

FACULDADES INTEGRADAS DE CARATINGA

FACULDADE DE CIÊNCIA DA COMPUTAÇÃO

PADRÕES DA WEB SEMÂNTICA E ONTOLOGIA COM USO DO PROTÉGÉ PARA RECUPERAÇÃO EFICIENTE DE INFORMAÇÕES DO SISTEMA ACADÊMICO ADX.

LEONARDO ALVES COTTA DE SOUZA

Caratinga

211

Leonardo Alves Cotta de Souza

**PADRÕES DA WEB SEMANTICA E ONTOLOGIA COM USO DO PROTÉGÉ PARA
RECUPERAÇÃO EFICIENTE DE INFORMAÇÕES DO SISTEMA ACADÊMICO
ADX.**

Monografia submetida à Comissão examinadora designada pelo Curso de Graduação em Ciência da Computação como requisito para obtenção do grau de Bacharel orientada pelo Prof. Glauber Luis da Silva Costa.

Caratinga

Leonardo Alves Cotta de Souza

**PADRÕES DA WEB SEMANTICA E ONTOLOGIA COM USO DO PROTÉGÉ
PARA RECUPERAÇÃO EFICIENTE DE INFORMAÇÕES DO SISTEMA
ACADÊMICO ADX.**

Monografia submetida à Comissão examinadora designada pelo Curso de Graduação em Ciência da Computação como requisito para obtenção do grau de Bacharel orientada pelo Prof. Glauber Luis da Silva Costa.

Prof. Glauber Luis da Silva Costa (Orientador)

Faculdades Integradas de Caratinga

Prof. Msc. Fabrícia Pires Souza Tiola

Faculdades Integradas de Caratinga

Prof. Msc. Jacson Rodrigues Correia da Silva

Faculdades Integradas de Caratinga

Caratinga, 7/12/211

DEDICATÓRIA

Dedico meu trabalho primeiramente a Deus, ao meu pai, Edeilson Cotta de Souza, à minha mãe (em memória), Regina Alves Cotta de Souza e, principalmente, à minha avó, Maria da Penha Cotta de Souza. Todos sempre estiveram ao meu lado e se propuseram a me ajudar para chegar onde cheguei. Dedico a eles minha vitória e agradeço por fazerem parte dela.

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter chegado até aqui, à minha família, aos meus amigos e colegas por me apoiarem e incentivarem para que nunca desistisse do meu objetivo, ao meu orientador Glauber que se propôs, com dedicação, me ajudar e mostrar o caminho através de seus conhecimentos, para que o trabalho fosse realizado.

Obrigado a todos.

RESUMO

Web Semântica é uma evolução da Web atual, com ela, agentes de softwares na Web podem localizar informações obedecendo a semântica imposta na busca, graças aos padrões de desenvolvimento da mesma.

Foi realizado neste trabalho o uso padrões já definidos para a Web Semântica apresentando, através de um sistema simples de busca, a viabilidade destes conceitos em um sistema de gerenciamento acadêmico.

A metodologia empregada baseia-se em um referencial teórico para empregar todo o conceito nele descrito, aplicando-o em um sistema simples de busca, a fim de obter resultados viáveis e objetivos.

É apresentada a criação de ontologia usando ferramenta Prótegé(Ferramenta para criar ontologias), que por sua vez, define a URI (Identificador único de recursos) do projeto, bem como importação de vocabulário, como o Dublin Core (padrão de metadados) e criação de classes. Com base de dados Mysql (Sistema gerenciador de banco de dados) foi realizado a inserção de dados através da API RAP (API PHP para trabalhar com padrões da Web Semântica) na linguagem PHP (Processador de Hipertexto), que possibilitou as consultas no banco de dados usando a linguagem de consulta à RDF (Descritor de recursos) chamada SPARQL (Protocolo de consulta e descritores de recursos). Foi necessário o uso de lógica, para reconhecimento da linguagem natural, visto que não feito o uso de reconhecimento da linguagem natural.

Com os resultados obtidos, usando padrões da Web Semântica para recuperar informações acadêmicas de alunos, pode-se notar que estes padrões quando obedecidos corretamente podem apresentar resultados positivos, retornando as informações de forma correta e objetiva. O uso destes padrões requer mudanças consideráveis na forma como são projetados e desenvolvidos os sistemas atualmente.

Palavra-chave: Web Semântica, Ontologia, Metadados, Sistema Acadêmico, Informações Semânticas.

ABSTRACT

Semantic Web is an evolution of the Web today, with it, software agents can find information on the Web obeying the semantics imposed in the search, thanks to the development patterns of the same.

This work was carried out using standards already set for presenting the Semantic Web through a simple search, the viability of these concepts in an academic management system.

The methodology is based on a theoretical framework to employ the concept described in it, applying it in a simple search in order to obtain viable results and objectives.

You see the creation of ontology using Protégé tool (Tool to create ontologies), which in turn sets the URI (unique resource identifier) of the project as well as import of vocabulary, such as Dublin Core (metadata standards) and creation of classes. With MySQL database system (database manager) was performed to enter data via the API RAP (PHP API for working with Semantic Web standards) on PHP (Hypertext Processor), which enabled the queries in the database data using the query language for RDF (resource descriptor) called SPARQL Protocol (consultation and resource descriptors). It was necessary to use logic to natural language recognition, since not made use of natural language recognition.

With the results obtained using Semantic Web standards for retrieving academic information of students, may be noted that these standards when properly obeyed may have positive, returning the information accurately and objectively. The use of these standards requires considerable changes in how they are designed and engineered systems today.

Keyword: Semantic Web, Ontology, Metadata, System Academic, semantic information.

LISTA DE SIGLAS

ADODB	- Ativo Dado Objeto Banco de dados
API	- Interface de aplicação programação
DTD	- Definição Tipo Documento
HTML	- Linguagem de marcação de hipertexto
IA	- Inteligência Artificial
IRI	- Identificadores de Recursos internacionalizados
OWL	- Linguagem Ontológica Web
PHP	- PHP Processador de Hipertexto
RDF	- Framework Descritor de Recursos
SGML	- Linguagem padrão generalizada de Marcador
SPARQL	- Linguagem de consulta para descritor de recursos
SQL	- Linguagem estruturada de consultas
TAG	- Etiqueta ou palavras chave
URI	- Identificador de recursos Uniforme
W3C	- World Wide Web Consortium
WWW	- World Wide Web
XHTML	- Linguagem extensível Marcador de Hipertexto
XML	- Linguagem extensível Marcador
XSL	- Linguagem extensível folha de estilo

LISTA DE TABELAS

Tabela 1: Definição padrão de Namespaces no SPARQL.....	40
---	----

LISTA DE ILUSTRAÇÕES

Figura 1: Camadas da Web Semântica.....	21
Figura 2: Catálogo de produtos em XML.....	29
Figura 3: Conflitos com Namespaces do XML.....	30
Figura 4: Catálogo de produtos usando grafo RDF.....	32
Figura 5: Tripla RDF.....	33
Figura 6: Catálogo de produtos usando RDF/XML.....	36
Figura 7: Exemplo de RDF e URIs.....	38
Figura 8: Busca Nome e e-mail utilizando RDQL.....	41
Figura 9: Escrever uma Consulta Simples RDQL.....	42
Figura 10: Nível de expressividade OWL.....	46
Figura 11: Novo projeto no Protégé.....	51
Figura 12: Escolhendo Modelo Protégé.....	51
Figura 13: Criando URI no Protégé.....	52
Figura 14: Sub-Linguagem OWL.....	53
Figura 15: Expressividade Protégé.....	53
Figura 16: Ambiente Protégé Classes.....	54
Figura 17: Definindo URI no Protégé.....	62
Figura 18: Definição da Sublinguagem do OWL.....	63
Figura 19: Definindo Nível de expressividade utilizando modelo lógico no Protégé.....	63
Figura 20: Importação do Padrão(Namespaces) Dublin Core.....	64
Figura 21: Definindo propriedades da classe Aluno.....	65
Figura 22: Definindo propriedades da classe Curso.....	66
Figura 23: Inserindo Curso no Protégé.....	66
Figura 24: Inserindo Aluno no Protégé.....	67
Figura 25: Busca Interna na Ontologia Protégé.....	67

