6
Desenvolvimento do Protótipo

Este capítulo trata do desenvolvimento de um protótipo para a arquitetura proposta. Inicialmente, será feita uma descrição de um estudo de caso ao qual a arquitetura é aplicada e em seguida é apresentada a especificação do protótipo, exibindo diagramas UML. Na parte final, são discutidos os detalhes de tecnologia envolvidos na implementação.

6.1.
Estudo de Caso

Esta seção apresenta um estudo de caso para aplicação do sistema LORIS. Trata-se da comunidade que compõe o projeto PGL (*Partnership in Global Learning*) [3].

6.1.1.
Partnership in Global Learning

O projeto PGL representa uma iniciativa internacional idealizada para produzir tecnologia avançada e educação distribuída em escala global. É uma colaboração entre escolas de ensino médio, universidades e empresas voltadas para pesquisa, desenvolvimento de aplicações e treinamentos na área de *e-learning*. Esta parceria pretende criar uma grande comunidade onde, a tecnologia contribuirá no desenvolvimento econômico, social e cultural.

As atividades do PGL incluem treinamento de professores do ensino médio em projeto instrucional on-line, desenvolvimento de conteúdo e educação superior/pesquisa. As universidades que fazem parte do projeto são instituições muito prestigiadas e com experiência em projetos de *e-learning*.

Os membros fundadores do PGL têm como objetivo comum o compartilhamento de seus materiais didáticos. Os membros deste grupo representam uma comunidade de *e-learning* de fato. As instituições que consistem nos membros fundadores do PGL são:

PUC-Rio - Certificação Digital Nº 0310839/CA

- University of Florida (UFL), Gainsville, USA.
- Fundação Getúlio Vargas (FGV), São Paulo, Brasil.
- Universidade Estadual de Campinas (Unicamp), Campinas, Brasil.
- Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brasil.
- Instituto Tecnológico de Estudos Superiores de Monterrey (ITESM), Monterrey, México.

6.1.2. Ambiente PGL

Como forma de avaliar o funcionamento da arquitetura proposta foi desenvolvido um protótipo a ser executado num ambiente que representa a participação de três sítios do PGL (A, B e C, que correspondem às instituições ITESM, UFL e FGV, respectivamente). A figura 6.1 apresenta a configuração adotada nesse ambiente e ilustra um processo de troca de metadados e LOs, onde um usuário submete uma consulta ao mediador que, por conseguinte, submete a consulta aos demais repositórios da rede.

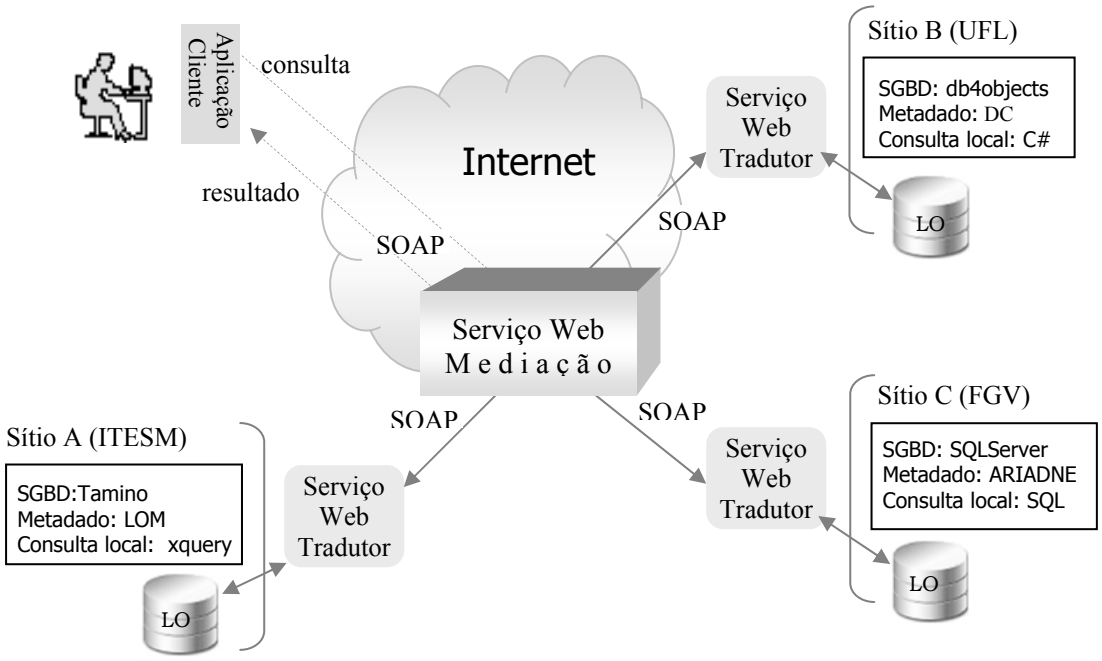

Figura 6.1: Configuração do ambiente.

PUC-Rio - Certificação Digital Nº 0310839/CA

Conforme ilustrado na Figura 6.1, o protótipo inicial consta do desenvolvimento de tradutores para os três sítios mencionados anteriormente, além do desenvolvimento dos componentes envolvidos na mediação.

6.2. Análise do Sistema LORIS

Esta seção descreve a especificação do protótipo desenvolvido para o LORIS. A subseção 6.2.1 apresenta os subsistemas do LORIS. Cada subsistema é composto de módulos, para os quais são apresentados os diagramas de classe e seqüência em UML.

6.2.1. Subsistemas do LORIS

Esta seção descreve a documentação do sistema LORIS, tendo sido dividido em duas partes: subsistema de mediação, subsistema de tradução e subsistema GUI (aplicação gráfica).

6.2.1.1. Subsistema de Mediação

Este subsistema compreende os módulos responsáveis pela mediação das fontes. É subdividido em: módulo de mediação principal e módulo de sub-mediação.

Módulo de Mediação Principal

O módulo de mediação principal recebe a consulta submetida pelo usuário através da interface gráfica (subsistema GUI ou aplicação externa). É responsável pelo processamento da consulta e sua reescrita para o modelo correspondente a cada padrão.

