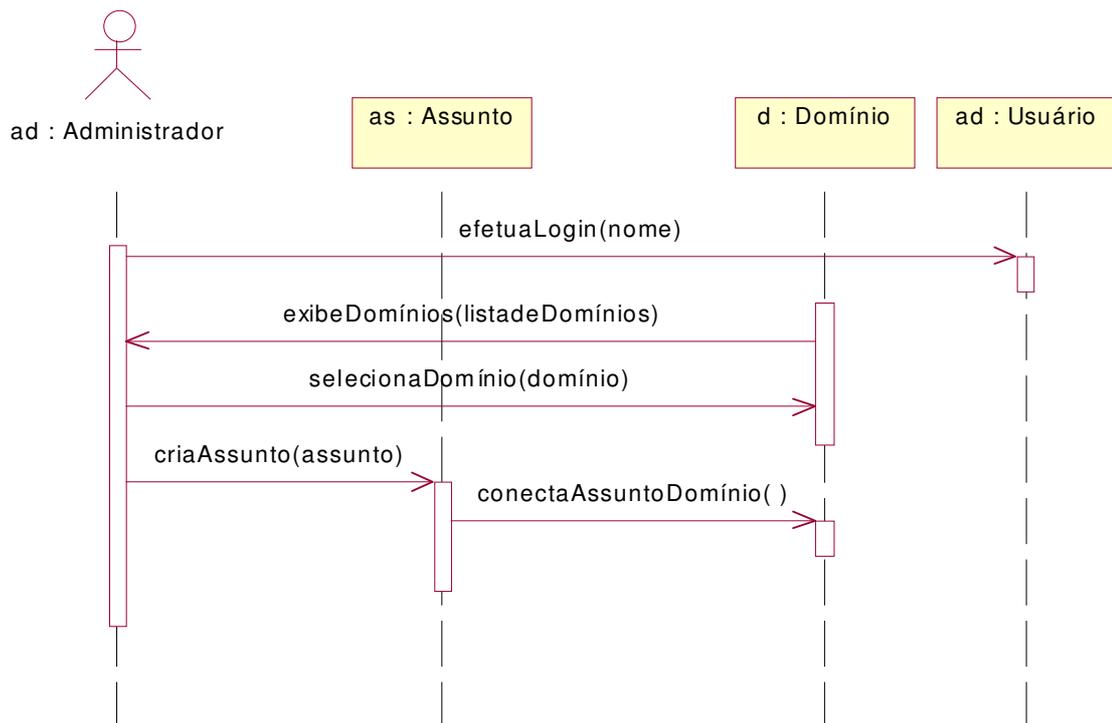


Figura 42 – Diagrama de Seqüência – Criar Assunto

UTILIZAR CURSO

CENÁRIO 1 – UTILIZAR CURSO

Contexto – O aluno acessa o sistema para utilizar um curso. O sistema exibe os domínios existentes. O aluno seleciona o domínio. O sistema mostra os cursos existentes para o domínio selecionado. O aluno escolhe o curso. O sistema apresenta as aulas pertencentes ao curso e o aluno seleciona a aula que pretende estudar. O sistema informa os materiais

didáticos pertencentes à aula. O aluno seleciona o material didático e o sistema exibe o material selecionado (Figura 43).

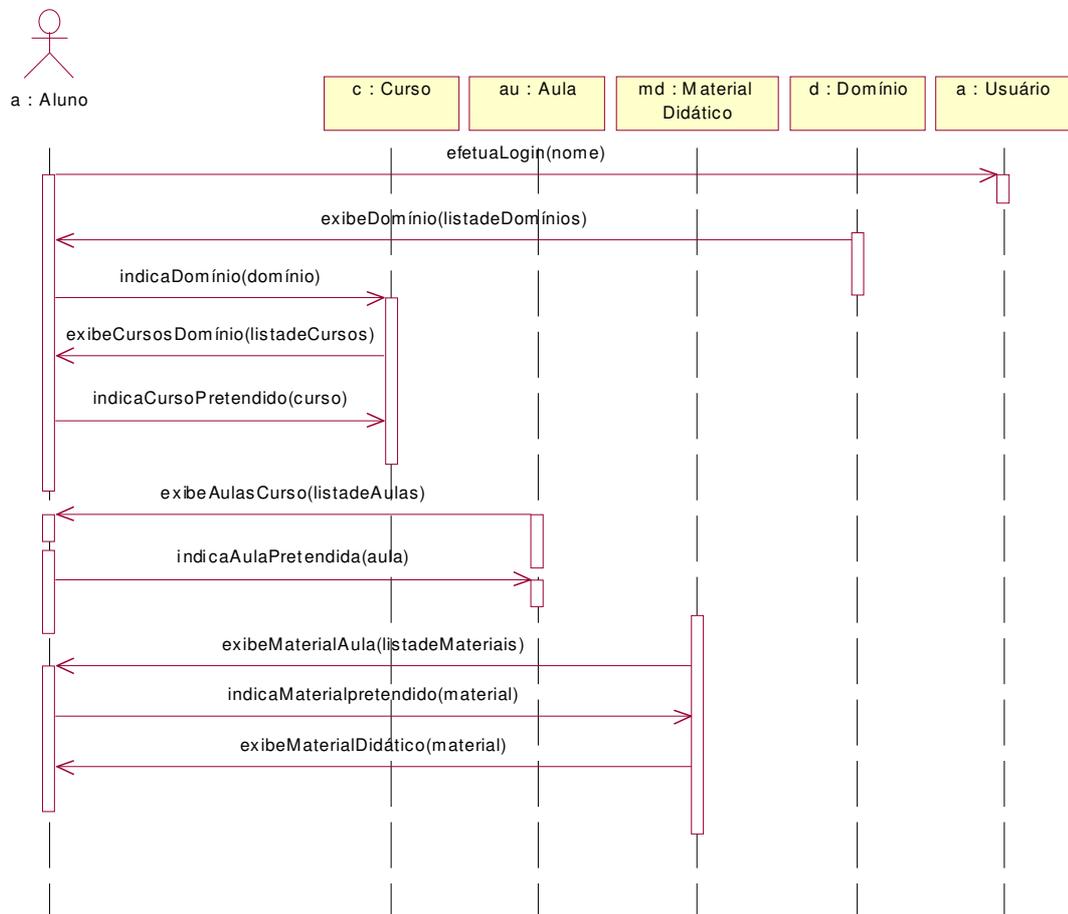


Figura 43 – Diagrama de Seqüência – Utilizar curso

6.4 ARQUITETURA GERAL DO SISTEMA

Segundo Booch (2000), para modelar a arquitetura de um sistema é necessário capturar decisões sobre os requisitos do sistema e seus elementos lógicos físicos. Deve-se também fazer a modelagem dos aspectos estruturais e comportamentais do sistema.

A Figura 44 ilustra a arquitetura do SGC, composta de três camadas: Camada de Apresentação, Camada de Aplicação e Camada de Armazenamento. A Camada de Apresentação contém os componentes que fazem a interface do sistema com os usuários e é relativamente livre dos processos ligados à aplicação. Esses componentes repassam as solicitações de tarefas para a Camada de Aplicação. Essa Camada possui a estrutura lógica da aplicação, cujas regras comandam o processamento do sistema. É formada por três componentes que representam o controle dos usuários, dos componentes de aprendizagem e da ontologia do sistema. É ela também que se comunica com a Camada de Armazenamento, por trás da aplicação, mediante interface com o banco de dados, fornecida pelo fabricante do banco. A Camada de Armazenamento representa um banco de dados relacional.

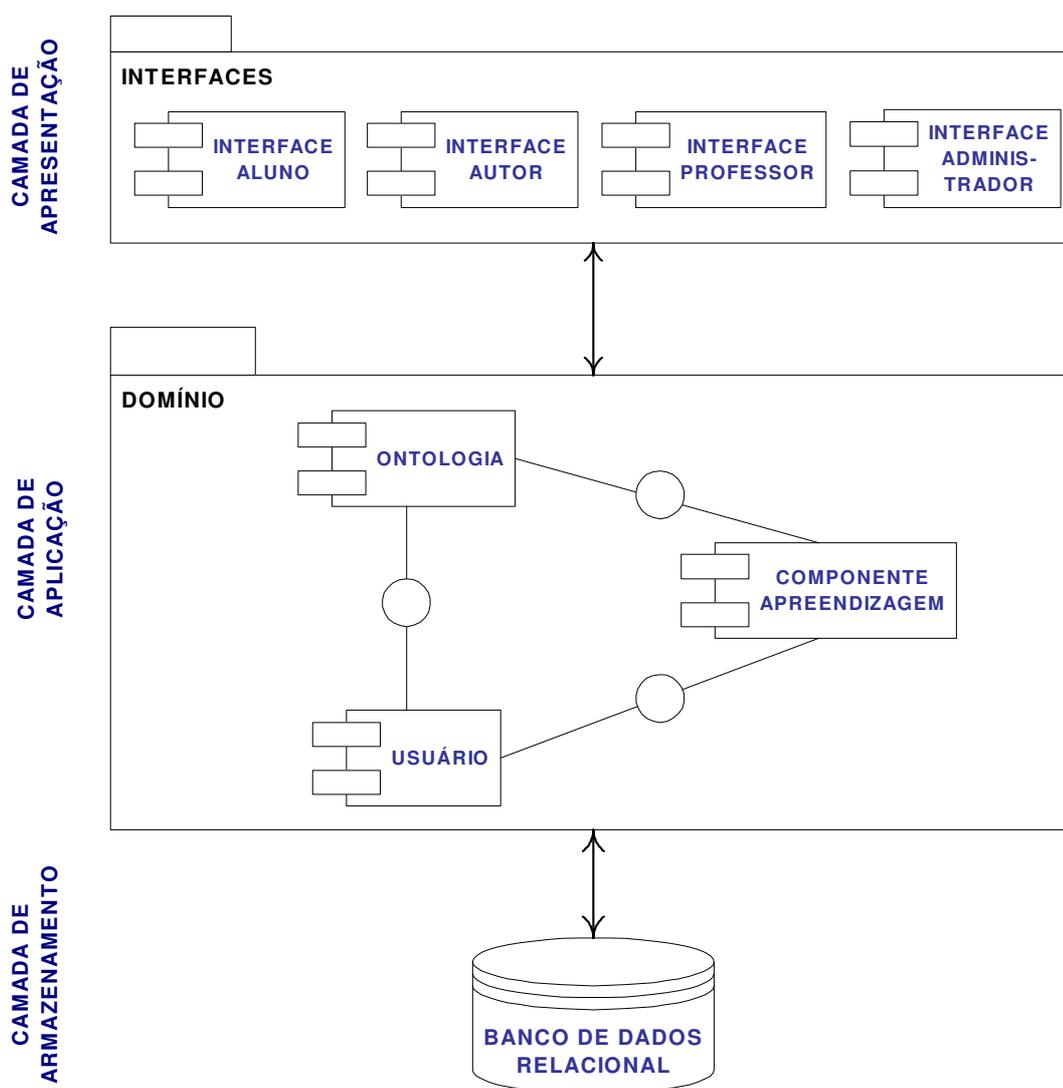


Figura 44 – Arquitetura geral do Sistema

6.5 PROJETO INTERNO - COMPONENTES

Segundo D’Souza (1998), cada componente da aplicação deve ser projetado e implementado como um conjunto de classes que realizam as interfaces solicitadas externamente e as interações internas necessárias.

A seguir, apresenta-se o projeto interno dos três componentes da Camada de Aplicação, com uma representação gráfica e de uma descrição sucinta. O capítulo 7 conterà a análise detalhada de cada componente

6.5.1 COMPONENTE – USUÁRIO

A Figura 45 ilustra a implementação do componente Usuário. A interface Usuário define o comportamento comum aos usuários do tipo Professor, Aluno e Administrador. Cada tipo de usuário é definido por uma classe, que realiza todos os métodos comuns definidos na interface, acrescidos dos métodos específicos para cada tipo de usuário. Por exemplo: a classe Professor implementa os métodos “setSenhaProfessor()” e “getSenhaprofessor()”, pertinentes a essa classe, além daqueles definidos na interface Usuário.

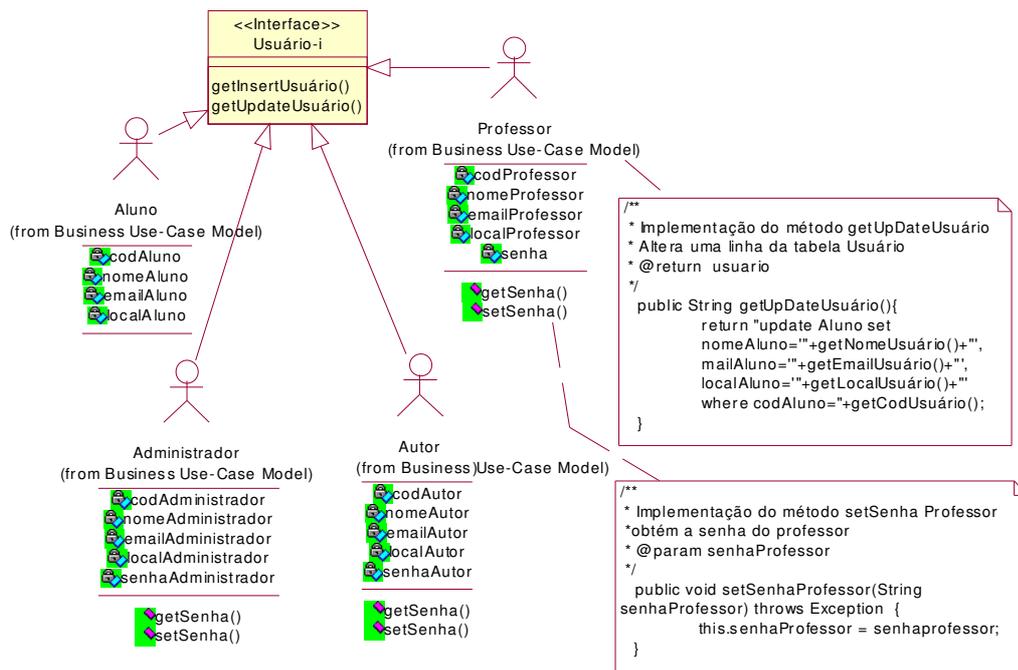


Figura 45 – Projeto interno do Componente Usuário

6.5.2 COMPONENTE – ONTOLOGIA

Na Figura 46 encontra-se a implementação do componente Ontologia. A interface Ontologia define o comportamento comum aos tipos de ontologia Domínio e Assunto. Ambos são definidos por classes, que realizam todos os métodos comuns definidos na interface, acrescidos dos métodos específicos para cada tipo de Ontologia. Por exemplo: a Ontologia de Assunto, implementa os métodos definidos na interface Ontologia, e também os métodos “setCodDominio()” e “getCodDominio()”, que são específicos da classe Assunto.

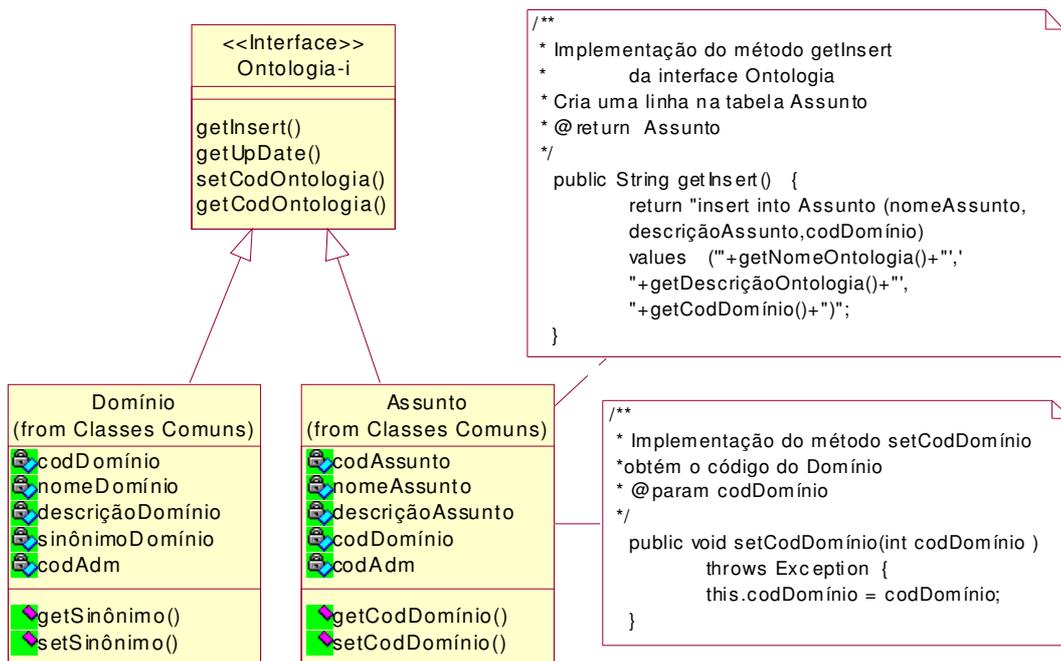


Figura 46 – Projeto interno do componente Ontologia

6.5.3 COMPONENTE – COMPONENTE DE APRENDIZAGEM

A Figura 47 representa a implementação do componente Componente de Aprendizagem. A interface Componente de Aprendizagem define o comportamento comum aos Componentes de Aprendizagem do tipo Curso, Aula e Material Didático, os quais de definem mediante classes, que realizam todos os métodos comuns definidos na interface, acrescidos dos métodos específicos para cada tipo de Componente de Aprendizagem. O Componente de Aprendizagem do tipo Material Didático, por exemplo, além de implementar os métodos definidos na interface, também implementa os métodos “setConteúdoMaterial()” e “getConteúdoMaterial()”, pertinentes apenas à classe Material Didático.

