

4 Integração DLMS e LMS

Neste capítulo define-se inicialmente a arquitetura proposta, que visa integrar repositórios de Bibliotecas Digitais e de Ambientes de Aprendizagem, podendo os mesmos estar armazenados em diferentes modelos e gerenciadores de banco de dados, em diferentes sistemas operacionais, etc. Em seguida, detalha-se a arquitetura, demonstrando cada camada utilizada na abordagem distribuída baseada em mediadores e tradutores. Por último, apresenta-se o caso de utilização desta arquitetura em UML.

4.1.Arquitetura Proposta

Como citado no Capítulo 2, a integração da informação é um dos problemas estudados na Área de Banco de Dados e tem por objetivo proporcionar uma visão homogênea coerente de dados armazenados em múltiplos e heterogêneos repositórios de dados.

A proposta de arquitetura desenvolvida neste trabalho visa integrar repositórios de Sistemas de Bibliotecas Digitais e Sistemas de Aprendizagem. É desejável que o usuário tenha uma visão única, homogênea e integrada, de modo que qualquer consulta feita pelo usuário implique em um resultado que recupere informações de todas as fontes de dados relativas à consulta. Deste modo, o usuário pode utilizar o LMS para fazer consultas ao material didático via LCMS, bem como via DLOMS. Alternativamente, o usuário pode utilizar o DLMS para fazer consultas via DLOMS, bem como via os LCMS. Finalmente, também haverá a possibilidade de o usuário fazer a consulta através de uma nova interface que engloba ambos os ambientes e retorna os dados integrados, sendo que todas estas interfaces acessam a ontologia global de integração. A Figura 10 apresenta estas alternativas conforme a arquitetura proposta.

