



CENTRO UNIVERSITÁRIO UNIVATES  
CURSO DE SISTEMAS DE INFORMAÇÃO

**ANÁLISE E IMPLEMENTAÇÃO DE UM SISTEMA INTEGRADO DE  
BUSCA BASEADO NOS PADRÕES DE METADADOS E  
PROTOCOLOS DE INTEROPERABILIDADE UTILIZADOS POR  
CATÁLOGOS ON-LINE DE BIBLIOTECAS E REPOSITÓRIOS  
DIGITAIS**

Willian Valmorbida

Lajeado, novembro de 2011

Willian Valmorbida

**ANÁLISE E IMPLEMENTAÇÃO DE UM SISTEMA INTEGRADO DE  
BUSCA BASEADO NOS PADRÕES DE METADADOS E  
PROTOCOLOS DE INTEROPERABILIDADE UTILIZADOS POR  
CATÁLOGOS ON-LINE DE BIBLIOTECAS E REPOSITÓRIOS  
DIGITAIS**

Monografia apresentada na disciplina de Trabalho de Conclusão de Curso II, do curso de Sistemas de Informação, do Centro Universitário UNIVATES, como parte da exigência para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Ms. Alexandre Stürmer Wolf

Lajeado, novembro de 2011

Willian Valmorbida

**ANÁLISE E IMPLEMENTAÇÃO DE UM SISTEMA INTEGRADO DE  
BUSCA BASEADO NOS PADRÕES DE METADADOS E  
PROTOCOLOS DE INTEROPERABILIDADE UTILIZADOS POR  
CATÁLOGOS ON-LINE DE BIBLIOTECAS E REPOSITÓRIOS  
DIGITAIS**

A banca examinadora abaixo aprova a Monografia apresentada a Graduação em Sistemas de Informação, do Centro Universitário Univates, como parte da exigência para obtenção do grau de Bacharel em Sistemas de Informação:

Prof. Ms. Alexandre Stürmer Wolf – orientador

UNIVATES

Prof. Ms. Luis Antônio Schneiders

UNIVATES

Prof. Ms. Paulo Roberto Mallmann

UNIVATES

Lajeado, 21 de novembro de 2011.

## AGRADECIMENTO

A Deus por cada novo dia que me é entregue.

Aos meus pais, Idoino Valmorbida e Liane Marta Valmorbida, que me encorajaram, compreenderam e não mediram esforços para me auxiliar em mais uma jornada, e que acima de tudo, foram responsáveis por uma formação pessoal sólida e responsável.

Ao professor, colega, orientador e grande amigo, Alexandre Stürmer Wolf, que esteve presente durante todos estes anos, sempre disposto, incentivando e auxiliando na busca de conhecimento e de uma boa formação profissional e pessoal. Agradeço a confiança e dedicação conferida.

As oportunidades de aprendizado, crescimento profissional e pessoal advindas do exercício das minhas atividades profissionais na Univates, proporcionadas e influenciadas por algumas pessoas especiais como as minhas chefes Ana Paula Lisboa Monteiro, Maristela Hilgemann Mendel e Carla Barzotto, professor Alexandre Stürmer Wolf, professor Candido Fonseca da Silva e demais colegas.

Aos verdadeiros amigos que sempre estiveram presentes nos momentos de alegria e de dificuldade.

A todos os colegas de trabalho e de graduação pelos bons momentos e trocas de experiência.

A todos os professores que estiveram disponíveis e dispostos a partilhar seus conhecimentos.

## RESUMO

Este trabalho apresenta a análise e desenvolvimento de um sistema destinado à integração de sistemas de informação utilizados pelo meio acadêmico e científico para a aquisição e distribuição de informações, sistemas estes representados principalmente por catálogos on-line de bibliotecas e repositórios digitais, porém extensível a qualquer outro serviço oferecido na *web* que trabalhe com uma estrutura de metadados e protocolo de interoperabilidade compatível com o sistema proposto. O principal objetivo do sistema desenvolvido é proporcionar uma interface uniforme de busca aos usuários finais, de modo que o sistema consulte dados provenientes de diferentes provedores de dados e direcione os usuários ao local onde determinada informação possa ser encontrada, diminuindo significativamente o tempo gasto pelos usuários com o acesso a vários sistemas para realizar buscas. O sistema desenvolvido baseou-se no conceito de federação de sistemas, fazendo uso, inicialmente, dos protocolos de interoperabilidade *Open Archive Initiative Protocol of Metadata Harvesting* (OAI-PMH) e *Search/Retrieve URL* (SRU), sendo o primeiro desenvolvido sobre o preceito de colheita de metadados e comumente utilizado por repositórios de objetos digitais, e o segundo baseado no conceito de requisições em tempo real aos servidores de dados, sendo comumente empregado em catálogos on-line de bibliotecas.

**Palavras-chaves:** Sistema integrado de busca. Metadados. Protocolos de Interoperabilidade.

## ABSTRACT

This work presents the analysis and development of a system destined for the information systems integration used by academic and scientific environment to acquisition and distribution of information, this systems mainly represented by online libraries catalogs and digital repositories, but extensible to any other offered service on the web where work with a metadata structure and compatible interoperability protocol with the proposed system. The main objective of the developed system is to provide a uniform search interface for end users, the system consult data from different data providers and link users to where the information can be found, reducing significantly the spent time by users with access to multiple systems to perform searches. The system developed was based on the concept of federation of systems, making use initially of Open Archive Initiative Protocol of Metadata Harvesting (OAI-PMH) and Search/Retrieve URL (SRU) interoperability protocols, the first built on the precept to metadata harvesting and commonly used by repositories of digital objects, and the second based on the concept to real-time requests in the data servers and commonly used in online libraries catalogs.

**Keywords:** Integrated system search. Metadata. Interoperability Protocols.

## LISTA DE FIGURAS

<b>Figura 1 - Estrutura de uma federação de serviços.....</b>	<b>23</b>
<b>Figura 2 - Metodologias de interoperabilidade: busca federada e harvesting.....</b>	<b>38</b>
<b>Figura 3 - Fluxo de informações do SRU.....</b>	<b>45</b>
<b>Figura 4 - Organização da mensagem em SOAP.....</b>	<b>49</b>
<b>Figura 5 - Ilustração do MARCXML como intermediário no processo de transformação de padrões e representações de metadados.....</b>	<b>60</b>
<b>Figura 6 - RDF representado por um grafo DLG.....</b>	<b>62</b>
<b>Figura 7 - RDF representado por um grafo DLG com várias declarações, formando um recurso.....</b>	<b>63</b>
<b>Figura 8 - Grafo de representação de um recurso utilizando URIs com o framework RDF.....</b>	<b>63</b>
<b>Figura 9 - Exemplo da estrutura de triplas do RDF.....</b>	<b>64</b>
<b>Figura 10 - Predicado RDF utilizando um elemento do padrão Dublin Core no formato de URI.....</b>	<b>65</b>
<b>Figura 11 - Representação do RDF utilizando apenas URIs.....</b>	<b>65</b>
<b>Figura 12 - Modelo de harvesting de metadados sobre repositórios dotados de Z39.50, SRW e SRU.....</b>	<b>69</b>
<b>Figura 13 - Exemplificação das tecnologias de programação utilizadas para desenvolvimento da interface.....</b>	<b>76</b>
<b>Figura 14 - Arquitetura MVC Model 1.....</b>	<b>78</b>
<b>Figura 15 - Arquitetura MVC Model 2.....</b>	<b>79</b>
<b>Figura 16 - Modelo do banco de dados.....</b>	<b>80</b>
<b>Figura 17 - Padrão das páginas a serem desenvolvidas.....</b>	<b>84</b>

<b>Figura 18 - Layout da interface do sistema.....</b>	<b>86</b>
<b>Figura 19 - Layout da interface do sistema.....</b>	<b>86</b>
<b>Figura 20 - Layout da interface e usabilidade.....</b>	<b>88</b>
<b>Figura 21 - Formulário de pesquisa.....</b>	<b>89</b>
<b>Figura 22 - Pesquisa simples exibindo uma notícia.....</b>	<b>90</b>
<b>Figura 23 - Exibição de uma notícia detalhada.....</b>	<b>90</b>
<b>Figura 24 - Percorrer autor.....</b>	<b>92</b>
<b>Figura 25 - Pesquisa simples.....</b>	<b>93</b>
<b>Figura 26 - Pesquisa simples: conteúdo original do registro aberto em uma janela iframe.....</b>	<b>93</b>
<b>Figura 27 - Pesquisa simples: resultado proveniente da Biblioteca Digital da Univates via protocolo OAI-PMH.....</b>	<b>94</b>
<b>Figura 28 - Pesquisa simples: resultado proveniente da SciELO via protocolo OAI- PMH.....</b>	<b>94</b>
<b>Figura 29 - Pesquisa simples: resultado proveniente da Library of Congress via protocolo SRU.....</b>	<b>95</b>
<b>Figura 30 - Pesquisa simples: estatística de acesso.....</b>	<b>95</b>
<b>Figura 31 - Erro de resposta à requisição SRU da Library of Congress.....</b>	<b>99</b>
<b>Figura 32 - Modelo da proposta baseada em harvesting de metadados para integração dos protocolos.....</b>	<b>101</b>
<b>Figura 33 - Modelo da proposta baseada na implementação do protocolo OAI-PMH sobre um catálogo on-line.....</b>	<b>104</b>
<b>Figura 34 - Modelo do sistema de busca implementado.....</b>	<b>106</b>



## LISTA DE CÓDIGOS

Listagem 1 - Exemplo de registro MARC 21.....	35
Listagem 2 - Exemplos de consulta CQL.....	44
Listagem 3 - Estrutura da URL do protocolo SRU.....	45
Listagem 4 - Exemplo real de requisição SRU, URL solicitada.....	46
Listagem 5 - Exemplo real de requisição SRU, retorno da requisição.....	48
Listagem 6 - Exemplo de uma requisição SRW.....	50
Listagem 7 - Exemplo de resposta de uma requisição SRW.....	51
Listagem 8 - Exemplo de resposta do protocolo OAI-PMH.....	55
Listagem 9 - Exemplo de representação de um registro MARC através de MARCXML. .....	61
Listagem 10 - Exemplo de representação do RDF utilizando XML (RDF/XML).....	64
Listagem 11 - RDF/XML utilizando o namespace Dublin Core.....	66
Listagem 12 - Exemplo de tag file utilizada na implementação do sistema.....	75
Listagem 13 - Cadastro de notícias: exemplo de utilização de tag files dentro de uma página JSP.....	75
Listagem 14 - Exemplo de configuração de um Data Source no arquivo context.xml....	82
Listagem 15 - Exemplo de código para obter um Data Source.....	83
Listagem 16 - Exemplo de código para obter uma conexão de um Data Source.....	84
Listagem 17 - Criação de um pool de threads.....	97
Listagem 18 - Thread que implementa a classe Callable.....	97

## LISTA DE TABELAS

<b>Tabela 1 - Os 15 elementos do esquema Dublin Core.....</b>	<b>31</b>
<b>Tabela 2 - Campos de metadados do esquema MARC 21.....</b>	<b>34</b>
<b>Tabela 3 - Parâmetros de uma requisição com a operação searchRetrieve .....</b>	<b>48</b>

## LISTA DE ABREVIATURAS E SIGLAS

AACR –	Anglo-American Cataloging Rules
ADL –	Advanced Distributed Learning
AGLS –	Australian Government Locator Service
AJAX –	Asynchronous JavaScript and XML
ANSI –	American National Standards Institute
API –	Application Programming Interface
ATG -	Art Technology Group
BDJur –	Biblioteca Digital Jurídica
BDTD –	Biblioteca Digital Brasileira de Teses e Dissertações
BDU –	Biblioteca Digital da Univates
CAN/MARC–	Canadian MACHine-Readable Cataloging
CGI -	Common Gateway Interface
CQL –	Common Query Language
CQL –	Contextual Query Language
CSS –	Cascading Style Sheets
DAO –	Data Access Object
DC –	Dublin Core
DCMI –	Dublin Core Metadata Initiative
DLG –	Directed Labelled Graph
DOM –	Document Object Model
DVPB –	Biblioteca Virtual Del Patrimonio Bibliografico
EAD –	Encoded Archives Description
EL –	Expression Language

GILS –	Government Information Locator Service
HTML–	Hypertext Markup Language
HTTP –	Hiper Text Transfer Protocol
IDE –	Integrated Development Environment
ISO/IEC –	International Organization for Standardization/The International Electrotechnical Commission
ISO –	International Organization for Standardization
ISBN –	International Standard Book Number
Java EE -	Java Enterprise Edition
JDBC –	Java Database Connectivity
JISTEM –	Journal of Information Systems and Technology Management
JNDI –	Java Naming and Directory Interface
JQuery UI –	JQuery User Interface
JSP –	Java Server Pages
JSTL -	JavaServer Pages Standard Tag Library Actions
LC –	Library of Congress
LC MARC –	Library of Congress MACHine-Readable Cataloging
MARC –	MACHine-Readable Cataloging
MARC 21 –	MACHine-Readable Cataloging XXI
MD5 –	Message-Digest algorithm 5
METS –	Metadata Encoding e Transmissão Standard
MODS –	Metadata Object Description Schema
MTD-BR –	Padrão Brasileiro de Metadados de Teses e Dissertações
MVC –	Model-View-Controller
NCSA –	Centro Nacional de Supercomputing Applications
NCSTRL –	Networked Computer Science Technical Report Library
NDLTD –	Networked Digital Library of Theses and Dissertations
NISO –	The National Information Standards Organization
OAI –	Open Archives Initiative
OAI-ORE –	Open Archive Initiative's Object Reuse and Exchange
OAI-PMH –	Open Archive Initiative Protocol of Metadata Harvesting
OCLC –	World's Libraries Connected
OPACs –	Online Public Access Catalogs
PHP -	Hipertext Processor

PLOS –	Public Library of Science
PMC –	PubMed Central
RDF –	Resource Description Framework
RDFS –	Resource Description Framework Schema
REST –	Representational State Transfer
RPN –	Reverse Polish Notation
SciELO –	Scientific Electronic Library Online
SGBD –	Sistema Gerenciador de Banco de Dados
SGML –	Standard Generalized Markup Language
SOAP –	Simple Object Access Protocol
SRU –	Search/Retrieve URL
SRW –	Search/Retrieve Web Services
SSJS –	Server-Side Javascript
TCP/IP –	Transmission Control Protocol/Internet Protocol
UNE-ISO –	Unificacion de Normativas Españolas-International Organization for Standardization
URI –	Uniform Resource Identifier
URL –	Uniform Resource Locator
URN –	Uniform Resource Name
USMARC –	United States MACHine-Readable Cataloging
W3C –	World Wide Web Consortium
XHTML –	eXtensible Hypertext Markup Language
XML –	eXtensible Markup Language
XPath –	XML Path Language

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>15</b>
<b>2 REPOSITÓRIOS DE METADADOS.....</b>	<b>20</b>
<b>2.1 Catálogos on-line de bibliotecas.....</b>	<b>20</b>
<b>2.2 Repositórios Digitais.....</b>	<b>21</b>
<b>2.2.1 Repositórios Institucionais.....</b>	<b>21</b>
<b>2.2.2 Bibliotecas digitais.....</b>	<b>22</b>
<b>2.3 Sistemas Federados.....</b>	<b>23</b>
<b>3 METADADOS.....</b>	<b>25</b>
<b>3.1 Dublin Core.....</b>	<b>29</b>
<b>3.2 MARC.....</b>	<b>33</b>
<b>4 INTEROPERABILIDADE.....</b>	<b>36</b>
<b>4.1 Z39.50.....</b>	<b>40</b>
<b>4.2 SRU e SRW.....</b>	<b>43</b>
<b>4.2.1 CQL.....</b>	<b>43</b>
<b>4.2.2 SRU.....</b>	<b>44</b>
<b>4.2.3 SRW.....</b>	<b>49</b>
<b>4.3 OAI.....</b>	<b>52</b>
<b>4.3.1 OAI-PMH.....</b>	<b>53</b>
<b>4.3.2 OAI-ORE.....</b>	<b>56</b>

<b>5 REPRESENTAÇÃO DE METADADOS.....</b>	<b>57</b>
<b>5.1 Linguagens de marcação.....</b>	<b>57</b>
<b>5.1.1 SGML.....</b>	<b>58</b>
<b>5.1.2 XML.....</b>	<b>58</b>
<b>5.2 MARCXML.....</b>	<b>59</b>
<b>5.3 RDF.....</b>	<b>61</b>
<b>6 IMPLEMENTAÇÃO DO SISTEMA.....</b>	<b>67</b>
<b>6.1 Visão geral.....</b>	<b>67</b>
<b>6.2 Ambiente de desenvolvimento.....</b>	<b>70</b>
<b>6.3 Linguagem de programação.....</b>	<b>70</b>
<b>6.4 Arquitetura.....</b>	<b>77</b>
<b>6.5 Modelo do banco de dados.....</b>	<b>79</b>
<b>6.6 Banco de dados e conexão.....</b>	<b>81</b>
<b>6.7 Layout da interface.....</b>	<b>84</b>
<b>6.8 Sistema e funcionalidades.....</b>	<b>85</b>
<b>6.9 Testes da pesquisa simples.....</b>	<b>98</b>
<b>7 CONCLUSÃO.....</b>	<b>100</b>
<b>7.1 Trabalhos futuros.....</b>	<b>106</b>
<b>REFERÊNCIAS.....</b>	<b>108</b>

## 1 INTRODUÇÃO

O advento das tecnologias da informação imprimiu fortes mudanças na forma pela qual as informações são distribuídas nos dias atuais. Como meio de disseminação de conhecimento, a *web*, tem trazido facilidades que extrapolam a tradicional forma de acesso à informação bibliográfica, de documentos, artigos, jornais, teses, entre outros. Além da mudança de paradigma sofrido pelas formas tradicionais de acesso à informação, as tecnologias da informação passaram a oferecer novos recursos, das mais variadas formas para publicação de informação, como documentos multimídia, *blogs*, fóruns, *wikis*, imagens em tempo real, imagens dinâmicas, entre outros. Desta forma, as informações acadêmicas e científicas ganharam amplo espaço de publicação e disseminação no meio digital, em complemento aos meios tradicionais de publicação, que possuem como características um maior tempo demandado para publicação, custos mais elevados e poder centralizado nas mãos de grandes editoras.

Segundo Marcondes e Sayão (2002), tal avanço tecnológico marcou significativamente a área científica, que partiu para a publicação eletrônica e gratuita de documentos em desgaste aos periódicos controlados por grandes editoras, fato que tem sido enxergado pela comunidade científica como uma forma de aumentar a visibilidade, disseminar o avanço das pesquisas e acelerar o avanço da ciência. Os autores citam que a partir de então surgiram iniciativas que buscam o desenvolvimento dos arquivos abertos como: *Networked Computer Science Technical Report Library* (NCSTRL), *Networked Digital Library of Theses and Dissertations* (NDLTD), *PubMed Central* (PMC), *Public Library of Science* (PLOS), *Open Archives Initiative* (OAI), entre outras.

Kuramoto (2006), cita que tal fato deu início no final da década de 90, frente às



dificuldades enfrentadas na publicação e aquisição de periódicos de cunho científico, surgindo então diversas frentes de apoio ao livre acesso à informação científica. A partir daí o modelo *Open Archives* (arquivos abertos) foi proposto e desta forma surgiram os repositórios digitais, também denominados de *e-prints*, como alternativa mais rápida e barata ao tradicional sistema de comunicação científica, sendo que diversas áreas do conhecimento aderiram à ideia, criando repositórios abertos.

O surgimento das novas tecnologias de informação permitiu a otimização da produção, acesso e disseminação da informação, mudando o conceito tradicional de informação bibliográfica baseada em documentos impressos. O acesso via *Internet*, a novos recursos informacionais, como hipertexto, hipermedia, listas de discussão, conferências virtuais, além da versão eletrônica de documentos impressos, tem se tornado uma realidade cada vez mais presente no dia-a-dia dos profissionais da informação (CRUZ, 2003, p. 47).

O meio digital eliminou as fronteiras geográficas e culturais para a publicação da informação, pois permite acesso global às informações disponibilizadas e de forma imediata, possibilitando desta forma, um *feedback* rápido sobre o conteúdo publicado. Para o meio científico estas características vêm ao encontro dos objetivos das publicações, pois os autores esperam ampla divulgação de suas pesquisas.

Em contrapartida, o meio digital tem disponibilizado enorme quantidade de informação de forma desordenada e sem garantia de qualidade, pois a mesma facilidade de publicação de informações que receberam as publicações científicas está ao alcance de cada indivíduo do planeta.

Castro e Santos (2009), citam que as formas de descrição bibliográficas vêm sofrendo grandes mudanças nos dias de hoje, devido a forte influência e tendência às tecnologias de informação e comunicação. Os autores acrescentam que acompanhada da forte tendência à disponibilização de conteúdo digital na *web* está a falta de padrões na descrição de tais recursos, caracterizando o fato de apenas uma pequena parcela de toda a informação estar concentrada em ambiente adequado e padronizado de modo a possibilitar a interoperabilidade, como é o caso de bibliotecas digitais.

Mesmo ao considerar apenas as informações acomodadas em ambientes adequados, a *web* não estava preparada para tal evolução, fato que ocasionou o desenvolvimento de diversos padrões e protocolos, destinados aos mais diversos fins, mas que possuem objetivos principais em comum, que é proporcionar o armazenamento adequado e a recuperação da

informação. Esta grande diversidade resulta na dificuldade de encontrar meios comuns para promover fácil divulgação e compartilhamento das informações.

Marcondes e Sayão (2002), citam que com o surgimento da *internet*, e com os mecanismos de publicação direta de documentos, o papel das bibliotecas vem mudando no enfoque da intermediação de acesso a documentos. Os autores citam o fenômeno conhecido como explosão informacional, caracterizado pela grande quantidade de informações produzidas e disponibilizadas na *web* por diferentes atividades. Tal fato torna necessário apenas conhecer o *Uniform Resource Locator* (URL) da informação desejada para que ela seja alcançada, porém, a facilidade de acesso conflita com a dificuldade de encontrar informações de qualidade, sendo que mecanismos de busca gerais não se apresentam como soluções ideais para o problema.

De acordo com Goulart e Hetem Júnior (2007), a *web* está cada vez mais presente e indispensável nas atividades de muitos usuários, porém essas atividades encontram-se prejudicadas pela dificuldade apresentada na localização de informação.

“A *Internet* não é como uma biblioteca: não existe ordem, a informação é disponibilizada de maneira caótica; – a informação disponibilizada é sobre uma infinidade de temas, sob os mais diferentes enfoques; isto é, um problema adicional para sua recuperação.” (MARCONDES; SAYÃO, 2002, p. 43).

Os repositórios digitais e todas suas variações surgiram de forma a resolver a problemática citada acima, objetivando organizar e qualificar a informação, porém logo começaram a tornar-se insuficientes perante as novas necessidades de recuperação de informação, pois até o momento era necessário acessar os sistemas individualmente e realizar as mesmas buscas em relação a um determinado conteúdo.

Segundo Marcondes e Sayão (2002), as bibliotecas convencionais passaram por diversas fases de desenvolvimento, sendo que uma destas consistiu na formação de cooperações. Atualmente, são as bibliotecas digitais que buscam constituir cooperação, de forma a alcançarem maior visibilidade para seu conteúdo. Tal fato é caracterizado por iniciativas como a *Open Archives Initiative* (OAI) (Seção 4.3) e por mecanismos de interoperabilidade como o protocolo *Open Archive Initiative Protocol of Metadata Harvesting* (OAI-PMH) (Seção 4.3.1), permitindo assim a criação de portais de busca que integram o conteúdo que está disperso em diferentes repositórios digitais.

Segundo Marcelino (2009), as fontes disponíveis para pesquisa na *internet* são muitas, no entanto, para obter resultados com eficiência e eficácia, existe a necessidade de realizar o procedimento de localização de informação de forma orientada, considerando neste processo a veracidade, confiabilidade, pertinência e atualidade dos dados obtidos. Desta forma é de grande importância filtrar a informação de forma adequada.

Surgiram então portais de busca especializados, sendo alguns focados em determinadas áreas do conhecimento, outros em determinados tipos de documentos, ou mesmo portais que buscam agregar todo conteúdo possível. Estes portais interoperam com sistemas provedores de dados, coletando seus metadados e disponibilizando-os em uma interface de busca, a qual, para os olhos do usuário final acaba parecendo apenas mais um sistema de busca, como tantos outros, porém, este ganha destaque pela riqueza de conteúdo disponível ao comparar com um repositório normal e pela qualidade de seu conteúdo ao ser comparado com mecanismos de busca gerais da *web*.

Toda esta evolução na área das publicações de conteúdo digital caminhou de forma paralela a evolução das bibliotecas tradicionais, as quais informatizaram seus catálogos e alcançaram a formação de redes de cooperação, através, principalmente, do protocolo de interoperabilidade Z39.50 (Seção 4.1). Porém pouca atenção foi dada a forte ligação existente entre as duas linhas, de modo que cada uma delas evoluiu desenvolvendo seus padrões e protocolos sem enxergar muita importância na outra. Entretanto, na atualidade, esta preocupação está tornando-se mais visível e tem-se buscado alternativas para possibilitar a integração de tais sistemas, contudo estas alternativas têm caminhado para iniciativas isoladas, pouco eficientes ou que contam com pouca adesão até o momento.

Tal possibilidade de integração eleva a um novo nível de interoperabilidade um portal de busca, pois permite agregar em um só mecanismo a possibilidade de recuperar informação de diversos repositórios de documentos digitais e de documentos físicos provenientes de catálogos de bibliotecas *on-line*.

A partir de tal problemática, foi proposto o desenvolvimento de um sistema de busca que possibilite a integração dos dados de tais sistemas de informação, utilizando os protocolos e padrões de metadados comuns nestes sistemas. Tendo por objetivo a disponibilização aos usuários, de um sistema que centralize a busca de informações, de modo a reduzir o tempo gasto no desempenho desta atividade, assim como oferecer conteúdo com determinado grau

de qualidade e confiabilidade, promovendo assim a disseminação do conteúdo científico desenvolvido pelas instituições.

Com vistas a apresentar os conceitos e tecnologias que permeiam este conjunto, assim como descrever o projeto proposto e o sistema desenvolvido, o presente trabalho encontra-se dividido conforme segue abaixo:

- No Capítulo 2 são apresentados alguns dos diversos modelos de repositórios de metainformação destinados a documentos acadêmicos e científicos em suporte físico e digital;
- O Capítulo 3 apresenta o conceito de metadados, assim como descreve dois dos padrões mais utilizados em repositórios digitais e catálogos on-line: *Dublin Core* e MARC;
- O Capítulo 4 apresenta o conceito de interoperabilidade, trazendo os protocolos e tecnologias correlatas mais utilizadas em repositórios digitais e catálogos *on-line*;
- No Capítulo 5 é abordado o conceito de representação de metadados através de linguagens de marcação e *frameworks* conceituais;
- No Capítulo 6 são apresentados os aspectos relativos à implementação do sistema de modo geral, tratando de assuntos como metodologia utilizada, linguagem de programação, arquitetura de desenvolvimento, banco de dados, *layout*, funcionalidades implementadas e resultados obtidos;
- Por fim, o Capítulo 7 trás uma breve conclusão em relação ao trabalho desenvolvido, assim como trabalhos futuros a serem desenvolvidos.

## 2 REPOSITÓRIOS DE METADADOS

Nas próximas subseções serão apresentadas as definições, segundo alguns autores da área, acerca de alguns tipos de repositórios de metadados existentes. É importante ressaltar que as fronteiras entre as definições não são bem definidas, gerando muitas vezes controvérsias entre diferentes autores. Tal fato ocorre também pelo fato de alguns repositórios serem construídos agregando funcionalidades de tipos distintos.

### 2.1 Catálogos *on-line* de bibliotecas

De acordo com Guedes (2002), os catálogos *on-line*, também conhecidos como *Online Public Access Catalogs* (OPACs), são disponibilizados visando facilitar a localização de documentos, remotamente ou localmente, por parte dos usuários.

Segundo Mey (1995, *apud* GONÇALVES, 2008), um catálogo *on-line* tem por objetivos disponibilizar aos usuários informações codificadas, organizadas e agrupadas por semelhança, possibilitando através deste, a consulta *on-line* a coleções de documentos de determinada instituição a qualquer usuário.

A finalidade dos catálogos é possibilitar aos usuários a identificação e a localização dos documentos, com maior rapidez e possibilidades, por meio de um registro ordenado dos documentos de uma coleção previamente descrita (representação bibliográfica), contendo os seus pontos de acesso necessários para a sua recuperação (autor, título, série e assunto) (GONÇALVES, 2008, p. 13-14).

Compreendendo um sistema mais complexo, Rowley (2002, *apud* WOLF; MONTEIRO; VALMORBIDA, 2009), determina que um sistema de catálogo *on-line* possa ser denominado também de biblioteca eletrônica quando agrega as funções de

processamento de encomendas, aquisição de materiais, catalogação, controle de circulação, controle de periódicos, informação gerencial, empréstimos entre bibliotecas entre outras.

## **2.2 Repositórios Digitais**

Segundo JISC (2005), repositórios digitais consistem em sistemas que agregam conteúdo digital dos mais variados tipos, proporcionando um conjunto mínimo de características como o de recuperação, controle de acesso, gerência de metadados, entre outros, tendo o intuito de apoiar determinado processo de negócio definido pela instituição mantenedora.