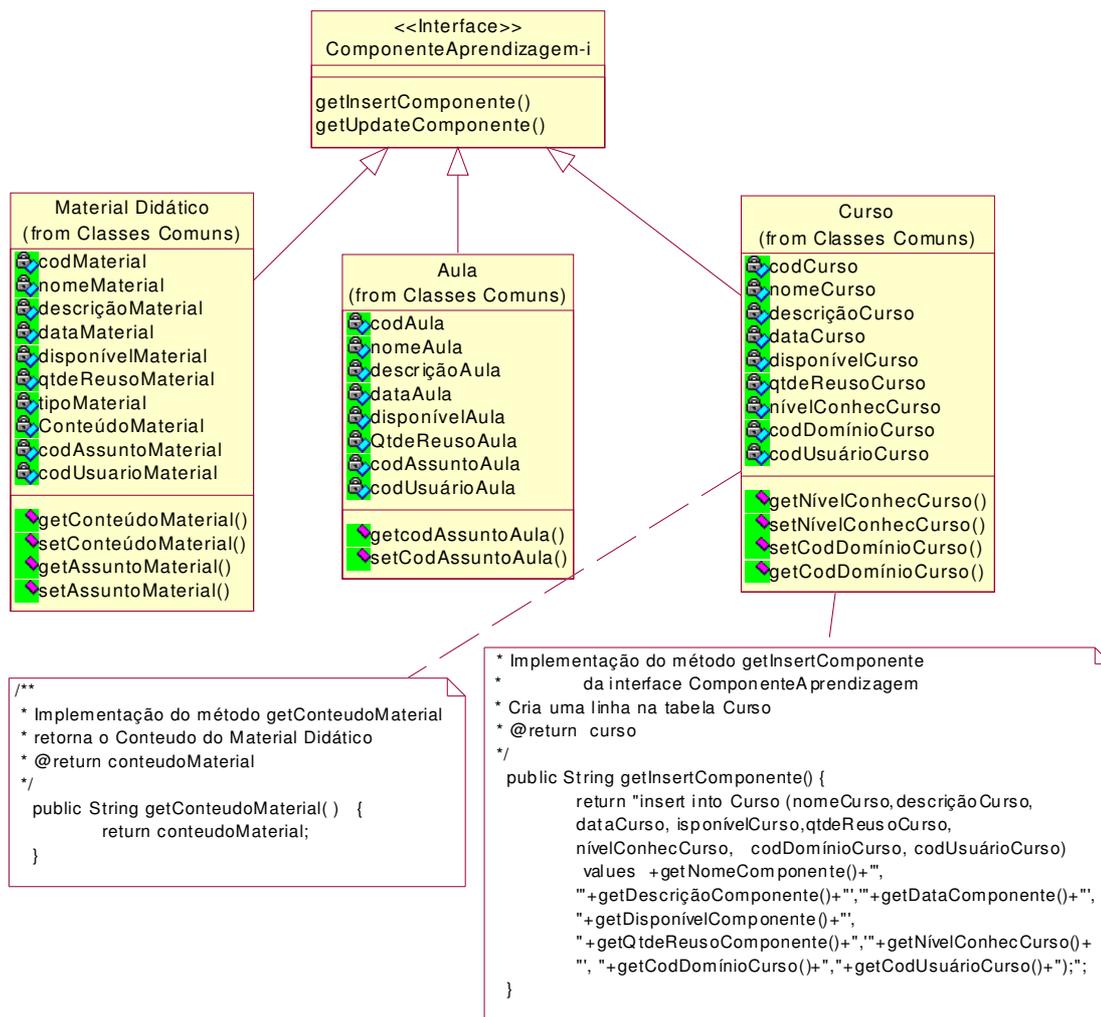


Figura 47 – Projeto interno do componente Componente de Aprendizagem

6.6 CONCLUSÃO

De acordo com Booch (2000), os diagramas de casos de uso são criados para definir o comportamento desejado do sistema; os diagramas de classes têm o objetivo de especificar o vocabulário do domínio; os diagramas de seqüência e de colaboração são definidos para especificar a forma como os itens de seu vocabulário trabalharão em conjunto para a execução desse comportamento. Para transformar esse projeto lógico em itens que “vivem no mundo dos bits”, a exemplo de executáveis, tabelas, bibliotecas, arquivos e documentos, é necessário criar alguns componentes ou reutilizar componentes antigos. Na UML, os diagramas de

componentes são empregados para facilitar a visualização do aspecto estático desses componentes físicos e seus relacionamentos e para especificar detalhes para a construção.

Este capítulo apresentou a modelagem do SGC, com aplicação da linguagem gráfica UML para a elaboração da estrutura de um projeto de *software*, usando-se a metodologia *Catalysis*.

O próximo capítulo versará sobre a implementação do protótipo do SGC, baseada nessa modelagem.

PARTE III

**IMPLEMENTAÇÃO DO SISTEMA GERENCIADOR
DE COMPONENTES**

7 IMPLEMENTAÇÃO DO SGC

O aplicativo SGC- Sistema Gerenciador de Componentes de Aprendizagem²⁰ foi construído com base na modelagem definida no capítulo anterior e tem como objetivo gerenciar o desenvolvimento de material didático. Trata-se de um protótipo, no qual foram desenvolvidas apenas algumas funções, escolhidas de forma a abranger os aspectos mais importantes da modelagem. Enquanto protótipo, tem a intenção de facilitar o entendimento do modelo e demonstrar que, por ser implementado através do modelo de componentes pode integrar-se a outros projetos da ACVA, conforme proposta inicial deste trabalho.

7.1 FERRAMENTAS UTILIZADAS

Para auxiliar no desenvolvimento da aplicação, utilizaram-se várias ferramentas: a linguagem Java, o *JSP*, o *JBuilder* e o *Tomcat*, e o banco de dados *SQL Server*.

Escolheu-se a linguagem de programação Java por ser orientada a objetos, considerada de alto nível, distribuída, interpretada, segura, portátil e de alto desempenho, que atende bem a sistemas orientados a componentes.

O *JSP (Java Server Pages)*²¹ é uma tecnologia para desenvolvimento de aplicações *WEB* que tem, dentre outras, a característica de portabilidade de plataforma, uma vez que é uma extensão do modelo de programação *Java*. Pode, portanto, ser executado em vários Sistemas Operacionais.

20. A documentação e códigos-fonte do sistema e a definição do banco de dados encontram-se no Anexo II

21. <http://www.jspbrasil.com.br:8081/jspbrasil/main/main.jsp>

Semelhante aos *servlets*, ele oferece mecanismos para captação de informações em formulários e em banco de dados no servidor, etc., produzindo com isso conteúdos dinâmicos. Tem a vantagem de ser facilmente codificado, pois sua estrutura permite separar a programação lógica (parte dinâmica) da programação visual (parte estática). Para construir uma aplicação usando *JSP*, escreve-se o texto em *HTML*, incluindo-se fragmentos do código Java e *tags* especiais, do tipo *Scriptlet* (<% e %>) Essa característica do *JSP* possibilita a criação, dentro de *scriptlets*, de objetos de uma classe que acessa uma base de dados. A utilização dos métodos desses objetos permite a gravação/recuperação de informações, que serão exibidas ao usuário.

Quanto ao *JBuilder*²², é um *software* que oferece um ambiente para desenvolvimento de aplicações para a *Web*, usando a linguagem de programação Java. O *Tomcat* é um *software*, desenvolvido pela Fundação Apache²³, que propicia a execução de aplicações para *Web*. Sua principal característica técnica é o uso da programação Java, mais especificamente das tecnologias de *Servlets*²⁴ e de *JSP*.

O *SQL Server* é um banco de dados relacional. Para viabilizar a implementação em um banco de dados relacional, fez-se necessário acrescentar as classes “*AulaMaterial*” e “*CursoAula*”, que controlam, respectivamente, o conjunto de Materiais Didáticos pertencentes a uma determinada Aula e o conjunto de Aulas pertencentes a um determinado Curso, conforme pode ser visualizado na Figura 48, que demonstra o diagrama de classes implementadas no *SQL Server*.

22. <http://www.borland.com/jbuilder/>

23. <http://www.apache.org/tomcat>

24. Um *servlet* é uma classe escrita em Java cujos objetos têm a finalidade de gerar documentos codificados em *HTML*.

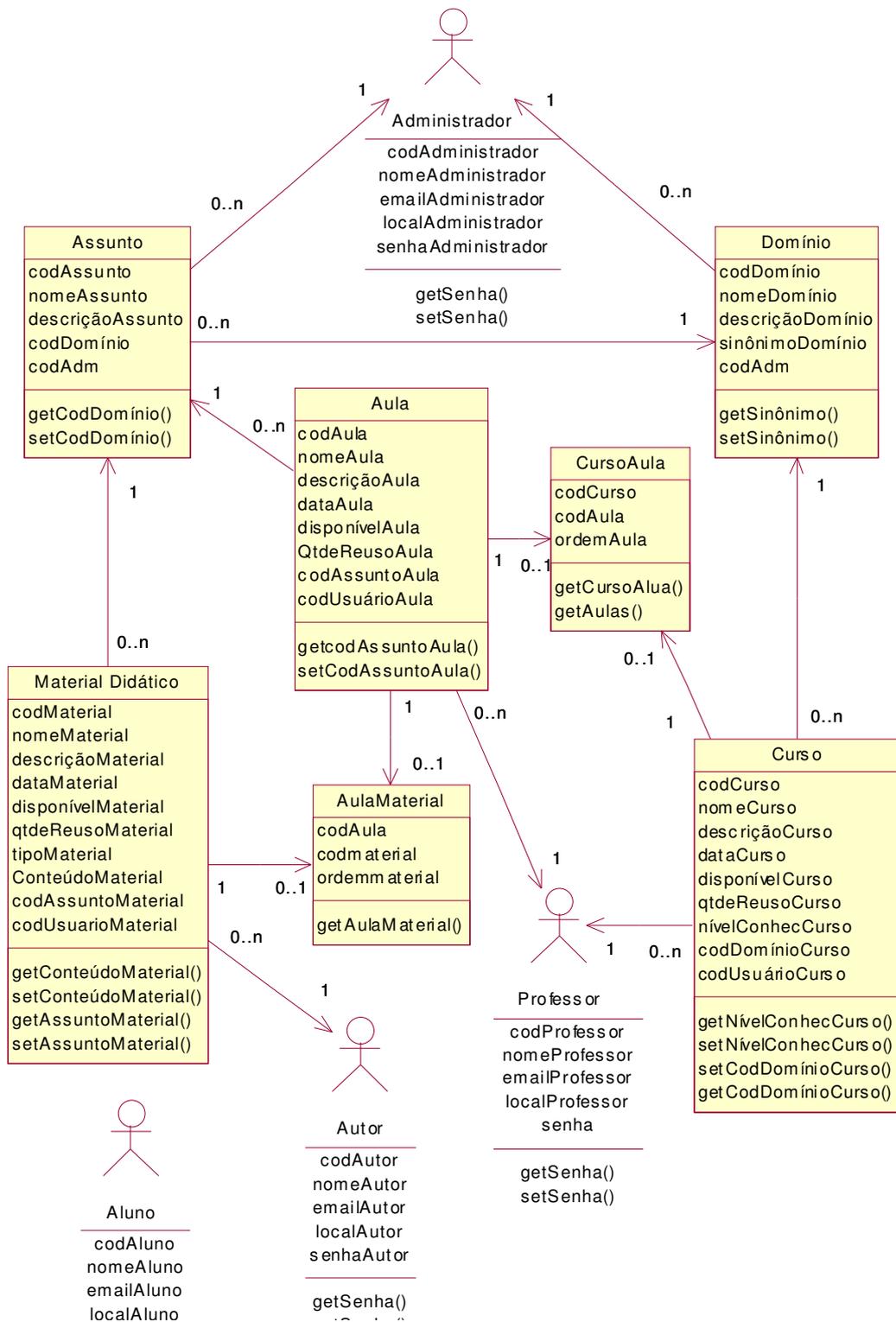


Figura 48 – Diagrama de Classes pertinentes ao banco de dados SQL Server

7.2 OS COMPONENTES DO SGC

Para Booch (2000), um componente é uma parte física e substituível de um sistema, que está em conformidade e proporciona a realização de um conjunto de interfaces. Os componentes podem, portanto, ser empregados para a modelagem dos elementos físicos que residem em um nó, como executáveis, bibliotecas, tabelas, arquivos e documentos. Um componente tipicamente representa o pacote físico de elementos lógicos, como classes, interfaces e colaborações. As interfaces ultrapassam as fronteiras lógica e física. A mesma interface utilizada ou realizada por um componente será utilizada ou realizada pelas classes que o componente implementa. Uma interface pode ser importada por um componente e exportada por outro. O fato de essa interface se encontrar entre dois componentes quebra a dependência direta entre eles.

O objetivo principal da fase de implementação do SGC foi criar um sistema a partir da junção de partes que trabalhassem juntas mas que pudessem ser utilizadas de forma separada e substituídas com facilidade, a exemplo do componente “AcessoBanco”, abaixo descrito.

```
package sgc;

/**
 * Title:
 * Description:
 * Copyright: Copyright (c) 2002
 * Company:
 * @author
 * @version 1.0
 */

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class AcessoBanco {
    public static Statement getStmt() throws Exception {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        String url = "jdbc:odbc:SGC";
        String user= "sa";
        String senha = "";
        Connection con = DriverManager.getConnection(url, user, senha);
        Statement stmt = con.createStatement();
        return stmt;
    }
}
```

Este componente foi criado para fazer conexão com o banco de dados *SQL Server*. Caso um outro banco venha a ser utilizado, troca-se apenas o componente “AcessoBanco” por outro que tenha sido desenvolvido para acessar um modelo deferente de banco de dados, sem que sejam necessárias outras alterações no sistema.

Seguindo o mesmo raciocínio usado na criação do componente “AcessoBanco”, desenvolveram-se outros componentes com o objetivo de atender às funções do SGC.

7.2.1 FUNÇÕES DO ADMINISTRADOR

CRIAR USUÁRIO

O SCG pode ser acessado por quatro tipos de usuários: Aluno, Administrador, Autor e Professor. Existem atributos e métodos comuns aos quatro e outros específicos de cada tipo. Nesse contexto, a fim de reaproveitar a definição de atributos e métodos comuns e preservar as diferenças, foi criada a interface Usuário, que define os atributos e métodos comuns a todos os tipos de usuários. Assim, cada tipo de usuário deu origem a uma classe, que implementa tanto os métodos comuns, definidos na interface Usuário, quanto os métodos e atributos específicos da própria classe.

A Figura 49 apresenta o diagrama de componentes que especifica a construção dos programas desenvolvidos para a criação de um objeto Aluno. Para incluir dados de um novo aluno, o sistema, usando o componente “CadAluno.jsp”, gera uma interface para o Administrador, que digita os dados solicitados. Esses dados são transferidos para o componente “RecebeCadAluno.jsp”, que utiliza o método “getInsertUsuário”, definido na interface Usuário e implementado pela classe Aluno, para inserir uma nova linha na tabela Aluno. O acesso ao banco de dados é feito através da classe “AcessoBanco”.

A inclusão no sistema de um novo professor, autor ou administrador é feita de forma similar à da criação de um aluno. A classe Professor, além dos atributos e métodos comuns da interface Usuário, possui atributos e métodos específicos, criados e implementados na própria classe.