Figura 26: Criando tabelas no banco de dados usando API RAP, na aplicação.....	70
Figura 27: Consulta RDQL criada na aplicação.....	72
Figura 28: Tela da Aplicação desenvolvida para demonstração.....	75
Figura 29: Apenas resultados de alunos com idade maior que 20 anos.....	75
Figura 30: Apenas alunos do curso de psicologia.....	76

SUMÁRIO

1	INTRODUÇÃO.....	14
2	REFERENCIAL TEÓRICO.....	17
2.1	Semântica	17
2.2	Web Semântica.....	18
2.2.1	Estrutura atual da Web Semântica	21
2.3	METADADOS.....	23
2.3.1	Padrões de Metadados.....	24
2.3.2	Dublin Core – DC.....	24
2.4	LINGUAGENS.....	26
2.4.1	XML.....	27
2.4.2	RDF.....	31
2.4.3	SPARQL – LINGUAGEM DE CONSULTA RDF.....	40
2.5	ONTOLOGIAS E OWL.....	43
2.5.1	OWL.....	45
2.6	SISTEMA ACADÊMICO ADX.....	48
3	FERRAMENTAS.....	49
3.1	Protégé.....	50
3.2	MYSQL.....	55
3.3	API RAP (RDF API).....	56
3.3.1	API RAP COM RDF E RDFS.....	57
3.3.2	API RAP E ARMAZENAMENTO PERSISTENTE.....	58
4	METODOLOGIA.....	60
4.1	CRIANDO UMA ONTOLOGIA OWL USANDO Protégé	61

4.1.1 Definindo URI da Ontologia.....	62
4.1.2 Importando vocabulário Dublin Core em Ontologia	64
4.1.3 Criando Classes.....	65
4.2 CRIANDO BASE MYSQL PARA GUARDAR A ONTOLOGIA.....	69
4.3 INSERINDO ONTOLOGIA NO MYSQL USANDO API RAP.....	69
4.4 LENDO MODELO NO BANCO DE DADOS UTILIZANDO API RAP.....	70
4.5 CONSULTAS SPARQL UTILIZANDO API RAP.....	71
4.5.1 Lógica para reconhecer a linguagem natural.....	71
4.5.2 Consultas usando API RAP E RDQL.....	72
4.5.3 Inserindo dados fictícios de alunos.....	73
5 RESULTADOS.....	74
5.1 RESULTADOS DA APLICAÇÃO DESENVOLVIDA.....	74
6 CONCLUSÃO.....	78
7 TRABALHOS FUTUROS.....	80
7.1 LINGUAGEM NATURAL.....	80
7.2 FRAMEWORK PARA CRIAÇÃO DE ONTOLOGIAS A PARTIR DE BANCO DE DADOS EXISTENTES. .	81
7.3 DESENVOLVIMENTO DE FRAMEWORK COMPLETO DA WEB SEMÂNTICA.....	81
8 REFERÊNCIAS.....	83

1 INTRODUÇÃO

Segundo Daconta, Obrst & Smith, (203), a Web sofreu dramaticamente uma mudança quanto a disponibilidade de informação eletrônica acessível e uma enorme quantidade de informações sobre os mais diversos assuntos que surgem todos os dias na Web.

Como as pessoas podem hoje acessar grandes quantidades de dados muito rapidamente, podem tornar-se sobrecarregadas com informação e, em alguns casos, a informação pode não ser útil para elas segundo Thuraisingham, (202). Com o objetivo de resolver estas questões, está se desenvolvendo a Web Semântica.

Com o crescimento da Web e a ausência de estratégias abrangentes e satisfatórias para recuperação eficiente de informações disponíveis em toda a Web, vieram os problemas na recuperação de informações nas páginas da Web, bem como o contexto da informação recuperada em relação à necessidade do usuário.

Atualmente a recuperação se dá somente por meio de ferramentas de busca, que utilizam palavras-chave extraídas do conteúdo dos documentos. Essa estratégia de recuperação é pouco eficaz, principalmente pela dificuldade de contextualização da informação, fazendo com que seja pobremente descrita e pouco passível de ser consumida por máquinas e seres humanos (Souza et al. 204).

As primeiras empresas que apresentaram uma solução para gerenciar esse excesso de informação foram o Alta vista, Yahoo, Google, filtrando as informações usando metainformações (metainformações são informações que descrevem outras informações, como palavras chave). Contudo, esse método se mostrou ineficiente, devido à quantidade de informação indesejada que o usuário comum recebe ao realizar uma simples busca (BRANDÃO & LUCENA, 202).

Uma solução para o problema foi proposta por Tim Berners-Lee (fundador da W3C - World Wide Web Consortium e criador da WWW - World Wide Web), defendendo a ideia de hipertexto semântico, nas quais os hiperlinks (ligações entre

páginas) expressariam relacionamentos entre os documentos. Contudo o HTML não oferece suporte para o tipo de construção Semântica devido ao fato que a linguagem é fechada e não há como criar TAGS específicas, sendo necessário o uso do XML (Linguagem extensível de marcação). O XML tem como utilizações básicas, o uso de TAGS específicas para descrever tipos de dados contidos nos documentos e Ontologias que definem conceitos de um domínio de conhecimento de forma a evitar ambiguidade da linguagem (BERNERS-LEE, 1998).

De acordo com Berners-Lee, (1998) a representação de conteúdo na Web se torna inteligível e processável por agentes de software graças a Semântica imposta pelo XML e abre caminho para novas tecnologias como RDF (*Resource Description Language*) e o mais recente OWL (*Web Ontology Language*), linguagens estas que permitem a expressividade usada na Web Semântica, mais detalhes no capítulo destinado a linguagens. Surge, neste contexto, o projeto da Web Semântica, onde as informações possuem significado bem definido, permitindo uma melhor interação entre máquinas e usuários.

O objetivo deste trabalho foi apresentar conceitos necessários para se compreender todo o processo que envolve o desenvolvimento usando os preceitos da Web Semântica, bem como linguagens específicas da mesma e ferramentas disponíveis.

A realização desta pesquisa se deu, com o intuito de propor uma nova funcionalidade para o sistema de gerenciamento acadêmico da Rede Doctum de Ensino/FIC, onde informações gerenciais possam ser dinamicamente criados de acordo com a necessidade do usuário de forma dinâmica. Para isso, viu-se a possibilidade com o uso de padrões da Web Semântica. Também visa a inserção da Web Semântica no meio acadêmico, para que possam ter mais trabalhos nesta área e, assim, contribuir para uma possível mudança na Web, como é conhecida atualmente.

Um dos objetivos do presente trabalho é o desenvolvimento de uma aplicação de buscas de informações acadêmicas dos alunos, a partir do que for solicitado pelos usuários, bem como a listagem de alunos de um determinado curso, por exemplo.