Martins, Rodrigues e Nunes (2011) caracterizam repositórios digitais como coleções de informação digital, constituídas com diferentes propósitos, podendo ser generalistas ou com conteúdo mais específico. Os autores relatam também que há diferentes definições para repositórios digitais devido a variedade de contextos, comunidades, objetivos e práticas ligadas à sua criação e funcionamento.

Nas subseções a seguir serão apresentados alguns dos tipos ou terminologias aplicadas a repositórios digitais.

### **2.2.1 Repositórios Institucionais**

De acordo com Mueller (2006), os repositórios institucionais, são um tipo específico de repositório digital destinado à publicação e manutenção de documentos produzidos em determinada instituição. Como exemplo, podem-se citar os repositórios de universidades. Estes possuem as funções de memória da instituição e com maior destaque, a de aumentar a visibilidade da instituição no que tange a divulgação da produção intelectual e científica da mesma.

### 2.2.2 Bibliotecas digitais

Segundo Castro e Santos (2009), na atualidade, as bibliotecas digitais apresentam-se e atuam na *internet* como um segmento que procura desenvolver, criar métodos e técnicas para a padronização dos recursos da informação.

De acordo com Cunha (1997, *apud* WOLF; MONTEIRO; VALMORBIDA, 2009), biblioteca digital tem como característica marcante a posse de uma coleção de documentos eminentemente digitais, independente se foram criados na forma digital ou digitalizados a partir de documentos impressos.

A DIGITAL LIBRARY FEDERATION (2004), define as bibliotecas digitais como organizações que possuem recursos para selecionar, estruturar, oferecer acesso, distribuir, preservar a integridade e garantir a persistência ao longo do tempo de objetos digitais, tornando-os facilmente e economicamente disponíveis para uso por uma comunidade definida ou conjunto de comunidades.

Uma biblioteca digital – no seu sentido pleno - não é meramente um repositório ou uma coleção de informações em formato digital; também não é somente uma tecnologia ou um conjunto de tecnologias que se pode avaliar isoladamente. Antes disso, é um sistema aberto, de múltiplas interligações e múltiplos subsistemas, envolvendo um ambiente organizacional, profissionais especializados provenientes de diversas áreas, recursos informacionais, usuários claramente definidos, tecnologia de informação, procedimentos, padrões e protocolos e, não menos importante, compromissos de longo prazo. Neste contexto, as normas, padrões, formatos e protocolos cumprem um papel de fundamental importância, já que estabelecem as regras pelas quais os objetos são descritos, identificados e preservados, seus dados são armazenados, e os sistemas aos quais estão inseridos se comunicam (SAYÃO, 2007, p. 19).

O termo biblioteca digital possui várias conotações de acordo com os diversos pontos de vista dos autores da área. Entretanto, o principal contexto é que este tipo de biblioteca implica em novas funções em relação ao armazenamento, organização e recuperação de informação, permitindo acesso remoto e simultâneo, disponibilizando serviços e produtos, possibilitando recuperar documentos completos e bibliográficos, possuindo diversos tipos de registros (música, imagem) e utilizando sistemas inteligentes que ajudam na recuperação da informação (CAMARGO; VIDOTTI, 2006, p. 105)

Apesar de toda facilidade de acesso e recuperação da informação proporcionada pelos modelos de bibliotecas digitais atuais, elas já não estão mais sendo suficientemente capazes de atender as demandas informacionais dos usuários, dificultadas principalmente pela dispersão das informações em diferentes bibliotecas digitais, afirmam Ferreira e Souto (2006). Este fator tem incentivado o surgimento de projetos de federação de bibliotecas digitais, os

quais visam integrar as informações provenientes de diversas fontes em uma única interface de localização.

### 2.3 Sistemas Federados

Ferreira e Souto (2006), definem uma federação como um grupo de instituições independentes, porém, que trabalham juntas formal ou informalmente, sendo que esta cooperação depende de um conjunto de serviços comuns e padronizados.

Marcondes e Sayão (2008) citam que entre as diversas formas de estruturar a arquitetura de uma federação de serviços, a maneira mais comum encontra-se estruturada em três camadas, como pode ser observado na Figura 1:

1. Camada de repositório digital: local autônomo que armazena as informações;
2. Camada de adaptação: camada responsável por tornar transparente as diferenças de modelos de dados e interfaces de consultas dos sistemas. Essa camada é representada principalmente por adaptadores ou mediadores, como por exemplo, os *harvesters*, que são responsáveis pela coleta dos metadados em repositórios heterogêneos e uniformizá-los em uma fonte de dados comum, sob um padrão de metadados comum;
3. Camada de federação: camada que corresponde a uma visão integrada dos dados, oferecendo-os em uma interface única.

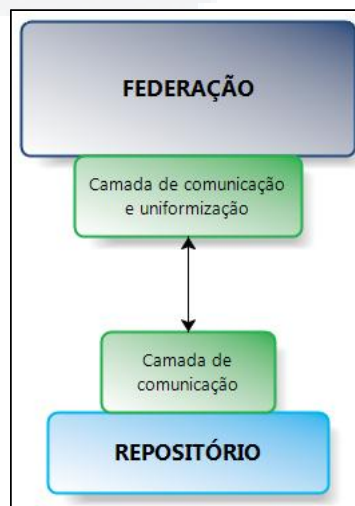


Figura 1 - Estrutura de uma federação de serviços.



Ferreira (2007) define federação a partir de uma de suas mais importantes características, que é a de compartilhamento e interoperabilidade dos conhecimentos gerados e armazenados em locais e plataformas heterogêneas. Cita ainda que o termo “busca federada” possui vários sinônimos, de acordo com diferentes autores: metabusca, busca distribuída, busca paralela, portal de busca, *broadcast search*, *cross-database search*, entre outros.



### 3 METADADOS

De acordo com Souza e Alves (2009), a crescente necessidade de disponibilização e recuperação de informações na *internet* vem em encontro à necessidade da utilização de metadados na organização de recursos digitais. Desta forma, a utilização de padrões de metadados na descrição de recursos possibilita aos sistemas de informação a integração e o compartilhamento desses recursos entre aplicações.

Segundo Wolf, Monteiro e Valmorbidia (2011), com o avanço dos meios digitais, o volume de informações disponível cresceu substancialmente, causando dificuldades para a disseminação de tais, sendo que, pode-se citar o grande número de detentores das informações, sua autonomia e a falta de uma estrutura para acolher esses dados como uma grande problemática. Com isso, o desenvolvimento de padrões para descrever essas informações de forma estruturada, tornou-se imprescindível para as Instituições que desejam disponibilizar os seus dados, ou mesmo utilizá-los de forma eficiente.

Souza et al. (2000 *apud* SOUZA; ALVES, 2009), define metadados como um conjunto de dados codificados e estruturados, que possuem o intuito de descrever as características de recursos de informação. Ele cita elementos como autor, título e assunto como exemplos de metadados, os quais podem ser usados para descrever um livro em um catálogo de uma biblioteca *on-line*, uma página de *internet*, um arquivo de computador, uma base de dados ou qualquer outro recurso digital.

Gil-Leiva (2007), afirma que o papel dos metadados é ordenar e descrever a informação contida em um documento entendido como objeto, de modo que os metadados possam revelar este objeto, tanto na descrição formal, quanto na análise de conteúdo, de forma a propiciar um acesso facilitado ao mesmo. Ele ainda descreve os metadados como

estruturas que possibilitam organizar a informação e conseqüentemente torná-la legível por máquina, de modo a tornar úteis os dados sob diferentes formas, de acordo com as necessidades de cada serviço de informação.

Segundo Duval et al. (2002), existem diversas formas de efetuar a associação de metadados a um objeto digital, sendo que dentre elas destacam-se:

- Metadados incorporados: os metadados estão incorporados na marcação do recurso, sendo criados no momento em que o recurso é criado, dado que o próprio criador do recurso normalmente é criador dos metadados. Esta forma de associação gera maior visibilidade, facilitando a localização por mecanismos de busca;
- Metadados associados: baseia-se em manter os metadados intimamente ligados ao objeto digital, mas não incorporados a ele. Uma das vantagens é que ao alterar os metadados, mantêm-se a estrutura original do objeto. Esta forma de associação exige a gestão dupla de recursos, ou seja, gestão do objeto digital e dos metadados, de forma separada. Outro fator importante é que dependendo da forma que é realizada esta associação, pode ou não tornar o recurso localizável por mecanismos de busca;
- Metadados de terceiros (*third-party*): os metadados são mantidos separados do objeto, em um repositório, normalmente armazenados em um Sistema Gerenciador de Banco de Dados (SGBD), tornando, desta forma, os metadados acessíveis apenas pelo próprio sistema do repositório ou um mecanismo de *harvesting* associado ao sistema, como por exemplo, um mecanismo baseado no protocolo OAI-PMH (Seção 4.3.1).

NISO (2004), define os metadados como informações estruturadas que descrevem, e tornam mais fácil de recuperar, usar ou gerenciar uma fonte de informação. Sendo o termo metadados comumente designado como dados sobre dados ou informação sobre a informação. Podem-se destacar três tipos principais:

- a) Metadados descritivos: são metadados que descrevem um recurso para fins de descoberta e identificação;

- b) Metadados estruturais: são utilizados para identificar a forma que os objetos são compostos;
- c) Metadados administrativos: possuem o objetivo de fornecer informações para o auxílio no gerenciamento de um recurso. Podem-se citar informações como a data de criação do recurso, o tipo do arquivo e outros dados de mesmo cunho. Os metadados administrativos subdividem-se ainda em outros subconjuntos, dos quais pode-se destacar:
- a) Metadados para gerenciamento de direitos: fornecem informações sobre os direitos de propriedade intelectual relacionado a um determinado recurso;
  - b) Metadados para preservação: guardam informações necessárias a gestão da preservação digital de um determinado recurso.

Todos os objetos de informação possuem três características básicas, que podem ser representadas por metadados, são elas: conteúdo, contexto e estrutura. O conteúdo está intimamente relacionado ao conteúdo em si do próprio objeto digital; o contexto refere-se a aspectos relacionados ao objeto (quem, o quê, por que, onde, como); já a estrutura refere-se ao conjunto de associações entre as informações do próprio objeto (GILLILAND-SWETLAND, 2000).

Gilliland-Swetland (2000), afirma que os metadados são de grande importância por promover a acessibilidade de objetos digitais, pois sua existência coerente pode:

- Aumentar de forma significativa a eficiência das pesquisas de mecanismos de recuperação de informação;
- Manter a relação entre objetos digitais, pelo fato de no meio digital existir uma grande facilidade em se desvincular objetos;
- Permitir divulgação ampla e personalizada de dados na *web*;
- Possibilitar a personalização dos metadados de acordo com a comunidade que os acessa ou de acordo com a necessidade descritiva dos objetos;
- Permitir a existência de várias versões de um mesmo objeto;

- Auxiliar no gerenciamento de direitos de acesso;
- Prover mecanismos para preservação digital.

Para Souza e Alves (2009), a organização das informações de forma consistente, ou seja, por meio de estruturas de metadados, é de grande importância quando busca-se a interoperabilidade entre aplicações sustentadas por plataformas de *hardware* e *software* diferentes.

Na mesma linha de pensamento de Souza e Alves (2009), Campos (2007) afirma que a padronização é a base para o entendimento dos metadados, sendo este um compromisso que deve ser realizado respeitando a estrutura do padrão escolhido. Esta padronização é responsável por permitir a reutilização e compartilhamento dos metadados, ou seja, é a base para permitir a interoperabilidade.

Desta forma, existem diversos conjunto de metadados pré-definidos, denominados esquemas de metadados (popularmente conhecidos como padrões de metadados), os quais normalmente são constituídos com um objetivo específico, o de descrever recursos de determinados tipos de objetos digitais. Dentre estes esquemas de metadados podem-se citar alguns mais populares (ARMS et al., 2002):

- *Dublin Core* (DC);
- *MAchine-Readable Cataloging XXI* (MARC 21);
- *Government Information Locator Service* (GILS);
- *Encoded Archives Description* (EAD);
- *Advanced Distributed Learning* (ADL).

NISO (2004), define um esquema de metadados como um conjunto de elementos de metadados concebidos para um determinado propósito, como o de descrever um determinado tipo de recursos de informação, sendo que a definição ou significado dos seus elementos é conhecido como a semântica do esquema e os valores apresentados para elementos de metadados são o conteúdo. Os esquemas de metadados costumam definir os nomes de seus elementos e sua semântica, porém muitas vezes eles determinam também regras para identificação, preenchimento e apresentação dos conteúdos. Também pode haver regras de

sintaxe, de como os elementos e seu conteúdo deve ser codificado. Esquemas de metadados, sem regras de sintaxe pré-definidas são denominados de esquemas de metadados de sintaxe independente, podendo estes ser codificados sobre sintaxes flexíveis como por exemplo, através de *Standard Generalized Markup Language* (SGML) (Seção 5.1.1), *Markup Language Extensible* (XML) (Seção 5.1.2), entre outros.

Nas próximas subseções serão apresentados conceitos e características dos padrões de metadados *Dublin Core* e *MAchine-Readable Cataloging* (MARC), os quais destacam-se no âmbito dos sistemas e protocolos estudados para o desenvolvimento deste trabalho.

### 3.1 Dublin Core

De acordo com NISO (2004), o esquema de metadados *Dublin Core*, ou *Dublin Core Element Set*, surgiu a partir de discussões em um *workshop* ocorrido em 1995 patrocinado pela *The World's Libraries Connected* (OCLC) e do *Centro Nacional de Supercomputing Applications* (NCSA). A *Dublin Core Metadata Initiative* (DCMI) é responsável pela continuação do desenvolvimento das especificações *Dublin Core*. O nome dado a este esquema de metadados advém do fato que o *workshop* que o deu origem foi realizado na cidade de Dublin, Ohio.

Segundo Gil-Leiva (2007), a *Dublin Core Metadata Initiative* é um grupo de trabalho constituído por bibliotecários, pesquisadores de bibliotecas digitais e provedores de informação que iniciou em Dublin (Ohio), em 1995, com o propósito de proporcionar recomendações sobre a descrição de recursos de informação e de seu intercâmbio. A versão 1.1 do conjunto de elementos de metadados *Dublin Core* passou em 2003 a ser uma norma internacional sob o número ISO 15836:2003 (UNE-ISO 15836:2007).

De acordo com NISO (2004), o objetivo original do padrão *Dublin Core* foi definir um conjunto de elementos que poderiam ser usados por autores para descrever seus próprios recursos na *web*, isso devido a grande proliferação de recursos digitais e a impossibilidade de profissionais de biblioteconomia em catalogar todos esses recursos. Desta forma, o objetivo era definir alguns elementos e algumas regras simples para serem utilizadas por pessoas comuns. A definição original do *Dublin Core* possuía 13 elementos, que mais tarde foram

aumentados para 15. São eles: título, criador, assunto, descrição, editor, colaborador, data, tipo, formato, identificação, fonte, idioma, relação, cobertura e direitos.

Os principais objetivos que motivaram o desenvolvimento do padrão *Dublin Core* segundo Weibel (1997), foram:

- Simplicidade de criação e manutenção;
- Semântica universal;
- Conformidade com as normas existentes e emergentes;
- Âmbito Internacional e aplicabilidade;
- Extensibilidade;
- Interoperabilidade entre coleções e sistemas de indexação.

Souza, Vendrusculo e Melo (2000), definem o *Dublin Core* como sendo um esquema de metadados planejado para facilitar a descrição de recursos eletrônicos, a tal ponto que é esperado que os próprios autores, sem conhecimento de catalogação, sejam capazes de utilizá-lo para descrever recursos digitais. Os autores destacam como principais características no padrão, a simplicidade na descrição dos recursos, semântica universal, escopo internacional e extensibilidade.

O padrão *Dublin Core* (DC) é um conjunto de quinze elementos metadados, os quais podem ser definidos como o mais baixo denominador comum para a descrição de recurso de informação, equivalente aos dados de uma ficha catalográfica. Suas características principais são: o entendimento semântico universal dos elementos metadados, o escopo internacional e a extensibilidade visando permitir adaptações às necessidades adicionais de descrição (SOUZA; ALVES, 2009, p. 192).

O principal objetivo que motivou a criação do esquema de metadados *Dublin Core* foi de facilitar a descrição de recursos de informação e promover a visibilidade dos recursos, permitindo sua recuperação por *engines* de busca, afirmam Souza e Alves (2009). Os quinze elementos do padrão tem por objetivo fornecer um conjunto mínimo de metadados para a descrição dos recursos digitais.

Segundo Rocha (2004), a criação do padrão *Dublin Core* levou em consideração as premissas de simplicidade, fácil utilização e escopo genérico. Os seus elementos foram agrupados em três grupos (Tabela 1): Conteúdo, propriedade intelectual e instância. Ao grupo

conteúdo pertencem os elementos: título, assunto, descrição, idioma, fonte, relação e cobertura (espacial ou temporal). Ao grupo de propriedade intelectual pertencem os elementos: criador (responsável intelectual pela criação do recurso), editor (quem tornou o recurso público), colaborador e direitos. Ao terceiro grupo, instância, pertencem os elementos: data, tipo (ex.: página da *web*, artigo, livro), formato (ex.: pdf, world, mp3) e identificador (ex.: URI, *International Standard Book Number* (ISBN)).

De acordo com Weibel (1997), atualmente o *Dublin Core* é formado por elementos e qualificadores, sendo que o esquema pode usar apenas os elementos, porém foi disponibilizada uma série de qualificadores com o objetivo de especificar mais as informações.

Tabela 1 - Os 15 elementos do esquema *Dublin Core*

<b>Conteúdo</b>	<b>Propriedade intelectual</b>	<b>Instanciação</b>
Title	Creator	Date
Subject	Publisher	Format
Description	Contributor	Identifier
Type	Rights	Language
Source		
Relation		
Coverage		

Fonte: Weibel (1997).

De acordo com DUBLIN (1999), cada elemento do *Dublin Core* é definido usando um conjunto de dez atributos da norma ISO/IEC 11179 (padrão ISO 11179) para a descrição dos elementos de dados, sendo eles:

- Nome: A etiqueta atribuída ao elemento de dados;
- Identificador: O identificador único atribuído ao elemento de dados;
- Versão: A versão do elemento de dados;
- Autoridade de registro: A entidade autorizada a registrar o elemento de dados;
- Idioma: A língua em que o elemento de dados é especificado;



- Definição: Uma afirmação que representa claramente o conceito e natureza essencial do elemento de dados;
- Obrigação: Indica se o elemento de dados é sempre obrigatório ou não;
- Tipo de dado: Indica o tipo de dados que pode ser representado no valor do elemento de dados;
- Ocorrência máxima: Indica um limite à repetição do elemento de dados;
- Comentário: Uma nota relativa à aplicação do elemento de dados.

Segundo Dziekaniak e Kirinus (2004), inicialmente o esquema de metadados *Dublin Core* apresentava a problemática de seus elementos serem muito simples e genéricos, de forma a não atender toda a necessidade na descrição de determinados recursos. Desta forma, foram desenvolvidos complementadores aos elementos do esquema, denominados de qualificadores. São de responsabilidade da DCMI a avaliação e inclusão de qualificadores, sendo dada preferência a qualificadores genéricos a qualquer aplicação e com nomenclaturas já utilizadas e mantidas por agências conhecidas dos usuários.

Segundo Dublin (2001), os qualificadores do *Dublin Core* pertencem a duas classes:

- Refinamento do Elemento: os qualificadores desta classe compartilham o significado do elemento ao qual pertence e tem objetivo de especificar mais seu elemento principal;
- Esquema de Codificação: os qualificadores desta classe determinam esquemas que auxiliam na interpretação de um elemento, como por exemplo: vocabulários controlados, notações formais ou regras de representação.

A partir das características do padrão *Dublin Core*, Wolf, Monteiro e Valmorbidia (2009), citam que tal padrão de metadados fornece simplicidade na descrição dos recursos digitais, fácil compreensão e manutenção, possibilidade de extensão do conjunto de metadados além de conferir uma padronização de escopo internacional.

### 3.2 MARC

De acordo com Furrie (2003), a partir da década de 1960 a biblioteca do congresso dos Estados Unidos da América começou a utilizar computadores e a projetar o *Library of Congress MACHine-Readable Cataloging* (LC MARC), que posteriormente evoluiu para MARC 21, vindo atualmente a tornar-se um dos esquemas de metadados mais utilizados por catálogos de bibliotecas no mundo.

Segundo Furrie (2003), o MARC 21 não é um padrão diferente do MARC, é apenas o termo utilizado para denotar o padrão MARC que uniu e eliminou as diferenças tidas entre os denominados *United States MACHine-Readable Cataloging* (USMARC) e *Canadian MACHine-Readable Cataloging* (CAN/MARC), duas versões do padrão MARC. Desta forma, poderão ser utilizados os dois termos (MARC e MARC 21) com o intuito de referenciar o mesmo padrão.

Gil-Leiva (2007), caracteriza o MARC como um conjunto de sinalizadores que utiliza números, letras e símbolos para acrescentá-los aos registros catalográficos, com o objetivo de tornar os registros catalográficos legíveis a computador.

MARC 21 (2000), define o MARC como um meio que pode ser utilizado para descrição de informação bibliográfica referente a livros, materiais de arquivos, manuscritos, arquivos de computador, mapas, músicas, materiais visuais e periódicos.

MARC é um acrônimo de *MACHine-Readable Cataloging*: um conjunto de padrões para identificar, armazenar, e comunicar informações bibliográficas em formato legível por máquina, de forma que diferentes computadores e programas possam reconhecer, processar e estabelecer pontos de acesso dos elementos que compõem a descrição bibliográfica (ALVES; SOUZA, 2007, p. 25).

De acordo com Alves e Souza (2007), o formato MARC 21 utiliza campos fixos de controle, os quais contêm informações codificadas usadas no processamento dos registros, sendo estas informações registradas nas *tags* 0XX. São utilizadas ainda listas auxiliares definidas pelo próprio formato para o preenchimento de alguns dos campos, como por exemplo: códigos para idiomas, país de publicação, área geográfica, entre outras. Já a partir do campo 010, eles ganham subcampos, podendo possuir indicadores, sendo estes números de um dígito que aparecem ocupando duas posições logo após o código do campo.

Cada campo de metadados no MARC é representado por uma *tag* numérica de três dígitos, devido a descrição do mesmo ser muitas vezes extensa. Cada *tag* pode ser dividir em subcampos, os quais são representados por um caractere e ainda cada *tag* pode estar associada a até dois indicadores, os quais são formados por um número de 0 a 9. Os subcampos são utilizados para subdividir com maior especificidade as *tags*, enquanto os indicadores servem para definir alguma informação em relação ao campo. As divisões de campos do MARC ocorrem conforme apresentado na Tabela 2, sendo importante ressaltar que a notação XX refere-se a um conjunto de campos, ou seja, 1XX refere-se a todos os campos pertencentes ao grupo 100: 100, 110, 130, etc (FURRIE, 2003).

Tabela 2 - Campos de metadados do esquema MARC 21

Tag	Descrição
0XX	Informações de controle, números e códigos
1XX	Entrada principal
2XX	Títulos, edição, impressão (em geral, o título, a declaração de responsabilidade, edição e publicação de informação)
3XX	Descrição física, etc.
4XX	Demonstrações, séries
5XX	Notas
6XX	Assunto e entradas adicionadas
7XX	Adicionado em outras entradas de assunto ou série
8XX	Série adicionada entradas (outras formas de autoridade)

Fonte: Furrie (2003).

Existem ainda os campos pertencentes ao grupo 9XX, os quais foram deixados para uso específico de cada utilizador, ou seja, eles podem ser definidos e utilizados conforme necessidade do usuário, assim como os grupos: 09X, 59X, etc (FURRIE, 2003).

Dziekaniak (2008) comenta que de modo geral o MARC possui flexibilidade tal que possibilita a criação de campos para uso local e a catalogação cooperativa através da implementação do protocolo Z39.50, tudo isso devido a sua estrutura enquanto padrão de metadados.

Na Listagem 1 é apresentado um exemplo de registro catalográfico do livro “Aprenda J2EE com EJB, JSP, Servlets, JNDI, JDBC e XML” expresso no padrão MARC 21.

```

001 167540
008 090520s2003 spba ### #001 0#por#d
041 1# $a por $h eng
090 $a 005 .133 $b A654a
240 10 $a Sams teach yourself J2EE in 21 days. $l Português
245 10 $a Aprenda J2EE com EJB, JSP, Servlets, JNDI, JDBC e XML / $c
Martin Bond ... [et al.] ; tradução: João Eduardo Nóbrega Tortello ;
revisão técnica: Álvaro Rodrigues Antunes
260 # $a São Paulo : $b Pearson Education do Brasil, $c 2003 .
300 $a xxxv, 962 p. : $b il. ; $c 24 cm + $e CD ROM
500 $a Inclui CD ROM
650 04 $a Java (Linguagem de programação de computador)
700 1# $a Bond, Martin, $d 1958-

```

Listagem 1 - Exemplo de registro MARC 21.

## 4 INTEROPERABILIDADE

Dziekaniak (2008), associa o conceito de interoperabilidade com a troca e compartilhamento de informações, processo o qual possibilita que bases de dados de sistemas diferentes, constituídos sob diferentes tecnologias possam trocar dados, de forma a permitir a disseminação das informações de forma ágil e eficaz. A interoperabilidade entre sistemas de informação é regida por protocolos, que são um conjunto de padrões convencionados, os quais contém regras que permitem a comunicação entre sistemas.

Sayão (2007), determina a interoperabilidade como sendo uma das coisas mais importantes que deve ser considerada no desenvolvimento de um repositório digital aberto, e afirma que a implementação de tal, exige a aplicação de uma gama de padrões abertos de modo a proporcionar de forma consistente as necessidades de interoperabilidade de um repositório digital de documentos abertos.

Seguindo a ideia de Sayão (2007), Alves e Souza (2007), determinam como essenciais a adoção de padrões, normas e modelos internacionais para a implementação de mecanismos que permitam o interoperabilidade de sistemas de informações.

Em contrapartida, Miller (2000), alega que para obter interoperabilidade não basta apenas assegurar uma forma padronizada de comunicação entre sistemas, sendo necessária uma harmonização mais ampla, no que refere-se a mudanças na forma de como trabalhar com a informação e de relacionar as instituições parceiras. O autor apresenta uma subdivisão da interoperabilidade nos seguintes aspectos:

- Interoperabilidade semântica: consiste basicamente na adoção de padrões de representação da informação, como por exemplo, a utilização de esquemas de metadados e tesouros;
- Interoperabilidade política/humana: trata dos fatores ligados a disponibilidade da informação de forma ampla versus a organização do pessoal técnico e usuários;
- Interoperabilidade intercomunitária: trata do acesso a informação por comunidades distintas e distribuídas;
- Aspectos legais da Interoperabilidade: trata das implicações legais do acesso livre das informações;
- Interoperabilidade internacional: trata da diversidade de padrões e normas, comunicação e linguística no âmbito internacional.

A interoperabilidade exige certo grau de cooperação entre as entidades envolvidas, Arms et al. (2002) cita três instâncias a serem consideradas neste ponto:

- a) Acordos técnicos: acordos que determinam formatos, protocolos, sistemas de segurança, de forma que as mensagens possam ser trocadas entre os sistemas;
- b) Acordos sobre conteúdos: abrangem os dados, metadados e acordos semânticos sobre interpretação das mensagens;
- c) Acordos organizacionais: determinam as regras básicas para acesso, alteração nas coleções e serviços, pagamento, autenticação, entre outros.

Segundo NISO (2004), existem duas abordagens principais de aplicação para o termo interoperabilidade, que são a *cross-system search* e o *harvesting* (colheita de metadados). O protocolo Z39.50 define a implementação de um *cross-system search*, no qual são compartilhadas informações através de requisições instantâneas ao servidor de dados, enquanto a abordagem da *Open Archives Initiative* (OAI) consiste na conversão dos metadados em um conjunto de elementos comuns que são disponibilizados por mecanismos de *harvesting*, sendo desta forma, colhidos e armazenados em bancos de dados, permitindo a busca dos metadados independentemente da comunicação com os repositórios originais.

Marcondes e Sayão (2001) atribuem sinônimos as duas abordagens apresentadas pela NISO (2004) ao citar que existem várias maneiras de promover a interoperabilidade a sistemas heterogêneos, porém estes podem ser agrupados em duas categorias, as quais o usuário acessa apenas uma interface: a busca em diversos servidores, comumente denominada *broadcast search* e a busca em uma base de metadados centralizada via *harvesting*.

Seguindo o mesmo ponto de vista dos autores anteriores, Suleman (2002), cita duas formas principais de promover a interoperabilidade entre sistemas, sendo elas a colheita de metadados e a federação, sendo o termo federação usado como sinônimo para *cross-system search*. É importante ressaltar que alguns autores atribuem o termo federação a promoção da interoperabilidade entre repositórios independentemente do protocolo e padrões utilizados. O autor define a federação como um método no qual os critérios de uma pesquisa são enviados para repositórios distribuídos, sendo seus resultados coletados, unidos e exibidos. Já na colheita de metadados, os metadados são mantidos em uma base local na qual ocorrem as consultas, sendo esta base atualizada pela coleta periódica de metadados em repositórios distribuídos. A Figura 2 exibe a estrutura das duas metodologias de interoperabilidade citadas por Suleman (2002).