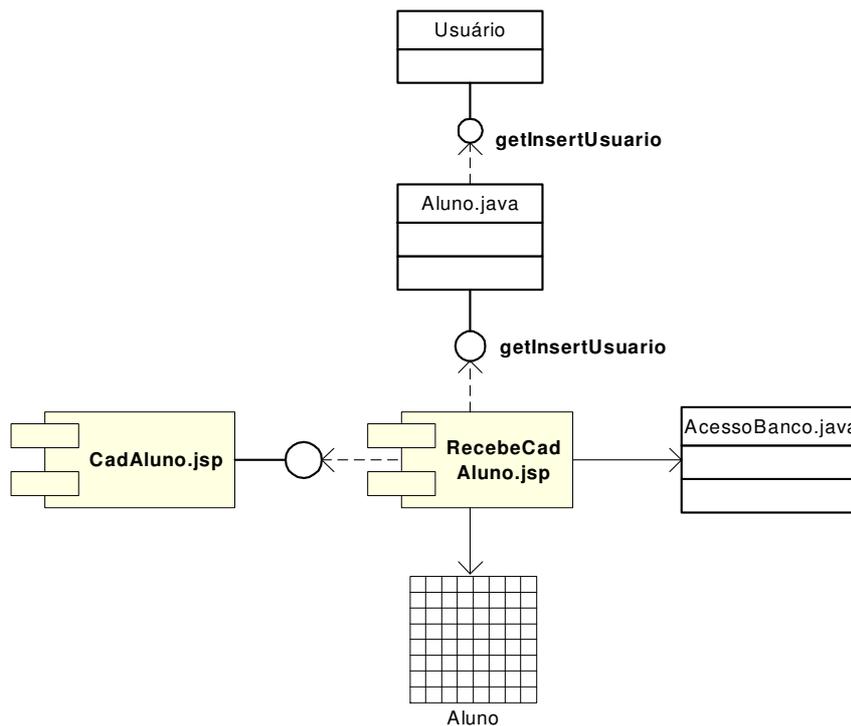


Figura 49 – Diagrama de componentes Criar Aluno

CRIAR ONTOLOGIA

Conforme visto anteriormente, para organizar os Componentes de Aprendizagem, de forma a facilitar a busca e possibilitar seu reuso, foram definidos dois tipos de Ontologia — Domínio e Assunto, que possuem atributos e métodos comuns e específicos. Por isso, foi definida a interface Ontologia, que descreve os atributos e métodos comuns a ambos. Os dois tipos deram origem a classes diferentes, que implementam os métodos definidos na interface Ontologia, adicionados aos métodos e atributos pertinentes a cada classe.

A Figura 50 apresenta o diagrama de componentes que especifica a construção dos programas desenvolvidos para a criação de um domínio. Para incluir dados de um novo domínio, o sistema, servindo-se do componente “CadDominio.jsp”, gera uma interface para o Administrador, que digita os dados solicitados. Esses dados são transferidos para o componente “RecebeCadDominio.jsp” que, pelo método “getInsert”, definido na interface Ontologia e implementado pela classe Domínio, insere uma nova linha na tabela Domínio, utilizando, para isso, a classe “AcessoBanco”. A criação de um objeto Assunto é feita de forma similar à de um objeto Domínio.

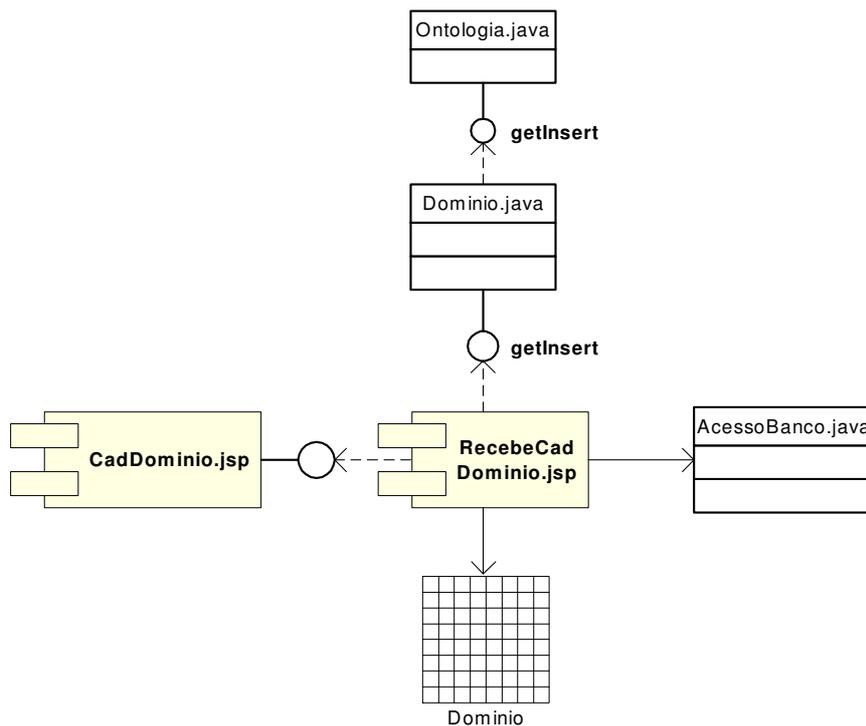


Figura 50 – Diagrama de componentes Criar Domínio

7.2.2 FUNÇÕES DO PROFESSOR

CRIAR COMPONENTES DE APRENDIZAGEM

Já foram definidos três tipos de componentes de aprendizagem: Curso, Aula e Material Didático; uma Aula é um conjunto de Materiais Didáticos e um Curso é um conjunto de Aulas. Esses três componentes foram implementados como classes diferentes, porém tiveram seus atributos e métodos de uso comum definidos em uma única interface, denominada Componente de Aprendizagem. Elas implementam os métodos definidos na interface, além dos atributos e métodos pertinentes a cada uma.

Na Figura 51 expõe-se o diagrama de componentes que especifica a construção dos programas desenvolvidos para a criação de um objeto do tipo Material Didático. Para incluir dados de um novo material, o sistema, usando o componente “CadMaterial.jsp”, gera uma interface para o Professor, que digita os dados solicitados. Esses dados são transferidos para o componente “RecebeCadMaterial.jsp” que utiliza o método “getInsert”, definido na interface Componente de Aprendizagem e implementado pela classe Material Didático, para inserir

uma nova linha na tabela Material Didático, utilizando a classe “AcessoBanco” para acessar o bando de dados.

A inclusão, no sistema, de um novo objeto do tipo Curso ou Aula é feita de modo semelhante à da inclusão de um novo Material Didático.

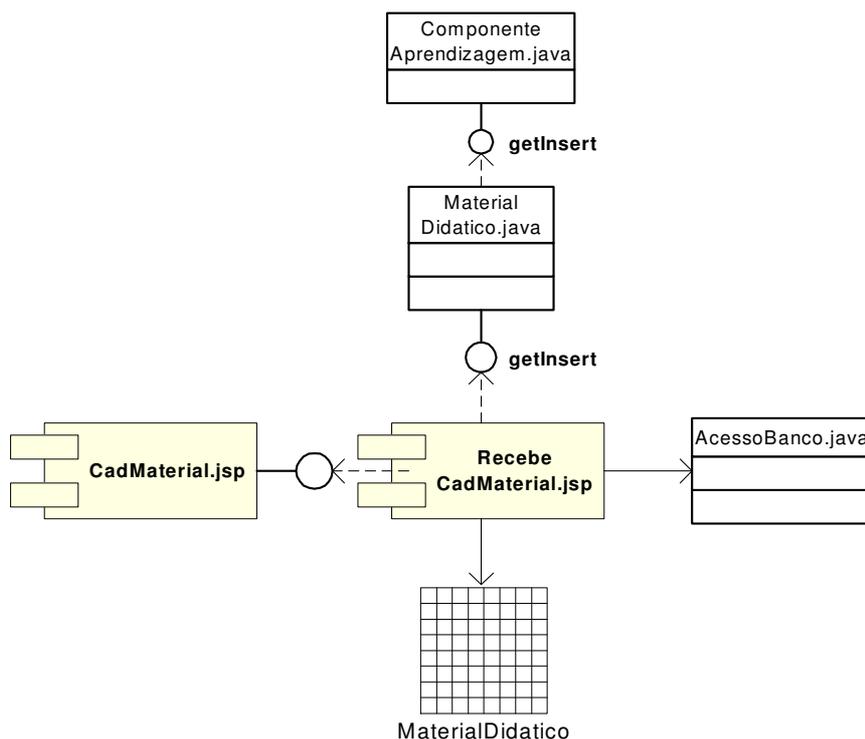


Figura 51 – Diagrama de componentes Criar Material Didático

ALOCAR MATERIAL EM AULA

Após a criação de uma Aula, o professor precisa alocar o material didático pertinente à aula criada e definir a ordem de apresentação desse material. A Figura 52 demonstra o procedimento para esse tipo de operação. Inicialmente, o sistema apresenta ao professor uma interface, por meio do componente “AlocarMaterialAulaAssunto.jsp”, onde ele seleciona um assunto. Esse assunto é recebido pelo componente “AlocarMaterialAulaAula.jsp”, que informa ao professor todas as Aulas cadastradas no sistema sobre o assunto indicado. O professor seleciona a aula que quer trabalhar. Essa informação é repassada ao componente “AlocarMaterialAulaMaterial.jsp”, que exibe todos os materiais existentes sobre o assunto. O

professor escolhe o material que quer alocar na aula recém-criada e, usando o componente “AlocarClassificarMaterialAula.jsp”, indica a posição do material na aula.

A função “Alocar Aula em Curso” é similar a função “Alocar Material em Aula”.

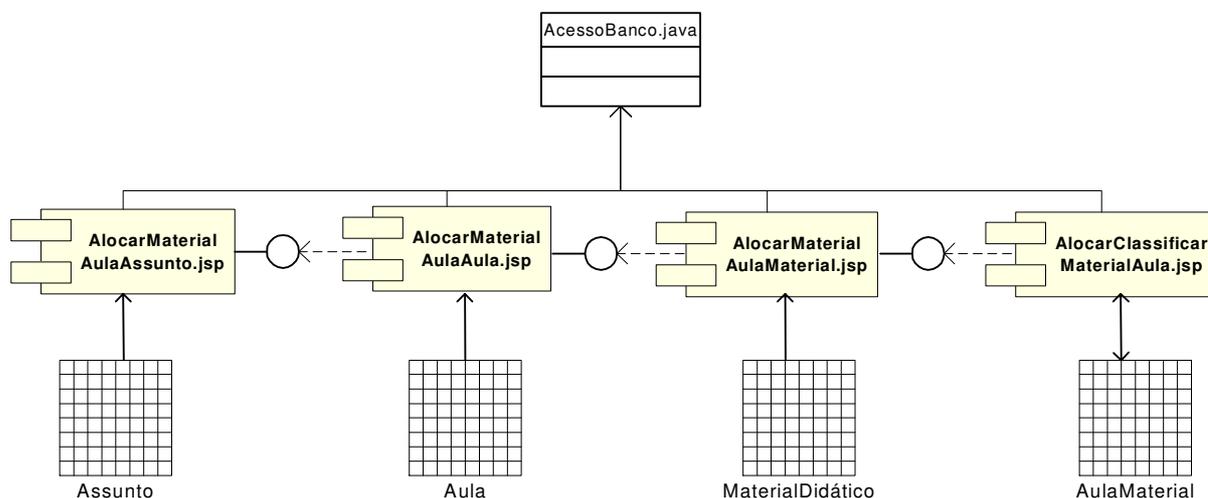


Figura 52 – Diagrama de componentes Alocar Material em Aula.

7.2.3 FUNÇÕES DO ALUNO

ASSISTIR A UM CURSO

A Figura 53 exibe o diagrama de componentes “Assistir Curso”, que especifica o comportamento do sistema quando um aluno o acessa, para assistir a um curso de um determinado domínio. Inicialmente, o componente “EscolherDominio.jsp” gera uma interface, onde o aluno pode selecionar o domínio a ser estudado. Este domínio é recebido pelo componente “EscolherCursoDominio.jsp”, que informa ao aluno todos os cursos pertencentes ao domínio selecionado. O aluno informa o curso escolhido, e a interface “EscolherAulaCurso.jsp” apresenta as aulas do curso. O aluno indica a aula que estudar e o sistema exibe os materiais didáticos existentes na aula selecionada. Finalmente, o aluno escolhe um material didático, que lhe é exibido pela ferramenta que deu origem a criação do material.

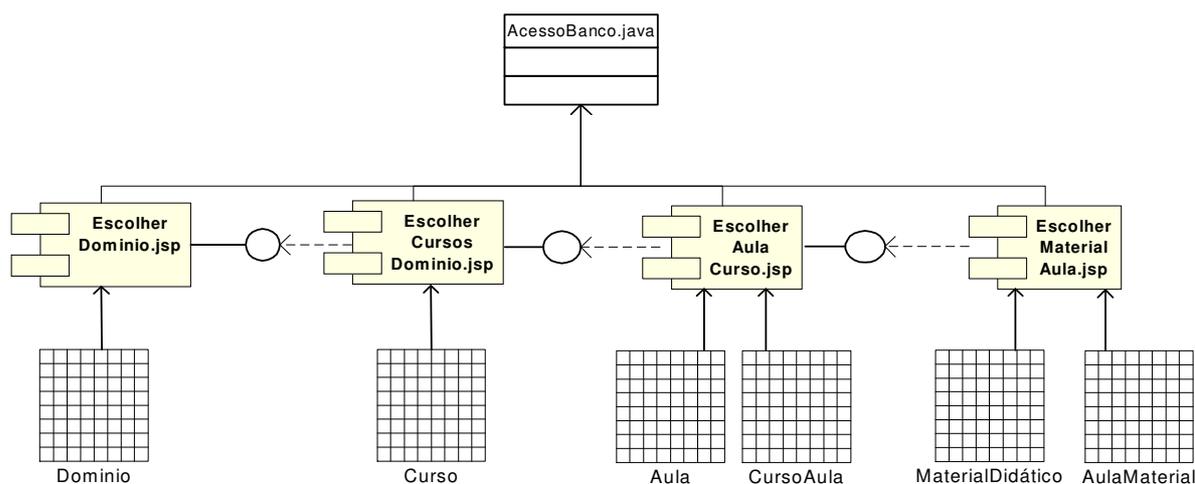


Figura 53 – Diagrama de componentes Assistir Curso.

7.3 O AMBIENTE SGC

A implementação de um protótipo do SGC objetiva demonstrar as principais interfaces de comunicação do usuário com o sistema. Para isso, selecionou-se uma parcela das funções definidas na modelagem. Visto que o controle de acesso ao sistema será desenvolvido em uma segunda versão e visando a simplificar a implementação, o componente Professor incorpora as funções do Professor e do Autor. A Figura 54 apresenta a interface principal do SGC onde podem ser observados os componentes Aluno, Professor e Administrador, conforme definidos na Camada de Apresentação da arquitetura do SGC (Figura 44).

Além do três componentes principais, o sistema oferece dois componentes auxiliares:

- **METODOLOGIA** – Esse componente, quando acionado, oferecerá ao usuário uma explanação do modelo desenvolvido neste trabalho e adotado na confecção do SGC.
- **DICAS GERAIS** – Esse componente, quando acionado, oferecerá ao usuário um texto explicativo de utilização do sistema.

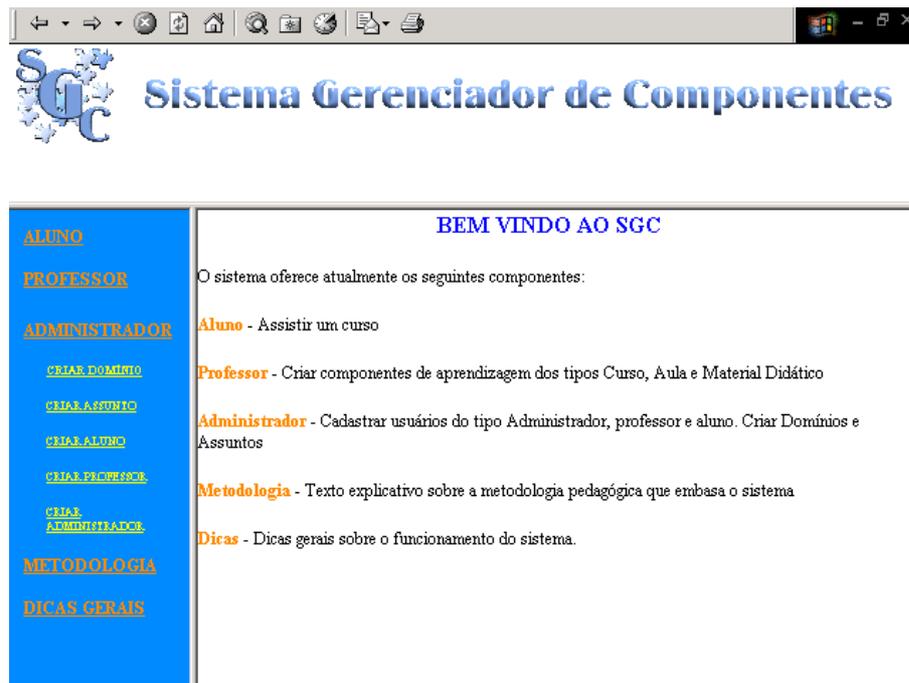


Figura 54 – Tela principal do SGC – Sistema Gerenciador de Componentes

7.3.1 FUNÇÕES DO ADMINISTRADOR

Foram implementadas, para o componente Administrador, as seguintes funcionalidades: Criar Domínio; Criar Assunto; Cadastrar Aluno; Cadastrar Administrador e Cadastrar Professor.

A figura 55 mostra a interface com o Administrador do sistema, na qual ele poderá cadastrar um novo domínio. O atributo Autor indica o responsável pela solicitação e é selecionado a partir da lista de professores cadastrados no sistema.



Figura 55 – Interface para criação de domínios

A Figura 56 apresenta a interface disponibilizada pelo sistema para a criação de um novo assunto. Um assunto faz parte de um domínio, o qual é indicado selecionando-se um item da lista dos domínios cadastrados no sistema.

