

Viviane Torres da Silva

**CONTENTNET: UM FRAMEWORK PARA  
INTEROPERABILIDADE DE CONTEÚDOS EDUCACIONAIS  
UTILIZANDO O PADRÃO IMS.**

DISSERTAÇÃO DE MESTRADO  
DEPARTAMENTO DE INFORMÁTICA  
PUC-RIO

Rio de Janeiro, 24 de março de 2000.

Viviane Torres da Silva

**CONTENTNET: UM FRAMEWORK PARA  
INTEROPERABILIDADE DE CONTEÚDOS EDUCACIONAIS  
UTILIZANDO O PADRÃO IMS**

Dissertação apresentada ao Departamento de Informática da PUC-Rio como parte dos requisitos para a obtenção do título de Mestre em Ciências em Informática.

Orientador: Carlos José Pereira de Lucena

Departamento de Informática

Pontifícia Universidade Católica do Rio de Janeiro

Rio de Janeiro, 24 de março de 2000

"Ontem você não fez.

Hoje você sente falta.

Amanhã você se arrependerá.

Porque tem que ser assim!?"

Viviane Torres da Silva

Ao meu marido Carlos Eduardo Marques de Macedo

Ao meu pai Joaquim Gomes da Silva

À minha mãe Rosalina Maria Torres da Silva

Ao meu irmão Luís Gustavo Torres da Silva

## Agradecimentos

Aos meus familiares por estarem sempre presentes nos momentos de felicidade e de tristeza com um palavra de apoio e por me incentivarem sempre a continuar.

Ao professor Carlos José Pereira de Lucena por sempre acreditar em mim, mesmo quando eu duvidava; por guiar meus passos e me carregar sempre que necessário; por ser esta pessoa tranqüila e serena que deixa qualquer tempestade parecer uma garoa; por ser fundamental.

Ao professor Hugo Fuks por me ajudar nas decisões de implementação junto ao ambiente AulaNet dando o suporte necessário para a implementação das ferramentas e pelas conversas e brincadeiras da hora do almoço.

À professora Clarisse Sieckenius de Souza por ser firme quando necessário, por ser flexível quando possível e pelas aulas de MGIU que tanto contribuíram no desenvolvimento das interfaces e dos testes realizados nesta dissertação.

Aos meus amigos Marcelo Blois, Ricardo Choren e Paulo Roberto Caroli por terem me ajudado a decidir pontos importantes desta dissertação; por terem me ajudado a decidir pontos importantes na minha vida; por terem tido muita paciência comigo; por terem transformado os dias difíceis em dias agradáveis; por terem sido exatamente o que são, meus amigos.

À amiga Milene Selbach Silveira por ter desenvolvido parte deste trabalho ao meu lado, o que foi essencial; por ter revisto toda a dissertação; por ter paciência e por estar sempre de bom humor.

À amiga Cláudia Mesquita por ter revisto grande parte desta dissertação, por ter me ajudado a superar meu nervosismo, por estar sempre disposta a ouvir minhas dúvidas e meus problemas.

## II

À amiga Vera A. S. Menezes por me ajudar em tudo que podia, por tentar me ajudar no que não podia e por sempre me receber com um sorriso quando eu tinha algo para pedir, o que eu mais fiz estes anos.

À Rafaela Ventura por ter sido a revisora desta dissertação.

Aos demais amigos do LES por reverem partes da minha tese, como Simone Barbosa e Sérgio Crespo; por terem sido "voluntários" nos testes das interfaces, Gustavo Robichez, Claudia Mesquita, Marcelo Blois, e por terem contribuído para transformar um ambiente de trabalho em um ambiente alegre e divertido.

Aos amigos do DI pelas conversas dos corredores, pelos trotes e brincadeiras que tanto contribuíram para a minha formação.

À meus amigos extra PUC por terem contribuído para a minha distração e por terem me aturado quando eu só conseguia falar sobre a tese.

Ao CNPq por ter dado o apoio financeiro necessário para a execução desta dissertação.

## Resumo

A *World Wide Web* foi, originalmente, organizada para que as informações nela contidas pudessem ser lidas por máquinas, sem que estas precisassem preocupar-se em interpretar o conteúdo. Com o crescimento do volume de informações existentes na *Web*, buscar, ter acesso a e obter qualquer informação da *Web* tornou-se extremamente difícil.

Visando à solução deste problema, o projeto IMS, um consórcio de organizações acadêmicas, comerciais e governamentais, vem desenvolvendo e propondo especificações para facilitar o crescimento e a viabilidade de atividades *online* na área de educação. Uma das atividades mais importantes do projeto é a aprendizagem na *Internet*, com a possibilidade de interoperabilidade entre diferentes ambientes para dar suporte à educação baseada na *Web*. Com base neste projeto e na necessidade de reutilização de conteúdos já existentes em servidores AulaNet, este trabalho apresenta um *framework* orientado a objetos que facilita a descrição, localização e o uso de conteúdos educacionais disponíveis em servidores compatíveis com o projeto IMS.

## **Abstract**

The World Wide Web original organization assumed that its stored information was to be read by machines without concern to the possibility of machines interpreting this information. The growth of the amount of information available on the Web made it more complex to search, have access to and to obtain information on the Web.

To cope with this problem the platform IMS has been proposed. A consortium formed by academic institutions, companies and government agencies is developing the platform. The goal of the platform is to promote the viability and growth of online activities in the area of education. One of the most important activities of the consortium is Internet based learning with the possibility of interoperability among different environments for Web-based education. Based on the proposed platform and the need for content exchange among different AulaNet servers we have developed an object oriented framework to enable the description, localization and use of educational content available in servers which are compatible with the IMS platform.



# Sumário

<b>SUMÁRIO</b>	<b>V</b>
<b>LISTA DE FIGURAS</b>	<b>IX</b>
<b>LISTA DE TABELAS</b>	<b>XI</b>
<b>LISTA DE GRÁFICOS</b>	<b>XI</b>
<b>CAPÍTULO 1</b>	<b>1</b>
<b>INTRODUÇÃO</b>	<b>1</b>
1.1 MOTIVAÇÃO	1
1.2 ORGANIZAÇÃO DO TRABALHO	3
<b>CAPÍTULO 2</b>	<b>4</b>
<b>AMBIENTE AULANET</b>	<b>4</b>
2.1 DESCRIÇÃO DO AMBIENTE AULANET	4
2.2 NOVO ATOR CATALOGADOR DE CONTEÚDO	8
2.3 NOVO PAPEL PARA O PROFESSOR	9
<b>CAPÍTULO 3</b>	<b>11</b>
<b>PADRÕES PARA REPRESENTAÇÃO DE CONTEÚDOS</b>	<b>11</b>
3.1 INTRODUÇÃO	11
3.1.1 Principais iniciativas	12
IMS	12
AICC	12

<i>IEEE</i>	13
3.2 PROJETO IMS	13
3.2.1 <i>Descrição do projeto IMS</i>	13
3.2.2 <i>Problemas ainda não resolvidos</i>	15
3.2.3 <i>O que vamos utilizar do projeto</i>	15
<b>CAPÍTULO 4</b>	<b>18</b>
<b>FRAMEWORK CONTENTNET</b>	<b>18</b>
4.1 INTRODUÇÃO A <i>FRAMEWORK</i>	18
4.2 DESCRIÇÃO DO <i>FRAMEWORK</i> CONTENTNET	21
4.3 ARQUITETURA UTILIZADA	22
4.4 DIAGRAMA DE CLASSES E DE INTERAÇÃO	26
4.4.1 <i>Módulo IServer</i>	27
4.4.2 <i>Módulo Content Search</i>	31
<b>CAPÍTULO 5</b>	<b>33</b>
<b>INSTANCIACÃO DO <i>FRAMEWORK</i> CONTENTNET</b>	<b>33</b>
5.1 - MÓDULO CONTENT SEARCH	33
5.2 - MÓDULO ISERVER	34
5.2.4 - <i>Ferramenta de cadastramento da meta-informação</i>	38
5.3 - DIAGRAMA DE FLUXO DE MENSAGEM	39
5.4 - OUTRAS INSTANCIACÕES	40
5.4.1 - <i>CompareContent</i>	42
5.4.2 - <i>GetContent</i>	42
5.4.3 - <i>SetContent</i>	43
<b>CAPÍTULO 6</b>	<b>44</b>
<b>INTERFACE DO CONTENTNET</b>	<b>44</b>
6.1 INTRODUÇÃO A CENÁRIO	44
6.2 INTERFACE DO PROFESSOR	45
6.2.1 <i>Perfil dos usuários</i>	46
6.2.2 <i>Cenários</i>	47

## VII

6.2.3 Características da ferramenta	50
6.2.4 Expectativas do usuário	51
6.2.5 Design da Interface	52
6.3 INTERFACE DO CATALOGADOR	57
6.3.1 Perfil dos usuários	58
6.3.2 Cenários	58
6.3.3 Características da ferramenta	60
6.3.4 Expectativas do usuário	61
6.3.5 Design da Interface	62
6.4 AVALIAÇÃO DA COMUNICABILIDADE	67
6.4.1 Aplicação de tags	67
6.4.2 Interpretação dos resultados	69
6.5 TESTES REALIZADOS	69
6.5.1 Interface do Professor	70
Cenário	71
Tarefas	71
6.5.2 Interface do Catalogador	74
Tarefa	74
<b>CAPÍTULO 7</b>	<b>77</b>
<b>CONCLUSÕES E TRABALHOS FUTUROS</b>	<b>77</b>
7.1 CONCLUSÕES	77
7.2 TRABALHOS FUTUROS	78
<b>APÊNDICE A</b>	<b>80</b>
<b>FORMULÁRIO DOS TESTES REALIZADOS</b>	<b>80</b>
A.1 FORMULÁRIO UTILIZADO NO TESTE DA INTERFACE DO PROFESSOR	80
A.2 FORMULÁRIO UTILIZADO NO TESTE DA INTERFACE DO CATALOGADOR	84
<b>REFERÊNCIAS</b>	<b>88</b>

## Lista de Figuras

Figura 1 - <i>Kernel e hot-spots</i> _____	19
Figura 2 - Processo de desenvolvimento de <i>framework</i> _____	20
Figura 3 - Interoperabilidade de conteúdo _____	21
Figura 4 - Pontos de flexibilização. MD-base de meta-informação, BD-base de conteúdo__	23
Figura 5 - <i>Framework: kernel</i> e pontos de flexibilização. _____	24
Figura 6 -Classes do ponto de flexibilização: CompareContent, GetContent, SetContent __	25
Figura 7 - Diagrama de Interação (processo cliente) _____	28
Figura 8 - Diagrama de Interação (processo servidor) _____	29
Figura 9 - Módulo <i>IServer</i> _____	30
Figura 10 - Módulo <i>Content Search</i> _____	32
Figura 11 - Diagrama de classes da instância do módulo <i>IServer</i> _____	37
Figura 12 - Ferramenta de catalogação _____	38
Figura 13 - Diagrama de fluxo de mensagem _____	40
Figura 14 - Tela 1 - Menu de opções _____	52
Figura 15 - Tela 2 - Seleção do tipo de conteúdo _____	53
Figura 16 - Tela 3 - Seleção dos atributos _____	54
Figura 17 - Tela 4 - Formulário _____	55
Figura 18 - Tela 5 - Resultado da pesquisa _____	56
Figura 19 - Tela 6 - Finalização _____	57
Figura 20 - Tela 1 - Menu de opções _____	63
Figura 21 - Tela 2 - Seleção do tipo de Conteúdo _____	64
Figura 22 - Tela 3 - Lista de Conteúdo _____	64
Figura 23 - Tela 4 - Atributos da Pesquisa _____	65
Figura 24 - Contato _____	66
Figura 25 - Tela 5 - Fim do processo de catalogação _____	66

## Lista de Tabelas

Tabela 1 - Mecanismos e seus recursos _____	7
Tabela 2 - Fatores de influência na implementação dos hots-spots _____	41
Tabela 3 - Relacionamento entre as <i>tags</i> e os problemas de interatividade e usabilidade. __	69
Tabela 4 - Perfil dos usuários da interface do professor_____	70
Tabela 5 - Perfil dos usuário da interface do catalogador _____	74

## Lista de Gráficos

Gráfico 1 - Análise da Comunicabilidade - Interface Professor_____	72
Gráfico 2 - Análise da Comunicabilidade - Interface Catalogador _____	75

# Capítulo 1

## Introdução

A *World Wide Web* foi, originalmente, organizada para que as informações nela contidas pudessem ser lidas por máquinas, sem que estas precisassem preocupar-se em interpretar o conteúdo. Com o crescimento do volume de informações existentes na *Web*, buscar, ter acesso a e obter qualquer informação da *Web* tornou-se extremamente difícil.

Visando à solução deste problema, o projeto IMS, um consórcio de organizações acadêmicas, comerciais e governamentais, vem desenvolvendo e propondo especificações para facilitar o crescimento e a viabilidade de atividades *online* na área de educação. Uma das atividades mais importantes do projeto é a aprendizagem na *Internet*, com a possibilidade de interoperabilidade entre diferentes ambientes para dar suporte à educação baseada na *Web*. Com base neste projeto e na necessidade de reutilização de conteúdos já existentes em servidores AulaNet, este trabalho apresenta um *framework* orientado a objetos que facilita a descrição, localização e o uso de conteúdos educacionais disponíveis em servidores compatíveis com o projeto IMS.

### 1.1 Motivação

Procurar um item específico na *World Wide Web* é um processo semelhante à procura de um livro em um ambiente desorganizado. Além disto, o número de recursos disponíveis na *Web* continua crescendo exponencialmente e, com eles, a necessidade de obtenção de mais informações sobre os recursos disponíveis [IMS, 2000a]. Assim como uma biblioteca utiliza

diversos catálogos organizados para encontrar um livro na prateleira, a *Web* também necessita de catálogos para localizar os recursos disponíveis nos diversos servidores. No caso da biblioteca, o conteúdo destes catálogos possui informação sobre os livros como seu autor, título e editora. No caso da *Web*, a informação contida nos catálogos é sobre os recursos, que no caso de recursos educacionais podem ser título, descrição e formato. Essa informação sobre informação tem o nome de meta-informação [IMS Metadata, 2000; W3C Metadata, 1999] ou, especificamente no contexto que abordaremos, "informação sobre conteúdos educacionais disponíveis nos servidores de conteúdo".

Com a descrição dos conteúdos utilizando-se de meta-informação, a busca por um conteúdo fica facilitada se for usada uma ferramenta de busca que trate deste tipo de informação. Com esta ferramenta, e dado que a meta-informação possui as características dos conteúdos disponíveis nos servidores, a busca pelos conteúdos torna-se mais eficiente.

O *framework* ContentNet dá suporte a instanciação de uma ferramenta de busca de conteúdos educacionais baseada em meta-informação. Esta ferramenta, parte localizada no servidor de pesquisa e parte nos servidores de conteúdo, estabelece a comunicação entre os servidores de pesquisa e de conteúdo, a fim de permitir o acesso às meta-informações e aos conteúdos disponíveis nestes servidores.

Para testar a viabilidade do *framework*, este foi instanciado para servidores AulaNet. Para tanto, foi necessária a implementação de uma segunda ferramenta que possibilitasse a catalogação de conteúdo. Esta ferramenta, que não faz parte do *framework*, foi implementada para permitir a adaptação do ambiente AulaNet ao padrão IMS. Junto com ela, foi implementada a interface de acesso do usuário, catalogador de conteúdo, à ferramenta de catalogação.

No teste de viabilidade do *framework*, surgiu, também, a necessidade de implementação da interface do usuário para acesso à ferramenta de busca. Esta interface, que não faz parte do *framework*, comunica o usuário que deseja um conteúdo à ferramenta de busca gerada na instância do *framework*.

## 1.2 Organização do trabalho

Este trabalho está organizado em sete capítulos, sendo que o primeiro compreende a introdução do trabalho. O segundo capítulo descreve o ambiente AulaNet para o qual o *framework* foi instanciado, dando ênfase à descrição dos tipos de conteúdo existentes no ambiente. Em seguida, no Capítulo 3, apresenta-se uma introdução aos projetos voltados à representação de conteúdos, dando ênfase a três principais projetos. Ainda neste capítulo, encontra-se a descrição do projeto e do padrão IMS utilizado neste trabalho.

O *framework* ContentNet é apresentado no Capítulo 4. Neste capítulo, encontra-se uma breve introdução ao conceito de *framework*, à arquitetura do *framework* proposto e aos diagramas de classe e de interação que compreendem o mesmo. No Capítulo 5, encontram-se os detalhes sobre a instanciação do *framework*, incluindo a ferramenta de catalogação de conteúdo usada no ambiente AulaNet e o diagrama de fluxo de mensagens trocadas entre o servidor de pesquisa e os servidores de conteúdo. Neste capítulo, aborda-se, também, um estudo sobre outras possíveis instâncias do *framework* ContentNet.

O Capítulo 6, por sua vez, descreve as interfaces geradas para as ferramentas mencionadas nos capítulos anteriores. Este capítulo introduz o conceito de cenário e, em seguida, apresenta o estudo realizado para a geração da interface do professor, usuário da ferramenta de pesquisa, e o estudo realizado para a interface do catalogador, usuário da ferramenta de catalogação. Por fim, este capítulo apresenta o método de avaliação da comunicabilidade e os testes realizados perante as interfaces projetadas. No último capítulo, Capítulo 7, estão as conclusões realizadas sobre o trabalho apresentado e os trabalhos futuros propostos.



## Capítulo 2

### Ambiente AulaNet

O ambiente AulaNet, por ser um ambiente educacional para a criação e manutenção de cursos, armazena, em sua base de dados, conteúdos educacionais. Assim, servidores AulaNet podem ser considerados servidores de conteúdo educacional. Com base nessa afirmação, servidores AulaNet foram utilizados a fim de validar<sup>1</sup> o *framework* ContentNet. O primeiro passo para a utilização do *framework* ContentNet, junto ao AulaNet, é a adaptação do ambiente em relação ao padrão IMS. Para tanto, fez-se necessário um estudo do ambiente com a finalidade de identificação dos tipos de conteúdo existentes. Uma vez identificados os tipos de conteúdo, foi necessária a criação de um novo ator, catalogador de conteúdo, e de um novo papel para o professor, pesquisador de conteúdo.

#### 2.1 Descrição do Ambiente AulaNet

O AulaNet[Lucena, 1998] é um ambiente educacional para a criação, manutenção, assistência e administração de cursos baseados na *Web*. Os objetivos do AulaNet são a adoção da *Web* como um ambiente educacional; a criação de uma transição viável da sala de aula convencional para a sala de aula virtual, oferecendo, assim, ao professor, a oportunidade de reutilizar seu material educacional existente em mídia digital (texto de aula, imagens, vídeo); e a criação de comunidades de conhecimento. Utilizando o AulaNet, o professor não precisa

---

<sup>1</sup> Entende-se por validação o teste proveniente da utilização do *framework* junto ao AulaNet.

conhecer nenhuma linguagem de programação voltada à *Internet* para criar, manter ou lecionar cursos à distância. O AulaNet separa o conteúdo de sua navegação, afastando, assim, o professor da tarefa de programação, o que permite que o professor focalize seu conhecimento na produção do conteúdo e não no aprendizado de tecnologias da *Internet*.

O AulaNet considera os seguintes atores envolvidos no processo de criação/aprendizado:

1. Administrador: É o facilitador da integração professor/curso/aluno que trata de questões de natureza eminentemente operacional, como inscrição de professores, admissão de alunos em cursos, etc.;
2. Aluno: É o usuário final do curso, representa o público-alvo para quem se destina o curso;
3. Professor: É o criador do curso, participa desde sua descrição inicial até a entrada dos conteúdos [Lucena, 1999b].

O AulaNet difere da maioria dos ambientes digitais de aprendizagem existentes (ex.: [Classnet, 1999][Web-Course, 1999][Vistual-U, 1999][WebCT, 1999]), pois estes utilizam metáforas físicas da escola tradicional: corredores, quadro negro, salas de aula, etc.. O AulaNet baseia-se em uma abordagem cooperativa, acreditando-se, assim, que o aprendizado e os aspectos intelectuais do trabalho estão se tornando o mesmo conceito. Esta abordagem cooperativa baseia-se na idéia de que para cooperar, as pessoas têm que se coordenar, e para se coordenar, as pessoas precisam comunicar-se. A fim de que o professor possa customizar seu curso de acordo com os objetivos do processo de aprendizado, o AulaNet oferece um conjunto de mecanismos de comunicação, coordenação e cooperação.

O mecanismo de comunicação inclui: ferramenta de envio de mensagem aos docentes, o que permite o contato dos aprendizes com o coordenador e com os possíveis docentes co-autores e instrutores do curso por *e-mail*; ferramenta para criação de grupo de discussão, o que possibilita a criação de um fórum de discussão entre os participantes do curso por *e-mail*; ferramenta para grupo de interesse, o que permite a criação de conferências assíncronas textuais na forma de discussão orientada; debate, o que possibilita a criação de conferências síncronas textuais por meio de *chat*; e ferramenta para contato com os participantes, o que permite que um participante possa contatar quem está conectado no ambiente.

O mecanismo de coordenação fornece: uma ferramenta para inclusão de avisos, o que possibilita a criação de avisos sobre o curso ou agendamento de eventos por meio de informes; ferramenta para criação de plano de aulas, o que permite a criação de uma estrutura básica para o acompanhamento dos materiais do curso; ferramenta para realização de tarefas, o que habilita a criação de tarefas (exercícios) a serem realizadas pelos aprendizes; ferramenta para criação de avaliações, o que permite a criação de exames para auto-avaliação dos aprendizes do curso; e uma outra ferramenta para o acompanhamento da participação, o que possibilita visualizar relatórios com as contribuições dos participantes do curso.

O mecanismo de cooperação agrupa: bibliografia, o que permite a criação de referências bibliográficas para o curso; *webliografia*, o que possibilita a utilização de URLs (*links* externos) no curso; documentação, o que permite a criação de conteúdos não ligados diretamente a uma aula do plano de aulas; co-autoria de docente, o que possibilita a indicação de outros docentes para co-autorarem o curso; co-autoria de aprendiz, o que permite a indicação de aprendizes para criarem conteúdos para o curso; e uma ferramenta para *download*, o que habilita o aprendiz a visualizar uma lista com todos os arquivos do curso[Lucena, 1999c].

Ao estudar os recursos disponíveis nesses três mecanismos, foi possível verificar que os conteúdos produzidos pelo professor referem-se a um material, uma aula, uma documentação, uma bibliografia, uma *webliografia*, uma avaliação, uma questão, uma tarefa e a um curso como um todo. Mais detalhadamente, entende-se por: conteúdo AulaNet um material, ou seja, um arquivo mais um texto explicativo; uma aula, um conjunto de materiais mais um texto explicativo; uma documentação, um conjunto de materiais mais um texto explicativo; uma bibliografia, um texto explicativo e um arquivo; uma *webliografia*, um texto explicativo e um *link*; uma avaliação, uma questão e uma tarefa, o que compreende textos; e um curso o que compreende o agrupamento de todos os conteúdos citados anteriormente.

Cooperação	Comunicação	Coordenação
<ul style="list-style-type: none"> <li>• Bibliografia ↗</li> <li>• Webliografia ↗</li> <li>• Documentação</li> <li>• Co-autoria de docente</li> <li>• Co-autoria de aprendiz</li> <li>• Download</li> </ul>	<ul style="list-style-type: none"> <li>• E-mail docentes</li> <li>• E-mail participantes</li> <li>• Grupo de Interesse</li> <li>• Grupo de discussão</li> <li>• Debate</li> </ul>	<ul style="list-style-type: none"> <li>• Avisos</li> <li>• Plano de aulas ↗</li> <li>• Tarefas ↗</li> <li>• Avaliações ↗</li> <li>• Acompanhamento da participação</li> </ul>

Tabela 1 - Mecanismos e seus recursos

Na Tabela 1, encontram-se os três mecanismos mencionados e seus recursos. Os recursos sinalizados são aqueles considerados como conteúdo. Nessa sinalização, não constam os conteúdos material e questão, uma vez que não são recursos AulaNet. Somente foram considerados conteúdos devido a material ser a menor representação de um conteúdo composto por arquivo, um material está presente em todos os conteúdos que utilizam arquivos, e a questão de prova ser a menor representação do conteúdo avaliação.

Por causa do grande número de servidores de informação disponíveis e da disseminação do ambiente AulaNet, o qual desde abril de 1998 possui mais de 1500 servidores instalados no país e fora dele [Lucena, 1999a], surgiu a necessidade de compartilhar conteúdos entre os diversos servidores, ou seja, surgiu a necessidade de utilizar-se recursos didáticos já publicados por diferentes autores em diferentes servidores. Visto que há muitos recursos de qualidade já disponíveis na *Web*, em servidores AulaNet ou não, porque não reutilizá-los? Esta necessidade gerou outras demandas devido à dificuldade de localização de conteúdo pelo qual se procura, e, mais difícil ainda, à maneira pela qual será utilizado este conteúdo, fazendo com que surgissem perguntas até então não respondidas. Como reutilizar as informações disponíveis? Como buscar as informações disponíveis? Como ter acesso às informações? Como publicar informações para facilitar a reutilização?

Dado que a filosofia do ambiente AulaNet está centrada em cursos, o nível de granularidade existente era o próprio curso. Em outras palavras, não era possível o acesso a cada item de conteúdo existente, mas tão somente ao curso como um todo. A fim de tornar possível a reutilização de todos os conteúdos existentes em ambientes AulaNet, independentemente da hierarquia existente entre eles, fez-se necessário alterar a granularidade para o menor nível

que possuísse semântica. Desse modo, a granularidade reduziu-se a cada recurso AulaNet e a cada conteúdo que compõe o curso, tornando possível, por exemplo, a reutilização de um curso, de um material e até mesmo de uma questão da auto avaliação. Uma vez definida a granularidade do conteúdo AulaNet, o próximo passo é catalogar os conteúdos produzidos pelos professores. A catalogação faz-se necessária para a pesquisa por conteúdo desejado. O pesquisador do conteúdo, novo papel do professor do ambiente AulaNet, utiliza os atributos da catalogação como, por exemplo, o título do conteúdo, descrição e formato, para identificar o conteúdo que deseja. A fim de que se catalogassem os conteúdos produzidos pelos professores, surgiu a necessidade da criação de um novo ator, o catalogador, em cada servidor AulaNet.