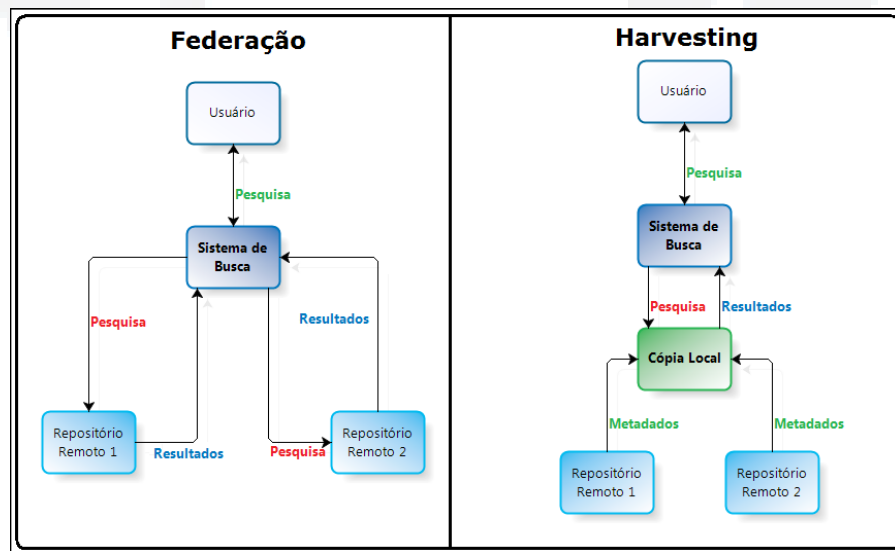


Figura 2 - Metodologias de interoperabilidade: busca federada e *harvesting*.

Fonte: Do autor, adaptado de Suleman (2002, p. 15).

Arms (2000) menciona que o termo biblioteca digital federada descreve um grupo de organizações, trabalhando juntas de maneira formal ou informal, que concordam em assumir uma série de serviços e padrões comuns, buscando a interoperabilidade entre elas. Seus parceiros podem ter diferentes plataformas, desde que implementem serviços compatíveis

com os padrões determinados. Necessitam concordar tanto com os padrões tecnológicos, quanto com as políticas e diretrizes (incluem, portanto normas de direitos autorais, segurança, privacidade e acordos financeiros).

A interoperabilidade promovida por uma federação do tipo *broadcast search* fornece o nível mais alto de interoperabilidade entre sistemas, porém as instituições envolvidas necessitam alcançar alto grau de colaboração dos participantes. Neste tipo de interoperabilidade, as organizações que participam do grupo concordam em manter seus serviços em conformidade com determinadas especificações. Um exemplo de federação de bibliotecas é encontrado com grupos de bibliotecas que compartilham seus catálogos através do protocolo Z39.50 (ARMS et al., 2002).

A *broadcast search* é caracterizada pela consulta distribuída em diversos servidores de dados disponíveis na *web* e sendo as respostas integradas e disponibilizadas ao usuário. Este tipo de interoperabilidade é comumente implementado com o protocolo Z39.50, especialmente por catálogos de bibliotecas, sendo um protocolo que apresenta a vantagem de facilmente poder agregar novos provedores de dados com um esforço mínimo para configuração, porém exigindo a aderência aos padrões utilizados (MARCONDES; SAYÃO, 2001).

De acordo com Arms et al. (2002), a interoperabilidade baseada em *harvesting* (colheita de metadados) proporciona um nível de interoperabilidade menor que a do tipo *broadcast search*, porém ao mesmo tempo exige menos esforço das organizações participantes, e por este motivo vem ganhando espaço. Um exemplo bastante difundido deste tipo de interoperabilidade está na utilização do protocolo OAI-PMH, com o qual as instituições tornam os metadados de seus repositórios disponíveis à coleta por sistemas prestadores de serviços.

Segundo Marcondes e Sayão (2001), a interoperabilidade denominada de *harvesting* consiste na coleta periódica de metadados e seu armazenamento em uma base centralizada, na qual o sistema realiza buscas, sem necessidade de efetuar consultas externas, como no caso da *broadcast search*. Esse tipo de interoperabilidade apresenta como vantagens um melhor desempenho nas buscas e facilidade na inclusão de um novo provedor de dados, porém o processo de sincronização das bases de dados pode ser oneroso.



Arms et al. (2002), apresenta mais uma abordagem de interoperabilidade denominada *gathering*, sendo esta alcançada de forma involuntária pela coleta de informações disponíveis publicamente na *web*, consistindo no nível mais baixo e que demanda de esforço mínimo. Exemplos deste tipo de interoperabilidade são encontrados em *engines* de busca disponíveis na *web*.

Nas próximas seções serão apresentados, de forma breve, a *Open Archives Initiative*, assim como conceitos e características em relação aos protocolos de interoperabilidade mais comumente utilizados no âmbito do compartilhamento de metadados de repositórios digitais e catálogos *on-line*, sendo eles: Z39.50, SRW, SRU, OAI-PMH e OAI-ORE.

#### 4.1 Z39.50

Segundo Rosetto (1997), o protocolo Z39.50 surgiu devido a necessidade da definição de um mecanismo que normalizasse a comunicação entre sistemas de computadores. Desta forma, a *National Information Standards Organization* (NISO) criou um comitê com o objetivo de definir um protocolo de recuperação de informação. Os estudos do comitê criado, tomou como ponto de partida os estudos da *Library of Congress* (LC) efetuados nos anos 70. Abaixo segue um breve histórico:

- 1970-1980 – Criação de um protocolo experimental como parte do *Linked System Project* (LSP) da LC;
- 1979 – A NISO estabelece um comitê para elaboração de um protocolo de recuperação de informação entre sistemas de computadores;
- 1988 – É lançada a primeira versão do protocolo Z39.50;
- 1991 – Ocorre a aprovação de duas normas complementares que acompanham o protocolo Z39.50: a ISO 10162, a qual define serviços e aplicação de pesquisa e recuperação e a ISO 10161-1, que especifica o protocolo para pesquisa e recuperação de informações;
- 1992 – É lançada a segunda versão do protocolo Z39.50, com muitas implementações;

- 1995 – É lançada a terceira versão do protocolo Z39.50, com poucas implementações, porém com ampliação de facilidades e inicia-se estudos para implementação da quarta versão do protocolo.

Após a aprovação da primeira versão do Z39.50 em 1988, Rosetto (1997) afirma que o comitê da NISO que havia sido criado especificamente com o intuito de desenvolver tal protocolo foi extinto. A norma continuou em desenvolvimento e a aprovação das novas minutas da norma é responsabilidade da NISO.

Segundo Rosetto (1997), o protocolo ANSI/NISO Z39.50 é bastante genérico, fato que permite que o mesmo seja implementado para interoperar em diferentes sistemas de computação, sistemas operacionais, equipamentos e sistemas de gerenciamento de dados. O Z39.50 vêm sendo usado principalmente para prover interoperabilidade a sistemas de bases de dados bibliográficos.

NISO (2002), define Z39.50 como um padrão que define as especificações de protocolos, regras e procedimentos para permitir que dois sistemas distintos possam comunicar-se com a finalidade de recuperar informação. Ele permite a utilização de uma única interface com o usuário para realizar a buscas em diversos sistemas simultaneamente, sendo independente de sua arquitetura e tecnologias.

De acordo com Gil-Leiva (2007), o Z39.50 consiste em um padrão internacional regido pela ISO 23950. Esse padrão define um protocolo de comunicação entre computadores que permite a recuperação de informações entre sistemas com plataforma de *hardware* e/ou *software* diferentes, tornando possível desta forma um sistema efetuar consultas em outro sem conhecer sua estrutura.

Segundo NISO (2002), o Z39.50 utiliza os protocolos de comunicação *Transmission Control Protocol/Internet Protocol* (TCP/IP) para alcançar a interoperabilidade entre os sistemas. A comunicação ocorre de forma transparente aos diversos sistemas, devido ao fato do protocolo padronizar a comunicação entre os mesmo, sendo possível desta forma, que sistemas construídos com tecnologias de hardware e software diferentes possam se comunicar, bastando para isso implementar as funcionalidades do protocolo.

O protocolo Z39.50 especifica como padrão para formulação de consultas, o uso da *type-1 query*, um padrão de consulta representado pela notação *Reverse Polish Notation*

(RPN), porém cada implementação do protocolo pode determinar seu suporte em relação a operadores e operandos do padrão (Z39.50, 2011).

NISO (2002), cita que o mecanismo de busca do protocolo Z39.50 permite a solicitação de alguns ou todos os registros que satisfazem os critérios de pesquisa, assim como é possível determinar os elementos a serem selecionados e o formato de retorno dos resultados. Abaixo são citadas algumas das funcionalidades do protocolo Z39.50:

- Classificação dos resultados;
- Eliminação de resultados dos resultados da pesquisa;
- Navegação através de índices como assunto, título, autor, entre outros;
- Controle de acesso através de autenticação e senhas;
- Explain, o qual permite que o cliente requirite informações do servidor a respeito de versão de protocolo, padrões suportados, entre outros, objetivando apoiar o cliente na utilização dos recursos do servidor;
- Serviços estendidos, os quais permitem ao cliente realizar solicitações de operações, tais como, salvar uma busca, reutilização ou execução de uma busca em períodos programados.

O protocolo Z39.50 oferece uma ampla gama de funcionalidades, fato que dificulta sua implementação, de modo que, determinadas implementações não seguem completamente a conformidade com o protocolo de modo a não interoperar corretamente com diversos sistemas. Além da implementação em si do protocolo, seu funcionamento correto ainda depende de que a catalogação aplicada aos registros siga as normas, pois falhas neste sentido também podem resultar em falhas de interoperabilidade entre os sistemas (NISO, 2002).

Segundo Trichkov (2004), as funções essenciais do protocolo Z39.50 relacionam-se com pesquisa e recuperação de informações de bancos de dados armazenados em diferentes sistemas, distribuídos em uma rede de computadores. O protocolo especifica estruturas de dados e regras que permitem troca de informações entre uma máquina cliente e uma máquina servidor para recuperação de informações. A especificação do protocolo não aborda a interação entre usuários e clientes/servidores Z39.50, apenas especifica a padronização

necessária para promover a interoperabilidade entre sistemas de computador. A comunicação descrita no padrão é orientado a conexão e stateful, ou seja, o cliente inicia uma sessão com o servidor e a conexão é mantida até que a associação seja encerrada. O protocolo também não suporta diretamente a difusão de pesquisas para múltiplos servidores, mas um cliente pode abrir sessões com vários servidores Z39.50 simultaneamente. Ainda, não é suportado pelo protocolo a manipulação de conjuntos de resultados, com tratamento de duplicação e apresentação para o usuário.

## 4.2 SRU e SRW

Segundo Taylor e Dickmeiss (2005), em meados do ano 2000 surgiram os protocolos *Search/Retrieve URL* (SRU) e *Search/Retrieve Web Services* (SRW), baseados no protocolo Z39.50, porém estes acompanharam a mudança que vinha ocorrendo na representação de registros catalográficos que trata-se da migração da utilização de arquivos binários para XML (Seção 5.1.2). Apesar de o SRW ter sido considerado mais adequado e mais seguro devido utilizar a tecnologia *Simple Object Access Protocol* (SOAP), o SRU tem alcançado maior aceitação devido à sobrecarga adicional do *XML-parsing* do SRW.

Nas próximas subseções serão apresentadas mais características em relação a estes dois protocolos de interoperabilidade baseados no Z39.50, que são considerados a próxima geração do protocolo Z39.50, assim como sobre a linguagem de consulta *Contextual Query Language* (CQL) (na versão 1.1 chamada de *Common Query Language*), adotada por estes dois protocolos.

### 4.2.1 CQL

A CQL é uma linguagem formal para representar consultas a sistemas de recuperação de informação, tais como índices na *web*, catálogos bibliográficos, entre outros. O objetivo da CQL é possibilitar uma estrutura de consulta legível e intuitiva, mantendo a expressividade de linguagens mais complexas, pois tradicionalmente as linguagens de consulta são poderosas e pouco intuitivas ou simples e de fácil utilização, desta forma no CQL buscou-se acomodar o melhor destes dois conceitos (THE LIBRARY OF CONGRESS, 2008).

Taylor e Dickmeiss (2005), citam que um aspecto importante da estrutura de busca e recuperação dos protocolos SRU e SRW em relação ao Z39.50, é que eles utilizam a linguagem CQL, que é muito mais intuitiva e é expressa em forma textual, enquanto o protocolo Z39.50 utiliza a *tipo-1 query*, a qual é expressa em forma binária. A linguagem CQL permite uma precisão muito maior. Na Listagem 2 são apresentados exemplos de consultas com CQL.

```
title=dinosaur
title=(dinosaur or pterosaur) and author=martill
dc.title=*saur and dc.author=martill
title exact "the complete dinosaur" and date < 2000
name=/phonetic "smith"
fish prox/distance<3/unit=sentence frog
```

Listagem 2 - Exemplos de consulta CQL.

Fonte: Taylor, Aylor e Dickmeiss (2005).

#### 4.2.2 SRU

O SRU é um protocolo desenvolvido para trabalhar no âmbito do protocolo *Hiper Text Transfer Protocol* (HTTP) e possibilita a recuperação de registros no formato MARC, utilizado principalmente por catálogos de bibliotecas, e no formato *Dublin Core*, utilizado largamente para a catalogação de arquivos digitais (MOREIRA; RIBEIRO, 2008).

De acordo com Moreira e Ribeiro (2008), o SRU utiliza o tipo de serviço *web Representational State Transfer full (REST-full)*, no qual os comandos são codificados do cliente para o servidor em uma cadeia de caracteres, na forma de uma URL. Cada valor forma um par com o nome do parâmetro (nome=valor), sendo que os parâmetros são preestabelecidos pelo protocolo. O servidor, por sua vez, recebe os parâmetros, os processa e retorna o resultado no formato XML (Seção 5.1.2). Na Figura 3 é apresentado o fluxo de informações entre o cliente e o servidor utilizando o protocolo SRU.

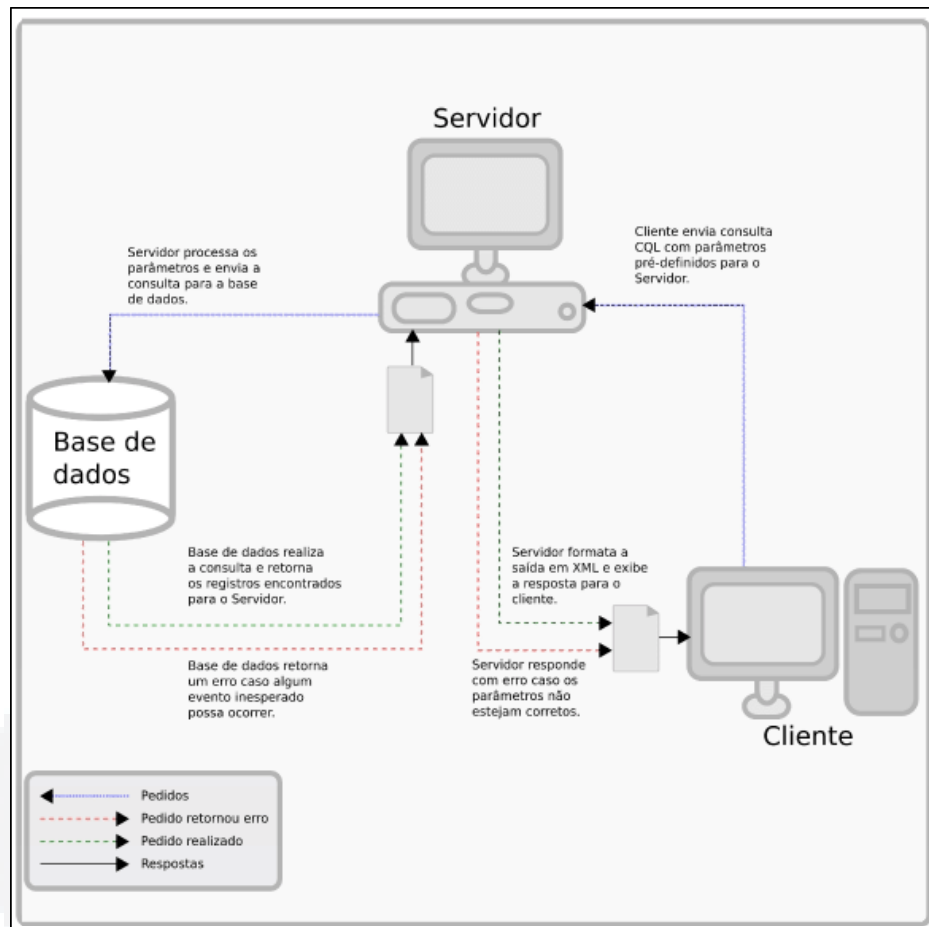


Figura 3 - Fluxo de informações do SRU.

Fonte: Moreira e Ribeiro (2008, p. 171).

Segundo Moreira e Ribeiro (2008), a estrutura da URL do protocolo SRU apresenta-se no formato apresentado na Listagem 3, onde o endereço do servidor compreende a sua URL, já a porta não é um parâmetro obrigatório e pode variar de acordo com a configuração de cada servidor. No caso da versão, o protocolo sempre responde a versão mais alta que seja compatível com a requisição e com o servidor, por exemplo, se a requisição se der na versão 1.1 e o servidor trabalha na versão 2.0 a resposta ocorrerá na versão 1.1 (MOREIRA; RIBEIRO, 2008).

```
Protocolo://EnderecoDoServidor:Porta/NomeDaBaseDeDados?
Versao=NumeroDaVersao&Operacao=TipoDaOperacao&Consulta=Consulta&Opcoes
Retorno=Opcoes&ParametrosRequisicao=Parametros
```

Listagem 3 - Estrutura da URL do protocolo SRU.

Fonte: Moreira e Ribeiro (2008, p. 172).

Moreira e Ribeiro (2008) especificam que o protocolo suporta três tipos diferentes de operação, sendo eles:

- *explain*: retorna informações sobre a situação do servidor, como por exemplo, versão, base de dados, serviços oferecidos, entre outros. Exemplo: <http://z3950.loc.gov:7090/voyager?operation=explain>;
- *scan*: retorna uma listagem dos registros encontrados para determinada consulta. Ele deve ser usado juntamente com a opção *scanClause*, como por exemplo: <http://z3950.loc.gov:7090/voyager?operation=scan&scanClause=ave>;
- *searchRetrieve*: é o tipo mais utilizado, pois é este quem determina a busca do termo solicitado e o retorno dos resultados encontrados. A resposta do servidor pode ocorrer nos formatos MARCXML (Seção 5.2), *Dublin Core* e *Metadata Object Description Schema* (MODS), entre outros, sendo que no caso de não ser definido o retorno ocorre no padrão *Dublin Core*. Exemplo: <http://www.loc.gov/z39voy?operation=searchRetrieve&query=mundo>.

Efetando-se um teste real no servidor da Biblioteca do Congresso dos Estados Unidos, com a URL apresentada na Listagem 4. A URL requisita ao servidor registros que possuam o termo *dinosaur*, limita a quantidade de registros a um, e define como padrão de metadados de retorno o *Dublin Core*.

```
http://www.loc.gov/z39voy?version=1.1&operation=searchRetrieve&
query=dinosaur&maximumRecords=1&recordSchema=dc
```

Listagem 4 - Exemplo real de requisição SRU, URL solicitada.

Na Listagem 5 é apresentada a resposta retornada pelo servidor SRU da Biblioteca do Congresso dos Estados Unidos. A resposta é composta por apenas um registro conforme determinado na solicitação, sendo este registro delimitado pela tag *record*. O conteúdo do registro foi filtrado e formatado de acordo com o padrão *Dublin Core*, conforme especificado no parâmetro *recordSchema* da requisição.

```
<zs:searchRetrieveResponse>
  <zs:version>1.1</zs:version>
  <zs:numberOfRecords>2108</zs:numberOfRecords>
  <zs:records>
    <zs:record>
      <zs:recordSchema>info:srw/schema/1/dc-v1.1</zs:recordSchema>
      <zs:recordPacking>xml</zs:recordPacking>
      <zs:recordData>
```

```

<sru_dc:dc xsi:schemaLocation="info:srw/schema/1/dc-
schema http://www.loc.gov/standards/sru/resources/dc-schema.xsd">
  <title>3-D dinosaur adventure [computer file].</title>
  <creator>Knowledge Adventure, Inc.</creator>
  <type>software, multimedia</type>
  <publisher>Glendale, CA : Knowledge
Adventure,</publisher>
  <date>c1995.</date>
  <language>eng</language>
  <description>
    Employs a dinosaur theme-park setting to introduce
    users to Triassic, Jurassic, and Cretaceous periods. Features
    hypertext dinosaur encyclopedia covering 150 million years of
    paleontology. Includes animated video simulations, three-dimensional
    dinosaur museum, narration, games, activities, and color
    illustrations.
  </description>
  <description>Ages 5 to 10.</description>
  <description>
    System requirements for PC: 486SX/25MHz processor
    or higher; 8MB RAM; Windows 3.1, 3.11, or 95; SVGA 256-color graphics
    adapter; hard drive with 5MB free space; double-speed CD-ROM drive;
    MPC-compatible sound card; mouse.
  </description>
  <description>
    System requirements for Macintosh: 68040 or Power
    PC processor; 8MB RAM; System 7.0 or higher; 256-color graphics
    capability; thirteen-inch color monitor or larger; hard drive with
    4MB free space; double-speed CD-ROM drive.
  </description>
  <description>Title from disc label.</description>
  <description>Ages 5 to 10.</description>
  <description>
    Employs a dinosaur theme-park setting to introduce
    users to Triassic, Jurassic, and Cretaceous periods. Features
    hypertext dinosaur encyclopedia covering 150 million years of
    paleontology. Includes animated video simulations, three-dimensional
    dinosaur museum, narration, games, activities, and color
    illustrations.
  </description>
  <subject>Dinosaurs--Juvenile software.</subject>

```



```

        <subject>Dinosaurs.</subject>
        <identifier>URN:ISBN:1569972133</identifier>
    </sru_dc:dc>
</zs:recordData>
    <zs:recordPosition>1</zs:recordPosition>
</zs:record>
</zs:records>
</zs:searchRetrieveResponse>

```

Listagem 5 - Exemplo real de requisição SRU, retorno da requisição.

Na Tabela 3 são apresentados os parâmetros que podem ou devem ser utilizados em uma requisição de *searchRetrieve*, sendo esta a responsável por solicitar os registros ao servidor. Para cada parâmetro é apresentado sua obrigatoriedade e uma breve descrição.

Tabela 3 - Parâmetros de uma requisição com a operação *searchRetrieve*

Nome do parâmetro	Obrigatório / Opcional	Descrição
<i>version</i>	Obrigatório	Versão do pedido. É uma declaração do cliente que quer a resposta seja menor, ou, de preferência igual a essa versão.
<i>query</i>	Obrigatório	Contém uma consulta expressa em CQL para ser processado pelo servidor.
<i>startRecord</i>	Opcional	A posição na sequência de registros do primeiro registro a ser retornado. A primeira posição na sequência é 1. O valor fornecido deve ser maior que 0. O valor padrão é 1.
<i>maximumRecords</i>	Opcional	Número de registros solicitados ao servidor. O valor deve ser 0 ou maior. O valor padrão, se não fornecido é determinado pelo servidor.
<i>recordPacking</i>	Opcional	Determina como o registro deve ser devolvido. Os valores definidos são <i>string</i> e <i>xml</i> . O padrão é <i>xml</i> .
<i>recordSchema</i>	Opcional	O esquema de metadados requisitado para a resposta do servidor.
<i>recordXPath</i>	Opcional	Uma expressão <i>XML Path Language</i> (XPath), a ser aplicado aos registros antes de serem entregues. A expressão deve ser aplicada relativa ao esquema fornecido no parâmetro <i>recordSchema</i> .
<i>resultSetTTL</i>	Opcional	O número de segundos que o cliente pede que o conjunto de resultados criado deve ser mantido. O servidor pode optar por não cumprir essa solicitação, e pode responder com um número diferente de segundos.

Nome do parâmetro	Obrigatório / Opcional	Descrição
<i>sortKeys</i>	Opcional	Contém uma sequência de chaves de ordenação a ser aplicadas aos resultados.
<i>stylesheet</i>	Opcional	Determina a URL para uma folha de estilos XML.
<i>extraRequestData</i>	Opcional	Fornecer informações adicionais de um perfil específico.
<i>Operation</i>	Obrigatório	A <i>string</i> : <i>searchRetrieve</i> .

Fonte: The Library of Congress (2008).

### 4.2.3 SRW

De acordo com Moreira e Ribeiro (2008), o SRW foi desenvolvido com os mesmos objetivos do SRU, sendo que o SRW diferencia-se do SRU apenas pelo fato de não trabalhar com URL, mas sim com SOAP.

Segundo Moreira e Ribeiro (2008), o SOAP apresenta algumas vantagens, como independência em relação à linguagem de programação, simplicidade, extensibilidade, possibilidade de realizar chamadas sobre o protocolo HTTP e ser estruturado em XML (Seção 5.1.2). Uma mensagem SOAP possui três elementos principais, como pode ser visto na Figura 4: envelope, cabeçalho e corpo. O envelope deve conter o cabeçalho (nem sempre necessário) e o corpo. Dentro do corpo encontra-se o *payload*, que é a informação que se objetiva transportar para o destino final. O corpo também pode transmitir e receber mensagens de erro e alerta através do recurso opcional *fault*, próprio para este fim.

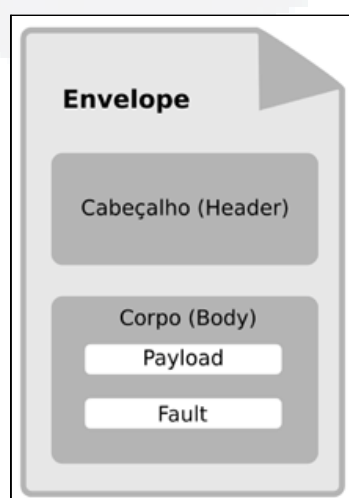


Figura 4 - Organização da mensagem em SOAP.

Fonte: Moreira e Ribeiro (2008, p. 176).

Na Listagem 6 é apresentado o código XML de uma requisição do protocolo SRW, que solicita ao servidor resultados que contenham o termo autor igual a “jones” e título contendo “smith”. É determinado também o retorno de no máximo 10 registros e que o mesmo ocorra no padrão MODS.

```
<SOAP:Envelope
xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP:Body>
    <SRW:searchRetrieveRequest
xmlns:SRW="http://www.loc.gov/zing/srw/">
      <SRW:version>1.1</SRW:version>
      <SRW:query>(dc.author exact "jones" and dc.title >=
"smith")</SRW:query>
      <SRW:startRecord>1</SRW:startRecord>
      <SRW:maximumRecords>10</SRW:maximumRecords>
      <SRW:recordSchema>info:srw/schema/1/mods-v3.0
      </SRW:recordsSchema>
    </SRW:searchRetrieveRequest>
  </SOAP:Body>
</SOAP:Envelope>
```

Listagem 6 - Exemplo de uma requisição SRW.

Fonte: The Library of Congress (2008).

Na Listagem 7 é apresentado um exemplo parcial de retorno de uma requisição a um servidor SRW. Os registros são delimitados no XML pela tag “record”, sendo que nesse caso o retorno ocorreu no padrão de metadados MODS.

```
<SOAP:Envelope
xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP:Body>
    <SRW:searchRetrieveResponse
xmlns:SRW="http://www.loc.gov/zing/srw/"
xmlns:DIAG="http://www.loc.gov/zing/srw/diagnostics/">
      <SRW:version>1.1</SRW:version>
      <SRW:numberOfRecords>2</SRW:numberOfRecords>
      <SRW:resultSetId>8c527d60-c3b4-4cec-alde-
1ff80a5932df</SRW:resultSetId>
      <SRW:resultSetIdleTime>600</SRW:resultSetIdleTime>
      <SRW:records>
        <SRW:record>
```

```

        <SRW:recordSchema>info:srw/schema/1/mods-
v3.0</SRW:recordSchema>
        <SRW:recordPacking>string</SRW:recordPacking>
        <SRW:recordData>
            &lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-
8&quot;?&gt;
            &lt;mods
xmlns:xlink=&quot;http://www.w3.org/TR/xlink&quot;
xmlns:xsi=&quot;http://www.w3.org/2001/XMLSchema-
instance&quot;
xmlns=&quot;http://www.loc.gov/mods/&quot;
xsi:schemaLocation=&quot;
http://www.loc.gov/standards/mods/mods.xsd&quot;&gt;
            &lt;titleInfo&gt;
                &lt;title&gt;Sound and fury :
                the making of the punditocracy /&lt;/title&gt;
            &lt;/titleInfo&gt;
            &lt;name type=&quot;personal&quot;&gt;
                &lt;namePart&gt;Alterman, Eric.&lt;/namePart&gt;
                &lt;role&gt;creator&lt;/role&gt;
            &lt;/name&gt;
            ...
        </SRW:recordData>
        <SRW:recordPosition>1</SRW:recordPosition>
    </SRW:record>
    ...
</SRW:records>
<SRW:diagnostics>
    <DIAG:diagnostic>
        <DIAG:uri>info:srw/diagnostic/1/59</DIAG:uri>
        <DIAG:message>Result set created with valid
partial results available
        </DIAG:message>
    </DIAG:diagnostic>
</SRW:diagnostics>
</SRW:searchRetrieveResponse>
</SOAP:Body>
</SOAP:Envelope>

```

Listagem 7 - Exemplo de resposta de uma requisição SRW.

Fonte: The Library of Congress (2008).

### 4.3 OAI

A *Open Archives Initiative* (OAI) desenvolve e promove padrões de interoperabilidade que visam facilitar a disseminação eficiente de conteúdo, tendo suas raízes no movimento de acesso aberto e repositórios institucionais (OPEN, 2010).