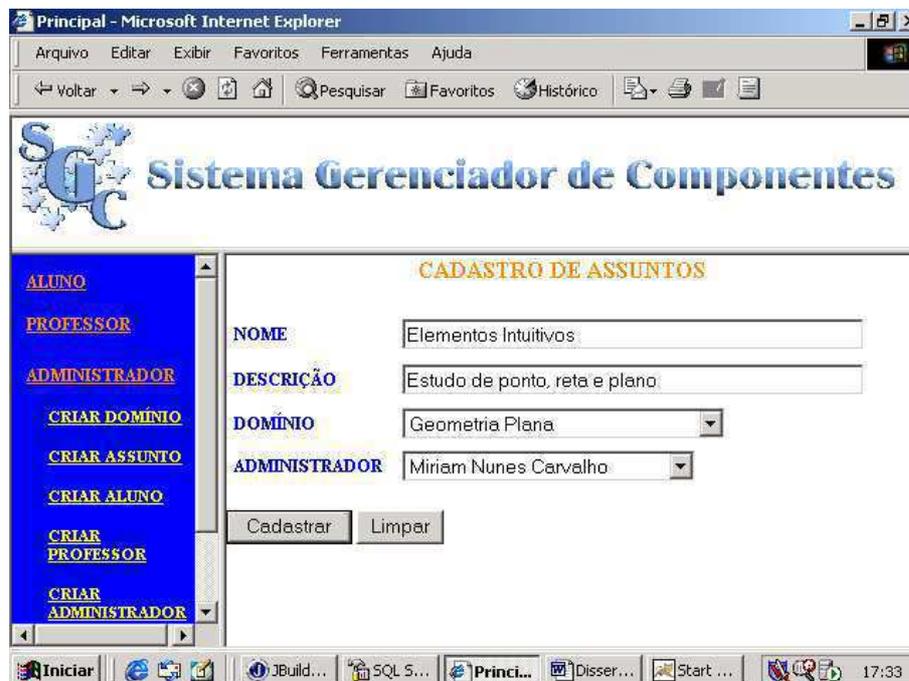


Figura 56 – Interface para criação de assuntos

A Figura 57 apresenta a interface, na qual o administrador poderá cadastrar um novo aluno. De modo semelhante poderá ser feito o cadastramento de administradores e professores.

The screenshot shows a web browser window with the title 'Sistema Gerenciador de Componentes'. The interface is divided into a blue sidebar on the left and a main content area on the right. The sidebar contains a menu with the following items: 'ALUNO', 'PROFESSOR', 'ADMINISTRADOR', 'CRIAR DOMÍNIO', 'CRIAR ASSUNTO', 'CRIAR ALUNO', 'CRIAR PROFESSOR', 'CRIAR ADMINISTRADOR', 'METODOLOGIA', and 'DICAS GERAIS'. The main content area is titled 'CADASTRO DE ALUNOS' and contains a form with the following fields: 'Nome' (text input with 'Luiza Nunes Carvalho'), 'Email' (text input with 'luiza@unitnet.com.br'), and 'Local' (dropdown menu with 'UNIT' selected). Below the form are two buttons: 'Cadastrar' and 'Limpar'.

Figura 57 – Interface para criação de alunos

7.3.2 FUNÇÕES DO PROFESSOR

Para o componente professor, foram implementadas as seguintes funcionalidades: Criar Material; Criar Curso; Criar Aula; Alocar Material Aula e Alocar Aula Curso.

A Figura 58 mostra a interface com o professor, na qual ele poderá criar um novo material didático. O material é classificado por assunto, selecionado pelo professor na relação de assuntos cadastrados no sistema, o que permite sua rápida recuperação, em caso de consulta, ou seu reuso por outro professor. O atributo “Disponível” oferece ao professor a opção de disponibilizar ou não o material para ser acessado pelo aluno. No atributo “ConteúdoMaterial”, o professor informa o “caminho” de acesso ao material, que pode ser um endereço local, ou seja, no HD do micro, em um servidor local, ou na Internet. O material será visualizado pelo aluno, por meio de software compatível com o tipo de arquivo no qual o material foi desenvolvido.



Figura 58 – Interface para criação de material didático

A Figura 59 apresenta a interface com o professor, na qual ele poderá criar uma nova aula. A função “Cadastrar Aula” visa a criar os dados cadastrais de uma aula e tem como resultado a inclusão de uma tupla no banco de dados, na tabela Aula.

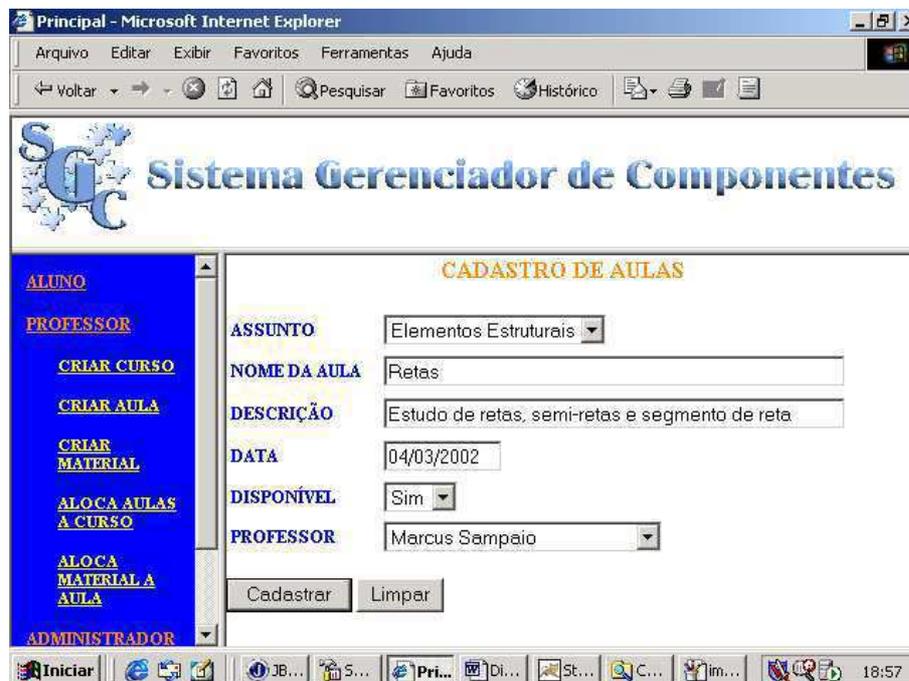


Figura 59 – Interface para criação de aulas

A figura 60 apresenta a interface com o professor, na qual ele poderá criar um novo curso. A função “Cadastrar Curso” visa a criar os dados cadastrais de um curso e tem como resultado a inclusão de uma tupla no banco de dados, na tabela Curso.

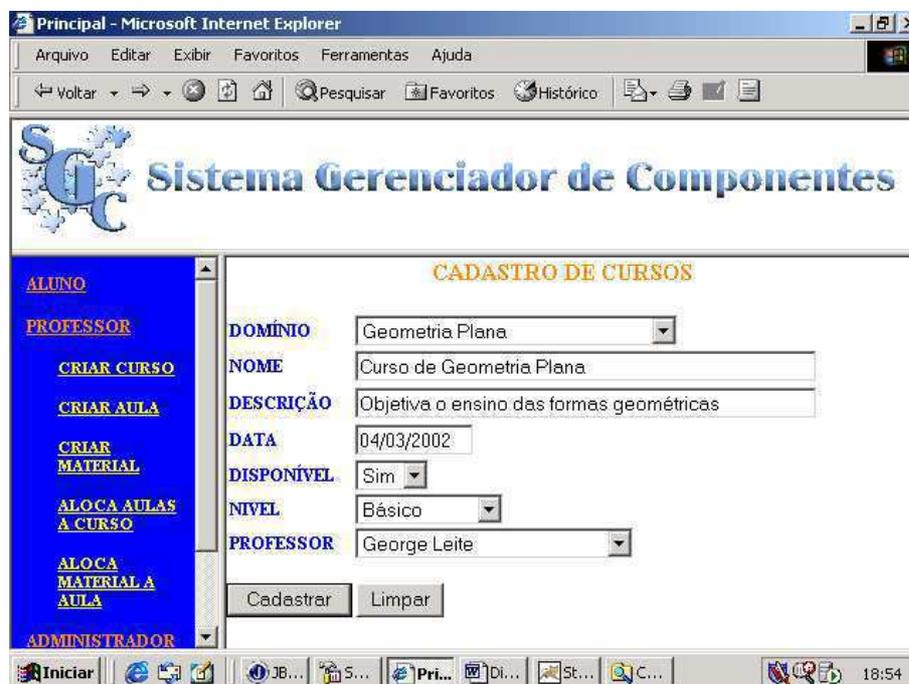


Figura 60 – Interface para criação de Cursos

Para alocar um ou mais materiais didáticos em uma aula, o sistema disponibiliza para o professor uma interface, onde ele, em uma primeira etapa, indica o assunto ao qual pertence a aula que deseja trabalhar (Figura 61). Na segunda etapa, o sistema disponibiliza todas as aulas pertencentes àquele assunto (Figura 62). Ao escolhê-la, o sistema informa todos os materiais existentes sobre o assunto em pauta (Figura 63) e o professor escolhe os que farão parte da aula em questão. Em seguida o professor indica a ordem em que o material será apresentado ao aluno (Figura 64). Essa função tem como resultado a inclusão, na tabela AulaMaterial, de tantas *tuplas* quantos forem os materiais selecionados.

O processo para a criação de um curso e alocação das aulas no curso é similar ao processo acima apresentado.



Figura 61 – Interface para alocar material em aula – Etapa 1



Figura 62 – Interface para alocar material a aula – Etapa 2



Figura 63 – Interface para alocar material em aula – Etapa 3



Figura 64 – Interface para organizar material em aula – Etapa 4

7.3.3 FUNÇÕES DO ALUNO

Não é objetivo deste trabalho oferecer material específico para cada aluno, e sim oferecer condições para que sistemas ou módulos que trabalham com controle de turmas e nível de conhecimento do aluno, a exemplo de um Sistema Tutor Inteligente (STI), assim o façam. Porém, foi implementada uma interface, na qual o aluno poderá selecionar e consultar o material cadastrado no sistema. Ele seleciona o domínio, o curso, a aula, e, por fim, o material que deseja consultar (Figuras 65 a 69c).