Na metodologia, foram apresentadas ferramentas e linguagens, foram utilizadas para o desenvolvimento de um sistema de busca semântico, bem como a criação de um modelo de dados, chamado “Ontologia”, que a partir do qual é possível extrair as informações nelas contidas, utilizando ferramentas criadas para este fim.

Foi concluído na execução deste trabalho que o uso de padrões da Web Semântica, capacita aplicações na web ao compartilhamento de recursos e informações, dessa forma tornando as aplicações e a Web como um todo mais eficiente.

2 REFERENCIAL TEÓRICO

Toda a parte referencial na qual esse trabalho está baseado, descrevendo as principais linguagens e ferramentas que são necessárias para a aplicação da Web Semântica, além das técnicas e padrões para da Web Semântica. São descritos conceitos sobre Semântica, a fim de esclarecer a importância da mesma. Introduz o conceito inicial da Web Semântica, bem como a estrutura atual da mesma de acordo com Hendler; Lassila, (201), os metadados e seus padrões. As linguagens de marcação como XML, RDF, e OWL bem como suas características e funcionalidades. Também pode-se verificar sobre a linguagem de consulta SPARQL que é uma linguagem de consulta a banco de dados, bem como suas características básicas para consultas em banco de dados relacional. É visto também sobre ontologias, bem como suas características e necessidade de criar um domínio de conhecimento.

2.1 Semântica

Segundo Roberta P. (201), pode-se afirmar que a Semântica busca descrever o “significado” das palavras e das sentenças, para ela não há consenso entre os semanticistas sobre o que se entende por significados. Segundo Irène T. (211), a Semântica tem por objetivo a descrição das significações próprias às línguas e sua organização teórica.

2.2 Web Semântica

A Semântica é uma palavra de origem grega (semantiké) que traduzida seria "a arte da significação". É o estudo do significado das palavras e de sua relação de significação nos signos linguísticos. Web Semântica é uma extensão da Web atual, porém, com todas as informações organizadas, de forma que não somente seres humanos possam entendê-la, mas principalmente máquinas que por sua vez auxiliarão em tarefas que atualmente são feitas manualmente.

Atualmente, as ferramentas de buscas existentes na Web, como o Google, por exemplo, dependendo do que se busca, não alcançam resultados satisfatórios logo de início, sendo necessário filtrar melhor a busca, isso ocorre porque os resultados de determinada pesquisa, podem não atender as necessidades do usuário.

Pressupondo então que, determinado usuário queira saber a palavra "manga" é uma fruta ou parte de uma camisa, pois, manga pode ser tanto manga de fruta ou manga de camisa, portanto se o usuário pesquisar por manga ele terá como resultado tudo que conter nome "manga" devido ao fato de não ser tratado a Semântica, ou seja, o sentido da pesquisa. Isto se deve graças à maneira como os documentos da Web são estruturados onde recuperação das informações se baseia em palavras-chave contidas em documentos ou páginas Web. Uma Web de dados e não de informação, pois não se extrai diretamente a informação desejada, sem nenhuma estratégia indexação dos documentos dificultando a determinação dos contextos das informações e a pertinência com relação ao que se está buscando.

Com uma Web Semântica, a Web passaria a ser uma Web de mais informações, nas quais não só apenas são retornadas informações satisfatórias e objetivas, como também poderá ser aplicada em serviços que rodam na Web. Um exemplo prático pode ser visto da seguinte forma: Determinado usuário quer comprar uma passagem aérea seguindo as restrições:

- O voo tem que ser à noite.
- Na classe econômica B.

- E com a companhia aérea que possui o preço da passagem mais barata.

O computador, em poucos momentos, retorna a busca com o nome da companhia que atende às restrições impostas, ficando a cargo do usuário apenas o ato de reservar seu lugar. Este serviço já é disponibilizado na Web atualmente, contudo, o próprio usuário é quem tem que procurar de acordo com o que deseja.

A Web Semântica consiste, na materialização da proposta de Tim Berners-Lee, o criador da Web, de dotar a Internet, com uma representação Semântica compartilhada, de forma que possa ser interpretada simultaneamente por seres humanos e máquinas, permitindo, assim, a inferência automática de conteúdo, futuros estados e ações. (Ramalho, 206)

Berners-Lee, (201) afirma que a Web Semântica ou Web 3.0, como é conhecida por ser a terceira geração da Web, não é uma Web separada, mas uma extensão da atual.

Ainda segundo Berners-Lee, (201), nela as informações terá significados bem definido, permitindo por sua vez uma melhor interação entre computadores e pessoas. Dessa forma, possivelmente ocorrerá uma transformação que virá a modificar a Web da forma em que a conhecemos. Sua proposta é de que programas e dispositivos especializados interajam por meio da infraestrutura dos dados da internet, trocando informações ou serviços a fim de automatizar tarefas rotineiras.

Segundo Breitman, (205), as informações contidas em documentos e páginas na Web, terão significado bem definidas usando linguagens de marcações Semânticas como XML. Será possível saber, por exemplo, o assunto tratado em cada página na Web ou em cada documento.

Web Semântica segundo Decker et al. (2000) é o nome do projeto, criado pelo W3C, que pretende embutir inteligência e contexto nos códigos XML(Linguagem de marcação) utilizados para confecção de páginas na Web, a fim de melhorar a forma com que programas interagem entre as páginas, e também possibilitar um uso mais intuitivo por parte dos usuários.

Contudo, até chegar a um patamar de confiabilidade aceitável para que seres humanos e máquinas possam interagir e trocar informações representadas semanticamente através da Web, existe um grande desafio da Web Semântica que consiste na interpretação automatizada a partir de conteúdos previamente estruturados (IS-KOLD, 208).

Para atingir esse propósito, Tim Berners-Lee, (201) afirma que uma padronização de tecnologias como linguagens de marcação e metadados, devem ser adquiridas por todos os usuários da Web para que seja possível a integração de todas as informações e serviços.

De acordo com Ramalho, (206) e Horrocks & Road, (202) A Web Semântica tem evoluído bastante com a utilização de ontologias, que agora desempenham um papel central. permitindo a troca das informações referentes ao conteúdo disponível na Web, funcionando como fonte de termos precisamente definidos, possibilitando o compartilhamento e a manipulação de informações que possam ser interpretadas computacionalmente de maneira automática à partir da lógica computacional.

A utilização de ontologias amplia o leque de opções de aplicações na Web, não apenas mantendo organizados os dados não estruturados, como, por sua vez, permite a acomodação destes dados em banco de dados subjacentes.

De acordo com Ramalho, (206) as linguagens de marcação e as ontologias contribuíram para que fosse possível tornar os computadores capazes de interpretar, através de regras lógicas, informações sobre determinado domínio qualquer, previamente bem definido. Será visto, nas próximas seções, a definição destes conceitos, e como trabalhar com eles de forma simples, usando ferramentas praticas.

2.2.1 Estrutura atual da Web Semântica

A Figura 1 mostra as camadas da Web Semântica conforme a visão de Lee; Hendler; Lassila, (201). Estas camadas estão dispostas de maneira crescente no que diz respeito à representatividade da Web e do seu conteúdo pela máquina e pela proximidade dessa representatividade, com o conceito que as pessoas tem do seu mundo.