Segundo Lagoze e Sompel (2001), a formação da *Open Archives Initiative* é o resultado da reunião ocorrida em outubro de 1999 em Santa Fé, que tinha por interesse tratar de assuntos relacionados a *E-Prints*. A iniciativa começou seus trabalhos com o objetivo de facilitar a formação de federações. A partir de então, a iniciativa desenvolveu diversos padrões e protocolos com tais objetivos, sendo os protocolos OAI-PMH e *Open Archive Initiative's Object Reuse and Exchange* (OAI-ORE) largamente utilizados nos dias de hoje.

Pavani (2007), descreve que a *Open Archives Initiative* foi criada em uma reunião da comunidade ligada à produção de conteúdos digitais e às bibliotecas digitais, motivada pela necessidade de localizar e identificar conteúdos gerados e armazenados em computadores ligados à *internet* no mundo inteiro. O protocolo criado a partir desta cooperação permite a coleta automática de metadados de forma a concentrá-los em catálogos coletivos, onde podem ser recuperados e conduzindo ao seu conteúdo completo.

O protocolo OAI-PMH oferece um grande diferencial, em relação a buscas baseadas em *web crawling*, pois o protocolo fornece dados estruturados, dando oportunidade para a criação de serviços mais ricos de pesquisa, facilitando também a análise e mineração de dados (HARVESTING, 2011).

Cerca de 75% de repositórios em todo o mundo (aproximadamente 85% no Reino Unido) fornecem uma interface que usa o protocolo OAI-PMH. Tais repositórios são designados *OAI-compliant*, disponibilizando seus metadados abertamente para serviços de colheita, geralmente retornando juntamente com os metadados uma URI para o conteúdo completo (HARVESTING, 2011).

Nas próximas subseções serão apresentados maiores detalhes a respeito dos dois principais protocolos surgidos a partir da iniciativa OAI.

### 4.3.1 OAI-PMH

De acordo com Marcondes e Sayão (2001), a principal motivação que levou ao surgimento do protocolo OAI-PMH foram as dificuldades encontradas para criar grandes federações. Desta forma, o OAI-PMH, foi desenvolvido com o intuito de ser menos oneroso que os protocolos já existentes, não necessitando do enquadramento em complexos acordos e permitindo um compartilhamento básico dos serviços oferecidos.

Segundo Wolf e Valmorbidia (2011), um dos principais protocolos desenvolvidos pela iniciativa OAI é o OAI-PMH, o qual possibilita aos participantes da federação compartilhar seus metadados, sendo que seu conceito subdivide os participantes em dois grupos distintos: os *Data Providers* e os *Service Providers*. Os *Data Providers* são repositórios que armazenam os recursos digitais e implementam o protocolo OAI-PMH como forma de expor seus metadados, já os *Service Providers* utilizam esse protocolo para coletar os metadados, possibilitando oferecer um serviço agregado sobre estes.

De acordo com Marcondes e Sayão (2002), o funcionamento do protocolo OAI-PMH determina que uma solicitação seja respondida ao provedor de serviços com o envio da resposta codificada em XML (Seção 5.1.2), em um formato estruturado de metadados. Essa troca de mensagens é efetuada entre os dois provedores de forma unidirecional, sendo que o provedor de serviços tem a responsabilidade de efetuar a solicitação e o provedor de dados responde com seus metadados. O protocolo OAI-PMH estabelece ainda o *Dublin Core* como seu conjunto mínimo de metadados a ser suportado pelos provedores de dados em resposta a uma solicitação de um provedor de serviços, porém segundo critérios de cada instituição, os provedores podem oferecer outros formatos de metadados mais amplos e complexos, como por exemplo, o MARC 21.

A chave do protocolo OAI-PMH para a coleta de metadados está nos metadados administrativos dos objetos armazenados no provedor de dados, citam Marcondes e Sayão (2002). Cada objeto do provedor de dados possui registros temporais, denominados *datestamps*, os quais determinam a data de criação e última alteração do objeto. Isso permite a coleta seletiva de metadados, de forma a permitir a sincronização dos registros entre um provedor de dados e um provedor de serviços. Além dos *timestamps*, cada objeto do servidor de dados possui um identificador único, normalmente formado pelo identificador do provedor de dados mais um identificador do objeto.

O protocolo prevê ao todo seis “verbos” que um provedor de serviços pode enviar a um provedor de dados para coletar metadados de documentos nele armazenados, são eles (MARCONDES; SAYÃO, 2002):

- *Identify*: obtém dados administrativos sobre o provedor de dados, sua política de publicação de documentos, seu escopo, etc;
- *ListSets*: lista os agrupamentos sob as quais os documentos são organizados no provedor de dados;
- *ListMetadataFormats*: lista os formatos de metadados por meio dos quais os metadados dos documentos armazenados no provedor de dados podem ser representados;
- *ListIdentifiers*: lista os identificadores dos registros armazenados no provedor de dados. Pode-se opcionalmente limitar estes registros a partir de um período temporal ou um *set* (conjunto/coleção de registro);
- *ListRecords*: lista os metadados dos registros armazenados no provedor de dados segundo um formato de metadados, especificando todos que pertencem a um *set* ou todos entre um período temporal;
- *GetRecords*: obtém os metadados de um registro armazenado segundo um formato de metadados, dado um identificador de registro.

A Listagem 8 apresenta o resultado de uma requisição ao protocolo OAI-PMH da Biblioteca Digital da Univates (BDU), utilizando os parâmetros *verb=ListRecords* e *metadataFormat=oai\_dc*.

```
<OAI-PMH xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2010-10-22T22:10:28Z</responseDate>
  <request metadataPrefix="oai_dc"
verb="ListRecords">http://www.univates.br/bdu_oai/request</request>
  <ListRecords>
    <record>
      <header>
        <identifier>oai:http://www.univates.br/bdu:369146285/67</
identifier>
```

```

        <timestamp>2009-04-16T20:03:50Z</timestamp>
        <setSpec>hdl_369146285_66</setSpec>
    </header>
    <metadata>
        <oai_dc:dc
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
            <dc:creator>WOLF, Alexandre Stürmer</dc:creator>
            <dc:creator>VALMORBIDA, Willian</dc:creator>
            <dc:creator>MONTEIRO, Ana Paula Lisboa</dc:creator>
            <dc:date>2009-03-31T13:54:52Z</dc:date>
            <dc:date>2009-03-31T13:54:52Z</dc:date>
            <dc:date>2009-03-31T13:54:52Z</dc:date>
            <dc:identifier>http://www.univates.br/bdu/handle/36914
6285/67</dc:identifier>
            <dc:description>This work seeks to present the
solution adopted by the Centro Universitário UNIVATES to
...</dc:description>
            <dc:language>pt_BR</dc:language>
            <dc:subject>Biblioteca Digital</dc:subject>
            <dc:subject>DSpace</dc:subject>
            <dc:subject>Biblioteca Univates</dc:subject>
            <dc:title>Biblioteca Digital da UNIVATES utilizando a
ferramenta DSpace</dc:title>
            <dc:type>Presentation</dc:type>
            <dc:type>Article</dc:type>
        </oai_dc:dc>
    </metadata>
</record>
...
</ListRecords>
</OAI-PMH>

```

Listagem 8 - Exemplo de resposta do protocolo OAI-PMH.

De acordo com Wolf, Monteiro e Valmorbida (2009), bibliotecas digitais, repositórios institucionais, repositórios digitais, entre outros, são pontos de concentração de informação, tendo como objetivos a disponibilização e acessibilidade da informação, desta forma o protocolo OAI-PMH pode ser utilizado para promover o intercâmbio de metadados dos objetos digitais destes repositórios, de forma a possibilitar uma maior visibilidade e disseminação da informação.



### 4.3.2 OAI-ORE

Segundo Maslov (2010), o intercâmbio de dados entre repositórios é um componente crítico para iniciativas de cooperação entre repositórios. Desta forma, o protocolo OAI-ORE, busca definir as normas e recomendações para descrever e permitir o intercâmbio de recursos digitais. Usado em conjunto com o protocolo OAI-PMH, é possível a replicação completa de conteúdo entre locais remotos e de forma automática. OAI-ORE descreve conjuntos de recursos *web* de forma padronizada e concisa. De acordo com o OAI-ORE, um recurso inclui qualquer objeto identificado por um URI e acessível através de HTTP. O OAI-ORE é projetado especificamente para descrever a localização e as relações entre esses recursos.

Segundo Lagoze et al. (2010), o OAI-ORE define as normas relativas à descrição e intercâmbio de agregações de recursos *web*. Assim como no mundo real no qual temos o costume de agregar coisas, como por exemplo, seleções de músicas, álbuns de fotografias, entre outros, na *web* tendemos ao mesmo costume, pois nela acumulamos URLs, fotos em sites de relacionamento, entre outros. Na *web* utilizamos normalmente a URI de uma página para identificar uma agregação inteira, porém esta URI identifica apenas a página e não a união de páginas ou recursos que compõe o recurso como um todo. Desta forma o OAI-ORE tem por objetivo fornecer uma forma de descrever e identificar recursos de forma agregada.

O fato das agregações não estarem bem definidas na *web*, limita o seu uso, especialmente quando tratam-se de processos automatizados, cita Lagoze et al. (2010). Um exemplo bastante claro é, por exemplo, um usuário querer imprimir várias páginas *web* ao mesmo tempo e as mesmas estarem agrupadas em *links*, o usuário terá que acessar cada página manualmente ou utilizar alguma ferramenta que possua esta funcionalidade; programas que transferem múltiplas páginas entre si devem se basear na *Application Programming Interface* (API) de determinado sistema; mecanismos de busca precisam utilizar heurísticas para agrupar páginas em documentos lógicos de forma aos resultados terem granularidade adequada.

## 5 REPRESENTAÇÃO DE METADADOS

Metadados podem ser representados de diversas formas, por meio de diferentes tecnologias, desta forma, pode-se objetivar apresentá-los para pessoas, devendo assim receber formatação adequada e legível para tais. Metadados são comumente representados através de linguagens de marcação, quando objetiva-se alcançar a interoperabilidade entre sistemas. Nas próximas subseções serão apresentados conceitos, tipos de linguagens de marcação e *frameworks* utilizados para representação de metadados.

### 5.1 Linguagens de marcação

De acordo com Almeida (2002), linguagem de marcação consiste em um conjunto de convenções utilizadas para codificar textos, sendo estas convenções responsáveis por especificar as marcas, dar significado e fazer a diferenciação entre marcas e texto.

Segundo Almeida (2002), tem-se buscado formas de aproveitar melhor a grande quantidade de informações disponíveis na *internet*, sendo que algumas propostas contemplam a utilização de linguagens de marcação como o XML em conjunto com padrões de metadados objetivando conferir maior semântica à *internet*.

De acordo com Bax (2001), a portabilidade de documentos, ou seja, a independência do documento em relação a *software*, *hardware* e sistema operacional, é propiciada pela utilização de padrões de marcação internacionais e abertos, como SGML, *Hypertext Markup Language* (HTML), *eXtensible Hypertext Markup Language* (XHTML), XML, entre outros.

A utilização de padrões, como os citados, permite que os documentos sejam interpretados por qualquer aplicação e ambiente que conheça o padrão usado na criação do documento.

### 5.1.1 SGML

De acordo com Almeida (2002), o SGML é uma linguagem de marcação definido pela ISO 8879, sendo um padrão de cunho internacional, não proprietário e de código aberto, utilizado largamente para a interoperabilidade de sistemas e descrição dos mais diversos tipos de documentos. Um dos principais objetivos o SGML é permitir o transporte de dados de um ambiente de *hardware* e *software* para outro sem perda de informações.

### 5.1.2 XML

Almeida (2002) cita que o XML, como uma linguagem de marcação que possui uma expressiva diferença em relação ao SGML. O XML possibilita ao próprio autor do documento definir as marcações. Tal característica confere ao XML maior flexibilidade na recuperação e disseminação de informações na *web*.

Segundo Pereira, Ribeiro Júnior e Neves (2005), a semântica do XML permita a descrição da estrutura e do conteúdo de recursos digitais e não a formatação, como é o caso do HTML. Tal fato facilita e propicia a difusão de informação na *web*. Os autores citam ainda como característica de grande importância no XML, a possibilidade de utilização de *namespaces*, fator que permite a integração de tal com outros padrões como por exemplo, o *Dublin Core*.

O XML é uma arquitetura que não possui elementos e marcas predefinidas. Não especifica como os autores vão utilizar metadados, sendo que existe total liberdade para utilizar qualquer método disponível, desde simples atributos, até a implementação de padrões mais complexos (ALMEIDA, 2002, p. 6).

Segundo Almeida (2002), o XML foi idealizado por Jon Bosak, engenheiro da *Sun Microsystems* que o derivou do SGML. Jon era usuário da SGML e apresentou ao W3C sua ideia de explorar o SGML em aplicações voltadas para *internet*. Daí nasceu o XML, em 1996, sendo que inicialmente era apenas uma simplificação do SGML, e, em fevereiro de 1998, o XML tornou-se uma especificação formal reconhecida pelo *World Wide Web Consortium* (W3C).

A linguagem XML permite associar atributos as marcações. O termo atributo é utilizado no contexto do XML para especificar propriedades ou características do elemento, sendo eles definidos como pares (nome, valor) (ALMEIDA, 2002).

De acordo com o W3 Consortium, entre os objetivos estabelecidos na especificação da linguagem XML, estão as seguintes características: ser diretamente utilizável na *Internet*; ser legível por humanos; possibilitar um meio independente para publicação eletrônica; permitir a definição de protocolos para troca de dados pelas empresas (independentemente da plataforma de *hardware* e *software*); facilitar às pessoas o processamento de dados pelo uso de softwares de baixo custo; facilitar a utilização de metadados que auxiliam na busca de informações; aproximar “produtores” e “consumidores” de informação (ALMEIDA, 2002, p. 7).

## 5.2 MARCXML

O MARCXML foi criado com vistas a tornar o padrão MARC mais flexível e extensível, de forma a moldar os dados às necessidades de cada usuário. O MARCXML é basicamente um esquema XML contendo dados do MARC e mantendo sua semântica. Este esquema de representação é importante quando da necessidade de transformação de um conjunto de metadados MARC em outros esquemas de metadados como, por exemplo, o *Dublin Core*, ou quando da necessidade de submeter os dados por determinados processos, como por exemplo, uma validação (THE LIBRARY OF CONGRESS, 2010).

No MARCXML todos os campos de controle, incluindo o líder são tratados como uma sequência de dados, enquanto os campos são tratados como elementos, com a sua marca e indicadores como atributos. Já os subcampos são tratados como subelementos, com o código de subcampo como um atributo (THE LIBRARY OF CONGRESS, 2010).

A Figura 5 ilustra o conceito de utilização do MARCXML como um padrão intermediário no processo de exportação e importação do padrão MARC 21 para outros padrões e representações de metadados e vice-versa.

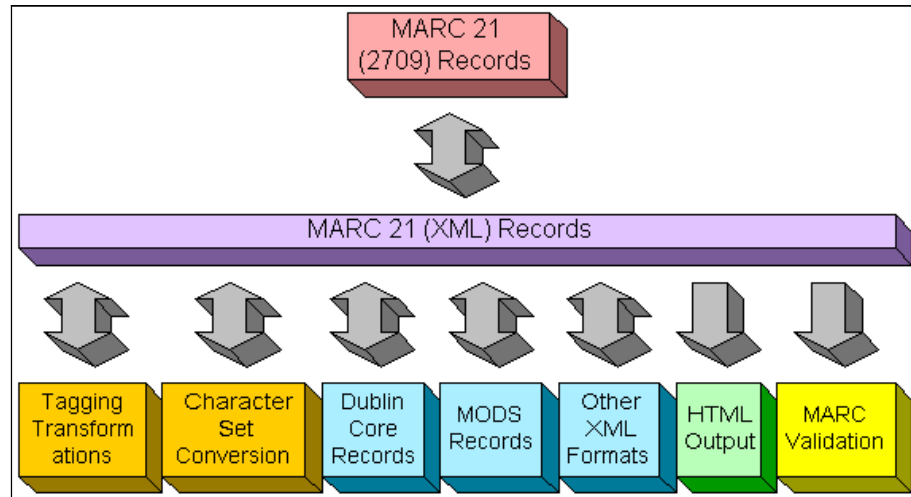


Figura 5 - Ilustração do MARCXML como intermediário no processo de transformação de padrões e representações de metadados.

Fonte: The Library of Congress (2010).

A The Library Of Congress (2010), apresenta como possíveis utilizações para o MARCXML:

- Representar um registro completo MARC em XML;
- Como extensão do esquema *Metadata Encoding and Transmission Standard* (METS);
- Representar metadados para *harvesting* via o protocolo OAI-PMH;
- Para a descrição de recursos original em sintaxe XML;
- Para o empacotamento de um recurso eletrônico com os metadados em XML.

A Listagem 9 apresenta uma parte de um registro MARC representado através de MARCXML, o qual é delimitado pelas *tags record* e representa os campos e subcampos do MARC pelas marcações *datafield* e *subfield* respectivamente.

```
<record>
  <leader>01142cam 2200301 a 4500</leader>
  <controlfield tag="001"> 92005291 </controlfield>
  <controlfield tag="003">DLC</controlfield>
  <controlfield tag="005">19930521155141.9</controlfield>
  <controlfield tag="008">920219s1993 caua j 000 0 eng
</controlfield>
  <datafield tag="042" ind1=" " ind2=" ">
```

```

        <subfield code="a">lcac</subfield>
    </datafield>
    <datafield tag="050" ind1="0" ind2="0">
        <subfield code="a">PS3537.A618</subfield>
        <subfield code="b">A88 1993</subfield>
    </datafield>
    <datafield tag="082" ind1="0" ind2="0">
        <subfield code="a">811/.52</subfield>
        <subfield code="2">20</subfield>
    </datafield>
    <datafield tag="100" ind1="1" ind2=" ">
        <subfield code="a">Sandburg, Carl,</subfield>
        <subfield code="d">1878-1967.</subfield>
    </datafield>
    <datafield tag="245" ind1="1" ind2="0">
        <subfield code="a">Arithmetic /</subfield>
        <subfield code="c">
Carl Sandburg ; illustrated as an anamorphic adventure by Ted Rand.
        </subfield>
    </datafield>
    ...
</record>

```

Listagem 9 - Exemplo de representação de um registro MARC através de MARCXML.

Fonte: The Library of Congress (2010).

### 5.3 RDF

De acordo com Manola e Miller (c2004), o *Resource Description Framework* (RDF) é uma linguagem para representar informações sobre recursos na *web*, mais particularmente metadados. Ele também pode ser utilizado para representar informações sobre objetos que podem ser identificados na *web*, mesmo não possuindo acesso direto a esses objetos, como por exemplo, informações de especificação, preço, disponibilidade de itens em um sistema de compras. O RDF destaca-se pela simplicidade com que busca estruturar o conteúdo contido na *web* sendo ele considerado um modelo de representação para descrição semântica de recursos.

Pereira, Ribeiro Júnior e Neves (2005) descrevem o RDF como uma linguagem gramática para estruturar metadados na *web* na descrição de recursos, promovendo desta forma a interoperabilidade entre sistemas de informação.

Segundo Dziekaniak e Kirinus (2004), o principal objetivo do RDF é a representação de metadados de forma a descrever um recurso. O RDF recebeu influência de várias fontes, das quais destacam-se: linguagens de marcação (HTML, XML, SGML), esquemas de metadados, ontologias, programação orientada a objetos, entre outras. O RDF é composto por três elementos principais: recurso, propriedade e tripla.

O RDF utiliza a notação de triplas, na qual cada declaração utiliza uma tripla para ser representada, sendo esta composta por sujeito, predicado e objeto. Um nó do grafo RDF pode participar de várias declarações, sendo desta forma fundamental o modelo de grafo, pois sua representação através de notações necessita da repetição dos nós que participam de várias declarações (MANOLA; MILLER, c2004).

Santanchè (2003), afirma que a tripla, um conjunto formado por sujeito, predicado e objeto, é a base da descrição de recursos em RDF. A representação do RDF ocorre por meio de *Directed Labelled Graph* (DLGs), que são grafos rotulados e direcionados, no qual o sujeito aponta para o objeto por uma aresta rotulada denominada de predicado, sendo ainda, o sujeito e o objeto, arestas rotuladas. A Figura 6 apresenta um exemplo de representação do RDF através de uma DLG, que neste caso representa o sujeito livro “Vida dos dinossauros” possuir como autor (predicado) o objeto “Sr. Horácio”.

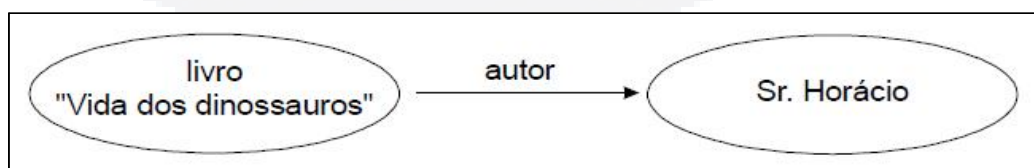


Figura 6 - RDF representado por um grafo DLG.

Fonte: Santanchè (2003).

A Figura 7 apresenta a mesma DLG da Figura 6, porém com mais declarações a respeito do mesmo sujeito, formando assim um recurso.



Figura 7 - RDF representado por um grafo DLG com várias declarações, formando um recurso.

Fonte: Santanchè (2003).

O RDF é destinado a situações em que as informações precisam ser processadas por aplicações e não apenas serem exibidas para pessoas. O RDF permite a interoperabilidade de informações entre sistemas através da disponibilização de recursos constituídos sob sua estrutura que define uma forma comum de expressar as informações. Ele baseia-se na utilização de identificadores da *web*, os *Uniform Resource Identifiers* (URIs), para identificar recursos e descrever suas propriedades (MANOLA; MILLER, c2004). O grafo DLG da Figura 8 descreve um recurso via RDF através de nodos, arcos e URIs.

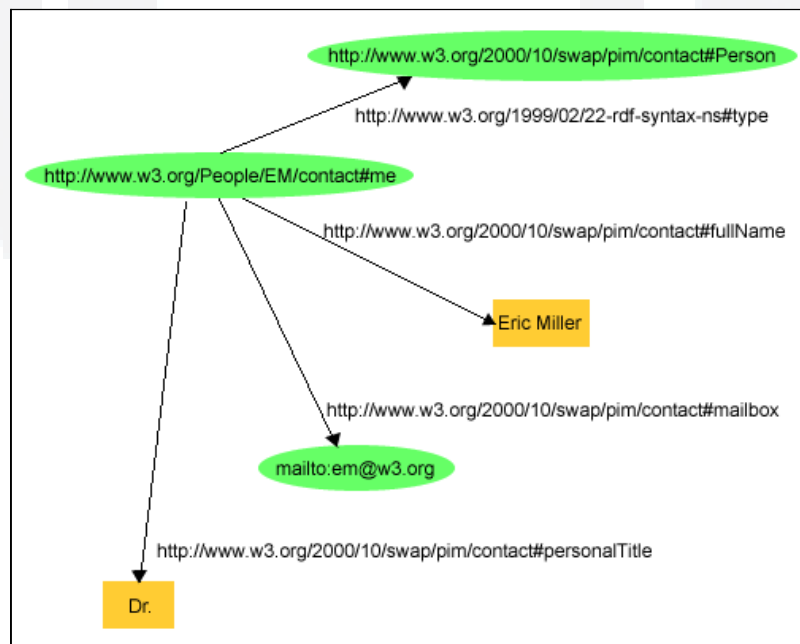


Figura 8 - Grafo de representação de um recurso utilizando URIs com o framework RDF.

Fonte: Manola e Miller (2003).

De acordo com Manola e Miller (c2004), o RDF também pode ser representado utilizando marcações XML, com a denominação de RDF/XML, sendo que o mesmo grafo representado na Figura 8 seria representado conforme a Listagem 10.



```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>
</rdf:RDF>

```

Listagem 10 - Exemplo de representação do RDF utilizando XML (RDF/XML).

Fonte: Manola e Miller (c2004).

Santanchè (2003), define o RDF, através da análise de sua sigla: descrição (*description*), recurso (*resource*) e *framework*.

Descrição: provem do fato do RDF basear-se na associação de descrições aos recursos, sendo estas compostas por declarações. Cada declaração define um valor para uma propriedade associada ao recurso e cada uma dessas declarações recebe uma denominação na terminologia RDF, a chamada tripla, de acordo com Manola e Miller (c2004). A Figura 9 representa a descrição de um recurso, descrito em português - O livro “Vida dos dinossauros” tem como autor o Sr. Horácio - através do *framework* RDF (SANTANCHÈ, 2003).

<b>Sujeito</b>	Recurso	Livro “Vida dos dinossauros”
<b>Predicado</b>	Propriedade	autor
<b>Objeto</b>	Valor	Sr. Horácio

Figura 9 - Exemplo da estrutura de triplas do RDF.

Fonte: Santanchè (2003).

Recurso: de acordo com Manola e Miller (c2004), um recurso representa genericamente qualquer coisa que pode ser identificado através da *web*, sendo que a sua identificação é realizada através de URIs, tendo o objetivo de identificar qualquer recurso *web* de forma única. Santanchè (2003), cita que o tipo mais popular de URI é um subconjunto denominado URL, o qual realiza a identificação de recursos por meio de sua localização física. O URI possui outro subconjunto denominado *Uniform Resource Name* (URN), que também tem por objetivo a definição de identificadores únicos de rede, porém sem estarem associados à localização física do recurso, permitindo desta forma a preservação do

identificador mesmo com a mudança de localização do recurso ou mesmo indisponibilidade do mesmo.

*Framework*: segundo Santanchè (2003), em uma declaração feita em RDF o sujeito e o predicado são identificados por URIs e o objeto pode ser identificado por um texto ou também por uma URI. Para o predicado, a URI permite fazer referência a propriedades descritas em esquemas, que tornam possível o estabelecimento de padrões para a descrição de recursos, como um exemplo claro deste tipo de utilização, pode-se citar o *Dublin Core*, o qual é um esquema de metadados que define um conjunto genérico de elementos utilizando o RDF para caracterizá-los na forma de propriedades. Um exemplo de propriedade do *Dublin Core* que está associado a uma URI é o autor/criador: <http://purl.org/dc/elements/1.1/creator>. A Figura 10 apresenta uma declaração RDF utilizando como predicado uma URI do esquema *Dublin Core*, sendo o sujeito também representado por uma URI na Figura 11, por uma recomendação do *framework* objetivando evitar ambiguidades.



Figura 10 - Predicado RDF utilizando um elemento do padrão *Dublin Core* no formato de URI.

Fonte: Santanchè (2003).

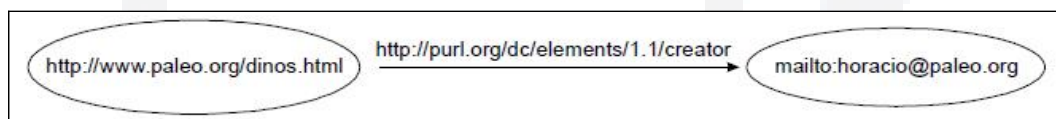


Figura 11 - Representação do RDF utilizando apenas URIs.

Fonte: Santanchè (2003).

De acordo com NISO (2004), o RDF, desenvolvido pelo W3C, é um modelo de dados para a descrição de recursos na *web* que fornece um mecanismo para integrar múltiplos esquemas de metadados. Em RDF um *namespace* é definido por uma URL que aponta para um recurso da *web*, o qual descreve o esquema de metadados que é usado na descrição. *Namespaces* múltiplos podem ser definidos, permitindo que os elementos de diferentes esquemas possam ser combinados na descrição de um único recurso. Descrições múltiplas, criadas em momentos diferentes, para diferentes fins, também podem ser ligados uns aos outros.

Segundo Souza e Alvarenga (2004), o RDF é uma recomendação do W3C para implementação de páginas *web* e descrição de recursos, devido a sua *web* semântica. O RDF estabelece um padrão para inserir padrões de metadados na codificação XML, denominada de *RDF Schema* (RDFS), desta forma, uma mesma descrição RDF pode utilizar mais de um padrão de metadados, sendo que cada padrão utilizado é definido como um *namespace*. A Listagem 11 apresenta a utilização de dois *namespaces* na descrição RDF, sendo o primeiro o *namespace* padrão do RDF (rdf) e o segundo, o *namespace* do padrão *Dublin Core* (dc).