No diagrama de classes ilustrado na figura 6.2, as classes em amarelo compõem o módulo de mediação principal (pacote *mediation*), enquanto as demais classes pertencem a outros módulos.

PUC-Rio - Certificação Digital Nº 0310839/CA

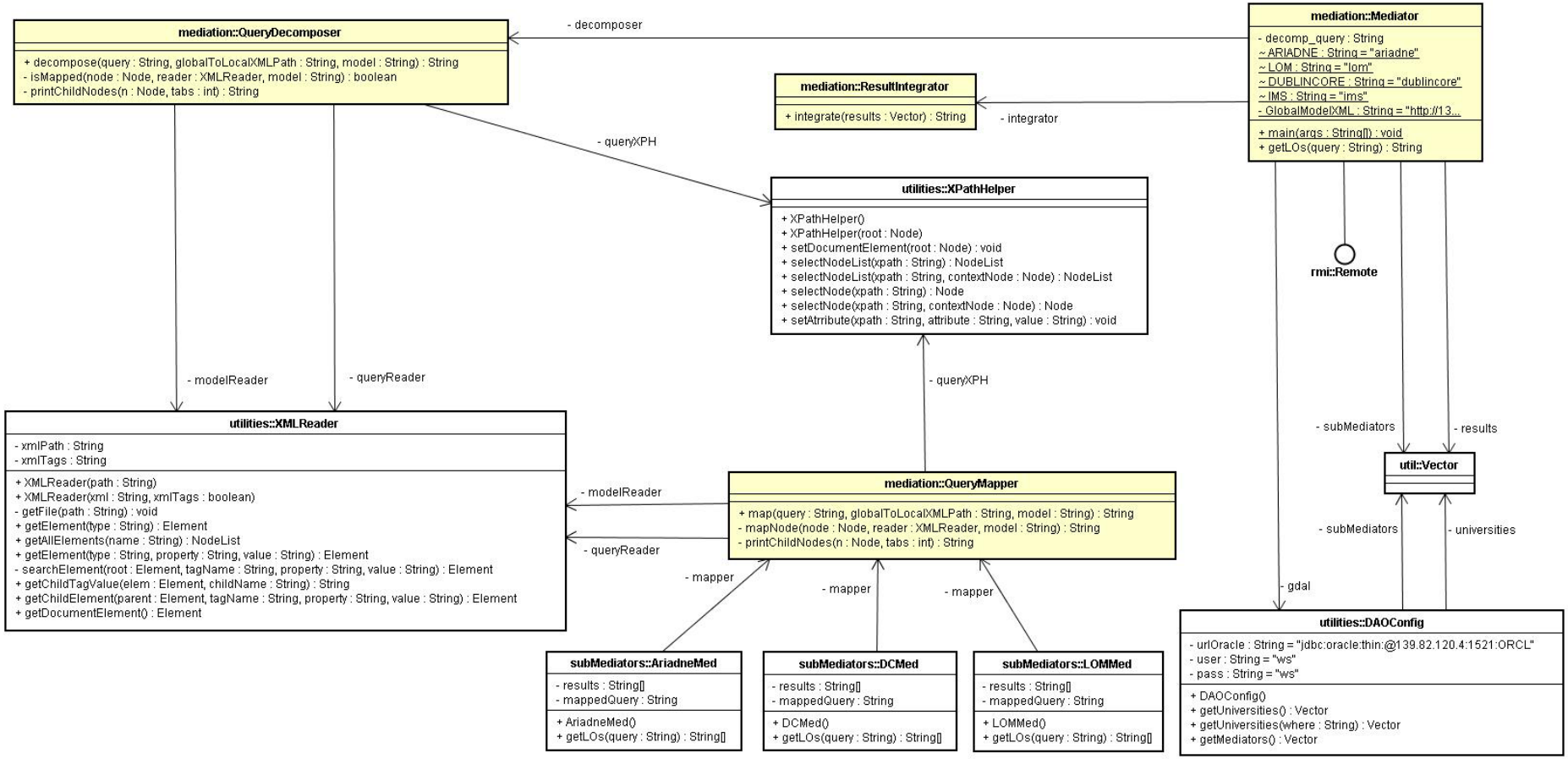

Figura 6.2: Diagrama de Classes da Mediação Principal

A classe *Mediator* é a classe principal deste módulo, sendo responsável pela gerência da mediação. O método *getLOs* recebe o conjunto de caracteres da consulta submetida pelo usuário e instancia as classes *QueryDecomposer* e *ResultIntegrator*. A partir disso, o método *decompose* (classe *QueryDecomposer*) é invocado para realizar a decomposição da consulta (reescrita) em sub-consultas para cada padrão em questão. Em seguida, o método *integrate* (classe *ResultIntegrator*) é invocado, sendo responsável pela integração dos resultados retornados pelas fontes.

A classe *Mediator* contém as instâncias que representam os submediadores, em *submediators* (classe *Vector*). Os submediadores serão detalhados no módulo sub-mediação.

A classe *QueryMapper* dispõe os métodos *map* e *mapNode* que são responsáveis pelo mapeamento da consulta do modelo global para o modelo do padrão correspondente.

PUC-Rio - Certificação Digital Nº 0310839/CA

Módulo de Submediação

O módulo de submediação é responsável pelo processo de mediação das fontes de um determinado padrão de metadados. Assim sendo, este módulo contém submediadores Ariadne, DC e LOM.

A figura 6.3 apresenta o diagrama de classes do módulo de sub-mediação (pacote *subMediators*). A classe abstrata *SubMediator* reúne os dados e métodos que cada sub-mediador precisa para realizar sua mediação. O atributo *xmlPath* (*String*) representa o caminho para o documento XML que contém o modelo de correspondência. O atributo *model* (*String*) indica o modelo (padrão) do sub-mediador. O método *getLOs* deve ser redefinido em toda classe herdeira da classe *SubMediator*. A classe armazena um vetor (classe *Vector*) que contém as universidades que adotam o padrão do sub-mediador em questão.