## **2.2 Novo Ator Catalogador de Conteúdo**

O papel do novo ator catalogador no ambiente AulaNet é, como o próprio nome já sugere, catalogar os conteúdos produzidos pelos professores, ou seja, fornecer uma descrição destes conteúdos. Em outras palavras, o papel do catalogador é cadastrar as meta-informações dos conteúdos existentes e atualizar o banco de dados que armazena esta informação à medida que novos conteúdos vão sendo criados pelos professores.

Cada servidor AulaNet deve possuir seu próprio catalogador que irá cadastrar a descrição de todo conteúdo que os professores do mesmo servidor produzirem e que quiserem deixar disponível para pesquisa, isto é, todo conteúdo que os professores quiserem disponibilizar para o acesso durante a pesquisa. Para tanto, o catalogador deve ter acesso a estes conteúdos e deve ser capaz de extrair as informações necessárias para a catalogação mediante observação do conteúdo. Caso o catalogador encontre alguma dificuldade em colher uma informação necessária, este deve ser capaz de contatar o professor que disponibilizou o conteúdo.

A meta-informação de um conteúdo cadastrada pelo catalogador deve seguir as normas do padrão IMS, pois todo o conteúdo, AulaNet ou não, deve seguir o mesmo padrão de catalogação. Assim, o catalogador deve ter um bom conhecimento sobre as características dos elementos da meta-informação do padrão IMS. A fim de auxiliar o catalogador em sua tarefa, a interface utilizada na catalogação deve prover toda a especificação necessária dos elementos

que compõem a meta-informação, como sua descrição, observações sobre o preenchimento e seus possíveis valores. Mais detalhes sobre a interface do bibliotecário encontram-se no Capítulo 6.

## **2.3 Novo Papel para o Professor**

Além da criação de um novo ator, criou-se, também, um novo papel para o professor do ambiente AulaNet. Agora, além de criar, dar manutenção e lecionar cursos, o professor poderá pesquisar por conteúdos existentes nos diversos servidores AulaNet. Para que um conteúdo existente em um servidor AulaNet faça parte da pesquisa, é necessário que o professor criador deste conteúdo disponibilize-o para pesquisa e, é necessário, também, que o catalogador do servidor AulaNet catalogue a meta-informação do conteúdo por completo. Caso a catalogação de um conteúdo ainda não tenha sido feita por completo, este conteúdo não fará parte da pesquisa.

O professor que deseja fazer a pesquisa utilizará o servidor AulaNet, no qual está cadastrado, e por meio deste servidor, terá acesso ao servidor de pesquisa do ContentNet. A ferramenta disponível no servidor de pesquisa possibilita que o professor procure por conteúdo AulaNet, em servidores AulaNet, e por outros conteúdos disponíveis nos servidores compatíveis com o padrão IMS dos quais possui conhecimento. Utilizando a ferramenta de pesquisa, o professor fornecerá a descrição do conteúdo que deseja obter e a ferramenta encarregar-se-á de pesquisar no servidor. A ferramenta trará para o professor uma lista com as descrições dos conteúdos similares ao que o professor deseja. O professor, por sua vez, selecionará uma das descrições disponíveis e a ferramenta buscará o conteúdo que possui tal descrição.

Não é necessário que o usuário da ferramenta, localizada no servidor de busca, conheça as características do padrão IMS. Para tanto, sua interface, descrita em mais detalhes no Capítulo 6, deve fornecer todo o auxílio necessário durante a seleção e o preenchimento dos elementos que farão parte da busca. No processo de pesquisa, o professor, usuário da ferramenta de pesquisa, deve ser capaz de selecionar grupos de atributos pelo qual deseja fazer a pesquisa. Esses grupos, por exemplo, características educacionais, informações gerais, características técnicas, descrevem os tipos de atributos que compõem o grupo. Dessa maneira, se o

professor desejar pesquisar por dados como título, idioma e palavras-chaves, deve selecionar informações gerais, e se desejar pesquisar por dados como o nível educacional do aluno que usará o conteúdo, e como o grau de dificuldade do conteúdo, deve selecionar características educacionais. Uma vez selecionados os grupos de atributos, o usuário deve fornecer valores para estes atributos. De posse destes valores, a ferramenta comporá a meta-informação do conteúdo que o professor deseja e iniciará a busca por conteúdos compatíveis com a especificação fornecida.

## Capítulo 3

### Padrões para representação de conteúdos

Grupos de organizações acadêmica, comerciais, governamentais e consórcios industriais estão colaborando para definir meios de possibilitar a interoperabilidade entre produtos e conteúdos educacionais. Destes grupos, os projetos IMS, AICC e IEEE são os principais. Neste trabalho, optou-se por utilizar um padrão definido no projeto IMS, como será mencionado a seguir.

#### 3.1 Introdução

O conceito de interoperabilidade não é novo. Este conceito originou-se na década passada[Pathware, 2000], na indústria comercial de aviação, sendo que o fórum "Aviation Industry CBT Committee" (AICC)[AICC, 2000] foi o primeiro a contribuir com especificações relevantes para interoperabilidade de sistemas.

Recentemente, o entusiasmo com este conceito agravou-se devido ao desenvolvimento da *Web*. A transição das tecnologias educacionais de ambientes fechados (LAN ou CD-ROM), nos quais soluções proprietárias eram aceitáveis, para ambientes abertos (*Web*) forçou o desenvolvimento de padrões com o objetivo de interoperabilidade.

Iniciativas emergiram com o intuito de desenvolver especificações estabelecendo relacionamento com outras iniciativas que, ao unirem-se, conseguiram resolver muitos problemas e desenvolver, conseqüentemente, especificações com mais rapidez e eficácia.



Dentro destas iniciativas estão o AICC, o IMS, o IEEE LTSC[IEEE, 2000], o W3C[W3C, 2000], o CEEdMA[CEEdMA, 2000], o ADL[ADL, 2000], ARIADNE[ARIADNE, 2000] e o MMI Workshop[MMI, 2000].

### 3.1.1 Principais iniciativas

#### IMS

O "Instructional Management Systems Global Learning Consortium" (IMS) vem desenvolvendo e promovendo especificações para facilitar a distribuição de atividades educacionais como localizar e utilizar conteúdos educacionais, e traçar o progresso, reportar a *performance* e as experiências do aprendiz entre sistemas administrativos e gerenciais.

O IMS tem dois objetivos principais:

- Definir o padrão tecnológico usado para interoperabilidade de aplicações e serviços em um aprendizado distribuído, e;
- Dar suporte a incorporações de especificações IMS em produtos e servidores voltados para a *Web*. IMS espera promover uma especificação adotada por todos, o que possibilitará que ambientes e conteúdos distribuídos, e produzidos por diferentes atores trabalhem junto, ou melhor, interoperem.

#### AICC

O "Aviation Industry CBT (Computer-Based Training) Committee" (AICC) é uma associação internacional de profissionais voltados para a área de treinamento baseado em tecnologia. As diretrizes do AICC são o desenvolvimento, distribuição, e avaliação do CBT e de técnicas de treinamento relacionadas, a fim de prover a interoperabilidade de tecnologias educacionais.

Os objetivos do AICC são:

- Assistir operadores de avião no desenvolvimento de suas diretrizes promovendo a implementação financeira e efetiva do CBT;
- Desenvolver diretrizes para possibilitar a interoperabilidade;
- Prover um fórum para discussão do CBT e de outras tecnologias de treinamento.

Durante os últimos cinco anos, a especificação para "Computer-Managed Instruction" (CMI) do AICC vem se estabelecendo como a especificação CMI mais compreensível[Pathware, 2000]. A AICC CMI define o processo de transferência de dados entre sistemas gerenciais e ambientes de suporte a aulas interativas. Define, também, um formato para estruturação dos cursos, fazendo com que estes possam ser trocados entre sistemas gerenciais diferentes.

## **IEEE**

O IEEE "Learning Technology Standards Committee" (LTSC) é um órgão aberto de padronização de tecnologias educacionais. A missão do grupo IEEE LTSC é desenvolver padrões tecnológicos, recomendar práticas, e guias para componentes de *software*, ferramentas, métodos tecnológicos e de *design* que facilitem o desenvolvimento, a manutenção e a interoperação de implementação de componentes e de sistemas educacionais e de treinamento.

Consórcios como o IMS e o AICC têm como objetivo padronizações no IEEE LTSC. O AICC já submeteu a especificação CMI enquanto que o IMS submeteu a especificação de meta-informação ao IEEE LTSC.

## **3.2 Projeto IMS**

### **3.2.1 Descrição do projeto IMS**

O grande problema com os sistemas de gerência de instrução é que cada um implementa as funcionalidades de seu sistema de gerenciamento de maneira própria. Por isso,

para os desenvolvedores de produtos e conteúdos, isso significa que desenvolver custa, geralmente, muito por diversas razões[IMS, 2000b]. Pensando nisso, o projeto IMS vem desenvolvendo alguns padrões:

- "IMS Learning Resources Meta-data Specifications" cria uma maneira uniforme de descrever recursos educacionais, tal que estes possam ser encontrados mais facilmente usando uma ferramenta de busca baseada em meta-informação;
- "IMS Enterprise Specification" visa a aplicações e serviços administrativos que precisam trocar informação sobre aprendizes, cursos, *performance*, etc. entre plataformas, sistemas operacionais e outros;
- "IMS Content & Packaging Specification" descreve uma estrutura de dados que será usada para prover interoperabilidade de conteúdos baseados na *Web* com ferramentas de criação de conteúdos, *software* de gerenciamento educacional e ambientes educacionais;
- "IMS Question & Test Specification" descreve uma estrutura de dados que será usada para prover interoperabilidade entre sistemas que utilizam questionários e testes;
- "IMS Learner Profiles Specification" focaliza-se em maneiras de organizar informações educacionais, tais que sistemas educacionais possam ser mais centrados em suas necessidades específicas.

O projeto IMS uniu-se a outras iniciativas, como a ADL, que usa as especificações definidas no projeto IMS; a ARIADNE; a AICC; a Dublin Core e a W3C, que cria especificações de baixo nível, como XML[XML, 2000], usadas pelo IMS.

O padrão de especificação de meta-informação, mencionado anteriormente, desenvolveu-se junto a ARIADNE. Este padrão incorpora os principais aspectos do Dublin Core, definindo extensões que são específicas para materiais educacionais e de treinamento. As especificações do projeto IMS também foram estendidas a partir do modelo AICC, acrescentando segurança e integração de sistemas de gerenciamento em um ambiente de educação. O IMS garante que os conteúdos que estiverem de acordo com as especificações do AICC irão operar em servidores que estiverem de acordo com as especificações do IMS.

### 3.2.2 Problemas ainda não resolvidos

O projeto IMS é muito novo e está em crescente desenvolvimento. A primeira versão de uma das especificações surgiu, em agosto de 1999, com a "IMS Learning Resources Meta-data Specifications". Em seguida, vieram a "IMS Enterprise Specification", a "IMS Content & Packaging Specification" e a "IMS Question & Test Specification". A única especificação que ainda não possui a primeira versão concluída é a "IMS Learner Profiles Specification".

O projeto IMS, como pôde ser visto anteriormente, não define o protocolo de comunicação utilizado entre os sistemas que trocarão conteúdo. Mais especificamente, tomando como base a pesquisa e a troca de conteúdo por uma ferramenta de busca, o projeto IMS não define como deve ser o protocolo usado na busca do conteúdo, nem o protocolo usado na troca do conteúdo entre um sistema e outro. Este projeto concentra-se na especificação da descrição da meta-informação e na descrição do próprio conteúdo a ser trocado, não estabelecendo, assim, o protocolo que viabilizará a interoperabilidade dos mesmos.

### 3.2.3 O que vamos utilizar do projeto

Visando à interoperabilidade de conteúdos educacionais, o *framework* ContentNet baseia-se na "IMS Learning Resources Meta-data Specifications". O *framework* utiliza a descrição de meta-informação proposta no projeto IMS para possibilitar a busca e a troca de conteúdos educacionais espelhados em diversos servidores de conteúdo. A partir deste momento, quando for mencionado o padrão IMS, subentender-se-á o padrão para descrição de meta-informação.

Reforçando o conceito de meta-informação que será usado adiante, encontra-se, a seguir, um resumo da descrição utilizada no padrão IMS.

#### Meta-informação

De acordo com o padrão proposto pelo IMS, as meta-informações são organizadas em categorias ou campos. Estes campos representam as características dos recursos. Cada

recurso, por sua vez, é descrito de acordo com a informação contida na combinação destes campos.

O processo de criação das meta-informações é expresso por esquemas. Os *esquemas* são estruturas lógicas construídas, a partir de *elementos*, de acordo com regras específicas que descrevem um conjunto particular de meta-informações e valores que serão usados para expressar a informação. Um esquema identifica, portanto, as propriedades, isto é, os elementos e as características destas propriedades que se aplicam a um recurso didático particular. Existem, ao todo, cinco esquemas de meta-informações: *Base*, *Item*, *Module*, *Tool* e *Master*<sup>2</sup>, sendo que o esquema *Master* apenas agrupa as propriedades e as características de todos os recursos didáticos. Os esquemas *Item*, *Module* e *Tool*, derivados do tipo *Base*, estendem este esquema, pois incluem outros elementos.

- *Base*: Implementação mínima do esquema *Master*. Este esquema cria um conjunto mínimo, presente em todos os esquemas, de propriedades necessárias para validar a meta-informação.
- *Item*: Descreve um elemento como, por exemplo, uma figura, um áudio, um vídeo. Um elemento é, tipicamente, um arquivo com o formato GIF, JPG, HTML, TXT e etc..
- *Module*: Provê relevantes meta-informações educacionais. O tipo *Module* também provê mais propriedades opcionais do que o tipo *Base*.
- *Tool*: Descreve ferramentas e funções utilizadas pelo usuário. Exemplos de conteúdos seriam uma calculadora e um editor de texto.

Devido ao número de elementos/campos existentes em uma dada meta-informação, estas foram agrupadas formando um total de nove grupos. Cada grupo representa um contexto descrito pelos elementos que são mais de sessenta só no esquema *Base*. Mais detalhes sobre a definição dos elementos da meta-informação podem ser obtidos em [IMS Metadata, 2000] e em [Silveira, 2000b].

---

<sup>2</sup> Não será traduzida a terminologia do padrão IMS para assegurar a aderência à padronização

O padrão IMS ainda não está completamente definido. Novos esquemas e elementos podem ser criados de acordo com as demandas provenientes da utilização do padrão. Para solucionar este problema, o padrão IMS propôs uma forma de armazenamento da meta-informação, independente do número de esquemas e elementos da meta-informação. Baseando-se nessa forma de armazenamento, em TIERS[TIERS, 1999], a estrutura da meta-informação foi armazenada.

Para a representação desta meta-informação, o padrão utiliza o formato "Resource Description Framework/eXtensible Markup Language" (RDF/XML). RDF[W3C RDF, 1999][RDF Schema, 1999][RDF Model, 1999] é um modelo usado para descrever informações sobre informações na *Internet*. Este modelo foi projetado para prover uma infra-estrutura de suporte às meta-informações que utilizam XML, formando uma comunidade de meta-informações por meio de muitas atividades baseadas na *Web*. XML é uma linguagem de marcação simples e muito flexível baseada em SGML[SGML, 1999], que possibilita a troca de informações presentes na *Web*, pois define a formatação de páginas HTML e a descrição de recursos.

Duas ferramentas foram utilizadas na manipulação da meta-informação: "LOM-IMS XML Toolkit"[Sun, 2000], para criação de arquivos XML contendo meta-informações de acordo com o padrão IMS, e "IBM's XML Parser for Java™ (XML4J)" [IBM, 2000], *parser* de arquivos XML para a linguagem Java™ utilizada no desenvolvimento do *framework* ContentNet.

## Capítulo 4

### **Framework ContentNet**

O *framework* ContentNet desenvolveu-se a partir de um estudo sobre as vantagens e desvantagens do desenvolvimento de *frameworks* para a construção de uma família de aplicações pertencentes a um mesmo domínio. Neste capítulo, apresentam-se a descrição do *framework* ContentNet, proveniente da análise do domínio da aplicação, assim como sua arquitetura e seus diagramas de classe.

#### **4.1 Introdução a Framework**

Um *framework* é um conjunto de classes que representa o *design* abstrato de uma família de problemas relacionados. Um *framework* é uma arquitetura projetada para possibilitar a reutilização máxima. Ele é representado por uma coleção de conjuntos abstratos e concretos de classes encapsuladas de maneira que subclasses possam ser especializadas para uma dada aplicação[Mattsson, 1996].

Um *framework* está conectado a um domínio específico e consiste em um modelo abstrato do domínio, um *design* reutilizável, e uma implementação do *design*. [Fayad, 1999].

Um *framework* deve ser utilizado como o ponto de partida para o desenvolvimento de um conjunto de aplicações relacionadas ao mesmo domínio e deve ser visto como um conjunto de classes que possuem funcionalidades reutilizáveis para um domínio específico de aplicações.

O *kernel* do *framework* representa a similaridade entre as aplicações enquanto o comportamento específico é dado pelos pontos de flexibilização ou *hot-spots*[Fontoura,1999]. O *kernel* do *framework* capta o controle geral da aplicação enquanto os pontos de flexibilização devem ser instanciados para um certo domínio da aplicação. Os *hot-spots* expressam aspectos do domínio do *framework* que não puderam ser antecipados no seu *design*. Na Figura 1, as classes que correspondem ao *kernel* estão envoltas por um quadrado, e as que correspondem ao *hot-spot* estão representadas em cinza.

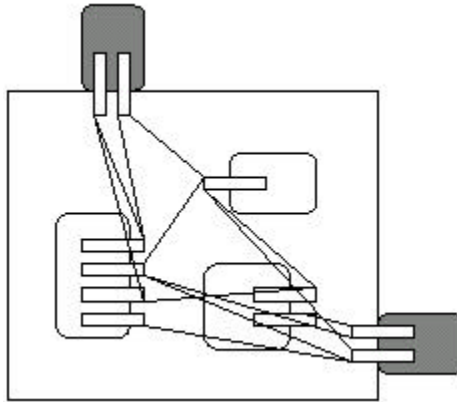


Figura 1 - *Kernel* e *hot-spots*

Os objetivos principais de um processo de desenvolvimento de *software* baseado em *frameworks* são:

- Manter o conhecimento da organização sobre o domínio da aplicação;
- Minimizar o número de linhas de código necessárias para implementar aplicações similares; isto é possível porque o *framework* abstrai as partes comuns das aplicações reduzindo a fração que deve ser implementada;
- Otimizar a generalização; para isto é necessário que o usuário do *framework* utilize-o em aplicações para as quais é destinado, e é necessário que o *framework* inclua um conjunto predefinido de subclasses que possam ser usadas sem a necessidade de alterá-las.

A utilização de *frameworks* traz as seguintes vantagens:

- Usabilidade do *design* e não só do código;
- Código já escrito e compilado;



- Mais foco em áreas específicas e menos foco em áreas comuns a outras aplicações; o usuário do *framework* concentra-se no desenvolvimento das áreas específicas à sua aplicação;
- Minimizar o número de linhas do código que devem ser implementadas se a aplicação requerer funcionalidades que são capturadas pelo *framework*.

Os elementos comuns no processo de desenvolvimento de um *framework* estão descritos na Figura 2:

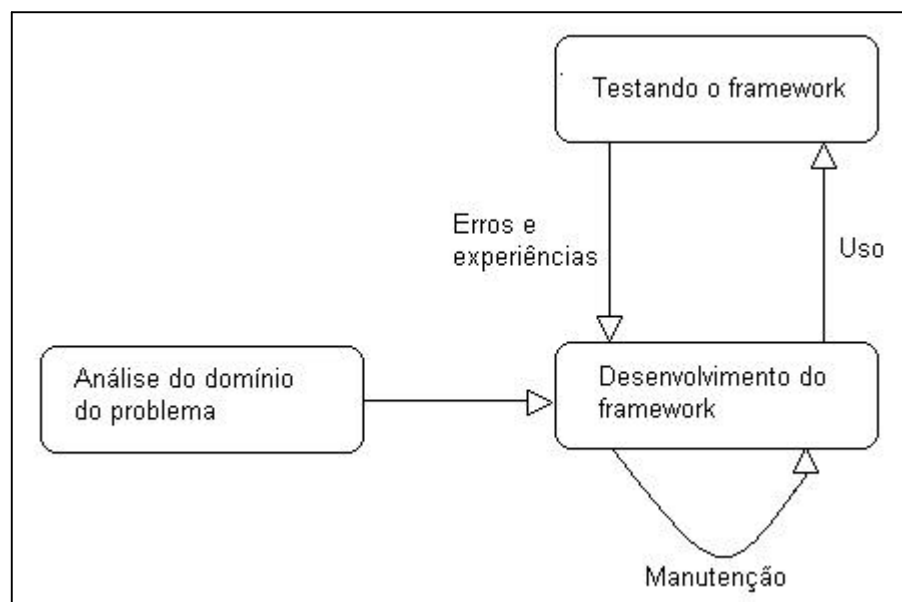


Figura 2 - Processo de desenvolvimento de *framework*

A análise do domínio do problema é o ponto de partida para o desenvolvimento de um *framework*. A partir das abstrações descobertas na análise do domínio, desenvolve-se sua primeira versão. Em seguida, deve ser desenvolvida uma ou mais aplicações baseadas no *framework*. Com isso, é possível testar se o *kernel* e os *hot-spots* do *framework* foram bem projetados e, assim, os problemas descobertos usando o *framework* são capturados e resolvidos na próxima versão. Depois de repetir esse ciclo algumas vezes, o *framework* adquire maturidade e pode ser usado para o desenvolvimento de aplicação no mesmo domínio[Fayad, 1999].

O *framework* ContentNet desenvolveu-se com base no processo descrito acima. Primeiramente, o domínio da aplicação que se desejava construir foi analisado e, em seguida,

produziu-se a primeira versão do *framework*. A fim de testar o *kernel* e os *hot-spots* projetados, foi instanciada uma aplicação, que será vista no próximo capítulo.

## 4.2 Descrição do *framework* ContentNet

O *framework* ContentNet visa resolver os problemas relativos a localizar, a avaliar, a prover acesso, e a manipular as informações disponíveis em servidores de conteúdo. O objetivo do ContentNet é encontrar, baseando-se em sua descrição, um conteúdo e transferi-lo do servidor de origem para o servidor de destino. Para tanto, o ContentNet utiliza o conceito de rede de servidores que publicam conteúdos, e que possibilitam o acesso a estes conteúdos a partir de um servidor de pesquisa. O servidor de pesquisa tem acesso a estes servidores de conteúdo utilizando um protocolo bem conhecido entre eles. Na Figura 3, está exemplificada a rede de servidores para a interoperabilidade de conteúdos.

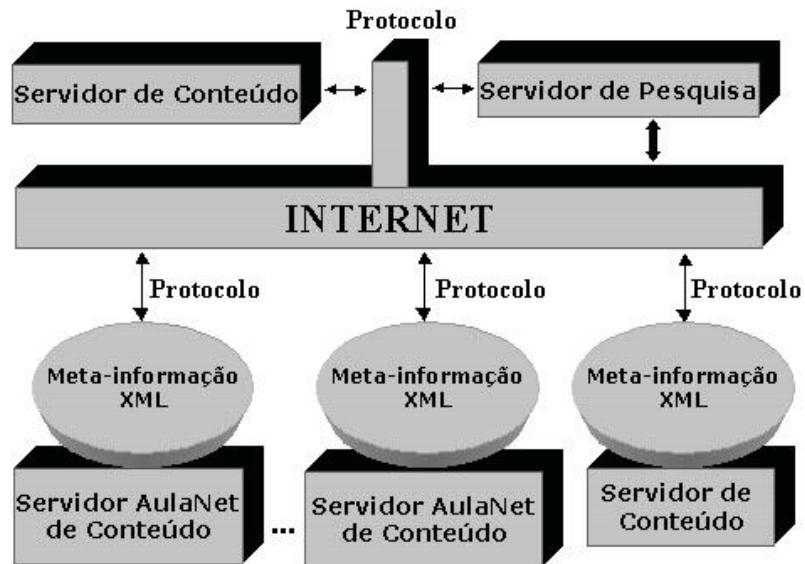


Figura 3 - Interoperabilidade de conteúdo

Para que o servidor de pesquisa conheça todos os servidores de conteúdo, cada servidor de conteúdo deve informar, ao servidor de pesquisa, que deseja fazer parte da rede de pesquisa ContentNet. O servidor de pesquisa, por sua vez, armazena uma lista dos servidores de conteúdo que fazem parte da rede ContentNet. Isto possibilita, a um usuário de um servidor

de conteúdo pertencente a rede ContentNet, ter acesso ao servidor de pesquisa, com o objetivo de localizar e obter um conteúdo que esteja localizado em um dos demais servidores de conteúdo pertencentes à rede. O servidor de pesquisa, por sua vez, contata os servidores de conteúdo cadastrados, procurando pelo conteúdo desejado pelo usuário. Os servidores de conteúdo fornecem, então, uma lista com descrições dos conteúdos similares que possuem. O servidor de pesquisa grupa as listas recebidas e fornece as opções ao usuário. O usuário escolhe, entre as opções, a que deseja, e o servidor de pesquisa busca o conteúdo no servidor de conteúdo que forneceu a opção. De posse do conteúdo, o servidor de pesquisa envia o conteúdo para o usuário, ou seja, para o servidor de conteúdo ao qual ele pertence.