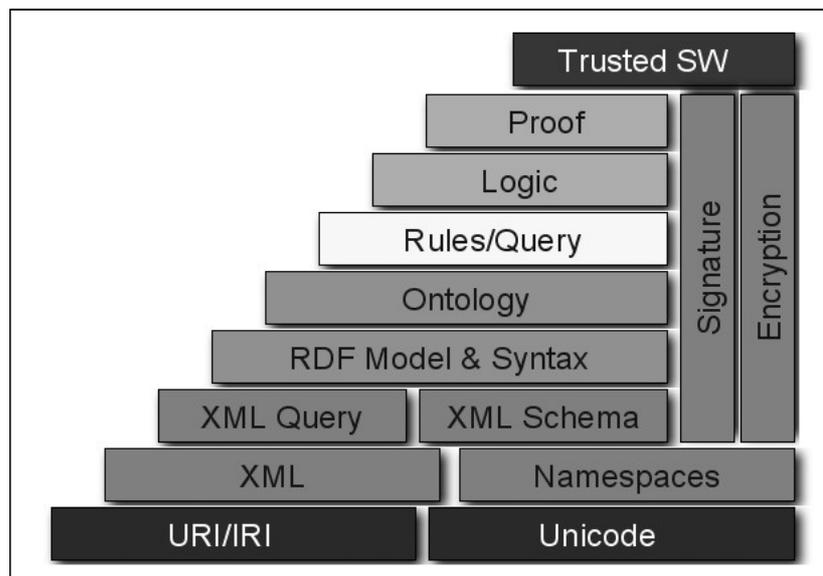


Figura 1: Camadas da Web Semântica - Fonte Lee; Hendler; Lassila (201).

Estas Camadas estão em constante evolução, sendo que a própria estrutura tem sofrido alterações razoavelmente freqüentes na medida em que alguns componentes são substituídos por outros mais eficientes.

Bravo, (210) realizou a seguinte descrição da estrutura:

- **URI (Uniform Resource Identifier ou identificadores únicos)**: é um identificador único que estabelece o uso de IRI (Internationalized Resource Identifiers ou identificadores Internacionais únicos) como padrão a fim de

garantir a codificação de caracteres internacionais e por sua vez identificar recursos na Web Semântica de forma única uma vez cada serviço ou documento na Web tenha seu próprio URI(Internationalized Resource Identifiers ou identificadores Internacionais únicos).

- **XML (Extensible Markup Language):** é uma linguagem de marcação extensível (ou flexível porque aceita criação de Tags personalizadas) que possui definições de metadados (dados sobre dados que tem a função de descrever características de determinado dado), que por sua vez permitem a integração de todas as definições feitas na Web Semântica com outros padrões baseados em XML.
- **RDF (Resource Description Framework):** linguagem criada para descrever objetos, documentos ou serviços com URIs utilizando vocabulários que podem ser referenciados por outros URIs e assim atribuir definições de tipo para recursos.
- **ONTOLOGY (Ontologia):** é a camada responsável pela ontologia, que por sua vez, suporta a evolução dos vocabulários entre ontologias diferentes, criadas com vocabulários diferentes. Esta camada é a responsável por definir o domínio de determinado assunto que se possa trate.
- **QUERY (Camada de Consulta):** Camada da Web Semântica responsável por realizar tarefas no banco de dados. Para esta camada foi desenvolvida uma linguagem própria que possibilita a realização de busca de Informações na Web Semântica em conjunto com todas as demais camadas.
- **RULE:** permite a troca de informações entre sistemas baseados em regras.
- **CRYPTO (Criptografia):** camada responsável pela segurança e que fornece padrões de assinatura digital e criptografia necessária para garantir a segurança na troca de informações da Web Semântica.
- **UNIFYING LOGIC (Lógica):** contém os padrões necessários para definir linguagens lógicas com expressividade suficientes para implementação de futuros novos mecanismos de inferência na Web Semântica.

- **PROOF:** é a camada responsável por prover os serviços corretos para a validação das definições feitas pela Web Semântica.
- **TRUST:** é responsável por oferecer a confiabilidade da Web Semântica através de mecanismos que permitem definir políticas de segurança e regras de acesso às fontes de informação.
- **USER INTERFACE & APPLICATIONS:** camada formada de acordo com os sistemas que utilizam a Web Semântica como plataforma de execução. Esta camada fica a critério da aplicação, onde será utilizado a Web Semântica. No caso deste trabalho é a interface de busca.

As Camadas “UNIFYING LOGIC”, “PROOF” E “TRUST”, ainda estão em desenvolvimento, segundo (Bravo, 210), e são consideradas plataformas informantes para pesquisa na Web Semântica.

Estas estruturas são a base da Web semântica, seguindo essa estrutura é possível criar aplicações semânticas obedecendo estas camadas que por sua vez em conjunto formam um novo padrão de desenvolvimento.

2.3 METADADOS

Os metadados são documentos na Web compostos por uma mistura de dados e “metadados” (dados sobre dados). Eles têm a função de especificar características dos dados, descrevendo a forma com que serão utilizados, exibidos, ou mesmo seu significado em um contexto (Breitman, 205).

“No contexto da Web, cinco aspectos devem ser destacados, Descrição de recursos: Informação expressa através de metadados, determinada pelo objetivo e tipo de recurso. Produção de metadados: produção de descrições de dados, que podem tornar-se um processo extremamente caro e cansativo, quando realizado manualmente, sendo a tendência realizar automaticamente esse processo incentivado pelo uso de XML e RDF, que

será visto mais à frente. Devido ao grande número de padrões de metadados existentes é possível descrever um recurso a partir de mais de um conjunto de descritores. Deve existir um conjunto de padrões de metadados específicos de forma a acomodar os diferentes tipos de recursos. Metadados são dados. Assim, apresentam também características de armazenamento e acesso, bem como dificuldades de interpretação de seu conteúdo” (Iannella, R., 1997).

2.3.1 Padrões de Metadados

Segundo Souza, Santos & Oliveira, (2000) os padrões de metadados fornecem definições de como formar uma rede a fim de automatizar registros e as propriedades incluindo também os dados de forma organizada e consistente.

A organização e descrição dos recursos e dados corretamente, podem contribuir para com resultados positivos para isso faz-se necessário o uso de um padrão, na Web Semântica o padrão usado é o Dublin Core segundo Breitman, (2005), este padrão é visto na próxima seção.

2.3.2 Dublin Core – DC

De acordo com Souza et al. (2000) Dublin Core é um padrão internacional para descrição de recursos de informação, considerado parte importante da infraestrutura da Internet, desenvolvido em 1994, por um grupo de bibliotecários e especialistas de conteúdo, encabeçado por Stuart Weibel, pesquisador da OCLC (Online Computer Library Center).

“É chamado "Dublin Core" por ter se originado em um workshop na cidade de Dublin, Ohio, Estados Unidos. Todas as ações e esforços são liderados pelo "Dublin Core Meta data Initiative" que é o responsável pelo desenvolvimento, padronização e promoção do conjunto de elementos metadados Dublin Core” (Souza et al. 2000).

O Dublin Core tem uma vantagem maior para descrever recursos eletrônicos disponíveis na internet, segundo (Weibel, 1997).

De acordo com Weibel, (1997) as principais características do Dublin Core são:

- Simplificar a criação e manutenção de recursos para que seja possível a recuperação efetiva dos recursos na rede.
- Capacitar o entendimento da semântica de forma objetiva nos documentos e páginas a fim de se obter com exatidão informações sobre os mesmos, como por exemplo, quem é o “criador” de uma página, ou documento, ou quaisquer informações relacionadas.
- Ser reconhecido em âmbito internacional e disponibilizado em varias linguagens se tornando multilíngüe e multicultural do universo de informação eletrônica.
- Extensibilidade fornece mecanismos para integração com outros padrões e dando a possibilidade para adicionar novos padrões e novos elementos de acordo com o perfil da aplicação.