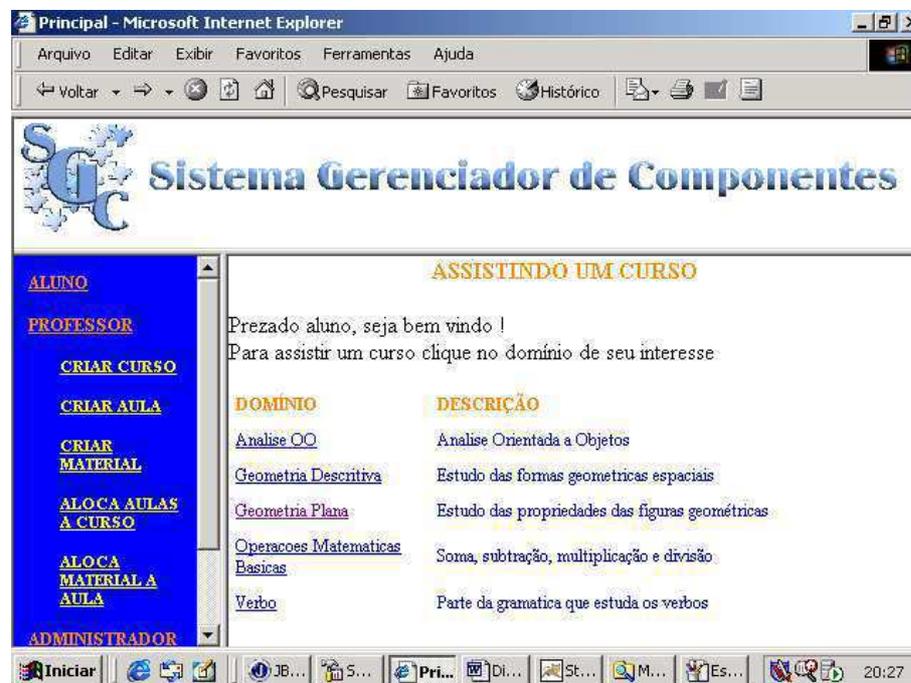


Figura 65 – Interface para o aluno escolher um domínio

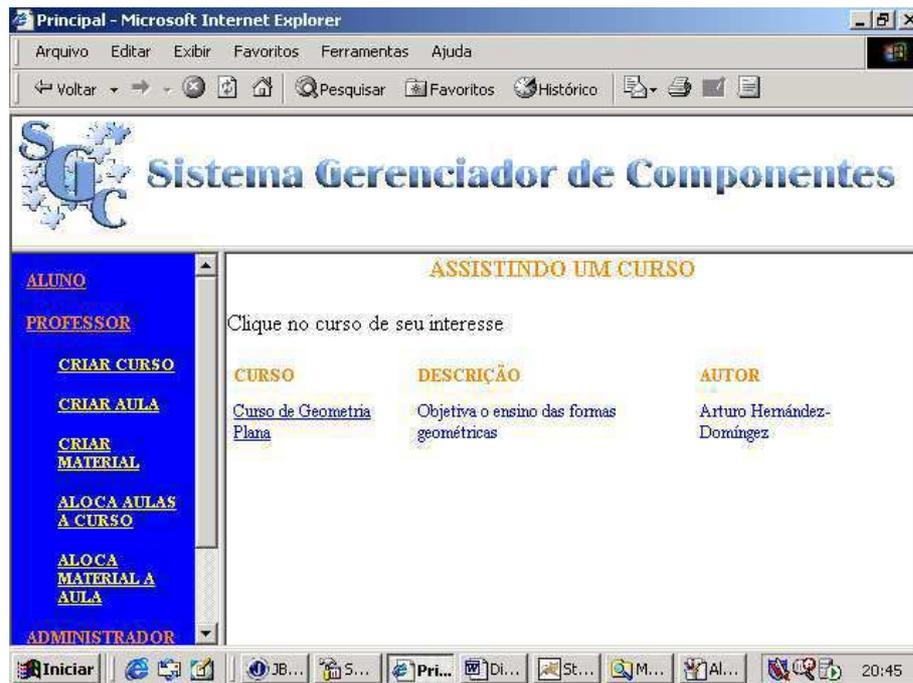


Figura 66 – Interface para o aluno escolher um curso

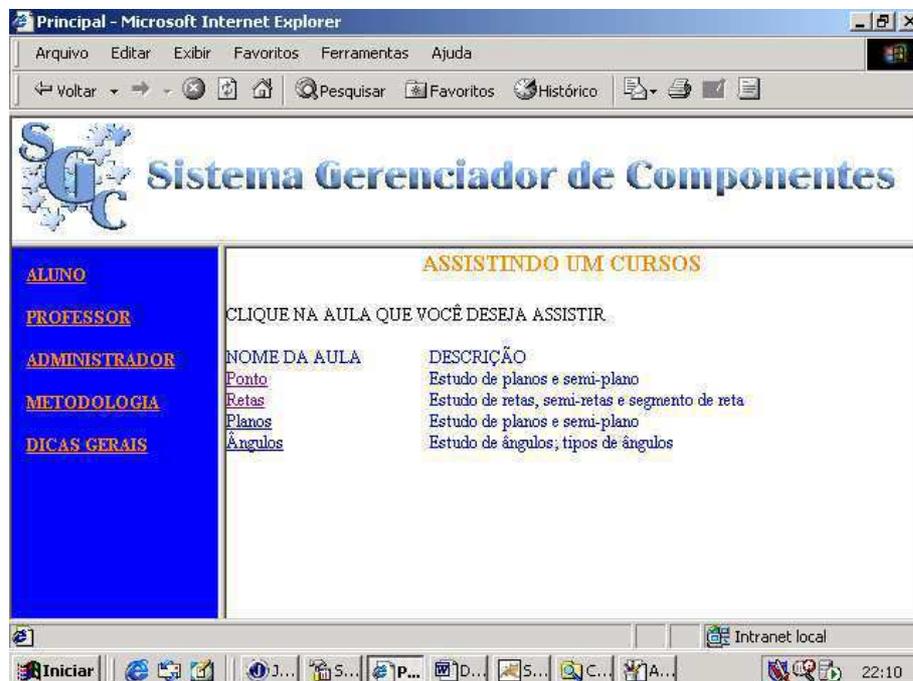


Figura 67 – Interface para o aluno escolher uma aula

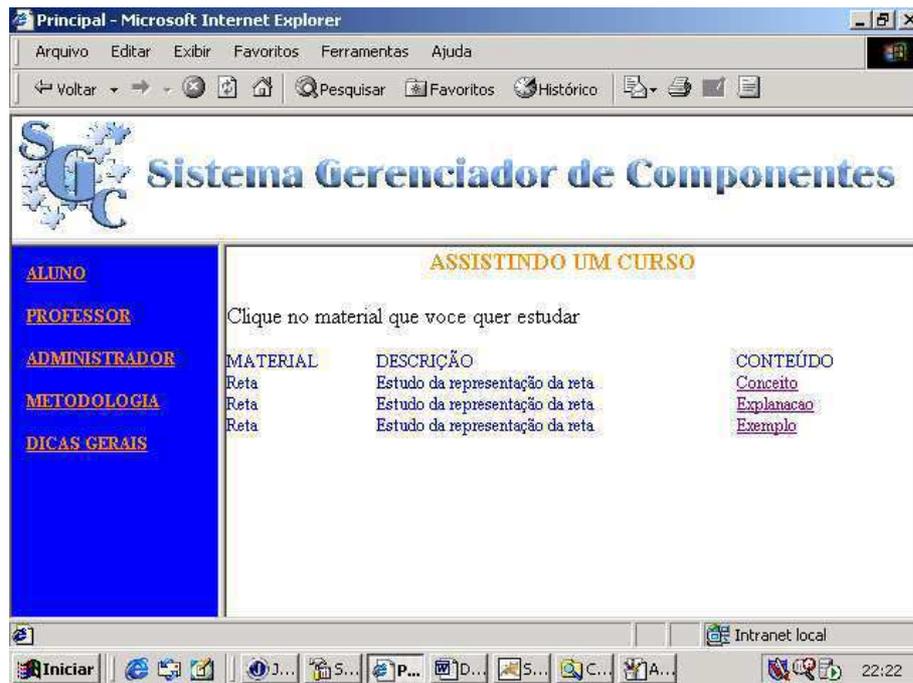


Figura 68 – Interface para o aluno escolher o material

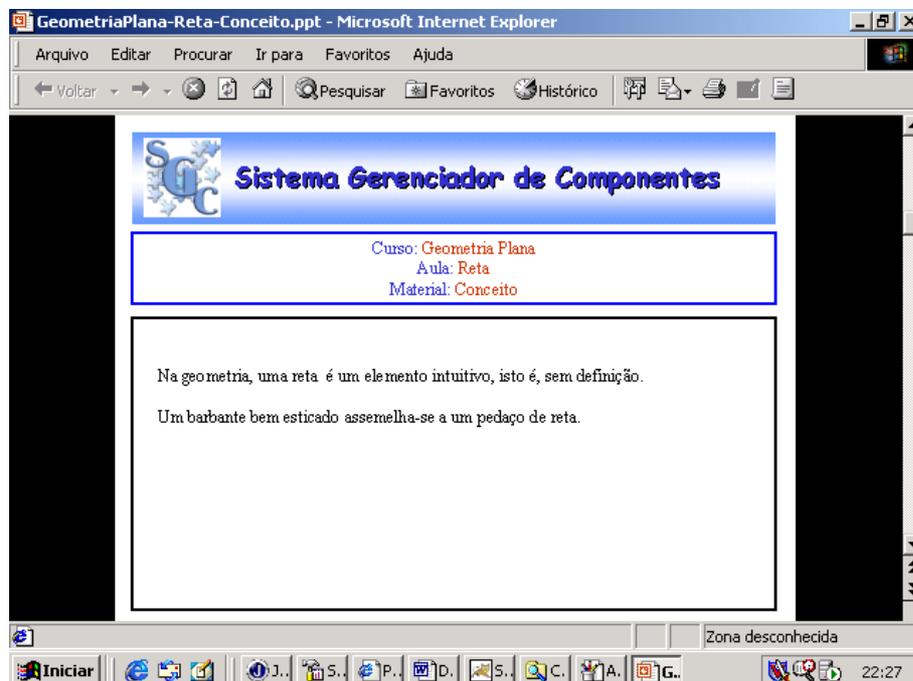


Figura 69a – Interface para exibir material para o aluno - conceito

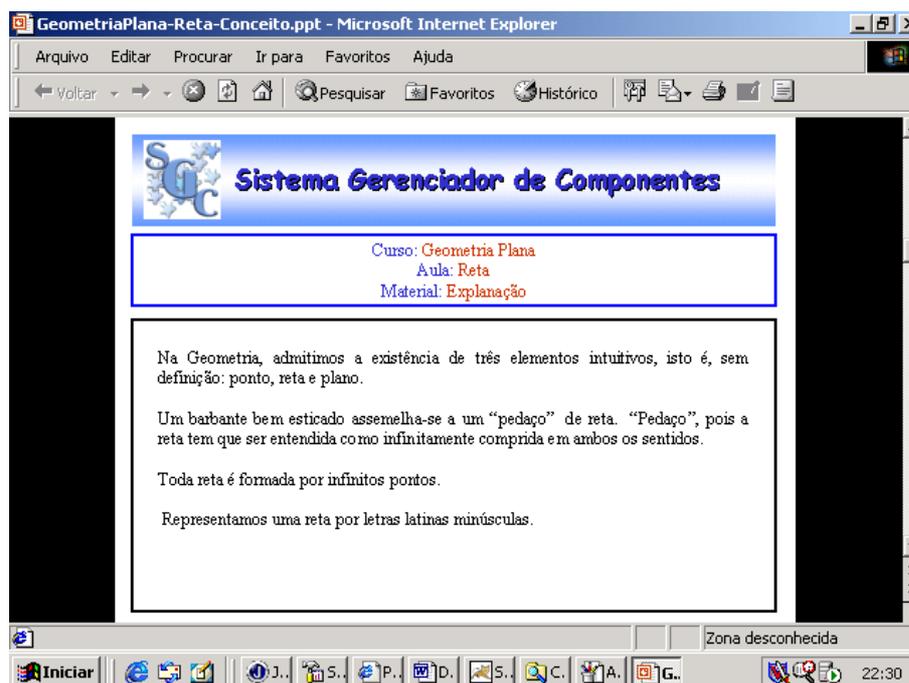


Figura 69b – Interface para exibir material para o aluno – Explicação

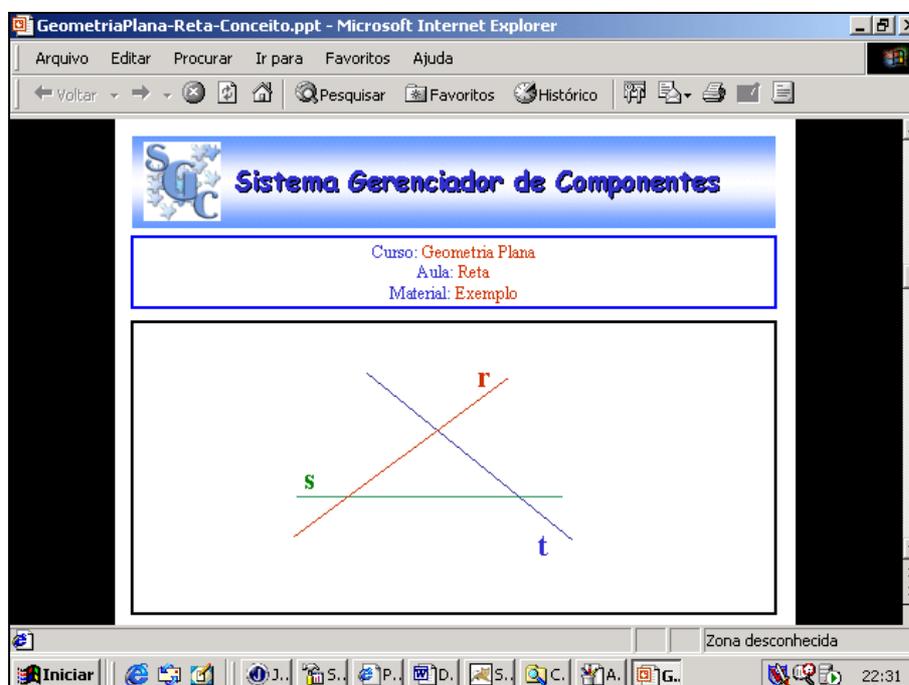


Figura 69c – Interface para exibir material para o aluno - exemplo

7.4 O SGC NO CONTEXTO DA ACVA

Na seção 2.5 apresentou-se o modelo proposto pela ACVA, cujo objetivo é desenvolver um ambiente de ensino e aprendizagem, destinado a grupos virtuais de estudantes, com suporte de um sistema de comunicação e controle por computador. Essa arquitetura foi dividida em quatro camadas, e cada uma pode ser considerada um subsistema. Conforme se observa na Figura 70, o SGC se integra ao projeto oferecendo uma solução para a camada de Serviços de Formação Básicos, mediante a criação de um modelo baseado em componentes para o desenvolvimento, organização e apresentação de material didático, e para a camada de Serviços de Suporte, valendo-se da apresentação de um modelo para armazenamento, em um banco de dados, do material desenvolvido pelo professor.

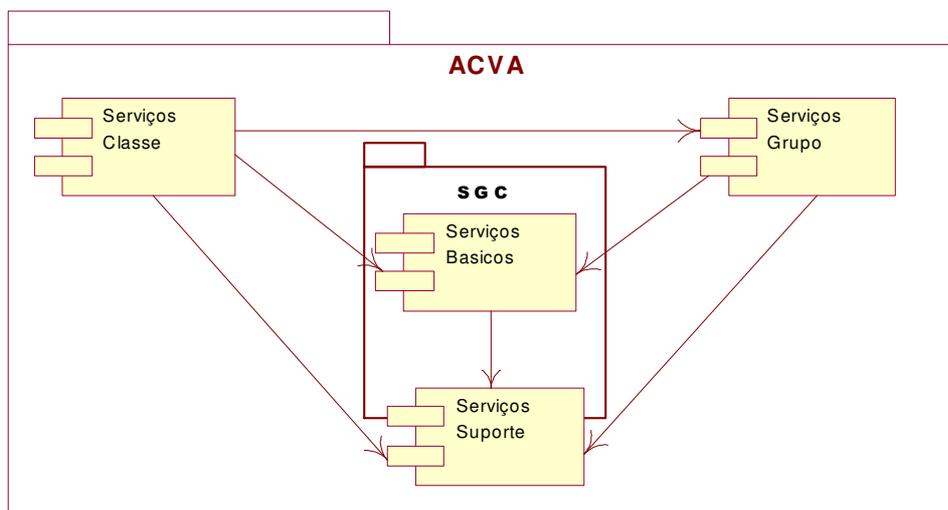


Figura 70 – Arquitetura SGC no contexto da ACVA.

A integração desses subsistemas proporcionará o funcionamento de um ambiente de ensino a distância, dotado de todas as funções propostas no projeto. O SGC fornece serviços às camadas de Serviços Grupo e Serviços Classe, usando as interfaces de seus componentes, como se mostrou na seção 5.5.

7.5 CONCLUSÃO

Este capítulo apresentou os aspectos de implementação, o protótipo do SGC e a integração do SGC, no contexto da ACVA.

É importante salientar que o SGC, da forma como foi projetado e implementado, tanto pode ser utilizado pela ACVA, fornecendo serviços às outras camadas dessa arquitetura, como pode, por meio de suas interfaces, oferecer serviços a qualquer sistema de ensino a distância. Além disto, a arquitetura do SGC (Figura 44) permite que seus componentes sejam utilizados e implementados, por qualquer outro sistema, não apenas de ensino a distância, conforme pode ser observado nos exemplos que seguem.

O componente Usuário (Figura 45), que é utilizado pelo SGC para manter o cadastro de professores, alunos e administradores do sistema, pode ser utilizado por qualquer outro sistema que necessite gerenciar informações sobre um grupo de pessoas como, por exemplo, um sistema de controle de vídeo locadora ou um sistema de controle de passagens aéreas.

O componente Ontologia (Figura 46), aqui utilizado para representar o conhecimento mediante a elaboração de um conjunto de conceitos existentes num certo domínio e a explicação das relações que podem surgir entre eles, pode ser utilizado, por exemplo, por um sistema de grade curricular.

O componente Componente de Aprendizagem (Figura 47) pode ser reutilizado por qualquer sistema de ensino informatizado, a distância ou presencial, a exemplo de um STI. Além disso, podem ser criadas novas classes que implementem a interface Componente de Aprendizagem como, por exemplo, Prova e Questão, onde uma prova de um determinado domínio é um conjunto de questões sobre determinados assuntos, pertencentes àquele domínio.

8 CONCLUSÃO

8.1 OBJETIVOS ALCANÇADOS

O principal objetivo deste trabalho foi a definir um modelo de organização de recursos didáticos associados a um domínio de conhecimento. O modelo desenvolvido é baseado em componentes e responde às questões apresentadas na introdução do trabalho, que nortearam a pesquisa. Foram alcançados os seguintes resultados:

- Pode-se organizar o material didático de forma clara e coerente, usando-se o conceito de mapas conceituais, o que evita a sobrecarga cognitiva e a desorientação do aprendiz;
- O uso de Ontologia, na definição do domínio e do assunto, proporciona um modelo de armazenagem organizado que facilita a pesquisa e o reuso do material instrucional;
- A divisão do material em partes pequenas, denominadas Componentes de Aprendizagem, estimula a criação e facilita o reuso;
- A definição dos Componentes de Aprendizagem em três níveis, Curso, Aula e Material Didático, proporciona a um professor ou a um sistema tutor organizar o material de forma individualizada, levando em consideração o nível de conhecimento do aprendiz.

O modelo desenvolvido, além de atender à proposta inicial do trabalho, apresentou resultados adicionais, a saber:

- A arquitetura do SGC, baseada em componentes, e a clara definição de interfaces permitirão seu uso por qualquer ambiente de ensino a distância, na preparação dos recursos didáticos.

- Os componentes, da forma como foram definidos e implementados, podem ser utilizados de forma independente por outros sistemas, conforme exemplificado na conclusão do capítulo 7.

8.2 CONTRIBUIÇÕES

As principais contribuições deste trabalho são:

- A definição de um modelo para organização de material didático utilizando, de forma associada, os conceitos de ontologia, mapas conceituais e componentes;
- A definição de um ambiente para proporcionar a criação de material didático com facilidades de criação, organização, identificação e reuso desse material;
- A possibilidade de integração do SGC com a ACVA, oferecendo, dessa forma, material instrucional para ser utilizado pelos outros módulos do projeto;
- A implementação do Sistema, baseada em componentes, que permite a reutilização por outros sistemas, tanto do material instrucional propriamente dito como das classes e interfaces definidas no SGC.

8.3 LIMITAÇÕES

Apesar de ter alcançado os objetivos propostos e oferecido contribuições para o processo de ensino/aprendizagem, em especial para o Ensino a Distância, este trabalho apresenta as seguintes limitações:

- A respeito da liberdade de escolha do *software* a ser utilizado na criação do material didático: embora essa decisão facilite o trabalho do professor, poderá criar para o aluno problemas de acesso ao material, já que ele precisará ter, se não o mesmo, algum *software* compatível com o tipo de arquivo no qual o material didático foi gerado.
- Sobre a documentação das classes e interfaces definidas no sistema: para que possam ser reutilizadas, precisam apresentar uma definição clara do que podem oferecer ao programador e de como devem ser implementadas, sem que seja necessário um estudo do

código-fonte. Por se tratar de um protótipo e por questões de tempo, não foi feito um trabalho nesse sentido.

8.4 TRABALHOS FUTUROS

Durante a realização da modelagem e da implementação do SGC, foram sendo identificadas atividades para trabalhos futuros:

- Concluir o SGC com todas as funções modeladas no projeto, inclusive as funcionalidades Metodologia e Dicas, que fazem parte da tela principal do sistema;
- Preparar documentação clara das classes e interfaces do sistema, de forma que elas possam ser reusadas por outros sistemas;
- Interagir com os alunos que estão desenvolvendo outros módulos do projeto ACVA, para que ele possa funcionar integralmente.

9 REFERÊNCIAS BIBLIOGRÁFICAS

AULANET. **Ambiente de Desenvolvimento de Cursos via Web**. Rio de Janeiro, PUC-Rio-LES - Laboratório de Engenharia de Software. Apresenta ferramenta para gerenciamento de ensino a distância. Disponível em: <<http://www.eduweb.com.br/scripts/aulanet>>. Acesso em: 10 fev. 2001.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: Guia do Usuário**. Tradução: Fábio Freitas da Silva. Rio de Janeiro: Campus, 2000. Título original: The Unified Modeling Language User Guide.

CABRAL, Anderson R. Yanzer. **Uma Proposta metodológica para Autoria em WBT**, 2001. 112 f. Dissertação (Mestrado em Ciência da Computação) – Escola da Ciência da Computação, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre.

D’SOUZA, Desmond Francis; WILLS, Alan Cameron. **Objects, Components, and Frameworks with ULM: The Catalysis Approach**. USA: Addison-Wesley, 1998.

D’SOUZA, Desmond Francis. **Objects, Components, Frameworks: Methods, Consulting, Training – “Looking for a few good minds!”**. Disponível em <<http://www.iconcomp.com>>. Acesso em: 23 set. 2001

GUARINO, Nicola. **Formal Ontology and Information Systems**. In FOIS’98: INTERNATIONAL CONFERENCE ON FORMAL ONTOLOGY IN INFORMATION SYSTEMS, 1998, Trento. Proceedings...Amsterdam: IOS Press, 1998. p.3-15.