### 4.3 Arquitetura utilizada

Como visto anteriormente, o *framework* ContentNet tem como objetivo a interoperabilidade de conteúdos entre quaisquer servidores de conteúdos educacionais que utilizem o padrão IMS. O processo de interoperabilidade de um conteúdo é complexo e pode ser dividido em 3 partes: encontrar o conteúdo, ter acesso a ele e cadastrá-lo.

Encontrar um conteúdo na *Web* não é tarefa fácil. Para que seja possível encontrar um conteúdo na rede, é necessário que se saiba descrever bem o conteúdo que se deseja e, tão importante quanto, é necessário saber onde procurá-lo. É necessário que os servidores de conteúdo sejam conhecidos e que utilizem o mesmo padrão de descrição de conteúdo. Uma vez encontrado o conteúdo, deve ser possível ter acesso a ele. O servidor que possui este conteúdo deve fornecer algum mecanismo de acesso, possibilitando, desse modo, que o pesquisador adquira o conteúdo. Junto com a possibilidade de aquisição, deve estar a possibilidade de cadastramento do conteúdo em outro servidor, efetivando, assim, a interoperabilidade do conteúdo.

A fim de cobrir o processo de interoperabilidade de um conteúdo, o *framework* ContentNet é composto de dois grandes módulos<sup>3</sup>: *Content Search* e *IServer*. O módulo *Content Search*,

---

<sup>3</sup> Módulo: Unidade planejada destinada a reunir-se ou ajustar-se a outras unidades análogas formando um todo homogêneo e funcional.

como o próprio nome sugere, encarrega-se do servidor de pesquisa enquanto que o módulo *IServer* encarrega-se da comunicação entre o servidor de conteúdo e o servidor de pesquisa. O *IServer*, *Intermediator Server*, envia, ao servidor de pesquisa, as informações fornecidas pelo servidor de conteúdo, *Content Server*, e envia, ao servidor de conteúdo, as informações disponibilizadas pelo servidor de pesquisa. Uma instância do módulo *IServer* deve ser criada sempre que um servidor de conteúdo desejar pertencer à rede ContentNet. Este módulo, ao ser instanciado, comunica, ao servidor de pesquisa, isto é, ao módulo *Content Search*, a existência de mais um servidor de conteúdo na rede ContentNet. Assim, o módulo *Content Search* precisa ser instanciado antes da primeira instância do módulo *IServer*.

Para que o *framework* possa ser instanciado em qualquer servidor de conteúdo educacional que utilize o padrão IMS, independentemente da forma de armazenamento dos dados, é necessário que, ao ser instanciado, sejam implementadas classes, junto ao módulo *IServer*, que façam a interface entre a instância do *framework* e o servidor de conteúdo. As classes implementadas no ponto de flexibilização do *framework* devem ser capazes de trazer as informações da base de dados do *Content Server* para a instância do *framework IServer*, e de gravar informações provenientes da pesquisa na base de dados do *Content Server*. Esta interface deve prover informações sobre a meta-informação de um conteúdo e sobre o conteúdo propriamente dito. A Figura 4 identifica onde estão os pontos de flexibilização mencionados.

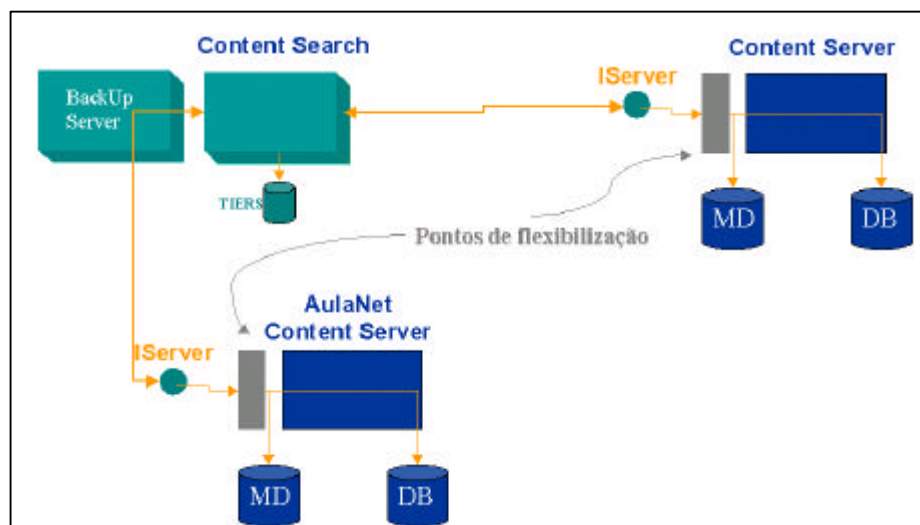


Figura 4 - Pontos de flexibilização. MD-base de meta-informação, BD-base de conteúdo

Por tratar-se de uma solução centralizada, faz-se necessária a existência de um *BackUpServer*, que deve assumir as ações do *Content Search*, sempre que este estiver impossibilitado.

O módulo *Content Search* e parte do módulo *IServer* encarregada do recebimento e do envio de informação fazem parte do *kernel* do *framework*. A outra parte do módulo *IServer*, que provê o acesso à base de dados do *Content Server*, buscando meta-informações ou conteúdos, e permitindo o cadastramento de um conteúdo proveniente da busca, compreende os pontos de flexibilização, ou *hot-spots* do *framework*. A Figura 5 ilustra o *kernel* do *framework* sinalizado por uma linha tracejada, e os pontos de flexibilização, representados por um caixa retangular, que comunica o *IServer* e o *Content Server*.

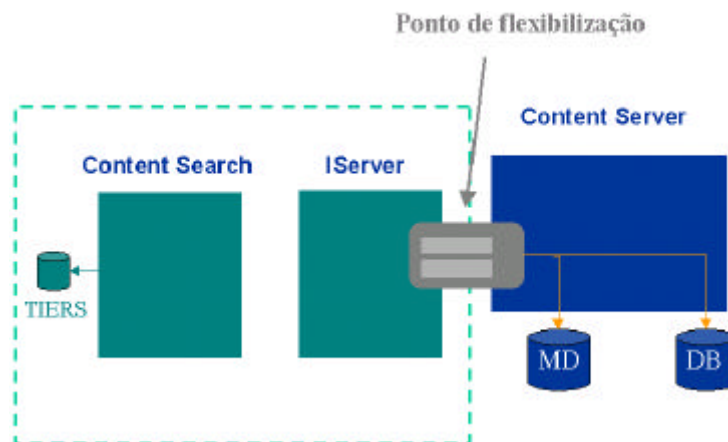


Figura 5 - *Framework: kernel* e pontos de flexibilização.

O módulo *Content Search* deve prover a busca por conteúdos educacionais<sup>4</sup>, localizados nos *Content Servers*. O usuário do *Content Search*, ao fazer uma busca por um conteúdo, gera uma lista com as características do conteúdo que deseja. Esta lista, meta-informação do conteúdo, é enviada para os *IServer* acoplados aos *Content Servers* que fazem parte da rede. Os *IServers* recebem um cópia da meta-informação e enviam-na para os correspondentes *Content Servers*. Quando o *Content Server* recebe uma meta-informação, ele procura, em sua base de dados, meta-informações com características semelhantes. Estas meta-informações são enviadas para o *IServer*, que as reenvia para o *Content Search*. Assim que o *Content Search* recebe todas as meta-informações enviadas pelos *IServers*, ele disponibiliza, para o usuário que está fazendo a busca, a lista das meta-informações. O usuário, por sua vez,

<sup>4</sup> Um conteúdo é composto de um ou mais arquivos.

seleciona a meta-informação que possui as mesmas características do conteúdo que deseja. O *Content Search* busca, então, o conteúdo desejado no servidor de conteúdo que enviou tal meta-informação selecionada.

Os módulos *Content Search* e *IServer* foram divididos em pequenas partes conforme a arquitetura cliente-servidor [Tanenbaum, 1995]. A essas partes, foi dado o nome de processo<sup>5</sup>. Os processos servidores estão destinados a receber mensagens, enquanto que os clientes estão destinados a enviar mensagens. Os servidores são processos que ficam sempre ativos desde a instanciação do módulo a qual pertencem. Já os processos clientes são chamados à medida que seus módulos necessitam enviar uma mensagem. O fluxo de mensagem trocada entre os módulos está representado na Figura 6.

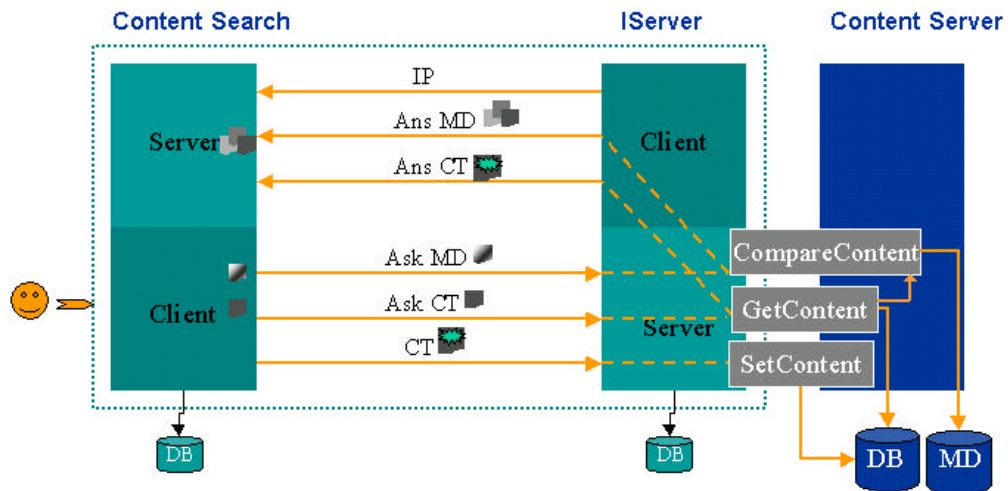


Figura 6 -Classes do ponto de flexibilização: CompareContent, GetContent, SetContent

Lista das mensagens trocadas entre o módulo *Content Search* e o módulo *IServer*:

- "IP" → Enviada pelo *IServer* quando este é instanciado e o *Content Server* deseja fazer parte da rede de busca do *Content Search*;
- "AskMD" → Pergunta enviada do *Content Search* para todos os *IServers* pedindo meta-informações similares a que o usuário deseja. O *IServer* recebe a meta-informação e contata a implementação do *hot-spot CompareContent* que fornece as meta-informações similares encontradas na base de dados do *Content Server*;

<sup>5</sup> Processo: programa em execução que possui espaço de endereçamento.

- "*AnsMD*" → Mensagem enviada do *IServer* para o *Content Search* como resposta a pergunta "*AskMD*", fornecendo a lista das meta-informações similares;
- "*AskCT*" → Pedido de um determinado conteúdo descrito na meta-informação que está sendo enviada junto com a mensagem. O *IServer*, que recebe a mensagem enviada pelo *Content Search*, contata a implementação do *hot-spot GetContent* que disponibiliza o conteúdo que possui tal meta-informação;
- "*AnsCT*" → Mensagem enviada do *IServer*, fornecendo o conteúdo ao *Content Search*;
- "*CT*" → Mensagem que leva o conteúdo do *Content Search* para o *IServer* que deseja o conteúdo. O *IServer* contata a implementação do *hot-spot SetContent* enviando o conteúdo que é cadastrado na base de dados do *Content Server*.

Os *hot-spots*, *CompareContent*, *GetContent* e *SetContent* são classes implementadas na instanciação do módulo *IServer* de acordo com a forma de armazenamento da meta-informação e do conteúdo na base de dados do *Content Server*. O *hot-spot CompareContent* tem como finalidade receber uma meta-informação e fornecer uma lista com meta-informações similares encontradas na base de dados. A classe *GetContent* deve disponibilizar o conteúdo cuja meta-informação foi recebida, e a classe *SetContent* deve cadastrar o conteúdo recebido.

#### 4.4 Diagrama de classes e de interação

Nesta seção, estão representados o diagrama de classes do *framework*, ou seja, o diagrama do módulo *Content Search*, e o diagrama do módulo *IServer*, antes de sua instanciação. Também serão apresentados os diagramas de interação dos processos cliente e servidor do módulo *IServer*. Os diagramas de interação dos processos do módulo *Content Search* não foram acrescentados, pois são muito similares aos diagramas de interação do módulo *IServer*.

#### 4.4.1 Módulo IServer

De acordo com o que foi visto em seções anteriores, o módulo *IServer* foi dividido em duas partes, as quais foram dados os nomes processo cliente e processo servidor. Em seguida, encontram-se a descrição das classes implementadas nesses dois processos e seus diagramas de interação e, por fim, o diagrama de classe do módulo *IServer*, sinalizando as classes que compreendem cada processo.

##### Processo Cliente

O objetivo deste processo, representado pela marcação com a linha tracejada na Figura 9, é enviar mensagens para o servidor *Content Search*. A classe *Client*, classe principal deste módulo, é a responsável pelo gerenciamento das mensagens. No envio de uma mensagem, a classe *Client* comunica-se com a classe *ConnectionC* para estabelecer uma conexão via *socket* com o servidor, e para criar os canais de comunicação mediante o acesso a classe *ChannelC*. Com a conexão estabelecida e os canais de comunicação criados, a mensagem é enviada por uma das classes derivadas da classe *Message*, conforme o tipo de mensagem a ser enviada. A classe *MsgAnsMD* envia a mensagem "*AnsMD*" junto com uma lista de meta-informações, a classe *MsgAnsCT* envia a mensagem "*AnsCT*" e um conteúdo, enquanto a classe *MsgIP* envia a mensagem "*IP*" fornecendo o endereço IP da máquina *Content Server*. A seguir, na Figura 7, encontra-se o diagrama de interação do processo cliente.



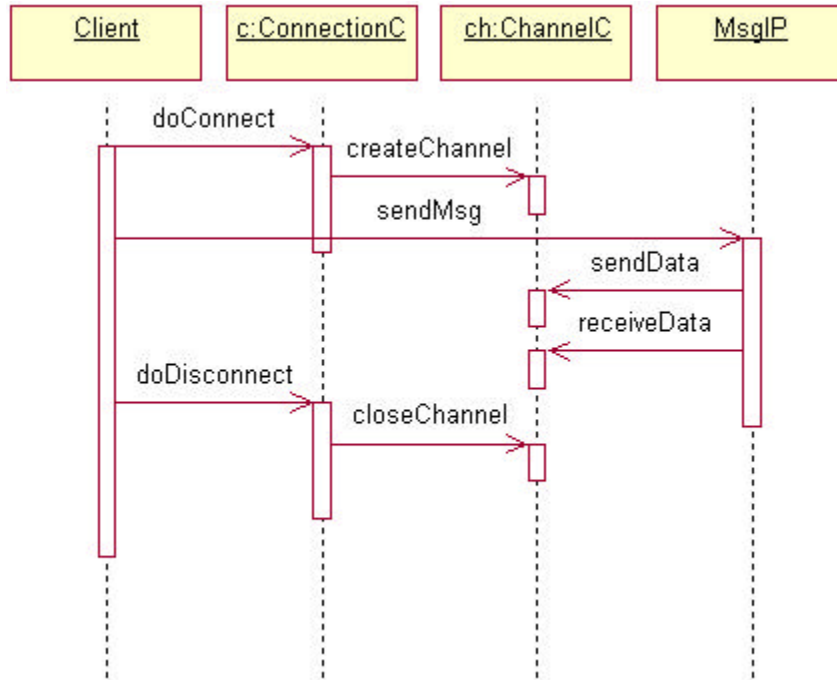


Figura 7 - Diagrama de Interação (processo cliente)

## Processo Servidor

O processo servidor, marcado pela linha pontilhada na Figura 9, é responsável pelo atendimento às solicitações feitas pelo módulo cliente do *Content Search*. A classe principal *Server* encarrega-se de estabelecer a conexão com o cliente *Content Search* por meio da classe *ConnectionS*, que criará um processo secundário, que ficará responsável por atender à solicitação. Assim, o processo principal volta a esperar uma nova solicitação enquanto o processo secundário, por meio da classe *ServerThread*, encarrega-se da nova solicitação. A classe *ServerThread* cria os canais de comunicação com o cliente utilizando a classe *Channel* e verifica o tipo de solicitação que o cliente *Content Search* pede na mensagem enviada. Se a mensagem enviada tiver sido "AskMD", a classe *CompareContent* é chamada. Esta classe recebe uma meta-informação e responde com zero ou mais meta-informações similares armazenadas na base de dados do *Content Server*. Se a mensagem tiver sido "AskCT", a classe *GetContent* é chamada. Esta classe recebe uma meta-informação e fornece o conteúdo, nela, descrito. E se a mensagem tiver sido "CT", é chamada a classe *SetContent* que recebe como parâmetro o conteúdo a ser cadastrado na base de dados do *Content Server*. No caso das classes *CompareContent* ou *GetContent* terem sido chamadas, a *ServerThread* chama, então,

a classe *Client* do processo cliente para enviar a informação de retorno das classes. O diagrama de interação deste processo encontra-se na Figura 8.

As classes *CompareContent*, *GetContent* e *SetContent* são classes abstratas, desse modo, não possuem implementação e devem ser implementadas de acordo com a instância do *framework*. Estas classes, *hot-spots* do módulo *IServer*, estão representadas por círculos na Figura 9.

Com o propósito de auxiliar na instanciação, foram usados os estereótipos *incomplete*, *extensible* e *unique*. O estereótipo *unique* indica que deve ser implementada uma única classe na instanciação. O *incomplete* possui significado parecido com o de UML padrão e indica que novas classes que satisfazem um relacionamento (generalização ou realização) podem ser adicionadas durante a instanciação do *framework*. O *extensible* indica que, na instanciação, novos métodos podem ser definidos para estender a funcionalidade da classe. Os estereótipos *incomplete* e *extensible* foram usados de acordo com a definição de Fontoura [Fontoura, 1999].

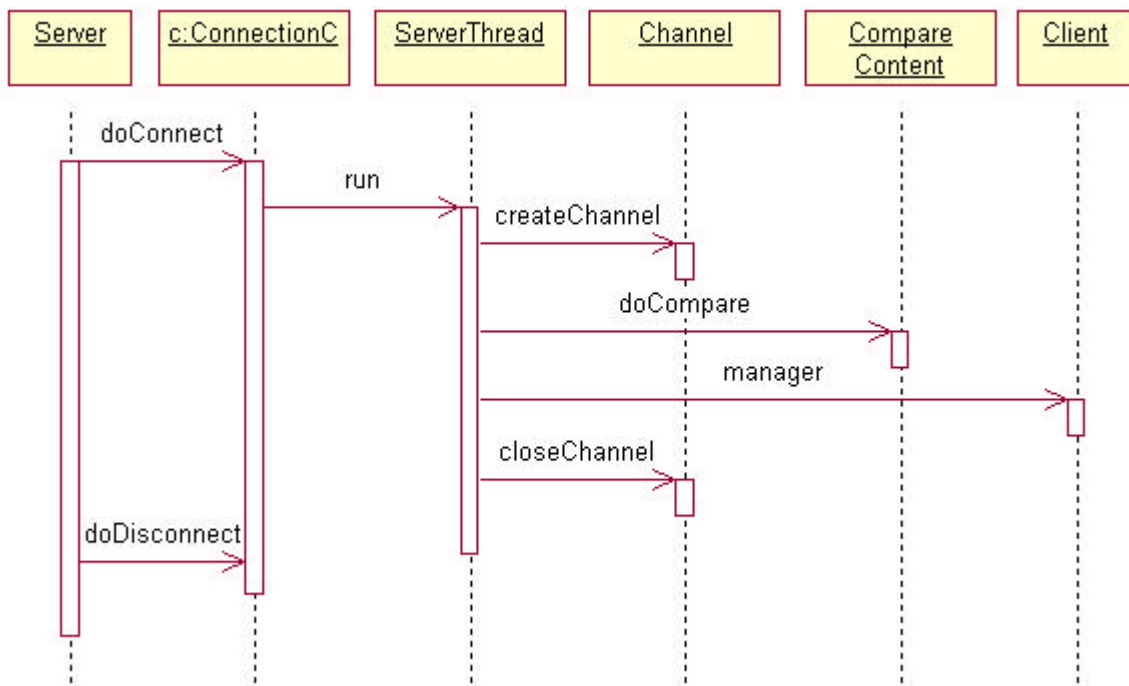


Figura 8 - Diagrama de Interação (processo servidor)

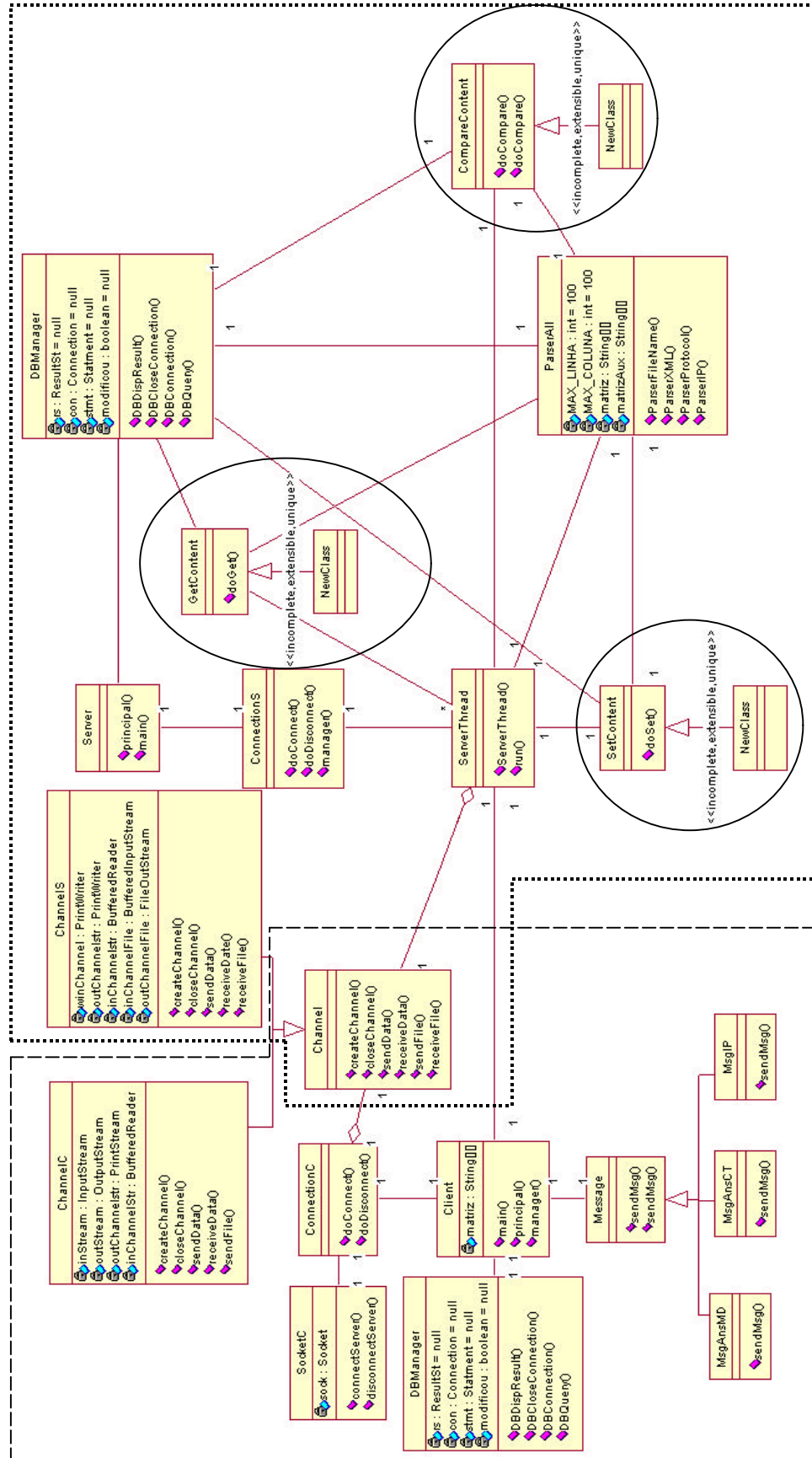


Figura 9 - Módulo IServer

#### 4.4.2 Módulo Content Search

Os processos cliente e servidor, divisões feitas no módulo *Content Search*, conforme visto anteriormente, estão descritos, a seguir, juntamente com as classes utilizadas em cada processo. Por fim, encontram-se o diagrama de classe do módulo *Content Search* e as sinalizações das classes utilizadas em cada processo.

##### Processo Cliente

O cliente do módulo *Content Search* é responsável pelo envio de mensagens para o servidor *IServer*. O processo cliente deste módulo é muito similar ao processo cliente do módulo *IServer* diferindo, no entanto, nas mensagens enviadas. Neste caso, as mensagens enviadas podem ser "*AskMD*", pedindo por uma lista de meta-informações, enviada pela classe *MsgAskMD*; "*AskCT*", pedindo por um conteúdo, enviada pela classe *MsgAskCT*; e "*CT*", enviando um conteúdo para ser armazenado na base de dados do *Content Server*, enviada pela classe *MsgCT*. A Figura 10 ilustra o processo cliente com uma marcação de linha tracejada.

##### Processo Servidor

Este processo também é bastante similar ao processo servidor do módulo *IServer*. A diferença está no tratamento das mensagens recebidas. O processo servidor deste módulo não chama a classe *Client* do processo cliente. Quando a mensagem "*IP*" é recebida, o endereço recebido, junto com a mensagem, é armazenado na lista dos servidores que fazem parte da rede ContentNet. Nas outras mensagens, "*AnsMD*" e "*AnsCT*", este processo apenas armazena a informação recebida em lugares conhecidos pelo módulo de interface do catalogador descrito no Capítulo 6. O módulo de Interface do catalogador cuidará destas mensagens recebidas pelo processo servidor, que está sinalizado com uma linha pontilhada na Figura 10.