“O padrão de metadados Dublin Core é um elemento simples, mas eficaz conjunto para descrever uma ampla gama de recursos de rede. O padrão Dublin Core inclui dois níveis: Simples e Qualificado. Dublin Core Simples inclui quinze elementos, o Qualificado Dublin Core inclui três elementos adicionais (Audiência Proveniência e Detentor de Direitos), assim como um grupo de refinamentos de elementos (também chamados qualificadores), que refinam a Semântica dos elementos de maneira que podem ser úteis para recurso descoberta. A Semântica de Dublin Core tem sido estabelecido por uma organização internacional. O grupo interdisciplinar de profissionais de biblioteconomia, ciência da computação, codificação de

texto, a comunidade do museu, e outras áreas afins do saber e da prática”.
(Dublin Core, 205)

O W3C procura prover padrões de compartilhamento de informações segundo A. Almeida et al. (1999) e o melhor e mais bem estruturado que caso do Dublin Core.

O Padrão Dublin Core possui documentação detalhada disponibilizada em sua documentação, onde podem ser encontradas as URIs de importação do padrão pode ser encontrado em (Dubin Core, 211).

2.4 LINGUAGENS

Atualmente a linguagem mais utilizada nas páginas da Web é o HTML, ou linguagem de marcação em texto, que é uma metalinguagem que vem da ideia de uma outra linguagem chamada SGML, responsável por descrever outras linguagens.

SGML tem como princípio de que documentos contêm uma estrutura, e o conjunto dessa estrutura faz referência à forma com que serão exibidos. A isto é dado o nome de TAGS (marcações sintáticas com função de descrever dados para manipulação de documentos). (A.ALMEIDA ET ALL, 1999).

De acordo com A. Almeida et. al. (1999) o HTML é um conjunto de TAGS(linguagens de marcação) com o princípio criado para exibição de documentos na Web. Ele herdou do SGML, a possibilidade de ser interpretado pelos navegadores, uma vez que o SGML define o HTML como fixo, deixando o HTML rígido, não permitindo a possibilidade de adição de novas TAGS.

Visto que o HTML, não possui características para descrever a Semântica dos conteúdos e seu respectivo significado dentro de um contexto, surgiu o XML que possui esta característica e outras que podem ser visto na próxima seção.

2.4.1 XML

O XML é recomendado formalmente pelo W3C conforme pode-se ver no endereço <http://www.w3.org/TR/202/REC-xhtml1-20281/>, para descrição Semântica dos dados, pois assemelha-se ao HTML e, ambas, são derivadas do SGML e contêm TAGS para descrever o conteúdo de um documento. Contudo a diferença de que o HTML tem como objetivo controlar a forma que os dados são exibidos. Já o XML foca na descrição dos dados que o documento contém, além de ser flexível no sentido de se poder ser acrescentadas novas TAGS a medida que forem necessárias, ao contrário do HTML.

“O grande diferencial do XML está na sua própria sintaxe: o produtor do documento escolhe suas próprias marcações e define como eles irão ser nomeados”. “Um documento escrito com a sintaxe XML é, na verdade, uma aplicação XML, com um conjunto de marcas escolhidas por seus criadores para aquele documento em particular” (TOLENTINO, 204).

O XML pode ser intercalado entre trechos do texto em determinada página na Web, mas ao contrário da HTML, são estruturados através de hierarquias que definem sua relevância, além atribuírem valores e relacionamentos contextuais no documento.

Um exemplo de marcação utilizando a marcação XML é visto mais a frente na Figura 2 de acordo com F. Oliveira, (204), exemplificando um Catálogo de produtos, de acordo com padrão da W3C. Pode-se apontar como os principais benefícios no uso do XML em aplicações Web:

- O acesso facilitado à informação através de atribuição Semântica aos dados.
- O desenvolvimento de aplicações flexíveis.
- A integração de dados de diferentes origens.

- A completa separação entre conteúdo, estrutura e formatação dos documentos.

De acordo com Souza et al. (204) o XML possui uma ótima flexibilidade, contudo não é o bastante, pois ele não é capaz de reconhecer a Semântica por ele imposta, ou seja, ele auxilia para que outro agente venha e possa semanticamente interagir.

Segundo Souza et al. (204), o padrão XML é aceito como o padrão emergente para troca de dados na Web. Mas, apesar de possibilitar aos autores a criação de suas próprias TAGS, em uma perspectiva computacional. Há muito pouca diferença entre as TAGS “<AUTHOR>” e “<CREATOR>”.

Para que as marcações Semânticas criadas sejam utilizadas de forma não-ambígua por comunidades maiores, são necessários alguns padrões de compartilhamento mais universais.

Será visto em seções seguintes sobre a linguagem RDF que será a responsável pela realização da tarefa.

2.4.1.1 URI e Namespaces no XML

De acordo com T. Berners-Lee et. al, (204), a Uniform Resource Identifier (URI) tem a função de identificar recursos abstratos ou físicos com uma sequencia de caracteres única, ou seja, o Identificador Universal de Recursos URI, provê uma forma única de identificação e localização de recursos da mesma forma como é feita com documentos, imagens e musicas através do URL.

De acordo com ETSI (211), o XML usa URIs para identificar claramente um módulo de XML, ou um objeto definido dentro de um arquivo XML, pois podem existir vários esquemas URI diferentes neste arquivo.

No XML, podem ocorrer conflitos quando diferentes documentos usam o mesmo nome referenciando e ambos possuem diferentes tipos de elementos ou serviços. Para que isso não ocorra são usados *namespaces* para evitar esse conflito de nomes. Será visto à frente sobre *namespaces*.

Abaixo na Figura 2 exemplo de arquivo XML sem tratamento de conflitos em seus módulos de acordo com F. Oliveira, (204).

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <catalogo>
3   <nome>Catalogo de produtos</nome>
4   <produtos>
5     <produto>
6       <nome>Arvore Rubro-Negra</nome>
7       <preco>20,00</preco>
8     </produto>
9     <produto>
10      <nome>Arvore Binaria</nome>
11      <preco>5,00</preco>
12    </produto>
13    <produto>
14      <nome>Arvore B</nome>
15      <preco>25,00</preco>
16    </produto>
17    <produto>
18      <nome>Grafo</nome>
19      <preco>15,00</preco>
20    </produto>
21    <produto>
22      <nome>Grafo Dirigido</nome>
23      <preco>20,00</preco>
24    </produto>
25  </produtos>
26 </catalogo>

```

Figura 2: Catálogo de produtos em XML – Fonte F. Oliveira, (204).

Na Figura 2, pode-se notar que todos os produtos estão dentro de um módulo, e todos os módulos são idênticos com mesmo nome “<produto>”, desta forma ocasionando, de acordo com F. Oliveira (204), conflitos de nomes.

De acordo com F. Oliveira, (204) e Souza et al (204) URIS são usados, então, para definição única de serviços ou objetos disponíveis para que assim não ocorram problemas na identificação destes, de modo garantir que não haja ambigüidades.

Segundo W3C (1999), *namespaces* é a coleção de nomes identificados por uma referência, estes que são usados no XML, como tipos de elementos e nomes de atributos. É ele que provê um método para evitar conflito de nomes de elementos.

Segundo Langer, (1969), *namespaces* é uma coleção de nomes identificados por uma referencia URI, usado nos documentos XML como prefixo dos nomes de elementos e atributos, colocando o atributo *namespace* no início da TAG de um elemento. Sintaxe: `xmlns:namespace-prefix="namespace"` Exemplo: `xmlns:f="http://www.w3schools.com/furniture"`. Langer, (1969) cita dois exemplos que deixa claro como deve ser feito para se eliminar conflitos com *namespaces*:

Eliminar conflito com namespaces

Ficheiro 1:

```
<h:table xmlns:h="http://www.w3.org/TR/html4/">
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>
```

Ficheiro 2:

```
<f:table xmlns:f="http://www.w3schools.com/furniture">
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

Figura 3: Conflitos com Namespaces do XML – Fonte Langer, (1969).