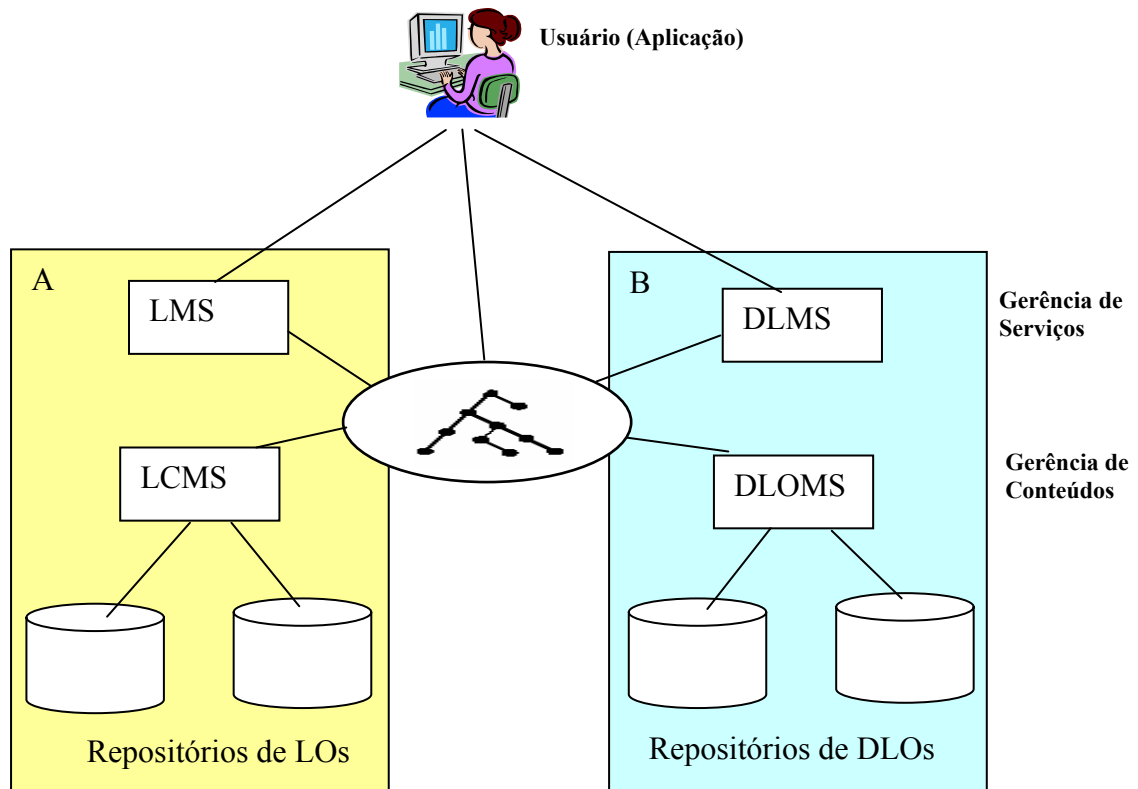


Figura 7 - Arquitetura Proposta

Uma das formas de se tratar esta integração, apresentada na Figura 10, é através da utilização de Mediadores (*Mediators*) e Tradutores (*Wrappers*) (Tzitzikas, 2002; Wiederhold, 1995), como apresentado no Capítulo 2.

A utilização dos Tradutores para encapsulamento dos repositórios de dados permite que eles sejam mantidos e evoluam com certa independência. Os Tradutores são utilizados como canal de comunicação independente do formato e da forma de implementação dos repositórios de dados.

Os Mediadores são utilizados para prover o acesso uniforme e integrado às informações, através dos Tradutores. Um mediador possui um conjunto de articulações que representam as relações entre seus termos e os termos de seus repositórios. O Mediador utiliza essas articulações para “montar” as consultas a serem submetidas aos repositórios de dados, com base nas consultas globais. Da mesma forma, o Mediador deve juntar as respostas de cada repositório de dados e retornar um resultado consolidado para a aplicação solicitante.

Através da aplicação da tecnologia de Mediadores pode-se garantir a interconectividade e interoperabilidade entre as bases.

A arquitetura baseada em Mediadores é a mais recomendável para a integração dos repositórios de LMS e DLMS, pelos seguintes motivos:

- Os esquemas dos metadados das DLs e dos Sistemas de Aprendizagem obedecem a padrões rígidos e são pouco mutáveis ao longo do tempo. Neste sentido, a arquitetura de mediador é a melhor, pois não terá muitas alterações dos esquemas.
- Somente as operações referentes a consultas são utilizadas na Integração dos Dados, cabendo aos sistemas locais a manutenção de seu acervo;
- Uma vez definido o modelo de dados do Mediador, acrescentar um novo repositório à Arquitetura pode ser resolvido com o desenvolvimento de um novo tradutor específico ou a utilização de um já existente, no caso desse tradutor obedecer aos padrões do repositório.
- Novas tecnologias, como ontologias, possibilitam maior interoperabilidade e melhor tratamento semântico na integração, constituindo boas opções na implementação da arquitetura.

Na sessão a seguir são detalhados os componentes tecnológicos da arquitetura de integração.

4.2.Componentes da arquitetura

A arquitetura de integração é composta de 4 camadas, apresentadas na Figura 11, sendo uma de Aplicação, uma de Mediação, uma de Tradução e uma de Fonte de Dados, onde estão os repositórios de cada sistema. A seguir, será detalhada cada camada da arquitetura.

4.2.1.Camada de Aplicação

A camada de Aplicação pode ser composta por diferentes tipos de aplicações, específicas de cada ambiente operacional, possuindo interfaces gráficas ou de caracteres no ambiente *Web*. Podem ser desenvolvidas com

linguagens do tipo C++, Java, PHP, Asp, etc. Esta camada acessa uma representação integrada das fontes de dados, através de uma linguagem de consulta. A fim de permitir uma consulta integrada aos repositórios dos Sistemas de Bibliotecas Digitais e Sistemas de Aprendizagem, é importante adicionar a funcionalidade de consulta integrada a estes ambientes, permitindo ao usuário fazer consultas aos conteúdos disponíveis e receber a resposta integrada.