HERNÁNDEZ-DOMÍNGUEZ, Arturo. **Systemes “Tuteur intelligent” et “Collecticiel(Groupware)”**: L’A.R.E.S.F.E.D. Architecture pour la reutilisation et l’exploitation des services de formation dans le contexte de l’Education à Distancia, 1995. Tese (Doutorado) - Doctorat de l’Université Paul Sabatier-Toulouse, França.

HERNÁNDEZ-DOMÍNGUEZ, Arturo. **Specification and implementation of an adaptable virtual class**. In: ED-MEDIA 97: WORLD CONFERENCE ON EDUCATION MULTIMEDIA AND HYPERMEDIA ON EDUCATIONAL TELECOMMUNICATIONS, 1997, Calgary, Canada. **Proceedings...** [S.l.]:[s.n.], 1997.

JSPBRASIL. – Tecnologia para desenvolvimento de aplicações WEB. Apresenta tutorial do *JSP-Java Server Pages*. Disponível em: <<http://jspbrasil.com.br:8081/jspbrasil/main/main.jsp>>. Acesso em: 02 abr. 2002.

LANDIM, Cláudia. **Educação a Distância**: Algumas Considerações. Disponível em: <<http://www.cciencia.ufrj.br/educnet/conceitd.htm>>. Acesso em: 31 mar. 2001.

LARMAN, Graig. **Utilizando UML e Padrões**: Uma Introdução à Análise e ao Projeto Orientados a Objetos. Tradução: Luiz Augusto Meirelles Salgado. Porto Alegre: Bookman, 2000. Título Original: *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design*.

LEARNING SPACE. **Web-based e-learning solution**. *Lotus Development Corporation*. Apresenta solução de ensino a distância. Disponível em: <<http://www.lotus.com/home.nsf/welcome/learnspace>>. Acesso em: 15 mar. 2001.

LEITE, Aury de Sá; OMAR, Nizam. **Representação de Conhecimento Pedagógico e Didático em Sistemas Educativos Inteligentes**. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 10., 1999, Curitiba. *Anais...* Curitiba: UFPR, 1999. p.153-160.

LOBO NETO, Francisco José da Silveira. **Educação a Distância**: Regulamentação, Condições de Êxito e Perspectivas. Disponível em: http://www.intelecto.net/ead_textos/lobo1.htm>. Acesso em 15 mar. 2002.

LUCENA, Carlos; FUCK, Hugo. **Professores e aprendizes na Web**: A educação na era da Internet. Rio de Janeiro: Ed. Clube do Futuro, 2000.

LUDWIG, Cláudio Michaelson; BORGES, Karen Selback. **Autoria e Navegação de Hiperdocumentos Educacionais e Utilização de Mapas Conceituais**. TISE97-Taller Internacional de Software Educativo, Santiago do Chile, dez.1997. Disponível em: <www.c5.cl/ieinvestiga/actas/tise97/trabajos/trabajo>. Acesso em: 27 jun. 2001.

MAGALHÃES, Léo Pin (Org.). **PROJETO SAPIENS – Sistema de Apoio à Aprendizagem**. São Paulo: Relatório Final de Atividades – UNICAMP (Processo nº 97/12807-1), 2001. Disponível em: <<http://www.dca.fee.unicamp.br/projects/sapiens/Reports/rf2000/rf2000.html>>. Acesso em: jul. 2001.

MARIETTO, Maria das G. Bruno; OMAR, Nizam. **Geração Dinâmica de Objetivos Instrucionais em Sistemas de Tutoria Inteligente**. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 10., 1999, Curitiba. *Anais...* Curitiba: UFPR, 1999. p.130-136.

MARIETTO, Maria das G. Bruno; OMAR, Nizam. **Uma metodologia para Estruturação de Sistemas de Tutoria Inteligente**. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 10., 1999, Curitiba. *Anais...* Curitiba: UFPR, 1999. p.121-129.

MENEZES, C. Silva et al. **Formação de Recursos Humanos em Informática Educativa**: Uma experiência com Educação à Distância. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 11., 2000, Maceió. *Anais...*Maceió: UFAL, 2000. p.175-182.

NOVAK, J.D. **Learning, Creating and Using Knowledge: Concept Map as Facilitative Tools in School and corporations.** Lawrence Erlbaum Associates. New Jersey, 1998.

NOVAK, J.D. **The Theory Underlying Concept Maps And How To Construct Them.** Disponível em: <<http://cmap.coginst.uwf.edu/info/index.html>>. Acesso em: 22 mai. 2001.

PLOTNICK, Eric. **Concept Mapping: A Graphical System for Understanding the Relationship Between Concepts,**ERIC Digests, New York, jun. 1997. Disponível em:<<http://ericir.syr.edu/ithome/digests.mapping.html>>. Acesso em: 30 jun. 2001.

REZENDE, Luiziana at al. **RCC: Auxiliando Professores na Tarefa Metacognitiva de Estruturação do Conhecimento.** In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 11., 2000, Maceió. **Anais...** Maceió: UFAL, 2000. p.251-258.

ROCHA, A.Regina at al. **A Construção de um Meta-Ambiente Educacional para Cardiologia.** In: CONGRESSO INTERNACIONAL DE INGENIERIA INFORMÁTICA, 6., 2000, Buenos Aires. Disponível em:<<http://www.cos.ufrj.br/~cardioed/paginas/meta-amb.htm>>. Acesso em: 18 out. 2001.

SANTOS, Neide at al. **Sistemas de Autoria para Cursos a Distância Apoiados em Tecnologias da Internet: Diretrizes para Seleção,** 1988. Disponível em: <http://www.cos.ufrj.br/~neide/artigos/analise_de_sistemas.htm>. Acesso em: 18 jul. 2001.

SANTOS, Neide at al. **Uma Taxonomia para o Domínio da Educação mediada pela Internet.** In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 12., 2001, Vitória. **Anais...** Vitória: SBIE2001, 2001. p.427-436.

SILVA, A.S., HERNÁNDEZ-DOMÍNGUEZ, A. **Desenvolvimento de um Sistema Tutor Inteligente baseado em Agentes no contexto de uma Aula Virtual Adaptada.** In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 12., 2001, Vitória. **Anais...** Vitória: SBIE2001, 2001. p.427-436.

SILVA, Aleksandra do Socorro. **TUTA – Um Tutor Baseado em Agentes no Contexto do Ensino a Distância,** 2000. 141f. Dissertação (Mestrado em Informática) – COPIN – Curso de Pós-graduação em Informática, Universidade Federal da Paraíba – Campus II –UFBP, Campina Grande.

SOUZA, Renato Rocha. **Usando Mapas Conceituais na Educação Informatizada Rumo a um Aprendizado Significativo.** Disponível em: <<http://www.edutecnet.com.br/Textos/Alia/MISC/edrenato.htm>>. Acesso em: 27 jun. 2001.

TAROUCO, Liane M. Rockenbach et al. **Construção Colaborativa de Mapas Conceituais: Similaridade Ideológica.** In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 11., 2000, Maceió. **Anais...** Maceió: UFAL, 2000. p.105-111.

TAVARES, O. Lira et al. **Ambiente de apoio à mediação da aprendizagem: Uma abordagem orientada por processos e projetos.** In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 11., 2000, Maceió. **Anais...** Maceió: UFAL, 2000. p.48-55.

TOMCAT – Tecnologia para execução de aplicações para web. Fundação Apache. Apresenta Tutorial do Tomcat. Disponível em: <<http://jacarta.apache.org/tomcat> >. Acesso em: 02 abr. 2002.

TOPCLASS. Gerenciador de ensino a distância. São Caetano do Sul, *MicroPower*. Apresenta ferramentas para gerenciamento de ensino a distância. Disponível em <http://www.micropower.com.br/elearning/topclass/index.asp> >. Acesso em: 15 fev. 2001.

WEBCT. WebCT na Universidade de São Paulo. São Paulo, Universidade de São Paulo. Apresenta tutorial do *WebCT*- Programa para criação de ambientes educacionais, desenvolvido pela *University of British Columbia*. Disponível em: <<http://jassy.redealuno.usp.br:8900/about.html>>. Acesso em: 15 mar. 2001.

ANEXO I – DESENVOLVIMENTO BASEADO EM COMPONENTES: O MÉTODO CATALYSIS

UML (Unified Modeling Language) é uma linguagem gráfica, padronizada pela *OMG (Object Modeling Group)*, bastante utilizada na elaboração de projetos de software. Pode ser empregada para a visualização, a especificação, a construção e a documentação de artefatos que façam uso de sistemas complexos de *software* (BOOCH, 2000). É uma linguagem muito expressiva, que abrange todas as visões necessárias ao desenvolvimento e implantação de um sistema. Independe de processos, podendo ser perfeitamente utilizada em processo orientado a casos de usos, centrado na arquitetura, interativo e incremental. Porém, a *UML* é apenas uma notação e, portanto, é somente uma parte de um método para desenvolvimento de *software* (BOOCH, 2000). Faz-se necessária, então, a escolha de uma metodologia que defina as etapas a seguir para a modelagem de um sistema.

Catalysis é uma metodologia que utiliza padrões, para o desenvolvimento sistemático de sistemas, baseada em objetos e componentes (D’SOUZA, 1998). Desenvolvida, originalmente por Desmond D’Souza e Alan Wills, usa notações com base no padrão *UML* e oferece um método completo, que possibilita especificar, documentar, implementar e testar um sistema desde a especificação do negócio até o código-fonte. Nessa metodologia, os processos de modelagem e projeto de um sistema são recursivos (Figura 71) ao longo da definição do modelo do negócio ou domínio, da especificação e da implementação dos componentes. Cada ciclo inclui uma revisão de resultados e riscos, levando a um refinamento das metas e planos para o próximo ciclo. A depender da complexidade do sistema, é possível seguir rotas diferentes e adotar uma sucessão de modelos e tarefas que se adaptem melhor as características do projeto. Os conceitos de tipo, modelos, colaborações, refinamento, *framework* e padrões podem ser aplicados a todos os níveis do processo (D’SOUZA, 1998).

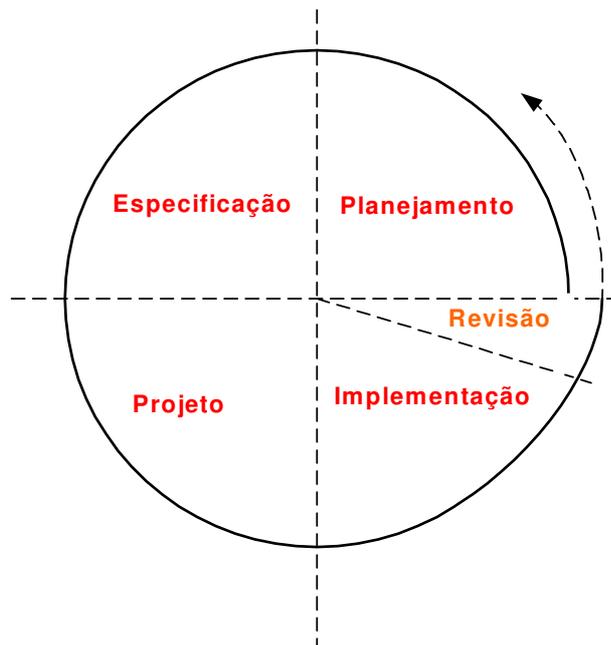


Figura 71 – *Catalysis* - Modelo de trabalho em espiral ²⁴

A metodologia *Catalysis* se baseia em três princípios:

- Abstração – A metodologia abstrai a complexidade do sistema no nível de código. Um exemplo de abstração comum é uma interface, que esconde informações sobre a implementação.
- Precisão – Abstração e precisão trabalham em níveis diferentes de detalhe. Com o uso de ferramentas apropriadas, as abstrações são representações precisas de eventuais implementações.
- Partes plugáveis – Todo o trabalho de desenvolvimento deve ser feito mediante a junção de partes existentes, sejam elas código de implementação, especificação de requisitos ou projetos padrões.

25. Fonte:(D'SOUZA, 1998, p.513)

Catalysis oferece um roteiro básico para modelar um sistema de negócios típico, do modelo até o código, usando a notação *UML*, e sistematizando-a de modo a estabelecer um relacionamento entre os documentos e uma semântica clara para os modelos (D'SOUZA, 1998). Esse roteiro pode ser adaptado, isto é, modelos de tipo, colaboração, refinamento, *frameworks* e padrões podem ser aplicados ao longo dos vários níveis do projeto, e podem ser ou não selecionados para a representação do problema, a depender da complexidade do negócio que se está modelando e do que se deseja representar. A figura 72 apresenta a consolidação desse roteiro, com as principais atividades de um sistema de negócios típico.

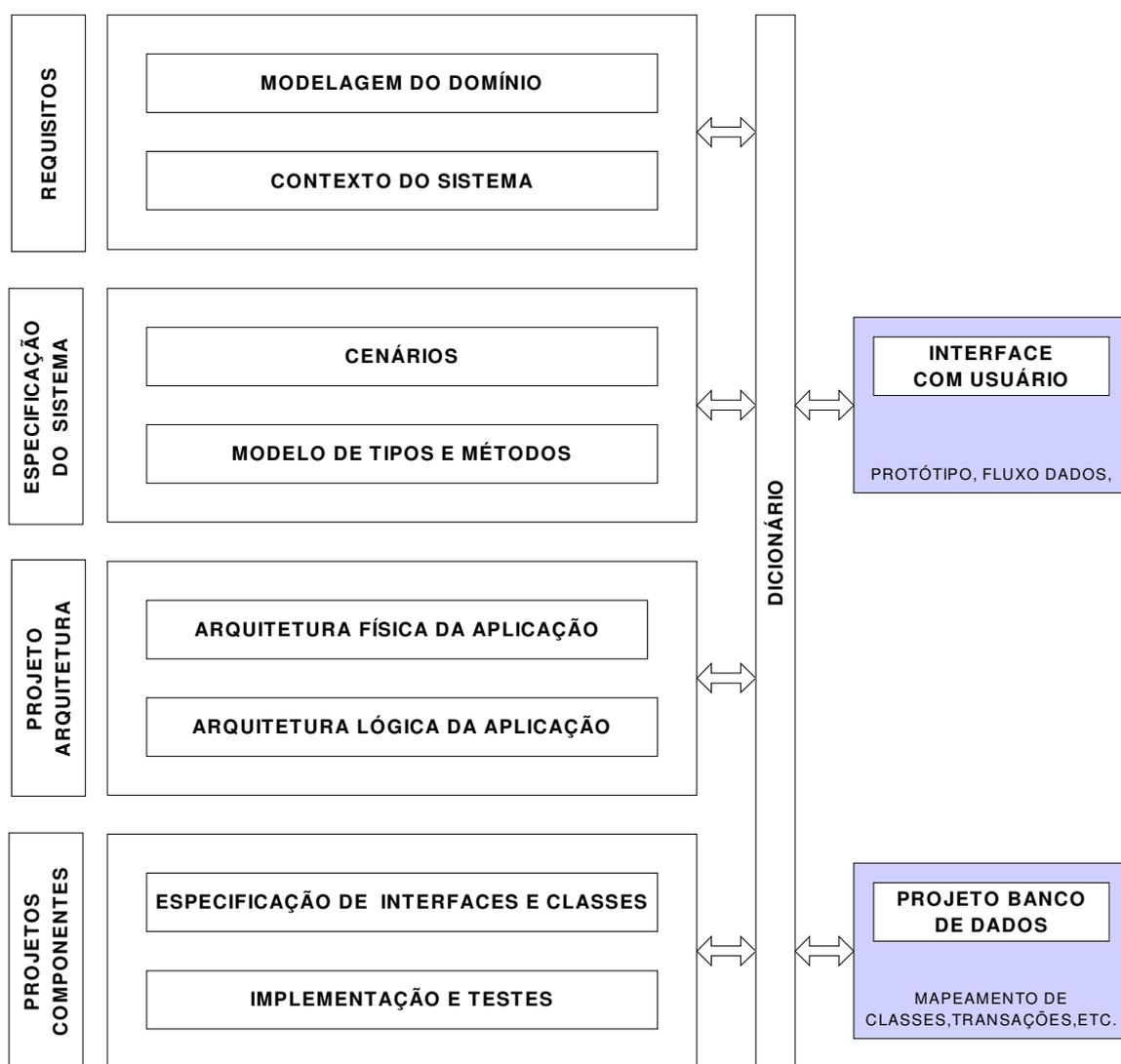


Figura 72 – *Catalysis* - Principais atividades para um típico sistema de negócios²⁵

26. fonte:(id. Ibid., p. 517)

As atividades de requisito são necessárias para a contextualização e o entendimento do problema. Nesse nível são usados os modelos do domínio, que devem refletir a visão do cliente em relação ao domínio do problema, e não a visão do programador. Por meio de modelos que expressem o contexto do sistema, como mapas conceituais e dicionários, é feito um mapeamento entre a visão interna do sistema e uma visão que seja mais natural ao cliente.

Os cenários descrevem a especificação, passo a passo, e podem ser usados desde o nível da especificação do negócio até as especificações das ações de implementação.

