

FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
CENTRO UNIVERSITÁRIO “EURÍPIDES DE MARÍLIA” – UNIVEM
PROGRAMA DE MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

DOUGLAS SANCHES DA CUNHA

UMA PROPOSTA DE ARQUITETURA DISTRIBUIDA PARA BIBLIOTECA
DIGITAL DE TESES E DISSERTAÇÕES USANDO ONTOLOGIA

MARÍLIA
2006

DOUGLAS SANCHES DA CUNHA

UMA PROPOSTA DE ARQUITETURA DISTRIBUIDA PARA BIBLIOTECA
DIGITAL DE TESES E DISSERTAÇÕES USANDO ONTOLOGIA

Dissertação apresentada ao Programa de Mestrado do Centro Universitário Eurípides de Marília, mantido pela Fundação de Ensino Eurípides Soares da Rocha, para obtenção do título de Mestre em Ciência da Computação. (Área de Concentração: Arquitetura de Computadores).

Orientador:
Prof. Dr. Marcos Luiz Mucheroni

MARÍLIA
2006

DOUGLAS SANCHES DA CUNHA

UMA PROPOSTA DE ARQUITETURA DISTRIBUIDA PARA BIBLIOTECA
DIGITAL DE TESES E DISSERTAÇÕES USANDO ONTOLOGIA

Banca examinadora da qualificação apresentada ao Programa de Mestrado da UNIVEM, /F.E.E.S.R., para obtenção do Título de Mestre em Ciência da Computação. Área de Concentração: Arquitetura de Computadores.

ORIENTADOR: _____
Prof. Dr. Marcos Luiz Mucheroni

1º EXAMINADOR: _____
Prof. Dra. Silvana Ap. B. Gregorio Vidotti

2º EXAMINADOR: _____
Prof. Dr. Edward Moreno

Marília, _____ de _____ de 2006

CUNHA, Douglas Sanches. **Uma proposta de Arquitetura distribuída para biblioteca digital de teses e dissertações usando ontologia**. 2006, 147f. Dissertação (Mestrado em Ciência da Computação) – Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2006.

RESUMO

Grupos de pesquisa, instituições e universidades estão desenvolvendo bibliotecas digitais mas ainda há muito para ser realizado, incluindo padronização, facilidades de gerenciamento e acesso, incorporação de mecanismos inteligentes de busca. Este trabalho investiga os conceitos da Web semântica e ontologia para implementar uma proposta de construção da base de conhecimento com os conceitos de uma biblioteca digital, usando conceitos e padrões que estão em pleno desenvolvimento, promovida pelo consórcio W3C. Os grupos que pesquisam e implementam estas bibliotecas devem evoluir para uma nova forma de gerenciamento dos documentos físicos e digitais já existentes, uma forma semântica de gerenciamento dos documentos com ontologia e não apenas de dados brutos. Usando as tecnologias de Web Services, foi desenvolvido um web service, formato de dados e ontologias em uma biblioteca chamada DLibOnto, que implementa estes conceitos aplicados a três áreas de pesquisa de Ciência da Computação: Realidade Virtual, Engenharia de Software e Arquitetura de Sistemas Computacionais, sobre estes conceitos foram validados e feita uma análise do sistema implementado DLibOnto.

Palavras-chave: biblioteca digital; internet; web semântica; interoperabilidade; ontologias; xml; web services; java.

CUNHA, Douglas Sanches. **Uma proposta de arquitetura distribuída para biblioteca digital de teses e dissertações usando ontologia.** 2006, 147f. Dissertação (Mestrado em Ciência da Computação) – Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2006.

ABSTRACT

Research groups, institutions and universities, are developing digital libraries but there is still a lot to be done as standardization, facilities for management and access and search intelligent mechanisms. This work investigates the concepts of semantics and ontology web in order to implement a purpose of construction of a knowledge base with the concepts of a digital library, using concepts and standards under development by W3C consortium. The groups which are researching and implementing these libraries must evolutes to a new way of management of the existent physical and digital documents, a semantic way of managing the documents with ontology and not only data. Using the web services technologies, there were developed web services, data format and ontology into a library named DLibOnto, which implements those concepts applied to three research areas of Computer Science: Virtual Reality, Software Engineering and Computer Systems Architecture, for those concepts they were validated and a analysis of DlibOnto was make.

Keywords: digital library; internet; web semantic; interoperability; ontology; xml; web services; java.

LISTA DE FIGURAS

Figura 1. Componentes chave da arquitetura de uma biblioteca digital.....	22
Figura 2. Arquitetura da Biblioteca digital da USP.....	36
Figura 3. Código XML.....	45
Figura 4. Associação de atributos aos elementos em XML.....	46
Figura 5. Documento DTD.....	47
Figura 6. Documento XML de acordo com o DTD.....	48
Figura 7. Modelo RDF.....	49
Figura 8. Declaração RDF.....	50
Figura 9. Sintaxe XML/RDF.....	51
Figura 10. Propriedade com valor estruturado.....	52
Figura 11. XML/RDF com valores estruturados.....	53
Figura 12. Coleção do tipo bag.....	54
Figura 13. Sintaxe XML/RDF tipo bag.....	54
Figura 14. Outro XML/RDF.....	54
Figura 15. Grafo da hierarquia de classes do esquema RDF.....	57
Figura 16. Grafo de um esquema RDF.....	60
Figura 17. Representação do esquema animal em RDF.....	60
Figura 18. As camadas da Web Semântica.....	62

Figura 19. Interação de agentes num sistema multiagentes.....	73
Figura 20. Definição OIL em RDF.....	82
Figura 21. Protege versão 3.1.1.....	83
Figura 22. Comunicação entre Web Services.....	89
Figura 23. Registros, pesquisa e consumo em Web Services.....	91
Figura 24. Envelope SOAP.....	93
Figura 25. Mensagem SOAP.....	94
Figura 26. Registro UDDI.....	98
Figura 27. Arquitetura da DLibOnto.....	106
Figura 28. Estrutura de hardware e conectividade da DLibOnto.....	109
Figura 29. Interfaces da DLibOnto.....	111
Figura 30. Ícones usados na interface da DlibOnto.....	112
Figura 31. Ontologia da DLibOnto.....	115
Figura 32. Modelo de negócio da DLibOnto.....	122
Figura 33. Diagrama de interfaces da DLibOnto.....	123
Figura 34. Diagrama de classes da DLibOnto.....	124
Figura 35. Diagrama de classes da DLibOnto.....	125
Figura 36. Diagrama de seqüência da página principal.....	125
Figura 37. Diagrama de seqüência da consulta de documentos.....	126

Figura 38. Diagrama de seqüência de cadastramento de usuário.....	126
Figura 39. Diagrama de seqüência de cadastro de documentos.....	127
Figura 40. Diagrama de seqüência para criar o XML.....	127
Figura 41. Diagrama de seqüência de envio de arquivo completo.....	128

LISTA DE TABELAS

Tabela 6.1. Taxonomia da Ciência da Computação.....	115
Tabela 6.2. Taxonomia da JBDigital.....	117
Tabela 6.3. Padrão Brasileiro de metadados de Teses e Dissertações	119

LISTA DE ABREVIATURAS E SIGLAS

BDB: Biblioteca Digital Brasileira

DCMI: Dublin Core Metadata Initiative

DHCP: Dynamic Host Configuration Protocol

DLF: Digital Library Federation

DLibOnto: Digital Library Ontology

HTML: Hyper Text Markup Language

IBICT: Instituto Brasileiro de Informação em Ciência e Tecnologia

JBDigital: Java Biblioteca Digital

JSP: Java Server Pages

OAI - PMH: Open Archives Initiative Protocol for Metadata Harvesting

RDF: Resource Description Framework

SOAP: Simple Object Access Protocol

UDDI: Universal Description, Discovery and Integration directory

URI: Uniform Resource Identifier

URL: Uniform Resource Locator

WSDL: Web Services Description Language

XML: Extensive Markup Language

SUMÁRIO

1. INTRODUÇÃO.....	14
2. BIBLIOTECAS DIGITAIS.....	19
2.1. Conceitos e definições das bibliotecas digitais.....	21
2.2. Breve histórico das bibliotecas digitais.....	26
2.3. Projetos de bibliotecas digitais.....	28
2.3.1. Biblioteca digital de Alexandria.....	29
2.3.2. Biblioteca digital de Standford.....	31
2.3.3. Biblioteca digital de vídeo Informedia.....	31
2.4. Iniciativa da OAI.....	32
2.5. Iniciativas nacionais.....	33
3. WEB SEMÂNTICA.....	38
3.1. Arquiteturas de metadados.....	39
3.1.1. Metadado embutido distribuído.....	41
3.1.2. Metadado externo centralizado.....	42
3.1.3. Metadado externo distribuído.....	43
3.2. Linguagem XML.....	44
3.2.1. Característica de linguagem.....	45
3.2.2. Documentos bem-formatados.....	46

3.3. Modelo RDF.....	48
3.3.1. Sintaxe RDF.....	51
3.4. Esquema RDF.....	56
3.5. Mecanismos de consulta em RDF.....	59
4. ONTOLOGIAS E WEB SEMÂNTICA.....	62
4.1. Representação do conhecimento.....	63
4.2. Conceito de ontologia.....	66
4.2.1. Elemento de representação comum: <i>Frames</i>	69
4.3. Agentes de software e agentes inteligentes.....	72
4.4. Interoperabilidade.....	74
4.4.1. A iniciativa Dublin Core.....	76
4.4.2. Protocolo OAI-PMH.....	78
4.5. Linguagens e ferramentas para criação de ontologias.....	79
4.5.1. Ferramenta Protégé.....	82
5. WEB SERVICES E ARQUITETURAS DISTRIBUÍDAS.....	87
5.1. Web Services.....	87
5.2. Protocolo SOAP.....	92
5.3. Linguagem WSDL.....	94
5.4. Registro UDDI.....	96

5.5. Tecnologia Java Server Pages.....	100
5.6. Computação distribuída e componente.....	102
6. BIBLIOTECA DIGITAL DLibOnto.....	104
6.1. Arquitetura Web distribuída da DLibOnto.....	105
6.2. Web Services da DLibOnto.....	107
6.3. Definição dos serviços da DLibOnto.....	110
6.4. Ontologia da DLibOnto.....	114
6.5. Análise da DLibOnto.....	121
7. CONCLUSÃO.....	129
Referencias Bibliográficas.	132
APÉNDICES.....	140

1. INTRODUÇÃO

A sociedade tem passado por profundas transformações e os diversos estudos e pesquisas sobre o desenvolvimento econômico e social indicam que, dos três fatores básicos de produção – o trabalho, o capital e o conhecimento, atualmente o principal é o conhecimento. A importância relativa entre eles, vem historicamente deslocando-se do primeiro, passando pelo segundo e concentrando-se nos últimos tempos no terceiro – o conhecimento, que se apóia em amplas e instantâneas infra-estruturas de informação.

O problema é tecnológico e sociológico porque ao iniciar essa análise depara-se com três problemas fundamentais: a distância física, a distância social e a distância temporal (RUGGLES, 1996).

Em consequência destas transformações, o meio digital passou a ser um espaço sem precedentes para a representação, armazenamento e recuperação de informações no contexto geral. Conseqüentemente, gerando discussões e estudos que promovem o desenvolvimento das bases de dados, tanto no que diz respeito aos processos de tecnologia da informação como sua gestão. Esses processos envolvem conhecimento e definição de arquiteturas de hardware e software, armazenamento, protocolos, recuperação de informação, padrão para descrição de objetos, elementos de metadados, entre outros, que garantam a integração das bases.

Uma economia baseada na informação começa a despontar nos negócios com investimentos globais. As instituições privadas e públicas estão competindo por meio de redes eletrônicas com mais agilidade. O crescente volume de informações faz com que as organizações responsáveis pelo armazenamento, gerenciamento e recuperação criem mecanismos, para possibilitar o uso dessa grande “massa de dados”, agregando novas arquiteturas e tecnologias de automação, proporcionando a interoperabilidade das informações com acesso on-line.

Não parece possível aumentar a quantidade e a qualidade da informação digital disponível sem que os que participam no processo da sua produção e distribuição (autores, editores, colaboradores, bibliotecas e etc.) recebam uma contribuição, seja ela financeira ou referência adequada e correspondente pela disseminação do conhecimento (LANCASTER, 1994).

A World Wide Web - WWW é hoje considerada a maior fonte de disseminação de informação nas principais áreas de conhecimento. O seu uso intensivo (estimada em mais de milhões de usuários em todo mundo) aliados ao seu crescimento exponencial (estima-se em torno de alguns bilhões de páginas atualmente disponíveis na rede) tem mudado drasticamente o comportamento da sociedade que dispõe atualmente de grande variedade de serviços e informações (comércio, turismo, notícias, bibliotecas etc.), simplificando e dinamizando muitas tarefas diárias.

Reunir pessoas com experiência e conhecimentos diferentes é uma das condições necessárias à criação de conhecimento. Aparentemente, tal afirmação parece bastante simples e prática. No entanto, a troca de experiência e conhecimentos entre pesquisadores e os profissionais, não é simples, principalmente devido às estruturas internas das organizações (NONAKA, TAKEUCHI, 1998).

Apesar do sucesso deste crescimento exponencial têm-se dificultado muito a localização, acesso, apresentação, recuperação e manutenção da informação para uso de um número tão grande de usuários. O compartilhamento de recursos distribuídos, autônomos e heterogêneos é disponibilizado, na maioria das vezes, sem a mínima padronização, que evidencia a problemática desta pesquisa.

A insatisfação dos usuários com relação às buscas na Internet está relacionada, sobretudo à demora na obtenção dos resultados e à recuperação de uma elevada quantidade de informações que, na maioria das vezes, não atendem às necessidades dos usuários. As restrições impostas pelas instituições ao acesso, somente aos usuários registrados. Em geral, a

apresentação dos documentos não segue padrão algum de conhecimento que possa ser apresentado, sem qualquer padrão de formatação, quando foi sua última atualização, sem qualquer referência sobre o autor ou fonte para validação sobre o conteúdo do documento.

Esta dissertação tem como objetivo estudar os métodos adotados por outras instituições que já construíram suas bibliotecas digitais. Pesquisar e avaliar as tecnologias e ferramentas que foram utilizadas nestas constituições, verificar também o contexto de uma biblioteca digital distribuída. Uma investigação sobre bibliotecas digitais do âmbito conceitual até o prático, levantando os recursos e requisitos básicos de sua gestão. Investigar algumas tecnologias que poderão agregar valores de funcionamento prático e arquiteturas computacionais distribuídas menos complexas. Este trabalho visa também, contribuir para a criação de uma biblioteca digital distribuída com código aberto (*Open Source*) usando Java.

Além das pesquisas e avaliações sobre as bibliotecas digitais já constituídas, investigar os conceitos da web semântica, assim como as tecnologias que poderão ser utilizadas para elaborar uma ontologia para uma biblioteca digital. Desenvolver uma arquitetura para biblioteca digital distribuída de teses e dissertações, utilizando os conceitos e tecnologias da web semântica usando ontologias para produzir um sistema web para ser utilizada na Internet por grupos de pesquisas, universidades e instituições que tem como objetivo a disseminação do conhecimento de suas pesquisas e artigos científicos publicados.

As principais bibliotecas digitais já implantadas têm-se ótima organização. O comum a todas elas é a ênfase colocada no acesso remoto ao conteúdo e os serviços das bibliotecas como fonte de informação. Possibilidade de reproduzir, mostrar e ampliar os serviços das bibliotecas tradicionais, aproveitando as potencialidades do armazenamento e comunicações digitais. Proporcionando um serviço mais dinâmico, com interfaces mais amigáveis aos usuários, promovendo o acesso aos documentos em formatos multimídia, (SANCHES and VEJA VALDES, 2004).

Usando a literatura revista foi possível justificar que a Web é um desafio para a comunidade científica na busca integrada, por não existir interoperabilidade semântica nas documentações. Tal integração engloba: o gerenciamento dos recursos, envolvendo a avaliação do conteúdo de seus relacionamentos; e a padronização dos recursos por meio da descrição de suas propriedades, arquiteturas e implementações de mecanismos que irão oferecer suporte à descoberta e recuperação destas documentações.

Várias iniciativas, como as desenvolvidas pelo W3C (2004), buscam por intermédio da criação de padrões, arquiteturas de metadados, serviços de inferência e ontologias, dentre outras, a melhor forma de tornar a informação também compreensível pela máquina.

Para que as bibliotecas digitais possam ser utilizadas com eficiência é necessário que existam as infra-estruturas que suportem o acesso e a difusão de seus serviços. É previsível que seus serviços, devido ao tipo de informação (digital e multimídia) que os integra, o elevado grau de distribuição no seu funcionamento, originem um grande volume de tráfego nas redes de comunicação. Assim, infra-estruturas de rede com maior largura de banda, velocidade, arquiteturas, topologia e protocolos de rede, ambientes distribuídos, técnicas de compressão e armazenamento dos dados etc., são condições prévias a existência de verdadeiras bibliotecas digitais.

Este trabalho desenvolve uma Biblioteca Digital contendo uma ontologia utilizando a linguagem Java e a ferramenta de ontologias Protégé, chamada de DLibOnto.

No segundo capítulo, é apresentado um breve histórico do desenvolvimento do homem com relação às bibliotecas, alguns conceitos considerados chave para as bibliotecas digitais, alguns dos principais projetos de bibliotecas digitais existentes em funcionamento e as iniciativas nacionais.

No terceiro capítulo, os conceitos sobre a Web Semântica, abordando as arquiteturas de metadados, apresentação da linguagem *eXtensive Markup Language* - XML, o modelo e o

esquema *Resource Description Framework* – RDF e seus mecanismos apropriados de consulta a este esquemas e modelos.

No capítulo seguinte é descrita a ontologia digital: seu significado e sua contribuição para a Web Semântica, na representação do conhecimento. A forma como que os agentes de software inteligentes poderão promover a interoperabilidade de informações entre os Web Services. As ferramentas e as linguagens que podem auxiliar na criação das ontologias, e a apresentação da ferramenta Protégé que foi utilizada para a elaboração da ontologia da biblioteca digital DLibOnto.

No quinto capítulo, os principais conceitos sobre a arquitetura distribuída dos Web Services. Protocolo, linguagem, registro e tecnologias que podem ser agregadas para uma composição segura, estável e dinâmica de quaisquer serviços web. As tecnologias Java que poderão ser utilizadas, para favorecer uma possível computação distribuída baseada em componentes.

No último capítulo está a descrição completa do projeto da arquitetura da biblioteca digital distribuída usando ontologia DLibOnto. Apresentando as ontologias que foram criadas de modo a atender as necessidades semânticas da Internet, de forma a promover a interoperabilidade semântica entre os web services.

2. BIBLIOTECAS DIGITAIS

A história intelectual do homem tem sido marcada por constante busca no desenvolvimento dos meios de registro e de organização de sua vida material e espiritual de existência: registro em papiro, tabula de argila, pergaminho, papel, caracteres eletrônicos; organização acessada por interfaces grafadas, manuscritas, impressas, digitais, analógicas, que traduz, captura, codifica o sentido da obra individual ou grupal para uso coletivo.

A configuração de uma nova tecnologia intelectual pelo surgimento da Internet, da qual passam a se beneficiar milhares de instituições com seus milhões de usuários, permite um novo campo de possibilidades à cultura. Seus efeitos, inimagináveis e imprevisíveis, dependem, porém, da associação dos atores que a exploram em redes humanas e não humanas (máquinas, dispositivos eletrônicos), fundadas no engenho, no talento e na paixão para prover interfaces mais amigáveis, que possibilitem o acesso a todo o conhecimento concebido, explorado, descoberto e inventado até o presente momento.

Buscar informações espalhadas na rede é um processo de tentativa e erro. Este processo de busca e acesso às informações foi denominado de navegação. Cada usuário da Web segue seu próprio rumo e toma diversos atalhos. Mas optar por permanecer “à deriva” não é uma opção certa. Procura-se, então, criar faróis, sinalizadores, mapas e bússolas, que possam servir de guia nesse oceano informacional. Como os velhos marinheiros, aventureiros que saíam em busca do desconhecido, com informações imprecisas sobre a rota a ser percorrida e o desejo em busca do novo. Ao navegarem, ao descobrirem, criavam e alimentavam os mapas. O início das navegações, do “surfear” na Internet, também é caracterizado pela absoluta imprecisão. Isso sem considerar na falta de informações em outra língua que não o inglês, uma vez que o que se via nos primórdios da Internet era, praticamente, *sites* somente em inglês.

Esses instrumentos de busca ajudam na descoberta dos *sites*, mas também se constituem em elementos auxiliares para se perseguir o ideal de racionalização do conteúdo disponível na rede. As chamadas bibliotecas digitais são criadas, então, como mais um passo na busca de facilitar essas navegações e podem ser identificadas como "*a pequena frota de pequenas arcas, botes ou sampanas*" segundo LÉVY (1999).

Com o desenvolvimento tecnológico alcançado na década de noventa proporcionou a derrubada dos obstáculos que impediam a popularização e o desenvolvimento adequado das bibliotecas digitais. Dentre os quais se podem citar: a) Redução do custo no armazenamento eletrônico; b) Recursos gráficos para apresentação mais agradáveis; c) Popularização dos computadores pessoais; d) Redes de computadores com maior velocidade na comunicação e com acesso a longas distâncias; e) Aumento no poder de processamento de computadores pessoais. Desta forma as bibliotecas passaram a utilizar os recursos oferecidos pelas tecnologias da computação.

Não é correto afirmar que as bibliotecas digitais são somente versões eletrônicas das bibliotecas tradicionais. Além de agregar novos conceitos e incorporar novos serviços e capacidades, um público diferente do que era atingido pelas tradicionais é servido pelas bibliotecas digitais. Parte destas pessoas não teria acesso às informações disponibilizadas pela biblioteca devido as suas limitações geográficas. Outra parte pode ser constituída por jovens pesquisadores que já nasceram na era da informação digital e que por comodidade dão preferência à manipulação e pesquisa de recursos digitais.

A seguir serão apresentados os conceitos básicos das bibliotecas digitais e a evolução cronológica pela classificação de gerações.

Com relação às bibliotecas tradicionais, as digitais ampliam o acesso dos usuários à medida que permite acesso, permanente aos dados, enquanto que a maioria das bibliotecas está fechada, quer seja durante a noite, em feriados ou finais de semana; ou mesmo quando se está a vários quilômetros de distância da biblioteca propriamente dita.

É importante ressaltar que as bibliotecas digitais devem ser entendidas como uma evolução qualitativa e quantitativa das bibliotecas tradicionais. A possibilidade de armazenar grande quantidade de informação em espaços reduzidos e do compartilhamento de coleções entre bibliotecas geograficamente distribuídas, indica o aumento quantitativo do acervo disponibilizado. A diversidade de formatos (multimídia) que pode ser oferecida, assim como novos tipos de serviços, caracteriza o aspecto qualitativo desta evolução.

2.1. Conceitos e definições das bibliotecas digitais

Algumas pessoas vêem bibliotecas digitais simplesmente como uma coleção de informações processáveis pelo computador ou um repositório para tais informações. Outros vêem a Internet inteira como uma biblioteca digital gigante, uma definição aceita é:

As bibliotecas digitais são as organizações que fornecem os recursos, incluindo a equipe especializada, para selecionar, estruturar, oferecer o acesso intelectual, interpretar, distribuir, preservar a integridade e assegurar a persistência integral das coleções de trabalhos digitais, de modo que estejam prontamente e economicamente disponíveis para o uso por uma comunidade definida ou conjunto de comunidades. (WATERS, 1998).

Pesquisadores das bibliotecas digitais sabem que há muitos conceitos importantes que devem ser considerados. Na figura 1, a seguir (ARMS, 2000), mostra alguns componentes que são usados em uma biblioteca digital. Eles têm três funções principais: ajudar os usuários a interagirem com a biblioteca digital, armazenar coleções de materiais e fornecer serviços.

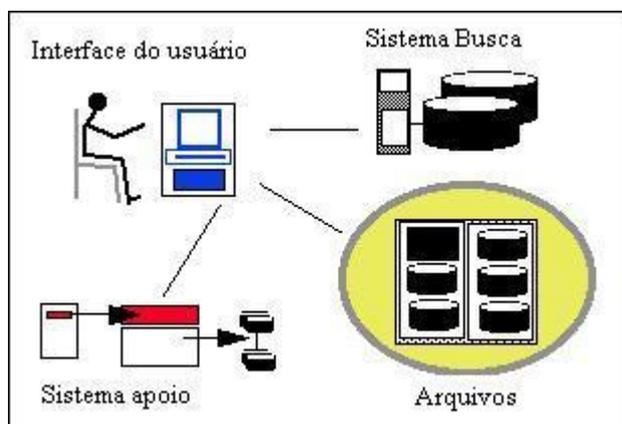


Figura 1 – Componentes chaves da arquitetura de uma biblioteca digital (ARMS, 2000)

Na terminologia de computação, alguém que interage com um computador é um “usuário”. Este termo engloba criadores, usuários de biblioteca, profissionais da informação e qualquer outro que acesse um computador. O computador utilizado para acessar uma biblioteca digital é chamado de “cliente”. Algumas vezes, clientes podem interagir com uma biblioteca digital sem o envolvimento de um ser humano. Muitos destes clientes são robôs que indexam automaticamente as coleções de uma biblioteca digital, por exemplo.

Dois “serviços” típicos fornecidos por bibliotecas digitais são: “serviços de busca”, que fornece catálogos, índices e outros serviços que auxiliam os usuários a encontrar informações; e “serviços de localização”, que são usados para identificar e localizar informação. Mas ainda sem qualquer órgão que as regule como padronização.

A criação da federação a *Digital Library Federation* - DLF (URL: <http://www.diglib.org/about/dldefinition.htm>), apresentou uma regulamentação importante ao contexto das bibliotecas digitais.

As instituições que fazem parte da DLF sentiram, no curso do desenvolvimento de seus programas, a necessidade de uma compreensão comum do que são bibliotecas digitais para que pudessem alcançar o objetivo eficaz de “federá-las”.

Outras definições focalizam em uma ou mais características incluídas na definição da DLF. Por exemplo, o termo “biblioteca digital” pode descrever simplesmente a noção da “coleção”, sem referência a sua organização, acessibilidade intelectual ou atributos de

serviços. Este é o sentido particular que é visto na Internet descrita como uma grande biblioteca digital. Mas as palavras podem se referir também à organização subjacente a coleção ou, mesmo mais especificamente, ao sistema de computação em que a coleção reside. O último sentido está mais claramente em uso na *National Science Foundation's Digital Library Initiative* - NSFDLI (URL: <http://www.dli2.nsf.gov>).

Para a DLF, “biblioteca digital” faz mais do que simplesmente enumerar características. Serve também como base para a perspectiva da DLF no objetivo das bibliotecas digitais e nas exigências funcionais para seu desenvolvimento. As breves considerações de determinadas características da definição ajudarão a explicar seu significado para a DLF. As seguintes características devem ser consideradas.

- As bibliotecas digitais são organizações que empregam e indicam uma variedade de recursos, especialmente os recursos intelectuais inerentes na equipe especializada, mas não necessitam ser organizadas no modelo das bibliotecas.
- Elas também devem preservar a integridade e a persistência. Cada uma das funções enumeradas na definição de “bibliotecas digitais” - selecionar, estruturar, oferecer o acesso intelectual, interpretar, distribuir, preservar a integridade, e assegurar a persistência - é tema aos especiais confinamentos e exigências de operar em uma rápida evolução eletrônica e ambiente da rede. A mudança contínua no ambiente significa que as últimas duas funções – assegurarem a persistência e a preservação de integridade - são especialmente difíceis de alcançar. Mas a DLF considera estas funções como centrais ao conceito de “biblioteca digital”.

As distinções entre as bibliotecas geralmente estão no foco do tema da matéria que define as coleções (por exemplo, medicina, arte, ciência, música, e outros), ou nas comunidades interessadas nos materiais coletados (por exemplo, pesquisa, faculdade,

público). A DLF está convencida que, com o amadurecimento das bibliotecas digitais, o princípio que define as políticas de suas coleções não é a “digitalidade” do material.

As bibliotecas em geral e as bibliotecas digitais particularmente, são organizações de serviços. As necessidades e os interesses das comunidades que elas servem determinarão finalmente a trajetória do desenvolvimento das bibliotecas digitais, incluindo o investimento feito no conteúdo e na tecnologia. A maioria das bibliotecas para a DLF é dedicada ao suporte à educação e pesquisa, e justificam seu investimento em desenvolvimentos digitais como um meio poderoso de realizar os objetivos institucionais das comunidades acadêmicas que elas servem.

- Para utilizar uma biblioteca, um leitor deve ir até ela. Porém, isto não é possível para a maioria das pessoas que necessitam usá-las. Uma biblioteca digital traz a informação ao usuário, sendo que o leitor nunca necessitará visitar uma biblioteca fisicamente. Em uma biblioteca digital basta um computador pessoal e uma conexão na rede.
- Os documentos em papel são convenientes para leitura, mas encontrar informações desejáveis em papel é muito difícil. Em muitos aspectos, os sistemas de computação são melhores que os sistemas e métodos manuais de busca de informação. Os sistemas de informação são particularmente úteis para trabalhos de referências que envolvem saltos de uma fonte de informação para outra.
- As bibliotecas possuem muitas informações que são únicas. Quando se colocam informações digitais sobre uma rede, elas são disponibilizadas para todos. Muitas bibliotecas digitais ou publicações eletrônicas são mantidas num local central, com poucas cópias duplicadas estrategicamente no mundo. Isto é uma grande vantagem em relação ao gasto excessivo de espaço físico para duplicação de material nas bibliotecas tradicionais.

- Muitas informações necessitam serem atualizadas constantemente. Materiais impressos são extremamente complicados de serem atualizados. Quando são, devem ser reimpressos por inteiro. Manter a informação atualizada é menos trabalhoso quando a versão definitiva está em formato digital e armazenada em um computador central. Muitas bibliotecas digitais mantêm versões on-line de diretórios, enciclopédias e outros trabalhos de referência.
- Na verdade, a porta de uma biblioteca digital nunca se fecha. O escopo das coleções expande os muros das bibliotecas tradicionais. Artigos privados em um escritório ou em uma biblioteca do outro lado do mundo estão à disposição facilmente através de uma biblioteca digital. Não quer dizer que bibliotecas digitais não falhem ou que a rede não entre em colapso, porém, comparando-se com as bibliotecas tradicionais, as bibliotecas digitais estão muitas mais disponíveis no momento e local que o usuário deseja.
- Um dos principais benefícios do uso de bibliotecas digitais está na economia financeira. Ainda hoje este custo não está bem definido. Entretanto, bibliotecas convencionais são muito caras. Elas ocupam construções caras sobre locais privilegiados. Grandes bibliotecas empregam centenas de pessoas. As bibliotecas não possuem recursos suficientes para adquirir e processar todo material que elas desejam. A publicação é algo bastante oneroso. As bibliotecas digitais, atualmente podem ter custos altos, isto é, mais caras que bibliotecas convencionais na fase de implantação. Porém, tudo que compõe uma biblioteca digital está cada dia mais barato. Como o custo da tecnologia que envolve uma biblioteca digital está caindo continuamente, o custo da biblioteca digital está se tornando cada vez menor. Entre outras coisas, os custos com distribuição e armazenamento de informação digital estão caindo gradativamente.

2.2. Breve histórico das bibliotecas digitais

Talvez por haver poucas comunidades e muito diferentes envolvidas na pesquisa, no projeto e na execução de bibliotecas digitais, existem algumas idéias diferentes sobre o que é uma biblioteca digital. Os bibliotecários e os cientistas da computação são os dois grupos principais que estão trabalhando em prol das bibliotecas digitais. Portanto, é relevante relatar as bibliotecas no âmbito das gerações, para melhor caracterizá-las, conforme descrito por (BAPTISTA, 2000).

Nas bibliotecas da primeira geração, todo o serviço é realizado sem a utilização de sistemas computacionais de informação. O acervo da biblioteca é catalogado manualmente em fichas ou cartões. A pesquisa é feita de modo manual. Os usuários devem preencher cartões em que estarão contidas suas informações pessoais, e informações sobre seus empréstimos e reservas. Os recursos disponibilizados pelas bibliotecas tradicionais geralmente são os seguintes: livros, revistas, jornais, jogos, mapas, material de áudio e vídeo. Os serviços oferecidos por elas são: empréstimo, reservas, pesquisas e a estrutura para o acesso físico ao acervo.

Questões sobre propriedade intelectual dos recursos são claramente definidas nas leis de *copyright* específicas para bibliotecas. Para as bibliotecas tradicionais, as noções de empréstimo e de reserva são importantes, pois os recursos são físicos e limitados.

Na segunda geração, as bibliotecas utilizam sistemas computacionais específicos para a informatização de seus serviços. Os cartões utilizados nas bibliotecas tradicionais são convertidos para meio eletrônico.

Há pacotes de software com o propósito de informatizar a indexação e pesquisa no acervo, largamente utilizada por bibliotecas em todo o mundo, como por exemplo, a *Open Public Access Library* - OPAC. Há uma preocupação com a produção de metadados e, mais

ainda, para que seja estabelecido um padrão na produção de metadados. Isto permite que tais sistemas possam interoperar com outros sistemas do mesmo gênero.

Na terceira geração as bibliotecas digitais agregam mais uma categoria de serviço às bibliotecas tradicionais. Além da busca e indexação, apresenta-se agora a recuperação da informação. Isto é possível, pois nesta geração o acervo da biblioteca é digital. Assim, após uma busca, a informação é recuperada a partir da própria máquina do cliente. Este tipo de sistema constrói uma idéia de biblioteca bastante distinta daquela apresentada nas gerações anteriores.

A quarta geração de bibliotecas agrega a terceira, a diferentes tipos de recursos tais como: áudio, vídeo, mapas e imagens, além dos documentos tipo texto. A estes diversos formatos no qual a informação pode se apresentar, dá-se o nome de multimídia. O processo de busca e indexação de informação multimídia é diferente do realizado para atributos de metadados textuais. Para as informações multimídia, utiliza-se a técnica de recuperação baseada em conteúdo, a ser detalhada mais adiante. Nesta abordagem, o processo de busca é diretamente dependente do tipo de informação que está sendo manipulado: imagem, som, texto e vídeo.

As iniciativas mais importantes para a construção de bibliotecas digitais tiveram raízes nos Estados Unidos. Em 1994, foi lançado um grande projeto, o Digital Library Initiative - DLI, envolvendo seis grupos de pesquisa nas seguintes universidades americanas: Universidade da Califórnia (Berkeley e Santa Barbara), Universidade de *Stanford*, Universidade de *Illinois* e Universidade *Carnegie-Mellon*. O projeto teve como objetivo buscar a evolução dos métodos de captação, armazenamento e organização da informação em formato digital e torná-los disponíveis para busca, recuperação e processamento por meio de redes de comunicação de forma amigável para o usuário. Ele foi financiado por alguns órgãos do governo americano sob a coordenação da *National Science Foundation* - NSF. Este projeto encontra-se em sua segunda fase, envolvendo várias universidades e com um foco de atuação

bem mais abrangente que o seu antecessor preocupando-se com os efeitos sociais trazidos pelas bibliotecas digitais em várias áreas. (URL: <http://www.dli2.nsf.gov>)

Atualmente, diversos sistemas de bibliotecas digitais estão em funcionamento pelo mundo. Bons exemplos são as disponibilizadas via Web pela *Association for Computer Machinery* - ACM, disponível (URL: <http://www.acm.org/dl/>) e pelo *Institute of Eletrical and Eletronic Engineers* - IEEE, disponível (URL <http://computer.org/dlsearch.htm>). O acervo de ambas, restrito a documentos tipo texto, é formado por artigos científicos publicados em conferências e revistas destas instituições. Aos usuários, são oferecidos serviços de navegação em toda a coleção, utilizando palavras-chave e restrições de campo. A cobrança pela utilização do material é feita por meio de uma assinatura anual.