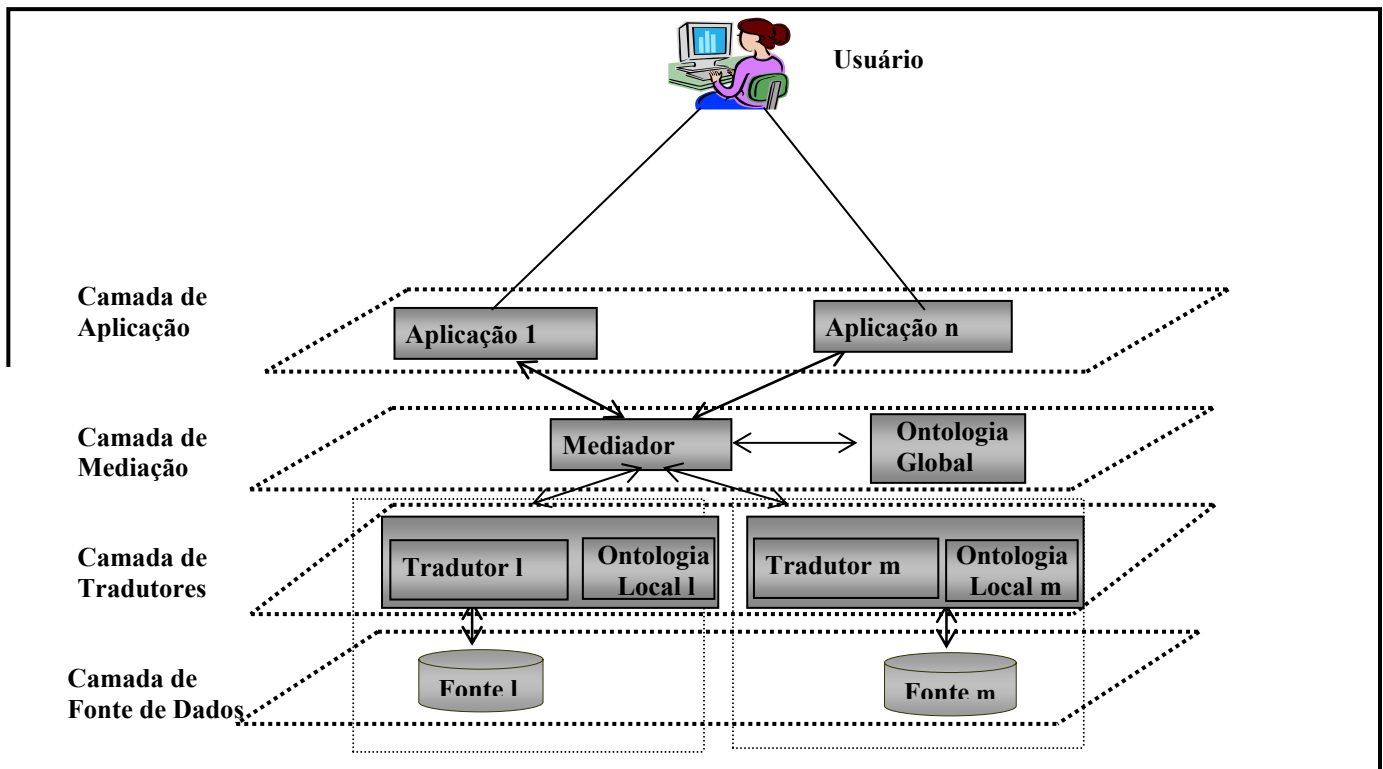


Figura 8 - Camadas com os componentes da arquitetura

4.2.2. Camada de Mediação

A camada de Mediação, através de sua linguagem padrão para aplicações, interpreta e traduz as consultas dos usuários, permitindo-lhes acessar fontes de dados de forma transparente e uniforme.

A camada de Mediação recebe a consulta da camada de Aplicação e faz a homogeneização das fontes de dados, que, em geral, são heterogêneas. As consultas são validadas face ao esquema global, que, por sua vez, estará definido através de uma ontologia global. O mediador identifica quais as fontes serão consultadas, transformando as consultas dos usuários em subconsultas, enviando-

as, através de uma codificação padrão, para o Tradutor correspondente de cada Fonte de Dados.