As classes *AriadneMed*, *DCMed* e *LOMMed* representam os sub-mediadores dos padrões Ariadne, DC e LOM, respectivamente. Essas classes invocam os métodos *map* e *mapNode* da classe *QueryMapper* (apresentada no módulo de mediação principal), com o objetivo mapear a consulta global para seu padrão correspondente.

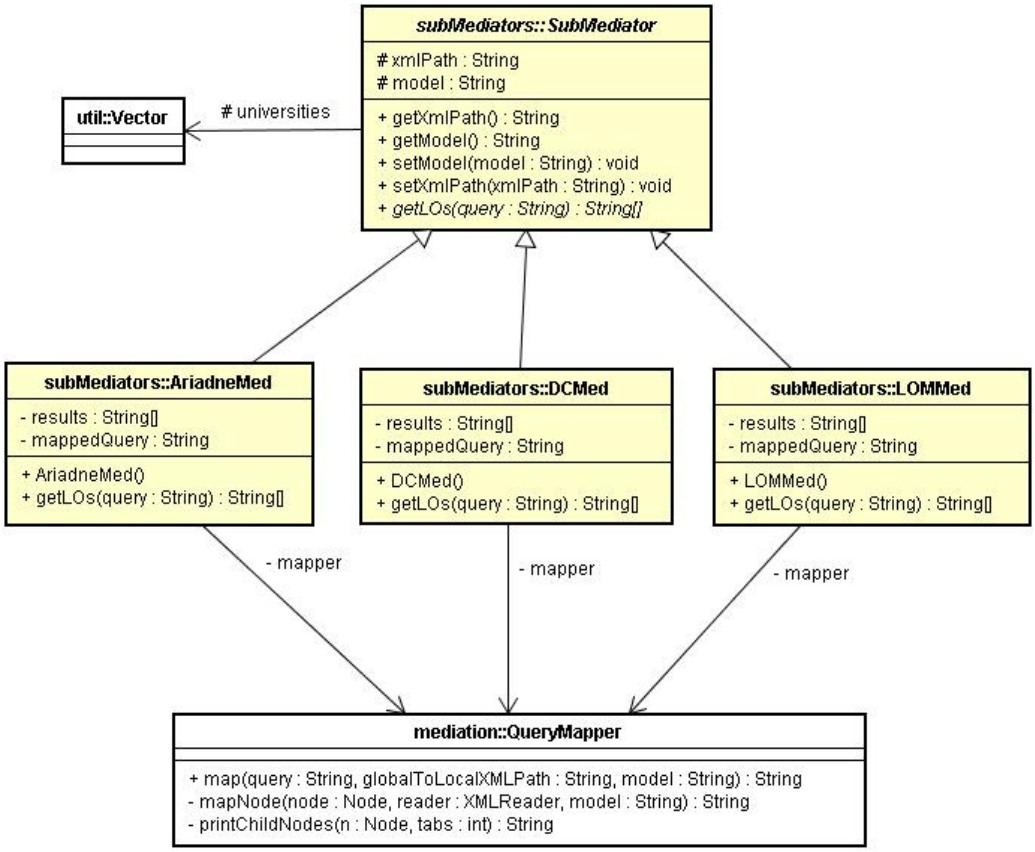

Figura 6.3: Diagrama de Classes do módulo de sub-mediação

6.2.1.2. Subsistema de Tradução

Este subsistema compreende as classes responsáveis pela chamada aos serviços *Web* providos pelos tradutores das fontes. A figura 6.4 apresenta o diagrama de classes do pacote *wrappers*.

As classes *ITESMStub*, *FloridaStub* e *WrapperFGVStub* foram geradas a partir das WSDLs dos tradutores dos sítios A, B e C, respectivamente. Estas classes implementam o método *execute(String xmlQuery)* da interface *Wrapper*, responsável pela tradução da consulta para o modelo de dados da fonte e pela conversão do resultado para o modelo global.

PUC-Rio - Certificação Digital Nº 0310839/CA

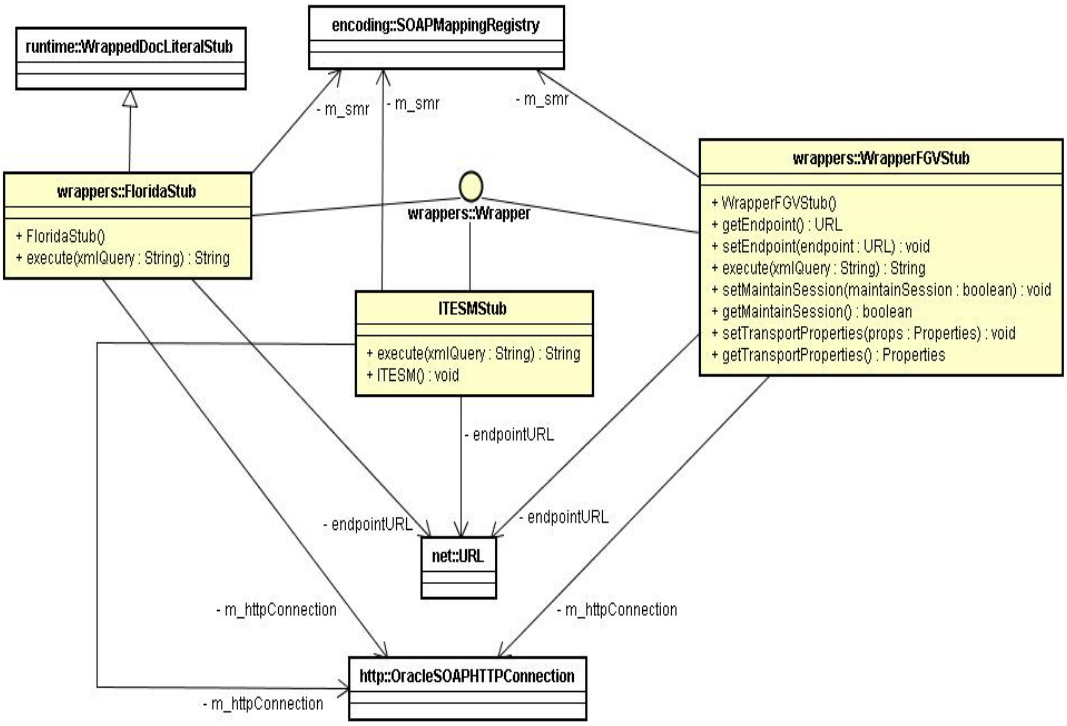

Figura 6.4: Diagrama de classes da tradução das fontes.