2.3. Projetos de bibliotecas digitais

O tamanho e a variedade de suas coleções fazem da Biblioteca de Congresso (Library of Congress) dos EUA a maior biblioteca no mundo. Englobam milhões de itens virtualmente, de todos os formatos, idiomas, e assuntos. Estas coleções são em sua maioria, selecionadas da acumulação geral da expressão humana. As coleções são amplas no escopo, englobando materiais de pesquisa em vários idiomas, muitos ainda manuscritos em diferentes tipos de mídia. Disponível em (URL: <http://www.loc.gov/index.html>)

Dos milhões de livros, fotografias, jornais, desenhos, manuscritos, livros raros, mapas, gravações de som e quadros móveis mantidos pela Biblioteca, somente uma pequena fração está na forma digital. Um dos principais componentes do programa de digitalização desta Biblioteca o *Americam Memory*, oferece coleções multimídia de documentos digitalizados, fotografias, gravações de som, filmes e texto das coleções da Biblioteca Americana. Por intermédio de uma concessão da Fundação *Ameritech Digital Library Competition* - ADLC, que permite a pesquisa pública nas bibliotecas acadêmicas, museus,

sociedades históricas, instituições arquivadas; e criam as coleções digitais de material de recurso primário para complementar o programa. A Biblioteca também coopera internacionalmente para colecionar leis digitalizadas, regulamentos, e outras fontes legais complementares no projeto *Global Legal Information Network - GLIN8* e a Rede Mundial de Informações Jurídicas – RMIJ.

2.3.1. Biblioteca digital Alexandria

Um projeto de projeção internacional é o da Biblioteca de Alexandria. Este projeto iniciou seus trabalhos em 1995, com a organização da biblioteca digital por meio de coleções de materiais e serviços geograficamente referenciados a fim de se obter acesso a essas coleções. O Projeto Biblioteca Digital Alexandria – ADL, que está alocado no campus da Universidade de Califórnia em Santa Bárbara, (*Alexandria Digital Earth Prototype*), disponível na (URL: <http://alexandria.sdc.ucsb.edu/>). É um consórcio de pesquisadores, desenvolvedores e educadores, transpondo os setores acadêmicos, públicos e privados, explorando uma variedade de problemas relacionados a uma biblioteca digital distribuída para informações geograficamente referenciadas. Por meio de redes de forma que os componentes da biblioteca podem ser distribuídos pela Internet, como também coexistindo em um único computador. Os meios geograficamente referenciados foram associados a todos os objetos na biblioteca com uma ou mais regiões.

O núcleo do Projeto Alexandria é um sistema de informação disponível em tempo real e *on-line* inspirado pelo Laboratório de Mapas e Imagem, *Map and Image Laboratory - MIL*, na Biblioteca de Davidson, na Universidade de Califórnia, Santa Bárbara. Ela provê acesso atualmente pela Internet para um subconjunto das propriedades, como também outros conjuntos de dados geográficos.

A arquitetura do sistema consiste em um componente de interface de usuário que suporta acesso amigável aos serviços da biblioteca por meio de uma combinação de linguagens textuais, visuais. Tem grande capacidade de navegação baseada em partições hierárquicas dos dados. Um componente de catálogo no qual são empregadas estruturas de índice e metadados provendo resposta rápida para as pesquisas dos usuários envolvendo busca de conteúdos. Tem um componente para armazenamento que provê acesso de alta velocidade para grandes coleções de materiais indexados regionalmente;

Há outro componente que permite aos bibliotecários e gerentes de sistemas a incorporação de novos artigos na coleção da biblioteca usando procedimentos que incluem digitalização, (re) formatação e extração de informações de catálogo. Cada componente tem uma interface para a rede que fornece protocolos para comunicação. O sistema é composto de subsistemas que são distribuídos para vários *sites*.

2.3.2. Biblioteca digital Stanford

A meta do projeto da Biblioteca Digital de Stanford é projetar e implantar a infraestrutura dos serviços necessários para colaborativamente criar, disseminar, compartilhar e administrar informações em um contexto de biblioteca digital, disponível na Infobus – *Stanford Digital Library Project Technologies*, disponível na (URL: <http://diglib.stanford.edu/>).

O projeto de Stanford foi consolidado a partir de três propostas coordenadas entre: Universidade da Califórnia em *Berkeley Digital Library Project* - UCB, disponível na (URL: <http://elib.cs.berkeley.edu/>), da Universidade da Califórnia em Santa Bárbara - UCSB e da Universidade de Stanford. Uma das metas principais é demonstrar as tecnologias que estão emergindo da Biblioteca Digital da Califórnia, implementar e avaliar estas tecnologias em um sistema de testes com ajuda do San Diego Supercomputer Center - SDSC.

O esforço de Stanford é desenvolver tecnologias básicas, que são exigidas para superar as barreiras mais críticas para bibliotecas digitais efetivas. Uma destas barreiras é a heterogeneidade de informações e serviços. Outro impedimento é a falta de mecanismos poderosos de filtro que permitam aos usuários encontrarem informações valiosas. O acesso contínuo a informação é restringido pela incapacidade das interfaces das bibliotecas e ferramentas que efetivamente operem em dispositivos portáteis. Uma outra dificuldade é a falta de uma infra-estrutura econômica sólida que encoraje os fornecedores a tornar as informações disponíveis e que dê garantias de privacidade aos usuários.

2.3.3. Biblioteca digital de vídeo Informedia

O projeto da *Informedia Digital Video Library*, biblioteca digital de vídeo é uma iniciativa de pesquisa da Universidade Carnegie Mellon que estuda a forma que bibliotecas digitais multimídia, disponível na (URL: <http://www.informedia.cs.cmu.edu/>). O projeto Informedia tem sido pioneiro em novas abordagens de representação e indexação automática de vídeo e áudio, navegação, visualização, busca e recuperação, além de encaixá-los em um sistema para uso na educação, informação e ambientes de entretenimento. Alguns projetos da Informedia são apresentados a seguir.

Informedia I foi o projeto original da NSF – financiado pela *Digital Library Initiative* - DLI, combinando exclusivamente reconhecimento de fala, compressão de imagem e processamento de linguagem natural para transcrever, segmentar e indexar automaticamente o vídeo linear.

Informedia II: Nesta fase transformam o paradigma de acesso a bibliotecas digitais de vídeos por intermédio de visões significativas e manipuláveis de conjuntos de documentos de vídeo, múltiplas consultas e resumos de grandes quantidades de vídeo de fontes heterogêneas distribuídas. Os trabalhos de informação vídeo são as tecnologias chave na

Informedia III, construídos com o avanço das pesquisas de visualização de informação para lidar efetivamente com múltiplos documentos de vídeo.

2.4. Iniciativa da OAI

A comunidade científica vê as publicações eletrônicas informais, conhecidos como *eprints archives*, na rede como um meio de aumentar sua visibilidade, acelerar o avanço da ciência e disseminar amplamente os resultados das pesquisas, vistas como patrimônio da humanidade. Ela soube, com muita propriedade, acercar-se das novas possibilidades abertas pelas tecnologias de informação de resultados das atividades de pesquisa (SENA, 2000).

Em um passo seguinte à criação dos arquivos *eprints*, pré-formatados, que a comunidade científica internacional se mobilizou para torná-los interoperáveis, isto é, passíveis de serem consultados simultaneamente. Esta interoperabilidade foi alcançada mediante a adoção de um conjunto de especificações técnicas e princípios organizacionais bastante simples, porém potencialmente poderosos e de grande alcance, no objetivo de integração desses arquivos. Esta iniciativa é conhecida como *Open Archive Initiative* - OAI, disponível na (URL: <http://www.openarchives.org/>).

A dimensão da iniciativa do OAI em nível mundial e sua cobertura regional e temática podem ser mais bem avaliadas consultando a lista dos arquivos disponíveis. Um repositório eletrônico aberto que apresenta características específicas permite que o autor faça a submissão de seus trabalhos em formato digital para cadastro, edite-os, receba críticas e contribuições de outros autores. Ao submeter um documento para armazenamento e disponibilizá-lo no arquivo eletrônico, um autor também o descreve, em um formato de catalogação, de onde serão extraídos os metadados como autor, título, idioma, assunto, que permitirão recuperar o documento. Os metadados são, portanto, obtidos como um subproduto da submissão de um documento. O *site* permite também a consulta e acesso direto aos

documentos eletrônicos nele armazenados. Um servidor compatível com o *Open Archives Initiative Protocol for Metadata Harvesting* - OAI PMH permite a exposição de metadados dos trabalhos nele armazenados para coleta automática (*harvesting*) e reuso por provedores de serviço de informação. Miores detalhes encontram-se na seção 4.4.2 – Protocolo OAI-PMH.

2.5. Iniciativas Nacionais

O crescimento acelerado do número de documentos digitais existentes na Internet, no território nacional, com a possibilidade de se multiplicar as cópias em versões dos mesmos documentos, com localizações diversas e variáveis no tempo, tem-se a idéia das grandes dificuldades na identificação, organização e descrição dos recursos disponíveis, diz o info filósofo Lévy a este respeito:

“[...] quem utilizar freqüentemente a Internet para pesquisar determinadas informações ou documentos certamente já ficou esmagado com a quantidade de informação (na maior parte dos casos é irrelevante) recolhida, nervoso com o tempo que precisou despende, perdido e frustrado por não ter encontrado algo que cuja existência não duvida, mas que se encontra escondido em alguma galáxia distante”. (LÉVY, 2000).

Alguns softwares para automação de bibliotecas como Aleph já estão em funcionamento em algumas universidades brasileiras, apresentando bons resultados na gestão da documentação física de biblioteca tradicional. Têm-se outros softwares como Gnuteca, trabalhando diretamente na Internet para armazenamento de conteúdo digital, assim como a biblioteca digital de teses e dissertações da Universidade de São Paulo (USP), nas áreas de humanas, exatas e biológica.

O Brasil está com um projeto em conjunto com o Ministério da Ciência e Tecnologia para construção da biblioteca universitária brasileira em 2010 (CUNHA, 1999). Os esforços estão atualmente direcionados aos estudos de ferramentas viáveis para sua

constituição. Poucas universidades estão participando desta, o que dificulta uma ação conjunta na adesão e participação ativa, conforme recomendado pelo IBCT:

“A continuidade desta trajetória é hoje a criação da Biblioteca Digital Brasileira (BDB). Produto de ampla articulação entre os principais atores em informação científica e tecnológica no país, esta iniciativa do Ibict tem dois eixos principais: objetiva criar facilidades para que a comunidade acadêmica brasileira possa publicar em meio eletrônico, diretamente na rede, e facilitar a esta comunidade formas de encontrar facilmente a informação relevante para suas atividades de ensino e pesquisa.”. (IBCT, 2001).

O Instituto Brasileiro de Informação em Ciência e Tecnologia - IBCT, por meio do projeto da Biblioteca Digital Brasileira - BDB, passou a fomentar o desenvolvimento de recursos para estruturar uma base de informações de interesse. Para texto completo, como teses e dissertações, artigos de periódicos, trabalhos em congressos, arquivos eletrônicos de *eprints*, integrando e provendo interoperabilidade entre estes recursos mediante acesso unificado aos mesmos, via única de interface Web. O projeto prevê o uso do OAI PMH como um dos mecanismos para prover interoperabilidade, coletando seus metadados para uma base comum.

Iniciativas pioneiras como o SCIELO disponível na (URL: <http://www.scielo.br>), oferece um portal que abrange dezenas de periódicos, associados a uma metodologia para publicar e prover acesso a periódicos eletrônicos em texto completo. Já é uma realidade no Brasil a publicação de textos completos, um interesse do C&T, Ministério da Ciência e Tecnologia de publicação na rede de Internet (PACKER, 1998).

A Universidade de São Paulo - USP implantou em 2001, a sua biblioteca Digital de Teses e Dissertações da Universidade de São Paulo com o objetivo de facilitar o acesso remoto a essa parte de sua produção intelectual. Engloba todas as áreas de humanas, exatas e biológicas, com diferentes estruturas e conteúdos.

A USP em 2001, possuía o maior sistema de pós-graduação do país e produzia cerca de 1.500 teses de doutorado e 2.600 dissertações de mestrado em 259 programas de pós-graduação. A diversidade e complexidade do sistema, já que não havia até o momento outra

iniciativa institucional desse porte no país, apresentaram vários desafios à equipe encarregado do seu desenvolvimento (SABER-USP, 2005).

Aliado às discussões da criação da Biblioteca Digital, foi idealizado o Portal do Conhecimento da USP, disponível na (URL: <http://www.saber.usp.br>), para ser constituído de diversas bibliotecas digitais, sendo a de teses e dissertações o seu primeiro produto. Com isso, todas as futuras bibliotecas digitais na universidade devem observar os conceitos do Portal (MASIERO, 2001).

A seguir é apresentada uma visão geral da arquitetura da biblioteca digital da USP. Nessa figura é possível verificar que os usuários utilizarão uma interface via Web para fazer suas interações de submissão e busca. Por meio dos servidores Web, é possível realizar todo o controle e administração das teses e dissertações. Os processos, programas, interfaces e os relacionamentos de toda a arquitetura podem ser vista na figura 2, que exemplifica a arquitetura da biblioteca digital da USP.

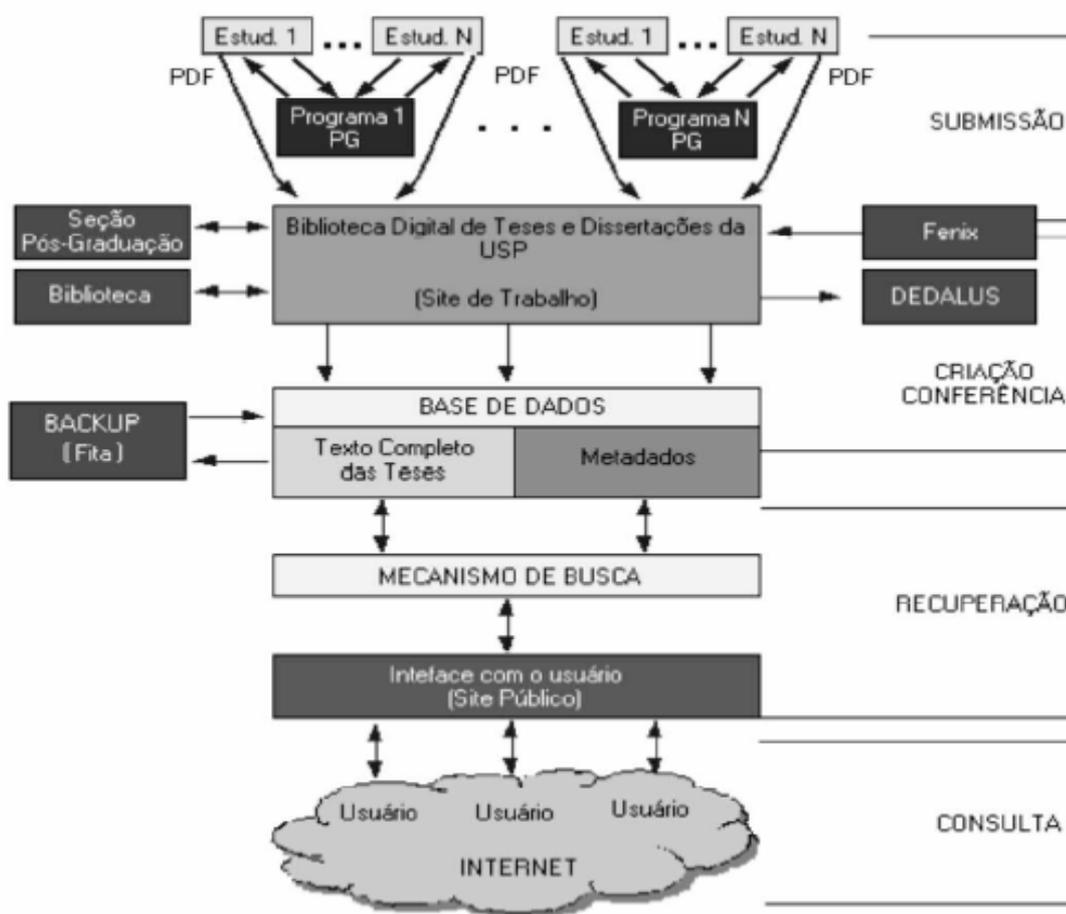


Figura 2 – Arquitetura da Biblioteca Digital da USP (SABER-USP, 2005).

Assim como o projeto da USP, outros projetos de bibliotecas digitais estão em andamento. Como comentado anteriormente, existe uma preocupação quanto à integração e a interoperabilidade de informações armazenadas em todas estas bibliotecas digitais. É correto afirmar que existe uma preocupação que os dados sejam interoperáveis mesmo quando suas estruturas de base e suas arquiteturas sejam diferentes.

“Bibliotecas digitais envolvem a integração de sistemas complexos, incluindo coleção de documentos com estruturas, mídias e conteúdos variados, além de uma mistura de componentes de hardware e software interoperando, ao longo de diferentes estruturas de dados, algoritmos de processamento e múltiplas pessoas, comunidades e instituições com diferentes objetivos, políticas e culturais”. (FOX e MACHIONINI, 2001).

Neste capítulo foram apresentados conceitos sobre bibliotecas digitais, as mudanças proporcionadas pelo avanço do uso delas na Internet. As definições que os órgãos

internacionais como a DLF, padronizando as características básicas de funcionamento de outras bibliotecas digitais. Apresentou-se também um breve histórico das bibliotecas até o aparecimento das bibliotecas digitais, seguindo para as principais bibliotecas digitais que foram as primeiras soluções implementadas.

No próximo capítulo serão apresentados os conceitos de fundamental importância para este trabalho. A Web Semântica têm um papel importante para o advento da Internet. A forma de abordagem no contexto dos metadados em relação à informação disponibilizada, representada pela linguagem XML que é considerada uma metalinguagem para a disseminação das informações, serão investigadas as arquiteturas e os tipos de metadados, todas as características da XML e arquitetura de RDF. Por fim a investigação da interoperabilidade e sua contribuição para a Web Semântica.

3. WEB SEMÂNTICA

A Web Semântica ou inteligente vêm-se apresentando como uma possível solução para ordenar o caos informacional existente na Internet.

Web Semântica é uma extensão da Internet atual, por apresentar a estrutura que possibilita a compreensão e o gerenciamento dos conteúdos armazenados na Internet independente da plataforma em que estes se apresentem. Seja texto, som, imagem e gráficos a partir da valorização semântica desses conteúdos, e por meio de agentes que serão programas coletores de conteúdo advindos de fontes diversas capazes de processar as informações e permutar resultados com outros programas (BERNERS-LEE, 2002).

Web Semântica, a exemplo da Web atual, é tão descentralizada quanto possível e que deverá manter a responsabilidade exigida por esta descentralização, procurando alcançar o ideal de consistência de interconexões, porém permitindo seu crescimento exponencial (BERNERS-LEE, 2005).

Para a implantação ou reorganização da Web Semântica há um contingente de pesquisadores trabalhando no *World Wide Web Consortium* - W3C, hospedado no Massachusetts Institute of Technology, Laboratory for Computer Science - MIT, nos Estados Unidos, no *Institute National de Recherche in Informatique et em Automatique*, na França e no Japão, a *Keio University Shanam Fujisawa*. O W3C é um fórum aberto de indústrias e organizações com a missão de elevar a Internet ao seu potencial máximo (W3C, 2004).

Uma troca e um correto uso dos dados baseados na Internet requerem informações sobre sua organização e seu significado. Estas informações, que são chamadas de contexto da informação, fornecem a base para a determinação de relacionamentos entre os dados e os aspectos do mundo real que eles descrevem. Para a explícita representação e troca deste contexto de informação são usados os metadados.

3.1. Arquiteturas de metadados

Os metadados são conjuntos de dados-atributos, devidamente estruturados, com base em padrões internacionais, para representar informações de um recurso informacional em meio digital ou não – digital, contendo uma série de características e objetivos (ROSETTO, 2003).

Ainda em (ROSETTO, 2003) os objetivos dos metadados são localizar, identificar e recuperar dados de um recurso informacional. Proporcionar controles de ordem gerencial e administrativo permitindo conexões e remissões *links* para pontos internos ou externos. Estes *links* possibilitam a interoperabilidade entre sistemas de informação, dentro de padrões. Informar sobre as condições de acesso e uso da informação.

Etimologicamente, metadados significa "dado sobre dado"; dado que descreve a essência, atributos e contexto de emergência de um recurso que caracteriza suas relações, visando seu acesso e uso potencial (FERREIRA, 1986)

Neste contexto, metadados refere-se a alguma estrutura descritiva e temática da informação sobre outro dado; que é usado para ajudar na identificação, descrição, localização e gerenciamento de recursos da Internet. Entretanto, eles podem ser aplicados em qualquer meio.

Segundo os metadados podem possuir a classificação de estrutural ou semântico. *Metadados estrutural* representa a informação que descreve a organização e estrutura dos dados gravados; por exemplo, informações sobre o formato, os tipos de dados usados e os relacionamentos sintáticos entre eles (IKEMATU, 2001). Em contraste, metadados semânticos fornecem informações sobre o significado dos dados disponíveis e seus relacionamentos semânticos.

Os dados que descrevem o conteúdo semântico de um valor de dado (como unidades de medida e escala), ou dados que fornece informações adicionais sobre sua criação

(algoritmo de cálculo ou derivação da fórmula usada), linhagem dos dados (fontes) e qualidade (atualidade e precisão). Faz-se necessário elaborar um modelo de metadados que descreva o contexto da informação de uma maneira não ambígua ou redundante. Uma conceitualização de um domínio específico de problema ou ontologias que forneçam um acordo comum de vocabulários para que os dados sejam referenciados. Assim, uma ontologia serve como uma base comum para a representação de dados e metadados.

Definir metadados tem sido uma tarefa difícil, pois as várias interpretações sobre o assunto estão relacionadas ao estágio da organização dentro da evolução e hierárquica da gestão do conhecimento (IKEMATU, 2001).

Descrever modelos que permitam uma associação flexível de metadados com os itens de dados disponíveis. As fontes de dados descrevem informações equivalentes diferentemente. Elas fornecem diferentes aspectos da mesma informação, e representam o mesmo aspecto do mundo real usando diferentes construções estruturais ou conceitos semânticos. Um objeto semântico representa um item de dado junto com sua base de contexto semântico que consiste de um conjunto flexível de meta – atributos que explicitamente descrevem a compreensão implícita sobre o significado do item de dado. Adicionalmente, cada objeto semântico possui um rótulo de conceito associado a ele, que especifica o relacionamento entre o objeto e os aspectos do mundo real que ele descreve. Estes rótulos são adquiridos de uma ontologia. A detecção e resolução destas heterogeneidades semânticas obviamente requerem conhecimento sobre a exata base semântica dos dados (BERNERS-LEE, 2005).

A forma pelas quais os metadados serão processados por uma aplicação, pode ser categorizada sob dois pontos de vista segundo Ahmed et. al. (2001). A primeira diz respeito à relação entre o metadado e o recurso que ele descreve. Desta maneira o metadado pode ser classificado como embutido ou externo. Já a segunda se preocupa com a relação entre o metadado e a aplicação que processa. Nesta forma de olhar para o metadado, ele pode ser classificado em centralizado ou distribuído.

3.1.1. Metadado embutido distribuído

O metadado embutido é aquele que está contido no conteúdo do recurso que ele estiver descrevendo. Consideram-se as duas formas de categorizar o metadado, é natural ao estarmos diante de um caso em que o metadado é embutido no seu recurso e também seja distribuído. Desta forma, o metadado estará embutido num conjunto de recursos que por sua vez estarão distribuídos sob o ponto de vista da aplicação que processa.

Um exemplo de metadado embutido distribuído é o *Resource Directory Description Language* - RDDL. O documento RDDL dispõe de um conjunto de informações sobre uma fonte que inclui: descrição sobre o recurso referenciado que contém uma descrição compreensível por humanos e uma lista de outros recursos relacionados (BORDEN & BRAY, 2002). Um documento RDDL é projetado para servir como o corpo de uma entidade que é referenciada por uma *Uniform Resource Identifiers* – URI, que está sendo usada como o nome de um XML *Namespace*.

RDDL é uma extensão do XHTML Basic 1.0 com um elemento adicional chamado *resource*. Esse elemento serve como um *xlink* para o recurso referenciado, e contém uma descrição compreendida por humanos e *links* compreendidos por máquinas que descrevem o propósito (*purpose*) da ligação e a natureza (*nature*) do recurso que está sendo associado. A natureza do recurso que está sendo associado é indicada no atributo *xlink:role* e o propósito do *link* é indicado pelo atributo *xlink:arcrole*.

O elemento *resource* está definido em um *namespace* cujo nome é “<http://www.rddl.org/>”. Este elemento representa um *xlink* simples, e faz uso dos atributos definidos no *namespace* do *xlink*. O elemento deve sempre ocorrer no conteúdo do elemento “*body*” do HTML. Já no XHTML o elemento *resource* pode ser usado em qualquer lugar onde é possível usar o elemento “<p>”.

Um exemplo da natureza de um recurso pode ser dado por um esquema XML que é projetado para ser usado com um namespace. Esta natureza pode ser dada como *xlink:role*=" <http://www.w3.org/2001/XMLSchema>". Em casos que só existe um único recurso com uma natureza em particular, o propósito daquele recurso que está sendo associado pode ser inferido desta natureza. Por exemplo, se existe um único recurso associado para um *namespace* em particular, cuja natureza indica que ele é um esquema XML, os aplicativos que irão processá-lo podem inferir que o seu propósito é para validar elementos naquele *namespace*.

3.1.2. Metadado externo centralizado

Metadado centralizado é aquele que é recuperado de uma única fonte por uma aplicação. Normalmente, essa fonte é algum tipo de repositório de dados, em que as aplicações são capazes de realizarem consultas sob um determinado conjunto de critérios. Sob a perspectiva da aplicação cliente, um repositório centralizado é simples de manipular, para isso, precisa ter conhecimento de como realizar consultas sobre ele e como interpretar os resultados.

Em algumas aplicações, tais como um sistema de documentos, o metadado está naturalmente centralizado, pois o recurso cujo metadado esta descrevendo, também se encontra centralizado. Contudo, em outras situações cuja intenção é proporcionar ao cliente um repositório central de metadados, é necessário para o provedor desta facilidade, agregar os metadados de um conjunto de recursos que estão distribuídos.

O principal problema de qualquer uma das abordagens citadas é manter a consistência entre os metadados e o recurso associado. É necessário manter uma identificação única que ligue a descrição com o recurso descrito. E esta ligação deverá ser mantida ao longo do tempo, ou seja, se o recurso tiver sua identificação alterada, seu metadado também deverá

refletir essa mudança. Outra preocupação é manter sempre consistente o conteúdo do recurso com as informações que estão contidas nos metadados. Sempre que o conteúdo do recurso for alterado, também deverá ser atualizada sua descrição.

3.1.3. Metadado externo distribuído

Do ponto de vista da aplicação cliente, o metadado é distribuído se ele for recuperado de múltiplos repositórios de dados. Quando se trabalha com vários repositórios, uma tarefa extra é atribuída à aplicação cliente que terá que ser capaz de estabelecer múltiplas sessões de consultas com cada repositório de dados, e também ser capaz de combinar os resultados de cada fonte antes de apresentá-los ao usuário.

Ao permitir que aplicações manipulem repositórios de metadados distribuídos, pode-se enriquecer a diversidade de fontes de onde os metadados poderão ser capturados. Por exemplo, receber metadados de fontes que não estão sob nosso controle e então combiná-los com a própria fonte.

Arquiteturas de metadados distribuídos abordam essa distribuição de diferentes maneiras. A forma mais comum é a arquitetura cliente – servidor. Desta forma seria definida, por exemplo, uma lista estática de servidores no cliente. Mas esta forma só seria interessante diante de poucas fontes de interesse. Uma arquitetura mais interessante é aquela usada pelas aplicações de ponto a ponto (*peer-to-peer*). Neste caso seria possível disseminar uma consulta ao longo de uma rede, e qualquer um dos nós seria capaz de responder a consulta e passá-la adiante.

3.2. Linguagem XML

Existem várias linguagens de metadados que podem ser utilizadas para marcar ou anotar estes recursos informacionais, como o XML e RDF. As linguagens de marcação (*Markup Languages*) evoluíram desde o *Standard Generalized Markup Language* - SGML, para o *HyperText Markup Language* - HTML em 1980 e *Extensible Markup Language* - XML em 1996.

Ao contrário da HTML que por meio das marcas pré-definidas, gerenciam os textos marcados e controlam sua representação estabelecendo ligações entre os documentos, a linguagem XML marca semanticamente um documento, que consiste num padrão utilizado para marcação de documentos que contém informações estruturadas, ou seja, documentos que contém uma estrutura clara e precisa da informação armazenada.

O objetivo é fornecer os benefícios da SGML, gerenciar o conteúdo e a estrutura do documento independente do seu tamanho e prover suporte a um número ilimitado de aplicações. Por ser baseada nas especificações da SGML, um sistema de marcação generalizado proporciona uma visão hierárquica da estrutura de um documento através de elementos denominados *tags* iniciais e finais.

Esta estrutura define e separa claramente o conteúdo, significado e apresentação. Assim os documentos em XML podem ser indexados com maior precisão que nas páginas plenas escritas em HTML.

O padrão XML permite troca de informações entre diversas plataformas e possibilita a descrição de dados em arquivos texto. “A linguagem XML apresenta-se como ótima ferramenta para a publicação de informações na Internet” (OLIVEIRA, 2002).

Pode ser utilizado de modo geral, no armazenamento de bases de dados e documentos estruturados e as aplicações construídas com XML criam novos padrões e linguagens (XML, 2003).

A criação de marcações definidas pelo próprio usuário é a melhor forma de proporcionar uma descrição dos recursos em termos de metadados. Também fornece uma linguagem sofisticada de folha de estilo, *eXtensible Stylesheet Language* - XSL, baseada no padrão *Document Style and Semantics Specification Language* - DSSSL que adiciona estilos visuais (cores, tipos de fontes, etc.) aos documentos Web. Desta forma, a formatação do documento é tratada separadamente de sua estrutura, suprimindo e resolvendo algumas das deficiências do HTML (XML, 2003).

A linguagem XML apresenta um modelo de dados simples, servindo de base para a construção de documentos mais complexos. Desta forma, um documento XML pode ser visto como uma estrutura em árvore, onde cada nó representa um elemento ou um componente lógico do documento.

3.2.1. Características da linguagem

XML é a representação textual do dado. O componente básico em XML é o *element*, isto é, o texto limitado entre delimitadores (tags) `<> ... </>` (incluindo os próprios delimitadores) tal como *peessoa*, *nome*, *idade* e *e-mail* no exemplo mostrado na figura 3, exemplo de código XML, que apresenta um código escrito na linguagem XML.

```
<peessoa>
  <nome> Douglas Sanches da Cunha </nome>
  <idade> 35 anos </idade>
  <e-mail>
professordouglas@femanet.com.br </email>
</peessoa>
```

Figura 3 – Código XML

O subelemento é usado também para descrever a relação entre um elemento e seus componentes. Assim, *idade*, *nome* e *e-mail* são subelementos de `<peessoa> ... </peessoa>`. Uma característica importante dessa linguagem, é que, ao contrário do HTML, o usuário pode definir seus próprios marcadores, onde letras maiúsculas e minúsculas são diferenciadas.

É possível associar atributos a elementos, muito embora este termo tenha a conotação de *propriedade* em modelos de dados. Um atributo é definido como um par (nome, valor), a exemplo de idioma, moeda e país no exemplo da figura 4, associação de atributos aos elementos em XML.

```
<livro>
  <titulo idioma= "Ingles"> CiberCultura
</titulo>
  <preço moeda= "Real"> 50.25 </preço>
  <editora pais= "Brasil"> Abril </editora>
</livro>
```

Figura 4 – Associação de atributos aos elementos em XML.

A linguagem XML deve respeitar duas restrições: *tags* devem ser corretamente aninhados e atributos devem ser únicos. Quando um documento atende a essas duas restrições diz-se que um documento é bem formado, sendo possível organizá-lo segundo uma estrutura de árvore.

3.2.2. Documentos bem-formados

Um documento é considerado bem formado se todos os elementos estão constituídos pelos pares de marcadores em torno do dado, e os atributos estão sintaticamente corretos. Todo documento XML válido começa com uma informação de cabeçalho responsável por:

- Descrever as regras estruturais que a marcação no documento almeja seguir;
- Listar os recursos externos (entidades externas) que constituem o documento;
- Declarar os recursos internos (entidades internas) que podem ser requeridos no documento;

- Listar os tipos de recursos que não são XML (anotações ou dados binários), que são encontrados no documento e para os quais aplicações de ajuda podem ser requeridas.

Essa informação de cabeçalho é mais conhecida como definição do tipo de documento *Document Type Definition* – DTD, que é utilizado como uma gramática para documentos XML. Assim, qualquer documento XML que faça referência a um DTD deve ser analisado, a fim de se verificar se o mesmo segue as regras especificadas naquele DTD, determinando se o mesmo é ou não um documento válido.

A figura 5 apresenta um exemplo de um documento DTD para um documento XML. Neste exemplo, um documento livro é constituído de elementos e estes, por sua vez, de outros sub-elementos.

```
<? xml version= "1.0"?>
<!DOCTYPE biblioteca [
<!ELEMENT livro (autor+, título, editora)>
<!ELEMENT autor (nome?, sobrenome)>
<!ELEMENT editora (nome_edit, endereço)>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT sobrenome (#PCDATA)>
<!ELEMENT nome_edit (#PCDATA)>
<!ELEMENT endereço (#PCDATA)>
<!ELEMENT título (#PCDATA)>
```

Figura 5 – Documento DTD

Um DTD pode ser considerado como um esquema de banco de dados (todo livro tem um ou mais autores, título), apesar de apresentar uma série de limitações quanto a esse tipo de utilização, porque ele só suporta um tipo de dado primitivo (PCDATA) i.e., *string*.

A sintaxe de um documento XML é armazenada em arquivos tabelas DTD, que determinam as regras, hierarquias e marcações criadas para caracterizar as informações do documento.

A figura 6 ilustra um documento XML contendo as informações de uma biblioteca, de acordo com o DTD da figura 5.

```

< biblioteca >
  <livro>
    <autor>
      <nome> Douglas </nome>
      <sobrenome> Sanches </sobrenome>
    </autor>
    <autor>
      <nome> Camila </nome>
      <sobrenome> Pitanga </sobrenome>
    </autor>
    <título> Arquitetura de Metadados </título>
    <editora>
      <nome_edit> Editora Abril Cultural </nome_edit>
      <endereço> Rua Benjamin,151 Assis </endereço>
    </editora>
  </livro>
  <livro>
    <autor>
      <nome> Juliana </nome>
      <sobrenome> Dias </sobrenome>
    </autor>
    <título> Banco de Dados </título>
    <editora>
      <nome_edit> Campus </nome_edit>
      <endereço> 133 Rui Barbosa, Assis </endereço>
    </editora>
  </livro>
</ biblioteca>

```

Figura 6 – Documento XML de acordo com o DTD.

3.3. Modelo RDF

O modelo de dados do *Resource Description Framework* – RDF fornece uma arquitetura abstrata e conceitual para o processamento e o intercâmbio de metadados, definindo uma sintaxe neutra como forma de representação de expressão. O modelo de dados do RDF pode especificar o relacionamento entre entidades e prover interoperabilidade estrutural, porém não fornece mecanismos para a declaração de propriedades e nem para a definição de relacionamentos entre tais propriedades e outros recursos (RDF, 2005).

Enquanto a função principal de metadados é descrever um documento por meio de atributos conferidos a um objeto, retratando suas características como dimensão, formato, autoria, localização e outro com o objetivo de intercambiar os dados. Já o RDF são grafos para descrever e intercambiar metadados.