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF SYSTEM "http://purl.org/dc/schemas/dcmes-xml-
20000714.dtd">
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntaxns#"
xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description
about="http://www.niso.org/standards/resources/Metadata_Demystified.p
df">
    <dc:title>Metadata Demystified</dc:title>
    <dc:creator>Brand, Amy</dc:creator>
    <dc:creator>Daly, Frank</dc:creator>
    <dc:creator>Meyers, Barbara</dc:creator>
    <dc:subject>metadata</dc:subject>
    <dc:description>Presents an overview of metadata conventions in
publishing.</dc:description>
    <dc:publisher>NISO Press</dc:publisher>
    <dc:publisher>The Sheridan Press</dc:publisher>
    <dc:date>2003-07</dc:date>
    <dc:format>application/pdf</dc:format>
  </rdf:Description>
</rdf:RDF>
```

Listagem 11 - RDF/XML utilizando o *namespace Dublin Core*.

Fonte: NISO, (2004).

## 6 IMPLEMENTAÇÃO DO SISTEMA

A partir do estudo efetuado acerca dos conceitos, padrões de metadados, protocolos de interoperabilidade e demais tecnologias que permeiam os sistemas de informação destinados à catalogação e disponibilização de informações digitais do meio acadêmico e científico, buscou-se a implementação do sistema proposto. Nesta seção serão apresentadas as características técnicas e funcionais do sistema implementado.

### 6.1 Visão geral

Observou-se a existência de uma ampla gama de esquemas de descrição de metadados e identificaram-se duas vertentes principais para os mesmos: os padrões voltados para a descrição de recursos digitais e os padrões voltados para a catalogação de documentos físicos.

Já em relação aos protocolos de interoperabilidade encontram-se menos opções, porém a maioria delas consiste em padrões internacionais, e acentuando a distinção encontrada entre os padrões de metadados, os protocolos de interoperabilidade também seguem basicamente duas vertentes: os protocolos voltados para a promoção de interoperabilidade de metadados de descrição de recursos digitais e os voltados para metadados de catalogação de documentos físicos.

As duas vertentes citadas nos parágrafos anteriores consistem apenas em uma percepção do autor deste trabalho, já que os padrões e protocolos estudados são genéricos o suficiente para serem utilizados de outras formas, como em algumas propostas conhecidas e ainda sendo perceptível uma tendência a utilização de forma mais generalizada dos mesmos.

A partir do estudo efetuado, foi possível avaliar as características dos padrões de metadados e protocolos mais utilizados no âmbito dos catálogos *on-line* de bibliotecas e repositórios digitais, de modo que, considerou-se a metodologia de colheita de metadados mais adequada para implementação proposta, devido ao fato de tornar menos oneroso o processo de recuperação de informações a partir de uma interface de busca, já que, não necessita de frequentes consultas a repositórios externos, que podem sofrer interferência em seu desempenho de vários fatores, possivelmente ocasionando uma maior demora na execução das consultas, além de tornar excessivamente onerosa a ordenação de resultados de consultas provenientes de várias fontes, já que para isso seria necessário recuperar todos os resultados possíveis e posteriormente efetuar sua ordenação. Porém a adoção da metodologia de colheita de metadados também possui desvantagens, como a de não possuir os dados atualizados em tempo real, dependendo de uma frequente sincronização, assim como pelo fato de não poder utilizar de todos os mecanismos de consulta já disponíveis nos protocolos Z39.50, SRW e SRU.

A partir da metodologia adotada, foi realizado o levantamento e a análise de requisitos, nos quais pode-se observar algumas necessidades básicas para o sistema, dentre as quais, encontram-se a escolha de um padrão de metadados principal, para o qual os dados coletados deverão ser convergidos. Desta forma, tornou-se necessário a implementação de uma estrutura de banco de dados que suporte tal padrão, assim como uma estrutura de software compatível e extensível aos demais.

A implementação do cliente OAI-PMH possibilita a integração nativa com diversas bases de dados de documentos digitais, como: BDU, *Scientific Electronic Library Online* (SCIELO), Biblioteca Digital Brasileira de Teses e Dissertações (BDTD), Biblioteca Digital Jurídica (BDJur), *Bioline International*, *Journal of Information Systems and Technology Management* (JISTEM) e diversos outros repositórios acadêmicos e científicos, nacionais e internacionais. Já o protocolo OAI-ORE não foi tomado como alternativa, devido ao fato de onerar de maneira significativa o processo de sincronização com o *data providers*, já que o mesmo efetua a sincronização dos metadados juntamente com os recursos digitais, assim como pelo fato de fugir do objetivo da proposta, já que não se pretende criar um repositório digital.

Desta forma, para viabilizar a localização de recursos digitais em repositórios que possuem interface para os protocolos Z39.50, SRW e SRU, será avaliada como primeira

alternativa a implementação da metodologia de *harvesting* de metadados sobre sua tecnologia, como pode ser observado no modelo apresentado na Figura 12.

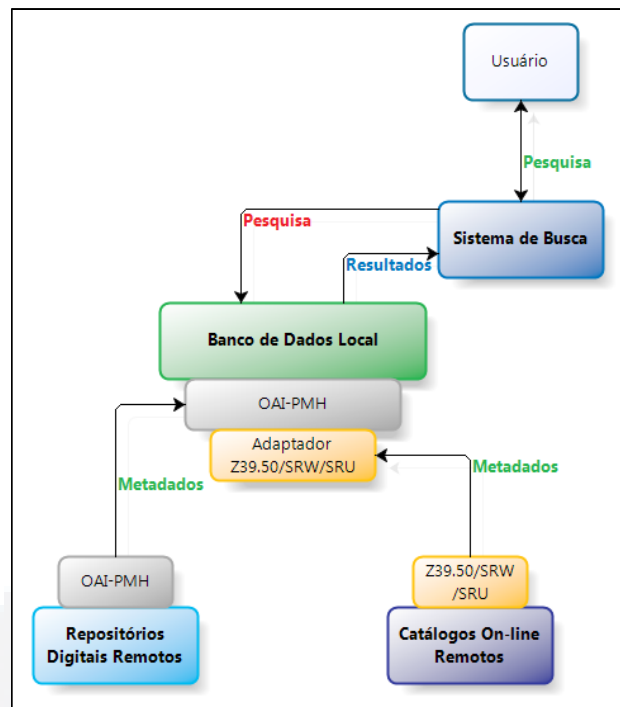


Figura 12 - Modelo de *harvesting* de metadados sobre repositórios dotados de Z39.50, SRW e SRU.

Como padrão base de metadados foi eleito o *Dublin Core* não qualificado, pelo fato de ser um padrão de metadados bastante simples e de âmbito mundial na descrição de recursos digitais, sendo utilizados pela maioria das aplicações voltadas para esta área e por ser um padrão tomado como base para a especificação de novos padrões, como é o caso do padrão Brasileiro de Metadados de Teses e Dissertações (MTD-BR), que foi desenvolvido no âmbito do projeto da Biblioteca Digital Brasileira de Teses e Dissertações. O *Dublin Core* é considerado o mínimo múltiplo comum dos padrões de metadados ao utilizar o protocolo OAI-PMH, desta forma, o repositório que possui este protocolo no mínimo suporta o *Dublin Core* não qualificado. Também foi levado em consideração sua simplicidade e generalidade no que refere-se à convergência de outros padrões de metadados para *Dublin Core*.

Ainda como requisito fundamental para o sistema, encontra-se a necessidade do desenvolvimento de uma interface de busca, assim como de um cadastro de repositórios de dados, e interfaces para cadastros auxiliares que possibilitem o gerenciamento e configuração de todas as atividades do sistema.

Para fins deste trabalho, frente à diversidade de padrões de metadados e de protocolos de interoperabilidade encontrados, tomam-se por base os padrões de metadados *Dublin Core* e MARC 21 e os protocolos de interoperabilidade OAI-PMH, Z39.50, SRW e SRU.

## 6.2 Ambiente de desenvolvimento

O sistema proposto foi desenvolvido utilizando o ambiente definido abaixo, sendo que algumas das tecnologias utilizadas serão melhor detalhadas nas próximas seções:

- *Hardware: notebook* com acesso a *internet*, sendo necessária a disponibilidade de comunicação em determinadas portas;
- Linguagem de programação: Java;
- Sistema gerenciador de banco de dados: PostgreSQL;
- *Integrated Development Environment (IDE)*: Netbeans;
- Servidor de aplicação: Apache Tomcat.

## 6.3 Linguagem de programação

Para implementação do sistema, foi adotada a linguagem de programação Java, mais especificamente, as tecnologias para desenvolvimento web: *Servlet*, *Java Server Pages (JSP)* e *JavaServer Pages Standard Tag Library Actions (JSTL)*. Segundo Temple et al. (c2004), a utilização destas tecnologias oferece diversas vantagens em relação ao uso de outras abordagens, sendo que as principais vantagens são herdadas da própria linguagem Java:

- a) Portabilidade: a aplicação desenvolvida pode ser implantada em diversas plataformas, como por exemplo, Windows, Unix e Macintosh, sem que seja necessário modificar ou mesmo reconstruir a aplicação;
- b) Facilidade de programação: a programação é orientada a objetos, com gerenciamento automático de memória;

- c) Flexibilidade: o Java encontra-se bastante difundido, contando com uma enorme comunidade de desenvolvedores, ampla documentação e diversas bibliotecas e códigos prontos, dos quais o desenvolvedor pode usufruir.

Jandl Junior (2009), afirma que os *Servlets* constituíram a primeira alternativa Java para a geração dinâmica de conteúdo para *web*, sendo que esta tecnologia oferecia algumas vantagens em termo de desempenho, independência de plataforma e arquitetura voltada para o desenvolvimento corporativo, em relação às tecnologias disponíveis na época, como *Common Gateway Interface* (CGI), *Hipertext Processor* (PHP) e o *Netscape Server-Side Javascript* (Netscape SSJS).

Ainda segundo Jandl Junior (2009), um *Servlet* consiste em uma classe Java que atua como um mini servidor, atendendo a pedidos específicos de múltiplos clientes, produzindo respostas aos *browsers* e operando como um componente em um servidor de aplicação *Java Enterprise Edition* (Java EE).

O desenvolvimento de aplicações *web* utilizando apenas *Servlets* é considerado oneroso, pois torna-se necessário construir toda estrutura HTML das páginas dentro da codificação Java, desta forma, a tecnologia JSP surgiu como uma evolução inovadora para o *Servlet*, a partir do aprimoramento de uma tecnologia para compilação de páginas proposta pela *Art Technology Group* (ATG) (JANDL JUNIOR, 2009). A proposta consistiu na integração da facilidade existente na construção de páginas HTML, com os benefícios funcionais do *Servlet*, desta forma, a tecnologia JSP tem como base o conceito de converter documentos HTML com marcações especiais, em *Servlets*. Gonçalves (2007), descreve que a primeira vez que uma página JSP é carregada pelo *container* JSP, o código Java é compilado gerando um *Servlet*, o qual é executado, sendo que as próximas chamadas, são enviadas diretamente ao *Servlet* gerado, não havendo mais a recompilação do código Java.

Segundo Temple et al. (c2004), as páginas JSP foram criadas para contornar algumas das limitações no desenvolvimento de aplicativos com *Servlets*. Em uma página JSP a formatação das páginas HTML pode ser separada da programação do sistema, permitindo sua modificação sem afetar o restante da aplicação.

De acordo com Jandl Junior (2009), a JSP utiliza marcações semelhantes as do XML para permitir a junção de código Java com elementos HTML, sendo que atualmente constitui



a tecnologia padrão da camada de apresentação de aplicações *web* construídas sobre a arquitetura Java EE. Os principais benefícios obtidos na utilização da JSP são:

- Separar as camadas da aplicação;
- Flexibilizar o design de páginas;
- Simplificar o processo de desenvolvimento;
- Facilitar testes de aplicações.

A tecnologia JSP, por sua vez, trouxe consigo novas problemáticas, sendo a maior delas a quantidade de código Java inserido dentro das páginas JSP, causando a diminuição da manutenibilidade dos sistemas. Desta forma, ao longo do tempo, foram sendo incorporadas à JSP novas tecnologias e funcionalidades. Neste contexto merecem destaque a *Expression Language* (EL) e o suporte a *tags*, o qual deu origem a JSTL.

A partir da versão 1.1, a JSP ganhou suporte a *custom tags*, ou seja, ações customizadas e reutilizáveis. As *custom tags* podem formatar e processar dados, repetir ações, acessar banco de dados e realizar outras operações (JANDL JUNIOR, 2009). A possibilidade de criação de *custom tags* levou a uma proliferação de *taglibs* (conjunto de *tags* personalizadas), de modo que surgiu a necessidade de padronizar seu uso, surgindo então a JSTL, cujo objetivo é permitir a criação de páginas JSP empregando exclusivamente a estrutura de *tags*, ou seja, sem o uso de *scriptlets* (isto é, sem a presença de código Java) (JANDL JUNIOR, 2009).

Segundo Jandl Junior (2009), a JSTL disponibiliza um conjunto de *custom tags* destinadas a realizar tarefas comuns, tais como: repetição, tomada de decisão, seleção, acesso a banco de dados, internacionalização e processamento de documentos XML. Ele também cita que o uso desta *taglib* proporciona vantagens como melhor legibilidade, reusabilidade, separação entre a lógica de negócio e a apresentação, entre outros.

A JSTL encapsula as funcionalidades mais comuns existentes em muitas aplicações JSP, de modo a permitir a criação de páginas JSP sem código Java, aumentando assim a legibilidade do código e a interação entre os desenvolvedores e *webdesigners*, proporcionando assim maior rapidez no desenvolvimento de sistemas web (GONÇALVES, 2007).

A partir da versão 2.0 da JSP foi introduzida a possibilidade da criação de *custom tags* a partir de *tag files*, os quais consistem em arquivos JSP que podem conter ações JSP, declarações, *scriptlets*, expressões EL, marcações fixas HTML ou XML e também fazer uso de *JavaBeans* e *taglib* como a JSTL. A utilização de *tag files* permite a criação de *tags* simples, com atributos ou com corpo, dotados ou não de repetições, removendo completamente *scriptlets* e expressões EL das páginas JSP (JANDL JUNIOR, 2009).

Ainda na versão 2.0 da JSP, em conjunto com a versão 2.0 da JSTL, foi introduzida a EL, como uma linguagem de acesso a dados que permite realizar operações aritméticas e lógicas, bem como a leitura de propriedade de *JavaBeans*. Sua utilização facilita o acesso às informações presentes no servidor durante a apresentação de páginas (JANDL JUNIOR, 2009).

Jandl Junior (2009) cita que a EL oferece diversas vantagens sobre o uso de *scriptlets*, como maior concisão, coerção automática e uso indireto da API Java, porém é menos flexível e realiza apenas um conjunto limitado de ações, em geral restritas aos métodos padronizados pela especificação dos *JavaBeans*.

Para Gonçalves (2007), a EL consiste basicamente em declarações que disponibilizam uma sintaxe mais simples para executar algumas das ações antes realizadas através de *scriptlets*, de modo a simplificar certas características do desenvolvimento de aplicações *web*, principalmente no que tange a legibilidade de código.

Durante o desenvolvimento do sistema, buscou-se alcançar algumas características, tais como reusabilidade e manutenibilidade. Desta forma foram adotadas algumas tecnologias e premissas para o desenvolvimento do mesmo. Foi definido a não utilização de *scriptlets* dentro das páginas JSP, de modo a facilitar a manutenção de código, assim como flexibilizar o *design* das páginas, sendo necessário para isto à adoção de tecnologias como as *custom tags*, *tag files*, JSTL e EL. Foram utilizadas as funções de repetição e tomada de decisão da JSTL em substituição aos *scriptlets*, assim como a função de internacionalização disponível na mesma.

A internacionalização do sistema foi adotada tendo em vista a larga escala de utilização que tais sistemas recebem, sendo necessário desta forma, que a arquitetura do sistema suporte tal característica.

A interface do sistema foi construída utilizando maciçamente *tag files*, na forma de componentes reutilizáveis, com sua lógica implementada através das tecnologias JSTL e EL. Na Listagem 12 pode ser observada a *tag file* utilizada para a criação de campos de entrada de texto, a qual possui embutida toda a lógica de definição, preenchimento e validação dos campos. Esta combinação permitiu a construção de páginas JSP limpas, livres de código Java, padronizadas e de fácil manutenção, conforme pode ser observado na Listagem 13, a qual apresenta o código fonte da JSP do cadastro de notícias.

```

<%@attribute name="name" required="true" %>
<%@attribute name="value" %>
<%@attribute name="help" type="java.lang.Integer" %>
<%@attribute name="required" type="java.lang.Integer" %>
<%@attribute name="validation_class" %>
<%@attribute name="location" %>
<%@attribute name="readonly" type="java.lang.Integer" %>
<%@attribute name="date" type="java.lang.Integer" %>
<%@attribute name="type" %>
<%@attribute name="limit" %>
<%@attribute name="div_id" %>
<c:if test="${date == 1}">
    <script type="text/javascript">
        $(function() {
            $("#${name}").datepicker({dateFormat: 'dd/mm/yy'});
        });
    </script>
</c:if>
<c:if test="${not empty limit}">
    <script type="text/javascript">
        $(function() {
            $("#${name}").limit('${limit}');
        });
    </script>
</c:if>
<div id="${div_id}">
    <label for="${name}">
        <fmt:message key="${location}.${name}" bundle="${bundle}" />
        <c:if test="${required == 1}">
            <span id="required_marc" *</span>

```

```

        </c:if>
    </label>
    <input id="${name}" name="${name}" ${readonly == 1 ? "readonly" :
    ""} type="${not empty type ? type : 'text'}" value="${value}"
    class="${required == 1 ? 'required' : ''} ${validation_class} ui-
    widget-content" />
    <span id="_${name}" class="help">
        <c:if test="${help == 1}">
            <fmt:message key="${location}.${name}.help" bundle="${
    {bundle}" />
        </c:if>
    </span>
</div>

```

Listagem 12 - Exemplo de *tag file* utilizada na implementação do sistema.

```

<jsp:directive.include file="../../layout/importsTags.jsp" />
<c:set var="location" value="form_admin.cadNews" scope="request" />
<c:set var="url" value="news.do" scope="request" />
<w:title />
<div>
    <form action="${url}" name="cadastro" id="cadastro" method="post">
        <w:hidden value="SAVE" />
        <w:input name="id" value="${x.idNews}" readonly="1"
        validation_class="ui-widget-header" />
        <w:input name="title" value="${x.title}" required="1" />
        <w:text name="content" value="${x.content}" />
        <w:input name="initialdate" value="${x.initialDate}" date="1"
        help="1" />
        <w:input name="finaldate" value="${x.finalDate}" date="1"
        help="1" />
        <w:input name="level" value="${x.level}" help="1"
        validation_class="digits" />
        <w:checkbox name="restricted" checked="${x.restricted ? 1 : 0}"
        help="1" />
        <br />
        <w:focus field="#title" />
        <w:menuButtons />
    </form>
</div>

```

Listagem 13 - Cadastro de notícias: exemplo de utilização de *tag files* dentro de uma página JSP.

A página JSP, apresentada na Listagem 13, a qual faz uso de *tag files*, conforme exemplificado na Listagem 12, resultou no formulário apresentado na Figura 13, que possui ajuda para os campos, validação e componentes visuais complexos.

The screenshot shows a web form with the following elements:

- Título:** A blue header bar.
- Código:** A text input field.
- Título: \*** A text input field with a red error message: "Este campo é obrigatório."
- Conteúdo:** A large text area with a scroll bar.
- Data Inicial:** A date input field with a red error message: "Data inicial de exibição."
- Data Final:** A date input field with a calendar dropdown showing "Agosto 2011".
- Nível:** A dropdown menu with a red error message: "Entre apenas com dígitos."
- Restrita:** A grid of numbers from 1 to 31, with the number 21 highlighted in yellow.
- Buttons:** "Salvar", "Novo", and "Pesqui" buttons.

Figura 13 - Exemplificação das tecnologias de programação utilizadas para desenvolvimento da interface.

Em conjunto com as tecnologias Java, utilizadas para o desenvolvimento da aplicação, fez-se uso de *Javascripts* a partir da biblioteca *jQuery*, buscando-se obter maior interatividade e dinamismo nas páginas da aplicação.

Segundo Silva (2010), *jQuery* pode ser definida como simplicidade de desenvolvimento, ao ponto que reduz em algumas linhas a quantidade de código que seria necessário ao utilizar *Javascript* puro. O autor cita que *jQuery* pode ser utilizada em uma aplicação para, dentre outras coisas:

- Adicionar efeitos visuais e animações;
- Acessar e manipular *Document Object Model* (DOM);
- Requisições assíncronas com o servidor através de *Asynchronous Javascript and XML* (AJAX);
- Prover interatividade;
- Alterar conteúdos;
- Modificar apresentação e estilização;
- Simplificar tarefas específicas de *JavaScript*.

Silva (2010), caracteriza a biblioteca *JQuery* da seguinte forma:

- Utiliza seletores *Cascading Style Sheets* (CSS) para localizar elementos da estrutura de marcação HTML da página;
- Possui arquitetura compatível com instalação de *plug-ins* e extensões em geral;
- É indiferente a inconsistências de renderização entre navegadores;
- É capaz de interação implícita, isto é, não há necessidade de construção de *loops* para localização de elementos no documento;
- Admite programação encadeada, ou seja, cada método retorna um objeto;
- É extensível, pois admite criação e inserção de novas funcionalidades na biblioteca existente.

Desta forma, a biblioteca *JQuery* foi largamente utilizada para desempenhar várias funções no que refere-se à exibição de mensagens, chamadas Ajax, efeitos de animação e alteração de conteúdo. Foram utilizados ainda os *plug-ins* para *JQuery*: *fancybox*, *jquery.validate* e *fg.menu*, com o objetivo de permitir a criação de janelas *iframe*, validação de campos e construção de menus, respectivamente.

A utilização das tecnologias Java para desenvolvimento *web* levam a necessidade da utilização de um servidor de aplicação que implemente as funcionalidades do JEE essenciais à aplicação. Desta forma, o Tomcat da Apache foi escolhido como servidor de aplicação, para tratar as requisições dos clientes, pelo fato do mesmo consistir em um software livre, implementar todas as funcionalidades do JEE necessárias para o projeto, ser um dos *containers* recomendado pelo Java e por já se ter um conhecimento prévio no projeto de aplicações com o mesmo.

#### 6.4 Arquitetura

Para implementação do sistema, foi adotado o paradigma de desenvolvimento e *design Model-View-Controller* (MVC), mais especificamente o MVC *Model 2*. Segundo Gonçalves (2007), o MVC é um paradigma de desenvolvimento que propõe a separação da aplicação em três camadas distintas: *Model* (modelo), *View* (apresentação) e *Controller* (controlador). A

*Model* representa os dados da aplicação, manipulando-os e aplicando operações, sendo representada por classes que trabalham no armazenamento e recuperação de dados. A *View* é responsável pela apresentação visual dos dados fornecidos pela camada *Model*. Já o *Controller* é responsável pelo intermédio dos dados entre as camadas *Model* e *View*, sendo encarregado de receber solicitações do cliente, providenciar os dados necessários e encaminhar a resposta.

De acordo com Gonçalves (2007), dois modelos para o desenvolvimento de aplicações surgiram para o Java a partir da incorporação do paradigma MVC:

*MVC Model 1*: também chamada de *page-centric*; esta arquitetura é mais comum no desenvolvimento de aplicações *web*, por ser mais fácil de implementar. Consiste no desenvolvimento de uma aplicação utilizando basicamente um conjunto de páginas JSP, como pode ser observado na Figura 14, sendo indicada para aplicações pequenas que contém um número limitado de usuários e contém pouca lógica de negócio.

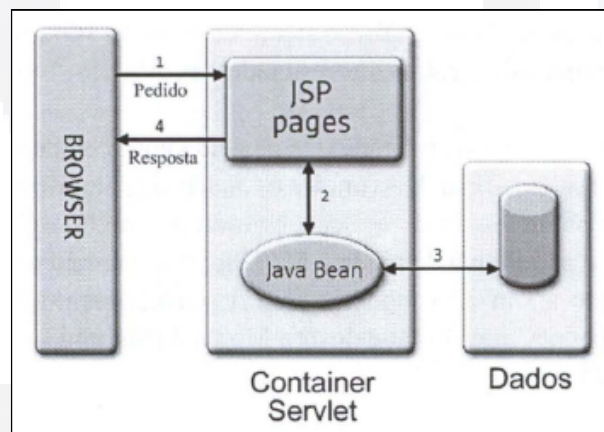


Figura 14 - Arquitetura MVC *Model 1*.

Fonte: Gonçalves (2007, p. 388).

*MVC Model 2*: arquitetura focada na utilização de *Servlets* e páginas JSP, sendo o *Servlet* usado como controlador (*Controller*), de modo a receber as requisições do usuário e fornecer os dados para a camada de apresentação (*View*), sendo esta desenvolvida em JSP, como pode ser observado na Figura 15. Este modelo é mais robusto que o *Model 1*, sendo indicado para projetos maiores e mais complexos.

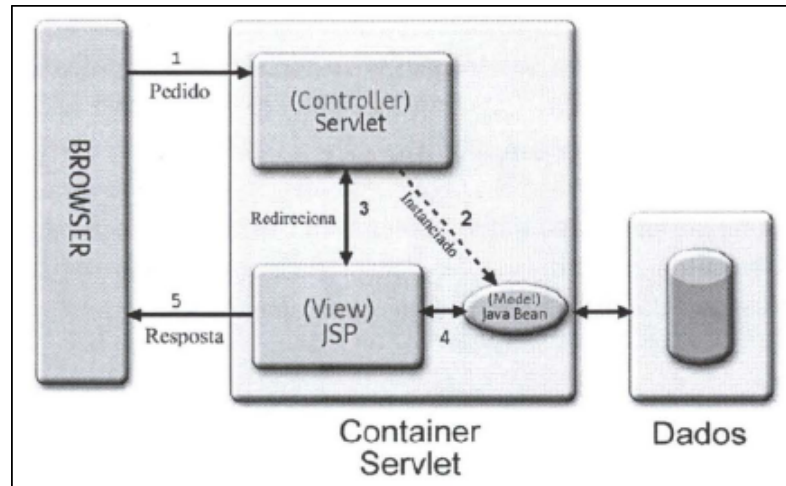


Figura 15 - Arquitetura MVC Model 2.

Fonte: Gonçalves (2007, p. 393).

Em conjunto com o MVC Model 2, adotou-se o padrão *Data Access Object* (DAO), sendo este, segundo Gonçalves (2007), o padrão mais utilizado para acesso a dados contidos em bancos de dados. Ele fornece uma interface independente para persistir objetos de dados. O padrão DAO tem como premissa a ideia de colocar todas as funcionalidades encontradas no desenvolvimento de acesso e trabalho com dados em um só local, tornando mais simples a sua manutenção. Tipicamente um DAO inclui métodos para inserir, selecionar, atualizar e excluir objetos de um banco de dados.

## 6.5 Modelo do banco de dados

Com base nos requisitos especificados para o projeto foi proposto o modelo do banco de dados apresentado na Figura 16, o qual contempla o armazenamento dos dados coletados pelo sistema de *harvesting* e a parte gerencial do sistema.



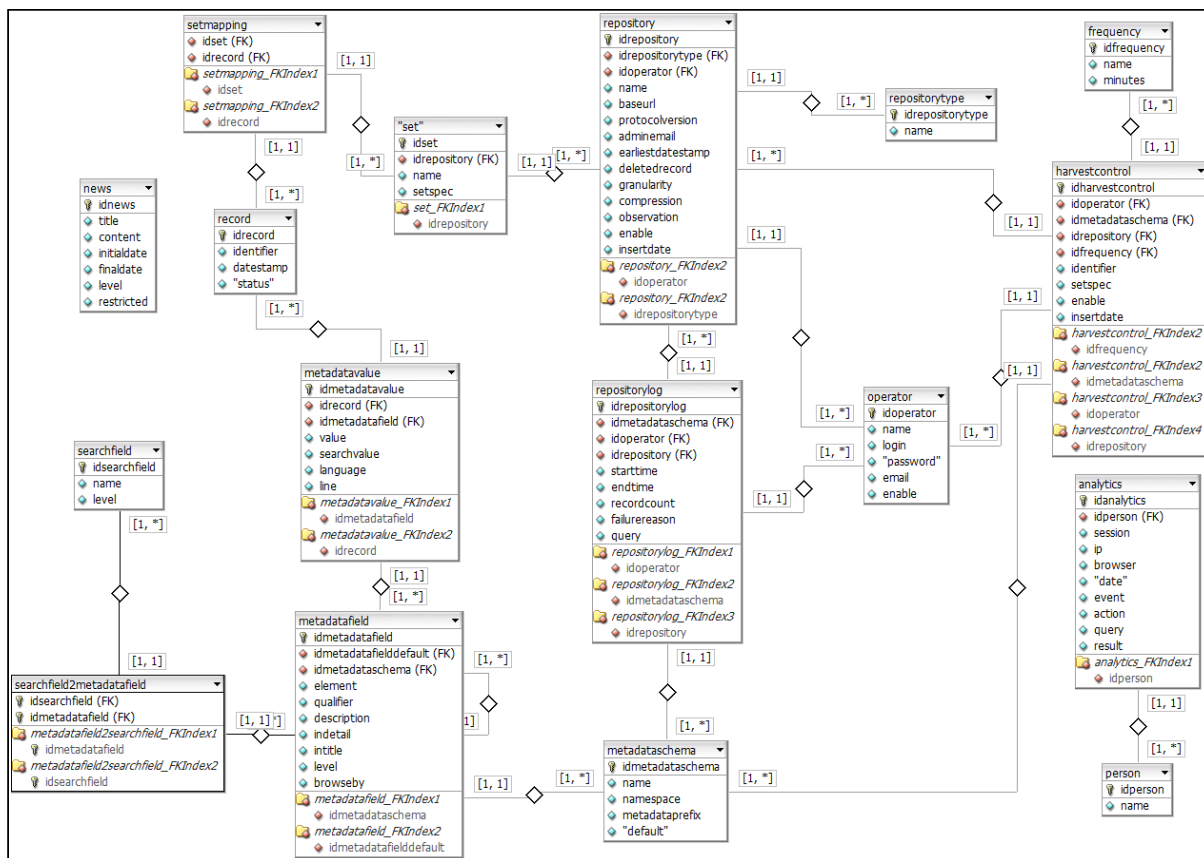


Figura 16 - Modelo do banco de dados.

As tabelas *set*, *setmapping*, *record* e *metadatavalue* são modeladas para armazenar os metadados coletados pelo mecanismo de *harvesting*, os quais serão disponibilizados para recuperação pela pesquisa do sistema. A forma que foi realizada a modelagem dos campos de metadados, objetivou permitir a extensibilidade do padrão de metadados utilizado e também a repetição ilimitada de registros de determinados campos; isto é possível pelo fato de não ser utilizado um modelo convencional de tabelas de dados para a gravação dos metadados, sendo que as tradicionais colunas da tabela foram transformadas em linhas, com apenas um campo texto para gravação do conteúdo e os títulos das colunas foram transformados em uma tabela secundária, a *metadatafield*, permitindo desta forma o acréscimo de novos campos sem necessitar de alterações no sistema.

De modo a permitir a utilização de múltiplos padrões de metadados, a tabela *metadatafield* possui uma relação com ela mesma, de modo a permitir a definição da relação de um campo qualquer com um campo do padrão principal determinado, o *Dublin Core*.

A tabela denominada *repository* armazena informações de funcionamento dos repositórios de dados utilizados pelo sistema. Esta tabela está relacionada a uma tabela de

*logs*, a *repositorylog*, na qual serão armazenados os dados referentes aos *harvestings* efetuados no repositório.

A tabela *harvestingcontrol* possui grande importância para o funcionamento do cliente do protocolo OAI-PMH, pois nela serão definidas as frequências de atualização dos dados dos repositórios de acordo com o *set* ou identificador desejado. As atualizações de *harvesting* são realizadas de forma incremental, ou seja, são coletados os dados dos servidores periodicamente, sendo a busca determinada pela data do último *harvesting* efetuado com sucesso, registrado na tabela *repositorylog*.

Como funcionalidades a parte dos objetivos principais do sistema, a tabela *news* foi criada para permitir o cadastro de notícias a serem apresentadas a usuários e administradores do sistema; já a tabela *analytics* armazena as operações realizadas pelos usuários na busca, de modo a permitir avaliar o comportamento do usuário em relação ao sistema.

## 6.6 Banco de dados e conexão

O software gerenciador de banco de dados escolhido para ser utilizado na aplicação foi o PostgreSQL, pelo fato de ser um software livre e por se ter experiência em sua utilização.

O PostgreSQL é um SGBD relacional, utilizado para armazenar informações de soluções de informática em todas as áreas de negócios existentes, bem como administrar o acesso a estas informações (MILANI, 2008).

Para efetuar o gerenciamento das conexões com o banco de dados foi utilizada a biblioteca *Jakarta-Commons Database* disponível no *Tomcat*. Esta biblioteca é responsável pelo gerenciamento de um *Pool* de conexões. Sua implementação utilizando *Data Sources* elimina a necessidade de código *Java Database Connectivity* (JDBC) para carregar o *driver*, utilizar o *Drive Manager* e manter informações de conexão na classe responsável pela conexão. A utilização de *Data Sources* em conjunto com um *Pool* de conexões impacta significativamente no desempenho da aplicação, uma vez que as conexões são criadas de uma maneira mais eficiente.

*Pool* de conexões reduzem expressivamente o tempo de conexões estabelecidas criando uma conexão física no início do sistema. Quando uma aplicação requer uma conexão, uma destas conexões físicas é provida a essa aplicação. Em um sistema comum, sem o *pool* de conexão, quando a aplicação termina de usar a conexão, este

a desconecta. Porém, no caso de uma conexão física, essa é devolvida somente para o pool de conexões, onde espera o próximo pedido da aplicação para um novo acesso ao banco de dados (GONÇALVES, 2007, p. 240).

O *Data Source* é configurado no arquivo *context.xml*, que encontra-se no diretório *META-INF* da aplicação. O código da Listagem 14 representa o arquivo *context.xml*, onde é configurado o *Data Source*. Nele são determinadas informações de usuário e senha do banco de dados, *driver* de conexão, endereço URL da conexão com o banco de dados, quantidade máxima de conexões ativas e inativas, entre outras.