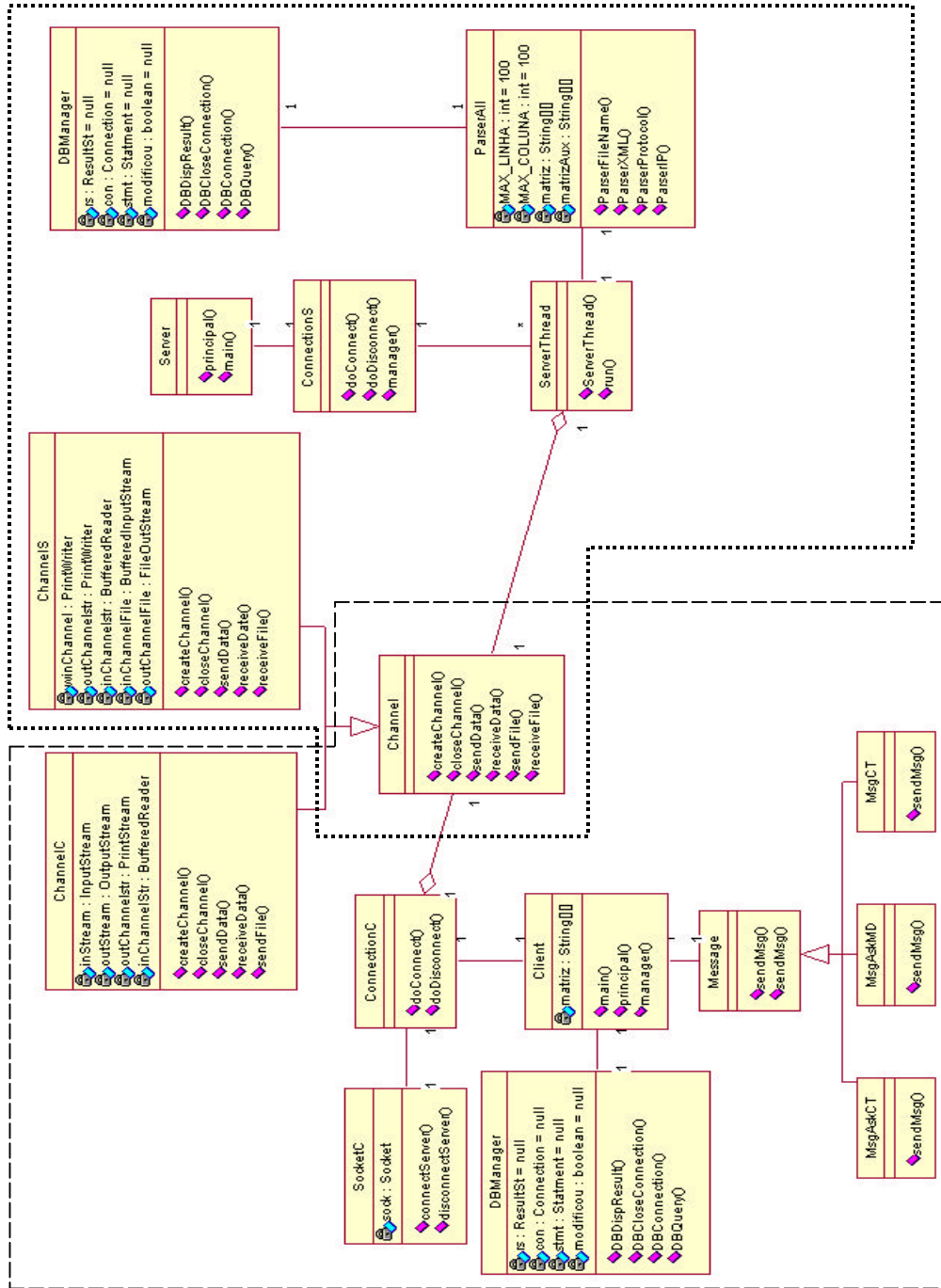


Figura 10 - Módulo *Content Search*

## Capítulo 5

### Instanciação do *Framework* ContentNet

O *framework* ContentNet foi instanciado a fim de verificar sua viabilidade testando seu *design*, no que diz respeito à definição do *kernel* e dos *hot-spots*, e testando sua implementação. Na instanciação, foram usados 3 servidores AulaNet e um servidor de pesquisa, já que a solução centralizada foi adotada no *design* do *framework*.

A instanciação do *framework*, como mencionado no Capítulo 4, prevê duas etapas. A primeira etapa é a instanciação da parte do *framework* que irá compor o servidor de pesquisa, isto é, a instanciação do módulo *Content Search*. A segunda etapa da instanciação corresponde ao módulo *IServer* nos servidores de conteúdo que farão parte da rede ContentNet. No teste realizado, foram instanciados 3 módulos *IServer*, um em cada servidor AulaNet.

#### 5.1 - Módulo Content Search

A instanciação deste módulo é mais simples do que a do módulo *IServer*, no que diz respeito aos *hot-spots*. Como o módulo *Content Search* não possui pontos de flexibilização, não é necessária, então, a implementação de nenhum *hot-spot*. Contudo, é necessária a implementação da interface do pesquisador acoplada a este módulo. Esta interface é implementada a fim de fornecer, ao processo cliente do *Content Search*, mensagens para serem enviadas aos *IServers*. Ao instanciar o módulo *Content Search*, seu processo servidor prepara-se para receber mensagens enviadas pelos *IServers*, só terminando sua execução

quando o servidor de pesquisas for desligado. Já o processo cliente somente começa sua execução quando há necessidade de envio de mensagem para os *IServers*. Uma vez enviada a mensagem, o processo cliente termina sua execução.

A primeira mensagem que o processo servidor recebe do módulo *IServer*, ou seja, do servidor de conteúdo, é a mensagem informando que este deseja fazer parte da rede de busca. Outras trocas de mensagem só surgem depois que um usuário do *Content Search* inicia sua pesquisa, estimulando o processo cliente pelo uso da interface.

## 5.2 - Módulo *IServer*

O módulo *IServer* deve ser instanciado junto a um servidor de conteúdo. Este módulo destina-se a fazer a comunicação entre o servidor de pesquisa e o servidor de conteúdo. Nesta comunicação, o módulo *IServer* recebe as mensagens do servidor de pesquisa e aciona uma das implementações dos três *hot-spots*, *CompareContent*, *GetContent* e *SetContent*, de acordo com a mensagem recebida. Desse modo, ao instanciar este módulo, os três *hot-spots* devem ser implementados de acordo com a forma de armazenamento do conteúdo e de sua meta-informação no servidor de conteúdo. Assim como ocorre no módulo *Content Search*, o processo servidor começa sua execução ao ser criada a instância do módulo, e só termina sua execução quando o módulo *IServer* for parado. Já o processo cliente é estimulado pelo processo servidor quando este deseja enviar mensagens e não é estimulado por uma ação do usuário.

A seguir, encontram-se detalhes sobre como deve ser a implementação dos pontos de flexibilização e detalhes da implementação utilizada nos servidores de conteúdo AulaNet. Todos os três *IServers* instanciados nos servidores AulaNet tiveram a mesma implementação no que diz respeito aos três *hot-spots*.

### 5.2.1 - *CompareContent*

A classe *CompareContent*, implementação do *hot-spot CompareContent*, tem como objetivo fazer a comparação entre meta-informações e fornecer as similares. Esta classe

abstrata possui um método chamado *doCompare*, que recebe como parâmetro uma meta-informação e o tipo do conteúdo descrito na mesma. A implementação deste método deve comparar a meta-informação recebida com as meta-informações do mesmo tipo existentes na base de dados do servidor de conteúdo. De acordo com o resultado da comparação, o método deve retornar zero ou mais meta-informações equivalentes à recebida.

A implementação do *hot-spot* deve fornecer uma classe que possua o relacionamento de herança com a classe *CompareContent*. A classe proveniente da instanciação do *hot-spot* deve herdar o método *doCompare* e implementá-lo. Durante a instanciação, de acordo com o servidor AulaNet, a classe implementada foi a *CompareAulaNetContent*. A classe *CompareAulaNetContent*, além de implementar o método herdado *doCompare*, implementou outros métodos e instanciou outras classes.

O método *doCompare*, implementado de acordo com o AulaNet, utiliza um algoritmo de comparação[Cormen, 1990b] que compara cadeias de caracteres, *strings*. Este algoritmo compara um *string* com o outro, e se eles forem 65% ou mais similares, os *strings* são, então, considerados. A percentagem 65% foi usada devido a problemas que possam ocorrer na digitação e à comparação entre dois textos complexos, que não necessitam ser exatamente iguais para serem considerados. O algoritmo lê a meta-informação recebida e busca o *string* de um elemento, o *string* de um dos atributo da meta-informação, e compara-o com o *string* do mesmo atributo de outra meta-informação, verificando, em seguida, se são 65% similares. Por exemplo, o algoritmo compara o título descrito em uma meta-informação com o título descrito em outra meta-informação, sendo título um atributo da meta-informação. Após a comparação de todos os elementos, o algoritmo verifica se o número de ocorrência de similaridades entre os atributos é maior ou igual a 65%. Se for, conclui-se, então, que as meta-informações são similares. Por fim, o método *doCompare* retorna todas as meta-informações similares.

### 5.2.2 - GetContent

O objetivo da classe *GetContent*, implementação do *hot-spot* *GetContent*, é buscar um conteúdo na base de dados do servidor de conteúdo. A instanciação da classe *GetContent* deve ser feita pelo relacionamento herança. A classe implementada deve herdar a classe



*GetContent* e implementar o método *doGet*. O método *doGet* desta classe abstrata recebe como parâmetro o tipo do conteúdo desejado e a meta-informação deste conteúdo, e sua implementação deve retornar o conteúdo que possui tal descrição.

Na instanciação desta classe do *IServer* junto ao servidor AulaNet, criou-se a classe *GetAulaNetContent*, que herdou e implementou o método *doGet*. O método *doGet* verifica qual o tipo de conteúdo que deve procurar e percorre a meta-informação recebida em busca do atributo que identifica o conteúdo descrito pela meta-informação. De posse do identificador e do tipo do conteúdo é possível encontrá-lo na base de dados.

Um conteúdo AulaNet pode ser composto de um texto explicativo e zero ou mais arquivos, conforme descrito no Capítulo 2. O texto explicativo, que está armazenado no banco de dados AulaNet, mas não é um arquivo propriamente dito, é transformado em um arquivo com extensão XML e seu DTD é gerado. Este arquivo XML, juntamente com o DTD e os outros arquivos que formam o conteúdo, retorna como parâmetro de saída do método *doGet*.

### 5.2.3 - SetContent

Como o próprio nome sugere, o objetivo da classe *SetContent*, implementação do *hot-spot SetContent*, é cadastrar um conteúdo. O método *doSet* desta classe recebe, como parâmetro de entrada, o conteúdo a ser cadastrado e o seu tipo. Sua implementação deve cadastrar o conteúdo recebido.

A implementação deste *hot-spot* para os servidores AulaNet gerou a classe *SetAulaNetContent*, que herda da classe abstrata *SetContent* e implementa o método *doSet*. O método *doSet*, assim que recebe o tipo do conteúdo, verifica se este conteúdo é proveniente de servidores AulaNet. Caso seja, o método não só cadastra o conteúdo como também o texto explicativo recebido na mensagem. Para o auxílio neste cadastramento, a classe *SetAulaNetContent* implementa outros métodos.

Na Figura 11, encontra-se o diagrama de classes deste módulo com as devidas implementações dos *hot-spots*.

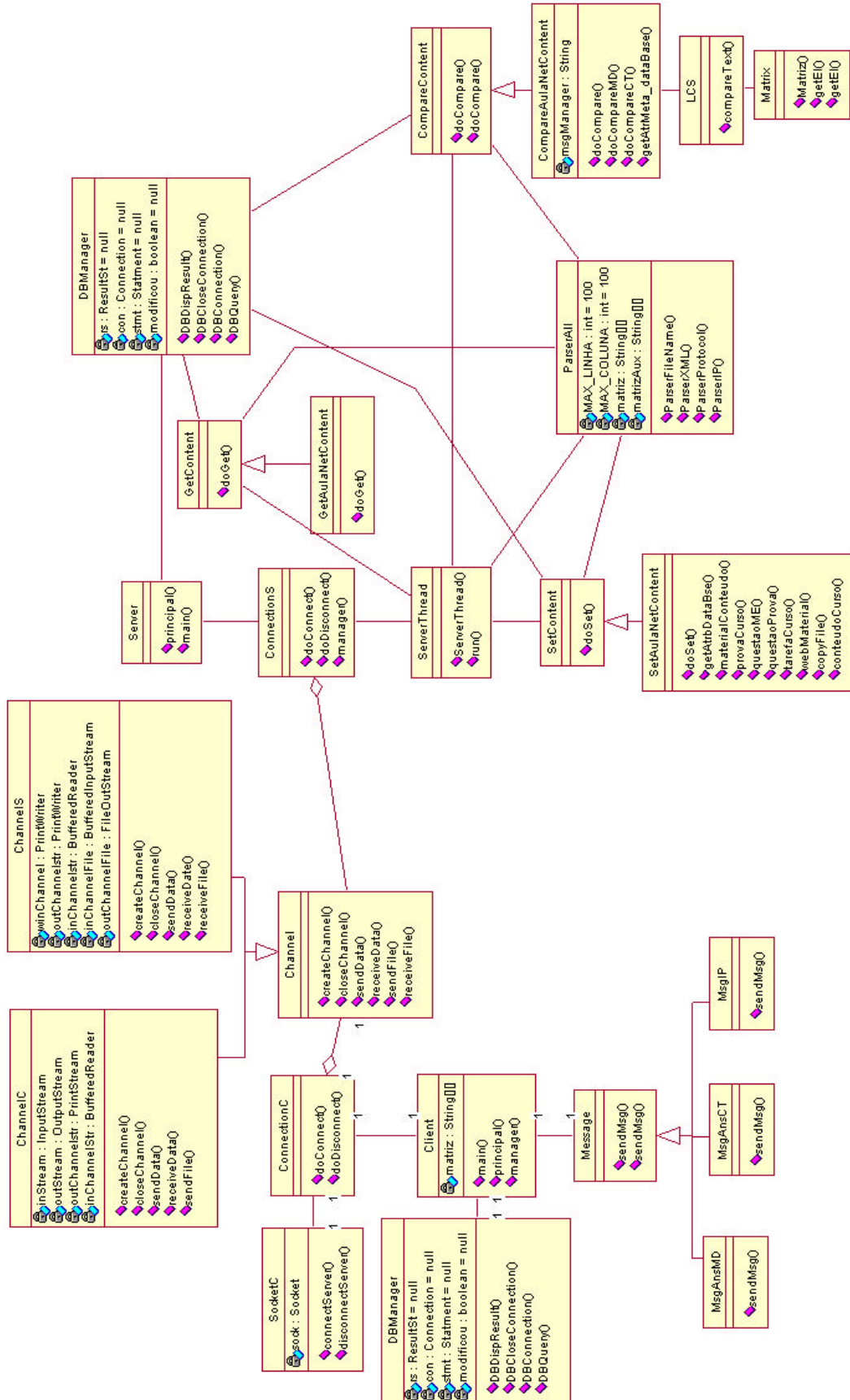


Figura 11 - Diagrama de classes da instância do módulo *IServer*

A instanciação do módulo *IServer* para conteúdos AulaNet provocou outra demanda além da implementação dos *hot-spots*. Houve a necessidade da criação de uma ferramenta capaz de cadastrar a meta-informação sobre os conteúdos existentes na base de dados AulaNet .

#### 5.2.4 - Ferramenta de cadastramento da meta-informação

O ambiente AulaNet, usado no teste do *framework* ContentNet, teve de ser adaptado. O *framework* ContentNet baseia-se no padrão IMS, o AulaNet, no entanto, não seguia o padrão. Este padrão pressupõe que todos os conteúdo cadastrados possuam uma meta-informação relacionada. Para a adaptação do ambiente AulaNet ao padrão, a ferramenta de cadastramento da meta-informação foi criada. Esta ferramenta, que não faz parte do *framework* ContentNet, possibilita que o catalogador cadastre meta-informação para todos os conteúdos existentes na base de dados. A ferramenta atua em duas etapas conforme ilustrado na Figura 12.

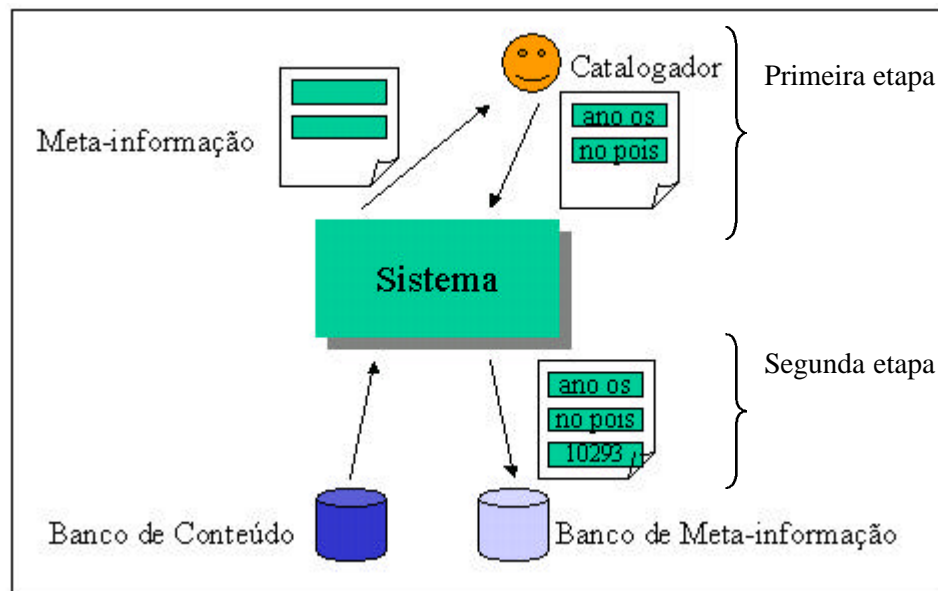


Figura 12 - Ferramenta de catalogação

Durante o processo de cadastramento da meta-informação, o sistema fornece ao catalogador um formulário contendo os atributos da meta-informação. Na primeira etapa, o catalogador preenche os atributos e envia o formulário para o sistema. Os atributos da meta-informação que necessitam de conhecimento sobre o conteúdo são cadastrados pelo catalogador. Na segunda etapa, o sistema acrescenta na meta-informação, preenchida pelo catalogador, o

atributo de identificação do conteúdo. Este atributo é catalogado pelo sistema, pois é um dado que só o sistema conhece.

De acordo com o padrão IMS, para que uma meta-informação seja válida, nem todos os elementos presentes na meta-informação de um conteúdo precisam ter um valor associado[IMS Metadata, 1999]. Uma meta-informação válida é aquela que descreve, minimamente, um conteúdo. Apenas os conteúdos que possuem meta-informação válida podem fazer parte da rede de pesquisa. Então, logo após a devida catalogação, o conteúdo passa a estar apto a ser pesquisado.

Os conteúdos catalogados são aqueles existentes na base de dados do ambiente. Contudo, os conteúdos provenientes da pesquisa, aqueles adquiridos por professores usando a ferramenta de pesquisa, não são catalogados pelo catalogador. Os conteúdos adquiridos pela ferramenta não poderão fazer parte da pesquisa novamente, por isso não são catalogados.

Sabendo-se como deve ser a instanciação dos módulos, encontra-se, na próxima seção, a descrição do fluxo de mensagem trocada, representado pelo diagrama de interação entre os processos associados a estes módulos.

### **5.3 - Diagrama de fluxo de mensagem**

Como visto anteriormente, o módulo *Content Search* é instanciado antes do primeiro módulo *IServer*. O processo servidor do módulo *Content Search* entra em execução antes do processo servidor do módulo *IServer*. A primeira mensagem que o processo servidor do módulo *Content Search* recebe é a mensagem "IP", enviada pelo processo cliente do módulo *IServer* a pedido do processo servidor deste módulo. Em seguida, as mensagens trocadas são provenientes do módulo *Content Search*. Mediante interação do usuário, o processo cliente envia as mensagens "AskMD" e "AskCT". Estas mensagens são tratadas e respondidas pelo *IServer*. A única mensagem enviada pelo módulo *Content Search* que não necessita de interação do usuário é a mensagem "CT". Ela é enviada, automaticamente, pelo processo cliente assim que o processo servidor do *Content Search* recebe o conteúdo e deseja enviá-lo para o servidor de conteúdo do professor.

Na Figura 13, a seguir, encontra-se o diagrama de fluxo de mensagens trocadas entre os módulos *Content Search* e *IServer*. Neste diagrama, estão explicitados o tempo de vida dos processos cliente e servidor dos módulos.

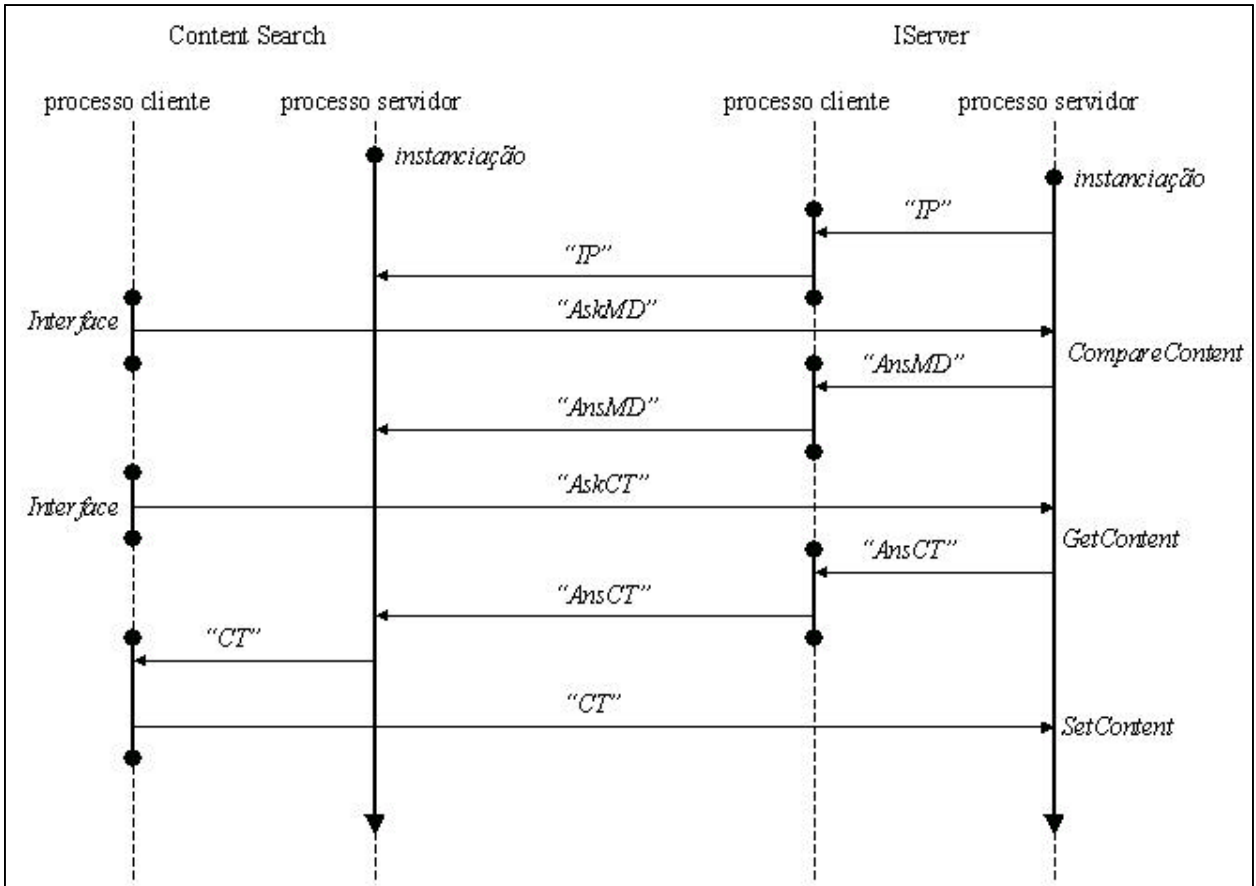


Figura 13 - Diagrama de fluxo de mensagem

## 5.4 - Outras instanciações

Na instanciação do *framework* ContentNet, seus *hot-spots* foram implementados de acordo com as características do ambiente AulaNet. As implementações destes *hot-spots* poderiam sofrer muita alteração, caso estes fossem instanciados para outros ambientes. A mudança na implementação dos *hot-spots* seria tão grande quanto as diferenças entre o ambiente AulaNet e o ambiente da instanciação.

A seguir, encontram-se detalhes sobre os fatores que podem influenciar na implementação dos *hot-spots*, conforme descrito na Tabela 2.

<b>Hot-spots</b>	<b>Fatores que podem influenciar na implementação</b>
<b>CompareContent</b>	Forma de armazenamento da meta-informação Método de comparação usado na meta-informação
<b>GetContent</b>	Forma de armazenamento da meta-informação Forma de armazenamento do conteúdo Método de identificação do conteúdo por meio da sua meta-informação
<b>SetContent</b>	Forma de armazenamento do conteúdo

Tabela 2 - Fatores de influência na implementação dos hot-spots

Os armazenamentos das informações (conteúdos) e das meta-informações (descrições dos conteúdos) são fatores que influenciam a implementação dos *hot-spots*. A forma de armazenamento destes itens depende da implementação do ambiente que pode ser tão diferente quanto se queira da forma de armazenamento dos itens no ambiente AulaNet. Como consequência, a busca e a aquisição da informação e da meta-informação dependem da implementação do ambiente.

Como exemplo, no ambiente AulaNet, as meta-informações estão armazenadas em arquivos. O banco de dados apenas fornece qual é e onde está o arquivo de meta-informação de tal conteúdo. Isto implica que, para acessar um atributo da meta-informação, seja necessário ir ao banco de dados colher os dados (qual é e onde está a meta-informação), e, em seguida, percorrer o arquivo da meta-informação até encontrar o atributo desejado.