As regras para construção de RDF são: Fonte (*Resource*), qualquer coisa que possa ter uma identificação na rede: *Uniform Resource Identifier* – URI; ou um elemento individual de um documento XML – Propriedade, é uma fonte que possui um nome e pode ser usado como uma propriedade e – Declaração (*statement*) que consiste numa tripla de uma fonte, uma propriedade e um valor, partes estas conhecidas como “sujeito”, “predicado” e “objeto” de uma Declaração (*statement*), como representado na figura 7, a seguir.

- **Tripla (Sujeito, Predicado, Objeto)**
 - **Sujeito** -> **Recurso**
 - **Predicado** -> **Propriedade**
 - **Objeto** -> **Recurso ou um literal**

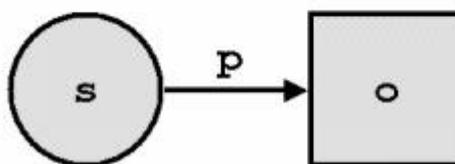


Figura 7 – modelo RDF.

Conforme a figura 7, para um objeto existe um atributo A cujo valor é V, sendo que objetos e valores são intercambiados no RDF, em que um objeto pode ser um valor. Os grafos RDF podem identificar qual o tipo do objeto e estabelecer relações à sua definição (BRAY, 2004).

O modelo de dados primitivo é representado por meio de um *Directed Labeled Graphs* – DLG, que consiste de três tipos de objetos que descrevem inter-relacionamentos entre recursos em termos de propriedades e valores nomeados:

- **Recurso (*resource*)**: tudo que é descrito por meio de expressões RDF é chamado de recurso. Um recurso pode ser tanto um documento eletrônico, uma coleção, uma página ou um *site* na Internet, como também um objeto não acessível diretamente na Web (ex.: livro impresso). Recursos são sempre nomeados através de URI, que permite a introdução de identificadores para qualquer entidade;

- **Propriedade (*Property*):** é um atributo ou característica que descreve o recurso, ou seja, o seu metadado. Uma propriedade representa também o relacionamento entre recursos.

- **Declaração (*Statement*):** corresponde à associação de um recurso específico, uma propriedade e o valor dessa propriedade para esse recurso. Uma declaração é dividida em: *sujeito, predicado e objeto*, onde o objeto dessa declaração pode ser um outro recurso ou um literal. Desta forma, um objeto pode ser especificado por URI, uma cadeia de caracteres ou outro tipo de dado primitivo definido pela XML.

Os elementos do DLG são representados por nós e arcos, onde:

- Nó (representado por uma elipse): identifica o recurso;
- Nó (representado por um retângulo): contém o valor;
- Arco (representado por uma seta): identifica a propriedade. A direção da seta é importante, pois o arco sempre inicia com o tema e aponta para o objeto da declaração;
- Nó (representado por uma elipse vazia): identifica um recurso anônimo.

Na figura 8, aparece a representação de uma declaração: “*Douglas Sanches é o autor desta dissertação <http://www.fema.edu.br/jbdigital/exemplo.htm>”.*

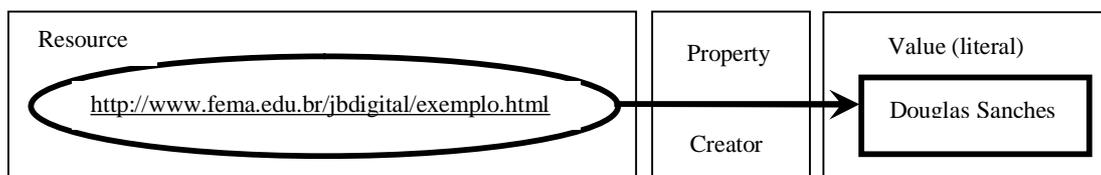


Figura 8 – Declaração RDF.

A propriedade **Creator** descreve o recurso na declaração que é formada por:

- **Resource:** <http://www.fema.edu.br/jbdigital/exemplo.htm> à Sujeito
- **Property:** creator à Predicado
- **Value (recurso ou literal):** Douglas Sanches à Objeto

Várias implementações podem ser desenvolvidas utilizando RDF nível simples ou Schema RDF e aplicações XML. O modelo adotado para a Web Semântica parte do modelo RDF. Este modelo básico contém apenas conceito sobre asserção (*assertion*) e de “quotation”, criando asserções sobre asserções (BERNERS-LEE, 2005).

Ainda em Berners-Lee (2005), no artigo “*Semantic Web Road Map*”, implantações e aplicações como conversão de linguagem, leis da lógica, com o objetivo de imprimir lógica aos documentos; *predicado lógico* (not, and, or) e *leis de quantificação* (para todo x y (x)). O autor cita a introdução da *linguagem de provas* e de *busca e índices de termos* evoluindo para *vocabulário*.

3.3.1. Sintaxe RDF

A sintaxe RDF apresentada no escopo desse artigo é a sintaxe de serialização por expressar a completa capacidade do modelo de dados RDF, demonstrando clareza da estrutura do modelo. Uma entidade simples com valores simples é utilizada quando se deseja mencionar apenas uma característica sobre um literal de um determinado recurso (W3C, 2004). Na figura 9, a representação utilizando a sintaxe XML/RDF.

```
<?xml version= "1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description about =
"http://www.femanet.com.br/jbdigital/exemplo.htm">
    <dc:Creator>Douglas Sanches </dc:Creator>
  </rdf:Description>
</rdf:RDF>
```

Figura 9 – Sintaxe XML/RDF.

Tal representação especifica dois *namespaces* identificados pelo *token* “xmlns”, a saber: **rdf**. O primeiro identificado pelo URI “<http://www.w3.org/1999/02/22-rdf-syntax-ns#>”, que referencia as definições do modelo básico do RDF; o segundo identificado pelo URI “<http://purl.org/dc/elements/1.1/>”, que referencia o esquema de definição dos elementos, ou seja, os *tokens* criados pelo autor desta expressão.

A *tag* “rdf:Description about” menciona o recurso a ser descrito. A *tag* “dc:Creator” especifica as propriedades referentes ao recurso citado com o valor Douglas Sanches.

Já uma entidade estruturada é utilizada quando se deseja mencionar mais de uma característica sobre um literal de um determinado recurso. A entidade estruturada pode ser representada como um outro recurso e quando esse recurso não possui um nome (recurso anônimo), sua representação é feita através de uma elipse vazia. A figura 10, apresenta um grafo com propriedade com valor estruturado a partir do exemplo da declaração a seguir: “A pessoa cujo nome é Douglas Sanches e e-mail dscunha@femanet.com.br, é criador do recurso ‘<http://www.fema.edu.br/jbdigital/artigo.htm>’ intitulado de Metadados”.

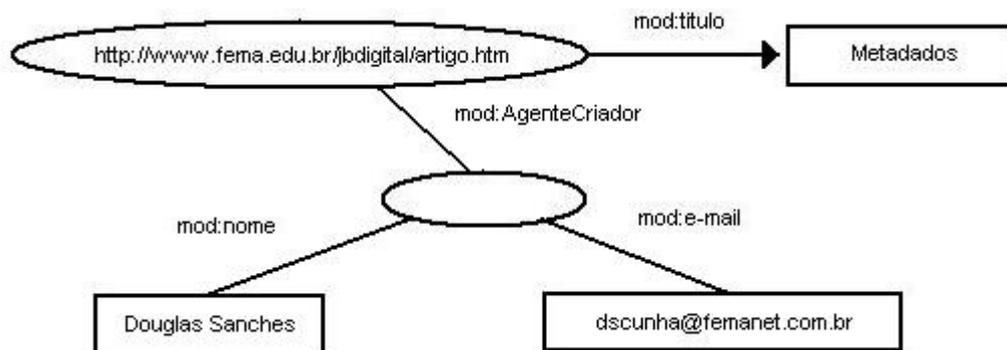


Figura 10 – Propriedade com valor estruturado.

A seguir está a figura 11, que representa a sintaxe RDF/XML com os valores estruturados.

```
<?xml version= "1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:mod = "http://www.fema.edu.br/moddec/definition">
  <rdf: Description about =
"http://www.fema.edu.br/jbdigital/artigo.htm">
    <mod:titulo > Metadados </mod:titulo >
    <mod:AgenteCriador>
      <mod:nome>Douglas Sanches</mod:nome >
      <mod:e-mail>dscunha@ femanet.com.br</mod:e-mail>
    </mod:AgenteCriador>
  </rdf:Description>
</rdf: RDF>
```

Figura 11 – XML/RDF com valores estruturados.

Na representação anterior foi utilizado um novo *namespace* denominado “mod” definido pelo criador da expressão e que possui o esquema de definição no endereço “<http://www.fema.edu.br/moddec/definition>”.

A entidade de coleção de objetos, que tem a habilidade de criar coleção de valores, pois, permite que uma declaração possa ser efetuada sobre um conjunto de valores referentes a uma propriedade. O RDF define três tipos de recipientes de objetos representados pela figura 12 (que apresenta coleção do tipo **Bag**) que especifica que o curso representado pelo recurso “www.fema.edu.br/jbdigital/curso/pg” é constituído pelos alunos Patrícia, Luiza e Beth.

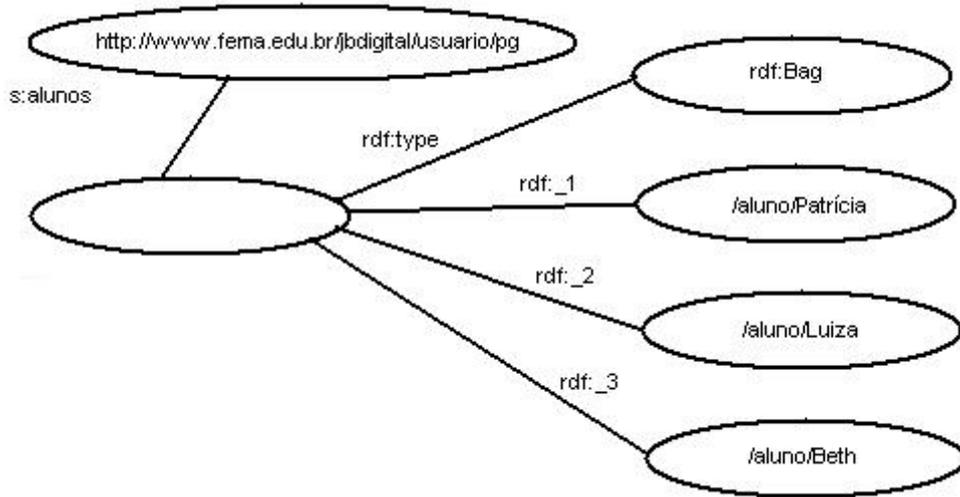


Figura 12 – Coleção do tipo Bag.

A propriedade “rdf:type” especifica o tipo de coleção “rdf:Bag” que está sendo utilizado. Cada elemento da coleção é nomeado através de propriedades numeradas como “_1”, “_2”, “_3” e assim sucessivamente. A representação utilizando a sintaxe RDF/XML é mostrada na figura 13.

```

<rdf:RDF
  xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:s="http://fema.edu.br/schema/"
  <rdf:Description about =
    "http://www.fema.edu.br/jbdigital/usuario/pg">
    <s:alunos>
      <rdf:Bag>
        <rdf:li
resource="http://www.fema.edu.br/alunos/Patrícia">
          <rdf:li resource="http://www.fema.edu.br/alunos/Luiza">
            <rdf:li resource="http://www.fema.edu.br/alunos/Beth">
          </rdf:li>
        </rdf:li>
      </rdf:li>
    </rdf:Bag>
  </s:alunos>
</rdf:Description>
</rdf:RDF>

```

Figura 13 – Sintaxe XML/RDF tipo Bag.

A **Sequence** é uma lista de recursos ou valores ordenados utilizada para declarar que uma propriedade pode ser composta de múltiplos valores que obedecem a uma determinada ordem como, por exemplo, alfabética ou numérica de valores. Este tipo de coleção permite valores duplicados.

A **Alternative** é outra lista de recursos ou valores que apresentam valores possíveis para uma propriedade, proporcionando uma livre escolha de qualquer item da lista.

Por fim as declarações sobre declarações é o modelo RDF que permite não apenas descrever recursos, mas também descrever as próprias declarações. Com isso, o modelo de uma declaração é necessário para a criação de novas declarações sobre o modelo.

O RDF apresenta o mecanismo de reificação de modo a permitir que um *statement* também possa ser tratado como um recurso. É definido a partir das propriedades: *subject*, *predicate*, *object* e *type*.

- **Subject**: define o recurso que está sendo descrito pela declaração modelada, isto é, o valor da propriedade *subject* é o recurso sobre o qual a declaração original é feita.
- **Predicate**: identifica a propriedade original da declaração modelada. O valor da propriedade *predicate* é um recurso representando uma propriedade específica na declaração original
- **Object**: identifica o valor da propriedade em uma declaração modelada. O valor da propriedade objeto é o objeto na declaração original
- **Type**: o valor dessa propriedade descreve o tipo do novo recurso. Todas as declarações reificadas são instâncias do `rdf:statement`. Um recurso do tipo `df:statement` é, por definição, composto pelas propriedades `rdf:subject`, `rdf:predicate` e `rdf:object`.

Considere o exemplo a seguir, expresso esse mecanismo, onde: “*O artigo <http://www.fema.edu.br/teste.pdf> cujos autores são: Douglas Sanches, Cíntia Carvalho e*

Diomara Reigato foi aceito por <http://www.fema.edu.br/semanainfo2005>". A figura 14 apresenta um exemplo XML/RDF.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:mod="http://www.fema.edu.br/moddec/definition">
  <rdf:Description>
    <rdf:subject resource = "http://www.fema.edu.br/teste.pdf"/>
    <rdf:predicate resource = mod:autor />
    <rdf:object>
      <rdf:bag>
        < rdf:li resource="Douglas Sanches"/>
        < rdf:li resource="Cíntia Carvalho"/>
        < rdf:li resource="Diomara Reigato"/>
      </rdf:bag>
    <rdf:object>
      <rdf:type resource="http://www.w3.org//1999/02/22-rdf-syntax-
ns#Statement"/>
    <mod:aceito_por>"http://www.fema.edu.br/semanainfo2005"</mod:aceito_por>
  </rdf:Description>
</rdf:RDF>

```

Figura 14 – Outro XML / RDF.

3.4. Esquema RDF

O Esquema RDF provê um mecanismo para a declaração de propriedades de recursos (título, autor etc.), relacionamentos entre essas propriedades, classes de recursos que podem ser descritas (livros, pessoas, páginas Web etc.), combinações possíveis entre classes e propriedades, etc. Basicamente, o esquema RDF provê um sistema de tipos básicos que é empregado na construção dos esquemas (RDF, 2005).

Conforme descrito no modelo RDF, os recursos podem ser instâncias de uma ou mais classes, indicado pela propriedade de *rdf:type*. Classes são frequentemente organizadas de forma hierárquica. Por exemplo, uma classe Cachorro poderia ser considerada uma

subclasse da classe Mamífero que é uma subclasse da classe Animal. Isto significa que qualquer recurso do tipo *rdf:type* Cachorro é também do tipo *rdf:type* Animal. A especificação RDF descreve uma propriedade, *rdfs:subClassOf* para especificar tais relacionamentos entre classes.

O sistema de tipos é especificado nos termos do modelo de dados básico do RDF como *recursos* e *propriedades*, segundo uma hierarquia de classes apresentada na figura 15. As setas indicam que o objeto da qual elas partem possui o tipo do objeto para o qual está apontando.

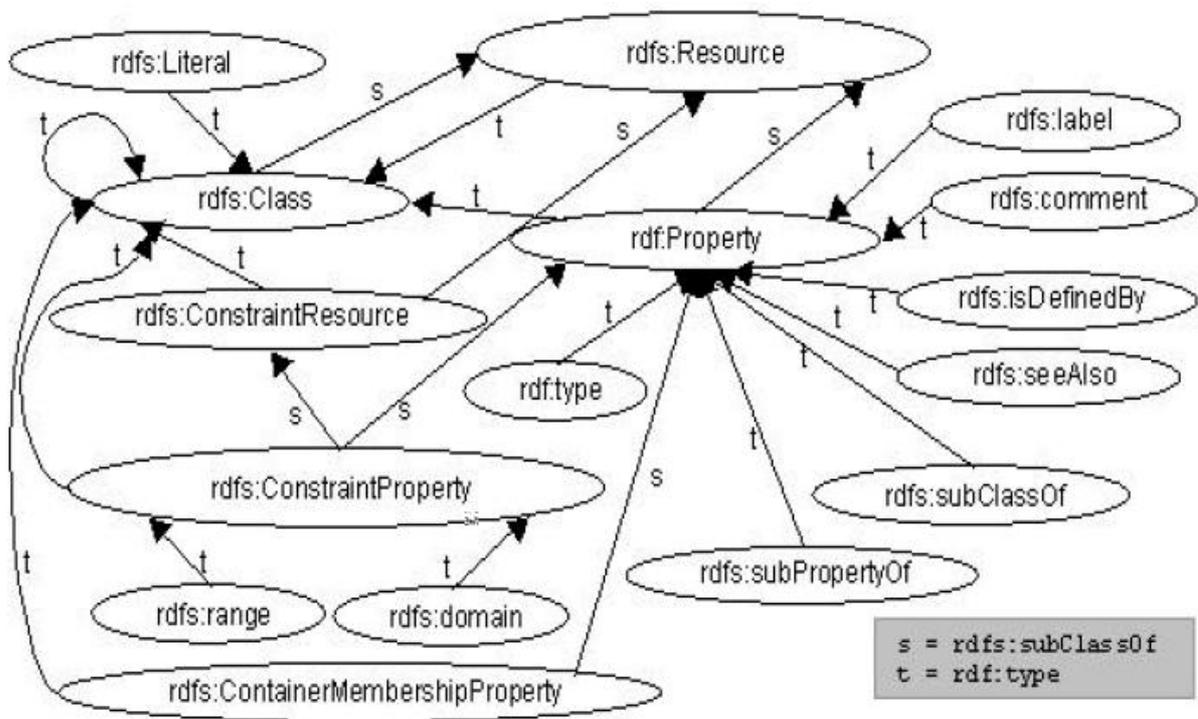


Figura 15 – Grafos da hierarquia de classes do esquema (RDF, 2005).

As classes, propriedades e restrições são descritas sucintamente a seguir:

- **rdfs:Resource** - representa qualquer objeto descrito por expressões RDF;
- **rdf:Property** - representa uma propriedade de um recurso;
- **rdfs:Class** - é um tipo de recurso que representa o conceito de classe de objetos;

- **rdf:type** - determina que um recurso é membro de uma classe, possuindo todas as suas características. Um recurso no modelo RDF pode ser uma instância de mais de uma classe;
- **rdfs:subClassOf** - é um tipo de propriedade que declara a relação de sub-classe / superclasse entre duas classes. Uma classe pode ser sub-classe de mais de uma classe no modelo RDF;
- **rdfs:subPropertyOf** – declara um relacionamento de especialização entre duas propriedades;
- **rdfs:seeAlso** - é uma propriedade usada para declarar que um recurso pode prover informações adicionais a respeito do recurso;
- **rdfs:isDefinedBy** – é uma sub-propriedade da propriedade rdfs:seeAlso que define o objetivo do recurso.
- **rdfs:ConstraintResource** - é uma subclasse de Resource, cujo objetivo é prover um mecanismo que possibilite ao processador RDF determinar as demais classes definidas que estabelecem regras de restrições;
- **rdfs:ConstraintProperty** - é um recurso subclasse de rdf:property. Todas as instâncias dessa classe são usadas para especificar restrições. É também subclasse de rdfs:ConstraintResource, correspondendo a um subconjunto que representa propriedades;
- **rdfs:range**: é uma instância da classe rdfs:ConstraintProperty que é usada para restringir valores de propriedades. O valor da propriedade range é sempre uma classe;
- **rdfs:domain** - é uma instância da classe rdfs:ConstraintProperty usada para especificar a classe na qual uma determinada propriedade pode ser aplicada. Esta

definição difere da Orientado a Objetos, onde o domínio encerra os possíveis valores de uma determinada propriedade.

O RDF permite modelar uma aplicação a partir de um esquema conceitual em *Unified Model Language* - UML, representando hierarquias entre classes e seus relacionamentos.

3.5. Mecanismos de Consulta em RDF

As instâncias oriundas de aplicações RDF podem ser consultadas segundo duas abordagens: a visão dos metadados RDF como um Banco de Dados relacional ou XML e a visão dos metadados RDF como uma base de conhecimento.

A primeira abordagem é semelhante às consultas efetuadas em bancos de dados com características relacionais ou XML, onde a ênfase é dada às operações em nível de estrutura do documento. Neste caso, as descrições RDF são tratadas como base de dados relacionais, ou seja, a aplicação RDF é vista como uma instância XML e não como um modelo de metadados visando prover interoperabilidade. As linguagens de consulta incluídas nesse contexto fornecem construtores necessários à realização de consultas a documentos e coleções de um esquema específico, onde são conhecidos as propriedades e os relacionamentos entre recursos. A figura 16, mostra o exemplo de um grafo de um esquema RDF.

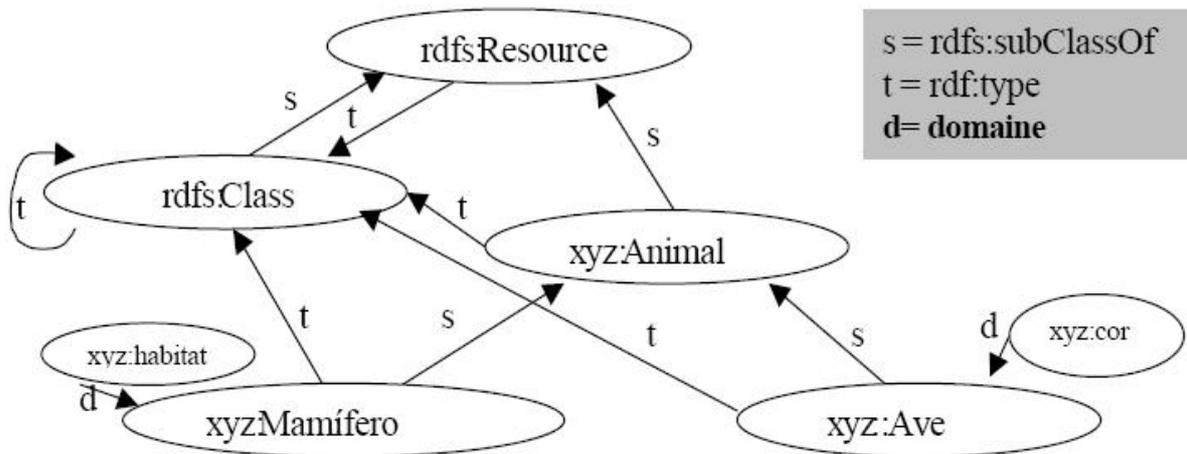


Figura 16 – Grafo de um esquema RDF.

Na figura 17, a representação do esquema Animal em RDF, respeitando o RDF

Schema.