```
<?xml version="1.0" encoding="UTF-8"?>
<Context antiJARLocking="true" path="/searcher" >
  <Resource name="jdbc/searcher"
    auth="Container"
    type="javax.sql.DataSource"
    username="postgres"
    password="postgres"
    driverClassName="org.postgresql.Driver"
    url="jdbc:postgresql://localhost:5432/searcher"
    maxActive="20"
    maxIdle="5" />
</Context>
```

Listagem 14 - Exemplo de configuração de um *Data Source* no arquivo *context.xml*.

O *Data Source* é recuperado através do *Java Naming and Directory Interface* (JNDI). Na Listagem 15 pode ser visualizada a implementação de uma classe Java capaz de fazer a recuperação do *Data Source*. Nesta classe é criada uma instância estática de um *ServiceLocator* que armazena um *DataSource*, o qual é obtido pelo método *lookup* executado sobre o contexto da aplicação.

```
import javax.sql.*;
import javax.naming.*;

public class ServiceLocator {
  private static ServiceLocator instance;
  private ServiceLocator() {}
  public DataSource getDataSource() {
    DataSource ds = null;
    try{
      Context ctx = new InitialContext();
      Context envCtx = (Context) ctx.lookup("java:comp/env");
      ds = (DataSource) envCtx.lookup("jdbc/searcher");
    }
  }
}
```

```

    }catch(NamingException ne){
        //tratamento da exceção
    }
    return ds;
}
public static ServiceLocator getInstance(){
    if(instance == null){
        instance = new ServiceLocator();
    }
    return instance;
}
}

```

Listagem 15 - Exemplo de código para obter um *Data Source*.

Após a recuperação do *Data Source* é necessário apenas obter sua conexão com o banco de dados, conforme pode ser visto no exemplo apresentado na Listagem 16. O *Data Source* é obtido através do acesso estático ao método *getInstance* da classe *ServiceLocator*, apresentada na Listagem 15, e aplicado sobre seu resultado o método *getDataSource*. Neste *Data Source* obtido, estará a conexão com o banco de dados, retornada pelo método *getConnection*.

```

import java.sql.*;
import javax.sql.*;
public class Conexao {
    private static Conexao instancia;
    private Conexao(){}
    public Connection criarConexao(){
        try{
            DataSource ds =
ServiceLocator.getInstance().getDataSource();
            if(ds != null){
                return ds.getConnection();
            }
        }catch(SQLException sqle){
            //tratamento da exceção
        }
        return null;
    }
    public static Conexao getInstancia(){
        if(instancia == null){

```

```

        instancia = new Conexao();
    }
    return instancia;
}
}

```

Listagem 16 - Exemplo de código para obter uma conexão de um *Data Source*.

## 6.7 Layout da interface

O layout da interface do sistema foi definido seguindo o padrão especificado na Figura 17, sendo dividido em quatro partes principais: Cabeçalho, Menu, Corpo Principal e Rodapé. No Cabeçalho foi reservado um espaço para um *banner*, de modo a identificar o sistema. Na área de *menu* encontram-se disponíveis os *links* que dão acesso às funcionalidades do sistema. O Corpo Principal é um espaço dinâmico de acordo com o menu ou a tarefa realizada. Por fim, o Rodapé possui espaço reservado para disponibilização de algumas informações como *copyright*, versão, contato, entre outros. O *menu* foi definido com orientação horizontal, objetivando obter mais espaço disponível para a apresentação dos resultados das pesquisas e interação com o usuário.

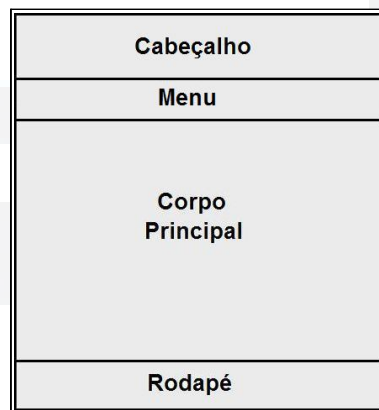


Figura 17 - Padrão das páginas a serem desenvolvidas.

O *layout* da interface foi desenvolvido utilizando *JQuery User Interface (jQuery UI)* em conjunto com CSS, de forma a tornar o *layout* totalmente independente da aplicação e de fácil personalização, uma vez que faz uso de arquivos externos separados do código fonte da aplicação.

De acordo com Wellman (2009), o *JQuery UI* é uma extensão da biblioteca *JQuery*, que busca disponibilizar um conjunto de componentes ricos e interativos, construídos para melhorar a interface com o usuário das aplicações *web*.

O CSS é o padrão W3C de definição da apresentação visual para as páginas *web* (GRANNELL, 2009). O HTML foi concebido como uma linguagem de *markup* estrutural, mas as demandas feitas pelos usuários e pelos designers incentivaram os fabricantes de *browsers* a suportar e desenvolver *tags* orientadas à apresentação.

Ainda segundo Grannell (2009), o conceito por trás do CSS é eliminar a apresentação e separar o desenho do conteúdo, deixando com que o HTML ou o XHTML trate a estrutura, e o CSS, de forma separada, para aplicar a apresentação visual.

Desta forma, o *JQuery UI* foi utilizado como base para o layout da aplicação, sendo que necessidades não contempladas pelo mesmo foram definidas em um documento CSS separado. Tais características atribuem à aplicação grande interatividade e riqueza visual, assim como permitem fácil customização e garante uma interface padronizada já que o *JQuery UI* foi desenvolvido sobre os preceitos da W3C.

## 6.8 Sistema e funcionalidades

Com o intuito de validar a proposta apresentada foi desenvolvido o sistema de acordo com as especificações levantadas e do estudo realizado. Nesta seção serão apresentados detalhes funcionais sobre a implementação realizada.

O *layout* do sistema desenvolvido foi padronizado seguindo algumas características básicas, conforme especificado na seção 6.7. O *layout* da interface foi dividido em quatro partes principais, sendo o menu em orientação horizontal, conforme pode ser visto na Figura 18.

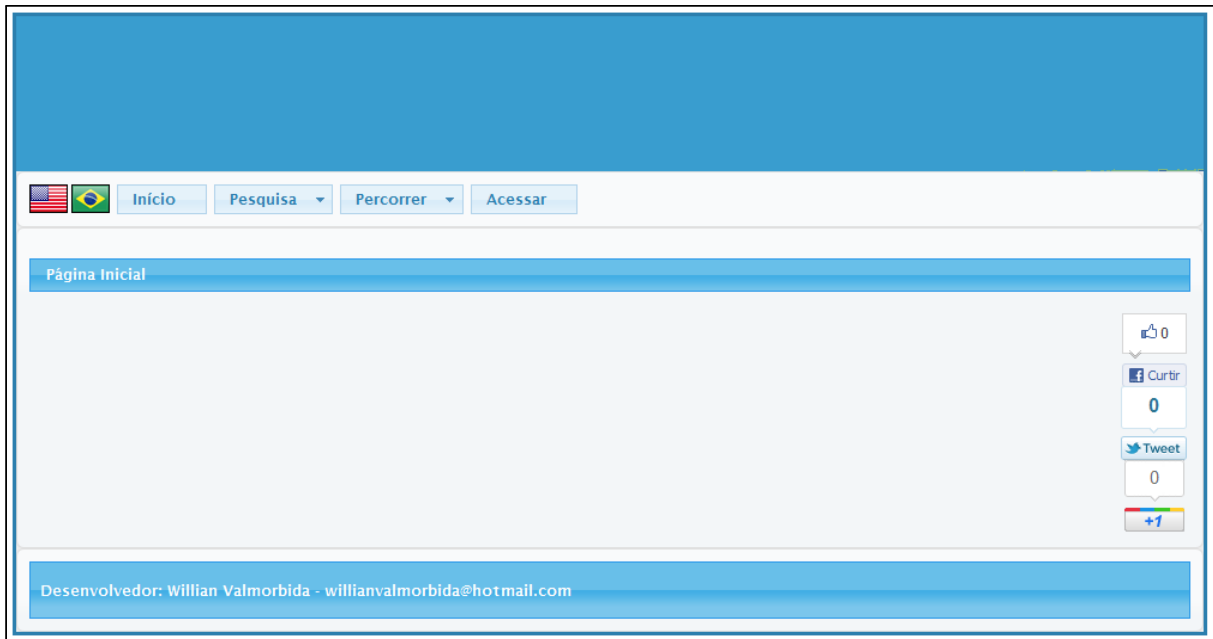


Figura 18 - *Layout* da interface do sistema.

Ainda na Figura 18, assim como na Figura 19, pode-se observar que o sistema foi internacionalizado, disponibilizando inicialmente os idiomas português brasileiro e inglês. Além da internacionalização, o sistema conta com uma integração básica com algumas das redes sociais mais utilizadas nos dias atuais, visto que, este tipo de integração pode consistir em uma forma simples e barata de promover a disseminação de informações, assim como a utilização do sistema.

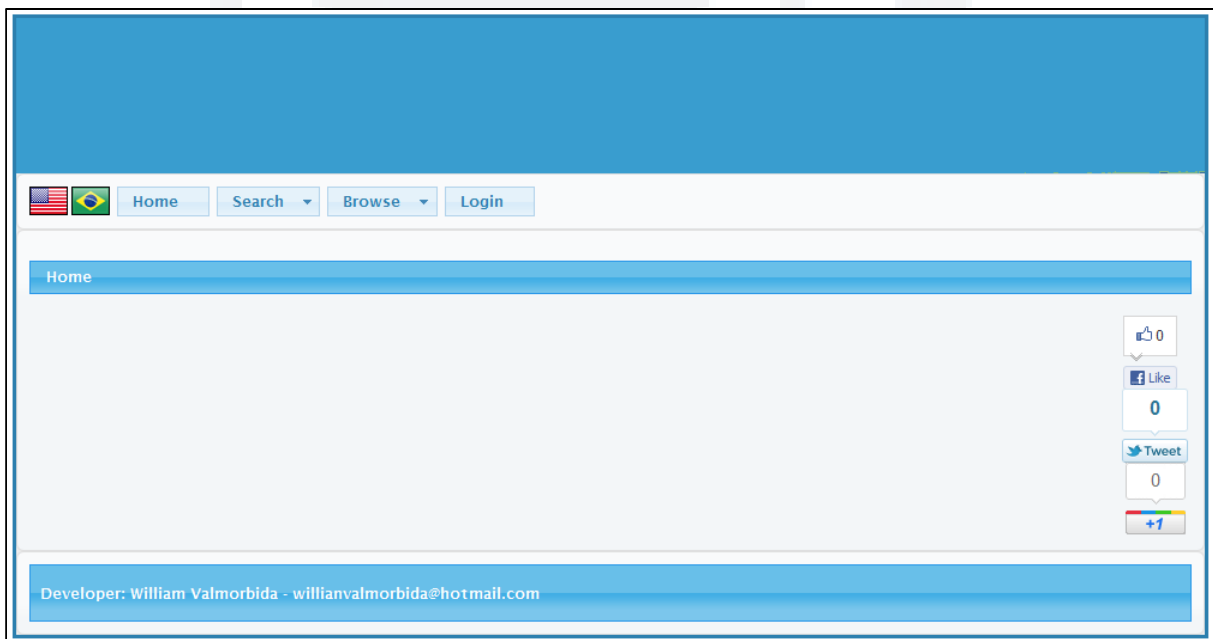


Figura 19 - *Layout* da interface do sistema.

Na Figura 20, é apresentada uma tela simples de cadastro, o cadastro de notícias. Nesta figura podem-se destacar algumas características importantes em relação a *layout* e também a usabilidade do sistema:

- Os campos dos cadastros receberam uma marcação com um asterisco para campos obrigatórios, como é o caso do campo “Título”;
- Os campos desabilitados para edição receberam um fundo com outra tonalidade de cor para destacá-los, como é o caso do campo “Código”;
- Os campos, que possuem necessidade para tal, receberam um auxílio de preenchimento que aparece dinamicamente na tela, quando o campo recebe o foco, como pode ser visto no campo denominado “Data Inicial”;
- Para facilitar o preenchimento dos campos do tipo data, foi disponibilizado um componente de calendário, como pode ser observado no campo “Data Inicial”;
- Ao acionar a ação de salvamento de um cadastro, os campos são validados antes do envio do formulário, e o usuário é informado campo a campo sobre os erros que ocorreram, com uma mensagem ao lado do campo correspondente, como é o caso do campo “Título” que não foi preenchido no exemplo da Figura 20. Este tipo de validação também é utilizado para tipos de dados como inteiros, e-mails, entre outros.



Figura 20 - *Layout* da interface e usabilidade.

O acesso a cada uma das funcionalidades do sistema, realizado pelo menu, é direcionado sempre ao formulário de busca, o qual dispõe normalmente de duas ações: “Pesquisar” e “Novo”. A ação “Novo” irá direcionar o usuário ao formulário de cadastro da correspondente funcionalidade, o qual sempre possui uma ação para realizar o caminho inverso.

Os resultados das pesquisas são apresentados em tabelas com linhas intercaladas de cor diferente, para permitir melhor visualização dos resultados aos usuários, conforme pode ser observado na Figura 21. Tais tabelas são providas das informações de quantidade de registros recuperados, quantidade de páginas e página de resultados atual. A primeira coluna das tabelas sempre apresenta as ações disponíveis para cada resultado, com um *hint* para facilitar sua identificação. A paginação dos resultados só é exibida quando os resultados ultrapassam uma página, sendo a quantidade de resultados por página limitado a uma configuração incluída no arquivo descritor da aplicação (*web.xml*).

Pesquisa de Repositório

**Código:**

**Tipo:** --- Selecione ---

**Base URL:**

**Nome:**

**E-mail do administrador:**

Encontrado(s) 35 resultado(s) em 3 página(s). (Página atual: 1)

Ação	Código	Tipo	Nome	Base URL	E-mail	Ativo?
	19	OAI-PMH	Ação Educativa	http://www.bdae.org.br/dspace-oai/request	dspace@acaeducativa.org	Sim
	20	OAI-PMH	BDJur no STJ	http://bdjur.stj.gov.br/oai/request	fbotelho@stj.gov.br	Sim
	21	OAI-PMH	BDTD Ibict	http://oai.ibict.br/myoai/oai2.php	equipebtdt@ibict.br	Sim
	22	OAI-PMH	DSpace@FGV	http://virtualbib.fgv.br/oai/request	biblioteca.digital@fgv.br	Sim
	23	OAI-PMH	Bioline International	http://www.bioline.org.br/oai	bioline.international@utoronto.ca	Sim
	3	OAI-PMH	WesScholar	http://wescholar.wesleyan.edu/cgi/oai2.cgi	support@dc.bepress.com	Não
	4	OAI-PMH	SciELO - Scientific Electronic Library Online	http://www.scielo.br/oai/scielo-oai.php	scielo@scielo.org	Sim
	36	SRU	Libris	http://libris.kb.se/sru/swepub		Sim
	16	SRU	CUNY Union Catalog	http://apps.appl.cuny.edu:5661/U-CUN01		Não
	6	OAI-PMH	CADAIR: Aberystwyth University repository	http://cadair.aber.ac.uk/dspace-oai/request	cadair@aber.ac.uk	Sim
	7	OAI-PMH	British History Online	http://www.british-history.ac.uk/oai/oai.aspx	british-history@sas.ac.uk	Sim
	8	OAI-PMH	ePrints Soton	http://eprints.soton.ac.uk/cgi/oai2	eprints@soton.ac.uk	Sim
	9	OAI-PMH	Roehampton Research Papers	http://rrp.roehampton.ac.uk/cgi/oai2.cgi	support@dc.bepress.com	Sim
	10	OAI-PMH	CCLRC ePublication Archive	http://epubs.cclrc.ac.uk/oai	epublications@rl.ac.uk	Sim
	1	OAI-PMH	Biblioteca Digital da UNIVATES	http://www.univates.br/bdu_oai/request	bdu@univates.br	Sim

Páginas: ◀ ◀◀ 1 2 3 ▶▶ ▶

Figura 21 - Formulário de pesquisa.

Como uma funcionalidade extra para o sistema, foi desenvolvido um gerenciador de notícias, o qual permite a definição do período de validade da notícia, a sua ordem de prioridade em relação a outras e o nível de restrição, ou seja, se é pública ou aparece apenas para operadores do sistema. Um exemplo de funcionamento das notícias pode ser visualizado na Figura 22 e Figura 23.

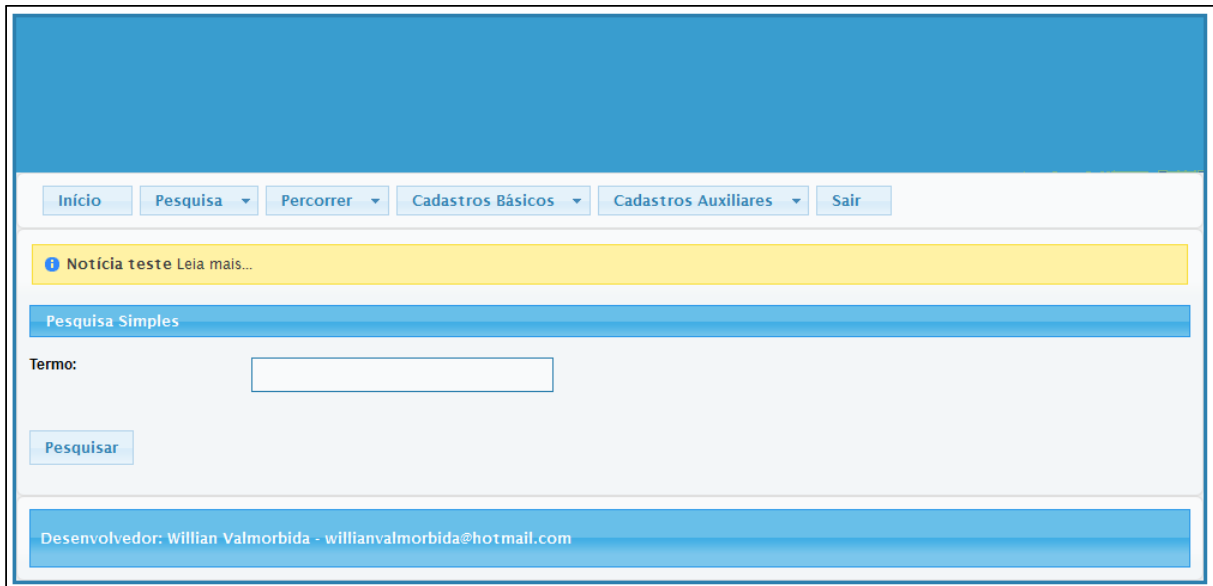


Figura 22 - Pesquisa simples exibindo uma notícia.

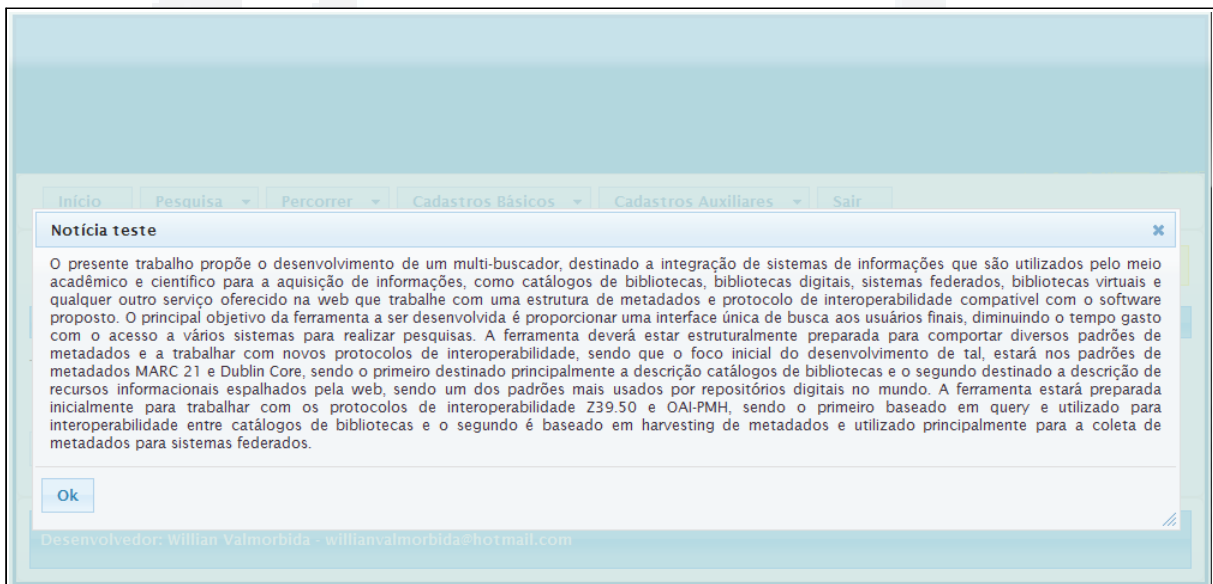


Figura 23 - Exibição de uma notícia detalhada.

Objetivando prover um suporte completo às funcionalidades do sistema, foi desenvolvida a manutenção completa para os seguintes cadastros: operador, tipo de repositório, frequência, campo de metadados, esquema de metadados, configuração de pesquisa, notícia, controle de *harvesting* e repositório, sendo que o termo manutenção, para efeitos deste capítulo consiste nas funcionalidades de cadastro, alteração, exclusão e pesquisa.

O cadastro de repositório consiste em um formulário comum, com entrada para as informações administrativas necessárias aos repositórios. Ele possui uma funcionalidade que permite a recuperação automática dos dados de determinado repositório, realizada através de uma chamada AJAX a camada de negócio do sistema, que por sua vez, executa uma chamada

do verbo *identify* ao protocolo OAI-PMH do repositório informado através da base URL, sendo esta funcionalidade disponível apenas para os repositórios OAI-PMH complacentes.

No formulário de cadastro de campos de metadados é possível determinar para cada campo se o mesmo aparece no título do resultado da pesquisa ou em seus detalhes, assim como a prioridade de ordenação em que irá ser exibido. Ainda neste formulário é possível configurar para quais campos será disponibilizado automaticamente a funcionalidade de percorrer. Tal funcionalidade é complementar a pesquisa simples, permitindo a listagem e a pesquisa de valores de determinados campos, sendo que a partir dela, pode-se chegar aos registros relacionados. Esta funcionalidade está disponível de forma fixa para as listagens de repositórios e coleções, e pode ser customizada para qualquer outro campo de metadados disponível no sistema. A Figura 24 apresenta um exemplo de utilização da funcionalidade de percorrer, sendo que a listagem apresentada foi filtrada por um determinado termo.

A funcionalidade de percorrer não suporta repositórios remotos, ou seja, os que possuem interação com o sistema através do uso natural dos protocolos Z39.50, SRW e SRU, pelo fato de não ser uma funcionalidade padrão implementada para quaisquer campos dos protocolos em questão.

The screenshot displays the search interface of the BDU system. At the top, there is a navigation bar with buttons for 'Início', 'Pesquisa', 'Percorrer', 'Cadastros Básicos', 'Cadastros Auxiliares', and 'Sair'. The 'Percorrer' dropdown menu is open, showing options: 'Percorrer Repositório', 'Percorrer Coleção', 'Percorrer por Título', 'Percorrer por Autor', and 'Percorrer por Assunto'. The 'Percorrer por Autor' option is selected, and the search term 'w' is entered in the 'Termo:' field. A 'Pesquisar' button is located below the search field. A yellow warning banner states: 'Esta funcionalidade não contempla repositórios remotos'. Below this, a message indicates 'Encontrado(s) 20 resultado(s) em 2 página(s). (Página atual: 1)'. The main content area shows a table of authors with the following names: Barrett, Willian; Coelho, Willian Marinho Dourado; Destro, Marco Willians Baena; Easterling, Willian E.; Farah, Marcio Willian; Gaspari, Jefferson Willians de; Hayasida, Willian; Jorge Willian Leandro Nascimento; Melo, Willian Augusto de; Moraes, Willian Bucker; Nahas, Willian Carlos; Nascimento, Jorge Willian Leandro; Nogueira, Douglas Willian; Nunes, Willian Mário de Carvalho; and Pomella, Alan Willian Vilela. At the bottom of the table, there is a pagination control showing 'Páginas: 1 2'. The footer of the page contains the text: 'Desenvolvedor: Willian Valmorbida - willianvalmorbida@hotmail.com'.

Figura 24 - Percorrer autor.

A manutenção do controle de *harvesting* é uma funcionalidade importante para o sistema, pois é através dele que são cadastrados agendamentos, por frequência de tempo, de *harvestings* automáticos, que especificam repositórios, *sets*, e registros. Na pesquisa do controle de *harvesting* foi disponibilizada uma ação que permite rodar manualmente o *harvesting* de determinado agendamento. Esta ação realiza uma requisição AJAX a camada de negócios do sistema, efetuando a importação ou atualização de todos os *sets* e registros presentes no repositório de acordo com a restrição determinada no cadastro.

Foi desenvolvida também uma interface de pesquisa simples (Figura 25) a qual permite aos usuários recuperar registros dos diversos repositórios cadastrados no sistema de forma unificada. Os registros recuperados apresentados na Figura 25, Figura 26, Figura 27,

Figura 28 e Figura 29 são provenientes de três bases de dados diferentes (Biblioteca Digital da Univates, SciELO e *Library of Congress*), sendo que as duas primeiras estão integradas ao sistema através do protocolo OAI-PMH e a terceira através do protocolo SRU.

A visualização dos resultados da pesquisa simples possui um conceito de apresentação diferente do utilizado para as buscas dos formulários administrativos. Os resultados recuperados são detalhados ao receberem um clique, sendo que o registro selecionado é expandido, enquanto o anteriormente detalhado é recolhido. Este conceito foi aplicado objetivando manter a tela organizada e limpa, assim como manter o usuário sempre na mesma página do sistema. Quando o resultado possui um link para seu conteúdo original, ele é aberto em uma janela *iframe*, conforme apresentado na Figura 26.

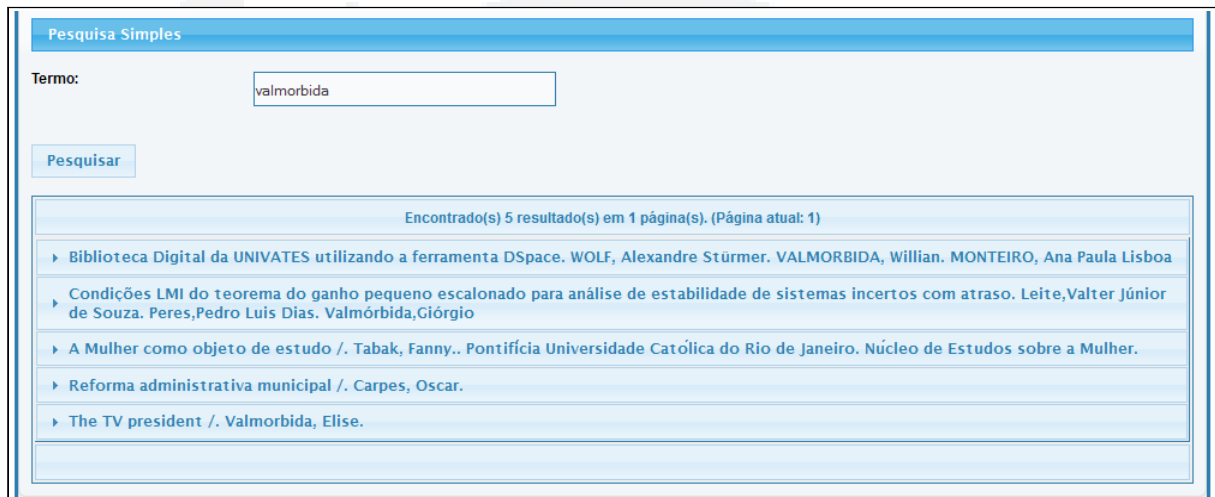


Figura 25 - Pesquisa simples.



Figura 26 - Pesquisa simples: conteúdo original do registro aberto em uma janela *iframe*.

Encontrado(s) 5 resultado(s) em 1 página(s). (Página atual: 1)

▼ Biblioteca Digital da UNIVATES utilizando a ferramenta DSpace. WOLF, Alexandre Stürmer. VALMORBIDA, Willian. MONTEIRO, Ana Paula Lisboa

<b>Título:</b>	Biblioteca Digital da UNIVATES utilizando a ferramenta DSpace
<b>Autor:</b>	WOLF, Alexandre Stürmer. VALMORBIDA, Willian. MONTEIRO, Ana Paula Lisboa
<b>Descrição:</b>	This work seeks to present the solution adopted by the Centro Universitário UNIVATES to supply the need of making available all the intellectual and scientific production of the Institution in global scale, through a Digital Library. Among several available tools, it was chosen a free software with possibilities of expansion and adaptation to the presented needs, called DSpace. DSpace is a digital repository developed under the partnership of MIT and HP with the objective of keeping, preserving, managing and disseminating the intellectual product of the organizations. For being a complex system, several other tools and technologies are used to confer functionality and enlarge its performance resources, which will be briefly mentioned. To explain the term "Digital Library", it will be presented categories of Libraries defined by some authors. Esse trabalho visa apresentar a solução adotada pelo Centro Universitário UNIVATES para suprir a necessidade de disponibilização da produção intelectual e científica da Instituição em escala global, através de uma Biblioteca Digital. Dentre várias ferramentas disponíveis, foi escolhido um software livre com possibilidades de expansão e adequação às necessidades apresentadas, chamado DSpace. O DSpace é um repositório digital desenvolvido através da parceria entre o MIT e a HP com o objetivo de guardar, preservar, gerir e disseminar o produto intelectual das organizações. Por ser um sistema complexo, várias outras ferramentas e tecnologias são utilizadas para dar funcionalidade e ampliar seus recursos de atuação, que serão brevemente abordadas. Para esclarecer o termo "Biblioteca Digital" serão apresentadas categorias de Bibliotecas definidas por alguns autores.
<b>Assunto:</b>	Biblioteca Digital. Biblioteca Univates. DSpace
<b>Tipo:</b>	Article. Presentation
<b>Acesse:</b>	<a href="#">Clique Aqui</a>
<b>Idioma:</b>	pt_BR
<b>Data:</b>	2009-03-31T13:54:52Z. 2009-03-31T13:54:52Z. 2009-03-31T13:54:52Z
<b>Repositório:</b>	Biblioteca Digital da UNIVATES
<b>Coleção:</b>	WET - Workshop de Engenharia e Tecnologia

▶ [Condições LMI do teorema do ganho pequeno escalonado para análise de estabilidade de sistemas incertos com atraso. Leite,Valter Júnior de Souza. Peres,Pedro Luis Dias. Valmórbida,Giörgio](#)

▶ [A Mulher como objeto de estudo /. Tabak, Fanny.. Pontifícia Universidade Católica do Rio de Janeiro. Núcleo de Estudos sobre a Mulher.](#)

▶ [Reforma administrativa municipal /. Carpes, Oscar.](#)

▶ [The TV president /. Valmorbida, Elise.](#)

Figura 27 - Pesquisa simples: resultado proveniente da Biblioteca Digital da Univates via protocolo OAI-PMH.

Encontrado(s) 5 resultado(s) em 1 página(s). (Página atual: 1)

▶ Biblioteca Digital da UNIVATES utilizando a ferramenta DSpace. WOLF, Alexandre Stürmer. VALMORBIDA, Willian. MONTEIRO, Ana Paula Lisboa

▼ [Condições LMI do teorema do ganho pequeno escalonado para análise de estabilidade de sistemas incertos com atraso. Leite,Valter Júnior de Souza. Peres,Pedro Luis Dias. Valmórbida,Giörgio](#)

<b>Título:</b>	Condições LMI do teorema do ganho pequeno escalonado para análise de estabilidade de sistemas incertos com atraso
<b>Autor:</b>	Leite,Valter Júnior de Souza. Peres,Pedro Luis Dias. Valmórbida,Giörgio
<b>Descrição:</b>	A equivalência entre resultados para análise de estabilidade baseados no método de Lyapunov de sistemas com atraso precisamente conhecidos e resultados de estabilidade robusta de sistemas de comparação sem atraso, empregando condições do teorema do ganho pequeno escalonado, foi apresentada recentemente na literatura. A extensão desses resultados para tratar sistemas com atraso e incerteza em domínios politópicos é a principal contribuição deste artigo. A partir da definição de uma realização genérica para um sistema, são apresentadas desigualdades matriciais lineares com variáveis extras, equivalentes a condições de ganho pequeno escalonadas. Devido à inserção dessas variáveis extras, condições independentes e dependentes do atraso para análise de estabilidade de sistemas com atraso em domínios politópicos podem ser elaboradas empregando matrizes de Lyapunov dependentes de parâmetro, resultando em condições menos conservadoras que outras já apresentadas na literatura, como ilustrado por exemplos numéricos.
<b>Assunto:</b>	desigualdades matriciais. Estabilidade robusta. teorema do ganho pequeno. sistemas com atraso
<b>Tipo:</b>	journal article
<b>Acesse:</b>	<a href="#">Clique Aqui</a>
<b>Editor:</b>	Sba: Controle & Automação Sociedade Brasileira de Automatica
<b>Idioma:</b>	pt
<b>Data:</b>	2007-12-01
<b>Formato:</b>	text/html
<b>Repositório:</b>	SciELO - Scientific Electronic Library Online
<b>Coleção:</b>	Sba: Controle & Automação Sociedade Brasileira de Automatica

▶ [A Mulher como objeto de estudo /. Tabak, Fanny.. Pontifícia Universidade Católica do Rio de Janeiro. Núcleo de Estudos sobre a Mulher.](#)

▶ [Reforma administrativa municipal /. Carpes, Oscar.](#)

▶ [The TV president /. Valmorbida, Elise.](#)

Figura 28 - Pesquisa simples: resultado proveniente da SciELO via protocolo OAI-PMH.

Encontrado(s) 5 resultado(s) em 1 página(s). (Página atual: 1)

▶ Biblioteca Digital da UNIVATES utilizando a ferramenta DSpace. WOLF, Alexandre Stürmer. VALMORBIDA, Willian. MONTEIRO, Ana Paula Lisboa

▶ Condições LMI do teorema do ganho pequeno escalonado para análise de estabilidade de sistemas incertos com atraso. Leite,Valter Júnior de Souza. Peres,Pedro Luis Dias. Valmórbida,Giörgio

▼ A Mulher como objeto de estudo /. Tabak, Fanny.. Pontifícia Universidade Católica do Rio de Janeiro. Núcleo de Estudos sobre a Mulher.

<b>Título:</b>	A Mulher como objeto de estudo /
<b>Autor:</b>	Tabak, Fanny.. Pontifícia Universidade Católica do Rio de Janeiro. Núcleo de Estudos sobre a Mulher.
<b>Descrição:</b>	At head of title: Centro de Ciências Sociais, Departamento de Sociologia e Política, Núcleo de Estudos sobre a Mulher.. Includes bibliographies.
<b>Assunto:</b>	Women--Brazil--Social conditions.. Women--Political activity--Brazil.
<b>Tipo:</b>	text
<b>Editor:</b>	Rio de Janeiro : Pontifícia Universidade Católica do Rio de Janeiro, Divisão de Intercâmbio e Edições,
<b>Idioma:</b>	por
<b>Data:</b>	1982.
<b>Repositório:</b>	Library of Congress

▶ Reforma administrativa municipal /. Carpes, Oscar.

▶ The TV president /. Valmorbida, Elise.

Figura 29 - Pesquisa simples: resultado proveniente da *Library of Congress* via protocolo SRU.

As principais atividades realizadas pelo usuário na pesquisa simples, bem como nas opções de percorrer, são registrados pelo sistema para uso estatístico, sendo que possuir estas informações possibilita atribuir ao sistema mecanismos que melhoram a interação com o usuário. Inicialmente foi implementado e disponibilizado um gráfico de acesso/utilização para cada registro, como apresentado na Figura 30. É importante destacar que esta funcionalidade está presente apenas para os registros que tiveram seus metadados colhidos e que conseqüentemente possuem um identificador, possibilitando a criação de uma relação com as ações efetuadas pelos usuários.

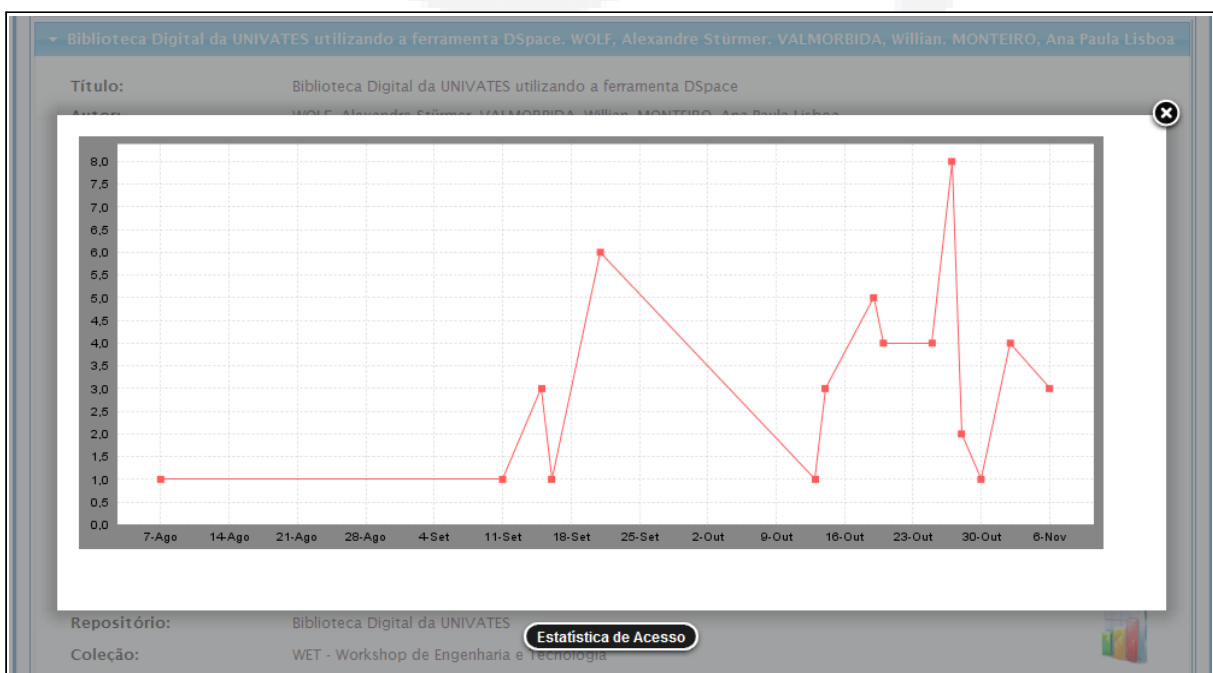


Figura 30 - Pesquisa simples: estatística de acesso.



A integração do sistema com os protocolos Z39.50, SRW e SRU foi implementada utilizando seu modo nativo e não por meio da colheita de metadados, conforme havia sido definido na proposta inicial. No Capítulo 7 são apresentados de forma detalhada os motivos que levaram a esta alteração no projeto.

O formato de integração adotado com os protocolos Z39.50, SRW e SRU limita a funcionalidade do sistema em ordenar os resultados, pelo fato de tornar necessário recuperar todos os registros correspondentes à pesquisa realizada, de todos os repositórios determinados, unir com os registros recuperados da base local e então ordenar os registros para apresentar ao usuário. Esta tarefa é de certa forma trivial, porém computacionalmente cara, pelo fato de que a integração de diversos repositórios resulta em quantidades consideráveis de registros.

Outra característica técnica relativa a esta forma de integração adotada, é a necessidade de a cada consulta realizada verificar a existência de resultados em cada um dos repositórios, efetuando requisições a cada um deles e aguardando o resultado para retornar aos usuários. Deste modo, implementou-se tal funcionalidade através de um *pool* de *threads*, de forma a paralelizar a execução das requisições a estes servidores, diminuindo consideravelmente o tempo de resposta. Esta integração foi desenvolvida de forma que, primeiramente é verificada a quantidade de registros que correspondem à consulta em cada repositório, e posteriormente são recuperados apenas os registros necessários calculados a partir da paginação.

O *pool* de *threads* implementado é exemplificado na Listagem 17 e Listagem 18, sendo criado a partir do *factory newCachedThreadPool* da classe *Executores*, criando assim um *pool* sem limite e com reutilização automática das *threads* que já finalizaram a execução, sendo que as *threads* que não são utilizadas por 60 segundos são automaticamente removidas. A classe da *thread* criada, implementa a interface *Callable* e o método *call*, retornando um tipo de dados especificado. Após a submissão das tarefas ao *pool* de *threads* os resultados são retornados a um objeto do tipo *Future*, o qual também permite monitorar e cancelar tarefas.