Uma outra opção de armazenamento da meta-informação é não utilizar arquivos e cadastrá-la no banco. Os atributos da tabela do banco corresponderiam aos atributos da meta-informação. Cada entrada, ou linha, da tabela corresponderia a uma meta-informação. Assim, acionar um atributo significa encontrar a meta-informação que se deseja e, para aquela linha, "olhar" a coluna da tabela que corresponde ao atributo.

A seguir, encontra-se uma análise sobre outras possíveis implementações dos *hot-spots* *CompreContent*, *GetContent* e *SetContent* para ambientes cujas semelhanças com o AulaNet sejam pequenas.

#### **5.4.1 - CompareContent**

Além da forma de armazenamento da meta-informação, o método de comparação usado na implementação deste *hot-spot* pode ser diferente do utilizado para o AulaNet. No caso do AulaNet, são selecionadas as meta-informações dos conteúdos que fazem parte da pesquisa. Para estas, o método utilizado compara o *string* da meta-informação recebida com o *string* da meta-informação do banco, para um mesmo atributo, verificando se são semelhantes. Outro método poderia verificar se os *strings* são iguais. Assim, só as meta-informações exatamente iguais seriam consideradas. Ainda outra opção seria utilizar qualquer um dos outros métodos localizados em [Cormen, 1990a], por exemplo.

#### **5.4.2 - GetContent**

Tanto a forma de armazenamento da meta-informação quanto a forma de armazenamento do conteúdo influenciam este *hot-spot*. Outro fato que influencia sua implementação é como se faz a identificação do conteúdo por meio de sua meta-informação. No caso do AulaNet, o conteúdo é identificado por um atributo da meta-informação. Em uma nova implementação deste método, para identificar um conteúdo, pode ser necessário percorrer toda a meta-informação recebida comparando-a com todas as meta-informações existentes no banco de dados (ou em um arquivo de acordo com a forma de armazenamento).

Uma outra diferença está na composição do conteúdo. Para o AulaNet, um conteúdo pode ser formado por mais de um arquivo, sendo que um deles é gerado, dinamicamente, mediante consulta ao banco de dados. Para outra implementação, um conteúdo pode ser composto de apenas um arquivo, sem a necessidade de geração dinâmica.

### 5.4.3 - SetContent

Neste *hot-spot*, como sua tarefa é bastante simples, não há nenhum outro ponto que possa influenciar sua implementação além da forma de armazenamento do conteúdo. No caso do AulaNet, ao conteúdo armazenado, associa-se um campo identificando se este conteúdo foi adquirido por pesquisa ou se foi criado no AulaNet, no qual está armazenado. Para o AulaNet, conteúdo adquirido não fica disponível para pesquisa. Para pesquisa, ficam disponíveis apenas os conteúdos criados no próprio AulaNet. No caso de outra implementação, pode ser decidido que mesmo os conteúdos adquiridos fiquem disponíveis para pesquisa.



## Capítulo 6

### Interface do ContentNet

A interface descrita neste capítulo foi projetada para a instância do *framework* ContentNet e, desse modo, apoia-se nas características do *layout* da interface utilizada no ambiente AulaNet. A engenharia da interface baseou-se no uso de cenários. Com base nos cenários desenvolvidos para o professor e para o catalogador, foram construídas as duas interfaces utilizadas no ambiente. A fim de validar o *design*, testes foram aplicados tanto na interface projetada para o professor quanto na interface projetada para o catalogador. Neste capítulo, encontram-se os cenários estudados, a descrição da interface do professor e do catalogador, assim como os testes realizados.

#### 6.1 Introdução a Cenário

Cenário é uma descrição, em forma de narrativa, de uma atividade, ou seja, é uma descrição do que as pessoas fazem e do que experimentam quando usam uma aplicação ou um sistema [Carroll, 1995]. Cenários são descrições parciais do sistema e do comportamento do ambiente expressas em termos concretos de acordo com certas situações. Estas descrições são parciais, porque não especificam, necessariamente, todos os estados que compreendem um comportamento, nem especificam, necessariamente, todos os atributos do estado, mas, mesmo assim, projetam uma descrição concreta da atividade na qual o usuário encontra-se [Benner, 1993].

Cenários são instrumentos para descrever e esclarecer as propriedades relevantes do domínio da aplicação, descobrir requisitos ainda não observados do sistema, formar alternativas de *design* e validá-lo.

Na construção dos cenários, um ponto importante é o nível de abstração no qual o cenário é descrito. É crucial que cenários sejam expressos em termos concretos, ainda que em um nível arbitrário de abstração. A intenção do cenário é permitir que seu comunicador provenha apenas aquelas partes da descrição que são relevantes. Apesar de cenários incorporarem abstrações, é importante que não sejam tão abstratos e tão gerais a ponto de ignorarem fatores importantes na situação.

Outro ponto importante na construção dos cenários é a definição de seu foco. O foco é consequência de uma intenção do comunicador do cenário. Uma mudança de foco, feita pelo comunicador, poderá causar comportamentos indevidos e inesperados [Benner, 1993]. O comunicador do cenário deve levantar os pontos mais importantes da interface e fazer os cenários em cima destes pontos.

Cenários podem ser construídos pelo próprio *designer* da interface ou pelo usuário do sistema. Neste trabalho, os cenários foram desenvolvidos pelo *designer*. De acordo com Silveira et al., em [Silveira, 2000a], os usuários do sistema, geralmente, descrevem apenas cenários livres de erros, enquanto que os *designers* estão mais atentos a encontrar situações difíceis, nas quais ocorrem problemas, e possuem conhecimento de técnicas de manipulações de erros.

A fim de projetar as interfaces do ContentNet, estudaram-se os perfis dos usuários de cada interface, assim como as características do *software* e as possíveis expectativas do usuário em relação à interface. Estas informações estão descritas nas seções seguintes.

## **6.2 Interface do Professor**

Ao projetar a interface do professor, usuário da ferramenta de busca e aquisição do ContentNet, o primeiro ponto estudado foi o perfil deste usuário. Em seguida, foram desenvolvidos cenários com base neste perfil e, por fim, levando-se em consideração as

características da ferramenta e as expectativas que o *designer* supõe que o usuário tenha em relação à interface, esta foi projetada.

Os dados colhidos nas etapas do processo de engenharia da interface foram influenciados pelo ambiente AulaNet. No estudo do perfil dos usuários, dos cenários, das características da ferramenta e das expectativas do usuário, levou-se em consideração que a interface seria construída para a instância do *framework*. Estes dados não foram extraídos apenas do *framework*, mas também da sua instância. Dessa maneira, os dados colhidos não são, obrigatoriamente, necessários para a construção da interface de uma outra instância do *framework*.

### 6.2.1 Perfil dos usuários

Os usuários do ambiente AulaNet nem sempre são informados de todas as funcionalidades e ferramentas oferecidas pelo *software* antes de começarem a utilizá-lo. Eles não recebem treinamento e nem sempre utilizam-se do *tutorial* fornecido pelo ambiente. Com base nesta informação, foi possível definir grupos de usuários: os que conhecem o ContentNet, ou seja, que já receberam alguma informação de forma oral ou escrita, e os que não conhecem. Destes usuários, alguns podem ter utilizado o ContentNet, o que permite a definição de quatro grupos de usuários: os que conhecem e já utilizaram; os que conhecem e nunca utilizaram; os que não conhecem e que já utilizam sem saber; e os que não conhecem e nunca utilizaram. Os cenários descritos fundamentam-se nestes quatro grupos de usuários.

A ferramenta de busca e aquisição de conteúdo do ContentNet pode ser utilizada por pessoas que estão, realmente, interessadas em montar um curso no ambiente AulaNet ou pessoas que estão utilizando o ambiente para experimento. Tanto um grupo de usuários quanto o outro está interessado em pesquisar e em adquirir um conteúdo para montar um curso.

Estes usuários, em geral, são conhecedores do ambiente AulaNet, possuem informação sobre os tipos de conteúdos existentes no ambiente e sabem o que cada um significa. Os usuários não possuem conhecimento sobre meta-informação, nem sobre padrões para descrição de conteúdos. Não são conhecedores de informática, mas possuem algum conhecimento sobre *Internet*. São conhecedores de ferramentas de busca, sabem que, para buscar uma informação

utilizando uma ferramenta de busca, é necessário fornecer pelo menos uma palavra que identifique a informação desejada.

## 6.2.2 Cenários

Os cenários descritos para a interface do professor pesquisador de conteúdo, projetados pelo *design*, levam em consideração os perfis estudados. Foi descrita uma tarefa que abrangesse todo o processo de busca e aquisição de um conteúdo. Utilizando-se esta tarefa, quatro cenários foram construídos, um para cada perfil analisado.

Os cenários foram construídos de forma que a abstração não ignorasse pontos importantes da situação e que não descrevesse pontos que não fossem importantes. Além disto, tentou-se manter o foco na tarefa executada pelos usuários.

### Tarefa

Obter uma bibliografia cujas características sejam: para adquiri-la, nenhum pagamento é necessário; sua palavra-chave é mecânica; o idioma no qual ela foi escrita é o português; o grau de dificuldade no uso da bibliografia é alto.

### Cenário 1 (usuário não conhece e nem nunca utilizou a ferramenta de pesquisa)

João entra no ambiente AulaNet utilizando seu *login* e sua senha. Na lista de opções, localizada no menu principal, verifica a opção "Pesquisar e adquirir conteúdo". João, que está montando um curso sobre Mecânica Geral, e deseja obter uma bibliografia sobre o tema, verifica a possibilidade de pesquisar e adquirir um conteúdo no ambiente.

João seleciona a opção "Pesquisar e adquirir conteúdo". Na nova página, lê uma mensagem informando que a pesquisa no ContentNet é realizada por meio de seleção de um dos conteúdos localizados na lista de opções a seguir. Desta lista, João seleciona o conteúdo "Bibliografia" e uma nova página é gerada. Nesta nova página, João verifica uma lista de opções e uma mensagem. A mensagem informa que, para pesquisar um conteúdo, ele deve selecionar atributos dentro da lista de opções que farão parte da pesquisa. A mensagem

informa ainda que, selecionando o título do atributo, uma janela de ajuda abrir-se-á com informações sobre o atributo. Após ler as informações de ajuda de todos os atributos, João seleciona aqueles que deseja e continua a pesquisa. A página seguinte fornece um formulário com campos para valores a serem dados aos atributos selecionados. João sente dificuldade no preenchimento dos campos e seleciona, então, a ajuda localizada no título do atributo. Após o preenchimento dos campos, ele continua sua pesquisa em uma nova página.

Nesta nova página, uma lista de opções e uma mensagem aparecem, novamente, para João. A mensagem informa que, a seguir, encontra-se o resultado da pesquisa, isto é, as informações sobre os conteúdos provenientes da busca. João seleciona cada um dos resultados a fim de ver sua descrição. João verifica, então, que, dentre as opções, não está a bibliografia que deseja. João volta à página de seleção de atributos, faz uma nova seleção e a página do formulário é gerada novamente. Ele preenche o formulário e continua a pesquisa. Na página dos resultados, verifica que, devido ao preenchimento impróprio de um campo, ainda não consegue encontrar a bibliografia. Ele volta à página anterior, a com o formulário, e altera o preenchimento de um campo. No resultado da pesquisa, João, após ter lido as características/atributos de cada resultado, encontra a bibliografia que deseja. Seleciona, em seguida, esta bibliografia e segue para outra página. Nesta última página, é informado de que o conteúdo desejado encontra-se a sua disposição no seu ambiente AulaNet. João entende que conseguiu obter a bibliografia que desejava e finaliza a busca.

**Cenário 2 (usuário não conhece, mas já utilizou a ferramenta, apesar de não saber que estava utilizando uma ferramenta de busca)**

Maria entra no ambiente AulaNet utilizando seu *login* e sua senha. Na lista de opções, localizada no menu principal, verifica a opção "Pesquisar e adquirir conteúdo". Maria, que já havia selecionado esta opção, mas não entendera o seu objetivo, lembra-se de que, por esta opção, era possível selecionar um conteúdo e alguns atributos. Ela entende, então, que o conteúdo selecionado deveria ser o conteúdo a ser pesquisado. Maria resolve selecionar a opção "Pesquisar e adquirir conteúdo", pois está montando um curso sobre Mecânica Geral e deseja obter uma bibliografia sobre o tema.

Na nova página, seleciona o conteúdo "Bibliografia", após ler as explicações da mensagem localizada antes da lista de opções. Outra página é gerada e Maria depara-se com a lista dos atributos. Após ler algumas informações de ajuda, Maria seleciona os atributos que deseja. A página seguinte fornece um formulário com campos para valores a serem dados a estes atributos. Maria sente dificuldade no preenchimento de todos os campos e seleciona, então, a ajuda localizada no título dos mesmos. Maria verifica que selecionou mais atributos do que o necessário. Após o preenchimento dos campos necessários, ela continua a pesquisa.

Na nova página, uma lista de opções e uma mensagem aparecem novamente para Maria. A mensagem informa que, a seguir, encontra-se o resultado da pesquisa, ou seja, as informações sobre os conteúdos provenientes da busca. Maria seleciona cada um dos resultados a fim de ver sua descrição. Ela encontra o conteúdo que deseja e segue para uma nova página. Nesta última página, é informada de que o conteúdo desejado encontra-se a sua disposição no seu ambiente AulaNet. Maria entende que conseguiu obter a bibliografia que desejava e finaliza a busca.

Cenário 3 (usuário conhece, isto é, sabe da existência, mas nunca utilizou a ferramenta)

José, que leu o *tutorial* do ambiente AulaNet, sabe da existência da ferramenta de pesquisa. Porém, José não leu todo o *tutorial* sobre o assunto, leu, tão somente, a parte que informava sobre o preenchimento dos campos.

José entra no ambiente AulaNet utilizando seu *login* e sua senha. Ele procura entre as opções da lista, localizada no menu principal, a opção "Pesquisar e adquirir conteúdo". José, que está montando um curso sobre Mecânica Geral e deseja obter uma bibliografia sobre o tema, seleciona a opção "Pesquisar e adquirir conteúdo".

Na nova página, seleciona o conteúdo "Bibliografia" sem ler as explicações da mensagem localizada antes da lista de opções. Na nova página que é gerada, José depara-se com a lista dos atributos. Sem necessitar ler a mensagem, ele seleciona alguns dos atributos que deseja, mas ainda falta, entretanto, um atributo a ser selecionado. Como ele está em dúvida sobre qual atributo fornecer-lhe-á a semântica pretendida, resolve ler as informações de ajuda de alguns dos atributos. Após a seleção, continua a pesquisa. A página seguinte fornece um formulário

com campos para valores a serem dados a estes atributos. José sente dificuldade no preenchimento de alguns dos campos e seleciona, então, a ajuda localizada no título do atributo. Após o preenchimento dos campos e a seleção do botão "Continuar", uma nova página é gerada.

Nesta nova página, José encontra o resultado da pesquisa, isto é, as informações sobre os conteúdos provenientes da busca. Ele seleciona o conteúdo que deseja após ter visto sua descrição. Nesta última página, é informado de que o conteúdo desejado encontra-se a sua disposição no seu ambiente AulaNet. José entende que conseguiu obter a bibliografia que desejava e finaliza a busca.

#### Cenário 4 (usuário conhece e já utilizou a ferramenta)

Ana, sabendo da existência da opção "Pesquisar e adquirir conteúdo", entra no ambiente AulaNet utilizando seu *login* e sua senha. Como ela está montando um curso sobre Mecânica Geral e deseja obter uma bibliografia sobre o tema, seleciona a opção "Pesquisar e adquirir conteúdo".

Na nova página, Ana seleciona o conteúdo "Bibliografia" sem ler as explicações da mensagem localizada antes da lista de opções. Na nova página que é gerada, seleciona os atributos que deseja que façam parte da busca e continua a pesquisa. Na página seguinte, ela preenche os campos com as informações que possui. Após o preenchimento dos campos, uma nova página é gerada e Ana encontra o resultado da pesquisa. Após ver as descrições dos conteúdos, seleciona o conteúdo que deseja e segue para uma nova página. Nesta última página, é informada de que o conteúdo desejado encontra-se a sua disposição no seu ambiente AulaNet. Ana entende que conseguiu obter a bibliografia que desejava e finaliza a busca.

### **6.2.3 Características da ferramenta**

As características desejadas, extraídas dos cenários, para a ferramenta encontram-se listadas a seguir:

- Simplicidade: O processo de busca e aquisição de um conteúdo deve ser simples. O número de tarefas realizadas deve ser pequeno e essas tarefas não devem ser complexas;
- Facilidade: Os textos utilizados na ferramenta devem estar bem escritos e devem fornecer a explicação necessária para a realização da tarefa. Sempre que necessário, devem existir textos de ajuda ao usuário. As tarefas não devem exigir conhecimento em meta-informação ou características do padrão IMS;
- Eficiência: A ferramenta deve garantir que, após o usuário percorrer algumas telas e realizar algumas tarefas, este consiga o desejado. A ferramenta deve fornecer como resposta todos os conteúdos similares ao que o usuário deseja;
- Rapidez: O usuário deve poder selecionar o que for necessário para a pesquisa em curto espaço de tempo, a tarefa realizada pelo usuário em determinada página deve ser feita de forma rápida.

#### **6.2.4 Expectativas do usuário**

O usuário da ferramenta espera que o processo de busca e aquisição de um conteúdo, utilizando o ContentNet, seja análogo a de outras ferramentas de busca disponíveis na *Web*. O usuário acredita que, fornecendo alguma característica do conteúdo, ele consiga adquiri-lo. Com base neste fato, foram observadas as seguintes expectativas que o *designer* espera que o usuário tenha sobre a ferramenta:

- Simplicidade: Com pouca interação entre o usuário e a ferramenta, esta deve ser capaz de fornecer o conteúdo desejado;
- Facilidade: As tarefas realizadas devem ser fáceis ou devem ser facilitadas por algum mecanismo de ajuda;
- Eficiência: A ferramenta deve encontrar o conteúdo desejado e deve cadastrá-lo no ambiente AulaNet;
- Rapidez: O tempo decorrido desde a primeira interação com a ferramenta até a aquisição do conteúdo deve ser curto.



### 6.2.5 Design da Interface

Após a análise do perfil do usuário, dos cenários construídos, das características da ferramenta e das expectativas do usuário, a interface da ferramenta de busca e aquisição foi gerada. Todos os elementos de interface foram projetados com base nestas análises. Neste trabalho, encontra-se a última versão da interface. As versões intermediárias, assim como todos os comentários sobre sua evolução, podem ser encontradas em [Silveira, 2000b] .

A seguir, encontram-se as telas utilizadas na ferramenta e a justificativa do uso dos elementos de interface.

A tela representada na Figura 14 é a primeira tela apresentada para o usuário após o fornecimento do *login* e da senha. Nesta tela (menu principal), estão todas as ações que o usuário pode realizar. Assim, neste menu, foi incluída a opção "Pesquisar e adquirir conteúdo".

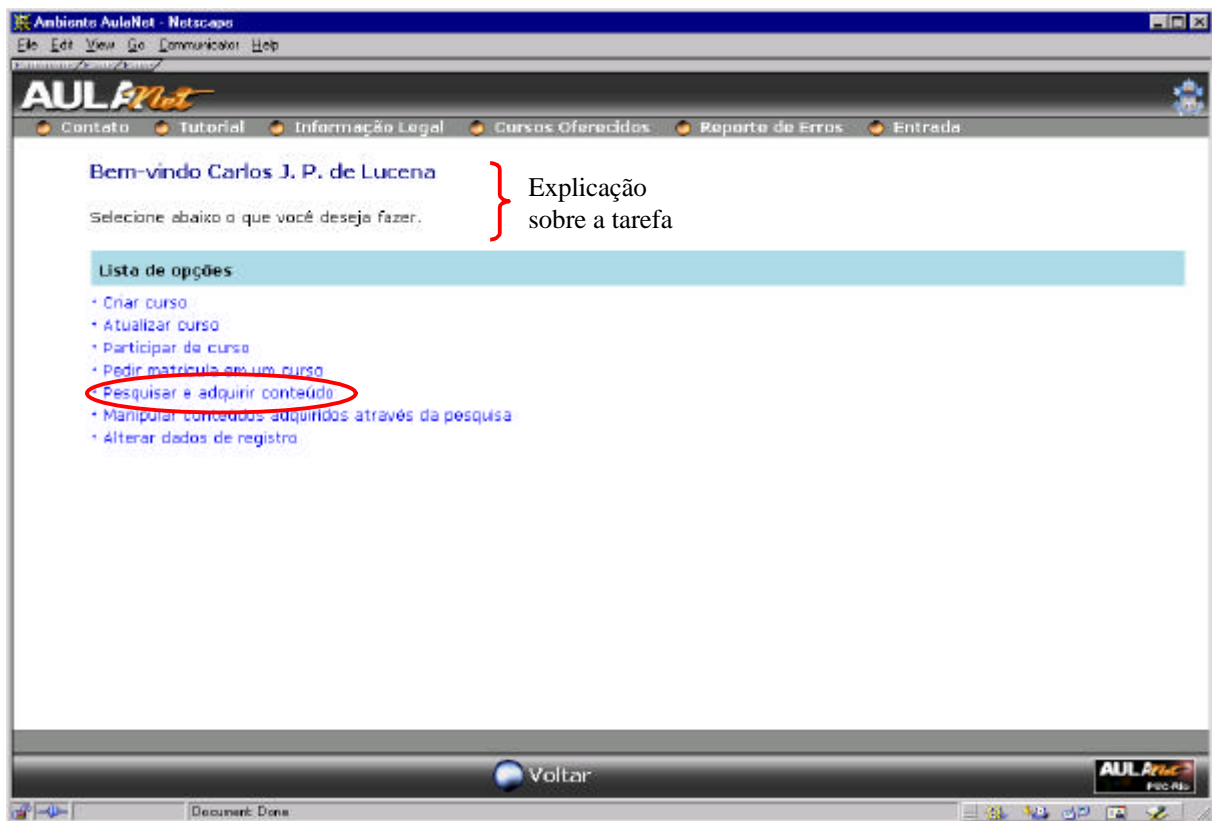


Figura 14 - Tela 1 - Menu de opções

Após a seleção da opção "Pesquisar e adquirir conteúdo", ao invés de um formulário no qual o usuário forneceria os dados para a pesquisa, foi incluída uma nova etapa. Esta etapa deve-se à complexidade da busca de um conteúdo genérico, tendo como informação apenas os dados do formulário, e à complexidade do padrão IMS, o que possibilita a busca de vários tipos de conteúdos. Nesta etapa, o usuário seleciona o tipo de conteúdo que deseja procurar. Na análise dos cenários construídos, verificou-se que esta etapa incluída não prejudicaria o usuário em sua tarefa, mas sim ajudá-lo-ia a concentrar-se no conteúdo que procurava.

Representada na Figura 15, encontra-se a lista com os tipos de conteúdo. Junto com as opções de tipo de conteúdo, encontra-se uma breve explicação sobre alguns destes tipos. Esta explicação só faz-se necessária em casos de nomes ambíguos, como documentação e material, e no caso de questão, pois é um tipo proveniente de outro tipo, auto-avaliação.

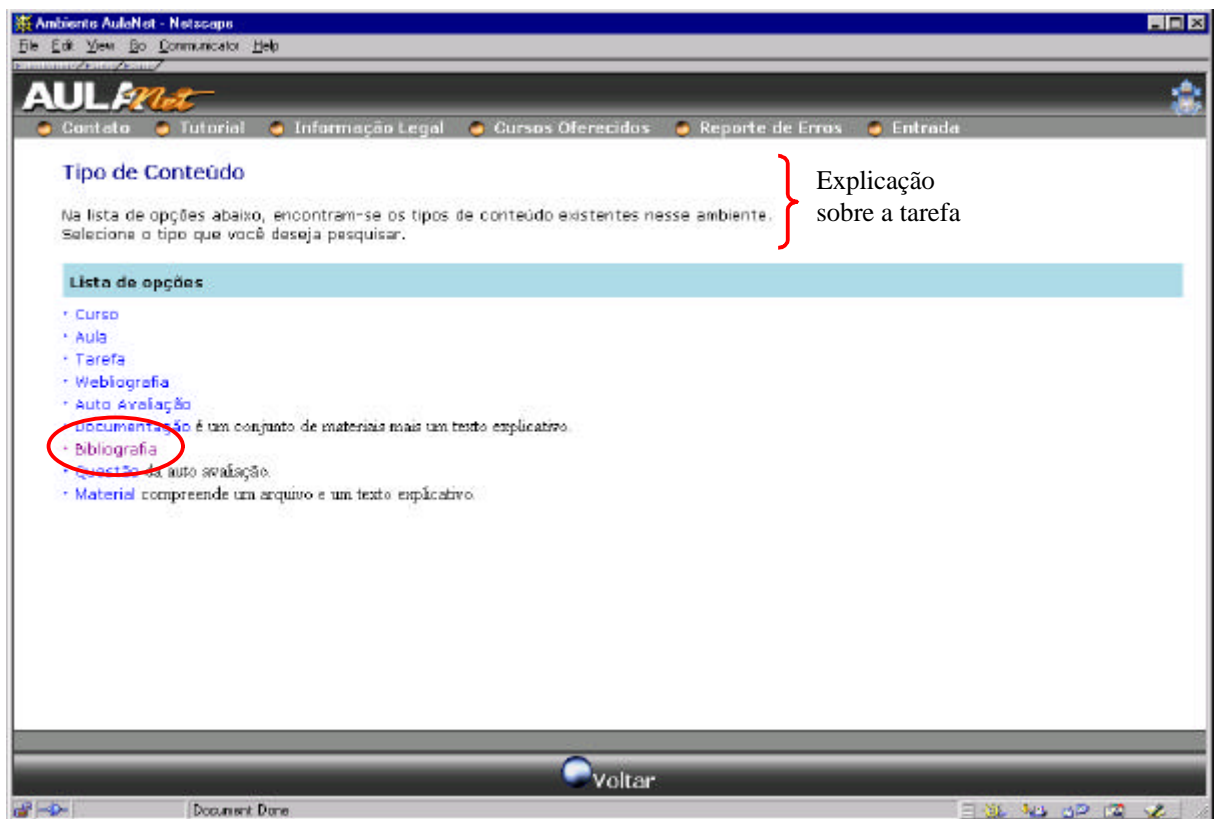


Figura 15 - Tela 2 - Seleção do tipo de conteúdo

Selecionado o tipo do conteúdo, uma nova etapa intermediária foi incluída. Antes do preenchimento dos campos, será necessária a seleção dos campos a serem preenchidos. Esta etapa foi incluída, novamente, devido à complexidade do padrão IMS. Este padrão caracteriza um conteúdo por inúmeros atributos, que, além de serem muitos, são muito complexos. Caso

esta etapa fosse suprimida, um formulário muito grande e complexo seria gerado para o usuário, o que dificultaria, conseqüentemente, seu preenchimento. Outro fator importante seria o tempo para a construção da página.