A camada Mediação é a responsável por disponibilizar uma API (*Application Program Interface*), que possibilita às diversas aplicações consultarem diferentes Fontes de Dados, como se estivessem em um ambiente local e homogêneo. Para cumprir esse objetivo, um Mediador deve possuir:

- Um modelo de dados;
- Uma linguagem de consulta para interação com as aplicações;
- Um protocolo de comunicação para envio de consultas e recebimento de dados dos Tradutores.

Neste trabalho será detalhado o modelo de dados de integração.

4.2.2.1. Modelo de dados

Para dar suporte à Arquitetura de Mediadores é necessário um modelo de dados global (comum) que seja flexível, que ofereça uma coleção rica de estruturas e que forneça informações sobre as estruturas (meta-informações) (Vidal1997).

Neste trabalho foi definido um modelo comum com os principais conceitos envolvidos nos padrões de metadados utilizados em Ambientes de Aprendizagem e em DLs. Este modelo foi representado através de ontologia, provendo uma visão integrada dos repositórios que fizerem parte da integração. Esta ontologia foi definida na linguagem OWL (OWL,2004), que é uma linguagem voltada para a descrição de classes e relacionamentos que são inerentes a documentos e aplicações *Web*. Esta linguagem pode também ser utilizada para formalizar um domínio através da definição de classes e suas propriedades, definir indivíduos e afirmar propriedades sobre eles, além de prover raciocínios lógicos sobre estas classes e indivíduos, de acordo com o grau permitido pela semântica formal da OWL.

O modelo de metadados global da arquitetura, representado na Figura 12, foi baseado no MARC, LOM e Dublin Core, padrões de metadados mais utilizados nas áreas de Biblioteca Digital e *E-learning*, considerando, também, a experiência adquirida na utilização destes padrões.

A ontologia global e as ontologias locais (de cada padrão), apresentadas na próxima seção, foram definidas utilizando a ferramenta Protégé (Protégé, 2006). A representação da ontologia Global em OWL, se encontra no Apêndice D.

Ontologia Global



Figura 9 – Modelo de Dados de Integração do Mediador

Neste modelo foi definido um esquema onde tem-se as classes TipoPesquisa, Repositórios e Documento_RIOS. Este esquema atenderá às consultas por Autor, Título, Assunto, Idioma e Data, definidos pela classe TipoPesquisa e suas subclasses PAutor, PTitulo, PAssunto, PIdioma e PData respectivamente. Posteriormente, o modelo pode ser facilmente estendido para outros tipos de consultas.

Na classe Repositório estão todos os repositórios que poderão ser acessados na integração, e na classe Documento_RIOS tem-se os principais elementos de metadados do modelo de integração.

Além de definir o modelo global (ontologia global), é necessário estabelecer correspondências entre o esquema global e os esquemas das fontes locais. Para isto, foram definidas regras de mapeamento de modo a garantir a visão integrada das fontes de dados. Todas as consultas requeridas pelo usuário são construídas no formato do modelo global pelo mediador e, através do mapeamento, estas consultas são transformadas em sub-consultas baseadas nos padrões de metadados de cada base de dados contida na arquitetura.

Neste trabalho, o mapeamento foi representado através de ontologias, permitindo que o usuário obtenha acesso global aos repositórios sem a necessidade de conhecer a estrutura exata dos padrões de metadados. Utilizou-se para este mapeamento a linguagem OWL, que descreve a ontologia comum e possui propriedades especiais para a definição de equivalências entre as classes. Essas propriedades definem que uma classe ou propriedade particular de uma ontologia seja dita equivalente a uma determinada classe ou propriedade de outra ontologia.

Após a definição da ontologia global, em cada tipo de dado foi definida a sua correspondência com as fontes de dados que compõem a arquitetura. Por exemplo, na ontologia global tem-se a classe Assunto, que, no padrão MARC, está representado pela classe Assunto, que tem as subclasses AssuntoNomePessoal600, AssuntoEntidadeColetiva610, AssuntoEventos611, AssuntoTituloUniforme630, AssuntoTopico650, AssuntoNomeGeografico651, TermoControlado653; no padrão LOM é representado pelas classes PalavraChave e PalavraChaveTax; e no padrão Dublin Core está representado pela classe Assunto. Para uma melhor compreensão, na Figura 13 é apresentado um exemplo de mapeamento realizado neste trabalho, onde tem-se as regras de mapeamento da ontologia global para as ontologias locais da classe Assunto definidas no Protégé.