```
...
ExecutorService threadPool = Executors.newCachedThreadPool();
List<Future<Integer>> lThreads = new ArrayList<Future<Integer>>();
for (Repository r : lRepository) {
    SRU sru = new SRU();
```

```

...
Future<Integer> resultsSRU = threadPool.submit(new
SRUSearch(sru));
lThreads.add(resultsSRU);
}
for (Future<Integer> f : lThreads) {
try {
results.add(f.get());
} catch (Exception e) {
results.add(0);
e.printStackTrace();
}
}
...

```

Listagem 17 - Criação de um *pool* de *threads*.

```

public class SRUSearch implements Callable<Integer> {
...
public Integer call() throws Exception {
return SearchRequestNumberOfRecords();
}
public int SearchRequestNumberOfRecords() {
int results = 0;
...
return results;
}
...
}

```

Listagem 18 - *Thread* que implementa a classe *Callable*.

O sistema implementado possui áreas de acesso restrito, de modo que foi necessária a implementação de restrições de acesso a tais áreas. Estas restrições foram implementadas de modo a permitir o acesso a apenas usuários cadastrados no sistema, mediante autenticação utilizando login e senha, a qual foi armazenada no banco de dados unicamente codificada via *Message-Digest algorithm 5* (MD5), uma função de *hash*, caracterizada pela unidirecionalidade, ou seja, não possui caminho inverso que permita sua transformação no texto que a deu origem.

A restrição de acesso a páginas não autorizadas do sistema ocorreu através de um filtro que implementa a interface *javax.servlet.Filter* disponível na API da tecnologia *Servlet*. Sua

implementação provê um mecanismo que permite desacoplar e isolar determinados comportamentos, pelo fato de filtros serem executados antes e após determinada requisição, tendo por objetivo interceptar requisições, efetuar determinado processamento e liberar o processamento normal, passando a requisição ao *Servlet* ou JSP responsável.

Para seu funcionamento é necessário mapeá-lo no descritor da aplicação, assim como ocorre com os *Servlets*, sendo que para um filtro deve ser definido um ou mais padrões de URL que serão interceptados. Desta forma, foram mapeadas as URLs restritas e redirecionadas no filtro para a página de login do sistema, quando do usuário sem permissão para a página requisitada.

## 6.9 Testes da pesquisa simples

Visando validar a integração proposta e desenvolvida, buscou-se cadastrar alguns repositórios que possuem interfaces para os protocolos OAI-PMH, assim como SRU.

Dentre os repositórios utilizados e complacentes com o protocolo OAI-PMH pode-se citar: BDU, SciELO, BDTD, BDJur, Boline International e JISTEM. Destes, foi realizado o *harvesting* completo apenas da BDU, pelo fato da enorme quantidade de registros presente no restante dos repositórios. Alcançou-se cerca de 42.000 registros disponíveis para consulta na base de dados local, através do *harvesting* de algumas coleções dos repositórios citados acima.

Foram também utilizados três repositórios SRU complacentes para os testes: Libris, *Biblioteca Virtual Del Patrimonio Bibliografico* (DVPB) e *Library of Congress*, sendo desconhecida a quantidade total de registros disponíveis para consulta nestes repositórios.

De modo a exemplificar, pesquisou-se o termo *administration* no sistema, sendo retornados cerca de 20.000 resultados, unindo os registros coletados dos repositórios via OAI-PMH, lembrando que foram *harvestings* parciais, e os registros retornados pelos três repositórios consultados via SRU.

Durante o progresso dos testes observou-se determinada instabilidade no tempo de resposta de alguns repositórios consultados via SRU, sendo em alguns casos, a resposta obtida após aproximadamente 60 segundos, onde geralmente o tempo obtido entre a requisição e a

resposta variou de 6 e 10 segundos. Estes testes consideraram requisições de 15 registros obedecendo à determinada restrição de consulta. Já em relação ao repositório local, o tempo de resposta obtido foi consideravelmente inferior.

É importante ressaltar que os testes foram realizados utilizando uma conexão *wireless* comum e não foram verificados se os tempos obtidos não receberam outras influências a não ser o processamento dos repositórios externos e do sistema local.

Em consultas via SRU à *Library of Congress* deparou-se com um problema de resposta, no qual o serviço simplesmente respondia eventualmente um erro para a requisição, sendo testado diversas vezes com a mesma requisição e em algumas obteve-se êxito e em outras não. Acessando a URL solicitada por um *browser* obteve-se o mesmo resultado, conforme pode ser observado na Figura 31. Esse tipo de instabilidade resulta em problemas de funcionamento no sistema implementado, já que em um primeiro momento pode-se obter “x” resultados e já ao acessá-los via paginação a resposta pode não ser obtida. De modo a reduzir a chance de ocorrer tal problemática, o sistema realiza novamente a chamada para verificar a existência dos registros e ignora os que resultaram em erro, porém isso pode causar insegurança ao usuário ao passo que em um primeiro momento o sistema pode informar ter localizado “x” registros e já em um segundo a quantidade pode diminuir ou aumentar para a mesma consulta.



Figura 31 - Erro de resposta à requisição SRU da *Library of Congress*.

## 7 CONCLUSÃO

A partir do estudo e da implementação efetuada, pode-se observar algumas características importantes em relação aos protocolos e padrões estudados, sendo tais características fundamentais para permitir um desenvolvimento adequado sobre a proposta apresentada. Tal proposta consistiu na busca por permitir a integração dos padrões e protocolos mais conhecidos voltados a catálogos de bibliotecas e repositórios de documentos digitais, de modo a alcançar determinado grau de interoperabilidade, permitindo assim a criação de uma busca unificada entre tais sistemas, melhorando o grau de experiência dos usuários na busca e recuperação de informações acadêmicas e científicas.

Os protocolos OAI-PMH, OAI-ORE, Z39.50, SRW e SRU foram identificados como os mais difundidos, utilizados e implementados em catálogos de bibliotecas e repositórios digitais, sendo que encontram-se intimamente relacionados a alguns padrões de metadados, sendo eles normalmente genéricos, como o MARC 21 e o *Dublin Core*.

Tanto o protocolo OAI-PMH, quanto Z39.50 e seus derivados, tiveram suas raízes na busca por alternativas para proporcionar a interoperabilidade entre repositórios de dados, sendo que eles chegaram a resultados diferentes e eficazes para a resolução de problemas semelhantes, desta forma, pode-se dizer que ambas as abordagens têm mérito em seus próprios contextos, formando a base para a promoção da interoperabilidade, descoberta e distribuição de metadados no meio em que estão inseridos.

O protocolo OAI-PMH possui foco na simplicidade, objetivando facilitar a interoperabilidade de dados através do conceito de colheita de metadados, já o Z39.50, SRW e SRU, possuem foco voltado para a complexidade do padrão de metadados MARC 21 e a um modelo de requisições de busca. Devido a estas diferenças conceituais e apesar de

proporcionarem a possibilidade de alcançar resultados semelhantes, estes protocolos possuem algumas características técnicas em dissonância que dificultam sua integração. Tais características são citadas abaixo e foram responsáveis pelo rumo tomado na solução aqui apresentada.

Podem-se apresentar várias soluções diferentes para a proposta, sendo algumas de alta complexidade e outras de mais simples. Partiu-se então do princípio que tal solução teria que ser genérica o suficiente para ser implementada sobre quaisquer serviços dos protocolos analisados, de modo que deixasse de ser mais uma implementação isolada e específica para o problema.

A solução tomada como base, inicialmente, teve por intuito buscar a possibilidade de permitir a coleta de metadados, utilizando o conceito de *harvesting*, sobre os repositórios que disponibilizam os protocolos Z39.50, SRW e SRU, de modo a permitir uma maior flexibilidade no que tange a recuperação das informações, como pode ser visualizado no modelo apresentado na Figura 32. Esta metodologia foi proposta pelo fato do conceito empregado nos protocolos Z39.50, SRW e SRU ser dependente de diversos requisitos técnicos que podem onerar o tempo de resposta das buscas, como por exemplo, largura de banda, tempo de resposta dos servidores remotos, quantidade de servidores remotos a serem consultados simultaneamente, frequência de requisições, entre outros. Estes protocolos apresentam também dificultadores para execução de determinadas operações, como por exemplo, a ordenação de resultados de busca, já que torna necessário colher todos os metadados de todos os repositórios que atendem ao padrão da busca realizada, para posteriormente efetuar sua ordenação. O conceito de interoperabilidade baseado em *harvesting*, também possui alguns dificultadores técnicos, como espaço de armazenamento, capacidade de processamento local, atualização periódica, entre outros.

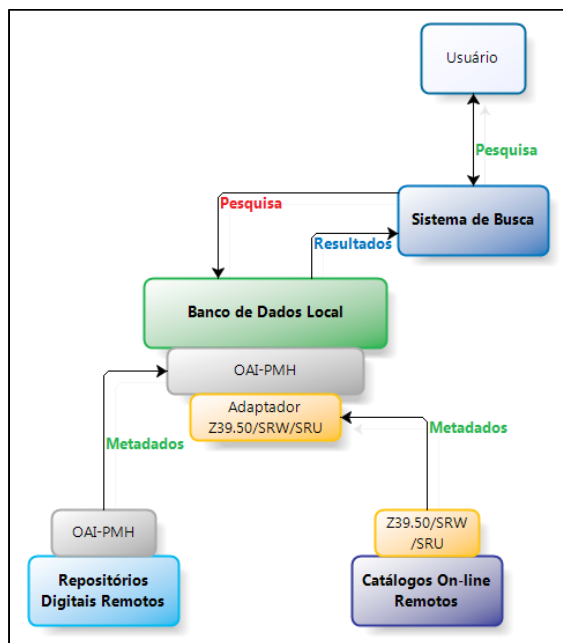


Figura 32 - Modelo da proposta baseada em *harvesting* de metadados para integração dos protocolos.

A partir do levantamento das características técnicas dos protocolos Z39.50, SRW e SRU, foram encontradas algumas dificuldades, devido aos mesmos não terem sido projetados para trabalhar com o conceito de colheita de metadados e conseqüentemente não possuir algumas características necessárias para viabilizar sua implementação de modo genérico. A implementação de tal solução leva normalmente a demandar adaptações no protocolo ou base de metadados em cada repositório que se busca interoperar, fato que torna a solução inviável de ser aplicada em larga escala.

Algumas formas de efetuar tal implementação foram propostas, porém todas elas demandam da necessidade de disponibilizar algumas características não nativas dos protocolos em questão, portanto, de eficiência duvidosa.

A implementação do conceito de colheita incremental, que é o padrão do protocolo OAI-PMH, gera a necessidade de disponibilizar nos repositórios a possibilidade de consulta por data de criação e alteração dos registros, remover o limite de resultados, que mesmo sendo elevado, pode levar a omissão de resultados, característica esta, não desejada para a implementação em questão. Ainda, torna-se necessário a disponibilização de um identificador único para cada registro do repositório, de modo a permitir sua sincronização, sendo ainda desejável que o protocolo informe os registros excluídos de modo a estes poderem ser tratados nos resultados das buscas do sistema integrador.

Os requisitos para a colheita incremental de metadados são possíveis de serem implementados, e alguns já estão disponíveis em determinados repositórios, porém a implementação destes requisitos pode ser dificultada por fatores comerciais, pelo fato da maioria dos sistemas de gerenciamento de bibliotecas serem proprietários e conseqüentemente de código fechado, impossibilitando tal tarefa. Ainda, é necessário levar em consideração que cada implementação dos protocolos, assim como das suas bases de dados e estrutura de metadados utilizados, possuem diferenças naturais que podem impactar no desempenho da implementação.

Foi levantada uma segunda possibilidade para permitir a colheita de metadados sobre os protocolos Z39.50, SRW e SRU, que consiste na colheita completa ou total, sendo caracterizada pela necessidade de colher todos os metadados de cada repositório a cada sincronização dos dados. Esta implementação esbarra nas mesmas questões comerciais da possibilidade anterior, porém pode demandar de um menor número de adaptações no protocolo. Para isso é necessário apenas permitir a recuperação de todos os registros do repositório, sendo através de um identificador único, coletando registro a registro a partir de uma lista pré fornecida, ou permitindo que o protocolo retorne todos os registros do repositório, eliminando-se o limite global da busca.

Tal alternativa pode ser mais simples de ser implementada do que a anterior, porém ela tende a ser consideravelmente mais cara computacionalmente, podendo tornar o processo de colheita ineficiente devido a demasiada carga gerada pelo excessivo número de solicitações necessárias aos *data providers*. A sincronização dos dados também tende a ocorrer em intervalos de tempo maior ao comparar com uma colheita incremental, já que demanda de maior tempo para ser executada, pois mesmo em *data providers* que possuem uma interface OAI-PMH pode-se observar uma visível diferença nos tempos de resposta para as requisições de um *data provider* para outro, sendo que um teste realizado com a BDTD estimou-se a necessidade de vários dias, talvez semanas, para execução de uma colheita completa.

Pode-se pensar em diversas formas de propiciar a colheita de metadados a partir de tais protocolos, porém normalmente encontra-se algum dificultador, sendo que o desempenho desse tipo de implementação pode ainda variar dependendo da quantidade de registros disponíveis no repositório, a capacidade de hardware disponível no servidor, questões relacionadas a rede, como latência e banda, dentre outros. Desta forma, tal implementação



deve ser analisada caso a caso, de modo a alcançar o desempenho e nível de interoperabilidade desejado.

Frente a esta necessidade de efetuar adaptações, pode ser mais aconselhável, por questões de eficiência e simplicidade, desenvolver uma implementação do protocolo OAI-PMH diretamente sobre a base de dados do catálogo on-line, como pode ser observado no modelo da Figura 33. Esta alternativa possui um dificultador relacionado a possibilidade de acesso a estrutura do banco de dados de sistemas proprietários, além de também se apresentar como uma solução específica.

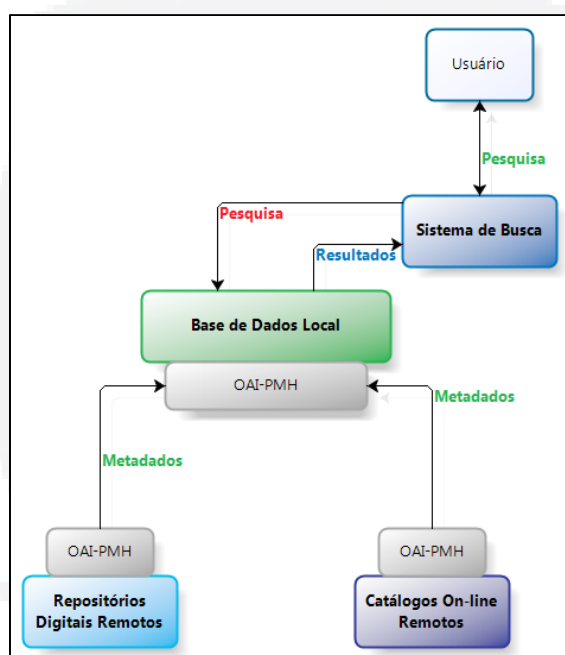


Figura 33 - Modelo da proposta baseada na implementação do protocolo OAI-PMH sobre um catálogo on-line.

Desta forma, devido aos inúmeros dificultadores encontrados na tentativa de possibilitar a colheita de metadados sobre o protocolo Z39.50, SRW e SRU, mudou-se a estratégia em relação à forma de trabalho com tais protocolos, não pela impossibilidade de sua implementação, mas pela especificidade necessária sobre cada repositório que se deseja trabalhar, de modo a tornar relativamente difícil encontrar uma alternativa totalmente genérica de efetuá-la.

A partir disto, foi decidido trabalhar com o Z39.50, SRW e SRU, respeitando suas características técnicas, de modo a efetuar uma implementação padrão de um dos protocolos, porém mantendo o objetivo inicialmente proposto, que consiste na disponibilização de uma busca integrada a partir de tais protocolos.

Dispondo de três alternativas semelhantes, Z39.50, SRW e SRU, buscou-se avaliar qual delas faria parte da implementação inicial, considerando suas características técnicas e recursos disponíveis, já que os mesmos trabalham de forma bastante semelhante. O SRW diferencia-se do SRU basicamente pelo fato de sua implementação ser baseada em *webservices*. O Z39.50, por sua vez, consiste em um protocolo mais antigo, que foi desenvolvido sem levar em consideração as características da *web*, necessitando de maior esforço para sua implementação e utilização. Optou-se então pela implementação do protocolo SRU, pelo fato de ser mais adotado por repositórios abertos, já que o SRW tem sido a opção de repositórios que buscam restringir o acesso aos seus dados.

Esta escolha, por sua vez, ocorre apenas para título de experimentação, já que pode-se ter os três protocolos trabalhando em harmonia, de modo a enriquecer ainda mais a quantidade de registros disponíveis para recuperação.

Partindo destas considerações, foram implementados clientes para os protocolos OAI-PMH e SRU, seguindo o funcionamento determinado por sua especificação, sendo o primeiro responsável pela colheita de metadados de uma ampla gama de repositórios digitais voltados as mais diversas áreas e espalhados pelo mundo, e o segundo, com um direcionamento um pouco menor, pelo fato de ser disponibilizado principalmente por catálogos de bibliotecas e por ser bastante recente.

A Figura 34 apresenta o modelo final adotado pela solução proposta neste estudo, sendo os metadados dos repositórios complacentes com OAI-PMH colhidos pelo agente de *harvesting* implementado no sistema proposto e os registros disponibilizados por meio dos protocolos Z39.50, SRW e SRU, considerando a implementação inicial apenas do SRU, consultados diretamente e em tempo real dos catálogos on-line remotos. O sistema de busca, por sua vez, une os resultados provenientes destas duas macro fontes e os organiza de modo a apresentá-los para o usuário final.

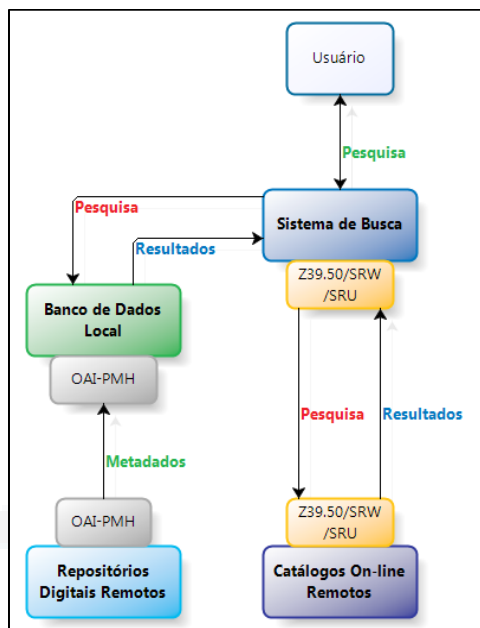


Figura 34 - Modelo do sistema de busca implementado.

A solução adotada, apesar de possuir alguns inconvenientes, demonstrou-se efetiva para a integração em uma busca unificada de repositórios digitais e catálogos on-line que possuem interface disponível dos protocolos OAI-PMH, Z39.50, SRW e SRU, sendo que, pode-se alcançar melhores resultados a partir do refinamento da solução proposta, assim como da integração com os demais protocolos estudados e não implementados no sistema de informação desenvolvido.

## 7.1 Trabalhos futuros

Como trabalhos futuros pode-se realizar diversas melhorias e inclusões de funcionalidades ao sistema desenvolvido, de modo a tornar o sistema mais completo e permitindo sua implantação para utilização em massa:

- Criar um agente de *harvesting* de metadados para trabalhar de acordo com os agendamentos cadastrados na base de dados;
- Otimizar o mecanismo de consulta implementado, com foco nas requisições do protocolo SRU, de modo a reduzir o tempo de busca;
- Implementar um formulário de busca avançada com mais opções de filtros e restrições de busca;

- Implementar de forma completa a linguagem de consulta CQL, como parte dos requisitos para implementação de uma busca avançada;
- Criar uma camada de ontologia utilizando o *framework* RDF, de modo a atribuir maior semântica aos metadados e propiciar a recuperação de resultados com maior inteligência;
- Implementar a integração com o protocolo de interoperabilidade Z39.50 pelo fato deste estar ainda muito presente em catálogos on-line de bibliotecas;
- Implementar a integração com o protocolo de interoperabilidade SRW por ser uma tendência sua maior utilização;
- Disponibilizar ferramentas de relatórios a partir dos metadados coletados assim como dos registros das operações realizadas pelos usuários nas ferramentas de busca;
- Prover aos usuários possibilidade de maior interação com o sistema, disponibilizando a criação de perfis permitindo a definição de preferências, armazenamento de consultas, favoritos, entre outros;
- Refinar a interface do sistema, de modo a torná-lo mais atrativo aos usuários;
- Disponibilizar mais opções de idioma, objetivando dar o sistema maior área de abrangência;
- Possibilitar que os usuários cadastrem citações em relação a determinado registro, de modo a formar uma base de referência para citações.