6.2.1.3. Subsistema GUI

Este subsistema compreende a aplicação da interface gráfica disponível como porta de entrada para o LORIS. Essa interface representa uma via de acesso comum para o serviço de mediação, contudo o acesso pode dar-se também através de uma aplicação externa.

A figura 6.5 apresenta o diagrama de classes do subsistema GUI. As classes *LORISWSStub* e *MediatorServlet* compõem o pacote *gui*. *MediatorServlet* representa o *servlet* que atende às requisições *Web* via o método *doPost*, o qual invoca o método *query* responsável pela consulta ao LORIS, os métodos *getResults* e *getallResults* são responsáveis pela obtenção dos resultados providos pelo LORIS.

O método *query* instancia a classe *LORISWSStub*, gerada a partir da WSDL do LORIS, e invoca o método *getLOS* que obtém acesso ao serviço de mediação. As classes *XMLReader* e *XPathHelper* pertencem ao pacote *utilities* e são usadas para a manipulação do resultado retornado pelo LORIS no formato XML.

PUC-Rio - Certificação Digital Nº 0310839/CA

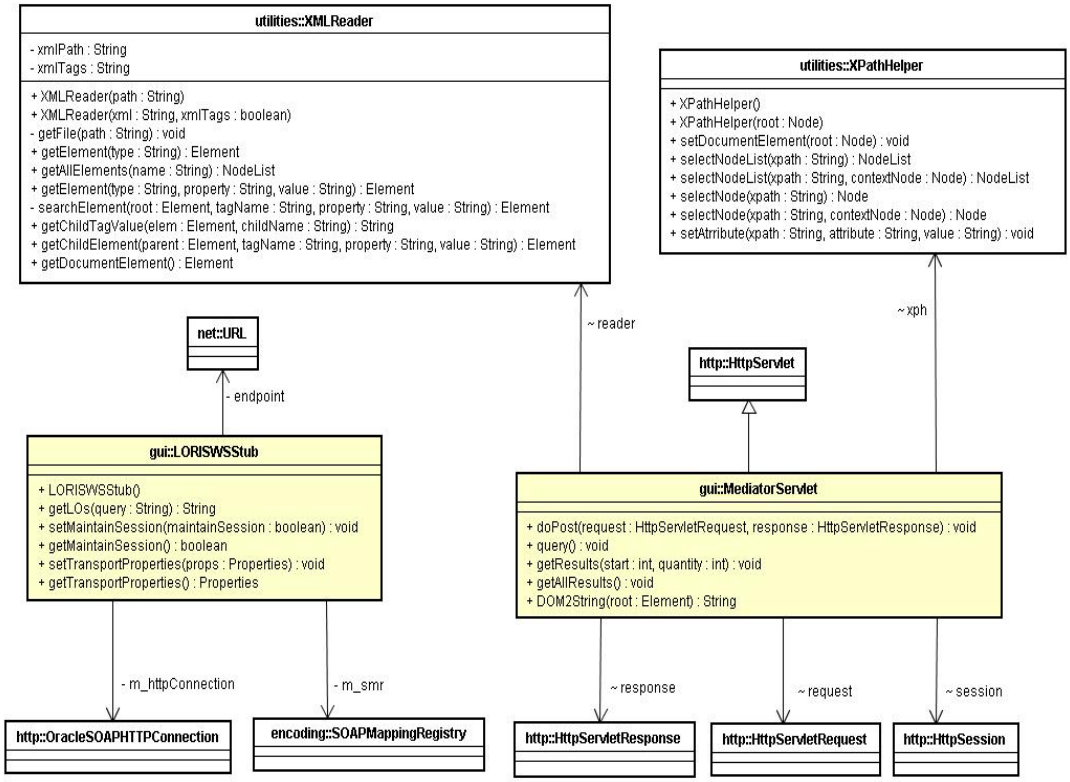

Figura 6.5: Diagrama de classes do subsistema GUI

6.2.1.4. Módulo Utilities

O pacote *utilities* representa um conjunto de classes utilitárias para a manipulação de documentos XML e acesso às informações de configuração do ambiente. A classe *XMLReader* provê acesso a documentos XML; a classe *XPathHelper* permite o acesso a elementos de um documento XML, a partir de uma expressão de caminho (XPath); a classe *DAOConfig* fornece acesso a informações de configuração mantidas numa base de dados; a classe *University* representa um sítio pertencente ao LORIS, contendo as informações necessárias.