Nas especificações do sistema são definidos os tipos com seus atributos e associações. Definir um tipo envolve identificar cada ação de que o sistema participa. O comportamento pode ser descrito por quadros de estados e cenários. Sistemas grandes podem ser estruturados em pacotes, para facilitar o trabalho de análise

A implementação interna do sistema (*Architectural Design*) é dividida em duas partes: arquitetura de aplicação e arquitetura do banco de dados. A primeira compõe a estrutura em pacotes e colaborações, implementando o negócio como uma coleção de colaborações de componentes. Usa a arquitetura técnica, isto é, a arquitetura e plataforma do *hardware*, que dá cobertura a todas as partes do sistema: plataforma de *hardware* e *software*, banco de dados e a escolha da arquitetura do componente. Inclui ainda regras de projeto e padrões. A arquitetura do banco de dados também deve ser pensada nesse estágio, incluindo o projeto do modelo, as dependências e as restrições.

A meta da implementação de componentes é definir uma estrutura interna e interações que satisfaçam a necessidade de comportamento, tecnologia e funcionamento exigidos para o componente. Em um sistema simples, cada tipo identificado no modelo será implementado como uma classe. Em componentes grandes, eles poderão ser divididos em classes, com responsabilidades para cada classe e com definição de diagramas de interação, de forma a realizar o comportamento especificado. Então, será desenhado o modelo de classes, que possui as classes que compõem o sistema, as interfaces que elas implementam e usam, seus atributos e operações e as referências entre elas.

ANEXO II – DIAGRAMAS DE SEQÜÊNCIA

Diagramas de seqüência e narrativas que não foram implementados no contexto do protótipo do SGC.

CONTEXTO – MANTER CADASTRO DE USUÁRIOS

CENÁRIO – ALTERAR USUÁRIO

O administrador efetua *login* e informa o nome do usuário. O sistema exibe os dados solicitados. O administrador os altera e o sistema efetua a alteração (Figura 73).

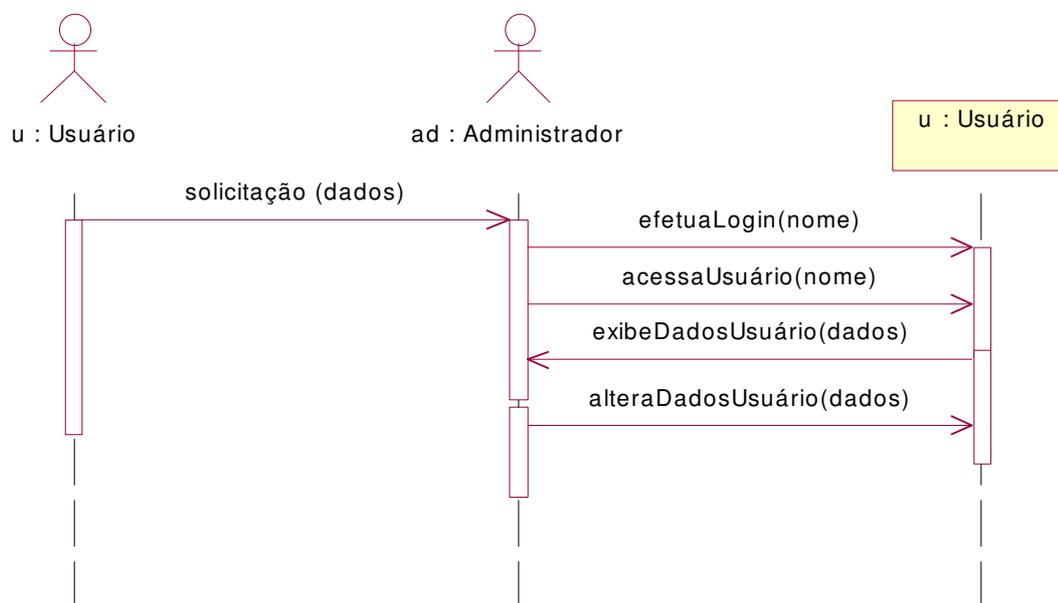


Figura 73 – Diagrama de Seqüência – Alterar Usuário

CENÁRIO – EXCLUIR USUÁRIO

Contexto – O administrador efetua *login* e informa o nome do usuário. O sistema exibe os dados solicitados. O administrador confirma a exclusão e o sistema realiza a operação (Figura 74).

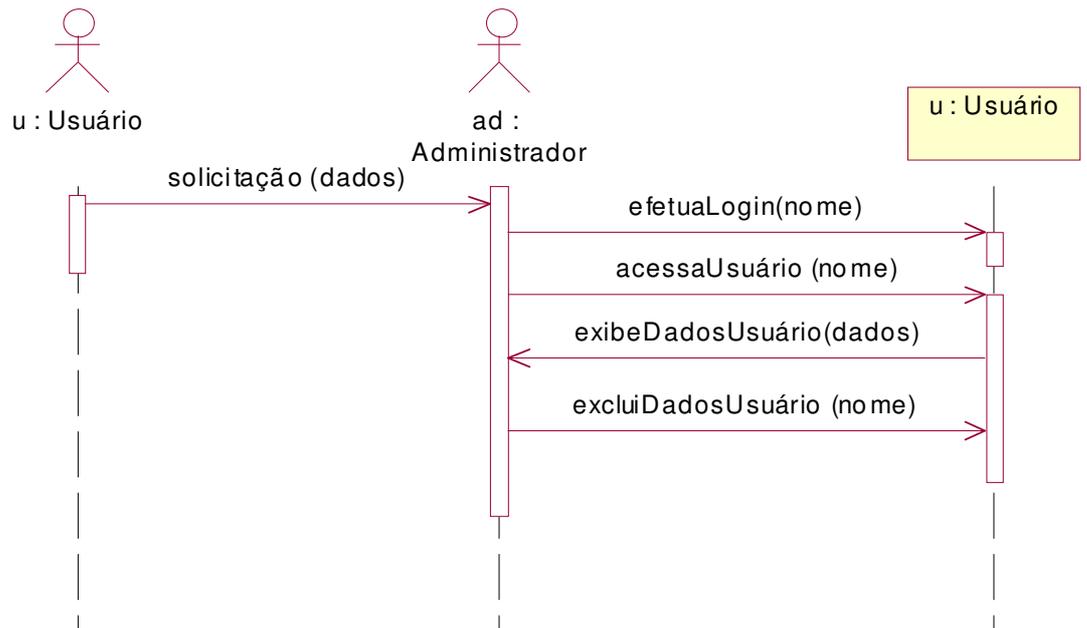


Figura 74 – Diagrama de Seqüência – Excluir Usuário

CONTEXTO – MANTER MATERIAL DIDÁTICO

CENÁRIO – ALTERAR MATERIAL DIDÁTICO

Um autor acessa o sistema para alterar um material didático; informa nome e senha e indica o material a ser alterado. O sistema expõe os dados do material. O autor os modifica e o sistema efetua a alteração (Figura 75).

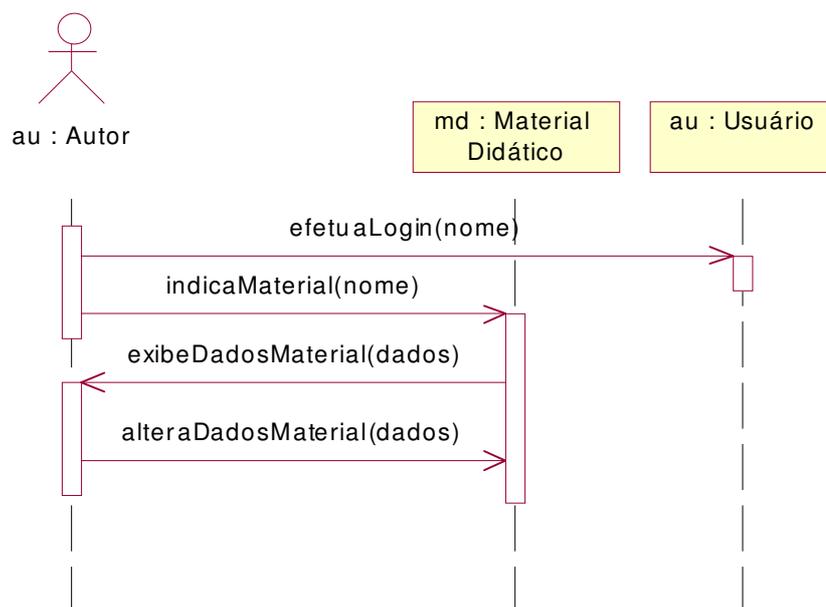


Figura 75 – Diagrama de Seqüência – Alterar Material

CENÁRIO – EXCLUIR MATERIAL DIDÁTICO

Um autor acessa o sistema para excluir um material didático existente; informa nome, senha e o material a ser excluído. O sistema apresenta os dados do material e solicita confirmação. O autor confirma e o sistema realiza a operação (Figura 76).

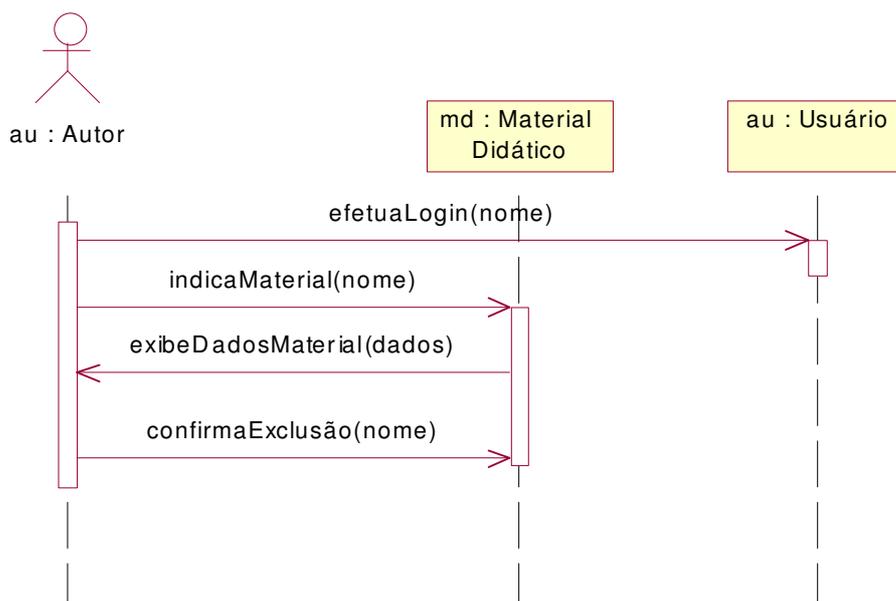


Figura 76 – Diagrama de Seqüência – Excluir Material

CONTEXTO – MANTER AULA

CENÁRIO – ALTERAR AULA

Um professor acessa o sistema para alterar uma aula, efetua *login* e informa a aula a ser alterada. O sistema exibe os dados. O professor altera os dados cadastrais de uma aula (Figura 77).

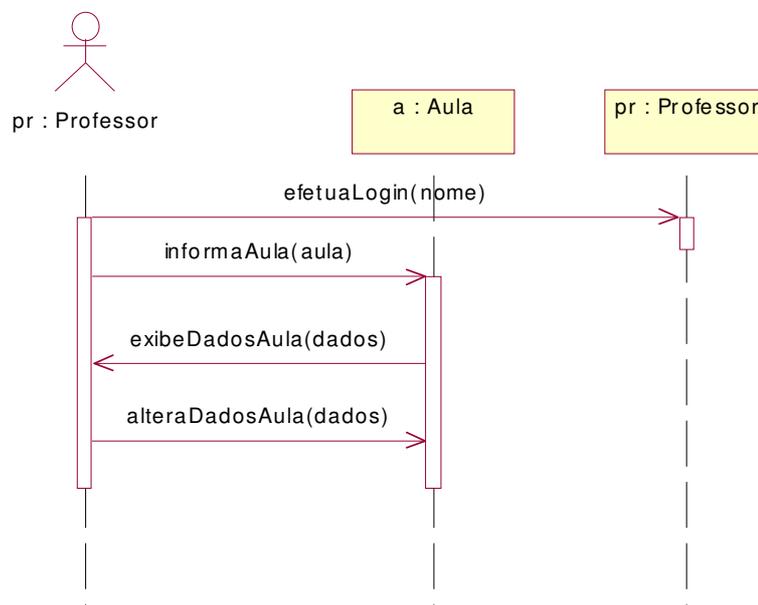


Figura 77 – Diagrama de Seqüência – Alterar Aula

CENÁRIO – EXCLUIR AULA

Um professor acessa o sistema para excluir uma aula, efetua *login* e informa a aula a ser excluída. O sistema apresenta os dados e o professor confirma a operação (Figura 78).

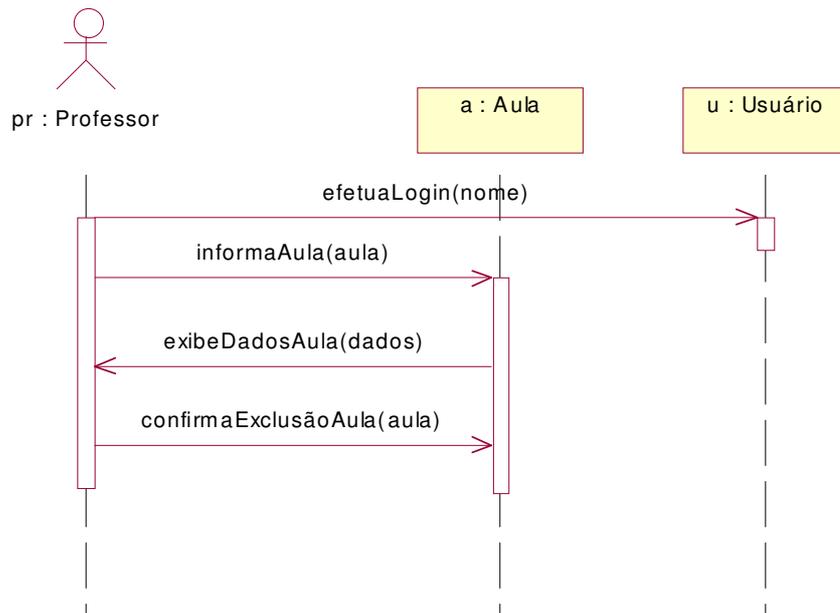


Figura 78 – Diagrama de Seqüência – Excluir Aula

CENÁRIO – EXCLUIR MATERIAL DIDÁTICO DE UMA AULA

Um professor acessa o sistema para excluir material didático de uma aula. Efetua *login* e informa a aula a ser trabalhada. O sistema informa os dados cadastrais da aula e a lista de material didático nela existente. O professor seleciona o(s) assunto(s) que excluirá de sua aula. O sistema desassocia o material didático da aula. (Figura 79).

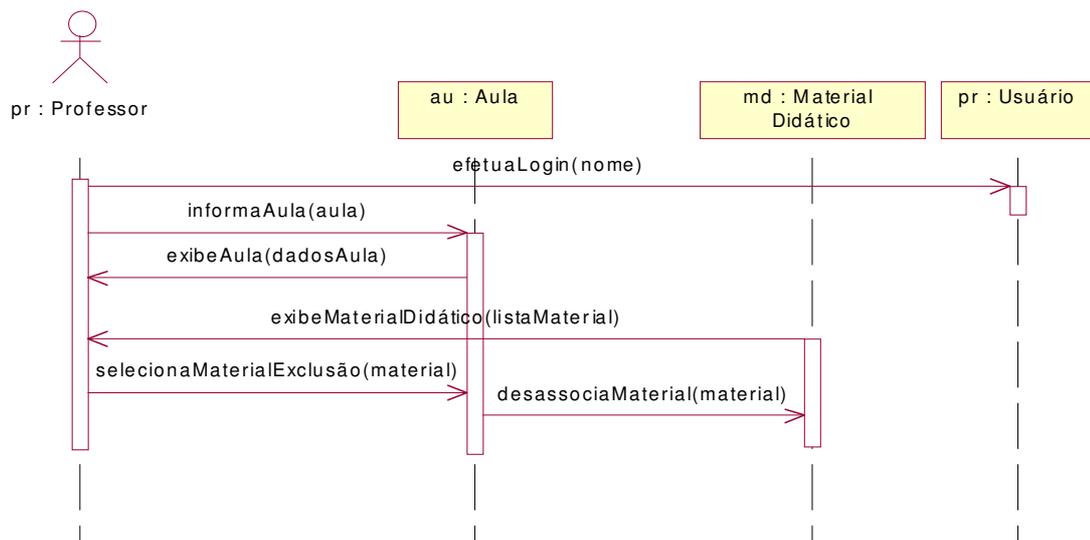


Figura 79 – Diagrama de Seqüência – Excluir Material de uma Aula

CONTEXTO – MANTER CURSO

CENÁRIO – ALTERAR CURSO

Contexto – Um professor acessa o sistema para alterar um curso. Efetua *login* e informa o curso a ser alterado. O sistema exibe os dados. O professor altera os dados cadastrais do curso (Figura 80).