```

rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  <rdf:Description ID="Animal">
    <rdf:type
      resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf
      resource="http://www.w3.org/TR/WD-rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description ID="Mamífero">
    <rdf:type
      resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:subClassOf resource="#Automóvel"/>
  </rdf:Description>
  <rdf:Description ID="Ave">
    <rdf:type
      resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:subClassOf resource="#Animal"/>
  </rdf:Description>
  <rdfs:Property rdf:ID="cor">
    <rdfs:comment>Cor da penugem </rdfs:comment>
    <rdfs:domain rdf:resource="#Ave">
    <rdfs:range
      rdf:resource="http://www.w3.org/2000/03/example/classes#String">
  </rdfs:Property>
  <rdfs:Property rdf:ID="habitat">
    <rdfs:comment>Local onde é encontrado </rdfs:comment>
    <rdfs:domain rdf:resource="#Mamífero">
    <rdfs:range
      rdf:resource="http://www.w3.org/2000/03/example/classes#String">
  </rdfs:Property>
</rdf:RDF>

```

Figura 17 – Representação do esquema Animal em RDF.

O objetivo maior da arquitetura RDF é definir um mecanismo para descrever recursos não vinculados a um domínio específico de aplicação. Como resultado do trabalho em conjunto com várias comunidades, o RDF recebeu a influência de várias fontes diferentes. As principais influências vieram das comunidades de padronização da Web na forma de metadados; de biblioteconomia; de estruturação de documentos na forma do SGML e XML; de representação do conhecimento e ainda de outras áreas de tecnologia que também contribuíram no projeto RDF: programação orientada a objetos, linguagem de modelagem e bancos de dados.

Na área de descoberta de recursos, a arquitetura RDF possibilita a implantação de mecanismos de pesquisa mais eficientes. Na área de catalogação, a arquitetura RDF pode ser utilizada para descrever os recursos de informação em um *site* da Internet, em uma biblioteca digital. Na área de agentes inteligentes, o RDF pode facilitar o intercâmbio de informações.

No capítulo 3 foi possível observar a importância da Web Semântica e sua proposta. Verificou-se a arquitetura de metadados, assim como seus tipos e formas de apresentação. O XML como uma linguagem mais adequada para a escrita dos metadados e flexibilidade na sua aplicação. Por fim, foi possível checar a importância do Schema RDF como responsável pela objetividade da semântica dos dados na Internet.

No próximo capítulo serão apresentados os conceitos de ontologias e a importância da representação do conhecimento, que pode ser representado em uma ontologia e a interoperabilidade de informações, por meio de algumas linguagens que representam à ontologia.

4. ONTOLOGIAS E A WEB SEMÂNTICA

Os computadores poderiam ter acesso a coleções estruturadas de informações (dados e metadados) e de conjuntos de regras de inferência que ajudem no processo de dedução automática para que seja administrado o raciocínio automatizado, ou seja, a representação do conhecimento (BERNERS-LEE, 2001).

A Web Semântica trará a estrutura ao conteúdo significativo (valorização semântica) de páginas da Internet, possibilitando um ambiente em que programas agentes ou agentes inteligentes irão procurar de uma página a outra, executando tarefas sofisticadas para os usuários. Mas para que isso se realize de forma dinâmica, os conceitos de ontologias são de extrema importância.

A figura 17 apresenta a proposta para a Web semântica, em forma de camadas, representando o relacionamento entre elas, mostrando a importância de cada uma delas na construção do futuro da Internet (KOIVUNEN and MILLER, 2006).

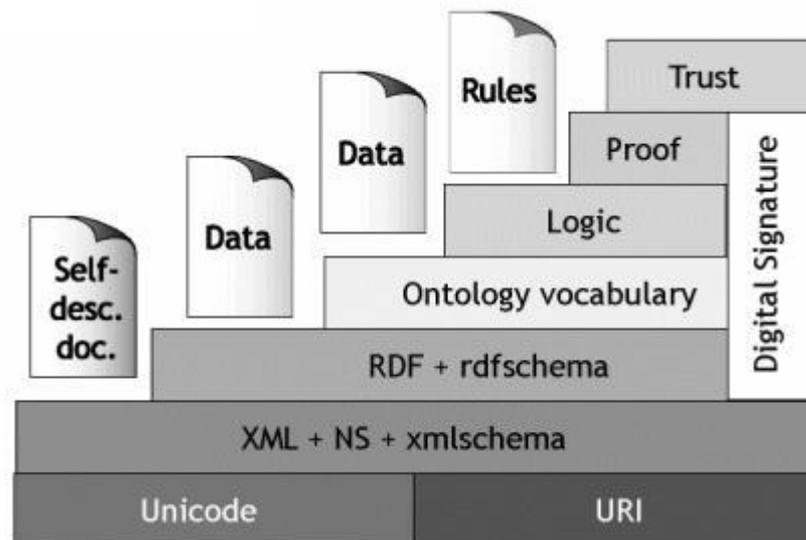


Figura 17 – As camadas da Web semântica (KOIVUNEN e MILLER, 2006).

A arquitetura da Web Semântica é processo em construção por grupos de trabalho da W3C (2005) com o objetivo de mapear as complexas relações semânticas, lógicas, de sintaxe e de apresentação dos documentos com algumas regras no novo espaço.

As regras são especificadas por meio de ontologias, que permitem representar explicitamente a semântica dos dados. Por meio dessas ontologias é possível elaborar uma rede enorme de conhecimento humano, complementando o processamento dos computadores e melhorando qualitativamente o nível de serviços na Web.

Pode-se complementar que a Web Semântica, compõe-se de três elementos: representação do conhecimento (*Knowledge representation*); Ontologias (*Ontologies*) e Agentes (*Agents*), apresentados a seguir.

4.1. Representação do conhecimento

O gerenciamento do conhecimento é um processo que identifica, avalia, captura a estrutura e difunde o conhecimento sobre determinados assuntos, que uma instituição gera durante suas atividades, para que possa ser aproveitado por qualquer indivíduo. Esse processo está baseado no estabelecimento e implantação de uma metodologia para coletar, analisar e catalogar as informações que estão dispersas em arquivos, documentos e na estrutura cognitiva das pessoas, tornando essas informações disponíveis para todos.

O objetivo do processo é capturar o conhecimento das pessoas com reconhecida capacidade em cada domínio ou assunto, e estruturar esse conhecimento de modo que possa ser armazenado e facilmente compreendido por outras pessoas. A obtenção de conhecimento é um dos maiores desafios para quem deseja implantar este processo.

A identificação de qual conhecimento é útil e que deve ser considerado, já exige um alto grau de preparo e definições metodológicas e conceituais muito claras. Pois, nem todo dado nos leva a ter informações úteis e nem toda informação é relevante para o assunto que se

deseja. Deve-se identificar como esse conhecimento se expressa e de qual local pode ser obtido (documentos, arquivos, sistemas, pessoas, etc.). Portanto, tem-se que avaliar se determinada informação traz de fato algum conhecimento para o domínio a ser considerado, de que tipo é essa informação e como ela pode ser relacionada com outras, para produzir conhecimento.

Identificado e avaliado como sendo de interesse para o assunto, necessita-se capturar esse conhecimento, ou seja, extrair e trazê-lo para dentro do processo de estruturação. A captura pode envolver diversas técnicas como entrevistas especializadas, observações locais, simulações de situações, análise de processos e atividades, verificação de documentos, análise automática de textos.

Uma vez que o conhecimento tenha sido obtido, deve ser estruturado, ou seja, categorizado e relacionado com outros. A categorização ou classificação do conhecimento é o primeiro passo para permitir sua perfeita caracterização e facilitar a sua localização. Todo conhecimento consiste em conceitos, definições, informações sobre sua história e relacionamentos entre esses conceitos e definições. A capacidade de identificar rapidamente relacionamentos existentes e de formar novos relacionamentos que se mostram úteis é o que caracteriza uma pessoa com um bom conhecimento (ALVARENGA, 2004).

Deve-se fazer uma consideração sobre a terminologia empregada. Sabe-se que os computadores armazenam em seus arquivos apenas *dados*. Quando se verifica o conteúdo desses arquivos de computador, seja por meio de telas, relatórios ou qualquer outro meio, o que se obtém são apenas conjuntos de signos, em um formato de seqüências que formam palavras, ou desenhos, mapas, sons, imagens etc. A *informação* que extraímos desse conjunto de signos está diretamente ligada à nossa capacidade de relacionar esses signos entre si, e entre esses signos e relacionamentos com o que está armazenado em nossa estrutura cognitiva (ALVARENGA, 2004).

Um relatório qualquer contendo números, pode não trazer nenhuma informação, mas pode carregar informações valiosas para quem souber interpretar esses signos, avaliar o seu conteúdo implícito e relacioná-los com outros signos. Esse é o processo básico para a obtenção de informações.

O computador é utilizado normalmente para armazenar dado. Por isso, o nome ainda hoje usado para denominar a área de qualquer instituição responsável pelo suporte de tecnologia de informação é o de Centro de Processamento de Dados. Quem trabalha nesses departamentos sabe apenas que o computador recebe determinado conjunto de signos, que são processados e armazenados de acordo com algumas regras e depois esses signos são armazenados em determinadas seqüências para serem exibidos em telas e relatórios. Somente as pessoas capazes de avaliar o seu conteúdo (os usuários) é que podem obter informações novas sobre algum aspecto do assunto.

Com isto pode-se dizer sobre o tratamento de conteúdo que, “a pessoa atualiza o seu conhecimento sobre o assunto, ou seja, acrescenta, altera, confirma ou nega a existência de relacionamentos” (RUGGLES, 1996). Comenta-se que uma pessoa tem conhecimento a respeito de algum assunto ou domínio quando, ao ser colocado em situações novas, ela demonstra habilidade para incorporar novos relacionamentos e modificar outros, agindo de forma mais eficaz.

Para fazer com que o computador nos auxilie na tarefa de obter mais conhecimento, tem-se que torná-lo capaz de subir um nível, naquilo que ele habitualmente realiza. Ou seja, passar do "processamento de dados" para o "processamento de informações". Tem-se que armazenar e divulgar informações e não apenas dados ou seqüências de signos. Isso significa registrar e processar basicamente relacionamentos entre dados. O computador passará a fornecer os “dados interpretados”, de acordo com as necessidades.

Portanto, quando se fala em bases de conhecimentos esta na verdade referindo-se as Bases de Dados e Bases de Informações. Armazenar definições, fatos, dados e

relacionamentos entre eles, que forneça novas informações sobre esses fatos, definições, dados e relacionamentos.

O Gerenciamento do Conhecimento é um conjunto de atividades, técnicas e métodos que garantem a obtenção, armazenamento e divulgação das informações que realmente propiciem um aumento do nível de conhecimento dos usuários (RUGGLES, 1996).

4.2. Conceito de Ontologia

Para o dicionário Aurélio ontologia é “1. s.f. Ciência do ser em geral. 2. Filos. Parte da metafísica que estuda o ser em geral e suas propriedades transcendentais”. Para outras enciclopédias, ontologia é “s.f. Tratado dos seres em geral; teoria ou ciência do ser enquanto ser, considerado em si mesmo, independentemente do modo pelo qual se manifesta”.

Apesar da palavra “ontologia” denotar uma teoria sobre a natureza do ser ou existência, em Inteligência Artificial ela pode ser interpretada como o conjunto de entidades com suas relações, restrições, axiomas e vocabulário. Uma ontologia define um domínio, ou, mais formalmente, especifica uma conceitualização (GRUBER, 1993).

O termo ontologia tem sua origem na Filosofia, mas foi redefinido para ontologias digitais onde é definida como “*a especificação explícita de uma conceitualização*”, (GRUBER, 1994), isto é, tem como pressuposto a existência de algo. Quando um domínio de conhecimento é representado através de um formalismo declarativo, o conjunto de objetos que pode ser representado é denominado de universo de discurso.

Assim, ontologia é a descrição explícita e precisa de conceitos e relações que existam em um domínio particular tal como uma organização, um domínio de estudo, uma área de aplicação. Por exemplo, a ontologia definida para um programa de computador deve incluir a especificação dos conceitos: programa, identificador, declaração, expressão, controles, operadores. O operador maior ou igual que (\geq), por exemplo, deve estar relacionado por um

tipo de relacionamento à classe de operadores relacionais. Identificadores devem ser definidos como string, etc.

Designa a faceta semântica da representação dos seres perante os entes, aquilo que se convencionou chamar de assuntos, conteúdos temáticos dos registros sobre a realidade (GUARINO, 1995). A ontologia, não é a semântica que estabelece uma ligação terminológica entre membros de uma comunidade podendo ser os membros, agentes humanos ou máquinas. No jargão dos pesquisadores de inteligência artificial, a ontologia pode ser representada em um documento ou arquivo que define formalmente a relação entre termos.

Uma ontologia fornece um entendimento sobre uma conceitualização compartilhada de um determinado domínio de aplicação. Os conceitos específicos numa ontologia fornecem vocabulário comum para que nenhuma negociação adicional seja necessária. Adicionalmente, a ontologia fornece informação sobre a representação do dado descrito sobre a base do modelo. Numa situação ideal, todas as instâncias que fazem uso dos dados e metadados de um determinado domínio devem aderir à ontologia correspondente (GUARINO, 1995).

O autor salienta ainda que se haja necessidade de integração e de uniformidade de linguagem e significado dos dados por meio da organização, os metadados devem ser os núcleos dos esforços de forma centralizada. Para que isto ocorra, a instituição deve ter pleno conhecimento das vantagens da organização dos dados. A informação baseada em metadados enriquece documentos com informações semânticas acrescentando explicitamente metadados aos recursos.

Uma ontologia requer o uso de um vocabulário específico para descrever uma realidade e mais um conjunto de axiomas lógicos necessários para dar semântica ao significado pretendido pelas palavras daquele vocabulário. Dessa forma, duas ontologias podem referir-se a uma mesma conceitualização, isto é, um só domínio de conhecimento, muito embora apresentem vocabulários distintos de representação, tal como o uso de dois idiomas distintos, para descrever uma mesma ontologia, por exemplo.

Normalmente, uma ontologia é organizada em hierarquias de conceitos, justamente pelo fato de não refletirem nenhum formalismo específico, e de representarem com frequência um vocabulário comum entre usuários e sistemas (CLARK, 1999).

Existem alguns níveis e tipos de ontologias. A vantagem de uma ontologia é de se lidar com conceitos, representando-os formalmente. Algumas propriedades de uma ontologia são compartilhamento e filtragem.

A propriedade de compartilhamento significa que um acordo deve existir entre diferentes agentes baseados no acordo de ontologias comuns, isto é, devem ter o mesmo entendimento sobre um dado conceito. Por exemplo, quando duas pessoas falam sobre “Mercado”, é preciso assegurar que ambas estejam tratando do mesmo conceito.

Uma ontologia contém os conceitos e relações relevantes a uma dada tarefa de modelagem, podendo conter informações de diferentes naturezas. Geralmente existem três tipos de informação numa ontologia, a terminológica, assertiva e a pragmática:

A terminológica constitui-se do conjunto básico de conceitos e relações da ontologia, denominada de camada de definição da ontologia;

A assertiva é denominada de camada de axiomas da ontologia, constitui-se do conjunto de assertivas que se aplicam aos conceitos e relações. Considere o exemplo:

- $\text{solteiro}(x) \equiv \text{homem}(x) \wedge \sim \text{casado}(x)$
- $\text{carnívoro}(x) \equiv \text{animal}(x) \wedge \text{come}(y) \wedge y = \text{'carne'}$
- $\text{herbívoro}(x) \equiv \text{animal}(x) \wedge \text{come}(y) \wedge y = \text{'vegetais'}$
- $\text{animal}(x) \equiv \text{carnívoro}(x) \vee \text{herbívoro}(x)$

E a pragmática que corresponde à camada de ferramentas e contém um conjunto de informações pragmáticas que não se enquadram em (a) ou (b). Contém a forma de expressar esse conceito na tela, por exemplo. Um conjunto de dados pragmáticos pode ser definido a partir de uma ontologia. Considere, por exemplo, o conceito classe definido em (a) por uma ontologia. Pode-se classificar nessa categoria a forma de apresentar os conceitos e relações

embutidas na ontologia ou mesmo a *Application Programming Interface* – API construída para tal, expressa na linguagem de definição de interface.

4.2.1. Elemento de Representação Comum: *Frames*

Os formalismos de representação de conhecimento orientados a classes e relações (frames, redes semânticas e lógicas de descrições), proporcionaram a oportunidade de se estruturar conhecimento com ontologias. Até o início dos anos 90, o conhecimento de um sistema especialista não podia ser reusado ou compartilhado, organizando-se em bases de conhecimento monolíticas e isoladas, escritas em diversas linguagens, sem interfaces de acoplamento, e, portanto, sem interoperabilidade, apesar dos formalismos lógicos serem, muitas vezes, similares (FARQUHAR, 1996). Sentiu-se a necessidade de reuso de conhecimento em muitos dos domínios modelados, especialmente em medicina, onde os conceitos são numerosos e cheios de relacionamentos. Precisava-se encontrar um substrato comum de representação, no qual o conhecimento pudesse ser especificado e facilmente convertido para outros formalismos.

O projeto KSE (*Knowledge Sharing Effort* – em português, esforço de compartilhamento de conhecimento), tornou-se um pioneiro ao escolher *frames* como base para a construção de ontologias reutilizáveis e ambientes para a sua especificação e reuso. Os *frames*, por sua simplicidade e expressividade, têm sido o elemento básico das ontologias. O formalismo comum de interoperabilidade entre bases de conhecimento sobre os mais diversos domínios, escritas sob a forma de redes semânticas, lógica de descrições e até dos próprios *frames* (FARQUHAR, 1996).

Os *frames* foram inspirados, na forma de como as pessoas resolvem problemas, trazendo da memória estruturas de padrões de situações passadas, e tentando instanciá-las no presente. É considerado um dos precursores dos atuais objetos imperativos, guarda

substancial semelhança com estes, pois possui: classes, atributos, instâncias de classes, herança e herança múltipla que são os *frames* (MINSKY, 1975).

As classes funcionam como os conceitos das ontologias, normalmente organizados em hierarquias. As classes ligadas com classes mais altas são subclasses destas classes mais altas, que, por sua vez, são superclasses destas classes mais baixas.

Já os atributos definem as características de membros de uma classe, preenchidas com valores dos tipos especificados para os mesmos. Este tipo pode ser um dos tipos básicos (inteiro, string, boolean, float, símbolo etc.), ou o tipo especial instância de classe que é responsável por uma função muito importante em sistemas de *frames*: definir relações entre as classes.

As instâncias das classes são objetos criados e modelados de acordo com a definição das características da classe (seus atributos).

A herança é relação do tipo *é-uma* entre as classes, bem semelhantes ao conceito de subclasse dos objetos. É a herança que define a hierarquia. Diferentemente da maioria dos sistemas orientados a objetos, pode existir herança múltipla.

Além da herança múltipla, os *frames* têm uma outra característica própria. São dotadas de estruturas pré-definidas, chamadas de facetas, para especificar restrições sobre os atributos. As facetas mais comuns, além do tipo, presentes em linguagens de *frames* são:

- Valor default, que atribui um valor para o atributo, caso nenhum tenha sido assinalado na criação de uma instância ou subclasse;
- Valores permitidos (*allowed-values*) para o atributo;
- Domínio, que determina conjuntos de valores esparsos, possíveis para um atributo (por exemplo, 1..100, ou seja, de 1 a 100);
- Classes permitidas (*allowed-classes*) para o atributo, se ele é do tipo instância de classe 3;

- Cardinalidade, determinando o número máximo e mínimo de elementos de um atributo, pois ele pode ser multivalorado;
- Documentação, para descrever o atributo;

Observa-se ainda, que a faceta (valor default), faz com que as instâncias sejam consideradas exceções à classe, que por sua vez, faz com que um sistema de *frames* se enquadre como uma lógica não mono tônica. Uma vez que um dado é acrescentado à base de conhecimento, a conclusão do raciocínio torna-se diferente da normalmente esperada.

A construção de ontologias comuns tem sido proposta como abordagem promissora para interoperabilidade de sistemas. Uma ontologia como uma formalização compartilhada de certo domínio de aplicação. Por exemplo, uma formalização de conceitos sobre informações em ciência e tecnologia poderia permitir que diversos sistemas desse domínio compartilhassem do mesmo vocabulário, comum ao assunto. Essas informações devem ser difundidas e divulgadas para que o maior número possível de pessoas possa utilizá-las e assim aumentar o seu próprio conhecimento.

Assim, o uso da ontologia facilita o entendimento dos diferentes níveis de abstração. Quando se utiliza um dado sistema, é possível observar e trabalhar com diferentes modelos do mesmo sistema, cada caracterizado por uma dada ontologia. Estes modelos, naturalmente, permanecem relacionados entre si.

No contexto da Web, a utilização de ontologias seria importante: porque permite que agentes de *software* compreendam a semântica embutida nas definições e vocabulário específicos com respeito a um domínio, sem ambigüidades, viabilizando o intercâmbio de informações por meio de consultas. Bases de conhecimento podem também ser criadas especializando e instanciando aquela ontologia, por meio de uma aplicação específica.

Para a construção de uma ontologia em um projeto de software, precisa-se tomar algumas decisões de projeto que determinam sua qualidade baseada em critérios como eficiência, legibilidade, portabilidade, extensibilidade, interoperabilidade e reuso. Nesta

seção, serão revistos os princípios para a construção de ontologias visando maximizar o seu reuso, e depois técnicas de desenvolvimento de ontologias. Alguns princípios, se usados com precisão, garantem sua qualidade (GRUBER, 1993).

4.3. Agentes de software e Agentes Inteligentes

A função dos programas agentes ou agentes inteligentes é coletar conteúdos na Internet a partir de fontes diversas, processarem a informação e permutar os resultados com outros programas permitindo por meio de linguagem que expressa inferências lógicas resultantes do uso de regras e informação como aquelas especificadas pelas ontologias. O princípio está, não no entendimento, pela máquina, daquilo que está escrito e sim no reconhecimento de provas escritas na linguagem estabelecida pela ontologia, em que os programas-agentes, pela inferência lógica retornam respostas ao que foi requerido, ou agente e consumidor podem alcançar entendimento compartilhado permutando as ontologias, que oferecem o vocabulário necessário para a discussão.

Foi formulada uma proposta por (MUCHERONI e TAMAE, 2003) para modelos que usam XML para construir conhecimento com três etiquetas fundamentais: intenção, linguagem e agente. O modelo de agente *Believe, Desire and Intention agent* - BDI, designa o conhecimento com as *tags* de crença, desejo e agente de intenção. A interação lógica indicará a criação ontológica em uma etiqueta.

Uma arquitetura de sistemas multiagentes, para manipular a informação referente a um conjunto de classes sobre um mesmo grupo, como, por exemplo, o grupo científico, com classes como artigos científicos, eventos, pesquisadores. A arquitetura visa recuperar, classificar e extrair dados de páginas pertencentes às classes a um grupo, e a motivação principal para o emprego de sistemas multiagentes são beneficiar-se dos relacionamentos entre as classes. A visão geral da arquitetura está ilustrada na figura 18 apresenta (FREITAS e BITTENCOURT, 2002).

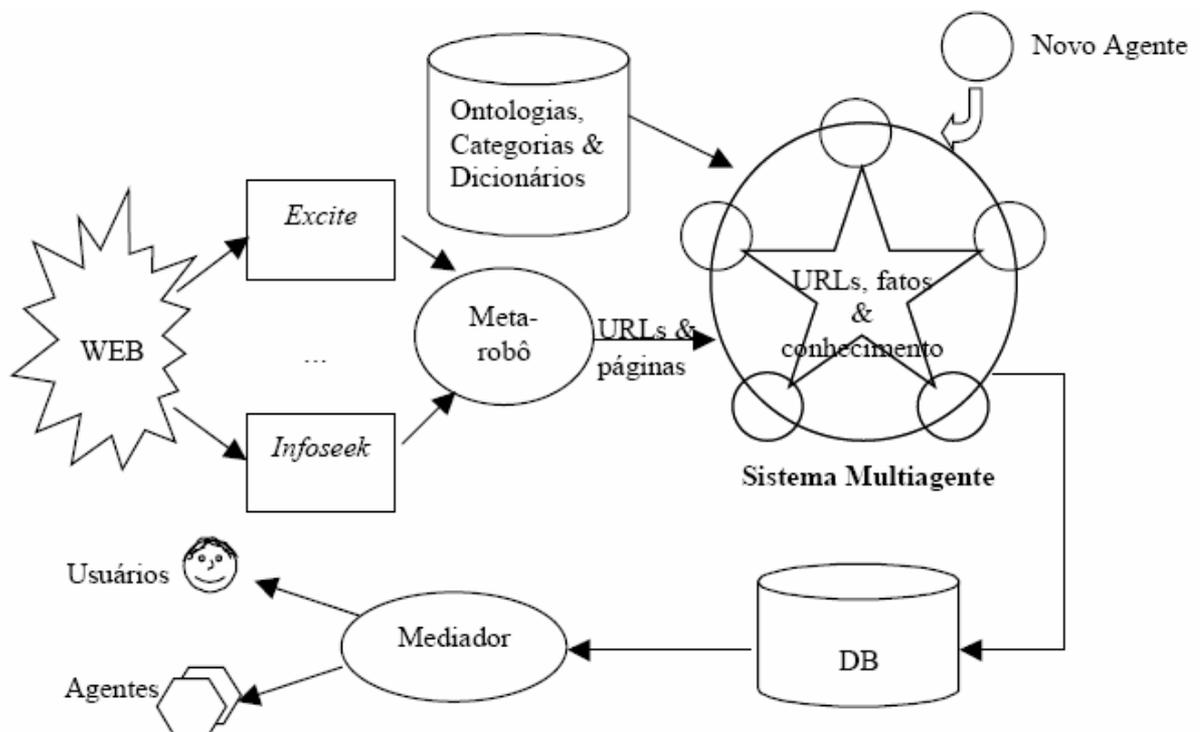


Figura 18 – Interação de agentes num sistema multiagentes (FREITAS e BITTENCOURT, 2002).

Na figura 18, a estrela indica troca de mensagens contendo regras de reconhecimento e fatos (conhecimento dos agentes), além das URL sugeridas entre os agentes. Cada agente possui um meta-robô, que consulta conecta-se a múltiplos motores de busca - como *Altavista*, *Excite*, *Infoseek* e outros - com palavras-chave que garantem cobertura em relação à classe de páginas processada pelo agente. (e.g., os termos *'call for papers'* e *'call for participation'*

para o agente CFP). As classificações e dados extraídos das páginas são postos num banco de dados, acessível por um mediador.

As ontologias servem como vocabulário de comunicação entre agentes, como também na definição e organização apropriadas de conceitos, relações, e restrições. A ontologia do domínio deve ser bastante detalhada para garantir precisão à classificação por conteúdo (FREITAS e BITTENCOURT, 2003).

Outra vantagem de se usar ontologias reside no ganho de expressividade e flexibilidade, uma vez que o conhecimento sobre uma classe não se circunscreve a termos e palavras-chaves como em engenhos de buscas, mas a qualquer fato que diga respeito às páginas, como estrutura, regiões, conceitos contidos nelas.

As ontologias provêm o mecanismo formal capaz de viabilizar o processamento semântico da informação através de uma máquina. O uso de ontologias permite que o entendimento compartilhado de termos possa ser utilizado por homens e programas para ajudar no intercâmbio de informações.

A Web Semântica juntamente com seus elementos e uma ontologia, não são suficientes para que a interação homem – máquina aconteça, portanto, a interoperabilidade é um conceito de fundamental importância neste processo.

A Internet é um ambiente composto por elementos das mais diversas tecnologias e, devido a sua abrangência universal e dependência da infra-estrutura de hardware, requer que seus componentes de software sejam extremamente modulares. Neste contexto, a colaboração entre os elementos é vital para que o conjunto deles produza os resultados esperados.

4.4. Interoperabilidade

Em uma situação ideal, diversos aplicativos devem interoperar, isto é, trocar dados e chamados a rotinas de forma transparente para o usuário. A interoperabilidade é definida

como, "em ter conexão efetiva com diferentes sistemas de computador, banco de dados ou redes a fim de apoiar a computação distribuída e ou o intercâmbio de dados" (ANDERSON e KUBIATOWICS, 2004).

Trata-se da capacidade de dois ou mais sistemas trocarem informações e usarem-nas de alguma forma. Esta necessidade de troca decorre dos sistemas que, para atender a determinadas funcionalidades indisponíveis no ambiente, tentam buscá-las em outros. Isto ocorre por diversas razões, sendo as principais: racionalização de recursos, modularidade das funções e prioridades de requisitos (ANDERSON e KUBIATOWICS, 2004).

Os dados são apenas um dos elementos deste ambiente. Porém, o conceito de dado abrange vários sub-elementos: números, datas, palavras, texto, imagem, vídeo, som, além de outros. Entre as diversas fontes de informação, pode haver diferentes sistemas, de sintaxe, de estrutura ou de semântica. Como na Internet as diferenças de sistemas (*hardware* e sistemas operacionais) são solucionadas por meio do uso padronizado do protocolo de comunicação em redes como: TCP ou UDP, no processo de interoperabilidade de dados, o desafio é vencer as barreiras da sintaxe, da estrutura e da semântica entre as fontes. Neste aspecto, quanto mais abrangentes forem as tecnologias empregadas no processo, mais recursos estarão disponíveis aos componentes envolvidos. As linguagens de marcação para descrição de informação ou conhecimento na Internet (HTML, XML e suas linguagens derivadas) atendem a estes desafios, cada uma com campo específico de aplicação.

A interoperabilidade sintática, diz respeito à forma como os metadados são codificados para a transferência, pode ser obtidos por meio do suporte da linguagem XML e seus vocabulários para definição de esquemas (DTD e XML Schema), como já apresentado anteriormente no capítulo três.

A interoperabilidade estrutural refere-se ao modelo de dados adotado para definir a estrutura dos metadados, que pode ser obtida por meio da sintaxe e do esquema RDF, porque esta especificação foi concebida para atender a esse requisito. Algumas lacunas da sintaxe

RDF, como a definição explícita de tipos de dados, podem ser resolvidas com a utilização em conjunto da especificação XML Schema. É importante ressaltar que na linguagem XML não há o conceito de classes e propriedades, conseqüentemente, não há o conceito de herança. Também nesta linguagem não há como definir regras de associação.

E na interoperabilidade semântica tem-se percepção do significado de cada metadado do domínio ou compreensão de seus conceitos. Também pode ser obtida pela especificação RDF, porém, uma solução mais abrangente, com relação à semântica, é a utilização de uma linguagem de definição de ontologias, a exemplo de *DARP Agent Markup Language* - DAML, que é uma aplicação XML/RDF.

Nessa linguagem há um conjunto mais completo de estruturas adequadas à interoperabilidade semântica. Convém observar que, em todas as linguagens de marcação, há algum mecanismo de definição de semântica. Até mesmo com a linguagem HTML isso pode ser observado quando se utiliza o *tag* META. O padrão *Dublin Core* - DC é um exemplo deste fato, quando utilizado desta maneira.

4.4.1. A iniciativa *Dublin Core*

A *Dublin Core Metadata Initiative* - DCMI, *Dublin Core* - DC é um foro aberto engajado no desenvolvimento de interoperabilidade on-line de metadados padrões que suportam um largo alcance de propósitos e modelos de negócio. Suas atividades incluem consenso dirigido trabalhando grupos, seminários globais, conferências, união de padrões e esforços educacionais para promover larga aceitação de padrões de metadados e práticas (DCMI, 2005).

O metadados do DC é usado para completar métodos existentes para buscar e indexar metadados baseados na Internet. As primeiras discussões que ocorreram foram concentradas, principalmente, em criar metadados para recursos eletrônicos. Entretanto, a

partir destas discussões, o consenso entre a comunidade de DC é que sistemas de descoberta de recursos podem e devem ser usados para descrever objetos físicos, digitais e reais. A maioria dos colaboradores do DCMI está envolvida amplamente no arquivamento ou catalogação de projetos que exigem o uso do metadados do DC, para habilitar grandes coleções de recursos de objetos para serem agrupados, nomeados, classificados e indexados de uma forma útil.

A característica do DC, que o torna um dos candidatos para utilização em sistemas de descoberta de recursos ainda em DCMI (2005), entra em várias categorias, como simplicidade, interoperabilidade semântica, consenso internacional e extensibilidade.

Com muita simplicidade ser utilizável por catalogadores como também por especialistas de descrição de recursos.

Permitir a interoperabilidade semântica na Internet, modelos de descrição discrepantes interferem com a habilidade para busca por meio de limites de técnicas. Promove uma compreensão comum do conjunto de descritores que ajudam a unificar outros dados padrões, aumentando a possibilidade de interoperabilidade semântica.

O reconhecimento de consenso internacional no âmbito da descoberta de recursos na Web é importante ao desenvolvimento da própria infra-estrutura. O DC se beneficia da participação ativa e da promoção de alguns países da América Norte, Europa, Austrália, e Ásia.

Prover a extensibilidade como alternativa econômica para modelos de descrição mais elaborados. Adicionalmente, inclui flexibilidade suficiente para codificar a estrutura e semântica mais elaborada inerente em padrões de descrição.

Portanto, as decisões quanto à linguagem de marcação a ser utilizada na Internet dependerão do tipo de interoperabilidade desejada no processo. A linguagem XML, com DTD ou XML *Schema*, aplica-se adequadamente à interoperabilidade sintática. RDF e RDFs aplicam-se para a interoperabilidade estrutural e casos não muito complexos de

interoperabilidade semântica, e, conseqüentemente também abrangem a interoperabilidade sintática. As linguagens para criação da ontologia como: DAML+OIL e OWL, aplicam-se adequadamente à conceituação de ontologias e casos complexos de interoperabilidade semântica.

Apesar de ser uma característica desejada nos sistemas, a interoperabilidade plena, dificilmente é atingida mesmo quando se trata de sistemas relativamente atuais. A interoperabilidade costuma ser impedida por barreiras de plataformas diferentes de hardware e também de software, incluindo sistemas operacionais, paradigmas de programação e de modelos de dados. Outra é a incompatibilidade dos modelos de dados subjacentes às aplicações. Os aplicativos de importação não poderão trocar os dados respeitando apenas as formas de representação.

4.4.2. Protocolo OAI - PMH

O protocolo OAI - PMH provê interoperabilidade não imediata, ou seja, não é um protocolo para busca on-line entre repositórios de *eprints*, bibliotecas digitais ou qualquer servidor na rede que queira expor, ou seja, tornar visíveis metadados de documentos nele armazenados para um programa externo que queira coletá-los.

Dentro da concepção OAI, existem as instituições chamadas provedoras de dados (*Data Providers*), que são bancos de documentos eletrônicos que oferecem facilidades para a publicação e armazenamento de documentos eletrônicos. A sua oferta em um servidor conectado à Internet, e as instituições provedoras de serviços (*Services Providers*), que coletam metadados de um ou mais provedores de serviço e com estes metadados prestam serviços de valor agregado. Portanto, ele define a troca de solicitações de metadados entre o servidor de *eprints* e um programa robô externo.

A troca de mensagens entre o servidor do provedor de dados e o programa robô externo do provedor de serviços para a transferência de metadados é unidirecional – o provedor de serviços faz solicitações ao provedor de dados, que responde enviando metadados. O OAI - PMH estabelece o *Dublin Core Metadata Element Set* na (URL: <http://purl.org/DC/documents/rec-dces-19990702.htm>) como conjunto mínimo de metadados a ser suportado.

4.5. Linguagens e ferramentas para criação de ontologias

Diversas linguagens e mecanismos para a definição de ontologias foram criados nos últimos anos. Algumas destas se destacam a Simple HTML Ontology Extensions - SHOE, XML-based Ontology Exchange Language - XOL, Ontology Inference Layer – OIL, DARP Agent Markup Language – DAML e Ontology Web Language - OWL.

A principal característica dessas linguagens está na capacidade de representar ontologias em RDF e RDFs, arquitetura já consagrada pela W3C para interoperabilidade de informações na Web. A seguir serão apresentadas algumas dessas linguagens que têm merecido maior destaque na literatura.

A linguagem SHOE é uma extensão para HTML (recentemente também utilizado em XML) que provê meios de incorporar conhecimento semântico de forma a ser entendido por máquina (robôs, agentes etc.) ou outros documentos WWW. Ela inclui um mecanismo de definição de ontologias, instâncias de dados em páginas Web e de classificação hierárquica de documentos HTML. Isto é feito a partir de categorias (classes) e regras que especificam relacionamentos e hierarquias entre instâncias, a partir de um conjunto de *tags* acrescidos ao HTML padrão.

A linguagem XOL é uma linguagem de especificação e intercâmbio de ontologias, especificado em XML/DTD e utiliza um modelo semântico baseado em *frames* denominado

Open Knowledge Base Connectivity – OKBC. Um arquivo XOL consiste de um módulo cabeçalho de definição, que provê meta informação sobre a ontologia, tal como nome e versão; classes e subclasses que permitem estabelecer hierarquias entre categorias de elementos; e *slots* que estabelecem propriedades aos elementos das classes; e definições individuais que permitem declarar nomes, descrições, informações sobre instância, valores, etc. às propriedades dos *slots*.

Na linguagem OIL, uma especificação de ontologias que reúne as seguintes características: primitivas de modelagem normalmente utilizadas em ontologias baseadas em *frames*; possui uma semântica bem definida, simples e clara baseada em descrição lógica; e apresenta suporte para dedução automática. Uma ontologia OIL contém descrições para classes, relacionamentos *slots* e instâncias. As classes podem se relacionar com outras classes através de uma hierarquia (classes/ subclasses) e através de relações binárias estabelecidas entre duas relações. Além disso, restrições de cardinalidade podem ser atribuídas aos relacionamentos.

A definição de uma ontologia na linguagem OIL é constituída de dois componentes: o primeiro que descreve as características da ontologia (*ontology container*), utilizando descritores do padrão *Dublin Core*; e o segundo (*ontology definitions*) que define o vocabulário particular daquela ontologia.

Uma característica importante dessa linguagem é que a mesma pode ser utilizada com a linguagem XML, muito embora esquemas XML não capturem totalmente a semântica embutida no OIL. Porém, sua integração com RDF e RDF Schema é bastante promissora. Do mesmo modo que um RDF Schema é utilizado para se auto definir, o mesmo também pode ser utilizado para definir outras linguagens de ontologia.

A linguagem DAML é uma iniciativa da agência DARPA que está sendo desenvolvida como uma extensão da XML e RDF. A mais recente iniciativa é oriunda da

combinação de DAML e OIL, uma linguagem que está sendo proposta como padrão para representação de ontologias e metadados pela W3C.

A combinação de DAML e OIL, denominada DAML+OIL, sofre muita influência do OIL original, embora não se utilize o seu conceito original de *frames*. É constituído de uma coleção de classes, propriedades (encapsulados numa coleção de axiomas e precedidos pelo *tag daml*) e objetos que são adicionados ao RDF e RDFS. Assim, declarações (*statements*) em DAML+OIL também são declarações RDF.

A linguagem OWL é baseada em *frames* implementando as primitivas definidas pelo OKBC, mas também definições em lógica de descrições. As definições de ontologias são geradas sobre XML e RDF. Na verdade, deriva de um consenso entre duas propostas, a OIL pela camada de inferência e a DAML (*DARPA Agent Markup Language*). Dividida em camadas de acordo com a complexidade. A divisão básica é composta por uma camada de núcleo (*core OIL*), que corresponde a RDFS - porém sem permitir reificação (transformar relações em objetos de uma linguagem), como em RDFS -, e outras camadas para permitir definições compatíveis com *frames* e lógica de descrições, disponível na (URL: <http://www.w3.org/TR/owl-features/>).

Os atributos ou propriedades, podem ter transitividade, simetria, atributos inversos, propriedades funcionais (se $P(x,y) \wedge P(y,x) \Rightarrow x=y$), funcionais inversas (se $P(x,y) \wedge P(z,x) \Rightarrow x=z$) e papéis. Garante completude, decidibilidade e toda a expressividade da lógica de descrições, almejando satisfazer engenheiros de conhecimento familiarizados com esta tecnologia. A expressividade torna-se ainda maior do que em OWL classes podem ser construídas por união, interseção e complemento, pela enumeração de instâncias e podem ter disjunções e a liberdade de usar RDF, inclusive permitindo novas metaclasses, já que elas são subclasses definidas em RDFS. Como pode ser visto no exemplo na figura 19 definição OWL em RDF para uma ontologia simples de uma classe Escola, com duas sub-classes primário e colégio, contendo uma propriedade endereço pertencente a classe escola.