A tela 3 (seleção dos grupos de atributos representada na Figura 16) apresenta uma lista com os grupos de atributos definidos no padrão. Uma pesquisa pode envolver atributos de diferentes grupos, por isso é permitida a seleção de mais de um grupo.

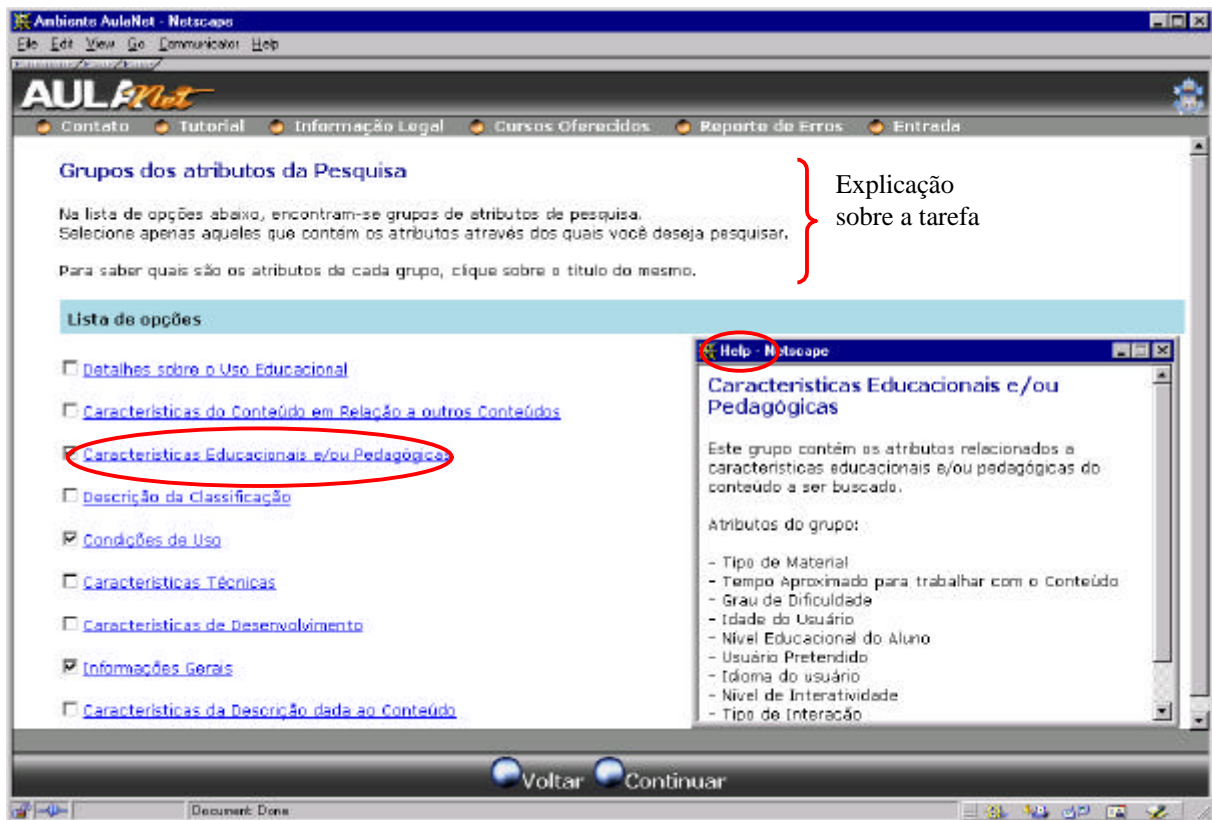


Figura 16 - Tela 3 - Seleção dos atributos

Para cada grupo de atributo, encontram-se uma ajuda com explicação sobre o grupo e uma informação sobre os atributos que o compreendem. Esta ajuda é disponibilizada em um nova janela do *browser* para que o usuário possa visualizar a lista dos grupos e a explicação sem se afastar de sua tarefa.

Selecionados os grupos de atributos, poderia existir uma tela para a seleção dos atributos destes grupos, o que acarretaria na diminuição dos campos do formulário. Apesar do preenchimento do formulário tornar-se mais simples, esta etapa foi suprimida, uma vez que entendeu-se que o benefício trazido pela inclusão desta etapa não compensava aumentar o

número de telas, o que provocaria o desgaste do usuário com o aumento da navegação (o que em termos de *Web* pode tornar-se crítico). Por outro lado, optou-se por permitir que os atributos do formulário que o usuário não quisesse pesquisar pudessem ser deixados em branco. A ferramenta de busca ignora estes atributos cujos valores não foram preenchidos.

Logo após à tela de seleção do grupo, encontra-se a tela contendo o formulário com os atributos da pesquisa. Esta tela, ilustrada na Figura 17, contém todos os atributos dos grupos selecionados. A fim de ajudar o usuário durante o preenchimento deste formulário, uma página com uma explicação sobre o preenchimento de cada atributo é gerada ao selecionar-se o título do mesmo.

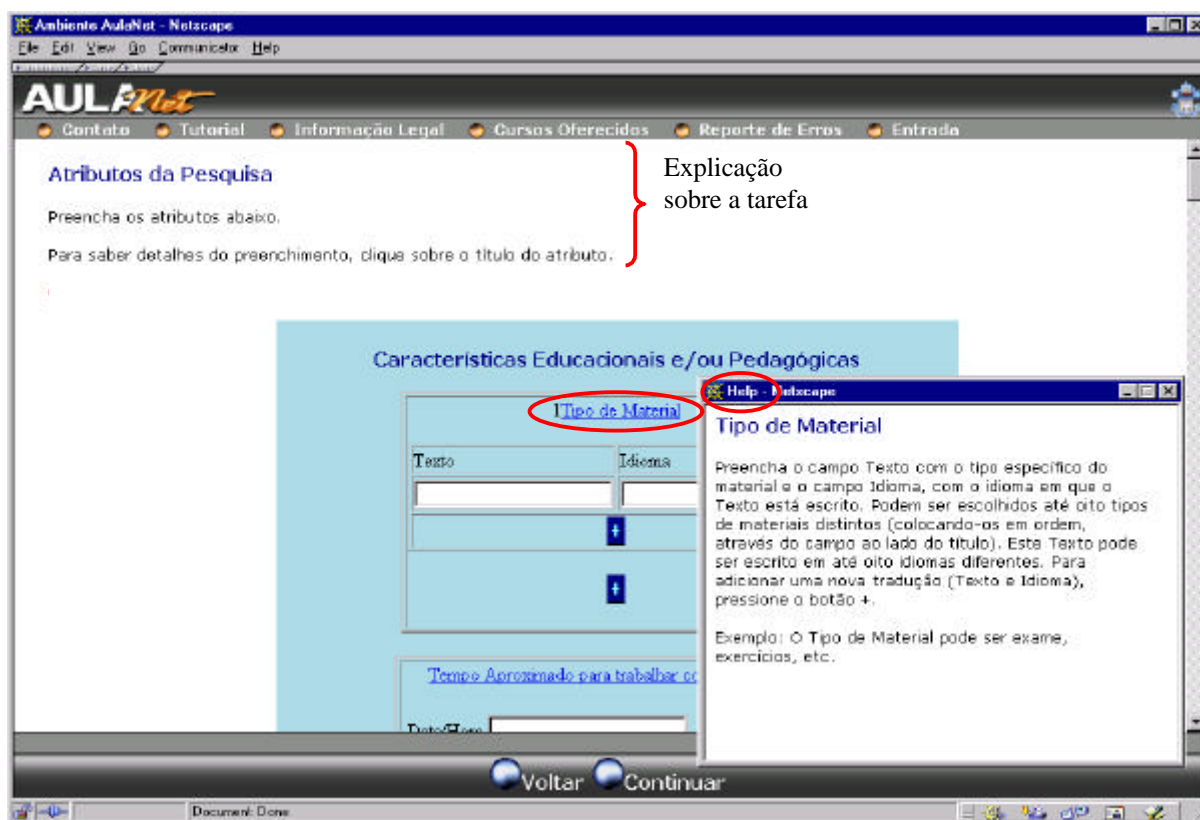


Figura 17 - Tela 4 - Formulário

Os elementos da tela da Figura 17 que não estão sendo mencionados foram projetados após o estudo do padrão IMS. Detalhes sobre estes elementos podem ser encontrados em [Silveira, 2000b].

De posse dos valores dos atributos, a pesquisa já pode ser realizada. Na tela seguinte, Figura 18, encontra-se a lista de opções com o resultado da busca. O resultado apresentado é o título

e a descrição do conteúdo colhidos da meta-informação. Selecionando-se o título do conteúdo, uma janela é aberta contendo toda a descrição do mesmo (meta-informação). Como apenas um conteúdo pode ser buscado de cada vez, só é permitida a seleção de um dos itens da lista.

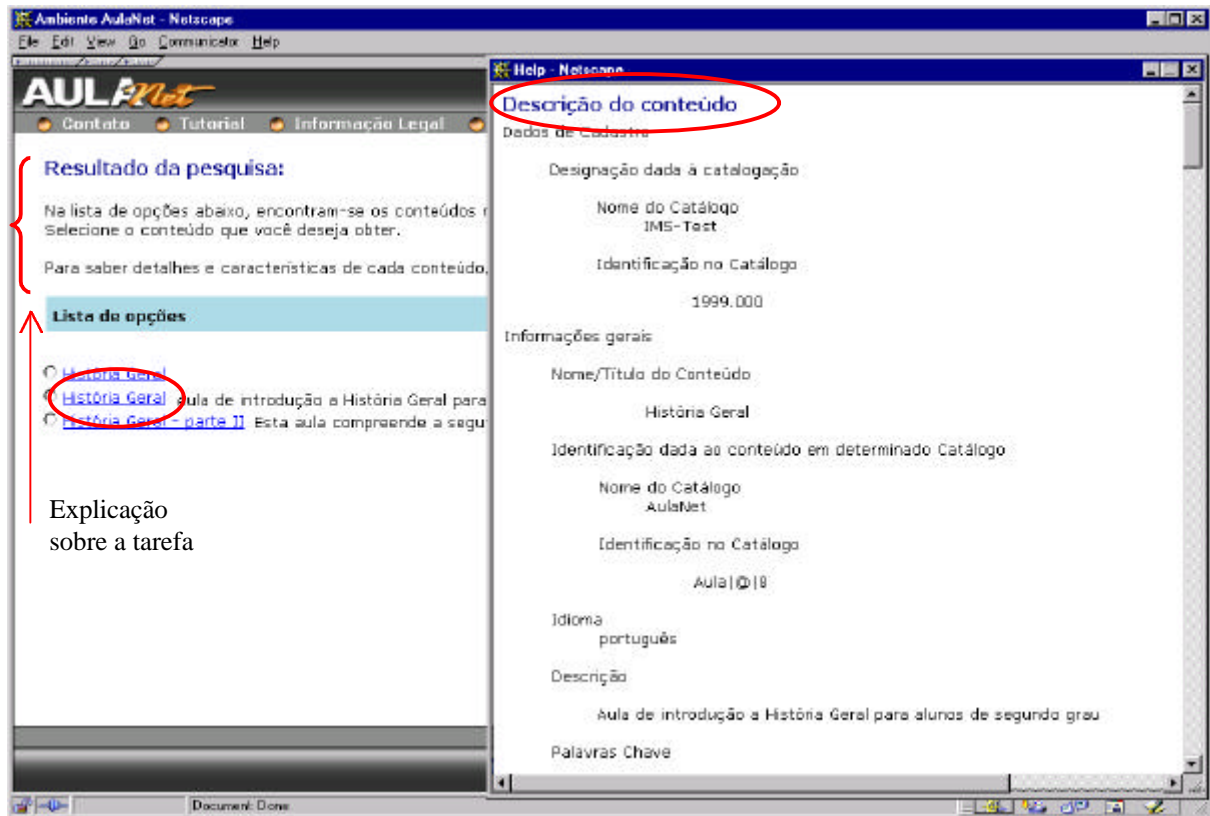


Figura 18 - Tela 5 - Resultado da pesquisa

A última tela (Figura 19) informa ao usuário de que o conteúdo foi adquirido e informa-o, também, como proceder para utilizá-lo.

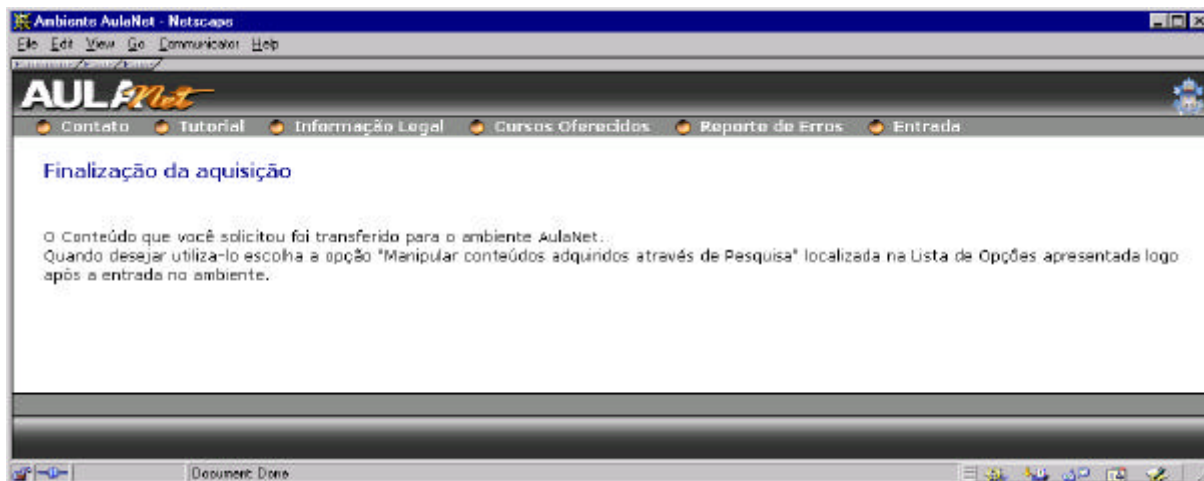


Figura 19 - Tela 6 - Finalização

É importante notar que todas as telas seguiram o *layout* do AulaNet. Além disso, em todas as telas apresentadas, encontra-se uma explicação sobre a tarefa a ser executada em cada tela. Encontra-se também o botão "Voltar", que foi incluído para possibilitar, ao usuário, voltar a página anterior e refazer suas ações. Este botão não foi incluído na última página, pois o usuário já adquiriu o conteúdo que selecionou e, para adquirir outro conteúdo, deve voltar a tela principal.

### 6.3 Interface do Catalogador

A interface do catalogador foi projetada utilizando-se o mesmo processo usado no projeto da interface do professor. O primeiro ponto estudado foi o perfil do usuário, a partir do qual foram construídos os cenários. Logo após, foram estudadas as características do *software* e as expectativas que o *designer* supõe que o usuário possua. Com base nestes dados, projetou-se a interface.

A ferramenta de catalogação para a qual a interface foi projetada não faz parte do *framework* ContentNet. Sabendo-se disto, os dados colhidos no processo de engenharia da interface não sofreram influência do *framework*. Como esta interface foi projetada para catalogar os conteúdos AulaNet, todas as análises realizadas foram feitas tendo em vista o novo ator catalogador do ambiente AulaNet.

### 6.3.1 Perfil dos usuários

Sabendo-se que os usuários do AulaNet nem sempre são informados de todas as funcionalidades e ferramentas oferecidas pelo ambiente, é necessário enfatizar que o catalogador deve ter um bom conhecimento sobre o AulaNet. Ele necessita saber de todas as funcionalidades e ferramentas existentes no ambiente, uma vez que podem influenciar a catalogação dos conteúdos. O catalogador deve conhecer, principalmente, todos os tipos de conteúdos AulaNet e todas as características dos mesmos.

O catalogador deve ter experiência na catalogação de conteúdos, principalmente na catalogação de conteúdos digitais. Ele deve ter, ainda, um bom conhecimento sobre o padrão IMS. Deve saber o que é meta-informação; deve conhecer os elementos nela utilizados, sua descrição, os valores que podem receber e quais destes elementos têm preenchimento obrigatório para validá-la.

O catalogador deve ser capaz de cadastrar as características do conteúdo após tê-lo assistido. Contudo, se ainda assim não for possível a catalogação de todos os elementos, ele deve ser capaz de contatar o professor que criou o conteúdo.

### 6.3.2 Cenários

Dado o perfil dos usuários descritos anteriormente, foram construídos os cenários. Os cenários construídos pelo *design* levam em consideração dois perfis bastante diferentes. No cenário 1, o usuário embora saiba catalogar um conteúdo, pouco conhece sobre o padrão IMS e sobre o AulaNet, o que não é recomendado para usuários da ferramenta. Já no cenário 2, o usuário conhece bem o padrão IMS e o AulaNet. É possível notar que estes cenários baseiam-se em perfis opostos. Por isso, acredita-se que, construindo estes cenários, é possível abranger a maioria dos problemas que possam ocorrer no processo de catalogação.

A tarefa descrita, a seguir, foi usada na construção dos dois cenários.

## Tarefa

Catalogar a aula "História Geral - parte II" após ter assistido a aula pelo AulaNet. Esta aula pertence ao curso "História" do professor José Silva.

### Cenário 1 (usuário pouco sabe sobre o padrão IMS e sobre o AulaNet)

João entra no AulaNet com seu *login* e sua senha. No menu de opções, que aparece a seguir, João encontra a opção "Participar de curso" e "Catalogar Conteúdo". João seleciona a opção "Catalogar Conteúdo" e uma nova janela é fornecida contendo uma explicação e uma lista de opções. Ele lê a explicação e seleciona "Aula" na lista de opções que contém os tipos de conteúdo.

Em seguida, em outra página, uma tabela é montada contendo todas as aulas cadastradas no ambiente AulaNet. João seleciona a aula desejada para a catalogação. Outra página é fornecida contendo uma explicação e um formulário. O formulário contém os atributos da catalogação. Ele verifica que não conhece a aula ainda, e não pode, então, catalogá-la. Seleciona, assim, a opção "Participar de curso" contida na página do formulário<sup>6</sup>. Uma nova janela é aberta e, após assistir o conteúdo, João volta à página do formulário.

João lê a explicação contida no formulário e começa a preencher os campos. Ele sente dificuldade no preenchimento e seleciona a ajuda. A ajuda explica como deve ser feito o preenchimento, porém, devido à complexidade do padrão IMS, ele não consegue preencher muitos campos. Aciona, então, a opção de envio de *e-mail* para o professor da aula e envia suas dúvidas. João passa à página seguinte na qual é informado de que terá de voltar ao ambiente para terminar a catalogação, pois não preencheu os campos necessários para a geração de uma meta-informação válida.

João não recebe todas as respostas esperadas do professor da aula, uma vez que as perguntas são muito específicas sobre catalogação, e o professor não conhece o padrão IMS. Então, João sente bastante dificuldade para cadastrar a meta-informação corretamente. Como caso

---

<sup>6</sup> Não será descrito o cenário do usuário assistindo o curso, pois este não faz parte da ferramenta de catalogação.



alternativo, João não consegue cadastrar a meta-informação corretamente devido à sua falta de conhecimento do padrão.

### Cenário 2 (usuário conhece bem o padrão IMS e o AulaNet)

Maria entra no AulaNet com seu *login* e sua senha. No menu de opções, Maria seleciona "Participar de curso" e assiste a aula "História Geral - parte II". Em seguida, volta ao menu principal e seleciona a opção "Catalogar conteúdo". Na próxima página, ela seleciona a opção "Aula" da lista de opções contendo os tipos de conteúdo.

Em seguida, uma tabela é montada contendo todas as aulas cadastradas no ambiente AulaNet. Maria seleciona a aula desejada para a catalogação. Outra página é fornecida contendo uma explicação e um formulário. O formulário contém os atributos da catalogação. Ela preenche alguns campos e entra em contato com o professor do conteúdo a fim de saber mais informações sobre a aula. Passa, então, à página seguinte na qual é informada de que terá de voltar ao ambiente para terminar a catalogação, pois não preencheu os campos necessários para a geração de uma meta-informação válida. Maria sai do AulaNet.

Mais tarde, após receber as informações necessárias do professor via *e-mail*, Maria volta ao AulaNet para terminar a catalogação da aula. Maria entra no AulaNet com seu *login* e sua senha. No menu de opções, seleciona "Participar de curso" e assiste a aula "História Geral - parte II". Em seguida, Maria volta ao menu principal e seleciona a opção "Catalogar conteúdo". Na próxima página, seleciona a opção "Aula", da lista de opções que contém os tipos de conteúdo. Em seguida, seleciona a aula "História Geral - parte II" que possui uma observação informando que está aula não possui meta-informação válida.

Na outra página, o formulário com os atributos aparece. Nele, alguns atributos já vêm preenchidos. Maria termina de preencher o formulário e passa para outra página, na qual é informada de que a meta-informação é válida.

### 6.3.3 Características da ferramenta

As características listadas abaixo são as desejadas para a ferramenta de catalogação do conteúdo:

- Simplicidade: A ferramenta deve fornecer tarefas simples a fim de que usuário atenda-se apenas às complexidades do padrão;
- Facilidade: A ferramenta deve facilitar a catalogação fornecendo, ao usuário, textos explicativos sobre o uso da ferramenta, sobre o padrão IMS e sobre características do ambiente AulaNet sempre que necessário;
- Eficiência: A ferramenta deve permitir catalogar o conteúdo de forma eficiente, isto é, dentro do padrão IMS. O catalogador também deve ser informado sobre quais conteúdos não possuem ainda meta-informação, quais já a possuem e, destas, quais estão completas;
- Rapidez: Deve ser possível executar as tarefas necessárias à catalogação de forma rápida. O usuário deve catalogar um conteúdo, executando poucas tarefas.

#### **6.3.4 Expectativas do usuário**

Catalogar um conteúdo não é uma tarefa simples. O usuário da ferramenta, por mais informação que tenha sobre o padrão IMS e por mais experiência que possua em catalogação de conteúdos digitais, pode, por ventura, ter dificuldade na realização de sua tarefa. Esta dificuldade pode estar na interpretação das características do conteúdo que está catalogando ou na própria dificuldade trazida pelo padrão. Tomando como base estas observações, a seguir, encontram-se as expectativas que o *designer* supõe que o usuário possua sobre a ferramenta:

- Com a ferramenta, deve ser possível visualizar todos os conteúdos a serem cadastrados;
- A própria ferramenta deve viabilizar o contato entre o catalogador e o professor do ambiente quando existirem dúvidas no preenchimento de algum campo da meta-informação;
- Uma página de ajuda sobre o padrão IMS deve poder ser consultada sempre que for necessário;

- Assim como o professor, este usuário espera que a ferramenta seja simples, fácil de ser usada, eficiente e que o processo de catalogação seja rápido devido à quantidade de conteúdos produzidos.

### **6.3.5 *Design* da Interface**

Analisados os itens mencionados anteriormente, foi construída a interface do catalogador na qual todos os elementos que compõem a interface foram projetados com base nesta análise.

As telas, encontradas a seguir, foram projetadas conforme o *layout* do ambiente AulaNet.

A tela representada na Figura 20 é a primeira tela apresentada para o usuário após o fornecimento do *login* e da senha. Como visto anteriormente, nesta tela, estão todas as ações que o usuário pode realizar. Por este motivo, a opção "Catalogar conteúdo" foi posicionada neste menu.