PUC-Rio - Certificação Digital Nº 0310839/CA

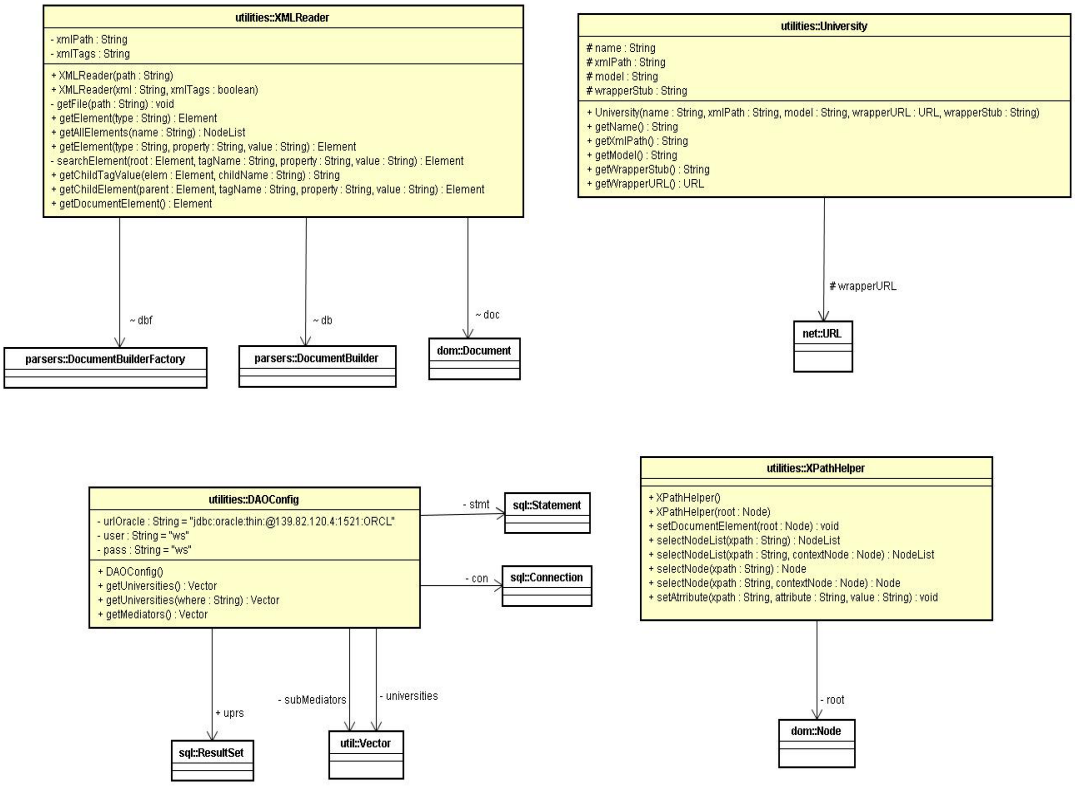

Figura 6.6: Diagrama de classes do pacote *utilities*.

6.3. Implementação do Protótipo

Esta seção apresenta os aspectos tecnológicos utilizados no desenvolvimento do protótipo. Desta forma, a subseção 6.3.1 apresenta a uma visão geral das tecnologias utilizadas. As subseções seguintes trazem uma descrição detalhada dos aspectos técnicos envolvidos na construção do protótipo.

6.3.1. Arquitetura do Protótipo

Com base nos serviços descritos no capítulo 5, foi realizado um levantamento para a busca de ferramentas, preferencialmente de domínio público, e tecnologias padrão de forma a oferecer interoperabilidade entre componentes de *software*.

A figura 6.7 apresenta uma visão geral das tecnologias utilizadas no desenvolvimento dos subsistemas que compõem o LORIS. Foi utilizada a

PUC-Rio - Certificação Digital Nº 0310839/CA

linguagem de programação Java para o desenvolvimento de todos os módulos do sistema, exceto pelo tradutor do sítio B que foi desenvolvido em C#.

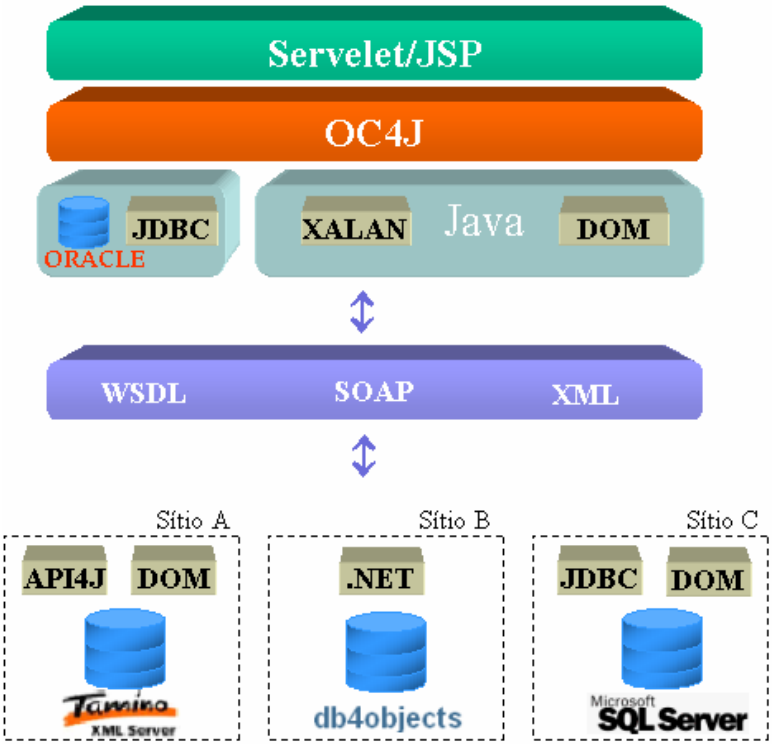

Figura 6.7: Tecnologias utilizadas no desenvolvimento do LORIS

6.3.2. Linguagem de Consulta

Para que seja possível a comunicação entre as fontes e o mediador principal é necessário que haja uma linguagem comum de consulta. No sistema LORIS foi criado um esquema (em XML Schema – Anexo D) para a representação da linguagem de consulta num documento XML.

A figura 6.8 apresenta a representação em XML do seguinte exemplo de consulta.

Quais LOs têm *nível de agregação* "1" ou "2" e possuem "banco de dados" como *palavra-chave*?

PUC-Rio - Certificação Digital Nº 0310839/CA

```
<?xml version="1.0" encoding="iso-8859-1"?>
<query>
        <andComposite>
                <condition>
                        <attribute>general/keyword</attribute>
                        <operator>=</operator>
                        <value>banco de dados</value>
                </condition>
                <orComposite>
                        <condition>
                                <attribute>general/aggregationLevel</attribute>
                                <operator>=</operator>
                                <value>1</value>
                        </condition>
                        <condition>
                                <attribute>general/aggregationLevel</attribute>
                                <operator>=</operator>
                                <value>2</value>
                        </condition>
                </orComposite>
        </andComposite>
</query>
```

Figura 6.8: Documento XML representando a consulta comum

A consulta apresenta-se na forma normal conjuntiva, sendo que o elemento *andComposite* abriga os termos da conjunção. Cada termo representa uma condição (*condition*) ou uma disjunção de condições (*orComposite*). Uma condição é formada pela tripla [atributo(*atribute*), operador(*operator*), valor(*value*)].