```

<?xml version="1.0"?>
<rdf:RDF
xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
xmlns="http://www.owl-ontologies.com/unnamed.owl#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Ontology rdf:about="">
    <owl:imports
rdf:resource="http://protege.stanford.edu/plugins/owl/protege" />
  </owl:Ontology>
  <owl:Class rdf:ID="Escola"/>
  <owl:Class rdf:ID="Colégio">
    <rdfs:subClassOf rdf:resource="#Escola"/>
  </owl:Class>
  <owl:Class rdf:ID="Primario">
    <rdfs:subClassOf rdf:resource="#Escola"/>
  </owl:Class>
  <owl:DatatypeProperty rdf:ID="Nivel">
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#Primario"/>
          <owl:Class rdf:about="#Colégio"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Nome_Escola">
    <rdfs:domain rdf:resource="#Escola"/>
  </owl:DatatypeProperty>
</rdf:RDF>

```

Figura 19 – Definição OWL em RDF.

4.5.1. A ferramenta Protégé

Na medida em que tais linguagens são utilizadas, tornam-se necessários mecanismos para edição e modelagem de ontologias que, a exemplo de ferramentas Case (as quais possibilitam o uso de ferramentas distintas de modelagem), permitam a utilização de várias linguagens semânticas na Web. O Protégé é uma ferramenta gráfica para a edição e teste de ontologias. Inclui um mecanismo de customização que permite a modelagem conceitual em várias linguagens semânticas, a exemplo de RDF, OIL e DAML+OIL e OWL, como apresentam a figura 20, Protégé versão 3.1.1.

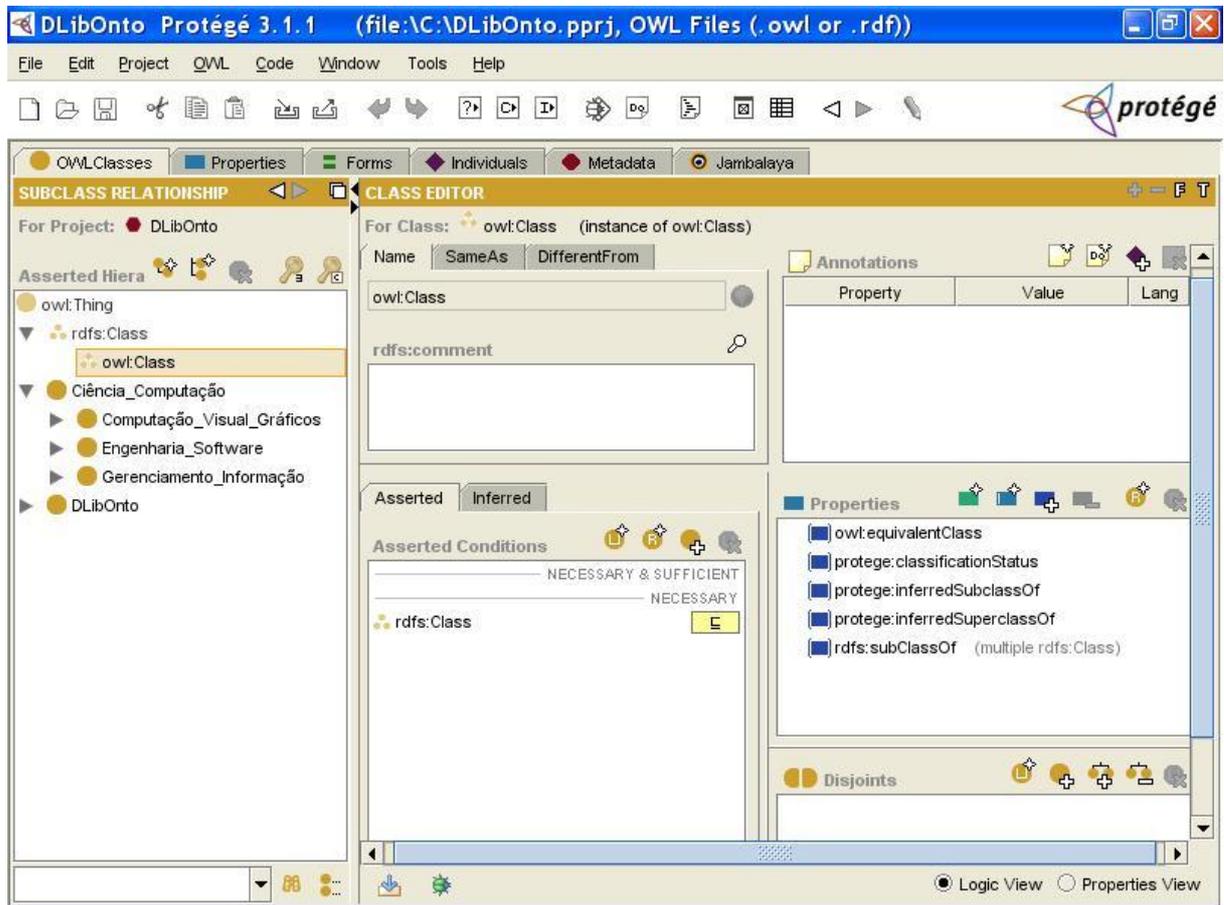


Figura 20 – Protégé na versão 3.1.1. (PROTÉGÉ, 2005).

O Protégé é uma ferramenta desenvolvida pela universidade de Stanford para auxiliar nas construções de bases de dados ontológicas. Baseados nos conceitos da Web Semântica, com geração automática de XML (PROTÉGÉ, 2005). A ferramenta foi desenvolvida com a linguagem Java é totalmente gratuita e livre (*open source*). É disponibilizada gratuitamente na Internet na (URL: <http://protege.stanford.edu/>).

O ambiente Protégé, em seu projeto original era uma ferramenta de aquisição de conhecimento limitada a um sistema especialista para oncologia. Ela foi se modernizando para acompanhar a evolução de tecnologia, para servir na aquisição de conhecimento diretamente dos especialistas de domínios. Com menos dependência de engenheiros de conhecimento, permitir diversos formalismos e estratégias de inferência, integrar tarefas (aquisição de ontologias e instâncias, ambiente de teste com inferência) num mesmo

ambiente, criar automaticamente formulários para entrada de conhecimento, acessar e combinar ontologias.

Pode-se considerar que os requisitos corretos foram levados em consideração durante sua evolução, do ponto de vista de engenharia de software. O Protégé sempre procurou crescer em número de usuários, passando por várias reengenharias e provendo ferramentas simples e configuráveis.

Ao perceber o potencial, a equipe responsável pelo desenvolvimento do Protégé, optou por abrir seu código. Assim, surgiu uma arquitetura integrável a diversas aplicações, via componentes que podem ser conectados ao sistema. Como consequência desta decisão e de sua difusão, componentes de vários matizes, elaborados por grupos de pesquisa de usuários, puderam ser adicionados ao sistema, sem necessitar o re-desenvolvimento. Foram aproveitados, por exemplo, o Jambalaya, um utilitário com animação e vários outros recursos em visualização de dados, e o Ontoviz, um componente que faz com que o gerador de gráficos Graphviz da AT&T, que produz gráficos com instâncias, heranças e outros tipos de relacionamento.

A melhor decisão do projeto Protégé está em seu modelo de conhecimento extensível: é possível redefinir declarativamente as classes primitivas *metaclasses* de um sistema de representação. Com efeito, um atributo de um *frame* numa classe definida por um usuário instância uma classe do sistema, a *metaclass*:*SLOT*, o mesmo ocorrendo com classes e facetas. O conjunto de *metaclasses* usados por default pelo sistema implementa características comuns a *frames*, tornando-o fácil de usar mesmo para usuários leigos. Todavia, se forem utilizadas *metaclasses* complexas e distintas das originais - como para definir classes em RDF, por exemplo - as instâncias alcançarão à expressividade e complexidade desejada. Por isso, o Protégé pode ser adaptado a diversos usos. A seguir algumas características fundamentais do Protégé:

- A linguagem axiomática PAL (*Protégé Axiomatic Language*), que permite a inserção de restrições e axiomas que incidem sobre as classes e instâncias de uma ou mais ontologias;
- A geração de arquivos de saída alteráveis, permitindo que sejam implementados componentes para traduzir o conhecimento para outros formalismos através das metaclasses. Atualmente podem ser criadas classes e instâncias em CLIPS – a base de conhecimento é gerada nativamente para esse motor de inferência. O mais popular quando se iniciou a construção do Protégé, RDF, OIL, XML, que são linguagem padrão ISO para descrever estruturas de conhecimento em páginas da Web através de componentes e metaclasses específicas. A saída pode ser armazenada ainda em outros formatos, como bancos de dados relacionais, ainda que as tabelas geradas não sejam normalizadas;
- Uma excelente interface para entrada de conhecimento, incluindo um gerador automático de formulários para as classes definidas, admitindo ainda a reposição da interface original por componentes mais adequados às aplicações específicas. Esta interface facilita sobremaneira o gerenciamento de conhecimento de uma ou mais ontologias;

Os conceitos de ontologia e suas construções por meio dos *frames*, que irão contribuir muito para a implementação dessa dissertação. Assim como a linguagem OIL e a ferramenta Protégé.

Com o advento da Web Semântica, a área da computação assim como a ciência da informação junta, poderá formatar uma proposta mais coesa, não definitiva, e apresentar algumas sugestões e técnicas, que poderão contribuir para um futuro desenvolvimento de ambas as áreas de forma cooperativa.

De fato, a Web Semântica só poderá de fato ser atingida a partir do interrelacionamento automático de ontologias, desenvolvidas de forma totalmente

independentes e específicas em seus domínios, servindo dessa forma para responder a uma consulta específica. Esse fato talvez justifique o desenvolvimento e proliferação de tantas linguagens e ferramentas para definição de ontologias e representação de conhecimento em torno da Web Semântica.

No capítulo seguinte serão apresentados os conceitos chave dos Web Services, que poderá prover toda a interoperabilidade necessária que uma biblioteca digital de teses e dissertações, ofereça seu acervo digital na Web.

5. WEB SERVICES E ARQUITETURAS DISTRIBUÍDAS

Neste capítulo são apresentados os conceitos de Web services e arquiteturas computacionais distribuídas. Para evidenciar como algumas tecnologias que estão relacionadas promoverão interoperabilidade em ambientes heterogêneos. A seguir é referenciado o motivo da utilização da linguagem Java como plataforma de desenvolvimento do estudo de caso, computação distribuída, construção de componentes, bem como sua utilização nesta linguagem. Quais os principais recursos, protocolos e eventos que poderão ser utilizados, os serviços que poderão ser explorados. Ao final deste capítulo é apresentado o uso destas tecnologias e também os serviços que poderão melhorar a distribuição das informações na Web.

5.1. Web Services

Criado em 1.887 pelo jovem oftalmologista polonês L.L. Zamenhof, o esperanto era a tentativa da criação de uma língua universal, que conseguisse acabar com as barreiras entre os idiomas. Com a necessidade de criar padrões para aplicativos na web, assumindo-os como uma linguagem global da web, o surgimento do *Web Services*, que podem ser, tecnologicamente falando, o esperanto da tecnologia da informação. Entretanto, não se trata de um produto e sim de um conceito, e se encaixa melhor na categoria de serviço.

Os *Web Services* pregam a integração total de sistemas, o que possibilita a conversa entre aplicações desenvolvidas em diferentes plataformas e linguagens, atuando como um tradutor universal ou um idioma que pode ser entendido por instituição, corporação ou usuário qualquer.

Os Web Services sugerem uma proposta para a integração de aplicações, baseada em XML, de modo que as aplicações de componentes possam interoperar de maneira independente de plataforma e de linguagem, em ambientes distribuídos.

A proposta dos Web Services é fundamentada nos seguintes pontos: a) devem ser francamente acoplados, b) devem suportar chamadas remotas de procedimentos, e c) devem oferecer serviços de alto nível, tudo isso se baseando em XML.

Na prática, os *Web Services* podem tratar de uma infinidade de aplicações, que vão desde uma lista de preços pela Internet para os diferentes clientes de uma empresa até a integração completa de toda uma cadeia de fornecedores de varejo. Passa também pelo pagamento automático, dados que vêm à tela. A promessa é que todas as informações contidas nas aplicações possam ser unificadas pela Internet, sem que haja necessidade de migração.

Com *Web Services*, é possível que novas aplicações possam interagir eficientemente e trocar informações entre si, com aquelas que já estão consolidadas, assim como, em sistemas desenvolvidos em plataformas diferentes (NEWCOMER, 2002).

Com Web Services ganhou-se mais importância no desenvolvimento e integração entre sistemas heterogêneos. Isso torna quase imperativo que as novas linguagens de desenvolvimento ofereçam suporte para a nova filosofia. Baseados nos protocolos SOAP, WSDL e UDDI, os Web Services podem ser escritos em praticamente qualquer linguagem. No, entanto, é muito mais prático criá-los e utilizá-los em plataformas que passou ferramentas específicas para esse fim como é o caso do Java e .Net.

Um *Web Services* é qualquer serviço de software que é disponibilizado na internet, que usa um sistema de troca de mensagens padronizadas por XML e que não é vinculado a qualquer linguagem de programação ou sistema operacional (CERAMI, 2002). Como apresentado na figura 21, comunicação entre *Web Services*.

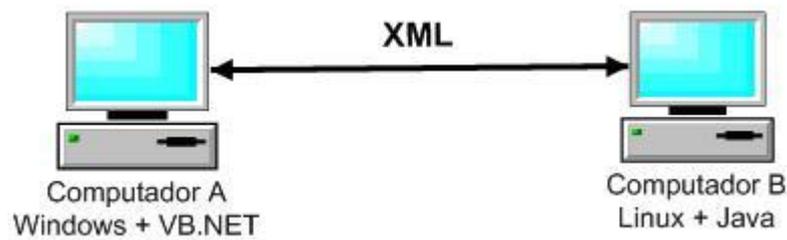


Figura 21 – Comunicação entre *Web Services* (CERAMI, 2002).

O conceito de *Web Services* desde seu início teve o suporte de um consórcio de empresas formado por Microsoft, IBM, Sun Microsystems, BEA Systems, Oracle entre outras. Atualmente *Web Services* é um padrão do W3C (NEWCOMER, 2002).

Este conceito tecnológico foi concebido com a finalidade de proporcionar um meio capaz de disponibilizar via Internet, os métodos para resolução de problemas computacionais. Podem-se proporcionar algumas soluções, oferecendo serviços de software remotamente, ou seja, utilizando esta tecnologia, possibilita a capacidade de desenvolver e disponibilizar componentes de software por meio de algum tipo de rede. Esta rede pode se apresentar como uma Intranet, Extranet ou ainda a Internet.

Na prática, um serviço de software é disponibilizado na Internet, e então ele recebe requisições de um cliente e retorna respostas para o mesmo, ou seja, nenhum detalhe de programação é exposto num *Web Services*, sendo que ele trabalha como uma caixa preta que recebe requisições, as processa e devolve os resultados.

Os clientes (consumidores) de um *Web Services* podem ser aplicações de usuários finais, que utilizam algum *browser* ou mesmo outros *Web Services*. Nesse segundo caso, um *Web Services* pode invocar métodos implementados em um outro, permitindo assim construir uma hierarquia de requisições, e dessa forma, possibilita a criação de serviços bastante complexos, baseados na invocação de outros mais simples (CERAMI, 2002).

Para se utilizar um serviço é necessário que haja uma forma de comunicação com ele. Os Web services utilizam o protocolo *Simple Object Access Protocol* – SOAP, para enviarem e receberem mensagens (CERAMI, 2002).

O uso do protocolo SOAP, é um muito importante na integração entre Web Services, porque além de passar seguramente pela barreira de segurança nos computadores (*firewall*), considerando que as mensagens são arquivos de texto trafegando na rede, a informação é entregue em XML, possibilitando que essa informação seja manipulada de forma conveniente ao consumidor no momento do recebimento da resposta gerada pelo Web Services.

Sempre que um Web Services é criado, deve ser disponibilizada uma descrição pública do mesmo. Esta descrição deve incluir uma documentação legível, para que outros desenvolvedores possam compreender quais são as funcionalidades oferecidas pelo Web Services, para que eles possam desenvolver soluções capazes de se integrar com o serviço criado. “Para que um Web Services seja funcional, ele deve obrigatoriamente ser capaz de se auto-descrever e deve ser publicado” (CERAMI, 2002).

Por meio da descrição do serviço, é apresentado também o conjunto de operações às quais o serviço dá suporte, contendo toda a informação necessária para interação com ele; incluindo o formato das mensagens trocadas, o protocolo de transporte utilizado e sua localização. Esta interface esconde os detalhes de implementação do serviço, permitindo sua utilização independentemente da plataforma de hardware ou software na qual ele é implementado ou da linguagem na qual ele é escrito. Como comentado anteriormente, todo este conjunto de informações é estruturado no padrão XML, auxiliando sua descrição “de posse desse arquivo de descrição, o cliente pode conhecer os detalhes dos métodos oferecidos pelo Web Services e gerar interfaces de acesso aos mesmos em tempo de execução” (CERAMI, 2002).

Para que o Web Services possa interagir com outras aplicações, ele deve de alguma forma publicar na rede para que sua descrição seja compartilhada. A padronização das

descrições de *Web Services* se dá por meio da *Web Services Description Language* - WSDL, que é a especificação que define como descrever *Web Services* em uma gramática XML padronizada.

Por fim, a principal tecnologia utilizada para publicar *Web Services* atualmente é o *Universal Description, Discovery and Integration* - UDDI, que pode ser visto como o padrão para registrar, descrever e localizar *Web Services* utilizando uma base de registro compartilhada entre os diferentes clientes.

Uma vez encontrado o *Web Services* apropriado, o cliente contata e invoca seus métodos. Na figura 22, ilustra o processo de registro, pesquisa e invocação de um *Web Services*.

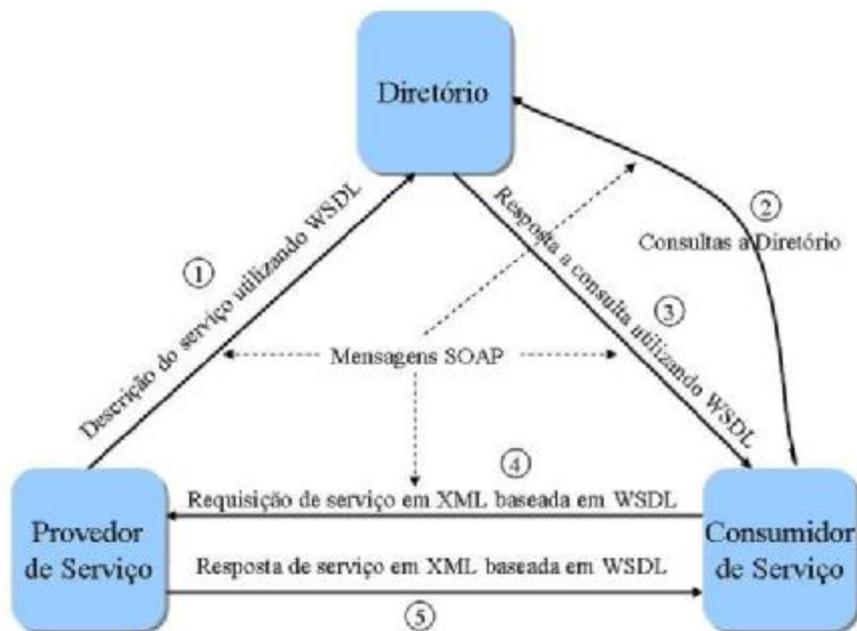


Figura 22 – Registros, pesquisa e consumo de *Web Services* (CERAMI, 2002).

O UDDI define a maneira de publicar e recuperar informações sobre os serviços Web. O termo “*Web Services*” descreve uma funcionalidade específica de uma organização, geralmente exposta por meio de uma conexão de Internet, que tem o objetivo de prover uma forma para que outra organização ou programa de computador possa usar o serviço. O UDDI

usa uma abordagem que conta com um registro distribuído de organizações e a descrição de seus serviços implementados em um formato comum que faz uso do XML.

5.2. Protocolo SOAP

A área de atuação de um protocolo SOAP é bastante abrangente, ele pode ser compreendido de diversas maneiras corretamente. É um protocolo de ligação, um mecanismo de RPC, um padrão de interoperabilidade, um protocolo de troca de documentos e é a linguagem universal para comunicações B2B (*Business to Business*) (CHAPPELL and JEWELL, 2002).

Em outras palavras, SOAP é um protocolo que permite acessar objetos através da Internet, que foi projetado para invocar aplicações remotas através de RPC ou para trocas de mensagens em um ambiente independente de plataforma e linguagem de programação. Sendo assim, SOAP é um padrão normalmente usado para comunicação entre Web Services, que garante a interoperabilidade e intercomunicação entre diferentes sistemas, por meio da utilização de XML e mecanismos de transporte padrões.

As principais características do SOAP são: um padrão definido pelo W3C baseado em XML, simples, leve, extensível e usa uma mensagem SOAP. Este protocolo talvez seja a mais importante das tecnologias que dão suporte a Web Services, isso porque não seria possível exibi-los, se não houvesse uma representação abstrata de dados e publicação de definições de interface (NEWCOMER, 2001).

Uma mensagem SOAP é um documento XML composto de três elementos principais: um *header*, que é o cabeçalho da mensagem e é opcional, o *body*, que é corpo da mensagem e um invólucro chamado de *envelope*, que encapsula os dois primeiros elementos e que contém dados da versão e formato de mensagem. Os componentes de uma mensagem SOAP são ilustrados graficamente na figura 23.

```

<? Xml version="1.0"?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
  Soap:encodingStyle="http://www.w3.org/2001/12/soap-
encoding">
  <soap:Header>      .....      </soap:Header>
    <soap:Body>      .....      ...
      <soap:Fault>    .....    <s/oap:Fault>
    </soap:Body>
  </soap:Header>
</soap:Envelope

```

Figura 23 – Envelope SOAP.

Os componentes de um protocolo SOAP são:

- **Envelope:** este é o elemento raiz da mensagem SOAP, sendo que o envelope pode conter declarações de *namespaces* e atributos adicionais também, como por exemplo, o *encoding style*, que é quem define o estilo de codificação do documento;
- **Header:** o cabeçalho é opcional na mensagem SOAP. Ele pode conter dados informativos quaisquer sobre a mensagem. Caso seja utilizado, o *Header* deve ser o primeiro elemento do *Envelope*.
- **Body:** Este elemento é obrigatório e contém a informação a ser transportada para o destinatário da mensagem, sendo que o elemento *Body* pode conter um elemento opcional *Fault*, usado para carregar mensagens de status e erros retornadas pelos destinatários ao processarem a mensagem.

Na figura 2.4 é apresentado um exemplo de código já estruturado de uma mensagem SOAP.