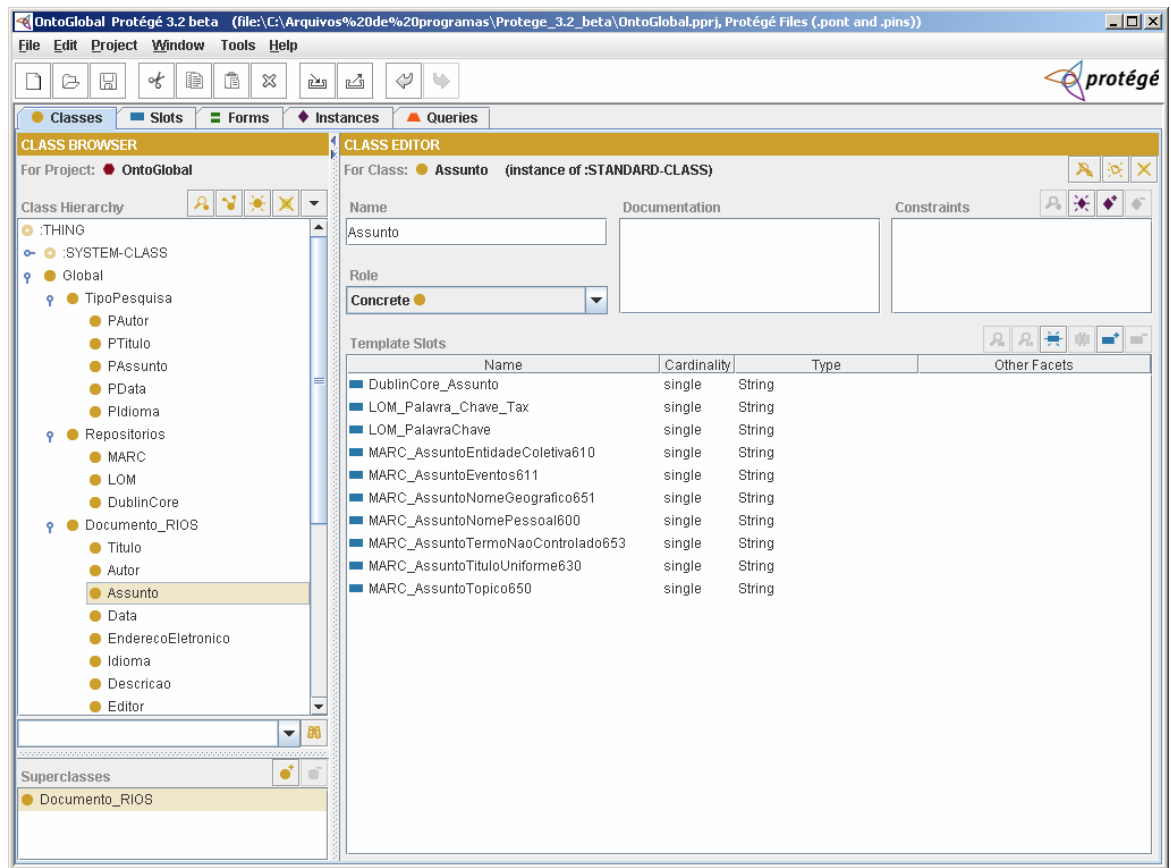


Figura 10 – Mapeamento de assunto do esquema global para os correspondentes termos dos esquemas locais

Pode-se observar que, para a classe Assunto na ontologia global, tem-se o mapeamento dos assuntos das ontologias locais, representados através de Slots, que é a propriedade que faz a ligação entre duas ou mais classes, ou seja, estas têm o mesmo significado. As ontologias locais, do Dublin Core, LOM e MARC, serão apresentadas na próxima seção.