PUC-Rio - Certificação Digital Nº 0310839/CA

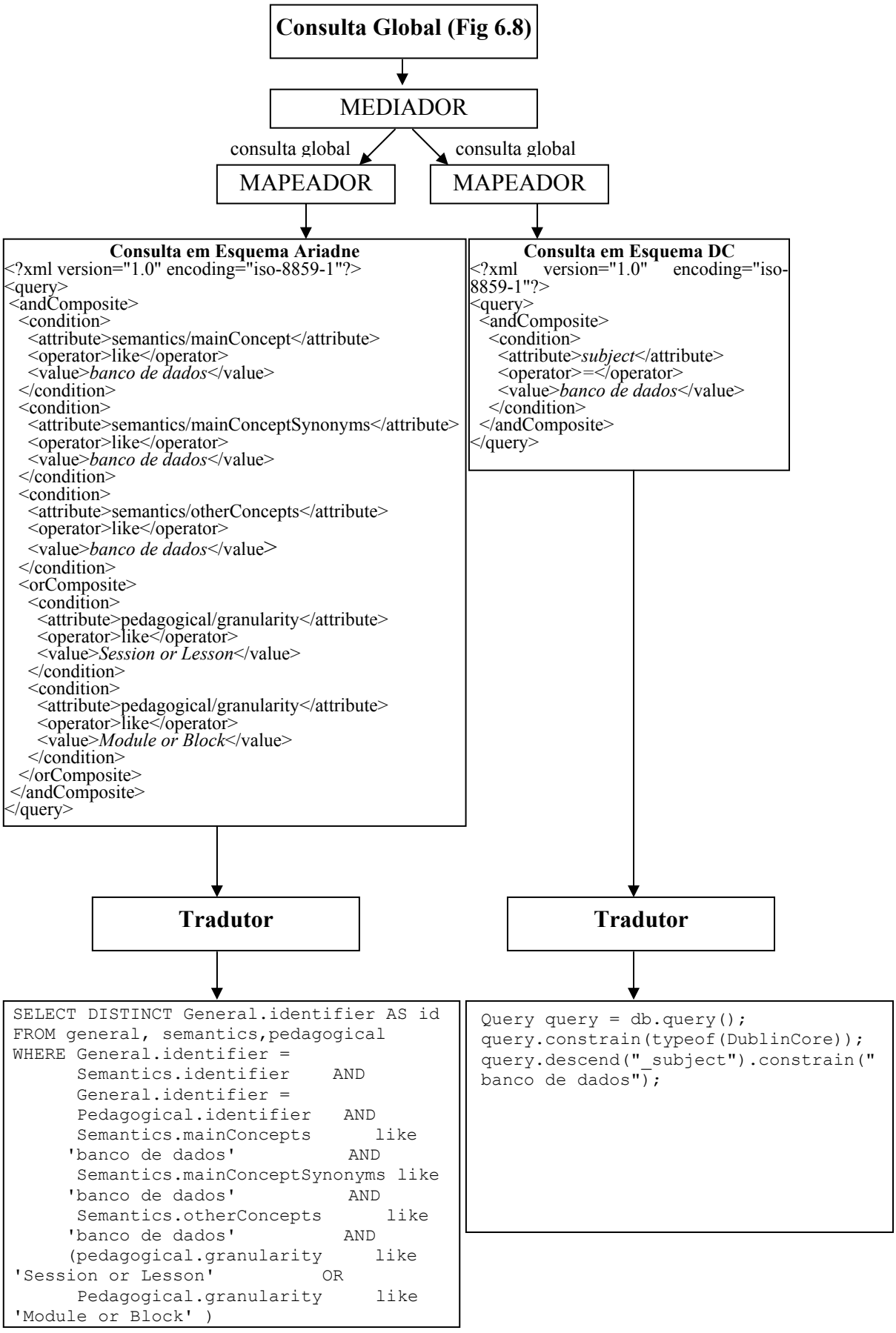

Figura 6.9: Consulta mapeada para as fontes

PUC-Rio - Certificação Digital Nº 0310839/CA

A figura 6.9 apresenta o mapeamento da consulta global (figura 6.8) para as consultas locais dos sítios PGL (exemplificados pelos sítios B e C).

O mapeamento ocorre em duas fases, na primeira etapa a consulta é mapeada de acordo com os elementos do esquema do padrão correspondente. Neste caso, por exemplo, o elemento *general/keyword* (modelo global) é mapeado para os elementos *semantics/mainConcept*, *semantics/mainConceptSynonyms* e *semantics/otherConcepts* no esquema Ariadne e para o elemento *subject* no esquema DC.

A etapa seguinte consiste na tradução da consulta para os modelos dos repositórios de LOs dos sítios B e C, respectivamente, SQL Server (linguagem SQL - Relacional) e db4Objects (linguagem OQL - OO).

6.3.3. Implementação do Subsistema de Mediação

A implementação do subsistema de mediação consiste no desenvolvimento de módulos de *software* capazes de prover serviços de mediação aos repositórios de LOs. A mediação principal constitui um serviço *Web*, cuja WSDL (Anexo A.1) contém a descrição de seus métodos.

A plataforma de desenvolvimento dos serviços *Web* inclui o uso do *Oracle Containers for J2EE* (OC4J) e do ambiente *JDeveloper*.

Para a manipulação de documentos XML foram usadas as APIs Xalan e DOM, responsáveis pelo processamento de expressões de caminho (XPath) e acesso aos elementos, respectivamente.

O módulo de mediação faz acesso a informações de configuração (ex: endereços das fontes, padrão utilizado, etc.) que residem num banco de dados Oracle 9i, através da API JDBC.

PUC-Rio - Certificação Digital Nº 0310839/CA

6.3.4. Implementação do Subsistema de Tradução

Para cada repositório que representa uma fonte de dados, existe um componente tradutor capaz de transformar as consultas derivadas do mediador para uma consulta específica de sua fonte de dados.

A definição dos métodos de cada tradutor é enviada para o mediador no formato WSDL. Desta forma, o mediador precisa implementar uma interface responsável por chamar as funções do serviço *Web* do repositório, gerando um *stub* do mesmo.