```

<env:Envelope xmlns:env="http://www.w3.org/2001/12/soap-
envelope">
  <env:Header>
    <n:broadcastService
xmlns:n="http://www.xmlbus.com/broadcastServices">
      <n:list> PDA, Cell, Email, VoiceMail, IM </n:list>
    </n:broadcastService>
  </env:Header>
  <env:Body>
    <m:Function
xmlns:m="http://www.xmlbus.com/broadcastServices/send">
      <m:message>Apresentação do TCC da
FEMA</m:message>
    </m:Function>
  </env:Body>
</env:Envelope>

```

Figura 24 – Mensagem SOAP.

O exemplo da figura 24 representa uma mensagem de único sentido para uma lista de mecanismos de comunicação. O bloco *Header* contém a lista dos dispositivos para os quais a mensagem é enviada. O bloco *Body* contém a mensagem que é entregue aos destinatários, que é um lembrete de uma reunião.

Os *namespaces* *broadcastService* e *function* são utilizados para determinar escopos à mensagem. O envelope referencia a versão 1.2 do *namespace Envelope*, <http://www.w3.org/2001/12/soap-envelope>.

5.3. Linguagem WSDL

A WSDL descreve o que o serviço executa no momento de invocar suas funções e qual a sua localização (CHAPPELL e JEWELL, 2002).

Em outras palavras, pode se dizer também que a descrição do Web Services representa um contrato entre o consumidor do serviço e o provedor do mesmo, visto que WSDL é um aspecto fundamental na arquitetura de Web Services, porque provê uma linguagem para descrição de serviços e uma plataforma para integração.

A linguagem WSDL é a especificação que define como Web Services devem ser descritos em uma gramática XML comum. Ela estabelece quatro pontos principais para descrição de Web Services (CERAMI, 2002).

- Interface de informação que descreva todas as funções disponíveis;
- Informação de tipos de dados para todas as mensagens de requisição e resposta;
- Informação sobre o protocolo de transporte a ser usado;
- Informação de endereço para localização do serviço especificado.

Tudo o que é definido em um arquivo WSDL é abstrato, sendo este arquivo simplesmente trás as definições de parâmetros e restrições para que a comunicação possa ser realizada em tempo de execução (CHAPPELL e JEWELL, 2002).

Os elementos da WSDL contêm descrições de funções e formatos presentes no serviço, tipicamente usando um ou mais XML Schemas que são trocados entre o provedor e o consumidor do serviço para que seja possibilitada uma comunicação entre eles. Normalmente, WSDL utiliza SOAP (NEWCOMER, 2001).

Esta especificação foi desenvolvida conjuntamente entre Microsoft, Ariba e IBM, sendo que a versão 1.1 da especificação foi submetida ao W3C, que a aceitou e a publicou em seu Web site W3C-WSDL. Vinte e duas outras empresas se juntaram a esta submissão, consistindo o maior número de membros que relataram conjuntamente uma submissão ao W3C (2005).

Ambas as partes que participam de uma comunicação entre Web Services devem ter acesso à mesma WSDL para habilitar a compreensão de informações pertinentes ao serviço. Em outras palavras, tanto o invocador quanto o receptor de uma mensagem envolvida em uma

interação entre Web Services devem ter acesso ao mesmo XML Schema. O invocador da mensagem deve conhecer qual o formato correto para o envio de mensagens, enquanto que o receptor deve saber como interpretar as mensagens recebidas corretamente. Com isso, as implementações dos Web Services podem ser ocultadas. Esta é a chave do sucesso do WSDL, ele prove um formato comum para codificação e decodificação de mensagens entre o serviço e virtualmente qualquer aplicação de *back-end*, tais como CORBA, EJB, JMS, sistemas de ERP e outras mais em geral (NEWCOMER, 2001).

5.4. Registro UDDI

O *Universal Description, Discovery and Integration* - UDDI surgiu de uma iniciativa da indústria de software, que visa estabelecer um padrão para descrição, descoberta e integração de serviços independentemente de plataforma. UDDI é focado no processo de descoberta de serviços no contexto da arquitetura orientada a serviços (CHAPPELL e JEWELL, 2002).

O UDDI é uma especificação para registros distribuídos com informações dos *Web Services*. É também um conjunto publicamente acessível de instâncias da especificação, que permite que organizações registrem informações sobre seus serviços Web, para que outras organizações possam pesquisá-los e utilizá-los. Os registros são usados para promover e descobrir estes serviços Web distribuídos.

O componente essencial do projeto UDDI é o registro de negócios UDDI (*Business Registration*), um arquivo XML usado para descrever uma entidade de negócio e seu serviço Web. Conceitualmente, a informação presente em um registro de negócio UDDI consiste de três componentes: páginas brancas (*white pages*) que inclui endereço, contato e identificadores conhecidos; páginas amarelas (*yellow pages*) que inclui uma categorização industrial baseado em taxonomias padrões; e páginas verdes (*green pages*) que inclui a informação técnica sobre o serviço sendo oferecido.

O esquema XML do UDDI define quatro tipos de informação que provê a variedade de informação, que um técnico necessitaria para poder usar um serviço Web. São elas: informações do negócio *business information*, informações do serviço *service information*, informações para a conexão *binding information* e informações sobre especificações para serviços.

As informações do registro de negócios do UDDI, as páginas amarelas, estão contidas numa estrutura XML chamada "*business Entity*". Essa estrutura serve como o nível mais alto de um conjunto de informações sobre uma unidade de negócio. As descrições técnicas e do tipo de negócio do serviço Web, as páginas verdes, estão presentes em subestruturas do "*business entity*", o "*business service*" e o "*binding template*". A estrutura "*business service*" é o local que é usado para agrupar uma série de serviços web relacionados a um processo de negócio ou uma categoria de serviço. Junto com o "*business service*" está uma ou mais descrições técnicas de serviços Web, são as estruturas do tipo "*binding template*". Elas contêm informações que são relevantes para programas que desejam se conectarem com um serviço Web remoto. Cada "*binding template*" contém um elemento especial chamado "*tmodel*", que é uma lista de referencias com informações sobre especificações que formam uma impressão digital do serviço Web.

Usando o serviço do UDDI, organizações podem registrar individualmente seus serviços Web. Esta informação pode ser adicionada ao registro de negócios UDDI, por meio da página ou por um programa que use a interface do UDDI. Esse registro é logicamente centralizado, mas é fisicamente distribuído. Ou seja, ele tem uma interface de consulta centralizada, porém os registros são replicados por vários nós. Uma vez registrada, a informação sobre o serviço Web é replicada para esses nós e fica disponível para consulta.

O UDDI é um serviço de descoberta que trabalha como um repositório de metadados para registrar e armazenar informações de descrição e localização de Web Services. O próprio repositório UDDI é implementado pelo Web Services, cujos métodos oferecidos são o

cadastro e a pesquisa no repositório de metadados. Dessa forma, clientes exploram o repositório UDDI, pesquisando todos os Web Services disponíveis.

UDDI é um conceito simples que implementa um registro distribuído entre muitos “nós”, que replicando se entre si, compartilham informações sobre serviços. O registro (ou diretório) de serviços UDDI, que é hospedado por diferentes repositórios na Internet, foi criado para facilitar e resolver o problema de descoberta de serviços entre empresas que realizam transações entre si (CHAPPELL e JEWELL, 2002).

O registro UDDI é para a arquitetura de *Web Services* o que é para o catálogo telefônico ou então um caderno de classificados, que são lugares onde se pode procurar por assuntos de interesse e obter informações de contato. O conceito apresentado até este momento, é exposto na figura 25 que mostra o esquema de funcionamento do registro UDDI.

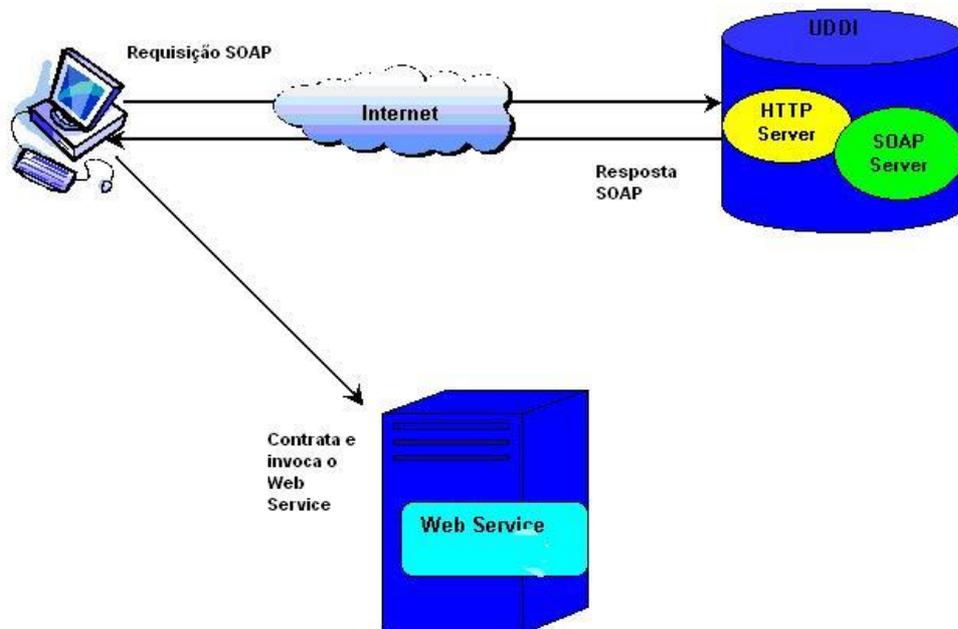


Figura 25 – Registro UDDI (CHAPPELL e JEWELL, 2002).

O potencial consumidor busca no registro UDDI informações de localização do serviço desejado por meio de uma requisição SOAP. Assim que recebe esta requisição, o UDDI busca em suas bases a localização do serviço solicitado encontrando-a, retorna esta

informação através de uma resposta SOAP. Com posse da localização do Web Services na Web, o consumidor inicia a transação com este serviço.

A especificação do UDDI consiste num XML *Schema* para as mensagens SOAP, e uma descrição da especificação da interface do UDDI. Juntos, eles formam um modelo básico de informações e uma arquitetura de interação que provê a habilidade para publicar dados de uma grande variedade de serviços web. O padrão de esquema XML foi escolhido pelo seu suporte a uma grande variedade de tipo de dados, bem como sua habilidade de facilmente descrever e validar informações baseadas em modelos de informação.

Desta forma, retendo a imagem dos Web Services, como um meio de aproximar diferentes aplicações e sistemas, está convicto que a adoção destes por parte da comunidade bibliotecária traria como consequência um aumento significativo da interoperabilidade entre os seus sistemas. Assim, melhorar significativamente os serviços já existentes, como o acesso aos registros bibliográficos e a informação relacionada com as obras do acervo digital.

As organizações buscam um consenso da indústria para facilitar as tomadas de decisão e incrementar a adoção das soluções, além de garantir a evolução contínua das tecnologias que suportam os Web Services. Fazem parte do consórcio de empresas como IBM, *BEA Systems*, *SUN* e *Novell*, todos partidários da linguagem Java. O W3C pretende desenvolver a interoperabilidade de especificações, software e ferramentas para o máximo aproveitamento das ferramentas baseadas em Internet.

O SOAP serve para a troca de informações em ambientes distribuídos; enquanto o WSDL atua como uma linguagem de descrição dos serviços e a UDDI é um conjunto de registros com função de "páginas amarelas" de serviços disponíveis.

Uma grande vantagem dos *Web Services* reside no fato da equipe de desenvolvimento poder focar seus esforços no sistema em si, praticamente sem se preocupar com meio de comunicação entre os processos. É neste ponto que o Java oferece muitas vantagens em relação às demais plataformas. Criado para garantir suporte e integração com os

Web Services, ele pode publicar um método de uma classe, bastando adicionar um modificador (atributo) ao mesmo. Com o Java não é necessário criar o WSDL ("contrato de especificação"), pois o mesmo é criado em tempo de execução e automático, utilizando a "reflexão". Isto significa que o WSDL estará sempre atualizando Java.

5.5. Tecnologia Java Server Pages

Apesar da criação de um Web Services não ser realizado de uma forma tão simples, o Java oferece mais opções para permitir as customizações dos aplicativos gerados. É possível controlar o WSDL e o SOAP construído, bastando utilizar outros modificadores dos métodos, podendo configurar o *namespace*, a utilização de sessão e a segurança. Outro ponto importante no ambiente é o compartilhamento de informações entre os Web Services e aplicações de Internet desenvolvidas em *Java Server Pages* - JSP. É possível compartilhar dados da sessão pelas variáveis de aplicação.

A JSP é uma tecnologia de geração dinâmica de documentos HTML ou XML que integra a plataforma Java. Um JSP é um arquivo texto com código HTML ou XML e *tags* JSP específicos, que podem ou não conter código Java, para a geração dinâmica do documento.

Quando uma aplicação cliente requisita a página JSP ao servidor Web pela primeira vez, a página é passada primeiramente para o *JSP Engine*, que gera automaticamente um *servlet* a partir do JSP e o executa retornando o resultado para o cliente. Nas próximas requisições ao JSP, o *servlet* é acionado diretamente.

Os elementos de um JSP de um código JSP podem ser compostos por:

- **Diretivas:** comandos que definem informações globais para a página;
- **Declarações:** declarações de variáveis e métodos com escopo da página;
- **Scriptlets:** trechos de código Java inseridos na página;

- **Expressões:** *strings* para serem incluídas na página resultante do processamento do JSP;
- **Ações:** comandos que executam ações durante o processamento do JSP.

Além disso, existem configurações avançadas no Web Services que podem ser utilizadas como: controle de buffer, utilização de transações e *cache*. Esta última é extremamente importante: é possível utilizar um controle de cacheamento das respostas. Mais uma vez, todas as configurações são feitas utilizando modificadores das classes e métodos dos *Web Services*. Esta variedade fornece um controle completo do funcionamento dos aplicativos sem abrir mão da facilidade e rapidez de implementação.

Todas essas vantagens existem quando se cria um *Web Services*, juntamente com o Java que também fornece várias ferramentas para facilitar a sua utilização. Para tal, é necessária a criação de um servidor de *proxy* (um computador que tem a função de gerenciar uma área de acesso rápido, informações já acessadas, normalmente de lugares cuja conexão é lenta) utilizando o seu WSDL. Java traz uma ferramenta que o cria automaticamente. Basta chamar na mesma hora de oferecer a *url* do *Web Services*, o *proxy* é criado automaticamente. Feito isto, basta compilar o código gerado e utilizar os métodos da plataforma.

Um dos grandes trunfos dos *Web Services* está na possibilidade de comunicação entre sistemas de várias plataformas e linguagens de forma transparente e simples, mas nem sempre o uso da plataforma é a melhor opção. Além do completo suporte a *Web Services*, a plataforma Java também oferece um recurso chamado *Remoting*.

O principal objetivo de qualquer plataforma intermediária (*middleware*) é a redução da complexidade na integração de sistemas e, por conseguinte criar uma homogeneidade artificial no seio de um conjunto de sistemas heterogêneos. Exemplos de plataformas de *middleware* são: o *Common Object Request Broker - CORBA*, o *Java 2 Enterprise Edition - J2EE*.

Este recurso tem o mesmo princípio dos *Web Services*, mas com a vantagem de permitir a escolha dos protocolos a serem utilizados. Enquanto a plataforma utiliza HTTP e SOAP, o Remoting permite a utilização de troca de mensagens binárias diretamente sobre o protocolo TCP/IP, além de utilizar o HTTP e o SOAP. Isto significa sacrificar a portabilidade dos serviços e em troca de um grande ganho de performance, além de ser ainda mais fácil de ser utilizado, pelo fato do *proxy* ser gerado automaticamente sem a necessidade do WSDL.

5.6. Computação distribuída e componente

O desenvolvimento de aplicações distribuídas é bastante complicado. Tão complexo que é necessário utilizar soluções adaptáveis, independentes de plataforma e desenvolver com simplicidade. As soluções que são descritas a seguir estão se tornando mais atraentes e necessárias em um número cada vez maior de aplicações corporativas e acadêmicas.

Para se obter vantagens da computação distribuída, ou seja, tolerância à falhas, melhor desempenho via paralelismo e melhor aproveitamento do hardware; todavia, poderá ser adquirida a partir de uma linguagem que ofereça os recursos apropriados para se tirar proveito destas vantagens.

Uma plataforma de linguagem que poderá prover tais objetos distribuídos, por sua vez, é um tipo de modelo de programação de aplicações distribuídas que diferem de modelos clássicos. Como acesso remoto ao banco de dados, invocação de métodos remotos e troca de mensagens, já que pode interagir com um objeto localizado em uma máquina remota como se ela fosse local e ao construir um objeto em uma dada máquina, pode-se transmitir para outra.

Algumas vantagens de utilização de objetos distribuídos são as abstrações (não há necessidade de conhecer os detalhes de implementação, apenas sua interface), a independência de localização e nomeação de recursos (acesso aos objetos independe da localização física e de informações de endereçamento), modularidade (todo objeto é entidade

autocontida). Por sinal, é relevante ressaltar a possibilidade de reutilização de bibliotecas pré-existentes com serviços e facilidades para plataformas distribuídas.

As arquiteturas de suporte a objetos distribuídos são constituídas por: a) mecanismo para especificação das interfaces dos objetos e geração de skeletons e stubs; b) gerente de objetos; c) um serviço de nomes; d) um protocolo de comunicação entre objetos remotos e uma rede para comunicação.

A tecnologia Java foi projetada visando também o ambiente de rede, com isso, as vantagens com a independência de plataformas, carregamento dinâmico de código por meio da rede e bibliotecas poderosas para especificação de arquiteturas distribuídas (ARNOLD e GOSLING, 1998).

Quando se apresenta objetos distribuídos em Java, duas soluções são apontadas inicialmente: *Enterprise Java Beans* – EJB e *Java Remote Method Invocation* - RMI. Ambos prometem a distribuição de serviços, no entanto, ainda estão se consolidando nos ambientes corporativos.

Em especial, o ambiente J2EE constitui uma das principais tecnologias utilizadas para o desenvolvimento de aplicações distribuídas e a tecnologia é ideal para a implementação de Web Services.

No próximo capítulo é apresentado o projeto da DLibOnto. Primeiramente é feita uma descrição completa dos objetivos da DLibOnto, e todas as etapas de importância para a implementação do projeto: arquitetura Web, arquitetura interação e Web services, ontologia DLibOnto, apresentação dos RDFs Schemas utilizando a linguagem OWL, definida com o auxílio da ferramenta Protégé, análise orientada a objetos e os resultados obtidos com a ontologia. Mostra-se também o projeto completo da arquitetura até a construção da biblioteca digital de teses e dissertação DLibOnto.

6. BIBLIOTECA DIGITAL DLibOnto

Vários usuários que atualmente utilizam à Internet realizam suas pesquisas na intenção de obter mais conhecimento, não imagina como se procede a conectividade, ou como se comportam os sistemas operacionais e equipamentos computacionais que são utilizados para oferecer essas facilidades. Sistemas que pertencem a grandes e pequenas empresas consorciadas, universidades públicas e privadas, enfim provedores com configurações de todos os tipos, conectadas por cabos coaxiais, fibras ótica e até mesmo sem fio (*wireless*), evidenciando grande complexamente interligação destas redes.

Recuperar rapidamente dados, dinamizar as buscas, manuais, documentos, imagens, som e informações, tem sido o grande interesse das bibliotecas digitais, independente do tipo de equipamento e conexão, realizados de Internet, por meio de PDA, Palmtop, Servidores ou microcomputadores. Facilitar os processos de busca e recuperação, tornando possível a interoperabilidade entre os *Web Services*. Por intermédio da interoperabilidade entre os *Web Services* está à chave para a troca permanente de informações e conhecimentos.

Isso não significa que os usuários consigam buscar e recuperar os assuntos de interesse com tanta facilidade. Mostra o quão difícil é buscar e recuperar a informação que deseja. Horas podem ser desperdiçadas na busca incansável para localizar, sem mesmo saber se vai obter o resultado esperado. Muita paciência e resignação são alguns dos adjetivos dados aos usuários pesquisadores da Internet.

A prioridade da biblioteca digital de teses e dissertações DLibOnto é a gratuidade de acesso aos documentos digitais, pelo simples conceito de disseminação de conhecimento de forma livre, não permitindo que os documentos digitais dos autores e colaboradores sejam alterados livremente, apenas acessados.

6.1. Arquitetura Web distribuída da DLibOnto

Partindo da necessidade, juntamente com a intenção, qualquer instituição ou usuário, justifica-se o uso de qualquer equipamento com acesso a Internet. Pressupondo que um usuário que esteja realizando alguma pesquisa, trabalhando, em casa ou no escritório, deseja receber informações sobre um assunto ou artigo que foi deixado disponível na biblioteca digital de teses e dissertações. Basta apenas que o usuário faça um acesso à DLibOnto de qualquer parte do mundo, desde que o equipamento esteja conectado na Internet, realizar uma busca por meio de campos no formulário específico.

Estas informações apresentadas são as mesmas informações solicitadas por qualquer outro usuário, localizado em qualquer outra região, que esteja fazendo acesso ao mesmo tempo. É mais cômodo para um docente, porque estas buscas auxiliarão os discentes em suas pesquisas. As referências utilizadas como material de apoio às pesquisas, levando em consideração à ética das mesmas utilizadas pelos alunos. Este mecanismo abre um portal de comunicação quase direta entre usuários, pelo intermédio da DLibOnto.

Uma instituição, seja ela privada ou pública, proverá suas informações, sendo elas teses e dissertações por meio da DLibOnto. Estes documentos de pesquisas deverão estar em formato digital contendo texto e imagens, permitindo que o conhecimento e as experiências adquiridas nos departamentos de pesquisa possam ser compartilhados com outros usuários. Os documentos poderão conter restrições, se estiverem no formato (PDF).

Segurança na recuperação de informações fica a cargo da arquitetura Web. Logo após a documentação estar finalizada pelo autor, é enviado para um processo computacional que armazena as informações em Bancos de Dados e o conteúdo completo anexado em diretórios específicos para segurança, organização e recuperação.

Esta arquitetura computacional está representada na figura 26, Arquitetura da DLibOnto na Web, apresentando sua arquitetura distribuída. Os processos evidenciam que a

interligação virtual proporcionada pela Internet favorece a conectividade de qualquer usuário e outros servidores Web.

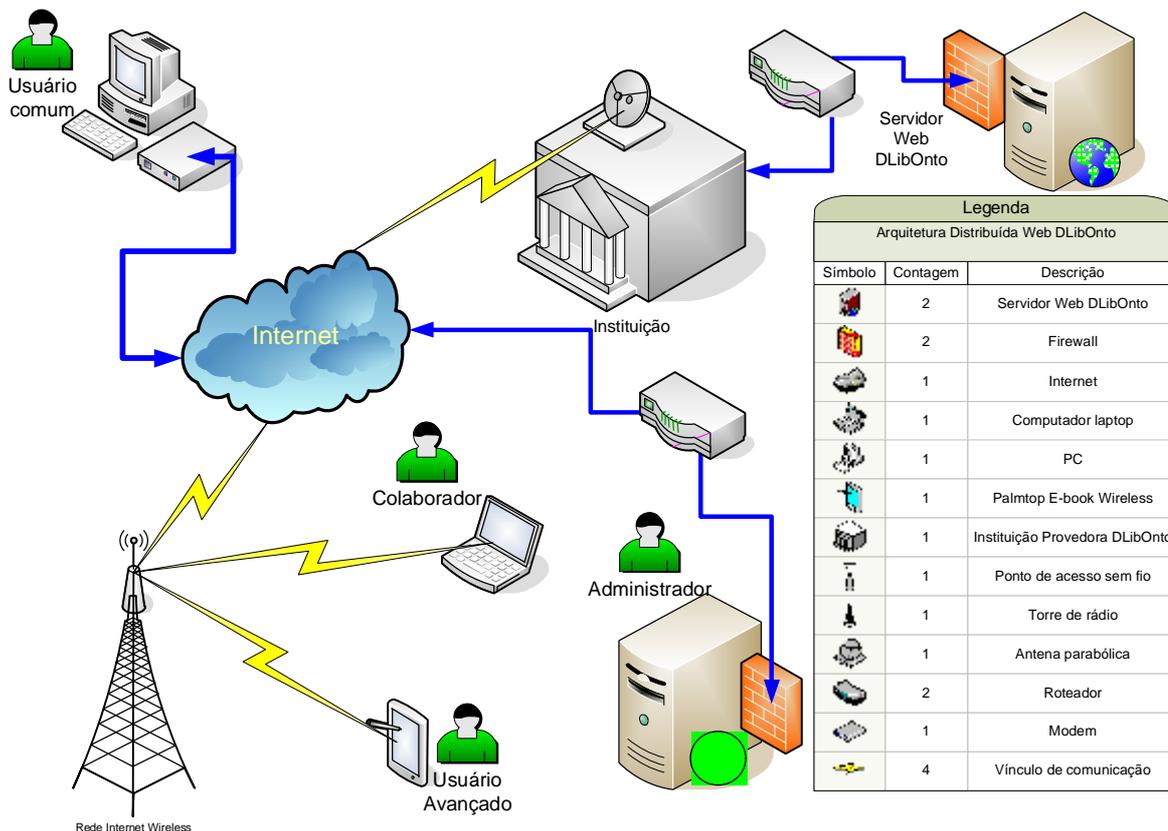


Figura 26 – Arquitetura da DLibOnto

As teses e dissertações estarão disponíveis nos servidores Web, onde estiver instalado a DLibOnto em formato específico com o mesmo padrão de configuração, mantendo a mesma estrutura ontológica, contendo os metadados específicos, oferecendo aos usuários o acesso à meta linguagem e as informações necessárias para realizar uma análise mais crítica e apropriada, como é apresentado na próxima seção, ontologia da DLibOnto.

Esta integração das bases de informações se define por interoperabilidade, desencadeando uma troca das bases de dados, sendo distribuída para todos que venham compartilhar das mesmas intenções. Outro aspecto importante para esta interoperabilidade é o acesso ao dado por qualquer tecnologia de microcomputador que utiliza para conectividade a Internet, não necessariamente por meio de um *Web Service* da DLibOnto.

Como já foi apresentado no capítulo cinco, os serviços UDDI, associado ao protocolo SOAP, está devidamente configurado no Web Service da DLibOnto, para que as informações ali depositadas pelo Sistema Web DLibOnto esteja pronto para ser consumido por outros Web Services. Lembrando que esta troca de informações é feito do modo não assíncrono, bastando apenas que agentes computacionais específicos dos próprios Web Services de busca realizem sua função.

Esta arquitetura supre a necessidade da consulta de qualquer dispositivo que esteja conectado a Internet. Bastará apenas que esses dispositivos façam o acesso a Internet, por meio de uma rede de comunicação, cabos metálicos, fibras óticas ou mesmo sem fio. Execute a procurar através de um site de busca específico. Que no momento que o site de busca tiver a referência do link dos documentos da DLibOnto, ele recupera o link certo e correspondente da mantenedora do documento original. Esta possibilidade está exemplificada também na figura 26, no momento em se observa a instituição com um Web Services DLibOnto em funcionamento.

Podem ser executadas tarefas como foram citadas acima por meio de laptop em meio à plantação de uma fazenda no Mato Grosso, ou mesmo por um professor da área de telecomunicações por meio de Palmtop em plena Avenida Paulista. Até mesmo uma secretária que não terminou de ler um artigo ou continuar a ler enquanto viaja para a Índia, por meio desta biblioteca digital.

6.2. Web Services da DLibOnto

Foram utilizados alguns recursos necessários e outros indispensáveis para a constituição deste Web Services. Dentre elas a arquitetura J2EE, em razão de vários aspectos citados no capítulo cinco anteriormente. Java oferece segurança, estabilidade e confiabilidade em relação a sistemas e aplicativos construídos para acesso a Internet.

Servidores de Internet poderão assumir a função da estrutura para a implementação dos sistemas que irão desempenhar a função de gerenciador dos dados, documentos, segurança e da semântica da DLibOnto. Neste servidor foi instalado softwares e aplicativos específicos para o acesso à Internet, devidamente configurado para atender o dinamismo das páginas, segurança e cacheamento de memória. Os requisitos básicos foram:

1. Sistema Operacional LINUX – DEBIAN
2. Servidor WEB Apache – TOMCAT
3. Banco de Dados – POSTGRES
4. J2EE – Java para soluções para Web Services
5. Firewall nativo DEBIAN, para segurança nos protocolos de acesso
6. RDFs Schemas das Ontologias DLibOnto
7. Repositório UDDI configurado, para manter os XML, RDFs e PDF.

A tecnologia envolvida foi Java J2EE, porque ela dispõe de um servidor apropriado para o gerenciamento de tais informações (*Java Web Services*). Este servidor Java prove automaticamente páginas dinâmicas desenvolvidas com a tecnologia JSP (*Java Server Pages*), acessando o Banco de Dados (*Postgres*), apropriado para tal necessidade. Permitindo maior flexibilidade aos desenvolvedores Web.

O acesso a DLibOnto é realizado totalmente por computador, via Web Browser. Para os administradores que mantém o cadastro das teses e dissertações, até os usuários que irão acessar o acervo digital por meio da Internet. A página de acesso a DLibOnto, estará disponível o dia todo, sete dias da semana, todos os meses, desde que o servidor seja mantido em funcionamento. Com uma estrutura de backup bem definida pelos administradores de rede, com um sistema de redundância instalado no próprio servidor Web, mantendo desta forma o acervo digital sempre disponível. A figura 27, que está na próxima página representa a estrutura de hardware e serviços que mantém.

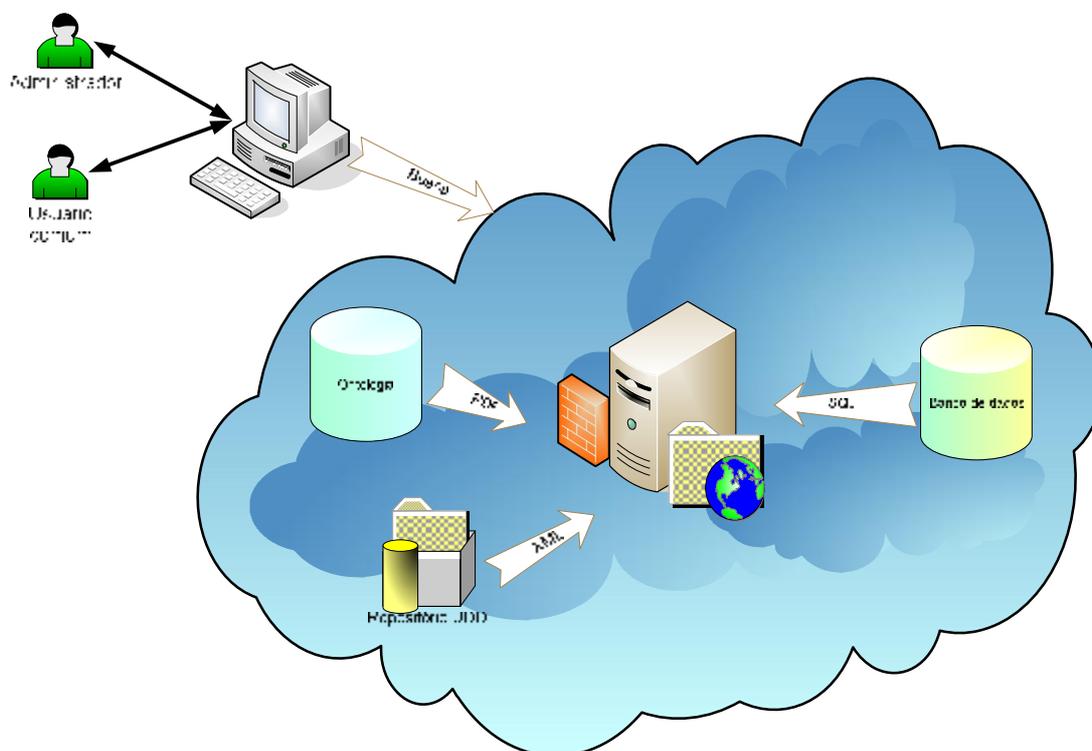


Figura 27 – Estrutura de Hardware e conectividade da DLibOnto.

Observando-se a figura acima pode concluir que qualquer usuário conectado a Internet poderá fazer uma pesquisa na DLibOnto diretamente no site, por meio de busca específica da biblioteca digital. Claro que a exatidão na procura, vai depender das características de informações fornecidas pelo usuário.

Estas duas arquiteturas apresentadas irão proporcionar ao usuário um aumento na sua base de dados pessoais contidos em seu microcomputador para futuras pesquisas. Ou simplesmente para referenciar algumas pesquisas em trabalhos próprios, até para outros pesquisadores que irão utilizar-se das mesmas referências. Outros poderão simplesmente transferir estas informações para os microcomputadores de outras pessoas ou empresas associadas a um grupo ou de outra biblioteca digital.

6.3. Definição dos serviços da DLibOnto

O modo de interação, de qualquer usuário com o sistema é realizado por meio de computadores, acessando o mesmo Web site. Por meio desta página dinâmica que estará disponível no Web Service DLibOnto, o usuário poderá interagir com seus serviços.

O sistema oferece duas interfaces distintas aos seus usuários. A primeira interface está diretamente relacionada às atribuições dos administradores do acervo digital. A outra está para os usuários comuns, que serão os consumidores do acervo digital. A interface dos administradores é responsável por cadastrar, gerenciar e manter as informações atualizadas e pertinentes aos documentos digitais.

A primeira interface é única para os administradores e para os usuários comuns. Assim, os administradores ou usuários comuns utilizarão uma interface de busca para localizar os documentos digitais dentro do acervo digital. Esta procura é realizada no banco de dados do servidor daquela DLibOnto especificamente. Ainda nesta interface é apresentada uma possibilidade ao administrador, de fazer um acesso restrito ao acervo, que lhe dará permissões para realizar as devidas atualizações.

Para os usuários comuns a interface de busca é livre, mas restrito, não permitindo o acesso as rotinas pertinentes aos administradores. Desta forma cabe aos usuários comuns fazer as buscas somente aos documentos ali cadastrados e também aos documentos que poderão ser apenas referenciados a outras bibliotecas digitais.

Para que o administrador tenha acesso à área restrita, um login e senha serão solicitados para a autenticação de entrada. Passada esta etapa pode-se realizar as ações de manutenção nos documentos.

Ainda na interface do administrador, têm-se alguns eventos importantes que diferenciam da interface dos usuários comuns. A seguir é apresentada a figura 28, contendo a

interface da DLibOnto. Desta forma a arquitetura de acesso é feita por qualquer dispositivo conectado na Internet, cujo acesso é por meio do Web Site da DLibOnto.

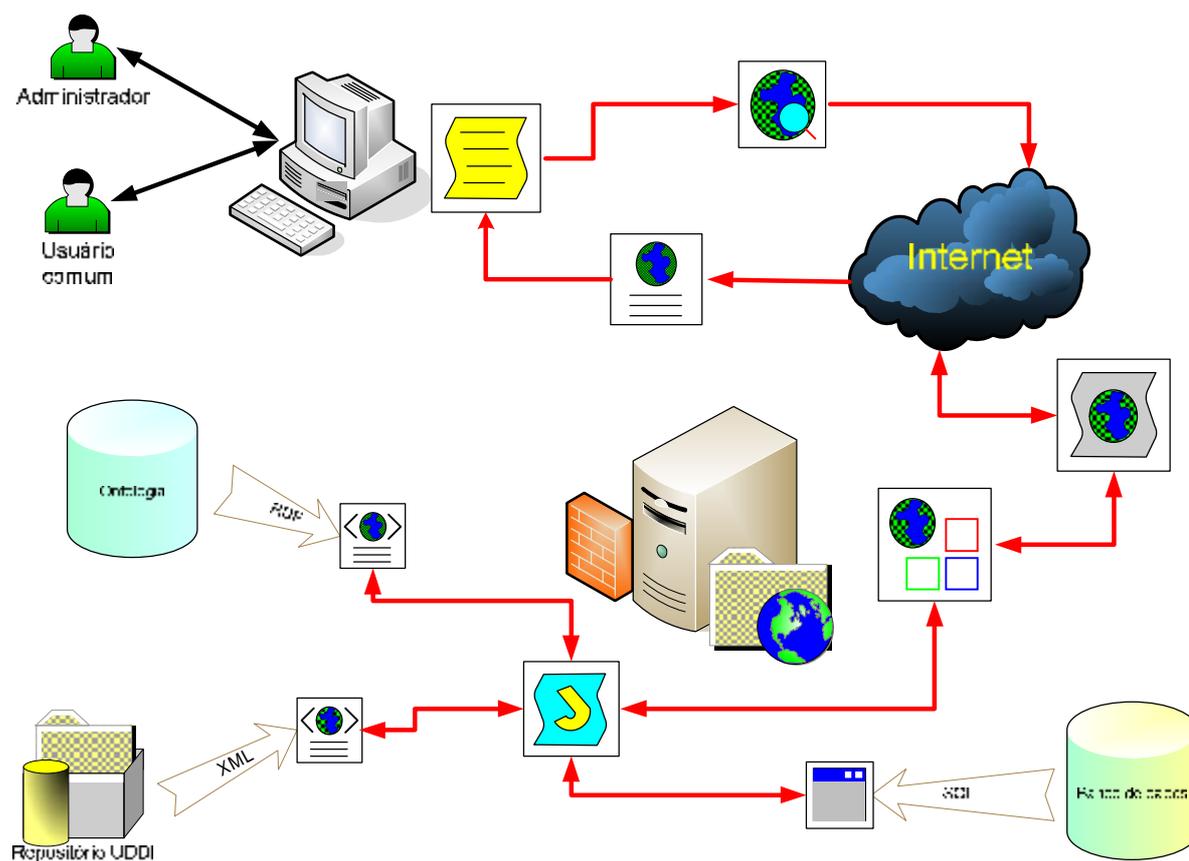


Figura 28 – Interfaces da DLibOnto.

A figura acima representada indica a estrutura de funcionamento das interfaces dos usuários. Os requisitos que são importantes para a construção do sistema e do funcionamento da DLibOnto. A seguir a figura 29, está à legenda, mostrando os ícones para a identificação dos processos executados.

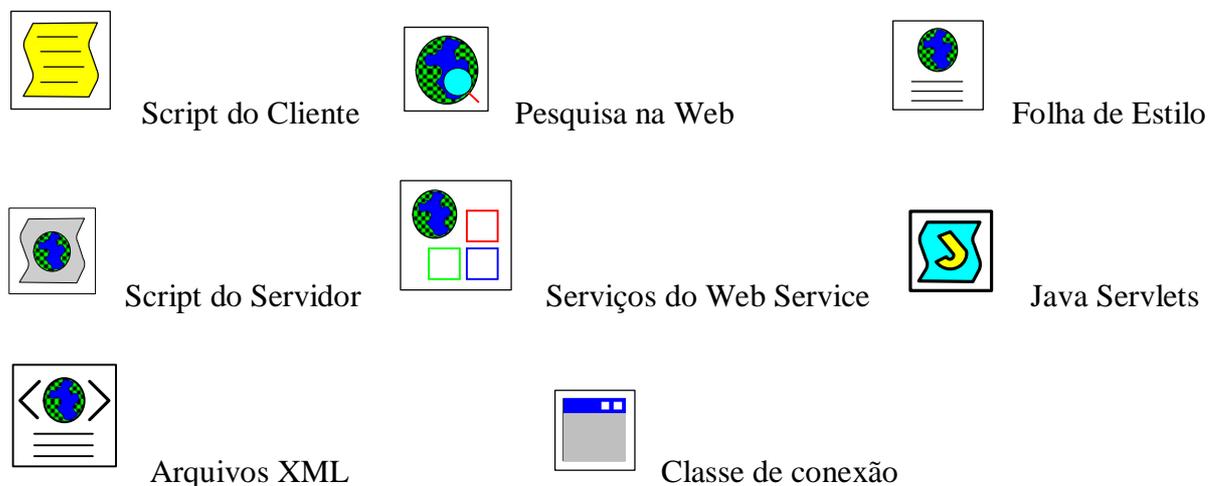


Figura 29 – ícones usados na interface DlibOnto.

A facilidade de acesso e interação são os objetivos principais a serem alcançados pelo sistema da DLibOnto. Neste esforço, deixar o acesso a todo o acervo digital cadastrado, no formato XML. Os motivos do uso do XML é a interação que poderá ser feita com outros sistemas computacionais do próprio usuário.

Aos administradores a possibilidade de cadastrar um documento digital novo no acervo, por meio de uma tela simples contendo todos os campos necessários para o preenchimento das informações pertinentes ao documento. Em outra tela, o administrador acessa a interface de busca dos documentos, exatamente igual à interface de usuários comuns, mas com a possibilidade de fazer a manutenção das informações que forem necessárias. Ao clicar no link de “manutenção” do documento digital em questão, o documento é recuperado do banco de dados, para a tela de cadastro com os campos correspondentes já preenchidos, para realizar uma eventual alteração.

Depois de realizado o cadastramento ou atualização das informações do documento do acervo digital, o aplicativo envia as informações para o banco de dados e realiza a geração ou atualização das mesmas informações no formato XML correspondente ao documento. O arquivo XML é depositado na pasta URI de endereço do Web Services, que o identificará e disponibilizará seu conteúdo para outros Web Services. Este arquivo é criado respeitando a

ontologia (RDF Schema), já pré-definido pelos administradores e armazenado no Web Services DLibOnto.

Após os arquivos XML gerados respectivamente para cada documento digital, o sistema acrescenta as informações para a localização destes documentos digitais dentro do arquivo que mantém a ontologia atualizada. Outros Web Services da Internet poderão fazer uso destas referências, que estão dentro deste arquivo.

Os usuários comuns que fizerem o acesso na página de busca da DLibOnto, dispara os serviços java (*servlet*) para a busca de informações de dados enviados pelo usuário no formato XML. Como já apresentado anteriormente no capítulo 4, o XML permite armazenar os dados de forma estruturada que poderão conter as informações mais bem definidas para uma busca mais refinada. Nessa transformação as informações de busca estarão estruturadas de forma atender esta requisição de serviço.

Para realizar busca por outros Web Services, a busca é interpretada pelos protocolos SOAP, utilizando a linguagem WSDL nos registros UDDI do Web Services da DLibOnto. Evidenciando que seus documentos estarão no formato RDF Schema do IBICT – Padrão de metadados de teses e dissertações, que irão funcionar como estrutura base do domínio de conhecimento.

Logo após a transformação dos dados, a busca é realizada em um banco de dados e RDF, devidamente preparado para retornar uma resposta rápida ao usuário. Se as informações foram encontradas no banco de dados, estes serão os metadados que irão passar por uma nova transformação dos dados em metalinguagem, obedecendo às regras do RDF Schema e uma folha de estilo. Desta forma os dados serão transferidos como resposta para o serviço (*servlet*) que foi disparado anteriormente.

6.4. Ontologia da DLibOnto

Na construção da ontologia da DLibOnto, foram usados os conceitos básicos, como já foi comentado no capítulo quarto. É levado em consideração um vocabulário de conceitos, definições e possíveis propriedades, que é representado por um modelo gráfico mostrando as possíveis relações entre os conceitos, forma mais clara e não ambígua. É importante realçar que, de posse dessa base de conhecimento formalizada como uma teoria lógica, a ontologia não descreve apenas conhecimento imediato.

Foram criadas duas ontologias, uma ontologia de domínio que expressa conceitos de domínio particular, descrevendo um vocabulário relacionado a um domínio mais genérico. Esta ontologia foi baseada no relatório definido em 2001 criados por uma força tarefa entre a IEEE e a ACM que se apresenta no artigo (COMPUTER CURRICULA, 2001).

A ontologia de aplicação expressa os conceitos dependentes do domínio e da tarefa particular da DLibOnto chamada JBDigital. Esses conceitos frequentemente correspondem aos papéis desempenhados na instituição. Esta ontologia foi baseada no MTD-BR do padrão brasileiro de metadados de teses e dissertação do IBICT.

Por questões de flexibilidade e dinamicidade as duas ontologias foram criadas separadamente, para permitir que a ontologia de domínio possa ser atualizada ou alterada totalmente sem ter que mexer na estrutura da ontologia de aplicação. Para que esta flexibilidade fosse alcançada foi feito um relacionamento entre as classes dessas ontologias. As classes que foram definidas na ontologia de domínio da área da Ciência da Computação, para este projeto da biblioteca digital de teses e dissertações, fecharam-se um escopo simplificado, envolvendo a Engenharia de software, Computação Virtual e Sistema de Informação, com algumas especializações.

Abaixo a figura 30, ontologia DLibOnto, representa a relação que existe entre as duas ontologias da Ciência da Computação e JBDigital pela classe <Programa>, respectivamente pela propriedade <Área>.

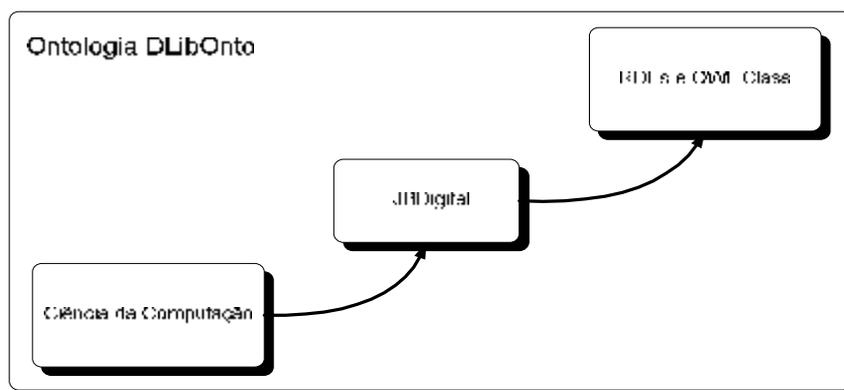


Figura 30 – Ontologia DLibOnto.