De acordo com Langer, (1969) diferenciando, antes do início de cada tabela, colocando o `<h:` e o `<f:` ele garantiu de que os mesmos não serão idênticos, podendo então ser tratados e referenciados de forma única, visto que se ambos estivessem sem esta forma única para serem usados, ou seja, um URI único, poderiam causar, de acordo com o autor, uma ambiguidade ao se tentar referenciar esses serviços, ou chegar a estas informações.

2.4.2 RDF

Conforme visto nas seções anteriores, usando apenas o XML não pode ser garantido um ideal de compartilhamento de dados, apesar da a linguagem garantir uma melhor estrutura e conseguir tratar ambiguidades.

Com o intuito de sanar esse problema, foi desenvolvido pela W3C o RDF baseado em Grafos Semânticos, que utiliza, segundo (W3C, 204a), a mesma filosofia por trás da linguagem orientada a objetos, fornecendo construções como herança e associação onde todas as entidades são representadas como nós em um grafo e, seus atributos são atributos, do nó.

RDF é um modelo de dados, padrão, para intercâmbio de dados na Web e tem características que facilitam a fusão de dados, mesmo se os esquemas subjacentes diferem, e apóia especificamente a evolução dos esquemas ao longo do tempo sem a necessidade de todos os consumidores de dados a serem alterados.

“RDF amplia a estrutura de ligação da Web para usar URIs para citar a relação entre as coisas, assim como as duas extremidades do link (isto é normalmente referido como um "triplo"). Usando este modelo simples, ele permite que os dados estruturados e semi-estruturados para ser misturado, exposta, e compartilhados entre aplicações diferentes.”(W3C,204b)

De acordo com W3C, (208), o RDF é um formato de dados para representação de informações na Web que define a sintaxe e a Semântica da linguagem de consulta SPARQL (SPARQL PROTOCOL AND RDF QUERY LANGUAGE) para RDF. De acordo com F. Oliveira, (204) na Figura 4 é possível visualizar o uso de grafo em RDF:

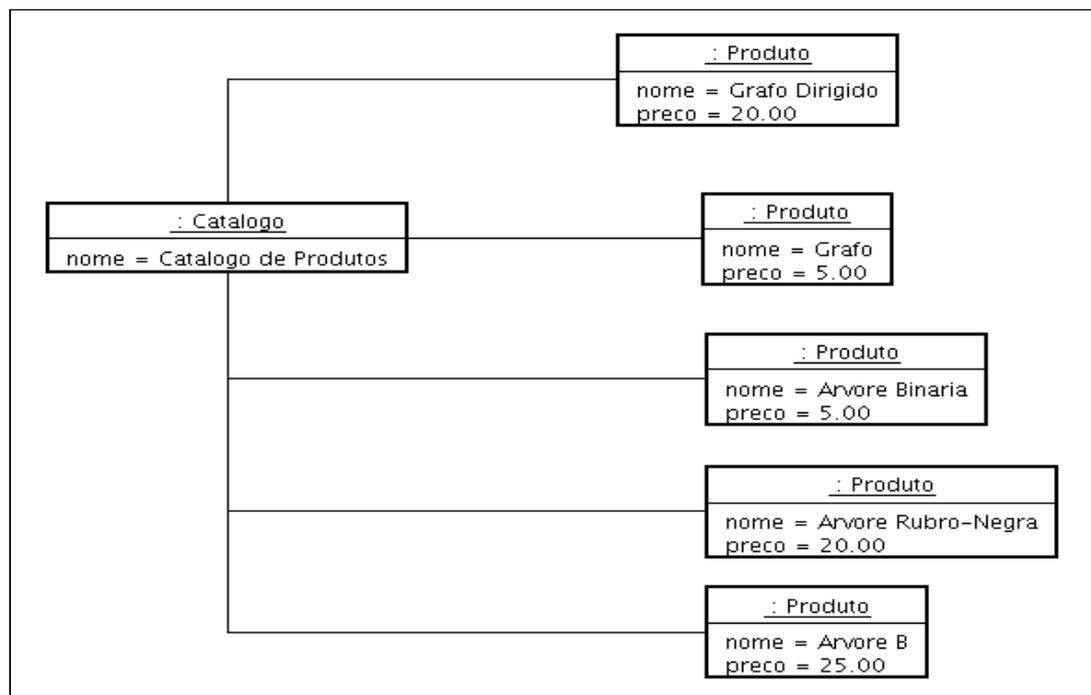


Figura 4: Catálogo de produtos usando grafo RDF – Fonte F. Oliveira, (204).

De acordo com a Figura 4, pode-se ver um grafo RDF de um catálogo de produtos onde se tem cada produto visto como uma ramificação do catálogo.

Será visto em “Gráfico Modelo de dados RDF”, como é a estrutura do RDF citada pelos autores a fim de esclarecer como são feitas.

2.4.2.1 Conceitos Básicos RDF

Para definir RDF é preciso que se tenha ciência de seus atributos, como de acordo com W3C, (204c.) esta linguagem é formada com a junção das seguintes ferramentas:

- Grafo de Modelo de Dados (*Graph data model*).
- Vocabulário baseado em URIs (*URI-based vocabulary*).

- Tipos de dados (*Datatypes*).
- Literais (*Literals*).
- Sintaxe de serialização XML (*XML serialization syntax*).
- Expressão de Fatos Simples (*Expression of simple facts*).
- Vinculação.

Sera visto nas próximas seções as devidas definições para estes conceitos.

2.4.2.2 Grafos Modelo de dados RDF

A Estrutura dos RDF é uma coleção de triplas, cada uma construindo um sujeito, um predicado e um objeto. Grafos RDF (Resource Description Framework), tem a finalidade de criar um ambiente gráfico para a construção de modelos.

De acordo com W3schools, (211), usando XML dentro do RDF é possível a troca de informações entre computadores que utilizam diferentes tipos de sistemas operacionais ou linguagens de aplicação, e esta junção é denominada por XML/RDF, onde, para representar as triplas formalmente, em linguagem compreensível pela máquina, o RDF utiliza o XML.

Segundo W3C, (204b) cada conjunto de triplas é chamado de RDF, que por sua vez, pode ser visto como um nó no qual cada tripla é representada em uma ligação. É desta visão que vem a ideia de gráfico como visto na Figura 5.



Figura 5: Tripla RDF – Fonte W3C, (204b)

Conforme demonstrado na Figura 5, presume-se que cada tripla é composta por um sujeito, um predicado e um objeto. É possível verificar que uma tripla tende

sempre em função do objeto relacionado pelo predicado identificando o Sujeito de acordo com W3C, (204b). Cada tripla representa:

- Um assunto.
- Um objeto.
- Um predicado (também chamado de propriedade).