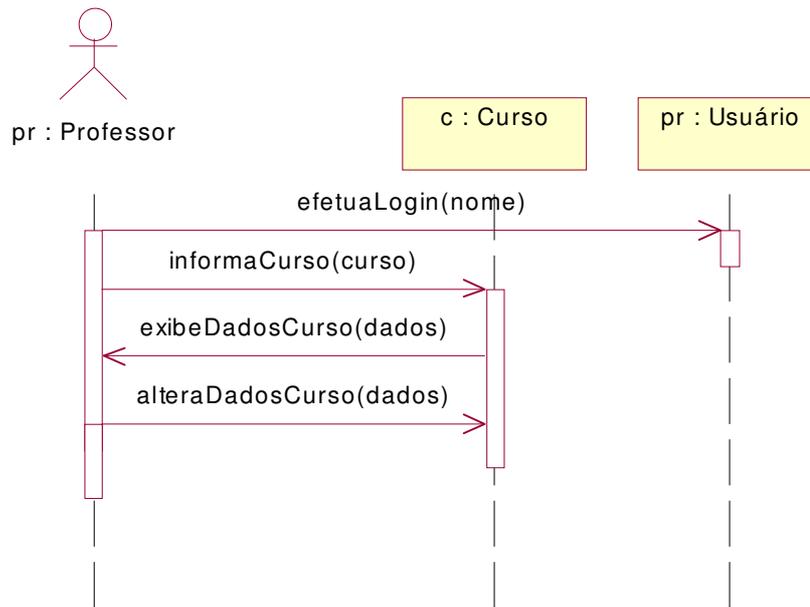


Figura 80 – Diagrama de Seqüência – Alterar Curso

CENÁRIO – EXCLUIR CURSO

Um professor acessa o sistema para excluir um curso. Efetua *login* e informa o curso a ser excluído. O sistema exibe os dados e o professor confirma a operação (Figura 81).

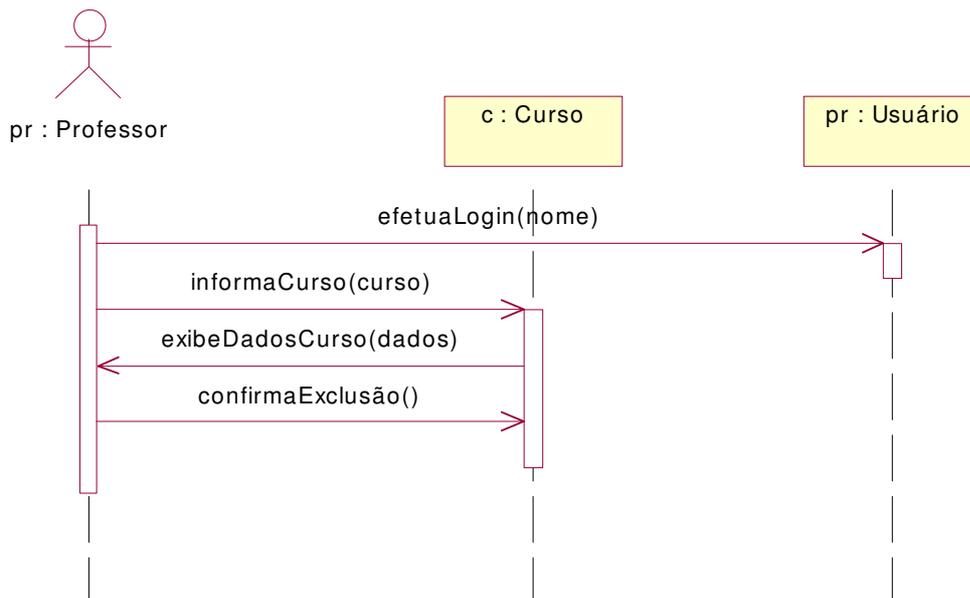


Figura 81– Diagrama de Seqüência – Excluir Curso

CENÁRIO – EXCLUIR AULAS DE UM CURSO

Um professor acessa o sistema para excluir aulas de um curso, efetua *login* e informa o curso a ser trabalhado. O sistema exibe os dados cadastrais do curso e a lista de aulas alocadas para o curso. O professor seleciona a(s) aula(s) que excluirá do curso. O sistema desassocia a aula do curso (Figura 82).

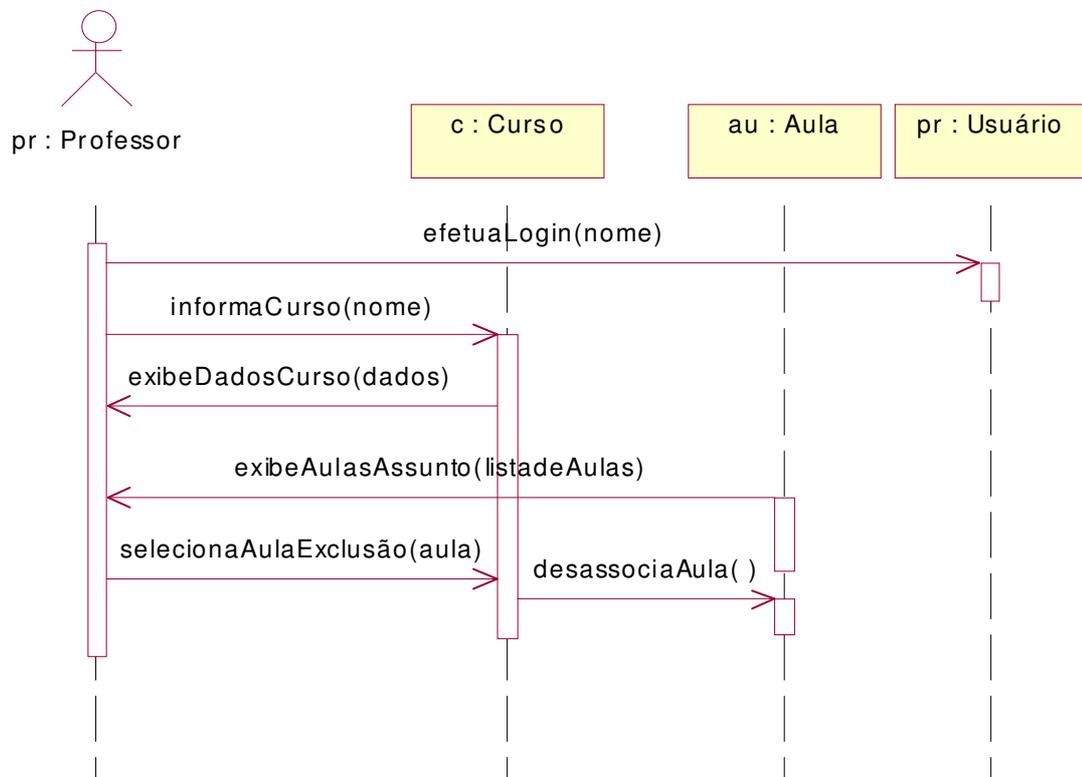


Figura 82 – Diagrama de Seqüência – Excluir Aula de um Curso

CONTEXTO – MANTER ONTOLOGIA

CENÁRIO – ALTERAR DOMÍNIO

O administrador acessa o sistema e informa o domínio a ser alterado. O sistema exibe os dados do domínio e o administrador os altera (Figura 83).

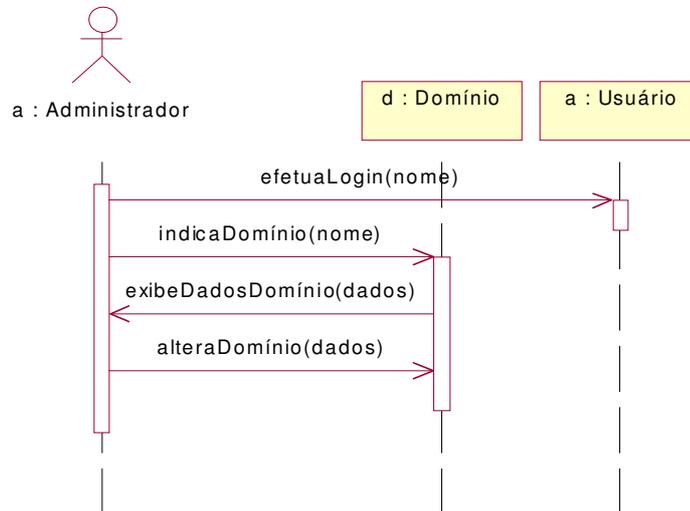


Figura 83 – Diagrama de Seqüência – Alterar Domínio

CENÁRIO – EXCLUIR DOMÍNIO

O administrador acessa o sistema e informa o domínio a ser excluído. O sistema exibe os dados do domínio. O administrador confirma a operação (Figura 84).

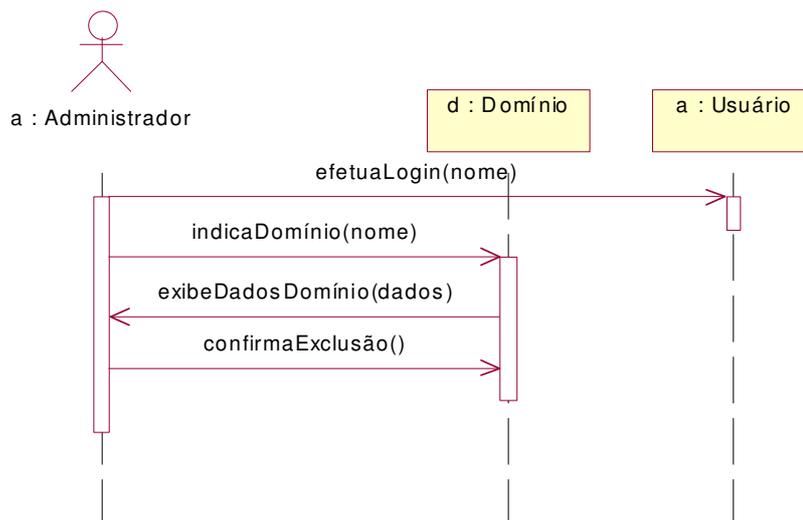


Figura 84 – Diagrama de Seqüência – Excluir Domínio

CENÁRIO – ALTERAR ASSUNTO

O administrador acessa o sistema para alterar um assunto. O sistema exibe os domínios existentes. O administrador seleciona o domínio. O sistema apresenta os assuntos para o domínio selecionado, o administrador seleciona o assunto a ser alterado, o sistema exibe os dados do assunto e o administrador os altera. (Figura 85).

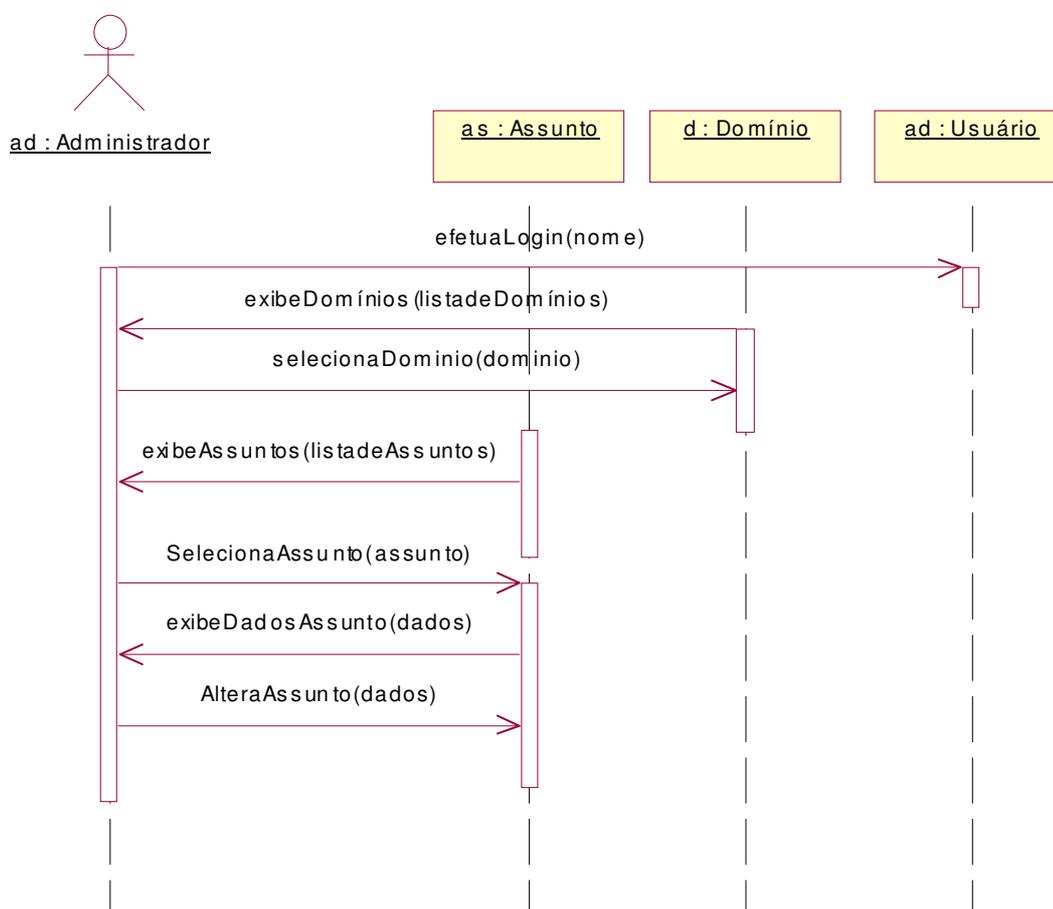


Figura 85 – Diagrama de Seqüência – Alterar Assunto

CENÁRIO – EXCLUIR ASSUNTO

O administrador acessa o sistema para excluir um assunto. O sistema exibe os domínios existentes. O administrador seleciona o domínio. O sistema mostra os assuntos para

o domínio selecionado. O administrador seleciona o assunto a ser excluído e confirma a operação (Figura 86).

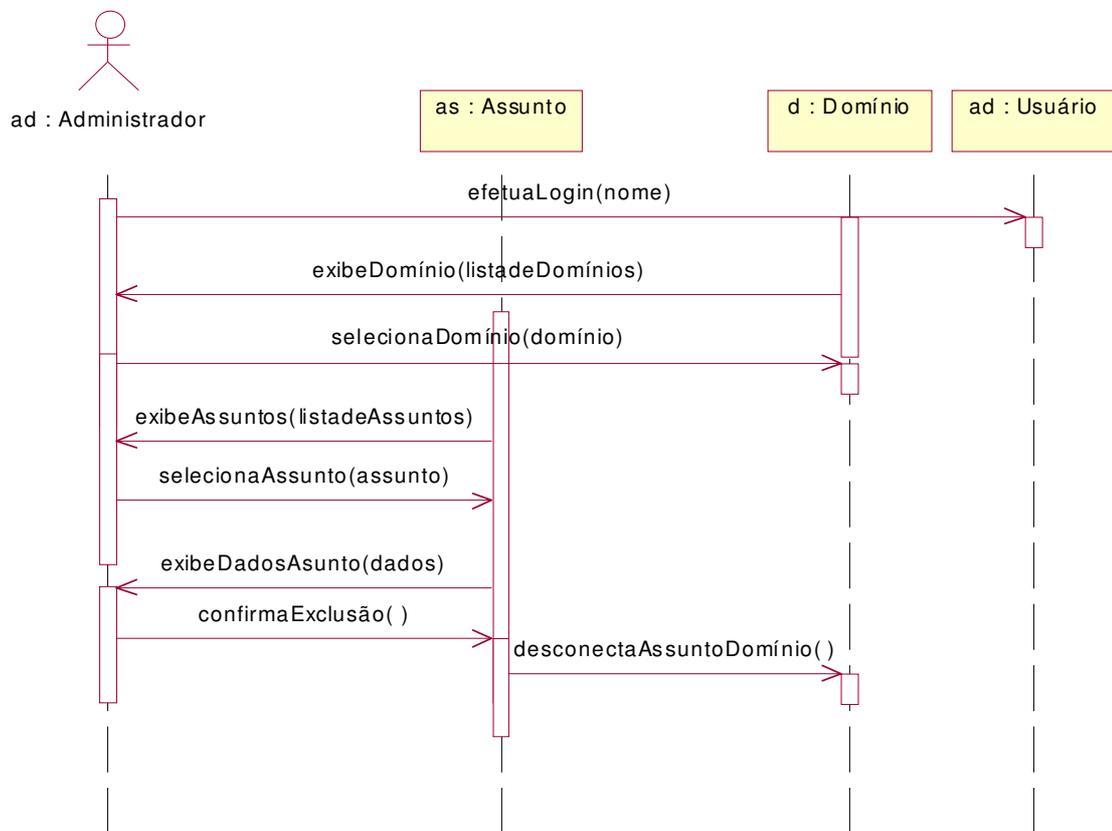


Figura 86 – Diagrama de Seqüência – Excluir Assunto

ANEXO III – CÓDIGO DE IMPLEMENTAÇÃO

Acompanha esta dissertação um CD, que contém as seguintes informações:

- Modelagem do SGC em *Rational Rose*
- *Script* do Banco de Dados do SGC
- Fontes dos programas do SGC, em Java e *JSP*