4.2.3. Camada de Tradutores

A camada de Tradutores recebe as subconsultas do mediador e acessa as regras de mapeamento do esquema global para o esquema da fonte de dados local, transformando as consultas para a linguagem correspondente de cada fonte de dados. O resultado da consulta é enviado ao mediador para que este possa centralizar as respostas dos diversos Tradutores, executar as operações necessárias (União, Interseção, Diferença, Seleção e/ou Projeção) e, então, enviá-las aos usuários que realizaram as consultas.

Como já definido, neste trabalho os esquemas locais estão representados por ontologia. As ontologias de cada padrão reúnem todos os conceitos presentes nas especificações dos mesmos. Nas figuras 14, 15 e 16, são apresentadas as ontologias locais para subconjuntos dos padrões de metadados Dublin Core, MARC e LOM, que, de acordo com observação empírica, são os mais utilizados nas áreas de Biblioteconomia e *E-learning*.

Na Figura 14 é apresentada a ontologia do padrão de metadado Dublin Core, padrão utilizado nos sistemas LMS e DLMS, que é composto por quinze elementos, onde cada um deles é uma subclasse da classe Dublin Core. Estes elementos estão detalhados no Apêndice C.

Ontologia Dublin Core

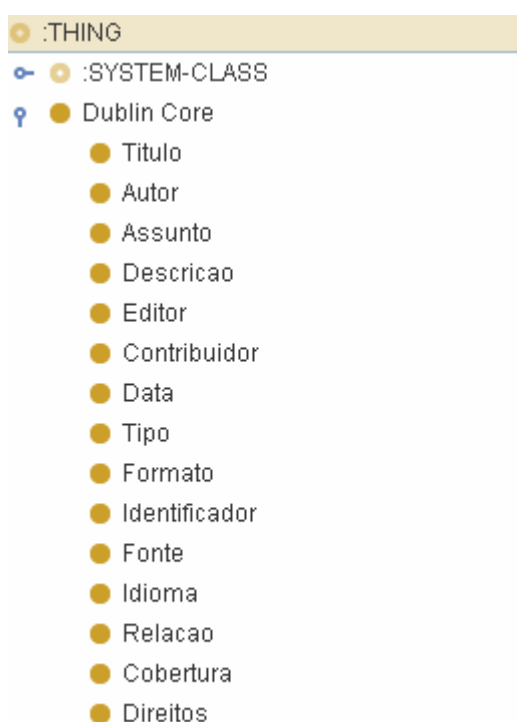


Figura 11 - Ontologia Dublin Core

Na Figura 15 é apresentada a ontologia do padrão de metadado MARC, padrão utilizado nos sistemas de bibliotecas em geral para representar os dados bibliográficos. Nos sistemas de bibliotecas digitais, que estão vinculados ao catálogo bibliográfico, ele é composto de *tags* (etiquetas), que representam os elementos bibliográficos de qualquer tipo de material. Por ser muito abrangente,

Ontologia LOM:**Figura 13- Ontologia LOM**

Na seção a seguir, os casos de uso do modelo de integração são formalizados em modelo UML.

4.3.Caso de Uso do Sistema Integrador

Nesta seção é apresentado o diagrama de casos de uso do sistema integrador, proposto pela arquitetura na seção anterior. O diagrama, além de oferecer uma visão geral das funcionalidades, configura-se como objeto de documentação do sistema, em conformidade com a linguagem UML (*Unified Modeling Language*) (BOOCH, 2000).

Foi identificado inicialmente, no nível conceitual, cinco atores que executam quatro casos de uso. Como pode-se verificar na Figura 17, os atores Administrador, Professor e Aluno são especializações do ator Usuário Sistema, ou seja, tipo de usuário que acessará o sistema através da aplicação de integração

desenvolvida neste trabalho. Neste caso, executa-se o caso de uso Validar Usuário. Já os Usuários LMS e Usuários DLMS podem fazer consultas através dos DLMS ou LMS, sendo que serão validados nos próprios sistemas. Todos os usuários precisam estar validados para utilizar o ambiente. Depois de validados, eles terão acesso à consulta integrada ou a administração do ambiente. A seguir, será feita uma breve descrição dos casos de uso.