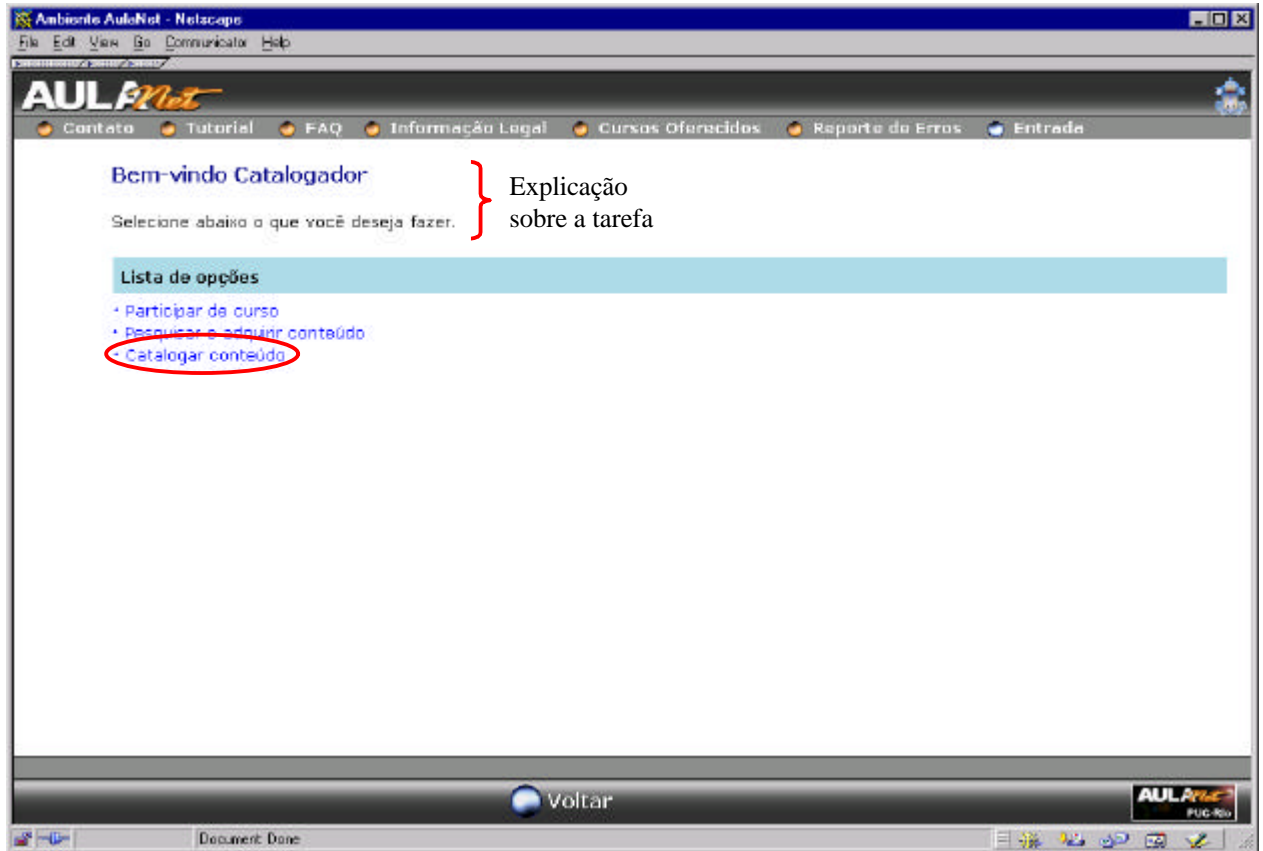


Figura 20 - Tela 1 - Menu de opções

Após o menu inicial, o catalogador poderia esperar uma lista dos conteúdos ainda não catalogados. Contudo, optou-se por incluir uma tela intermediária. Nesta tela, representada na Figura 21, o catalogador escolhe o tipo do conteúdo que deseja catalogar. Esta tela intermediária foi incluída a fim de facilitar a seleção do conteúdo desejado. O catalogador seleciona primeiro o tipo do conteúdo e, depois, o conteúdo, diminuindo o número de opções e facilitando, assim, a seleção.

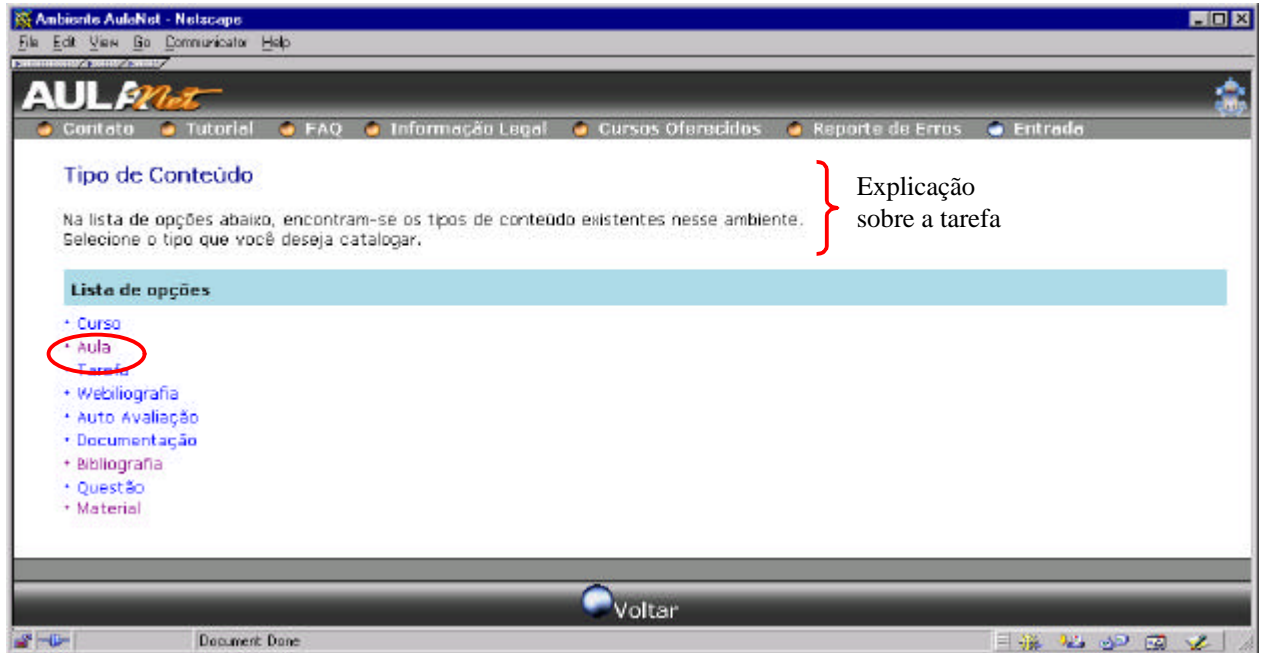


Figura 21 - Tela 2 - Seleção do tipo de Conteúdo

A Figura 22 ilustra a seleção do conteúdo a ser catalogado. Nesta tela, juntamente com o nome do conteúdo e sua descrição, fornecidos pelo professor que o criou, informa-se a etapa da catalogação do mesmo. A catalogação do conteúdo pode ainda não ter sido feita (primeira etapa); pode ainda não estar completa (segunda etapa); e pode ter sido finalizada, concluindo três etapas.

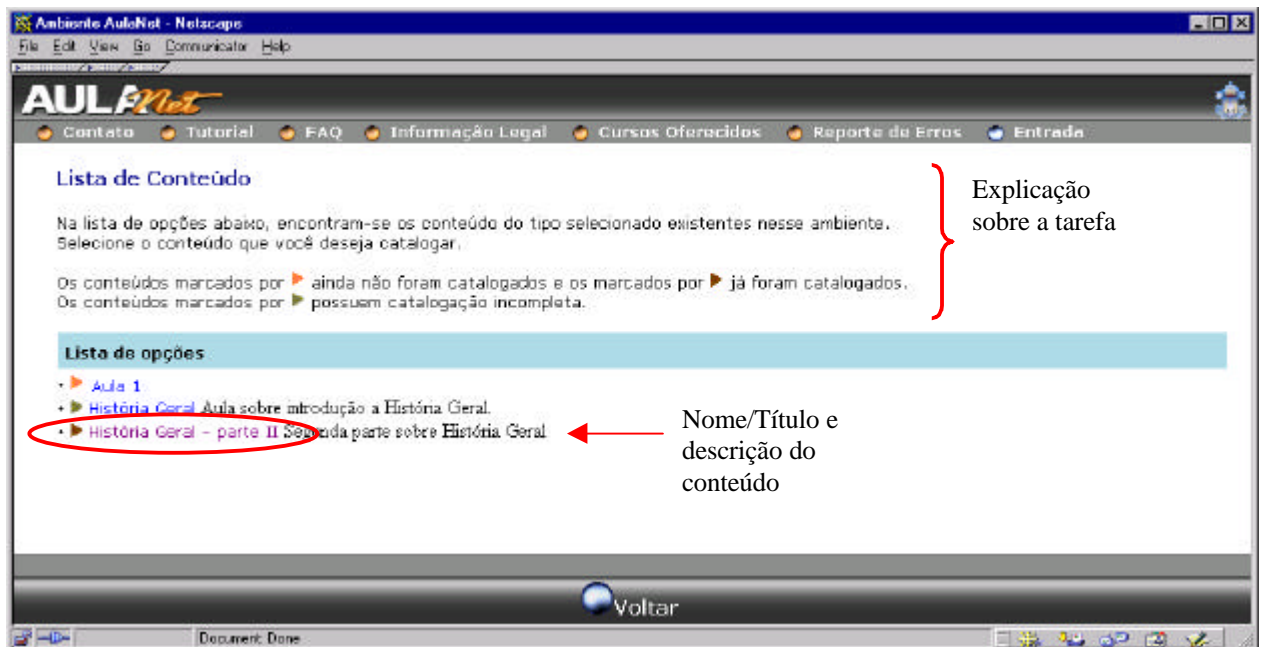


Figura 22 - Tela 3 - Lista de Conteúdo

Uma vez selecionado o conteúdo, o catalogador deve preencher o formulário com o valor dos atributos do conteúdo. A tela representada na Figura 23 é semelhante à tela utilizada pelo professor na Figura 17.

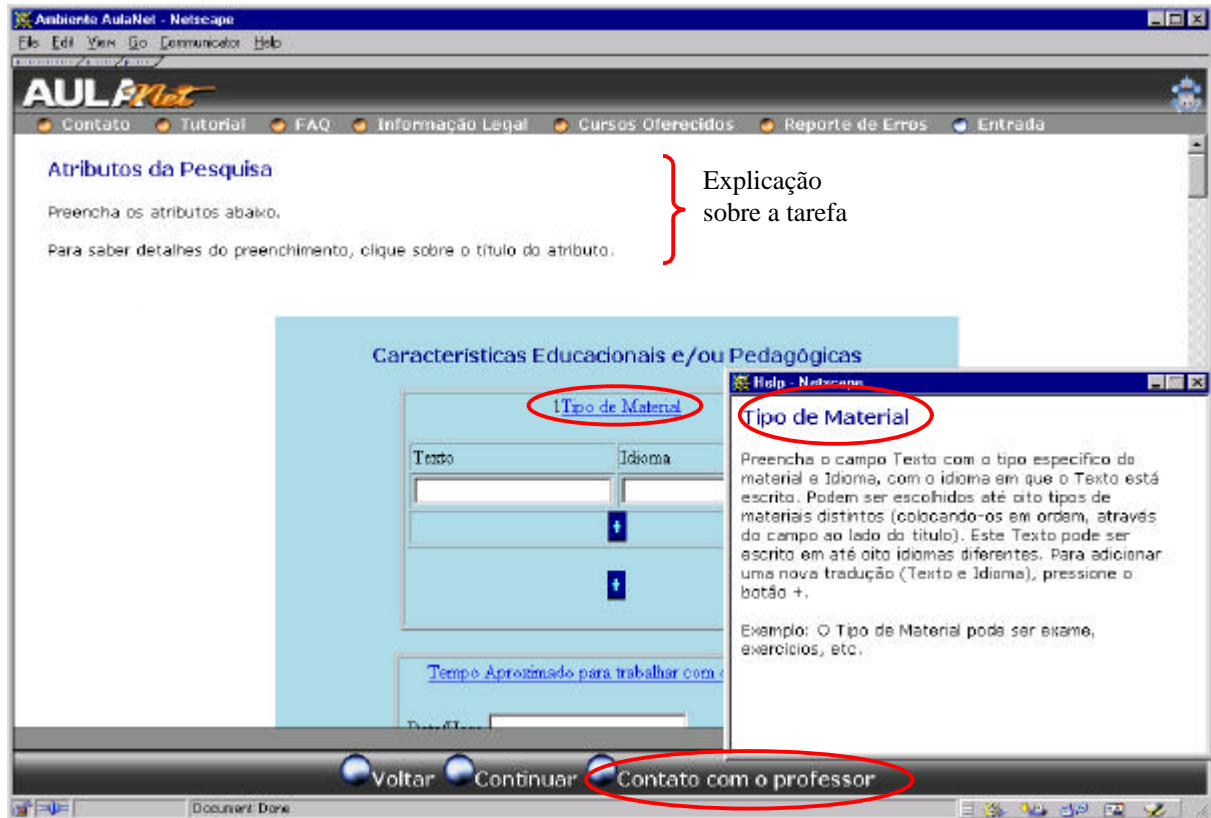


Figura 23 - Tela 4 - Atributos da Pesquisa

O que difere estas duas telas é a opção de envio de *e-mail* do catalogador para o professor que criou o conteúdo. Esta opção foi acrescentada para possibilitar, ao catalogador, esclarecer as dúvidas que possam surgir ao preencher os atributos do conteúdo. A Figura 24 ilustra a tela de envio de *e-mail*. Nela, o catalogador fornece o *e-mail* do professor, o assunto e o texto da mensagem.



Figura 24 - Contato

Após o preenchimento dos atributos da catalogação, a tela da Figura 25 é apresentada informando de que o processo de catalogação chegou ao fim.

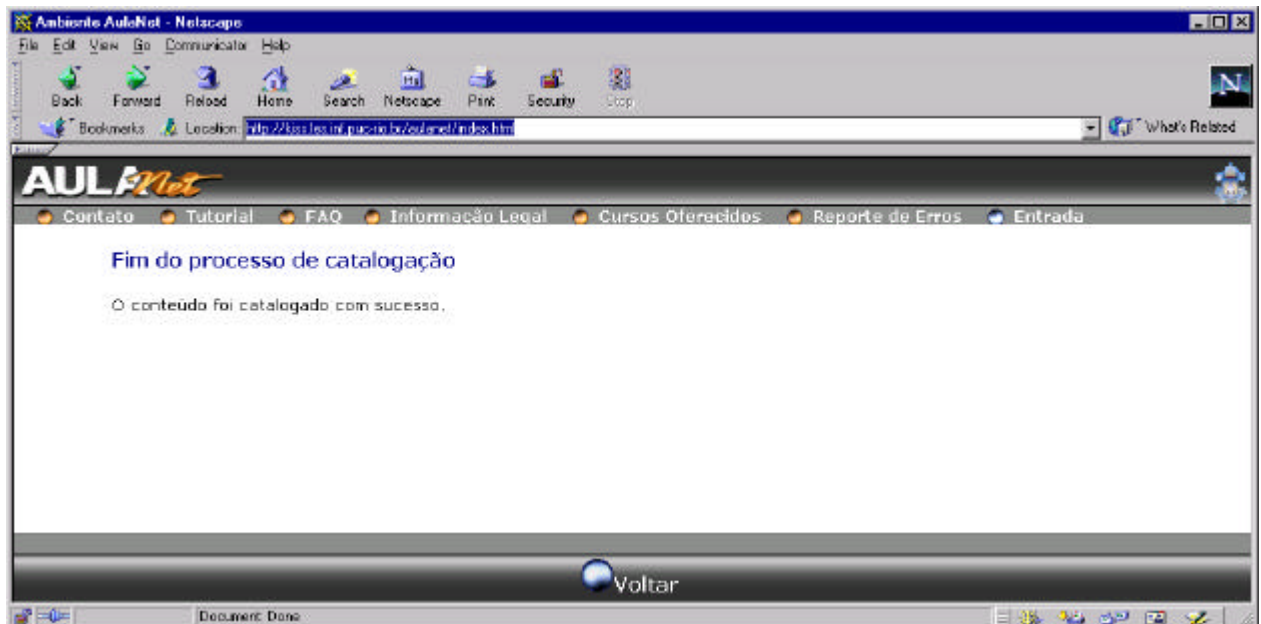


Figura 25 - Tela 5 - Fim do processo de catalogação

Em todas as telas apresentadas, o *layout* do AulaNet foi seguido e um texto foi incluído com o objetivo de explicar a tarefa a ser realizada.

Uma vez projetadas as interfaces do professor e do catalogador, testes foram realizados com o objetivo de verificar a usabilidade das mesmas. O método utilizado na aplicação e na análise deste testes foi o método de avaliação da comunicabilidade.

## 6.4 Avaliação da Comunicabilidade

Comunicabilidade é a propriedade de um *software* de transferir, eficientemente e efetivamente, para o usuário, os princípios interativos e intencionais do *design*. [Prates, 2000]

O método baseia-se na Engenharia Semiótica, pela qual o sistema é visto como uma mensagem do *designer* para os usuários [de Souza, 1999]. O objetivo deste método é prover uma maneira dos avaliadores identificarem pontos nos quais o *designer* possa ter falhado em transferir, para os usuários, suas mensagens, ou seja, pontos de conflito de *design*. O método provê, também, uma maneira pela qual os usuários possam comunicar-se com o *designer*, ainda que indiretamente.

Os usuários que fazem, espontaneamente, o teste, expressam suas expectativas, atitudes, interpretações, rejeições ou aprovações sobre a interface testada [Prates, 2000]. Para capturar estes sentimentos e identificar quando os conflitos ocorrem, o avaliador deve observar expressões que indiquem dúvidas, frustrações, confusão, etc..

O método descrito aplica-se a interfaces mono-usuários e consiste, basicamente, em três etapas: aplicação de *tags*, interpretação do resultado e modelagem semiótica. A terceira etapa não foi realizada, pois é necessário um especialista em Engenharia Semiótica para efetuar-la.

### 6.4.1 Aplicação de *tags*

Nesta primeira etapa, é necessário pedir que usuários realizem uma ação pré-definida usando a aplicação a ser avaliada. As ações do usuário precisam ser armazenadas usando algum *software* que seja capaz de gravar os movimentos executados pelo *mouse*, os eventos realizados pela interface, etc.. Ainda nesta etapa, os avaliadores devem associar expressões (*tags*) identificando "conversas" em potencial do usuário com a ferramenta no momento do teste, enquanto interagem com as gravações destes testes.

As *tags* são relacionadas a um conjunto de expressões que indicam uma reação do usuário a algo que aconteceu durante a execução da tarefa e a interação com o ambiente. Estas *tags* [de



Souza, 1999], listadas a seguir, foram selecionadas para expressar diferentes tipos de situações de ruptura na interação, ou seja, ruptura na comunicação *design*-usuário durante a interação, e atitudes de usuários que ocorrem durante a interação deste com a máquina[Prates, 2000]:

- *Where is? (What now?)* → O usuário está, aparentemente, buscando uma função específica. Como tem dificuldade em encontrá-la, executa uma busca temática ou sequencial em menus e barras de tarefas. Uma *tag* especial é incluída nesta categoria: *What now?* Esta *tag* aplica-se quando o usuário não sabe o próximo passo a ser dado;
- *What's this?* → O usuário encontra-se, aparentemente, explorando as possibilidades de interação para obter mais entendimento sobre uma determinada função;
- *Oops! (I can't do it this way. / Where am I?)* → O usuário verifica que executou uma ação incorreta e imediatamente a corrige. *I can't do it this way* indica que o usuário realiza uma seqüência de ações e volta ao passo original, pois acredita que tal direção não lhe levará à execução da tarefa. Esta categoria inclui também *Where am I?* na qual o usuário verifica que uma ação que era apropriada em um contexto não é mais;
- *Why doesn't it? (What happened?)* → O usuário recebe uma mensagem da interface que não era a esperada. Normalmente, ele insiste na ação. Já a *tag What happened?* é usada quando o usuário não recebe mensagem da interface e não consegue, então, compreender o que aconteceu;
- *Looks fine to me...* → O usuário encontra um resultado que acredita ser o certo, interpreta, entretanto, mal a mensagem fornecida. O usuário não verifica que o resultado não é o correto;
- *I can't do it.* → O usuário está impossibilitado de alcançar o objetivo, porque não sabe como ou não possui recursos para fazê-lo(tempo, paciência, etc.);
- *Thanks, but no, thanks. (I can do otherwise.)* → O usuário não utiliza o caminho mais saliente proposto pelo *design* para executar a ação corretamente. A diferença entre *Thanks, but no, thanks* e *I can do otherwise* é que, enquanto no primeiro caso existe uma explícita declinação do caminho saliente, no segundo, o caminho saliente não é percebido.

### 6.4.2 Interpretação dos resultados

A segunda etapa do método baseia-se na Tabela 3 e na observação do número de aparições de cada *tag* nos testes realizados. A Tabela 3 relaciona as *tags* utilizadas no método com os problemas de interatividade e usabilidade (navegação, atribuição de significado, cumprimento da tarefa, incompreensão/recusa no uso de *affordance*). Observando o número de aparições de cada *tag* e verificando a quais problemas relacionam-se, é possível construir um gráfico no qual estarão identificados os pontos críticos do *design* no contexto da tarefa.

	Navegação	Atribuição de significado	Cumprimento da tarefa	Incompreensão/recusa no uso de <i>affordance</i> <sup>7</sup>
<b>I can't do it.</b>			√	
<b>Looks fine to me...</b>			√	
<b>Where is? What now?</b>	√			
<b>What's this? Object or action?</b>		√		
<b>Why doesn't it? What happened?</b>		√		
<b>Oops! I can't do it this way. Where am I?</b>	√	√		
<b>Thanks, but no, thanks I can do otherwise.</b>				√

Tabela 3 - Relacionamento entre as *tags* e os problemas de interatividade e usabilidade.

### 6.5 Testes realizados

Os testes realizados aplicam-se às interfaces projetadas para o professor e para o catalogador. Os testes, aplicados a cada uma destas interfaces, consistem no preenchimento de

---

<sup>7</sup> *Affordance* é um termo que se refere às propriedades percebidas e reais de um artefato, em particular as propriedades fundamentais que determinam como este artefato pode ser utilizado [Norman, 1988].

um formulário e na utilização da ferramenta na realização de uma ou mais tarefas. Nos testes realizados com a ferramenta, utilizou-se o *software* Lotus® ScreenCam™ com o objetivo de gravar os movimentos do *mouse*, eventos da interface, etc..

Os formulários e as descrições das tarefas realizadas pelos usuário estão expostos no Apêndice A.

### 6.5.1 Interface do Professor

À interface do professor, aplicou-se um teste utilizando-se três usuário. Na Tabela 4, encontra-se o perfil destes usuários.

	<b>Conhecimento sobre o AulaNet</b>	<b>Papel no ambiente AulaNet</b>	<b>Conhecimento sobre o Padrão</b>	<b>Conhecimento sobre o ContentNet</b>	<b>Uso do ContentNet</b>
<b>Usuário 1</b>	Regular	Usuário	Baixo	Regular	Nunca utilizou
<b>Usuário 2</b>	Bom	Usuário Desenvolvedor	Nenhum	Regular	Nunca utilizou
<b>Usuário 3</b>	Excelente	Suporte Desenvolvedor	Regular	Bom	Nunca utilizou

Tabela 4 - Perfil dos usuários da interface do professor

O teste aplicado consiste na descrição de um cenário realizado pelo usuário e na execução de duas tarefas propostas pelo *design* utilizando a ferramenta. O cenário, assim como as tarefas, trata da busca de um conteúdo pelo fornecimento de valores para alguns atributos.

## Cenário

Nos cenários desenvolvidos pelos usuários, estes não mencionam nenhuma prévia seleção dos atributos que devem fazer parte da busca. Os usuários descrevem a necessidade de um único formulário contendo todos os atributos necessários para a pesquisa. Neste formulário, os valores dos atributos deveriam ser selecionados a partir de uma lista pré-definida ou deveriam ser fornecidos pelo próprio usuário. Os usuários mencionam que os atributos existentes no formulário que não foram preenchidos não devem ser considerados pela ferramenta.

A seleção prévia dos grupos de atributos utilizados na interface do professor foi projetada pelo *designer* para facilitar o preenchimento do formulário. Se não houvesse esta prévia seleção, o formulário contendo os atributos seria muito grande devido ao número de atributos existentes descritos pelo padrão IMS. Acredita-se, assim, que os usuários não mencionaram a seleção dos atributos para a busca por causa da falta de conhecimento do padrão.

Os usuários também não descrevem a seleção do tipo de conteúdo a ser buscado, uma opção utilizada pelo *designer* para facilitar a seleção dos atributos. Eles ainda não mencionam a possibilidade de ser encontrado mais de um conteúdo com as mesmas características. Conseqüentemente, não descrevem a seleção de um entre os conteúdos do resultado.

Os usuários também não mencionam a existência de ajudas sobre o preenchimento do formulário, nem sobre qualquer outra etapa do processo de busca. Acredita-se, também, que isto deve-se à falta de conhecimento do padrão IMS, já que a ajuda sobre a descrição dos atributos é essencial na visão do *designer*.

## Tarefas

Durante a execução das duas tarefas, enquanto os usuários eram filmados, dados foram colhidos com o objetivo de ajudar na análise dos filmes. Após a execução das tarefas, foram analisados, ao todo, seis filmes, dois por usuário, sendo um filme para cada tarefa. Como mencionado anteriormente, utilizou-se o método de avaliação da comunicabilidade.

Ainda na primeira etapa do método, as *tags* foram aplicadas a cada um dos filmes. Já na segunda etapa, houve a contagem das *tags* aplicadas para cada uma das tarefas, sendo que o pior caso de ocorrência de cada *tag* foi anotado. Relacionando-se estas ocorrências com a Tabela 3, foi possível produzir o

Gráfico 1.

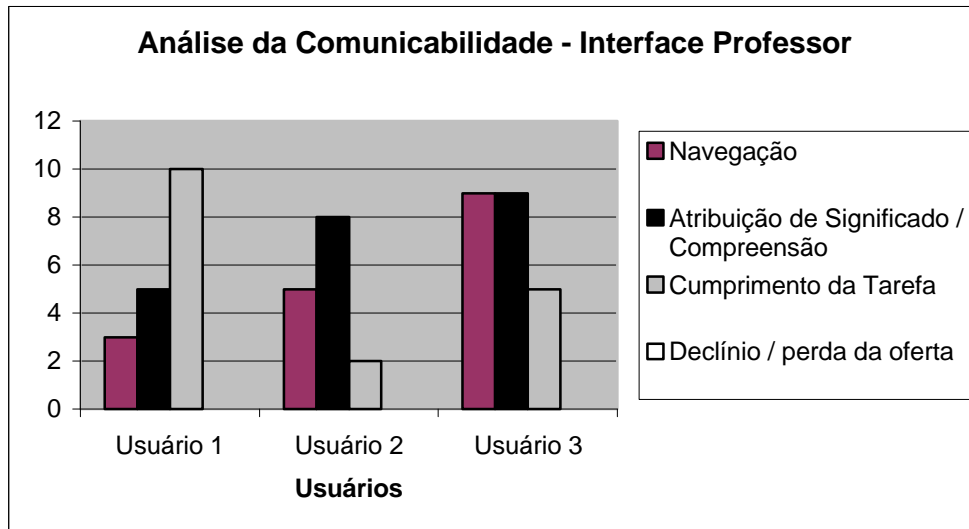


Gráfico 1 - Análise da Comunicabilidade - Interface Professor

Como é possível notar, nenhum usuário deixou de compreender ou recusou a oferta feita pelo *designer* durante a execução das tarefas. A seguir, encontra-se uma análise sobre a ocorrência dos problemas de interatividade e usabilidade.