O significado dos termos encontrados na Ciência da Computação está definindo no relacionamento entre os objetos e classes, subclasses e classes-pai. Esses sistemas são denominados taxonomia. Esse tipo de ontologia normalmente é estabelecido por sistemas orientados a objetos. Muitas ontologias existentes são definidas usando-se apenas esses relacionamentos hierárquicos. Relacionamentos simples entre categorias na taxonomia podem ser usados para relacionar e combinar dados. Portanto, os dados são inteligentes o suficiente para serem facilmente descobertos e sensivelmente combinados com outros dados. A seguir é apresentada na Tabela 1, a taxonomia da Ciência da Computação (GUARINO, 1999).

Tabela 1 – Taxonomia da Ciência da Computação

Ciência_Computação
Sistema de Informação
Armazenamento Informação Recuperação
Banco Dados Relacional
Bancos Dados Distribuído
Bibliotecas Digitais
Digitalização, armazenamento e intercâmbio.
Objeto Digital, combinação e pacotes.
Metadados, catalogação, Submissão Autores.
Nomeando, repositórios, arquivos.
Espaços (conceitual, geográficos, 2/3D, Realidade Virtual)
Arquiteturas (Agentes, Barramentos, Empacotadores, Mediadores), Interoperabilidade)
Serviços (Busca, ligação)

Gerenciamento dos direitos da propriedade intelectual, privacidade, proteção
Arquivamento e preservação, integridade
Hipermídia Hipertexto
Linguagem Consulta Banco Dados
Mineração Dados
Modelagem Dados
Processamento Transações
Projeto Banco Dados Físico
Projeto Banco Dados Relacional
Sistemas Banco Dados
Sistemas Informação Multimídia
Sistemas Modelos Informação
Computação Virtual e Gráficos
Fundamentos Técnicos em Gráficos
Sistemas Gráficos
Comunicação Gráfica
Modelagem Geométrica
Renderização Básica
Renderização Avançada
Técnicas Avançadas
Animação Computacional
Visualização
Realidade Virtual
Stereoscopic display
Force feedback simulation, haptic devices
Viewer tracking
Collision detection
Visibility computation
Time-critical rendering, multiple levels of details (LOD)
Image-base VR system
Distributed VR, collaboration over computer network
Interactive modeling
User interface issues
Applications in medicine, simulation, and training
Visão Computacional
Engenharia Software
Projeto Software
Usando API's
Ambientes Ferramentas Software
Processos Software
Especificações Requisitos Software
Validação Software
Evolução Software
Gerenciamento Projeto Software
Computação Baseada Componentes
Metodos Formais
Reusabilidade Software
Desenvolvimento Sistemas Especialistas

A seguir está a representação gráfica contendo a ontologia da JBDigital que apresenta todas as classes, contendo suas propriedades, evidenciando a ligação das superclasses com as subclasses correspondentes.

A relação existente entre as duas ontologias, foi feita por meio da classe Programa, através da propriedade Área. Que faz a ligação direta com ontologia da Ciência da Computação. Esta relação permite que no momento do cadastramento do documento a área de concentração da tese e dissertação, terá que ser escolhida de acordo com a leitura da ontologia de domínio que foi previamente definida pelos administradores da DLibOnto da instituição. A tabela 2, representa a taxonomia da JBDigital.

Tabela 2 – Taxonomia da JBDigital

JBDigital							
			Título				
			Idioma				
			Grau				
			Titulação				
			Resumo				
			Cobertura				
			Assunto				
			Data Defesa				
			Direitos				
Controle	Biblioteca Digital	Biblioteca Depositária	Local Defesa	Autor	Contribuidor	Programa	Agência Fomento
Nome	Nome	Nome	Cidade	Nome	Nome	Nome	Nome
Sigla	Sigla	Sigla	UF	Citação	Citação	Área	Sigla
URL	URL	URL	País	Lattes	Lattes		País
URL	URL	URL		CPF	CPF		UF
Documento	Documento	Documento					CNPJ
							URL
	Provedor Serviço			Afiliação	Afiliação	Instituição	
	Nome			Nome	Nome	Nome	
	Sigla			Sigla	Sigla	Sigla	
	País			País	País	País	
	UF			UF	UF	UF	
	CNPJ			CNPJ	CNPJ	CNPJ	
	URL			URL	URL	URL	

O construtor Protégé foi o aplicativo utilizado para a construção dessas ontologias. Além das facilidades de construção das ontologias deste aplicativo, suas ferramentas adicionais permitiram algumas validações e testes. Para realizar o teste foram cadastrados 10 teses e dissertações numa base de dados contendo os objetos e atributos respectivamente na JBDigital. A base de dados mantida pelo Protégé já está no formato XML e RDF. Foi possível realizar uma validação na entrada de dados, com formulários específicos criados dentro do Protégé. No momento da digitação foi possível verificar a taxonomia da ontologia de domínio JBDigital, sendo apresentada para a escolha da área específica.

Os dados são armazenados nos arquivos XML/RDFs específicos, já apresentando sua verdadeira característica ontologia desta ferramenta. Em formulário específico de busca, a ferramenta Protégé através de um motor, realizou a busca semântica entre as *tags* e propriedades do RDFs *Schema*. Recuperando as informações necessárias e precisas dentre as teses e dissertações que foram cadastradas.

Evidenciando a grande importância destas ontologias relacionadas para esta arquitetura. Porque desta ontologia sai a base do domínio do conhecimento gerado como regra para o filtro do Sistema Web DLibOnto.

O Sistema ainda oferece também uma proposta mais concreta, respeitando os órgãos competentes que já apresentam regras, já aprovadas por comitês responsáveis. Um destes órgãos é o IBICT – Instituto Brasileiro de Informação em Ciência da Informação em Ciência e Tecnologia, que aprovou uma lista de elementos do *Schema*, como padrão Brasileiro de Metadados de Teses e Dissertações. Desta forma, ficou claro que neste estudo de caso são adotados os padrões para a interação com qualquer outra biblioteca digital de teses e dissertações no âmbito nacional.

A partir do modelo de elementos do IBICT foi possível criar a estrutura do arquivo XML, que é gerado no momento do cadastramento da documentação no acervo digital, respeitando o RDF Schema da JBDigital. A tabela 3, têm a lista dos elementos de metadados.

Tabela 3 – Padrão Brasileiro de Metadados de Teses e Dissertações

Nro.	Nome elemento	Conteúdo	Padrão
1.	<Controle>	Dados do controle do registro de teses e dissertação	
1.1	<Sigla>	Sigla da Instituição Consorciada	
1.2	<DataAtualizacao>	Data em que foi realizada a última atualização do registro da tese ou dissertação	Protocolo OAI
1.3	<IdentificacaoDocumento>	Código que identifica a tese ou dissertação na base de dados da biblioteca	
1.4	<Tipo>	Tipo da fonte da informação	DCMI Type Vocabulary
2.	<BibliotecaDigital>	Nome da Instituição responsável pela publicação digital da tese ou dissertação	
2.1	<Nome>	Nome da Biblioteca Digital	
2.2	<Sigla>	Sigla da Biblioteca Digital	
2.3	<URL>	Endereço eletrônico do site da Biblioteca Digital	Padrão URI
2.4	<URLDocumento>	Endereço eletrônico da tese ou dissertação na Biblioteca Digital	Padrão URI
2.5	<ProvedorServiço>	Instituição que abriga a Biblioteca Digital	
2.5.1	<Nome>	Nome da Instituição	
2.5.2	<Sigla>	Sigla da Instituição	
2.5.3	<Pais>	País da Instituição	ISO 3166
2.5.4	<UF>	UF da Instituição	Unidade da Federação
2.5.5	<CNPJ>	CNPJ da Instituição	Padrão Ministério da Fazenda
2.5.6	<URL>	Endereço eletrônico do site da Instituição	Padrão URI
3.	<BibliotecaDepositaria>		
3.1	<Nome>	Nome da Biblioteca Depositária	
3.2	<Sigla>	Sigla da Biblioteca Depositária	Cadastro CCN/COMUT
3.3	<URL>	Endereço eletrônico do site da Biblioteca Depositária	Padrão URI
3.4	<NumeroChamada>	Código que identifica a tese ou dissertação no acervo da biblioteca depositária	
4.	<Titulo>	Título da tese ou dissertação	
5.	<Idioma>	Idioma da tese ou dissertação	ISO 639
6.	<Grau>	Grau acadêmico associado à tese ou dissertação	Tabela de Grau
7.	<Titulacao>	Nome do grau acadêmico associado à tese ou dissertação	
8.	<Resumo>	Resumo da tese ou dissertação	
9.	<Cobertura>	Escopo espacial ou temporal da tese ou dissertação	
10.	<Assunto>	Tópicos tratados na tese ou dissertação e a tabela de onde estes tópicos foram extraídos, quando for o caso	
11.	<LocalDefesa>	Local de defesa da tese ou dissertação	
11.1	<Cidade>	Cidade onde foi defendida a tese ou dissertação	
11.2	<UF>	UF da cidade onde foi defendida a tese ou dissertação	Unidade da Federação
11.3	<Pais>	País onde foi defendida a tese ou dissertação	ISO 3166

12.	<DataDefesa>	Data em que foi defendida a tese ou dissertação	ISO 8601
13.	<Autor>	Autor da tese ou dissertação	
13.1	<Nome>	Nome da pessoa física	
13.2	<Citacao>	Forma pela qual a pessoa deseja ser citada	
13.3	<Lattes>	Endereço eletrônico do currículo da pessoa na plataforma Lattes	Padrão URI
13.4	<CPF>	CPF da pessoa	Padrão Ministério da Fazenda
13.5	<Afiliação>	Instituição à qual a pessoa é afiliada	
13.5.1	<Nome>	Nome da Instituição	
13.5.2	<Sigla>	Sigla da Instituição	
13.5.3	<Pais>	País da Instituição	ISO 3166
13.5.4	<UF>	UF da Instituição	Unidade da Federação
13.5.5	<CNPJ>	CNPJ da Instituição	Padrão Ministério da Fazenda
13.5.6	<URL>	Endereço eletrônico da Instituição	Padrão URI
14.	<Contribuidor>	Contribuidor da tese ou dissertação e forma de participação (papel)	
14.1	<Nome>	Nome da pessoa física	
14.2	<Citacao>	Forma pela qual a pessoa deseja ser citada	
14.3	<Lattes>	Endereço eletrônico do currículo da pessoa na plataforma Lattes	Padrão URI
14.4	<CPF>	CPF da pessoa	Padrão Ministério da Fazenda
14.5	<Afiliação>	Instituição à qual a pessoa é afiliada	
14.5.1	<Nome>	Nome da Instituição	
14.5.2	<Sigla>	Sigla da Instituição	
14.5.3	<Pais>	País da Instituição	ISO 3166
14.5.4	<UF>	UF da Instituição	Unidade da Federação
14.5.5	<CNPJ>	CNPJ da Instituição	Padrão Ministério da Fazenda
14.5.6	<URL>	Endereço eletrônico da Instituição	Padrão URI
15.	<Programa>	Programa de pós-graduação onde a tese ou dissertação foi defendida	
15.1	<Nome>	Nome do Programa de Pós-graduação	Tabela CAPES
15.2	<Area>	Área de concentração do Programa de Pós-graduação	Classes DLibOnto
15.3	<Instituicao>	Instituição que oferece o curso de Pós-graduação	
15.3.1	<Nome>	Nome da Instituição	
15.3.2	<Sigla>	Sigla da Instituição	
15.3.3	<Pais>	País da Instituição	ISO 3166
15.3.4	<UF>	UF da Instituição	Unidade da Federação
15.3.5	<CNPJ>	CNPJ da Instituição	Padrão Ministério da Fazenda
15.3.6	<URL>	Endereço eletrônico do site da Instituição	Padrão URI
16.	<AgenciaFomento>	Agência de financiamento que apoiou financeiramente o autor	
16.1	<Nome>	Nome da Instituição	
16.2	<Sigla>	Sigla da Instituição	
16.3	<Pais>	País da Instituição	ISO 3166

16.4	<UF>	UF da Instituição	Unidade da Federação
16.5	<CNPJ>	CNPJ da Instituição	Padrão Ministério da Fazenda
16.6	<URL>	Endereço eletrônico da Instituição	Padrão URI
17.	<Direitos>	Informa as condições de distribuição, reprodução e utilização da tese ou dissertação	

6.5. Análise da DLibOnto

Para um melhor entendimento do projeto da DLibOnto, um modelo de negócio foi elaborado baseado nos requisitos levantados de acordo com um ambiente simplificado de uma biblioteca digital de teses e dissertações. Para tanto, é representado por um diagrama que representa melhor o objetivo do sistema, evidenciando que o projeto é de acesso livre a qualquer instituição. A figura 31, apresenta o modelo de negócio da DLibOnto.

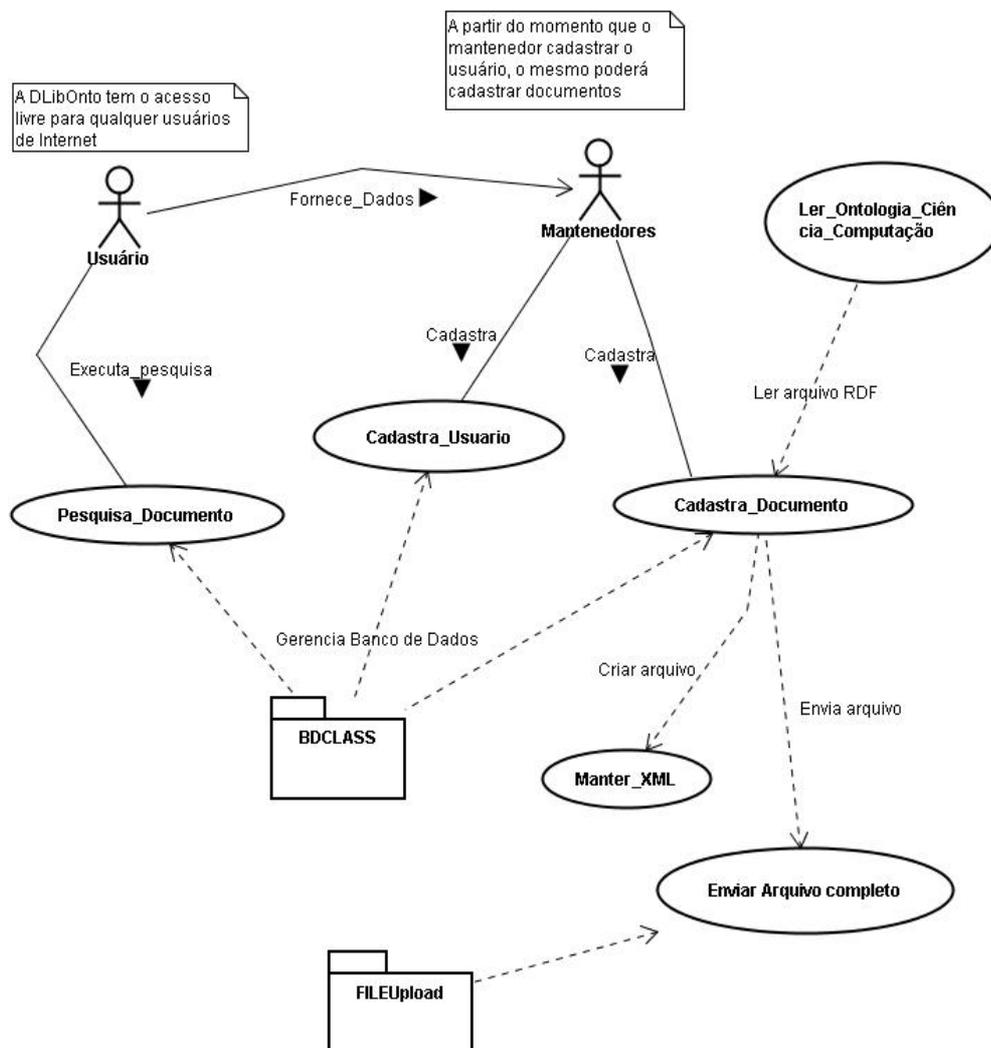


Figura 31 – Modelo de negócio da DLibOnto.

O diagrama anterior mostra exatamente quais serão os atores, os eventos, os pacotes que serão utilizados. Neste modelo, pode ser visto que alguns atores terão atribuições diferentes em relação ao outro.

Como já foi comentada anteriormente, a DLibOnto é um sistema Web com interface de uso diretamente no Web Browser, sendo utilizada e administrada pela mesma. Assim faz-se necessários apresentar outro diagrama que representa esta interação de interfaces, que apresenta as páginas que foram construídas com este objetivo. A seguir a figura 32, mostrar o diagrama de interfaces Web da DLibOnto.

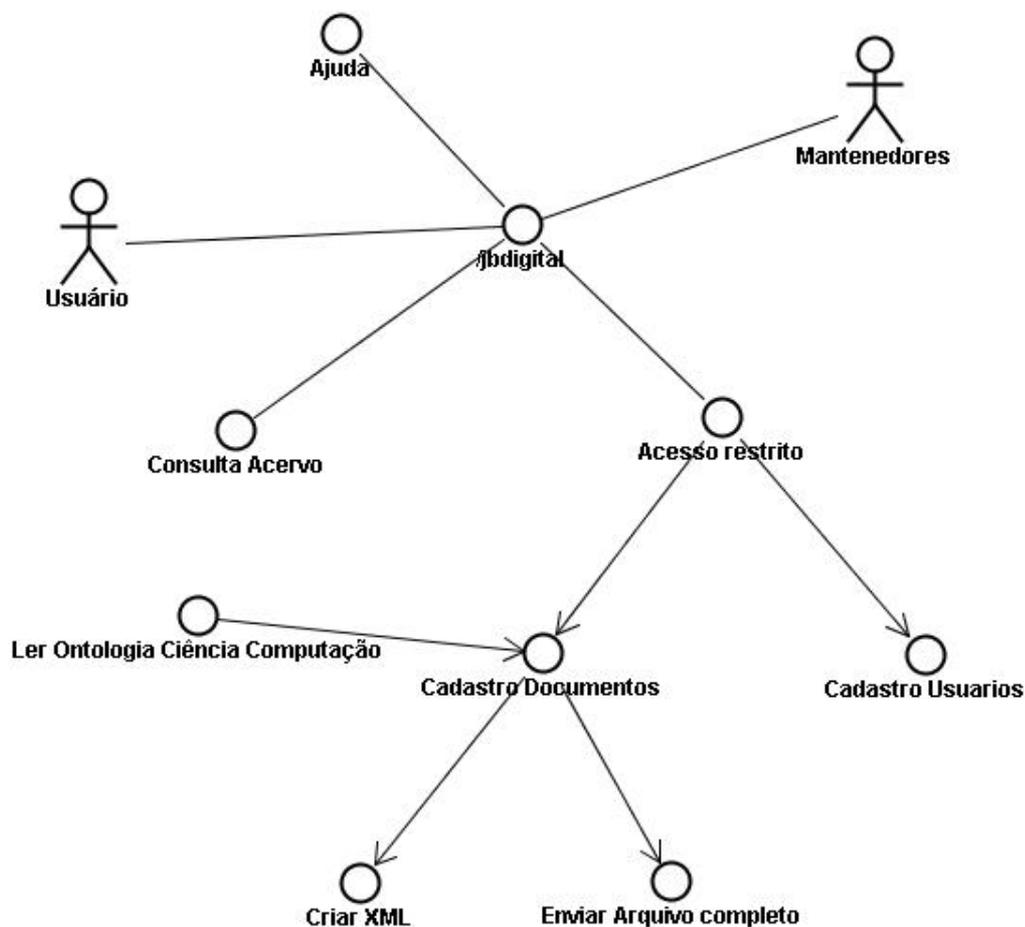


Figura 32 – Diagrama de interfaces da DLibOnto.

O diagrama de classes apresenta a forma concisa da manipulação de informações que a DLibOnto executa no momento que ela é utilizada. Neste diagrama é possível verificar como as informações irão fluir dentro do Web Services. Ainda neste diagrama poderão ser visualizados os eventos que ela realiza, na figura 33, que mostra o diagrama de classes.

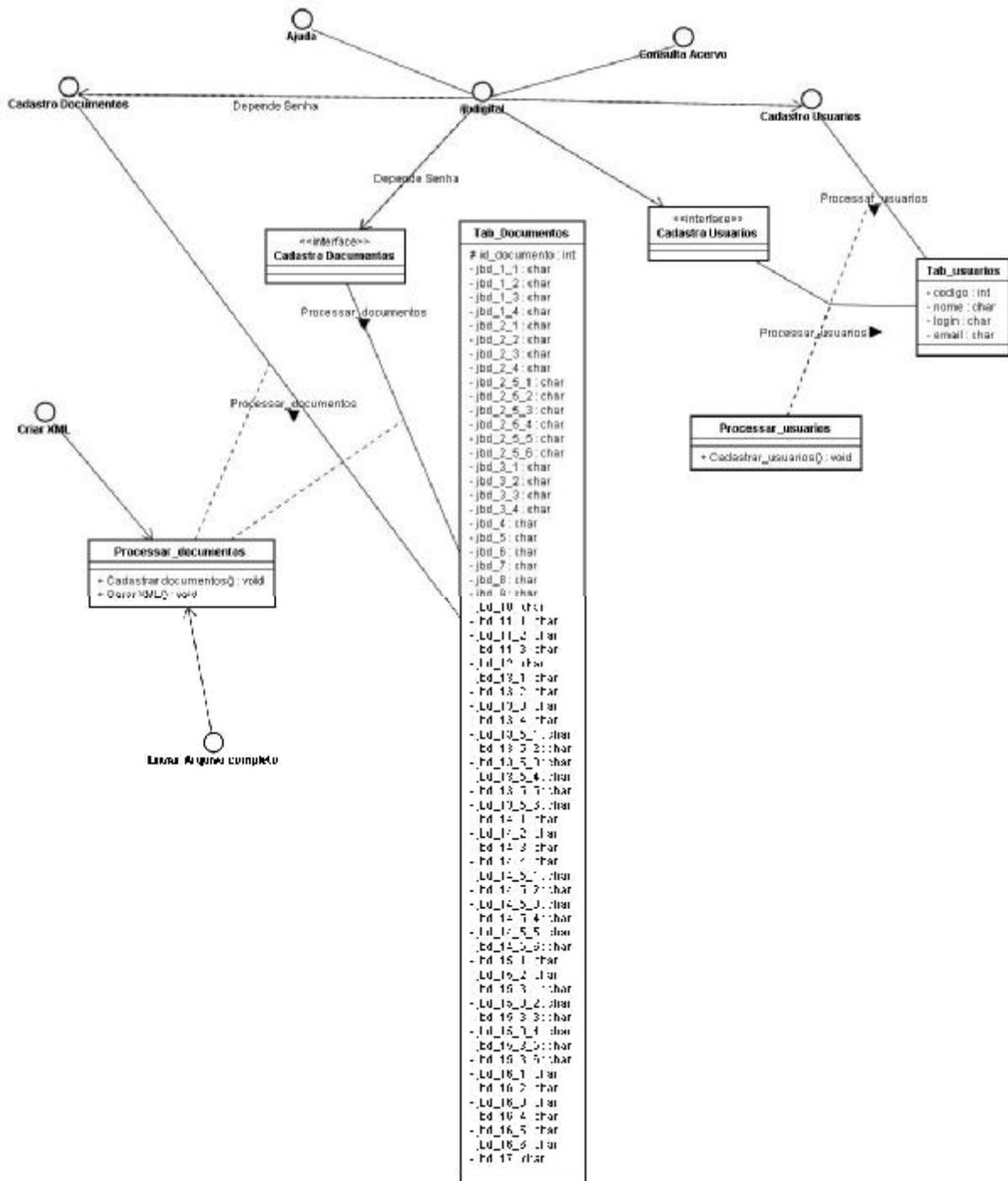


Figura 33 – Diagrama de classes da DLibOnto.

O próxima figura, o diagrama dos componentes que a JB-digital utiliza para realizar as operações de processamento dentro do Web Service. Estes componentes se tornaram necessários pelo fato de não precisar construir rotinas que já foram implementadas.

Realizando desta forma um reaproveitamento de um código já testado com sucesso, na figura 34 o diagrama de componentes da DLibOnto.

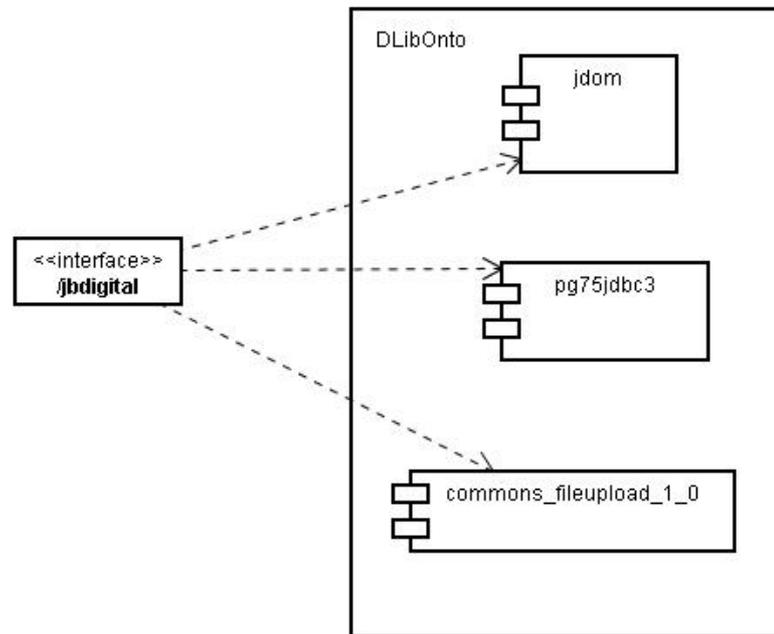


Figura 34 – Diagrama de componentes da DLibOnto.

A seguir são apresentados os diagramas de seqüência para melhor representar os eventos que são realizados pelos usuários. A figura 35 é a seqüência da página principal.

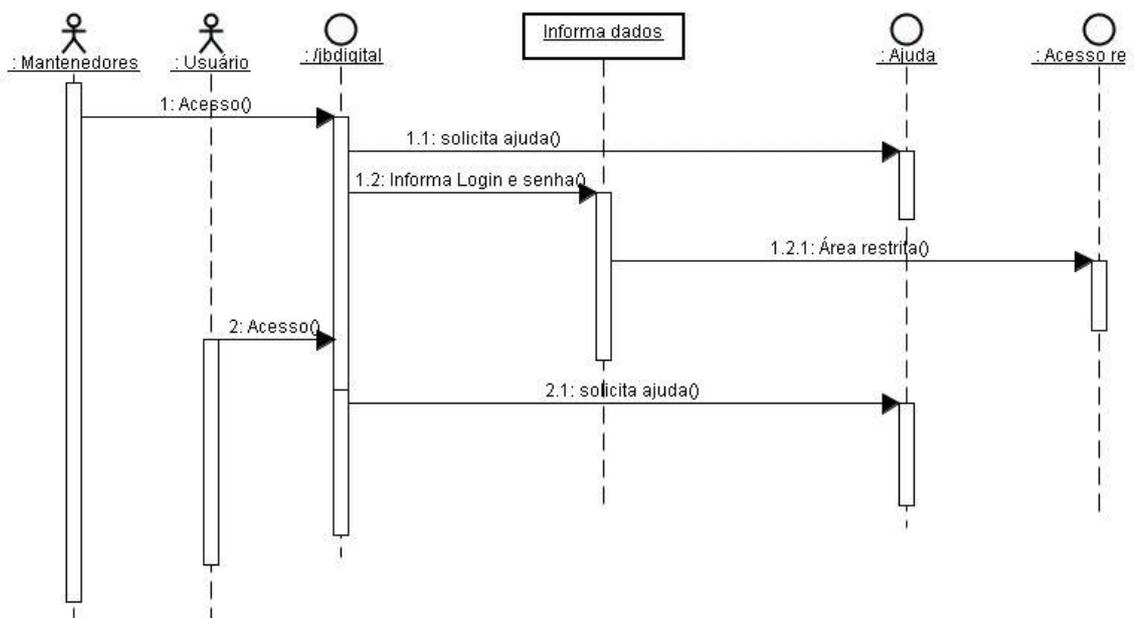


Figura 35 – Diagrama de seqüência da página principal.

O próxima figura apresenta o acesso à biblioteca digital de teses e dissertações, na figura 36.

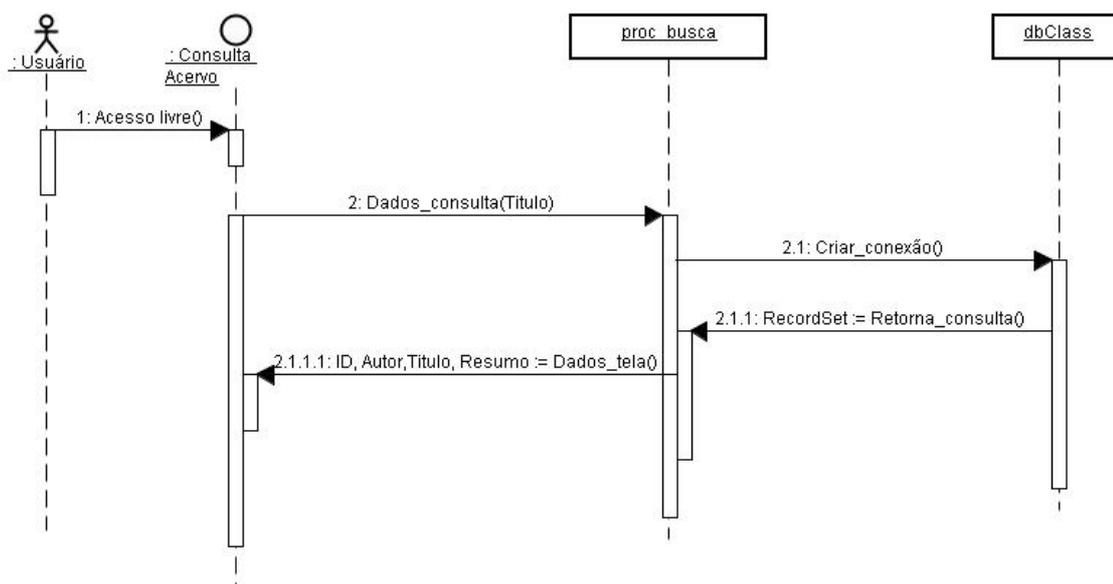


Figura 36 – Diagrama de seqüência da consulta de documentos.

O diagrama a seguir representa os usuários que irão ser cadastrados no banco de dados da biblioteca digital que terão a permissão de cadastrar novos documentos, na figura 37.

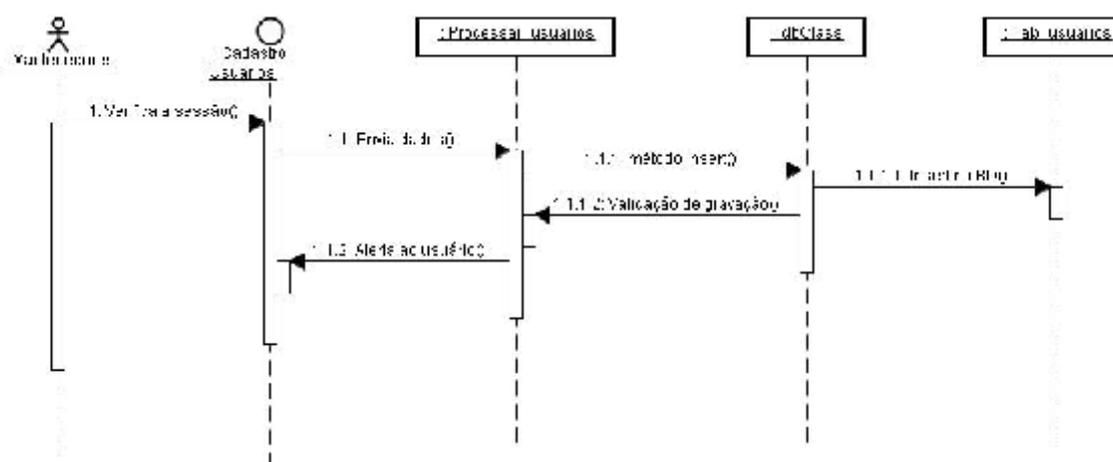


Figura 37 – Diagrama de seqüência de cadastramento de usuário.

O próximo diagrama apresenta a seqüência de cadastramento dos documentos na figura 38.

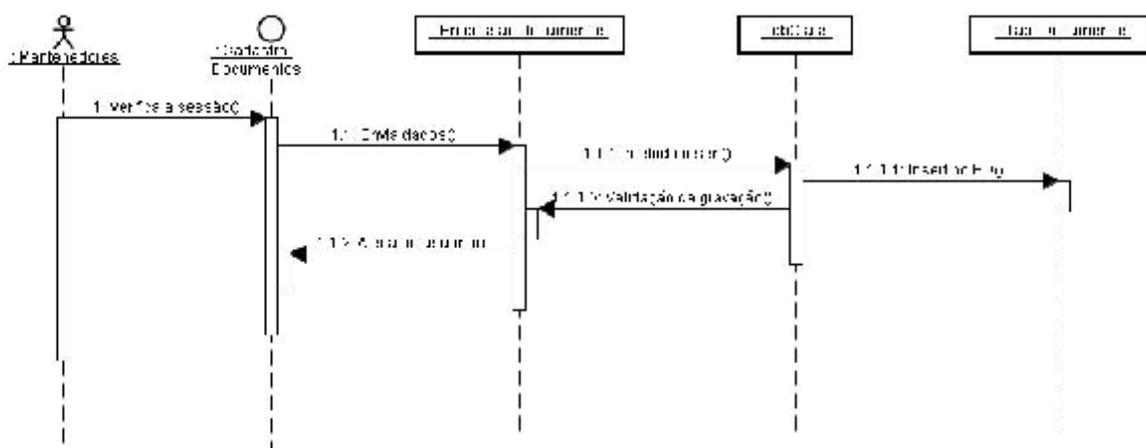


Figura 38 – Diagrama de seqüência de cadastro de documentos

A seguir é apresentado o diagrama que gera o arquivo XML diretamente na pasta do Web Service. Este arquivo é formatado por outro arquivo já existente no Web Service um JBDIGITAL.XSL (folha de estilo), que transforma qualquer arquivo XML lá cadastrado e apresentar o arquivo XML, com um layout mais bem formatado. A figura 39 representa este diagrama.

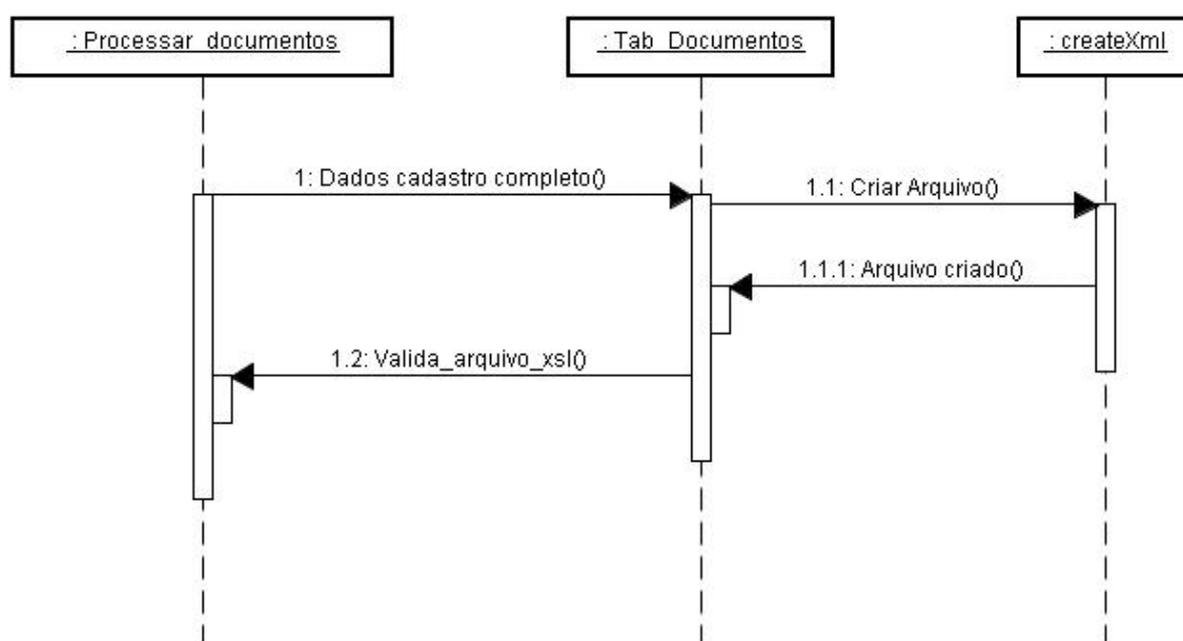


Figura 39 – Diagrama de seqüência para criar o XML

O diagrama de seqüência a seguir representa como é o envio do documento em formatado (PDF), que também é armazenado no Web Service, para que os usuários tenham acesso ao conteúdo total do documento digital, apresentado na figura 40.

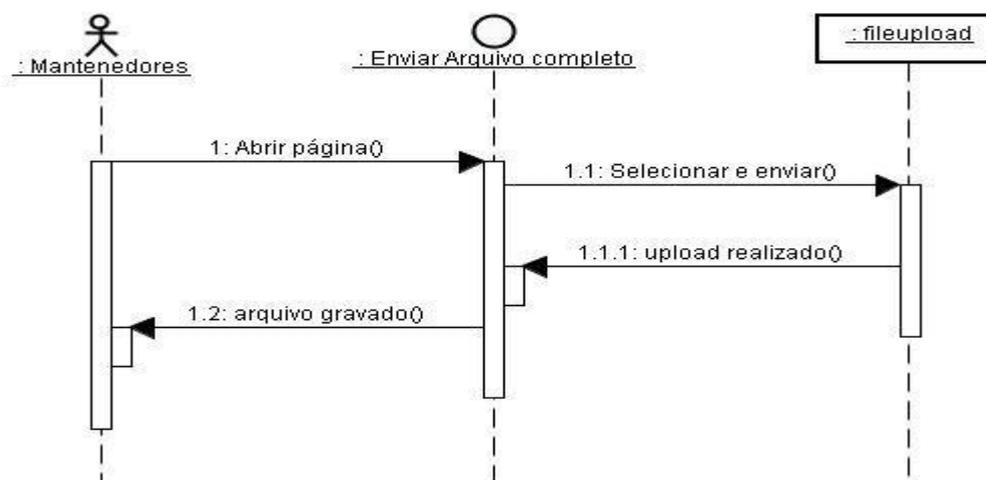


Figura 40 – Diagrama de seqüência de envio de arquivo completo

7. CONCLUSÃO

As tarefas básicas para um equipamento ligado na internet são: navegar, buscar, localizar processos e armazenar informações em algum domínio, entretanto, ainda tem-se a Internet sem domínios de conhecimento, estando estes distribuídos e sem nenhuma proposta de conhecimento que os relacionem, não se tratam apenas de links ou relações de dados, mas relações de metadados. Produções científicas são deixadas em sites de busca sem nenhum gerenciador de conhecimento que permita sua conexão com outros. Conforme Levy, sobre a cibercultura conceitua em seu livro, as associações e cooperações na realização de tarefas de busca de proliferação de informação na Internet irão quebrar o paradigma da individualidade virtual do conhecimento para fazê-lo junto.

A intenção de criar uma arquitetura para biblioteca digital de teses e dissertações para o armazenamento das pesquisas realizadas dentro de instituições, é promover acesso à produção científica concebida e publicada. Promover acesso a outras referências bibliográficas, proporcionando uma interoperabilidade de informações que poderão ser distribuídas pela Internet usando ontologia.

Elas têm a proposta de armazenar e dar acesso a volumes cada vez maiores de informação multimídia texto, imagem e som, em suportes digitais em diversos formatos a par com a existência de documentos. Apesar da distância física, que compromete muito qualquer tipo de interação, mesmo com avanços tecnológicos e ferramentas apresentadas nesta dissertação. Tende-se de alguma forma, diminuir as distâncias criando um elo entre as instituições e pesquisadores.

Elas estarão acessíveis aos seus potenciais utilizadores a qualquer hora e lugar. Contrariamente ao que até agora era normal, os utilizadores remotos da biblioteca pudessem obter não apenas informação secundária e de referência, mas também informação primária (conteúdo integral de documentos textuais, dados e informação, documentos multimídia, etc).

Mas até lá, as bibliotecas digitais permitirão de forma transparente para seus idealizadores, a pesquisa e o acesso às suas coleções locais ou qualquer outra fonte de informação existente nas redes de comunicação no qual estejam interligadas. A possibilidade de ligação virtual entre as bibliotecas transformará cada uma delas, num nó de uma biblioteca digital de escala planetária, por meio dos conceitos da Web Semântica, apresentados aqui.

Perante as revisões bibliográficas que foram realizadas, observou-se que alguns autores estão preocupados com metadados das informações armazenadas nas bibliotecas digitais. Não levando em consideração que uma ontologia cristaliza o conhecimento, permitindo melhores condições de busca e recuperação das informações já desenvolvido e conceituado para ontologias de informações digitais, por diversos autores, entre os quais neste trabalho destacam-se entre eles Nicolas Guarino e T.R. Gruber que mostraram a ontologia como necessidade para melhor exploração de conhecimento compartilhado, desde que a informação seja armazenada respeitando metadados estruturados com bases em ontologias.

É possível concluir que faltam ainda várias condições para que as bibliotecas digitais seja uma realidade utilizada em grande escala. O potencial tecnológico já existe (os problemas técnicos que ainda subsistem serão resolvidos a curto e médio prazo) e as mudanças sociais, organizações, culturais, necessárias são inevitáveis.

A inserção da biblioteca no mundo virtual exige, a despeito da sua complexidade, uma atitude distinta, bem como formas de pensar e agir diferenciadas. Esse novo cenário demanda aprendizagem contínua, ousadia e perspicácia dos profissionais da informação e da computação. Além dos desenvolvimentos tecnológicos novos e estudos interdisciplinares que contemplem a diversidade exigida nesse tipo de biblioteca.

Como uma contribuição a este esforço este trabalho implementou e analisou uma Biblioteca Digital contendo uma Ontologia DLibOnto, com as seguintes características. A construção dessa biblioteca digital de teses e dissertações DlibOnto, seguiu os padrões necessários para construção de um Web Services, ligado diretamente em um domínio de

Internet. Configurado adequadamente para atender as necessidades de interoperabilidade das informações com outros Web Services, implementado também com a plataforma distribuída Java J2EE para suporte computacional do Sistema DLibOnto.

O Sistema funciona acessando um banco de dados relacional Postgres com programação em JSP em três camadas (primeiro nível para página, segunda para processamento das classes internas e a terceira para acesso ao banco de dados). A ontologia DLibOnto foi definida com o suporte de um aplicativo chamado Protégé, que forneceu os eventos necessários para sua criação, relacionamentos e testes da sua validação. Esta ontologia DLibOnto está dentro do Web Services DLibOnto, assessorando o funcionamento do Sistema e oferecendo todo o conhecimento necessário das teses e dissertações ali depositadas. Elas serão recuperadas pelos usuários da própria DLibOnto, assim como outros usuários que desejam recuperar através de outros servidores Web.

REFERÊNCIAS BIBLIOGRÁFICAS

AHMED, K.; et al. Professional XML Meta Data, Wrox Press, 2001.

ALVARENGA, Lídia. **A teoria do conceito revistada em conexão com ontologias e metadados no contexto das bibliotecas tradicionais e digitais.** DataGramZero – Revista de Ciência da Informação, v.2, n.6, dez. 2001. Disponível em <http://www.dgz.org.br/dez01/Art_05.htm>. Acesso em Novembro 2003.

ALVARENGA, Lídia. **Representação do conhecimento na perspectiva da Ciência da informação em tempo e espaços digitais.** Enc. Bibli: R. Eletr. Bibliotecon. Ci. Inf., Florianópolis, n. 15, 2003. Disponível em <http://www.encontros-bibli.ufsc.br/Edicao_15/alvarenga_representacao.pdf>. Acesso em Dezembro. 2004.

ANDERSON, Dativ P.; KUBIATOWICZ, John. **The WorldWide Computer.** Disponível em: <<http://w3.org/DesignIssues/Semantic.html>>. Acesso em Novembro 2004.

ARIDOR, Y., FACTOR, M., TEPERMAN, A., **cJVM: a Single System Image of a JVM on a Cluster,** Proceedings, International Conference on Parallel Processing, 1999.

ARMS, William Y., **Key Concepts in the Architecture of the Digital Library.** D-Lib Magazine, July 1995. Disponível na URL: <<http://www.dlib.org/dlib/July95/07arms.html>>

ARMS, William Y.; **Digital Libraries.** The MIT Press, Cambridge, MA, 2000. ISBN 0-262-011880-8.

ARNOLD, A., GOSLING, J. **The Java Programming Language,** Book, Second Editon, Addison-Wesley Longman, Reading, MA, 1998.

BAEZA-YATES, Ricardo; RIBEIRO-NETO, Berthier. **Modern information retrieval,** New York, N.Y.: ACM Press, 1999.

BAPTISTA, C. **Steplib: A Digital Library for Spatio-Temporal and Multimedia Data.** Tese de doutorado, Universidade de Kent em Canterbury, 2000.

BECKS, Andreas; SKORZ, Stefan and TRESP, Christopher. **Semantic Structuring and Visual Querying of Document Abstracts in Digital Libraries**. In: Proceedings ECDL '98, Second european Conference, Heraklion, Crete, Greece, September 21-23, 1998.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. **The semantic web: a new form of web content that is meaningful to computers will unleash a revolution of new possibility**. s. Scientific American, New York, April 2002.

BERNERS-LEE, T. et al. **The semantic toolbox: building semantics on top of XML-RDF**. May 2003. Disponível em: <<http://www.w3.org/DesignIssues/Toolbox.html>>. Acesso em Fevereiro 2005.

BERNERS-LEE, Tim; HENDLER, James; LASSILA, Ora. **The semantic web**. Scientific American, New York, n. 5, May 2001. Disponível em: <<http://www.scian.com/2001/0501issue/0501berners-lee.html>>. Acesso em Fevereiro 2005.

BERNERS-LEE, Tim. **Semantic web road map**. Disponível em: <<http://w3.org/DesignIssues/Semantic.html>>. Acesso em Janeiro 2005.

BISHOP, P., Warren, N., **JavaSpaces In Practice**, Book, Addison-Wesley, 2002.

BRODGEN, Bill. **Java Developer's Guide to E-Commerce with XML and JSP**. São Paulo: Pearson Education do Brasil, 2002.

BORGHOFF, Uwe; PARESCHI, Remo. **Information technology for knowledge management**. Germany : Springer-Verlag, 1998.

BORDEN, J.;BRAY, T. **Resource Directory Language (RDDL)**. 2002 Disponível em: <<http://www.rddl.org>>. Acessado em Julho 2005.

BRAY, Tim. **What is RDF?** Disponível em: <<http://www.xml.com/pub/a/2001/01/24/rdf.html>>. Acesso em Agosto 2004.

CERAMI, Ethan. **Web Services Essentials: Distributed Applications with XML-RPC, SOAP, UDDI & WSDL**. USA: O'Reilly, 2002

CERT, **Cartilha de Segurança para Internet**, Versão 3.0, Disponível em: <<http://cartilha.cert.br>> . Acessado em Setembro de 2005.

COMPUTER CURRICULA – in: Computer Curricula, 2001, Disponível em <http://www.computer.org/portal/cms_docs_ieeecs/ieeecs/education/cc2001/cc2001.pdf>. Acessado em Março 2006.

CUNHA, Murilo Bastos da. **As tecnologias de informação e a integração das bibliotecas brasileiras**. Ciência da Informação, Brasília, DF, v. 23, n. 2, pp. 182-189, maio/ago. 1994. URL: <http://www.ibict.br/cionline>

CUNHA, Murilo Bastos da. **Biblioteca digital: Bibliografia internacional anotada. Ciência da Informação**, v. 26, n. 2, p. 195-213, 1997. URL: <http://www.ibict.br/cionline>

CUNHA, Murilo Bastos da. **Construindo o futuro: a biblioteca universitária brasileira em 2010**. Ciência da Informação, Brasília, DF, v. 29, n. 1, p. 71-89, 1999. URL: <http://www.ibict.br/cionline>

CUNHA, Murilo Bastos da. **Desafios na construção de uma biblioteca digital**. Ciência da Informação, Brasília, DF, v. 28, n. 3, p. 255-266, set./dez. 1999. URL: <http://www.ibict.br/cionline>

DATE, C. J. **Introdução a sistemas de bancos de dados**. Rio de Janeiro, RJ, p. 674, Campus, 1991.

DC. “**Dublin Core Metadata Initiative**”. Disponível em: <<http://dublincore.org/>>. Acessado em Dezembro 2004

DCMI - DUBLIN CORE METADATA INITIATIVE. **Dublin Core Qualifiers**. 2001. Disponível em: <<http://www.dublincore.org/documents/2000/07/11/dcmes-qualifiers>>. Acesso em: setembro de 2005.

DEITEL, Paul DEITEL Harvey. **XML How to Program**. USA: Prentice Hall, 2001.

DIAS, Eduardo José Wense. **Contexto digital e tratamento da informação**. Datagramazero: Ciência da Informação, v.2, n.5 out.2001. Disponível em: <http://www.dgz.org.br/out01/Art_01.htm>. Acesso em Novembro 2004.

FERREIRA, Aurélio Buarque de Holanda. **Novo dicionário de língua portuguesa**. 2ed. Rio de Janeiro : Nova Fronteira, 1986.

FOX, E. A.; MARCHIONINI, G. Digital Libraries. **Communications of the ACM**, v. 44, n. 1, p. 88-89, May 2001.

MULLENDER, S., **Distributed Systems**, Book, Second Edition, Addison-Wesley, ACM Press, 1993.

FOSTER, I., KESSELMAN, C., **The Grid: Blueprint for a New Computing Infrastructure**, Book, Morgan Kaufmann Publishers, 1999.

FARQUHAR, A. The Ontolingua Server: a Tool for Collaborative Construction, Computer Science Department, Stanford University. 1996.

FARQUHAR, A. 1997 Ontolingua tutorial. Disponível em <<http://www.ksl.stanford.edu/software/ontolingua/tutorial.pdf>>. Acesso em Março 2005.

FOSTER, I., KESSELMAN, C., **The Globus Project: A Status Report**", Proceedings IPPS/SPDP'98 Heterogeneous Computing Workshop, 1998, pp.4-18.

FREITAS, F.; BITTENCOURT, G. 2002. **Comunicação entre Agentes em Ambientes Distribuídos Abertos: o Modelo "peer to peer"**. Revista Eletrônica de Iniciação Científica (REIC). Ano II No. II Vol. II. Sociedade Brasileira de Computação (SBC). Brasil.

FREITAS, F.; BITTENCOURT, G. 2003. **An Ontology-based Architecture for Cooperative Information Agents**. IJCAI'2003. Acapulco. México.

GNUTECA - Disponível em: <<http://gnuteca.codigolivre.org.br>>. Acesso em Novembro 2004.

GUARINO, N. Formal Ontology, **Conceptual Analysis and Knowledge Representation**. International Journal of Human and Computer Studies, 1995

GUARINO, N. **Understanding, Building, and Using Ontologies: A commentary to “Using Explicit Ontologies in KBS Development”**, by van Heijst, Schreiber, and Wielinga. *International Journal of Human and Computer Studies*, 1997

GUARINO, N. **Formal Ontology and Information Systems**. *Proceedings of FOIS'98, Trento, Italy*, 1998

GUARINO, N., MASOLO, C., VETERE, G., **Ontoseek: Content-based access on the Web**. IEEE Intelligent System, 1999

GRUBER, T. R. *What is an ontology?* Disponível em < <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>>. Acessado em setembro de 2005.

GRUBER, T. R. **Ontolingua: a mechanism to suport portable ontologies**. Stanford: Knowledge Systems Laboratory Stanford University, 1992.

GRUBER, T. R. **A Translation Approach to portable ontology specifications**. Stanford: Knowledge Systems Laboratory Stanford University, 1993.

GRUBER, T. R. **Toward Principles for the Design of Ontologies Used for Knowledge Sharing**. Stanford: Knowledge Systems Laboratory Stanford University, 1994.

IBICT. **Editorial**. *Ciência da Informação*, Brasília, DF, v. 30, n. 3, p. 5-6, 2001. URL: <http://www.ibict.br/cionline>

IKEMATU, R. S. **Gestão de metadados: sua evolução na tecnologia da informação**. *DataGramaZero – Revista de Ciência da informação*, v.2, n.6, dez.2001. Disponível em <http://www.dgzero.org/dez01/Art_02.htm>. Acesso em: 11 junho 2004.

KOIVUNEN, M., MILLER, E. 2001. *W3C Semantic Web Activity*. Disponível em <http://www.w3.org/2001/12/semweb-fin/w3csw> >. Acesso em: 12 Janeiro 2006

KUMARAN, I., “**Jini Technology: An Overview**”, Book, Prentice Hall, 2001.

LANCASTER, F.W. **Ameaça ou oportunidade? O futuro dos serviços de biblioteca à luz das inovações tecnológicas**. Revista da Escola de Biblioteconomia da UFMG, Belo Horizonte, v. 23, n. 1, p. 7-27, 1994

LÉVY, Pierre. **Cibercultura**; Tradução Carlos Irineu da Costa. São Paulo: 2ª ed. Editora 34 Ltda, 1999

MAGAZINE, JAVA. **Proteja seus dados: Começando com a criptografia**. Neofício Editora, 2005.

MINSKY, M. **A Framework for Representation Knowledge**. In *The Psychology of Computer Vision*, p.211-281, McGraw-Hill, New York. USA, 1975.

MASIERO, Paulo César *et al.* **A biblioteca digital de teses e dissertação da Universidade de São Paulo**. *Ciência da Informação*. Brasília, v.30,n.3,p.34-41. set./dez. 2001

MUCHERONI, Marcos L., TAMAE, Rodrigo Y., BRACCIALLI, Tiago. **Auto-organizing Agents in Ontologies with Intentions**. UNIVEM, Marília, São Paulo, Brasil, 2003.

NEWCOMER, Eric. **Understanding Web Services, XML, WSDL, SOAP and UDDI**. USA: Independent Technology Guides / David Chappel – Series Editor, 2001.

NONAKA, I.; TAKEUCHI, H. **Criação de conhecimento na empresa**. Rio de Janeiro: Campus, 1998.

NSFDLI (National Science Foundation's Digital Library Initiative - <http://www.dli2.nsf.org>).

OLIVEIRA, D. H. de. **Introdução ao XML e suas aplicações**. Disponível em: http://www.xml.com.br/docs/intro_xml_apli.pdf>. Acesso em Janeiro 2002.

OPEN ARCHIVE INITIATIVE PROTOCOL FOR METADATA HARVESTING. Version 2. Disponível em: <http://www.openarchives.org/OAI/openarchivesprotocol.html>. Acesso em: 17 de julho de 2005.

PACKER, Abel *et al.* SciELO: **Uma metodologia para publicação eletrônica**. *Ciência da informação*. Brasília, v.27, n. 2, 1998. Disponível em <<http://www.scielo.br/pdf/ci/v27n2/scielo.pdf>>. Acesso em Setembro de 2005.

PROTÉGÉ, Natalya Fridman Noy, Ray W. Ferguson, Mark A. Musen. **The knowledge model of Protégé-2000: combining interoperability and flexibility**. Disponível em: <http://smi-web.stanford.edu/pubs/SMI_Abstracts/SMI-2000-0830.pdf>. Acesso em Janeiro 2005.

PROTÉGÉ, Natalya F. Noy, Michael Sintek, Stefan Decker, Monica Crubézy, Ray W. Ferguson, and Mark A. Musen, *Stanford University*. **Creating Semantic Web Contents with protege-2000**. Disponível em: <http://smi-web.stanford.edu/pubs/SMI_Abstracts/SMI-2001-0872.pdf>. Acesso em Janeiro 2005.

RDF: Resource Description Framework. **Model and Syntax Specification**. W3C Recommendation 22 february 1999. Disponível em: <<http://www.w3.org/TR/1999/REC-rdfsyntax-19990222>> Acesso em: junho 2005.

RDF: Resource Description Framework. **Schema Specification**. 1998. Disponível em: <<http://www.w3.org/TR/1998/WD-rdf-schema/>> Acesso em: julho 2005.

ROSETTO, Márcia. Metadados e formatos de metadados em sistemas de informação: caracterização e definição. São Paulo, 2003. 112 p. (Dissertação de mestrado apresentada ao Curso de Pós-Graduação da Escola e Comunicações e Artes da Universidade de São Paulo).

RUGGLES, R. **Knowledge Management Tools**. Boston: Batewurrth-Heinemann, 1996.

SABER-USP digital de teses e dissertações da USP. Disponível em: <<http://www.teses.usp.br>>, Acessado em Junho 2005.

SANCHEZ DIAZ, Marlery y VEGA VALDES, Juan Carlos. **Bibliotecas electrónicas, digitales y virtuales: tres entidades por definir**. *ACIMED*. (online). nov.-dic. 2002, vol.10, no.6 p.9-10. ISSN 1024-9435. Disponível em http://scieloprueba.sld.cu/scielo.php?script=sci_arttext&pid=S102494352002000600005&lng=es&nrm=iso>. Acesso em Novembro 2004.

SEMANTIC, Web. Disponível em <www.semanticweb.org>. Acesso em Dezembro 2004.

SENA, Natália Kneipp. **Open Archives: caminho alternativo para a comunicação científica**. *Ciência da Informação*. Brasília, DF, v. 29, n. 3, p. 71-78 set./dez. 2000.

SUN Microsystems, “**Computing at the Edge**”, White Paper, 2003. Disponível em <www.sun.com/servers/entry/lx50/pdfs/whitepapers/whitepapers.Edg e.pdf>

TYLER, David; CHAPPELL J. **Java Web Services**. USA: O’Reilly, 2002.

UDDI. **UDDI Technical White Paper**. 2000. Disponível em: <http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf>. Acessado em Novembro 2005.

W3C. Disponíveis em <www.w3.org>;<www.w3.org/People>. Acessados em Novembro de 2004.

WORLD WIDE WEB CONSORTIUM (W3C). **Resource Description Framework (RDF) Model and syntax specification**, 1999. Disponível em: <<http://www.w3c.org/TR/1999/RECrdf-syntax/>> Acesso em: agosto de 2005.

WITTEN, Ian; BAINBRIDGE, David. *How to Build a Digital Libray*. San Francisco, CA, 2003

WATERS, Donald J., **What Are Digital Libraries?** Digital Library Information Resources in Berkeley Digital Library SunSite. CLIR Issues, Number 4, July/August 1998. Disponível na URL: <<http://www.clir.org/pubs/issues/issues04.html>>

XML. Disponível em: <www.xml.com/pub/r/1246> ; <www.w3.org>. Acesso em Janeiro 2003.

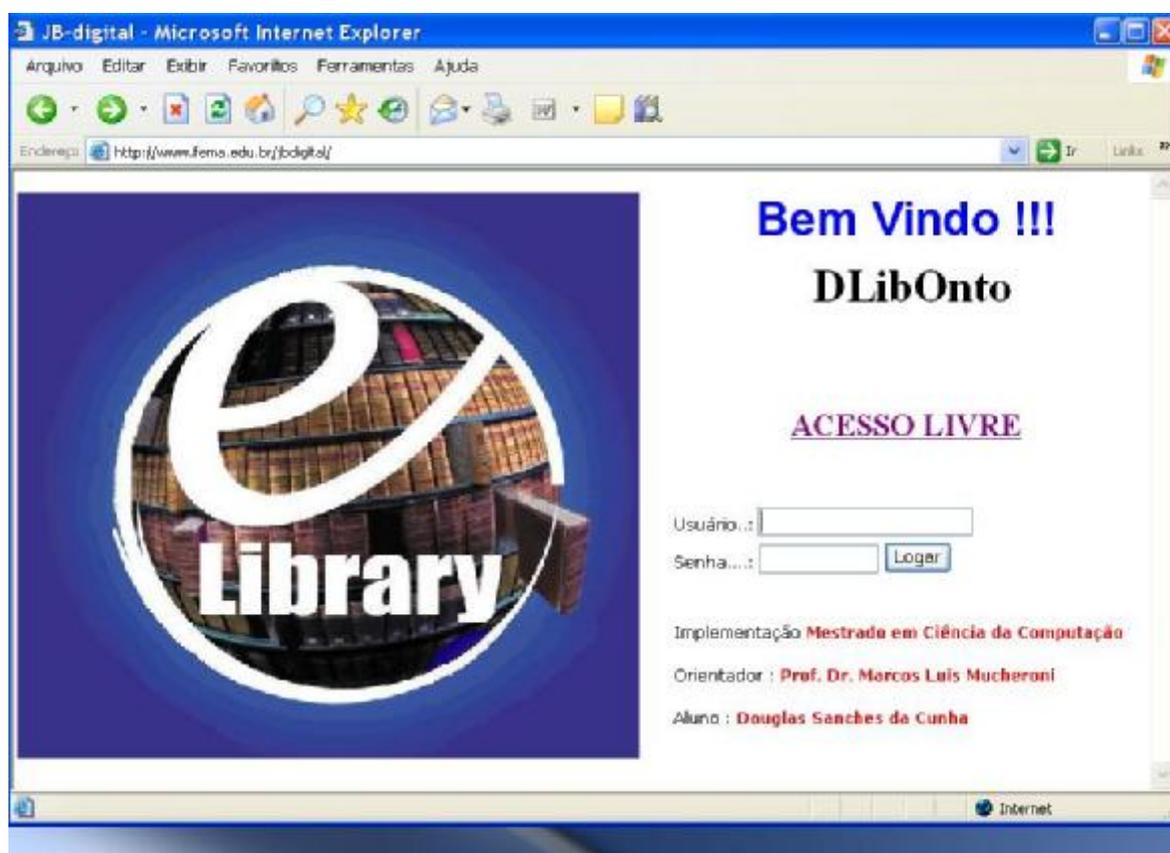
APÊNDICES

APÊNDICE A

O Sistema da DLibOnto está implementado no provedor FEMAnet no link:
www.fema.edu.br/jbdigital/.

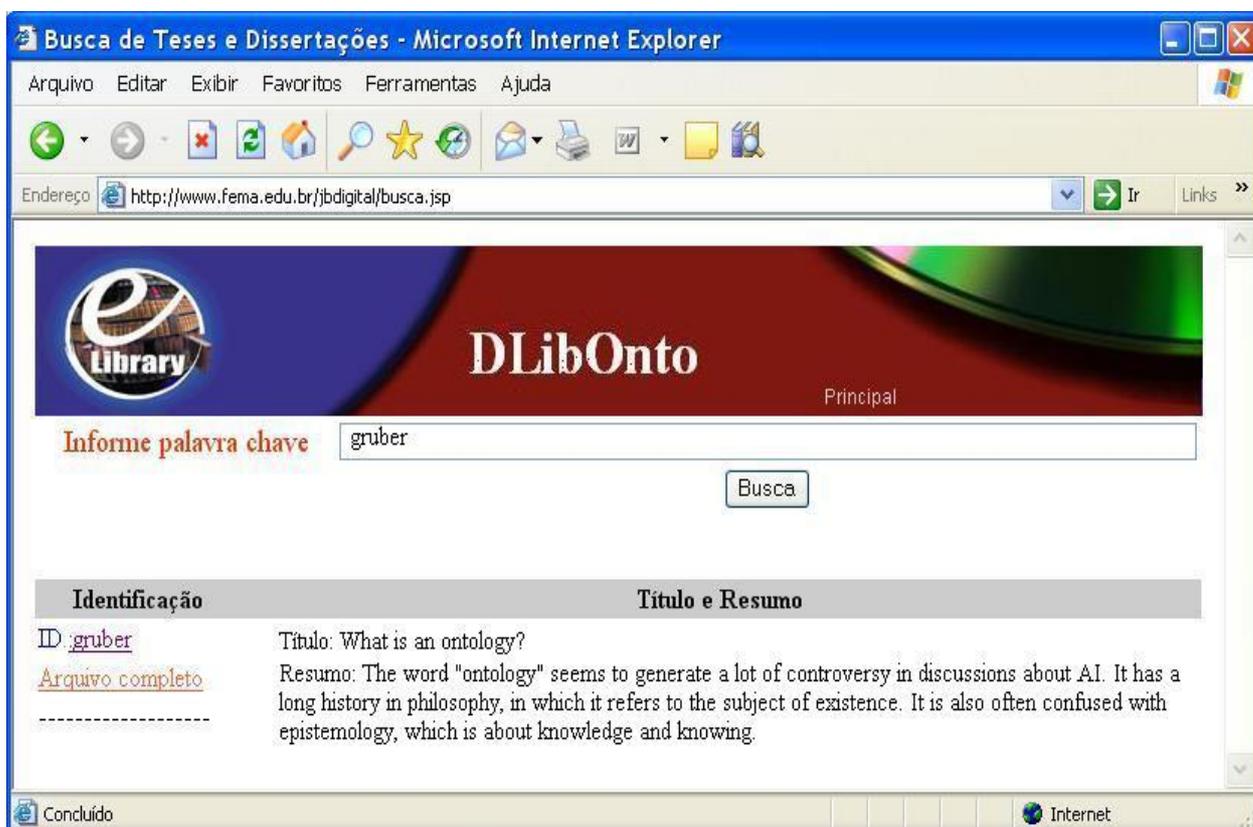
APÊNDICE B – Tela principal

Qualquer usuário que acessar o Acesso Livre, tem acesso a todo o conteúdo da DLibOnto, devendo preencher uma palavra para que sejam listados os conteúdos.



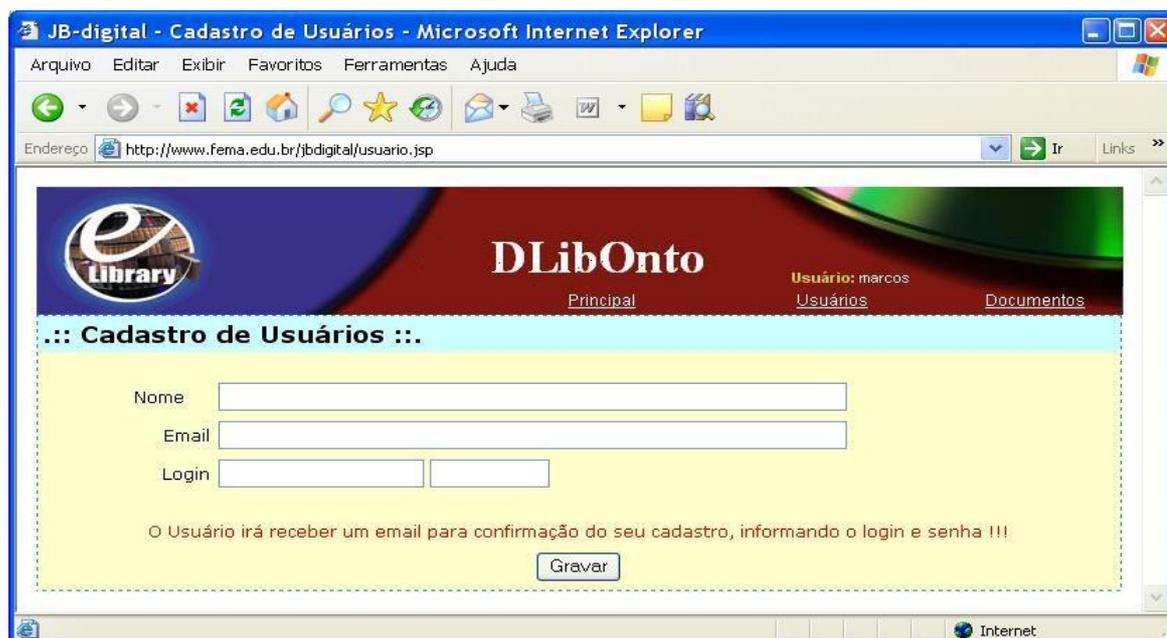
APÊNDICE C – Formatos da dissertação ou artigo cadastrado

A tela principal do sistema está representada abaixo, onde o usuário tem acesso a uma ajuda, acesso livre e login com senha para o acesso aos administradores da DLibOnto. Para o usuário que deseja acessar o acervo digital já cadastrado, basta apenas clicar no **ACESSO LIVRE**.



APÊNDICE D – Acesso aos autores e ao administrador

O administrador tem um o acesso restrito, se identificando com um login e senha, o administrador faz acesso às rotinas de cadastramento de usuários ou documentos, os autores cadastrados de usuários também tem este acesso.



JB-digital - Cadastro de Usuários - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço <http://www.fema.edu.br/jbdigital/usuario.jsp> Ir Links

DLibOnto
Principal Usuários Documentos

Usuário: marcos

:: Cadastro de Usuários ::

Nome

Email

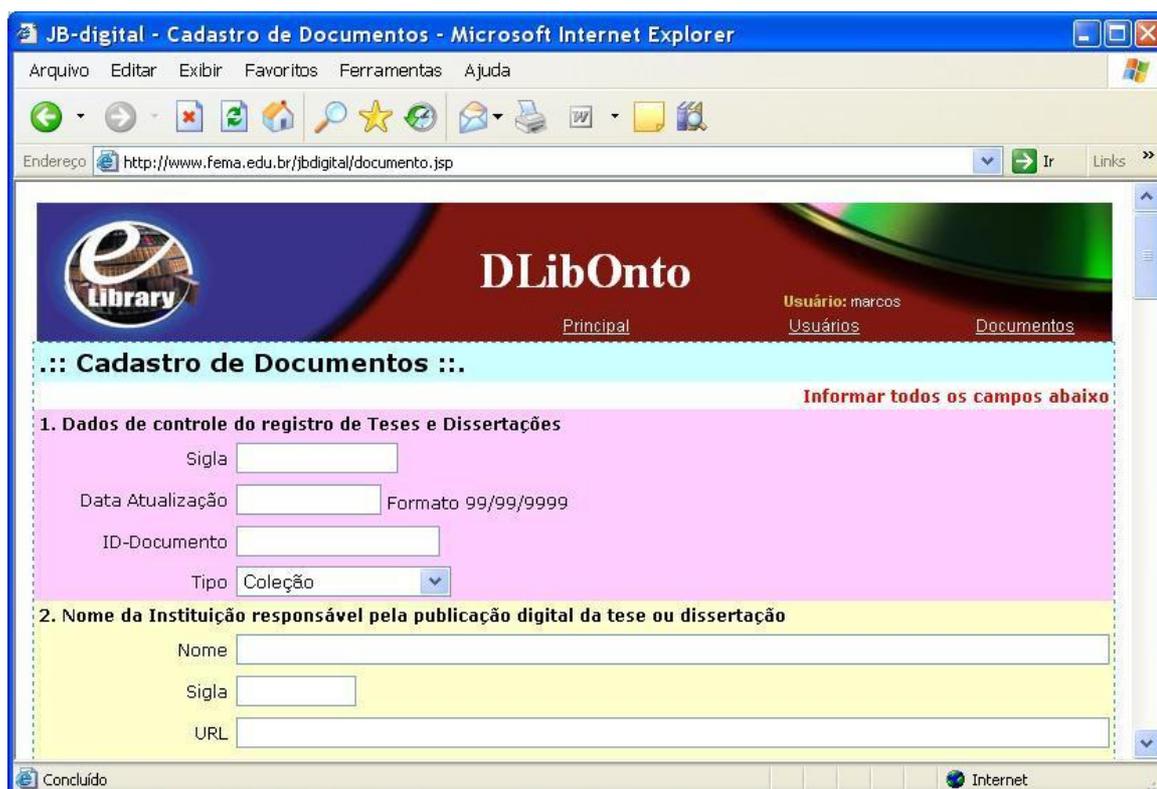
Login

O Usuário irá receber um email para confirmação do seu cadastro, informando o login e senha !!!

Gravar

Internet

Estes usuários terão acesso ao Cadastro de documentos.



JB-digital - Cadastro de Documentos - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço <http://www.fema.edu.br/jbdigital/documento.jsp> Ir Links

DLibOnto
Principal Usuários Documentos

Usuário: marcos

:: Cadastro de Documentos ::

Informar todos os campos abaixo

1. Dados de controle do registro de Teses e Dissertações

Sigla

Data Atualização Formato 99/99/9999

ID-Documento

Tipo Coleção

2. Nome da Instituição responsável pela publicação digital da tese ou dissertação

Nome

Sigla

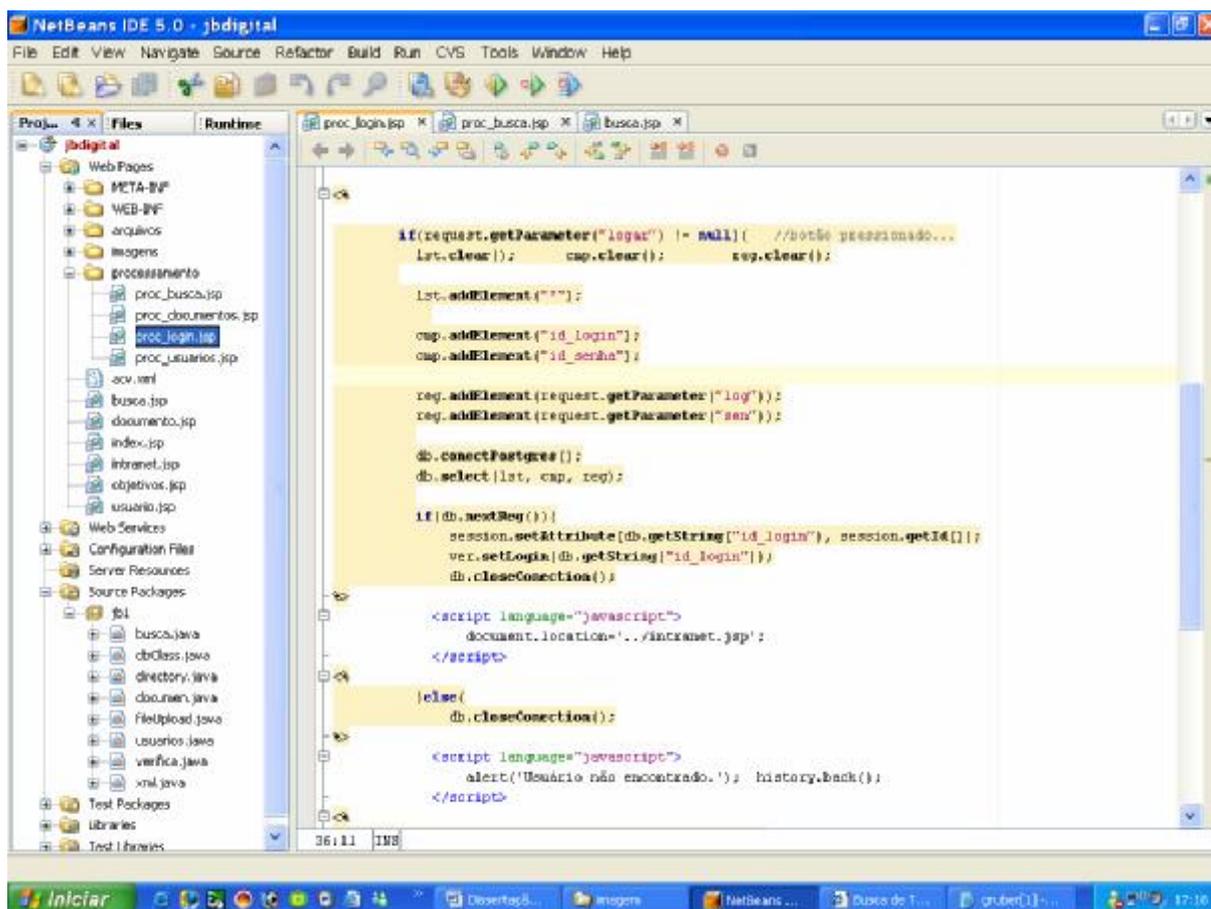
URL

Concluído

Internet

APÊNDICE E – Construções dos serviços usando NetBeans

A ferramenta utilizada para fazer a construção do site da DLibOnto, foi NetBeans 5.0, para a composição dos códigos em java, jsp e está disposta em três camadas exemplificada pelas telas a seguir:



APÊNDICE E1 – JAVA e JSP