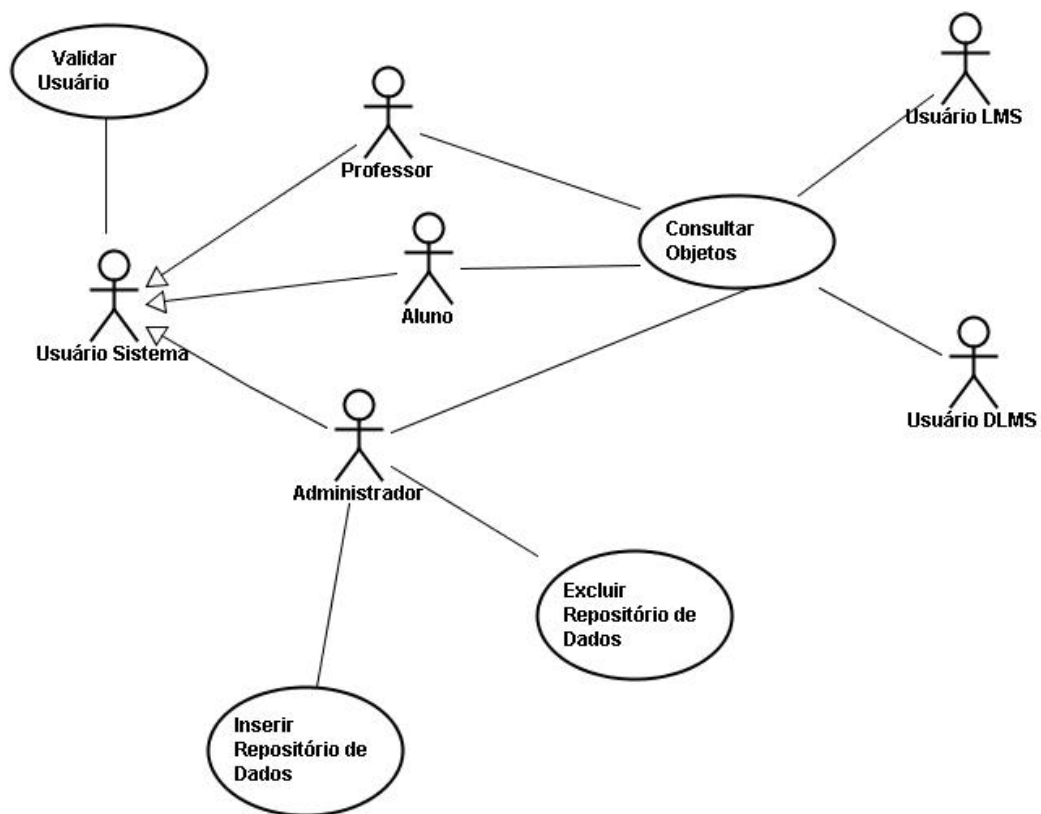


Figura 14 – Diagrama de Casos de Uso da Arquitetura Proposta

Validar Usuário

Consiste das funcionalidades para validação do terno <usuário, senha> de um usuário para posterior acesso aos recursos do ambiente e etapas dos processos aos quais estão vinculados.

Consultar Objetos

O principal caso de uso do sistema integrador consiste na pesquisa integrada aos objetos dos sistemas DLMS e LMS. A consulta é submetida pelo usuário através de uma interface gráfica e o sistema opera sobre as fontes integradas, gerenciando a entrega das sub-consultas e a composição dos resultados obtidos.

Inserir Repositórios de Dados

A fonte de dados e seu respectivo tradutor são integrados ao sistema através de informações relevantes para o funcionamento do mesmo, assim como, o mapeamento do esquema global para o local. Desta forma, a fonte de dados será considerada no processamento da consulta integrada.