Nesta especificação, a comunicação é feita de serviço *Web* para serviço *Web*, através dos protocolos SOAP e HTTP, e os repositórios estão de acordo com a linguagem de consulta comum a ser utilizada, a qual é representada pelo documento XML apresentado na seção 6.3.2.

6.3.4.1. Sítio A

No sítio A, um SGBD XML Nativo – *Tamino XML Server 4.2.1.2*, foi usado para o armazenamento dos LOs, descritos conforme o padrão IEEE LOM. Para acesso ao SGBD foi utilizada a API Tamino API4J. A API DOM e o *parser Xerces* foram usados no processamento dos dados XML.

6.3.4.2. Sítio B

No sítio B, foi utilizada a plataforma de desenvolvimento .NET, tendo C# como linguagem de programação e consulta. Foi utilizado o banco de dados orientado a objetos, *DB4Objects*, como repositório de LOs. O padrão de metadados utilizado por este sítio é o Dublin Core.

6.3.4.3. Sítio C

No sítio C, foi usado o SGBD Relacional SQLServer 2000 como repositório de LOs, os quais estão representados no padrão Ariadne. As APIs JDBC e DOM

PUC-Rio - Certificação Digital Nº 0310839/CA

foram usadas no acesso ao SGBD e na manipulação de documentos XML, respectivamente.

6.3.5. Implementação do Subsistema GUI

A construção da interface utilizou a tecnologia Servlet/Java Server Pages (JSP), residindo no servidor *Web* junto ao subsistema de mediação principal, sendo acessível através do endereço *http://139.82.111.103:8888/LORIS-LORISWS-context-root/*. O *servlet* roda no *Oracle Containers for J2EE* (OC4J).

A comunicação entre o subsistema GUI e o subsistema de mediação ocorre via o protocolo SOAP, uma vez que a mediação é um serviço *Web*.

A interface apresenta-se em duas telas que representa as formas de busca disponível: busca simples e busca avançada. As subseções seguintes apresentam as duas formas de busca.

6.3.5.1. Interface de Busca Simples

A figura 6.7 apresenta a tela para a busca simples, onde os LOs são pesquisados nos sítios da comunidade PGL. A pesquisa simples constitui uma forma de acesso simplificado a atributos relevantes (identificador, título e autor) do modelo comum.

PUC-Rio - Certificação Digital Nº 0310839/CA

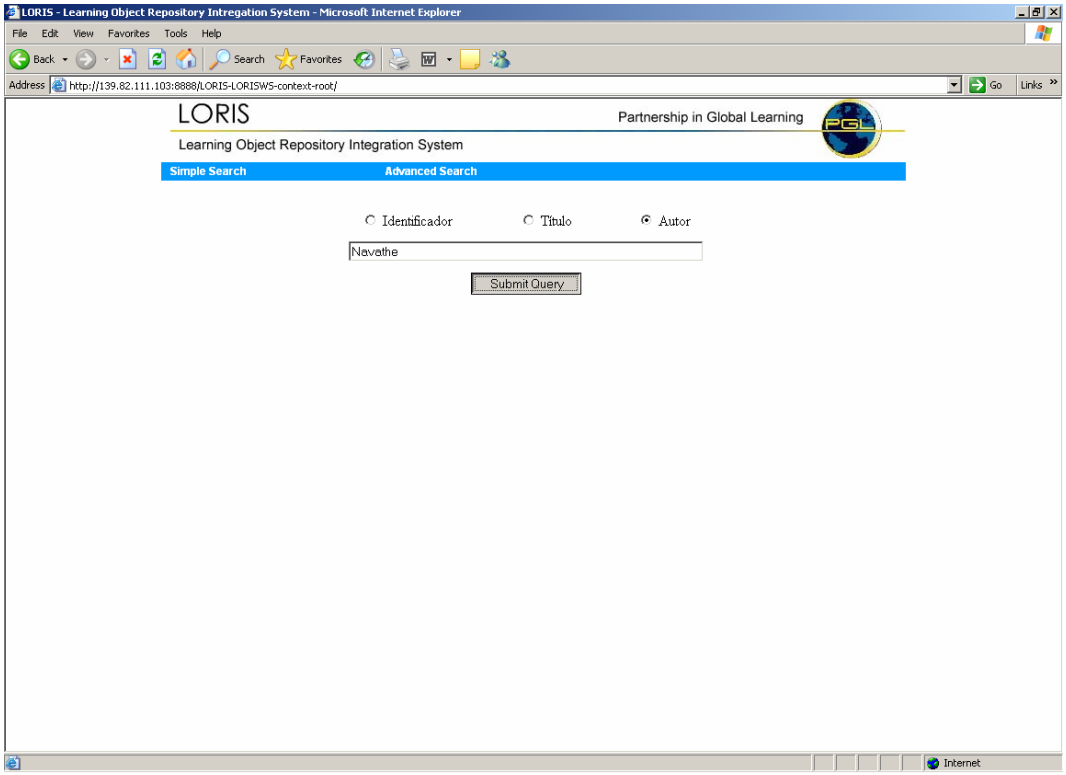

Figura 6.7: Tela da busca simples

6.3.5.2. Interface de Busca Avançada

A figura 6.8 apresenta a tela para a busca avançada de LOs. A pesquisa avançada constitui uma forma de acesso completo aos atributos que compõem o modelo comum. Neste caso, a busca realiza-se através de um conjunto de parâmetros que podem ser atributos mono valorados ou multivalorados. Quando mais de um atributo é requerido, a busca ocorre de forma restritiva, configurando-se como uma conjunção (E) dos atributos especificados na busca. Em se tratando de atributos multivalorados, cada um de seus valores de busca compõe um termo de OUs ou Es como parte da conjunção.

Na busca exemplificada na figura 6.8, a expressão de consulta teria a seguinte condição: (((palavra-chave='sql') **OU** (palavra-chave='consulta')) **E** (idioma='português') **E** (estrutura='Atômica')).