## REFERÊNCIAS

ALMEIDA, Maurício Barcellos. Uma introdução ao XML, sua utilização na Internet e alguns conceitos complementares. **Ciência da Informação**, Brasília, v. 31, n. 2, p. 5-13, maio/ago. 2002. Disponível em: <<http://revista.ibict.br/ciinf/index.php/ciinf/article/view/140/120>>. Acesso em: 08 ago. 2010.

ALVES, Maria das Dores Rosa; SOUZA, Marcia Izabel Fugisawa. Estudo de correspondência de elementos metadados: Dublin Core e MARC21. **Revista Digital de Biblioteconomia e Ciência da Informação**, Campinas, v. 4, n. 2, p. 20-38, jan./jun. 2007. Disponível em: <<http://www.sbu.unicamp.br/seer/ojs/include/getdoc.php?id=389&article=94&mode=pdf>>. Acesso em: 01 ago. 2010.

ARMS, William Y. et al. A Spectrum of Interoperability: the site for Science for Prototype for the NSDL. **D-Lib Magazine**, v. 8, n. 1, jan. 2002. Disponível em: <<http://www.dlib.org/dlib/january02/arms/01arms.html>>. Acesso em: 27 ago. 2010.

ARMS, William Y. Key Concepts in the Architecture of the Digital Library. **D-Lib Magazine**, jul. 1995. Disponível em: <<http://www.dlib.org/dlib/July95/07arms.html>>. Acesso em: 20 ago. 2010.

BAX, Marcello Peixoto. Introdução às linguagens de marcas. **Ciência da Informação**, Brasília, v. 30, n. 1, p. 32-38, jan./abr. 2001. Disponível em: <<http://revista.ibict.br/ciinf/index.php/ciinf/article/view/221/196>>. Acesso em: 06 ago. 2010.

CAMARGO, Liriane Soares de Araújo de; VIDOTTI, Silvana Ap. Borseti Gregório. Arquitetura da informação para biblioteca digital personalizável. **Encontros Bibli: Revista Eletrônica de Biblioteconomia e Ciência da Informação**, Florianópolis, n. esp., 1º sem. 2006. Disponível em: <<http://www.periodicos.ufsc.br/index.php/eb/article/view/346/389>>. Acesso em: 07 ago. 2010.

CAMPOS, Luiz Fernando de Barros. Metadados digitais: revisão bibliográfica da evolução e tendências por meio de categorias funcionais. **Encontros Bibli: Revista Eletrônica de Biblioteconomia e Ciência da Informação**, Florianópolis, n. 23, jan./jun. 2007. Disponível em: <<http://www.periodicos.ufsc.br/index.php/eb/article/view/318/390>>. Acesso em: 09 ago. 2010.

CASTRO, Fabiano Ferreira de; SANTOS, Plácida Leopoldina Ventura Amorim da Costa. Uso das tecnologias na representação descritiva: o padrão de descrição bibliográfica semântica MarcOnt Initiative nos ambientes informacionais digitais. **Ciência da Informação**, Brasília, v. 38, n. 1, p. 74-85, jan./abr. 2009. Disponível em: <<http://revista.ibict.br/ciinf/index.php/ciinf/article/download/1082/1312>>. Acesso em: 05 ago. 2010.

CRUZ, Angelo Antonio Alves Correa da et al. Impacto dos periódicos eletrônicos em bibliotecas universitárias. **Ciência da Informação**, Brasília, v. 32, n. 2, p. 47-53, maio/ago. 2003. Disponível em: <<http://revista.ibict.br/ciinf/index.php/ciinf/article/view/115/96>>. Acesso em: 06 ago. 2010.

DIGITAL LIBRARY FEDERATION. **A working definition of digital library [1998]. DLF**, 2004. Disponível em: <<http://diglib.org/about/dldefinition.htm>>. Acesso em: 27 ago. 2010.

DUBLIN CORE METADATA INITIATIVE. **Dublin Core Qualifiers (SUPERSEDED, SEE DCMI Metadata Terms)**. 2001. Disponível em: <<http://www.dublincore.org/documents/2000/07/11/dcmes-qualifiers>>. Acesso em: nov. 2010.

DUBLIN CORE METADATA INITIATIVE. **Dublin Core Metadata Element Set, Version 1.1: Reference Description**. 1999. Disponível em: <<http://www.dublincore.org/documents/1999/07/02/dces>>. Acesso em: 16 ago. 2010.

DUVAL, Erik et al. Metadata Principles and Practicalities. **D-Lib Magazine**, v. 8, n. 4, abr. 2002. Disponível em: <<http://www.dlib.org/dlib/april02/weibel/04weibel.html>>. Acesso em: 09 ago. 2010.

DZIEKANIAK, Gisele Vasconcelos et al. Uso do padrão MARC em bibliotecas universitárias da região Sul do Brasil. **Encontros Bibli: Revista Eletrônica de Biblioteconomia e Ciência da Informação**, Florianópolis, n. 26, 2º sem. 2008. Disponível em: <<http://www.periodicos.ufsc.br/index.php/eb/article/view/7198/6645>>. Acesso em: 07 ago. 2010.

DZIEKANIAK, Gisele Vasconcelos; KIRINUS, Josiane Boeira. Web semântica. **Encontros Bibli: Revista Eletrônica de Biblioteconomia e Ciência da Informação**, Florianópolis, n. 18, 2º sem. 2004. Disponível em: <<http://www.periodicos.ufsc.br/index.php/eb/article/view/155/5471>>. Acesso em: 07 ago. 2010.

FERREIRA, Sueli Mara S. P.; SOUTO, Leonardo Fernandes. Dos sistemas de informação federados à federação de bibliotecas digitais. **Revista Brasileira de Biblioteconomia e Documentação**, Nova Série, São Paulo, v. 2, n. 1, p.23-40, jan./jun. 2006. Disponível em: <<http://www.febab.org.br/rbbd/ojs-2.1.1/index.php/rbbd/article/view/3/20>>. Acesso em: 09 ago. 2010.

FERREIRA, Sueli Mara Soares Pinto. Repositório institucional em comunicação: o projeto Reposcom implementado junto à Federação de Bibliotecas Digitais em Ciências da Comunicação. **Encontros Bibli: Revista Eletrônica de Biblioteconomia e Ciência da Informação**, Florianópolis, n. esp., 1º sem. 2007. Disponível em:

<<http://www.periodicos.ufsc.br/index.php/eb/article/view/459/453>>. Acesso em: 09 ago. 2010.

FURRIE, B. **Understanding MARC bibliographic**: machine-readable cataloging. 7 ed. rev. Washington, D. C.: Library of Congress; Follet Software, 2003. Disponível em: <<http://www.loc.gov/marc/umb>>. Acesso em: 12 ago. 2010.

GIL-LEIVA, Isidoro. A indexação na internet. **Brazilian Journal of Information Science**, v. 1, n. 2, p. 47-68, jul./dez. 2007. Disponível em: <<http://www.brapci.ufpr.br/download.php?dd0=10327>>. Acesso em: 02 ago. 2010.

GILLILAND-SWETLAND, Anne. J. **Introduction to metadata**: Setting the stage. 2000. Disponível em: <<http://www.slis.kent.edu/~mzeng/metadata/Gilland.pdf>> . Acesso em: 16 ago. 2010.

GONÇALVES, Edson. **Desenvolvendo aplicações Web com JSP, Servlets, JavaServer Faces, Hibernate, EJB 3 Persistence e AJAX**. Rio de Janeiro: Ciência Moderna, 2007.

GONÇALVES, Maria Carolina. **A indexação em catálogos on-line em bibliotecas universitárias na percepção de usuários integrantes de grupos de pesquisa**: uma contribuição ao desenvolvimento de política de indexação na rede de bibliotecas da UNESP. 2008. 78 f. Dissertação, Marília, 2008. Disponível em: <[http://www.marilia.unesp.br/Home/Pos-Graduacao/CienciadaInformacao/Dissertacoes/goncalves\\_mc\\_me\\_mar.pdf](http://www.marilia.unesp.br/Home/Pos-Graduacao/CienciadaInformacao/Dissertacoes/goncalves_mc_me_mar.pdf)>. Acesso em: 18 nov. 2010.

GOULART, Elias Estevão; HETEM JÚNIOR, Annibal. Pesquisas na Web: estratégias de busca. **Revista Digital de Biblioteconomia e Ciência da Informação**, Campinas, v. 4, n. 2, p. 53-66, jan./jun. 2007. Disponível em: <<http://polaris.bc.unicamp.br/seer/ojs/viewarticle.php?id=92&layout=abstract>>. Acesso em: 09 ago. 2010.

GRANNELL, G. **O guia essencial de web design com CSS e HTML**. Rio de Janeiro: Ciência Moderna, 2009.

GUEDES, Joana Barbosa. **Catálogos online: disponibilização das bibliotecas brasileiras**. 2002. Disponível em: <<http://www.sibi.ufrj.br/snbu/snbu2002/oralpdf/70.a.pdf>>. Acesso em: 18 nov. 2010.

HARVESTING Repository Data and OAI-PMH. 2011. Disponível em: <<http://www.rsp.ac.uk/grow/registration/harvesting/>>. Acesso em: 29 ago. 2011.

JANDL JUNIOR, Peter. **Desenvolvendo aplicações web com JSP e JSTL**. São Paulo: Novatec, 2009.

JISC. **Digital repositories**: helping universities and colleges. 2005. Disponível em: <[http://www.jisc.ac.uk/uploaded\\_documents/JISC-BP-Repository%28HE%29-v1-final.pdf](http://www.jisc.ac.uk/uploaded_documents/JISC-BP-Repository%28HE%29-v1-final.pdf)>. Acesso em: 8 nov. 2011.

KURAMOTO, Hélio. Informação científica: proposta de um novo modelo para o Brasil.

**Ciência da Informação**, Brasília, v. 35 n. 2, maio/ago. 2006. Acesso em: 06 set. 2010. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0100-19652006000200010&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19652006000200010&lng=en&nrm=iso)>. Acesso em: 06 set. 2010.

LAGOZE, Carl et al. (Ed.). **Open archives initiative: object reuse and exchange**. ORE User Guide - Primer. Disponível em: <<http://www.openarchives.org/ore/1.0/primer>>. Acesso em: 13 set. 2010.

LAGOZE, Carl; SOMPEL, Herbert Van de. The Open Archives Initiative: Building a low-barrier interoperability framework. **JCDL**, 01 jun. p. 17-23, 2001, Roanoke, VA. Disponível em: <<http://www.openarchives.org/documents/jcdl2001-oai.pdf>>. Acesso em: 30 ago. 2010.

LIU, Xiaoming et al. Arc: an OAI service provider for digital library federation. **D-Lib Magazine**, v. 7, n. 4, abr. 2001. Disponível em: <<http://www.dlib.org/dlib/april01/liu/04liu.html>>. Acesso em: 27 ago. 2010.

MANOLA, F.; MILLER, E. **RDF Primer. c2004**. Disponível em: <<http://www.w3.org/TR/rdf-primer>>. Acesso em: 23 ago. 2010.

MARCELINO, Silvia Castro. A contribuição da biblioteca para a construção e difusão do conhecimento no Instituto Nacional de Pesquisas Espaciais (Inpe). **Ciência da Informação**, Brasília, v. 38, n. 2, p. 80-95, maio/ago. 2009. Disponível em: <<http://revista.ibict.br/ciinf/index.php/ciinf/article/download/1090/1325>>. Acesso em: 05 ago. 2010.

MARCONDES, Carlos Henrique; SAYÃO, Luís Fernando. Documentos digitais e novas formas de cooperação entre sistemas de informação em C&T. **Ciência da Informação**, Brasília, v. 31, n. 3, p. 42-54, set./dez. 2002. Disponível em: <<http://revista.ibict.br/ciinf/index.php/ciinf/article/view/149/128>>. Acesso em: 06 ago. 2010.

MARCONDES, Carlos Henrique; SAYÃO, Luís Fernando. Integração e interoperabilidade no acesso a recursos informacionais eletrônicos em C&T: a proposta da Biblioteca Digital Brasileira. **Ciência da Informação**, Brasília, v. 30, n. 3, p. 24-33, set./dez. 2001. Disponível em: <<http://revista.ibict.br/ciinf/index.php/ciinf/article/view/190/167>>. Acesso em: 09 ago. 2010.

MARCONDES, Carlos Henrique; SAYÃO, Luís Fernando. O desafio da interoperabilidade e as novas perspectivas para as bibliotecas digitais. **TransInformação**, Campinas, 20(2): 133-148, maio/ago., 2008. Disponível em: <<http://www.brapi.ufpr.br/download.php?dd0=5629>>. Acesso em: 01 ago. 2010.

MARC 21. Tradução e adaptação de Margarida M. Ferreira. Marília: Unesp, 2000. v. 1.

MASLOV, Alexey et al. Adding OAI-ORE Support to Repository Platforms. **Journal of Digital Information**, v. 11, n. 1, 2010. Disponível em: <<http://journals.tdl.org/jodi/article/view/749/640>>. Acesso em: 13 set. 2010.

MARTINS, Ana Bela; RODRIGUES, Eloy; NUNES, Manuela Barreto. **Repositórios de informação e ambientes de aprendizagem**: criação de espaços virtuais para a promoção da



literacia e da responsabilidade social. Disponível em: <<http://www.rbe.min-edu.pt/news/newsletter3/repositorios.pdf>>. Acesso em: 8 nov. 2011.

MILANI, A. Postgresql: guia do programador. São Paulo: Novatec, 2008.

MILLER, Paul. **Interoperability**: what is it and why should I want it. Ariadne, n. 24, jun. 2000. Disponível em: <<http://www.ariadne.ac.uk/issue24/interoperability>>. Acesso em: 20 ago. 2010.

MOREIRA, Walter; RIBEIRO, Thiago. Introdução ao uso dos protocolos SRU/SRW: ferramentas para a catalogação cooperativa. **Perspectivas em Ciência da Informação**, Belo Horizonte, v. 13 n. 3, set./dez. 2008. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1413-99362008000300011&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-99362008000300011&lng=en&nrm=iso)>. Acesso em: 06 set. 2010.

MUELLER, Suzana Pinheiro Machado. A comunicação científica e o movimento de acesso livre ao conhecimento. **Ciência da Informação**, Brasília, v. 35 n. 2, maio/ago. 2006. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0100-19652006000200004&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19652006000200004&lng=en&nrm=iso)>. Acesso em: 06 set. 2010.

NISO. **Z39.50**: A Primer on the Protocol. Bethesda: NISO Press, 2002. Disponível em: <[www.niso.org/publications/press/Z3950\\_primer.pdf](http://www.niso.org/publications/press/Z3950_primer.pdf)>. Acesso em: 11 ago. 2011.

NISO. **Understanding Metadata**. Bethesda, MD: NISO Press, 2004. Disponível em: <<http://www.niso.org/standards/resources/UnderstandingMetadata.pdf>>. Acesso em: 09 ago. 2010.

**OPEN Archives Initiative**. Disponível em: <<http://www.openarchives.org/>>. Acesso em: 30 ago. 2010.

PAVANI, Ana M. B. A produção científica disponível ao mundo: a tecnologia, a vontade e os acessos. **Encontros Bibli**: Revista Eletrônica de Biblioteconomia e Ciência da Informação, Florianópolis, n. esp., 1º sem. 2007. Disponível em: <<http://www.periodicos.ufsc.br/index.php/eb/article/view/464/467>>. Acesso em: 09 ago. 2010.

PEREIRA, A. M.; RIBEIRO JÚNIOR, D. I.; NEVES, G. L. C. Metadados para a descrição de recursos da Internet: as novas tecnologias desenvolvidas para o padrão Dublin Core e sua utilização. **Revista ACB**: Biblioteconomia em Santa Catarina, Florianópolis, v. 10, n. 1, p. 8-39, jan./dez. 2005. Disponível em: <<http://www.acbsc.org.br/revista/ojs/viewarticle.php?id=113&layout=abstract>>. Acesso em: 16 ago. 2010.

PEREIRA, Ana Maria; RIBEIRO JÚNIOR, Divino Ignácio; NEVES, Guilherme Luiz Cintra. Metadados para a descrição de recursos da internet: As novas tecnologias desenvolvidas para o padrão Dublin Core e sua utilização. **Revista ACB**: Biblioteconomia em Santa Catarina, v. 10, n. 1, p. 241-249, jan./dez., 2005. Disponível em: <<http://www.brapci.ufpr.br/download.php?dd0=11157>>. Acesso em: 02 ago. 2010.

ROCHA, Rafael Port da. Metadados, Web Semântica, Categorização Automática: combinando esforços humanos e computacionais para a descoberta e uso dos recursos da web. **Em**

**Questão**, Porto Alegre, v. 10, n. 1, p. 109-121, jan./jun. 2004. Disponível em: <<http://www.brapci.ufpr.br/download.php?dd0=9814>>. Acesso em: 04 ago. 2010.

ROSETTO, Márcia. Uso do Protocolo Z39.50 para recuperação de informação em redes eletrônicas. **Ciência da Informação**, Brasília, v. 26, n. 2, maio/ago. 1997. Disponível em: <<http://revista.ibict.br/ciinf/index.php/ciinf/article/view/391/351>>. Acesso em: 07 ago. 2010.

SANTANCHÈ, André. Aplicações educacionais na Web – o papel de RDF e Metadados. **SBIE**: Rio de Janeiro, nov, 2003. Disponível em: <<http://www.lis.ic.unicamp.br/~santanch/publications/SBIE2003-RDF-LOM.pdf>>. Acesso em: 09 ago. 2010.

SAYÃO, Luís Fernando. Padrões para bibliotecas digitais e interoperáveis. **Encontros Bibli**: Revista Eletrônica de Biblioteconomia e Ciência da Informação, Florianópolis, n. esp., 1º sem. 2007. Disponível em: <<http://www.periodicos.ufsc.br/index.php/eb/article/view/378/436>>. Acesso em: 07 ago. 2010.

SILVA, Maurício Samy. **jQuery**: a biblioteca do programador JavaScript. São Paulo: Novatec, 2010.

SOUZA, Marcia Izabel Fugisawa; ALVES, Maria das Dores Rosa. Representação descritiva e temática de recursos de informação no sistema Agência Embrapa: uso do padrão Dublin Core. **Revista Digital de Biblioteconomia e Ciência da Informação**, Campinas, v. 7, n. 1, p. 190-205, jul./dez. 2009. Disponível em: <<http://polaris.bc.unicamp.br/seer/ojs/viewarticle.php?id=209&layout=abstract>>. Acesso em: 02 ago. 2010.

SOUZA, Marcia Izabel Fugisawa; VENDRUSCULO, Laurimar Gonçalves; MELO, Geane Cristina. Metadados para a descrição de recursos de informação eletrônica: utilização do padrão Dublin Core. **Ciência da Informação**, Brasília, v. 29, n. 1, p. 93-102, jan./abr. 2000. Disponível em: <<http://revista.ibict.br/ciinf/index.php/ciinf/article/view/271/239>>. Acesso em: 07 ago. 2010.

SOUZA, Renato Rocha; ALVARENGA, Lídia. A Web Semântica e suas contribuições para a ciência da informação. **Ciência da Informação**, Brasília, v. 33, n. 1, 2004. Disponível em: <<http://revista.ibict.br/ciinf/index.php/ciinf/article/view/50/49>>. Acesso em: 08 ago. 2010.

SULEMAN, H. **Open digital libraries**. 2002. 190 f. Dissertation, Blacksburg, 2002. Disponível em: <<http://scholar.lib.vt.edu/theses/available/etd-11222002-155624/unrestricted/odl.pdf>>. Acesso em: 27 ago. 2010.

TAYLOR, Mike; DICKMEISS, Adam. Delivering MARC/XML records from the Library of Congress catalogue using the open protocols SRW/U and Z39.50. In: **WORLD LIBRARY AND INFORMATION CONGRESS: IFLA GENERAL CONFERENCE AND COUNCIL**, 71., 2005, Oslo, Norway. **Anais...** Oslo, Norway: IFLA, 2005. p. 1 - 9. Disponível em: <[http://archive.ifla.org/IV/ifla71/papers/065e-Taylor\\_Dickmeiss.pdf](http://archive.ifla.org/IV/ifla71/papers/065e-Taylor_Dickmeiss.pdf)>. Acesso em: 06 set. 2010.

TEMPLE, A. et al. **Jsp, Servlets e J2EE**. c2004. Disponível em:

<<http://www.inf.ufsc.br/~bosco/downloads/livro-jsp-servlets-j2ee.pdf>>. Acesso em: 15 set. 2009.

THE LIBRARY OF CONGRESS. **MARCXML**: MARC21 XML Schema. 2010. Disponível em: <<http://www.loc.gov/standards/marcxml>>. Acesso em: 09 set. 2010.

THE LIBRARY OF CONGRESS. **SRU**: Search/Retrieval via URL. 2008. Disponível em: <<http://www.loc.gov/standards/sru>>. Acesso em: 09 set. 2010.

TRICHKOV, Krasimir. **Standardization Approach for Information Retrieval in WAN**. 2004. Disponível em: <[ecet.ecs.ru.acad.bg/cst04/Docs/sIIB/36.pdf](http://ecet.ecs.ru.acad.bg/cst04/Docs/sIIB/36.pdf)>. Acesso em: 26 ago. 2011.

TROLL, Denise; MOEN, Bill. Report to the DLF on the Z39.50 Implementers' Group. **DLF**: 2001. Disponível em: <<http://www.diglib.org/architectures/zig0012.htm>>. Acesso em: 27 ago. 2010.

WEIBEL, S. et al. **RFC 2413 – Dublin Core Metadata for resource discovery**. 1998. Disponível em: <<http://www.ietf.org/rfc/rfc2413.txt>>. Acesso em: 16 ago. 2010.

WEIBEL, S. The Dublin core: a simple content description model for electronic resources. **Bulletin of the American Society for Information Science**, out./nov. 1997. Disponível em: <[http://findarticles.com/p/articles/mi\\_qa3633/is\\_199710/ai\\_n8767908](http://findarticles.com/p/articles/mi_qa3633/is_199710/ai_n8767908)>. Acesso em: 16 ago. 2010.

WELLMAN, Dan. **JQuery UI 1.6**: the user interface library for JQuery. Birmingham: Packt Publishing, 2009.

WOLF, Alexandre Stürmer; MONTEIRO, Ana Paula Lisboa; VALMORBIDA, Willian. Biblioteca digital da Univates utilizando o software DSpace. **Destaques Acadêmicos**, Lajeado, v. 1, v. 4, 2009. Disponível em: <[http://www.univates.br/files/files/univates/editora/revistas/destaquesacademicos/ano1n4/9\\_biblioteca\\_digital.pdf](http://www.univates.br/files/files/univates/editora/revistas/destaquesacademicos/ano1n4/9_biblioteca_digital.pdf)>. Acesso em: 03 nov. 2010.

WOLF, Alexandre Stürmer; MONTEIRO, Ana P. L.; VALMORBIDA, Willian. Implementação do padrão de metadados MTD2-BR para a ferramenta DSpace visando a interoperabilidade com a Biblioteca Digital Brasileira de Teses e Dissertações (BDTD). In: 8th CONTECSI (International Conference on Information Systems and Tecnology) and 22th World Continuous Auditing and Reporting Systems Symposium, 8., 2011, São Paulo. **Proceedings...** São Paulo, 2011.

WOLF, Alexandre Stürmer; VALMORBIDA, Willian. Desenvolvimento do Portal Rede Aquífero Guarani/Serra Geral para coleta e disseminação do conhecimento. In: 8th CONTECSI (International Conference on Information Systems and Tecnology) and 22th World Continuous Auditing and Reporting Systems Symposium, 8., 2011, São Paulo. **Proceedings...** São Paulo, 2011.

Z39.50. **Z39.50 Text**. Disponível em: <<http://www.loc.gov/z3950/agency/markup/01.html>>. Acesso em: 8 nov. 2011.