Excluir Repositórios de Dados

A fonte de dados e seu respectivo tradutor são removidas do sistema, não sendo consideradas no processamento da consulta integrada.

Os quadros 1, 2, 3 e 4 apresentam os fluxos típicos dos casos de uso do diagrama da Figura 17.

Caso de Uso	Validar Usuário
Ator(es)	Usuário Sistema
Descrição	Validação de usuário através do parâmetro <usuário,senha> para acesso ao Sistema integrador de repositórios.
Curso Típico dos Eventos	
Passo	Descrição
1	Sistema exibe janela de fornecimento do par usuário/senha
2	Usuário sistema informa usuário/senha
3	Sistema valida usuário/senha
4	Sistema recupera a categoria de usuário, o conjunto de operações que pode executar no ambiente

	* Fim do caso de uso *
--	------------------------

Tabela 3 – Descrição do Caso de Uso Validar Usuário

Caso de Uso	Consultar Objetos
Ator(es)	Professor, Aluno, Usuário LMS, Usuário DLMS
Descrição	A consulta é submetida pelo usuário através de uma interface gráfica e o sistema opera sobre as fontes integradas, gerenciando a entrega das subconsultas e a composição dos resultados obtidos.
Curso Típico dos Eventos	
Passo	Descrição
1	Sistema exibe tela de consulta
2	Aluno/Professor escolhe o tipo de consulta, os repositórios a serem pesquisados e digita o argumento de pesquisa
3	Sistema envia para o mediador o argumento de pesquisa e os repositórios que serão pesquisados
4	Mediador redefine a consulta global em subconsultas, faz o mapeamento e as distribui para os tradutores
5	Tradutor recebe a subconsulta do Mediador e, mediante conhecimento do esquema da base local, executa a recuperação dos dados da pesquisa
6	Tradutor recebe as respostas dos repositórios locais, traduz estas respostas para o modelo de consulta global e retorna a resposta para o mediador
7	Mediador consolida as respostas dos tradutores e envia a resposta integrada para a aplicação de consulta
	* Fim do caso de uso *

Tabela 4 - Descrição do Caso de Uso Consultar Objetos

Caso de Uso	Incluir repositório de dados
Ator(es)	Administrador

Descrição	Inclusão de repositórios de dados através de um registro de informações relevantes para o funcionamento do sistema. Desta forma, o repositório de dados será considerado no processamento da consulta.
Curso Típico dos Eventos	
Passo	Descrição
1	Administrador envia para o sistema as informações relativas ao repositório de dados que será incluído no esquema global da arquitetura de integração
2	Sistema inclui no modelo global (ontologia) as informações referentes ao repositório de dados que será incluído
3	Administrador envia para o sistema as informações relativas ao tradutor que será incluído.
4	Sistema inclui novo tradutor e o esquema local.
5	Administrador envia para o sistema as informações relativas ao mapeamento do esquema global para o esquema local do tradutor do novo repositório de dados.
	* Fim do caso de uso *

Tabela 5 - Descrição do Caso de Uso Incluir Repositório de Dados

Caso de Uso	Excluir repositório de dados
Ator(es)	Administrador
Descrição	Exclusão de repositórios de dados. O repositório de dados não existirá no modelo de integração nem, portanto, no processamento da consulta.
Curso Típico dos Eventos	
Passo	Descrição
1	Sistema solicita informações do repositório a ser removido
2	Administrador envia para o sistema as informações relativas ao repositório de dados que será excluído da arquitetura de integração
3	Sistema verifica se repositório já existe no modelo de integração
4	Sistema exclui do modelo global (ontologia) do mediador as

	informações referentes ao repositório de dados a ser excluído
5	Sistema exclui o tradutor do repositório de dados a ser excluído e o mapeamento do esquema global para o esquema local.
	* Fim do caso de uso *

Tabela 6 - Descrição do Caso de Uso Excluir Repositório de Dados

Neste capítulo foi apresentada a arquitetura proposta e suas camadas tecnológicas, bem como alguns exemplos aplicados a este trabalho. No próximo capítulo será apresentado o estudo de caso que utiliza esta arquitetura.