### Navegação

A ocorrência de problemas com a navegação surgiu na tela de seleção dos grupos de atributos. Os usuários não sabiam qual o grupo que deveriam selecionar, o que os levava a utilizar o *scroll*, passando de um em um procurando o grupo cujo nome trouxesse alguma similaridade com o nome dos atributos desejados. Como solução deste problema, propõe-se o uso de nomes mais claros para os grupos de atributos que indiquem o conjunto de atributos que compreendem o grupo. Outra solução seria informar o usuário de todos os atributos existentes no padrão IMS, ao invés de informar o grupo destes atributos. Esta solução não foi escolhida pelo *designer* devido ao grande número de opções de atributos existentes no padrão. Acredita-se que, selecionando o grupo, torna-se mais fácil encontrar o atributo que se deseja.

Encontraram-se problemas com a navegação também na utilização do formulário dos atributos. O usuário utiliza o *scroll* novamente procurando os atributos que devem ser preenchidos. Apesar dos atributos estarem posicionados em conjunto, informando o nome do grupo ao qual pertencem, isto é, os atributos do mesmo grupo aparecem seqüencialmente, os usuário não percebem este fato. A solução proposta para este caso é o uso de alguma identificação por cores para os grupos de atributos. Outra solução seria construir uma tela para cada grupo selecionado, ou seja, os atributos dos mesmo grupo ficam na mesma tela. Esta opção não foi a escolhida pelo *designer*, pois implicaria um aumento do número de telas em oito para fazer uma busca, caso o usuário selecionasse todos os atributos. O número de telas para uma ferramenta *Web* é crítico devido ao tempo necessário para carregar cada uma delas.

### Atribuição de significado / Compreensão

Problemas com a atribuição de significado e compreensão surgiram em dois momentos. No primeiro momento, os usuário necessitam selecionar a ajuda para verificar quais os atributos pertencentes a cada grupo. No segundo, os usuário necessitam selecionar a ajuda de cada atributo ao preencher o formulário. A seleção da ajuda, nestes dois momentos, deve-se à complexidade do padrão IMS.

O problema da seleção da ajuda para os atributos do grupo poderia ser solucionado listando-se os atributos que compreendem o grupo na interface de seleção do grupo. No entanto, esta solução não foi adotada pelo *designer*, pois a interface ficaria carregada demais. Além disso, foi possível verificar, por meio de teste com a segunda tarefa, que os usuários, após utilizarem a interface pela primeira vez, já não necessitam de ajuda com tanta frequência. O mesmo pode-se concluir para as ajudas selecionadas nos atributos sobre o preenchimento do formulário.

### Cumprimento da tarefa

Os problemas descritos pelo não cumprimento da tarefa ocorreram durante todo o processo de execução desta, agravando-se na execução da primeira tarefa, não devido à sua complexidade, mas devido aos usuários nunca terem usado a interface. Calcula-se que este problema agravou-se por dois motivos principais: as dificuldades do padrão utilizado e o não uso das

explicações existentes na interface (ajudas). Analisando-se os vídeos, é possível notar que o usuário não lê o que está escrito na interface e tenta adivinhar o que deve ser feito. Assim, a tarefa não é cumprida corretamente. Como solução, propõe-se formular explicações melhores e que chamem mais atenção ao usuário.

### 6.5.2 Interface do Catalogador

O teste realizado com a interface do catalogador foi aplicado a dois usuários. O perfil destes usuários está descrito na Tabela 5.

	<b>Conhecimento sobre o AulaNet</b>	<b>Papel no ambiente AulaNet</b>	<b>Conhecimento sobre o Padrão</b>	<b>Conhecimento sobre o ContentNet</b>	<b>Uso do ContentNet</b>
<b>Usuário 1</b>	Bom	Usuário Desenvolvedor	Nenhum	Regular	Já utilizou a pesquisa
<b>Usuário 2</b>	Regular	Usuário	Bom	Bom	Já utilizou a pesquisa

Tabela 5 - Perfil dos usuários da interface do catalogador

O teste aplicado consiste na execução de uma tarefa. No teste da interface do catalogador, não foram utilizados cenários, uma vez que catalogar um conteúdo digital não é uma tarefa da qual os usuários possam conhecimento. Os usuários que participaram do teste não sabiam escrever o processo de catalogação.

#### Tarefa

Enquanto os usuários eram filmados executando a tarefa, dados foram colhidos com o objetivo de ajudar na análise dos filmes. Após a execução das tarefas, foram analisados os dois filmes utilizando o método de avaliação da comunicabilidade.

*Tags* foram aplicadas aos dois filmes, e a soma da ocorrência de cada *tag* foi anotada. Relacionando-se estas ocorrências com a Tabela 3, foi possível produzir o Gráfico 2.

#### Gráfico 2 - Análise da Comunicabilidade - Interface Catalogador

Assim como ocorreu na interface do professor, nenhum usuário deixou de compreender ou recusou a oferta feita pelo *designer* durante a execução das tarefas. A seguir, encontra-se uma análise sobre a ocorrência dos problemas de interatividade e usabilidade.

#### Navegação

Problemas com a navegação foram observados tanto na seleção do conteúdo a ser catalogado quanto no preenchimento do formulário. Na seleção do conteúdo, devido às cores semelhantes utilizadas para elementos da interface, o usuário não sabia o que fazer. O usuário teve dificuldade em identificar as cores na identificação de catalogação incompleta e catalogação completa. O usuário, à princípio, entendeu que o conteúdo a ser catalogado já possuía sua catalogação completa, e assim, não teria mais o que fazer. Após a dúvida, o usuário verificou que a catalogação estava incompleta e o que teria de fazer era continuar a catalogação. Como solução a este problema, propõe-se trocar a cor de uma das identificações.

O problema de navegação que surgiu no formulário utilizado pelo catalogador foi o mesmo que surgiu no formulário utilizado pelo professor, já que estes formulários são semelhantes. Por isso, comentários para este problema já foram discutidos na seção anterior.



### Atribuição de Significado/Compreensão

A ocorrência de problemas com a atribuição de significado/compreensão deu-se no preenchimento do formulário. O usuário sente dificuldade no preenchimento do formulário e aciona a ajuda sempre que há necessidade.

Embora um dos usuários seja conhecedor do padrão IMS, é importante notar que ele também sente necessidade de acionar a ajuda, mesmo em um número menor de vezes.

### Cumprimento da tarefa

O problema com o cumprimento da tarefa deu-se devido ao preenchimento impróprio de campos do formulário. Os usuários, utilizando a ajuda ou não, preencheram campos incorretamente, embora achassem que o faziam corretamente. Estima-se que este problema possa ter ocorrido devido à explicação da ajuda não estar satisfatória e à dificuldade do padrão. Como solução, os textos contidos na ajuda devem ser revistos.

## Capítulo 7

### Conclusões e trabalhos futuros

#### 7.1 Conclusões

Utilizando-se RDF/XML e com base no projeto IMS, é possível construir ambientes que lidem com a meta-informação disponível em servidores de conteúdo. Pela meta-informação padronizada dos conteúdos, é possível encontrar um conteúdo desejado de forma mais eficiente. Com a informação de que existe o conteúdo que se deseja, torna-se viável a transferência deste conteúdo entre dois sistemas que se comunicam.

O *framework* ContentNet dá suporte à geração de uma ferramenta que possibilita o intercâmbio de conteúdos educacionais entre servidores de conteúdos que estejam de acordo com o padrão IMS. O *framework* proposto visa facilitar e viabilizar a pesquisa e a transferência destes conteúdos.

Implementando os *hot-spots* do *framework*, é possível sua adaptação no que diz respeito à forma de armazenamento dos conteúdos trocados, ao método utilizado na comparação das meta-informações e à identificação de um conteúdo dada sua meta-informação.

Devido à compatibilidade dos servidores AulaNet com os demais servidores que utilizam o conceito de meta-informação proposto no padrão IMS, e instanciando-se o módulo do *framework* ContentNet, responsável pelos servidores de conteúdo, é possível a criação de uma comunidade de conhecimento em torno de conteúdos educacionais.

Com o desenvolvimento do protocolo de comunicação entre os servidores de conteúdo e um servidor de pesquisa, contribui-se tanto para o desenvolvimento do ambiente AulaNet quanto para a evolução do projeto IMS, dos quais somos participantes. Ao ambiente AulaNet, acrescentou-se uma ferramenta que possibilita seus professores pesquisarem e buscarem um conteúdo. Ao projeto IMS, acrescentou-se uma ferramenta que estabelece o protocolo de troca de conteúdo.

A versão do protótipo foi desenvolvida de maneira autônoma e encontra-se em fase de integração com o AulaNet em sua versão 2.0.

## 7.2 Trabalhos futuros

A seguir, estão listados alguns trabalhos que podem ser realizados como continuidade do tema abordado nesta dissertação.

- Devido à possível atualização da versão 1.0 da meta-informação proposta pelo projeto IMS, faz-se necessária a criação de uma ferramenta ou a adaptação do *framework* com o objetivo de possibilitar a criação de novos elementos para a meta-informação e a retirada de elementos obsoletos. Para dar suporte a essa modificação, a criação e exclusão de elementos da meta-informação, o padrão IMS criou o repositório de estrutura TIERS. Baseando-se em TIERS, a criação da ferramenta e a adaptação do *framework* ficam facilitadas.
- Padrão IMS prevê a comercialização dos conteúdos trocados entre os servidores de conteúdo, o que não ocorre no caso do ContentNet. O ContentNet não lida com a comercialização do conteúdo, mas tão somente com a troca de um conteúdo entre um servidor e outro. Assim, seria um aperfeiçoamento do *framework* ContentNet o desenvolvimento de uma ferramenta ou a comunicação com alguma ferramenta que comercialize o conteúdo entre o servidor que possui o conteúdo e o servidor que deseja o conteúdo antes da troca efetiva do mesmo.
- As buscas feitas no ContentNet podem ser facilitadas caso exista uma aplicação que, com base nas pesquisas feitas anteriormente, informe ao usuário do *Content Search* o grupo de

características da meta-informação mais selecionados e os dados dos elementos mais usados. Para isso, é necessário que seja guardado um arquivo de *log* com todas as pesquisas feitas no *Content Search*.

- Atualmente o usuário do *framework Content Search* não possui alternativa entre adquirir um conteúdo que procura e apontar para o conteúdo encontrado. Se essas opções estivessem disponíveis, o usuário que quisesse acompanhar a atualização do conteúdo, optaria por utilizar um *link* para o mesmo, ao invés de adquiri-lo e armazená-lo. Com isso, a alteração do *framework* para inclusão desta opção seria de grande valia.
- A interface do professor usuário da ferramenta de pesquisa foi desenvolvida baseando-se na instância do *framework* para servidores AulaNet. Os tipos de conteúdos utilizados na ferramenta são os tipos de conteúdos AulaNet. Estudando-se mais esta interface, vê-se que é possível a geração de um *framework* para construção de interfaces com o propósito de interoperabilidade de conteúdo, como proposto em Silveira[Silveira, 2000b]. Acoplar este *framework* de geração de interface ao *framework* ContentNet é um trabalho importante a ser realizado. Com este acoplamento, a implementação de um dos *hot-spots* do *framework* de interface permite que o usuário procure conteúdos de qualquer tipo, e não só dos tipos oferecidos pelo AulaNet.
- O número de usuários que testaram tanto a interface do professor quanto a do catalogador foi bastante pequeno. Por este motivo, outros testes devem ser aplicados às interfaces, utilizando-se para tanto, um número maior de usuários. De posse dos resultados destes novos testes, será possível avaliar com mais precisão as interfaces projetadas.
- Com o objetivo de validar o *design* do *framework*, com relação a seus *hot-spots* e seu *kernel*, este deve ser instanciado para outras aplicações cujas características sejam bem diferentes das características do ambiente AulaNet.

# Apêndice A

## Formulário dos Testes realizados

### A.1 Formulário utilizado no teste da interface do professor

#### Interface do professor AulaNet utilizando o ContentNet

##### Formulário:

1) Você é usuário do AulaNet?

Não

Sim

Qual é seu grau de conhecimento?

Baixo

Regular

Bom

Excelente

Qual o seu papel no ambiente?

Usuário

Desenvolvedor

Suporte

Observações:

---

---

---

---

2) Você conhece o padrão IMS?

Não

Sim

Qual é seu grau de conhecimento?

Baixo

Regular

Bom

Excelente

3) Você conhece o ContentNet? (visão do professor)

Não

Sim

Sabe para que serve?

---

---

---

---

O ContentNet é uma ferramenta de busca de conteúdos AulaNet. Compreende-se por conteúdos AulaNet um curso, uma aula, uma bibliografia, uma *webliografia*, uma tarefa, uma auto-avaliação, uma questão, um material e uma documentação. Para pesquisar um destes conteúdos, é necessário preencher algumas características do mesmo com as informações conhecidas. Em seguida, a ferramenta encarrega-se da busca deste conteúdo, informando ao usuário os resultados encontrados. O usuário escolhe um dos resultados e a ferramenta busca o conteúdo correspondente.

---

O seu objetivo é executar as tarefas descritas abaixo e não testar a ferramenta. Sendo assim, concentre-se na execução da tarefa.

Nem todas as funcionalidades da ferramenta estão disponíveis, mas, para a realização das tarefas pedidas, estão disponíveis todas as necessárias.

---

### **Tarefa 1 (Cenário):**

Você é um professor do AulaNet que deseja fazer uma pesquisa procurando uma bibliografia. As informações que você possui sobre esta bibliografia são:

- para adquiri-la não é necessário nenhum pagamento;
- sua palavra-chave é mecânica;
- o idioma no qual ela foi escrita é o português;
- o grau de dificuldade no uso da bibliografia é alto.

Imagine que você irá projetar a interface do ContentNet. Descreva os passos necessários para a busca do conteúdo descrito acima. Preocupe-se apenas com a funcionalidade da interface.

---

---

Usando o ambiente AulaNet, execute as tarefas abaixo.

**Tarefa 2:**

- a) Busque uma bibliografia que trate do assunto mecânica, que seu tamanho não ultrapasse um arquivo e que seja destinada a adolescentes do segundo grau.
- b) De posse dos resultados da busca, adquira a bibliografia que possua os atributos pedidos.

**Tarefa 3:**

- a) Busque uma aula cujo título seja "História Geral". Este conteúdo não possui nenhum custo, é de fácil compreensão, é destinado a usuários do segundo grau e, no seu tipo de interação, a informação flui do recurso para o aprendiz.
- b) De posse dos resultados da busca, adquira a aula que possua restrições ao uso e que tenha um alto nível de interatividade.

Observações:

---

---

---

---



## A.2 Formulário utilizado no teste da interface do catalogador

### Interface do catalogador AulaNet utilizando o ContentNet

#### Formulário:

1) Você é usuário do AulaNet?

Não

Sim

Qual é seu grau de conhecimento?

Baixo

Regular

Bom

Excelente

Qual o seu papel no ambiente?

Usuário

Desenvolvedor

Suporte

Observações:

---

---

---

2) Você conhece o padrão IMS?

Não

Sim

Qual é seu grau de conhecimento?

Baixo

Regular

Bom

Excelente

3) Você conhece o ContentNet?

Não

Sim

Sabe para que serve? (visão do catalogador)

---

---

---

---

O ContentNet é um ambiente que permite a busca de conteúdos educacionais AulaNet. O ContentNet baseia-se na descrição do conteúdo que o usuário deseja para buscá-lo nos diversos servidores AulaNet. Sendo assim, todos os conteúdos AulaNet precisam ter catalogada sua descrição. A descrição do conteúdo é provida pelo catalogador de conteúdos AulaNet que, pelo consumo do conteúdo e contato com o professor que o criou, consegue obter todas as informações necessárias para a catalogação.

A descrição do conteúdo é feita de acordo com o padrão IMS. No padrão IMS, a todo conteúdo, é relacionada uma série de características como, por exemplo, o título do conteúdo, sua descrição, seu formato e o usuário pretendido para o mesmo. O papel do catalogador é atribuir valores a essas características.

---

O seu objetivo é executar a tarefa descrita abaixo e não testar a ferramenta. Sendo assim, concentre-se na execução da tarefa.

Nem todas as funcionalidades da ferramenta estão disponíveis, mas, para a realização das tarefas pedidas, estão disponíveis todas as necessárias.

---

Usando o ambiente AulaNet, execute a tarefa abaixo:

Usuário: biblio senha: biblio

**Tarefa 1:**

Suponha que você é um catalogador do ambiente AulaNet. Cadastre a descrição da bibliografia "Mecânica dos Sólidos". A seguir, encontram-se as informações que você possui sobre a bibliografia.

Esta bibliografia é composta de um conjunto de documentos, é destinada a alunos do primeiro grau, seu conteúdo é de difícil compreensão e é composta por documentos hipertexto e simulações. O professor que a desenvolveu, colocou restrições para seu uso e um custo para sua aquisição.

Observações:

---

---

---

---

## Referências

- [ADL, 2000] ADL - Advanced Distributed Learning. [online]. <<http://www.adlnet.org>> [Consulta: 24 de janeiro de 2000]
- [AICC, 2000] AICC - Aviation Industry CBT Committees. [online]. 16 de janeiro de 2000. <<http://www.aicc.org>> [Consulta: 26 de janeiro de 2000]
- [ARIADNE, 2000] Alliance of Remote Instructional Authoring & Distribution Networks or Europe [online] < <http://ariadne.unil.ch/project/main.content.html>> [Consulta: 06 de março de 2000]
- [Benner, 1993] Benner, K. M.; Feather M. S.; Johson W. L.; Zorman L. A. N. Prakash, C. Rolland, and B. Pernici editors, Information Systems Development Process, IFTIP Transactions A-30, Elsevier Science Publishers, setembro 1993.
- [Carroll, 1995] Carroll, John M. "Scenario-based design: envisioning work and technology in system development." New York: Wiley, 1995.
- [CEdMA,2000] CEdMA - Computer Education Management Association. [online]. <<http://www.cedma.org>> [Consulta: 26 de janeiro de 2000]
- [Classnet, 1999] Classnet. Iowa State University Computation Center. [online]. <<http://classnet.cc.iastate.edu>>. [Consulta: 10 de novembro de 1999]

- [Cormen, 1990a] Cormen, T. H.; Leiserson C. E.; Rivest R. L. "Introduction to Algorithms"  
London: McGraw-Hill, 1990
- [Cormen, 1990b] Cormen, T. H.; Leiserson C. E.; Rivest R. L. "Introduction to Algorithms"  
London: McGraw-Hill, 1990 (pp.301-328)
- [de Souza, 1999] de Souza, C. S.; Prates, R. O.; Barbosa, S. D. J. A Method for Evaluating  
Software Communicability. Em IHC'99 - II Workshop sobre Fatores Humanos  
em Sistemas Computacionais, 1999. Campinas, S.P.
- [Fayad, 1999] Fayad, M. E., Schmidt, D. C. Johnson, R. E.. "Building Application  
Frameworks". New York: Wiley, 1999.
- [Fontoura, 1999] Fontoura, M. F. M. C., "A systematic approach to framework development",  
Tese de Doutorado, 1999, Departamento de Informática, PUC-Rio.
- [IBM, 2000] "IBM's XML Parser for Java™ (XML4J) [online]  
<<http://www.alphaworks.ibm.com/tech/xml4j>> [Consulta: 07 de março de 2000]
- [IEEE, 2000] IEEE LTSC - IEEE Learning Technology Standars Committee. [online]. 5 de  
março de 2000. <<http://ltsc.ieee.org>> [Consulta: 03 de janeiro de 2000]
- [IMS Metadata, 2000] IMS Metadata. [online]. IMS Web Team, 25 de fevereiro de 2000.  
<<http://www.imsproject.org/metadata>> [Consulta: 06 de março de 2000]
- [IMS, 2000a] IMS Global Learning Consortium, Inc.[online]. IMS Web Team, 12 de janeiro  
de 2000. <<http://www.imsproject.org>> [Consulta: 19 de janeiro de 2000]
- [IMS, 2000b] IMS Faqs [online] IMS Web Team <<http://www.imsproject/faqs.html>>  
[Consulta: 04 de março de 2000]

- [Lucena, 1998] Lucena, C. J. P., Fuks, H. et al.; "AulaNet—An Environment for the Development and Maintenance of Courses on the Web". Em: ICEE'98—International Conference On Engineering Education, 1998. Rio de Janeiro.
- [Lucena, 1999a] Lucena, C. J. P., Fuks, H. et al.; "AulaNet technologies: future trends." Em: ICECE'99 - International Conference on Engineering and Computer Education, 1999, Rio de Janeiro.
- [Lucena, 1999b] Lucena, C. J. P., Fuks, H. et al.; "AulaNet: ajudando professores a fazerem o seu dever de casa." Em: Anais do XXVI Semish - Seminário Integrado de Software e Hardware, Sociedade Brasileira de Computação (sbc), 1999. Rio de Janeiro.
- [Lucena, 1999c] Lucena, C. J. P., Fuks, H. et al.; "O AulaNet e as novas tecnologias de informação aplicadas à educação baseada na Web." V Congresso Internacional de Educação À Distância, Associação Brasileira de Educação À Distância (abed), 1999. Rio de Janeiro.
- [Mattsson, 1996] Mattsson, M. "Object-Oriented Frameworks: A Survey of Methodological Issues", M.Sc. Dissertation, Department of Computer Science and Business Administration, University College of Karlskrona/Ronneby, LU-CS-96-197, 1996.
- [MMI, 2000] Metadata for Multimedia Information [online] Luc Van den Berghe, 20 de agosto de 1998 [Consulta: 06 de março de 2000]
- [Norman, 1988] Norman, D. "Psychology of Everyday Things". BasicBooks. HarperCollins Publishers, 1988
- [Pathware, 2000] Pathware 4 Attain. [online]. Macromedia, Inc. <[http://www.macromedia.com/software/pathware/productinfo/whitepapers/open\\_standards.html](http://www.macromedia.com/software/pathware/productinfo/whitepapers/open_standards.html)> [Consulta: 19 de janeiro de 2000]

- [Prates, 2000] Prates, R. O.; de Souza, C.S.; Barbosa, S.D.J. "A Method for Evaluating Software Communicability". Interact, 2000. New York.
- [RDF Model, 1999] RDF Model and Syntax Specification. [online] Ora Lassila, Ralph R. Swick, 24 de fevereiro de 1999. <<http://www.w3.org/TR/REC-rdf-syntax>> [Consulta: 03 de agosto de 1999]
- [RDF Schema, 1999] RDF Schema Specification. [online]. Dan Brickley, 03 de março de 1999. <<http://www.w3.org/TR/WD-rdf-schema>>. [Consulta: 20 de julho de 1999]
- [SGML, 1999] SGML. [online]. Charles F. Goldfarb, 27 de novembro de 1999. <<http://www.sgmlsource.com>> [Consulta: 10 de dezembro de 2000]
- [Silveira, 2000a] Silveira, M. S.; Monteiro, C. C. O.; Barbosa, S. D. J.; Souza, C. S.. "The Role of Design-Generated Scenarios in Developing Web Applications and their Help Systems". MCC09/00. Departamento de Informática. PUC-RJ. Rio de Janeiro, 2000.
- [Silveria, 2000b] Silveira, M. S "Frameworks para as Interfaces de Busca e Catalogação de conteúdos do ContentNet". MCC13/00. Departamento de Informática. PUC-RJ. Rio de Janeiro, 2000.
- [Sun, 2000] Sun Microsystems, Inc. [online] 27 de outubro 1999 <<http://www.sun.com/products-n-solutions/edu/announcements/toolkit.html>> [Consulta: 07 de março de 2000]
- [Tanenbaum, 1995] Tanenbaum, A. S.. "Distributed Operating Systems". New Jersey: Prentice Hall, 1995 (pp.: 50-68)



- [TIERS, 1999] Using TIERS [online]  
<<http://www.imsproject.org/technical/metadata/did163.html>> [Consulta: 02 de junho de 1999]
- [Virtual-U, 1999] Virtual-U. Simon Fraser University. [online]. <<http://virtual-u.cs.sfu.ca/vuweb>>. [Consulta: 10 de novembro de 1999]
- [W3C Metadata, 1999] Metadata at W3C. [online]. Ralph Swick, 20 de novembro de 1999.  
<<http://www.w3.org/metadata>> [Consulta: 05 de dezembro de 1999]
- [W3C RDF, 1999] W3C Resource Description Framework.[online]. Ralph Swick, 06 de dezembro de 1999. <<http://www.w3.org/RDF/>> [Consulta: 20 de julho de 1999]
- [W3C, 2000] W3C - World Wide Web Consortium. [online]. 24 de janeiro de 2000.  
<<http://www.w3.org>> [Consulta: 25 de janeiro de 2000]
- [Web-Couse, 1999] Web-Course-in-a-Box. Virginia Commonwealth Center. [online].  
<<http://views.vcu.edu/wcb>>. [Consulta: 10 de novembro de 1999]
- [WebCT, 1999] WebCT. University of British Columbia. [online]. <<http://homebrew.cs.ubc.ca/webct>>.[Consulta: 10 de novembro de 1999]
- [XML, 2000] eXtensible Markup Language. [online]. Dan Connolly, 25 de janeiro de 2000.  
<<http://www.w3.org/XML/>> [Consulta: 26 de janeiro de 2000]