PUC-Rio - Certificação Digital Nº 0310839/CA

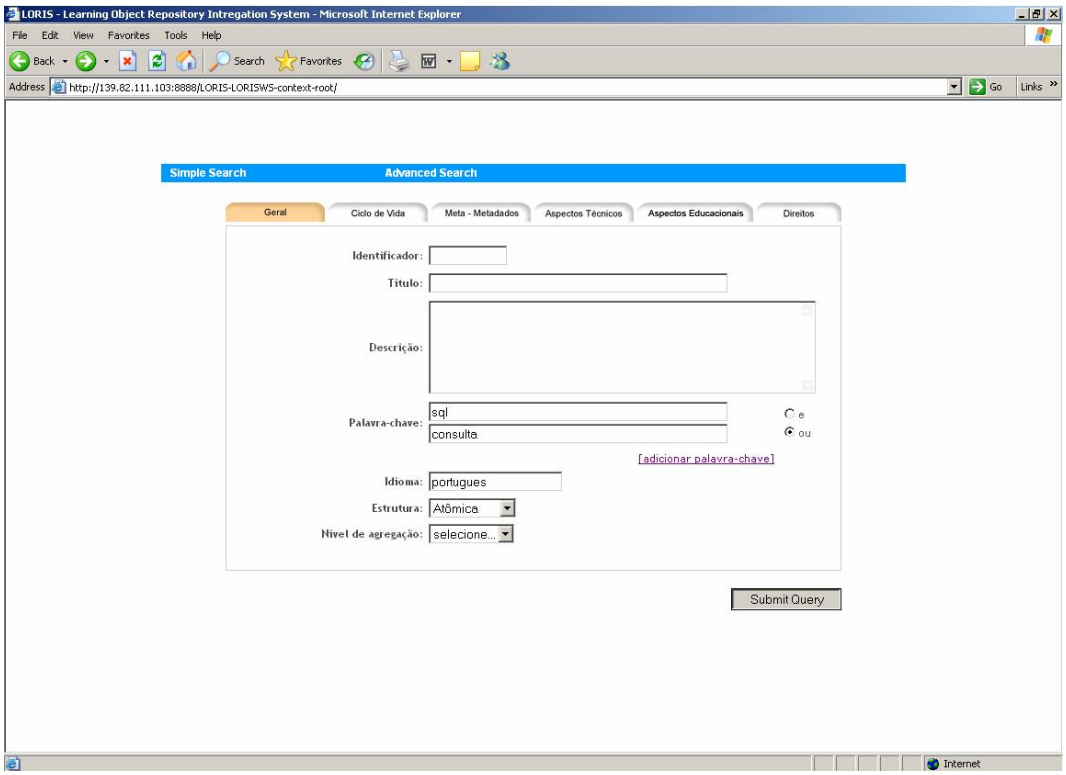

Figura 6.8: Tela de busca avançada

A figura 6.9 exibe a tela contendo os resultados da busca submetida na tela da figura 6.8.

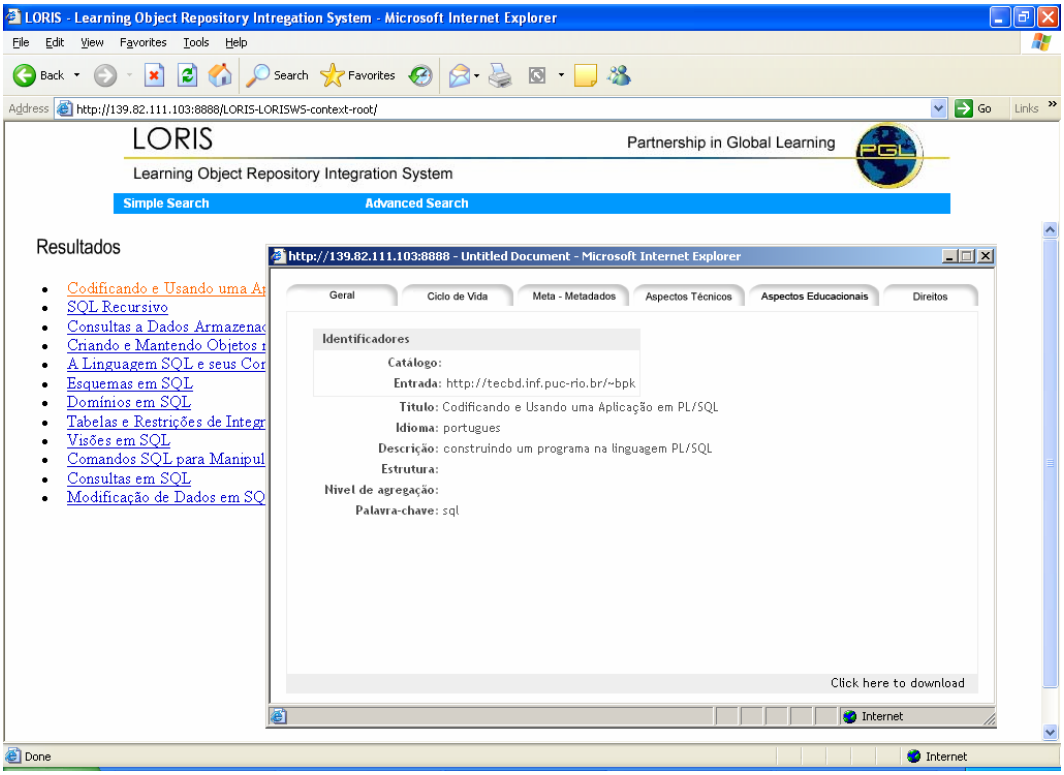

Figura 6.9: Resultados da busca avançada.

PUC-Rio - Certificação Digital Nº 0310839/CA

6.4. Considerações Finais

Este capítulo tratou do desenvolvimento de um protótipo para o LORIS, sendo o projeto PGL aplicado como estudo de caso do sistema. Foram apresentados os diagramas de classes que representam o funcionamento de cada componente da arquitetura, além das tecnologias utilizadas em cada fase. O protótipo desenvolvido mostrou a viabilidade da utilização de serviços *Web* na integração de informações, seguindo a abordagem baseada em mediadores.

PUC-Rio - Certificação Digital Nº 0310839/CA