```

<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<!--
The taglib directive below imports the JSTL library. If you uncomment it,
you must also add the JSTL library to the project. The Add Library...
action
on Libraries node in Projects view can be used to add the JSTL 1.1 library.
--%>
<!--
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
--%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <jsp:useBean id="cmp" scope="page" class="java.util.Vector" />
    <jsp:useBean id="reg" scope="page" class="java.util.Vector" />
    <jsp:useBean id="lst" scope="page" class="java.util.Vector" />
    <jsp:useBean id="db" scope="page" class="jbl.usuarios" />
    <jsp:useBean id="ver" scope="session" class="jbl.verifica" />

    <%
      if(request.getParameter("logar") != null){ //botão pressionado...
        lst.clear();      cmp.clear();      reg.clear();

        lst.addElement("");
        cmp.addElement("id_login");
        cmp.addElement("id_senha");
        reg.addElement(request.getParameter("log"));
        reg.addElement(request.getParameter("sen"));
        db.conectPostgres();
        db.select(lst, cmp, reg);
        if(db.nextReg()){
          session.setAttribute(db.getString("id_login"),
session.getId());
          ver.setLogin(db.getString("id_login"));
          db.closeConection();
%>
          <script language="javascript">
            document.location='../intranet.jsp';
          </script>
<%
        }else{
          db.closeConection();
%>
          <script language="javascript">
            alert('Usuário não encontrado. '); history.back();
          </script>
<%
          //response.sendRedirect("../index.jsp");
        }
      }
    %>
  </body>
</html>

```

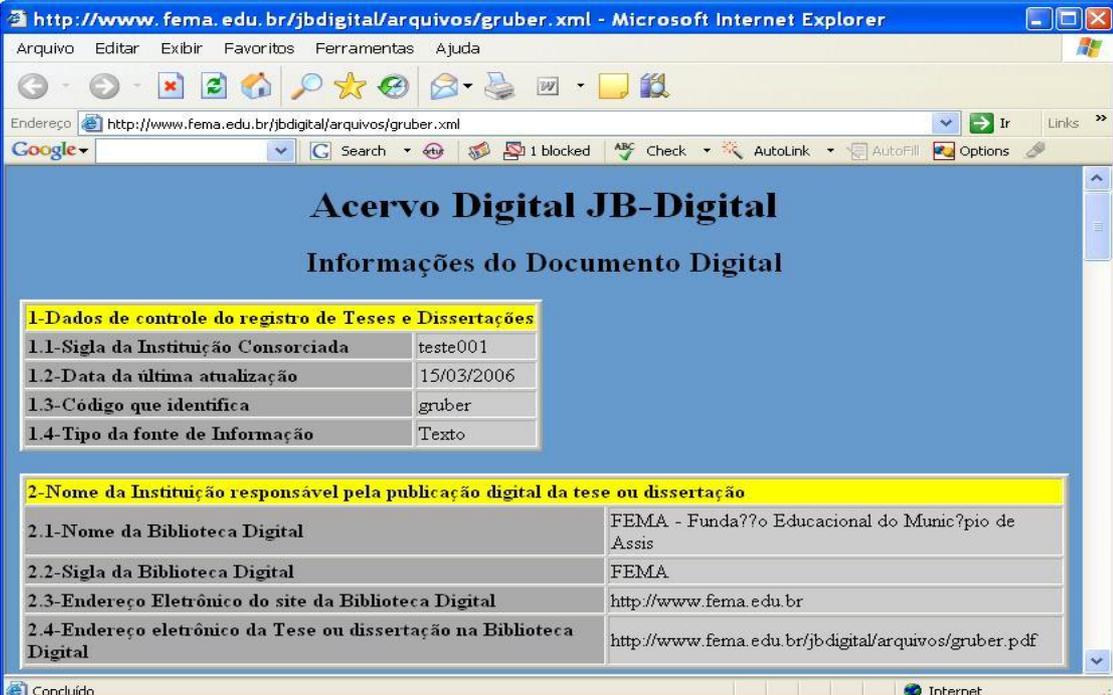
APÊNDICE E2 – XML e XSL

```

<?xml version='1.0' encoding='ISO8859-1'?>
<?xml-stylesheet href='jbdigital.xsl' type='text/xsl'?>
<JBDIGITAL>
  <Controle>
    <Sigla>teste001</Sigla>
    <DataAtualizacao>15/03/2006</DataAtualizacao>
    <IdentificacaoDocumento>gruber</IdentificacaoDocumento>
    <Tipo>Texto</Tipo>
  </Controle>
  <BibliotecaDigital>
    <Nome>FEMA - Fundação Educacional do Município de Assis</Nome>
    <Sigla>FEMA</Sigla>
    <URL>http://www.fema.edu.br</URL>
  <URLDocumento>http://www.fema.edu.br/jbdigital/arquivos/gruber.pdf</URLDocu
  mento>
  <ProvedorServico>
    <Nome>FEMA - Fundação Educacional do Município de Assis</Nome>
    <Sigla>FEMA</Sigla>
    <Pais>Brasil</Pais>
    <UF>SP</UF>
    <CNPJ>51.501.559/0001-36</CNPJ>
    <URL>http://www.fema.edu.br</URL>
  </ProvedorServico>
</BibliotecaDigital>
<BibliotecaDepositaria>
  <Nome>FEMA - Fundação Educacional do Município de Assis</Nome>
  <Sigla>FEMA</Sigla>
  <URL>http://www.fema.edu.br/jbdigital/arquivos/</URL>
<NumeroChamada>http://www.fema.edu.br/jbdigital/arquivos/</NumeroChamada>
</BibliotecaDepositaria>
  <Titulo>Texto do Gruber</Titulo>
</JBDIGITAL>

```

Apenas uma parte.



http://www.fema.edu.br/jbdigital/arquivos/gruber.xml - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço http://www.fema.edu.br/jbdigital/arquivos/gruber.xml

Acervo Digital JB-Digital

Informações do Documento Digital

1-Dados de controle do registro de Teses e Dissertações	
1.1-Sigla da Instituição Consorciada	teste001
1.2-Data da última atualização	15/03/2006
1.3-Código que identifica	gruber
1.4-Tipo da fonte de Informação	Texto

2-Nome da Instituição responsável pela publicação digital da tese ou dissertação	
2.1-Nome da Biblioteca Digital	FEMA - Funda??o Educacional do Munic?pio de Assis
2.2-Sigla da Biblioteca Digital	FEMA
2.3-Endereço Eletrônico do site da Biblioteca Digital	http://www.fema.edu.br
2.4-Endereço eletrônico da Tese ou dissertação na Biblioteca Digital	http://www.fema.edu.br/jbdigital/arquivos/gruber.pdf

Concluído Internet

APÊNDICE E3 – RDF e OWL

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.owl-ontologies.com/unnamed.owl#"
  xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Class rdf:ID="Realidade_Virtual">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Computação_Visual_Gráficos"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="BibliotecaDepositaria">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Biblioteca onde o exemplar físico da tese ou dissertação está
arquivado</rdfs:comment>
  <owl:Class rdf:about="#DLibOnto">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Lista de Elementos do Schema (MTD-BR) Padrão Brasileiro de Metadados
de Teses e Dissertações</rdfs:comment>
  </owl:Class>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Area">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Programa"/>
        <owl:Class rdf:about="#Ciência_Computação"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
</rdf:RDF>

```

Código com muitas páginas, aqui apenas uma parte.