A direção do arco é significativa, pois ele sempre aponta para o Objeto, visto que os nós de um grafo sempre serão os objetos, portanto, tudo o que se for denotado pelo predicado está sempre relacionado ao sujeito e ao objeto de acordo com W3C, (204b). Cada tripla representa uma afirmação sobre um relacionamento entre as entidades representadas pelos nós. De acordo com F. O. Gauthie, (211), as afirmações representadas pelas triplas no conjunto sujeito-predicado-objeto são chamados de *statements*.

2.4.2.3 Vocabulário baseado em URIs

Segundo (F. O. Gauthie), cada nó do grafo pode ser uma URI, um literal ou estar em branco (dessa forma não podendo ser identificado). Os predicados são identificados por URIs e podem ser também nós no grafo.

2.4.2.4 Tipos de dados

Segundo W3C, (204b) os tipos de dados utilizados por RDF representam valores como números inteiros, números reais e datas ,sendo que consiste de um espaço léxico, um espaço de valores e um mapeamento léxico-para-valor.

O tipo de dado é identificado por uma ou mais URIs. O RDF não fornece nenhum mecanismo para a definição de novos tipos de dados, cabendo ao XML a definição de uma estrutura de extensibilidade adequada para a definição de novos tipos de dados para uso em RDF.

2.4.2.5 Literais

São utilizados em RDF os literais que identificam valores como números e datas por meio de uma representação léxica e podem ser o objeto de uma declaração RDF e não o sujeito ou o predicado segundo (W3C, 204b).

Os literais podem ser:

- Plain – são strings relacionadas com tags opcionais da linguagem.
- Tipados – são strings combinadas com determinada URI de tipo de dados, dois exemplo segundo W3C, (204b) seria `<xsd:boolean, "true">` e `<xsd:boolean, "false">`.

2.4.2.6 Sintaxe de serialização XML (XML serialization syntax)

Existe uma forma recomendada de serialização de RDF em XML de forma a facilitar a troca de modelos RDF entre aplicações. (W3C, 204 d). Abaixo na Figura 6, de acordo com F. Oliveira, (204), um exemplo de como se pode representar um RDF em formato XML:

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <?xml-stylesheet type="text/xsl" href="catalog.xsl"?>
3 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4   xmlns:termos="http://www.ime.usp.br/termos/"
5   xmlns:catalogo="http://www.ime.usp.br/catalogo/">
6   <rdf:Description rdf:about="http://www.ime.usp.br/catalogo">
7     <catalogo:produtos>
8       <rdf:Bag>
9         <rdf:li>
10          <rdf:Description rdf:about="http://www.ime.usp.br/produtos/rubronegra">
11            <termos:preco>20,00</termos:preco>
12            <termos:nome>Arvore Rubro-Negra</termos:nome>
13          </rdf:Description>
14        </rdf:li>
15        <rdf:li>
16          <rdf:Description rdf:about="http://www.ime.usp.br/produtos/arvorebinaria">
17            <termos:nome>Arvore Binaria</termos:nome>
18            <termos:preco>5,00</termos:preco>
19          </rdf:Description>
20        </rdf:li>
21        <rdf:li>
22          <rdf:Description rdf:about="http://www.ime.usp.br/produtos/arvoreb">
23            <termos:nome>Arvore B</termos:nome>
24            <termos:preco>25,00</termos:preco>
25          </rdf:Description>
26        </rdf:li>
27        <rdf:li>
28          <rdf:Description rdf:about="http://www.ime.usp.br/produtos/graf">
29            <termos:preco>15,00</termos:preco>
30            <termos:nome>Grafo</termos:nome>
31          </rdf:Description>
32        </rdf:li>
33        <rdf:li>
34          <rdf:Description rdf:about="http://www.ime.usp.br/produtos/grafodirigido">
35            <termos:preco>20,00</termos:preco>
36            <termos:nome>Grafo Dirigido</termos:nome>
37          </rdf:Description>
38        </rdf:li>
39      </rdf:Bag>
40    </catalogo:produtos>
41    <termos:nome>Catalogo de produtos</termos:nome>
42  </rdf:Description>
43 </rdf:RDF>

```

Figura 6: Catálogo de produtos usando RDF/XML - Fonte F. Oliveira, (204).

De acordo com o exemplo citado anteriormente, pode-se verificar que na primeira linha, foi definida a versão do XML e a codificação na qual o documento se encontra, de acordo com definições estabelecidas pelo XML.

Na segunda linha pode-se verificar que está sendo feito uma referência a um catálogo de produtos no qual foi citado por F. Oliveira (204), e indicando o tipo (type) dessa relação, que é texto.

Da terceira à sexta linha constam à chamada do RDF no arquivo, onde a URI da sintaxe do RDF vista como `xmlns:rdf="http://www.w3.org/1999/2/22-rdf-syntax-ns#"`. Na 3ª linha, a URI dos termos vista como `xmlns:rdf="http://www.ime.usp.br/termos/"` na 4ª linha e, por último, a URI do catálogo vista como `xmlns:rdf="http://www.ime.usp.br/Catálogo"`.

Das linhas 7 a 40 entram os produtos conforme definição da estrutura do próprio XML, e finaliza na linha 41, identificando com nome para o catálogo de produtos, bem como finaliza corretamente na linha 42 e 43 a sintaxe do RDF.

De acordo com W3C, (210) é importante notar que todo o objeto que queremos utilizar para especificar um recurso no RDF/XML precisa ser unicamente identificado através de um URI (Identificadores uniformes de recursos) e sendo assim podemos representar recursos tanto dentro quanto fora da rede internet.

2.4.2.7 Expressão de Fatos Simples

Indicam relacionamento entre duas entidades que podem ser rerepresentados através de uma tripla RDF na qual o predicado nomeia o relacionamento e o sujeito e o objeto as duas entidades. Eles podem ser equivalentes a uma linha de determinada tabela no banco de dados que contenha duas colunas, uma indicando o sujeito e a outra o objeto ficando a cargo do predicado representar o nome da tabela segundo W3C, (204b).

Ainda, segundo W3C, (204b), no banco de dados a lógica de primeira ordem permite relações ou predicados com mais de duas entidades. Este tipo de relação ou predicado, por sua vez, é decomposto para representá-lo como uma tripla RDF. Dessa forma ele poderá fazer esta decomposição e introduzir em um nó em branco no grafo que por sua vez corresponde a uma tripla (conjunto de dados com URI única) que esteja armazenada em banco de dados.

O sujeito da tripla é o nó em branco, o predicado é dado pelo nome da coluna, e o objeto corresponde ao valor da célula. O mesmo se aplica para predicados da lógica de primeira ordem com mais de um argumento, como pode ser visto na Figura 7.

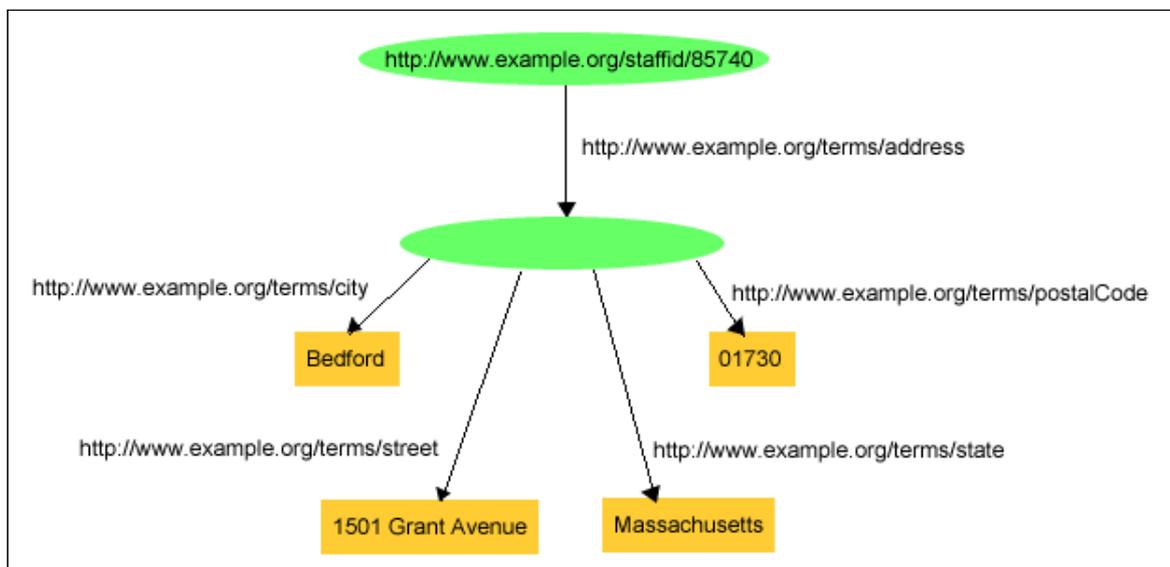


Figura 7: Exemplo de RDF e URIs – Fonte W3C. (204b).

Pode-se verificar, conforme vínculo na Figura 7, que através de determinada URI (Identificador único de recursos), vista como <http://www.example.org/staffid/85740>, o identificado foi o 85740 que, conforme pode-se verificar, buscava-se como objetivo o endereço de 85740 visto na aresta <http://www.example.org/terms/address>, que por sua vez pode-se chegar à informação desejada através das arestas <http://www.example.org/terms/city> para cidade, <http://www.example.org/terms/postalCode> para saber o código postal, <http://www.example.org/terms/street> para Rua e <http://www.example.org/terms/state> para estado. Desta forma chegou-se ao no objetivo que foi saber que o indivíduo único,

ou serviço representado na URI, está localizado na Cidade de Bedford, no código postal 1730, na rua 151 Grant Avenue e no estado de Massachusetts.

Na Figura 7 de acordo com W3C, (204b), a informação representada pode conter uma linha na tabela "*STAFFADDRESSES*" com uma chave primária "*STAFFID*" e colunas adicionais como estado, cidade e CEP (*STREET*). Para expressar relações mais complexas em RDF são feitas através de conjuntos (lógico-E) dos relacionamentos binários simples, pois, o RDF não expressa negação (NOT) ou Junção (OR).

De acordo com (204b), pode-se verificar que é possível representar relações de determinado banco de dados em RDF ou vice-versa, podendo, então, armazenar vários modelos RDFs, caso necessário, na Base de dados. Será visto mais a frente como trabalhar com consultas no banco de dados para que após a inserção do modelo RDF na base de dados, se possam recuperar as informações na seção linguagem SPARQL.

2.4.2.8 Vinculação

O Vínculo que uma expressão tem com outra expressão em RDF pode afirmar se ambas são verdadeiras ou falsas, ou apenas se uma delas for verdadeira ou falsa.

Segundo W3C, (204b), uma expressão RDF A é dito que, vincula-se à outra expressão RDF-B se todos os possíveis estados das coisas no mundo que fazem A verdadeira, também fazem B verdadeira, ou seja, se A é provada como verdadeira, então B pode ser inferida como verdadeira.

2.4.3 SPARQL – LINGUAGEM DE CONSULTA RDF

Nessa seção será destacado apenas como se realiza consultas utilizando a linguagem RDQL, pois o foco deste trabalho é na recuperação das informações, será proposta, para trabalhos futuros, a abordagem para estudo nas demais funcionalidades do RDQL. Para garantir essa integração, faz-se necessário uma linguagem de consulta, particularmente conhecida como SPARQL, projetada para atender esses requisitos pela W3C.

2.4.3.1 Namespaces no SPARQL

Segundo W3C, (208), o SPARQL assume ligação por padrão com os seguintes *namespaces*:

Prefixo (qualificador)	URI (namespace)
Rdf:	http://www.w3.org/1999/2/22-rdf-syntax-ns#
Rdfs:	http://www.w3.org/2000/1/rdf-schema#
Xsd:	http://www.w3.org/2001/XMLSchema#
fn:	http://www.w3.org/2005/xpath-functions#

Tabela 1: Definição padrão de Namespaces no SPARQL

Conforme apresentado na tabela 1, estes *namespaces* definem toda a sintaxe do RDF e XML Schema (Sintaxe do XML).

“SPARQL é usada para expressar consultas em diversas fontes de dados. Se os dados são armazenados nativamente no RDF, ele contém recursos

para consulta de padrões gráfico obrigatórios e opcionais juntamente com suas conjunções e disjunções. SPARQL também suporta valor extensível testes e restringindo consultas pelo gráfico fonte RDF. Os resultados das consultas SPARQL podem ser conjuntos de resultados ou gráficos RDF.”(W3C,208)

De acordo com M.Obitko, (207) e W3C, (208) SPARQL é uma linguagem SQL-Lite para consultas RDF, gráficos para expressar RDF na parte correspondente da consulta, explicando como seria uma consulta simples, um SELECT buscando nome e e-mails de acordo, como contido nos prefixos do Namespaces xmlns, descrito em “Dados” na Figura 8:

Dados:

```
@ Prefixo foaf: <http://xmlns.com/foaf/0.1/>.
_: A foaf: nome "Johnny Lee Outlaw".
_: A foaf: mbox <mailto:jlow@example.com>.
_: B foaf: nome de "Peter GoodGuy".
_: B foaf: mbox <mailto:peter@example.org>.
_: C foaf: mbox <mailto:carol@example.org>.
```

Consulta:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT? Nome? Mbox
ONDE
  {Foaf x: nome, nome?.
   ? X foaf: mbox mbox}
```

Resultado da Consulta:

nome	mbox
"Johnny Lee Outlaw"	<mailto:jlow@example.com>
"Peter GoodGuy"	<mailto:peter@example.org>

Figura 8: Busca Nome e e-mail utilizando RDQL - Fonte (W3C,208).

Segundo M.Obitko, (207), na primeira linha é definido de onde virão os dados, ou seja, o Namespaces, que é visto como “http://xmlns.com/foaf/0.1/”, nas duas últimas linhas é usado o prefixo “foaf”, conforme passado pelo Namespaces, para expressar um gráfico, ou seja, expressando os identificadores para “nome” e para “mbox”(Email), visto que estes identificadores comecem com o “?” (interrogação), que tem a função de identificar as variáveis. “Como o RDF trabalha com triplas, a

consulta está realizando buscas no recurso “?x”, que tem como predicado foaf:name e foaf:mbox, em busca dos sujeitos desse triplo que seriam ?Nome e ?Mbox.

De acordo com W3C, (208), padrões de triplas usadas em RDF pra cada sujeito, predicado ou objeto pode se ter uma variável. A fim de demonstrar, é feita uma consulta simples do título de um de um terminado livro, demonstrando, segundo o autor, a forma que deve ter uma consulta em RDQL na Figura 9:

Dados:

```
<http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> "Tutorial SPARQL".
```

Consulta:

```
SELECT? Título
ONDE
{
  <http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> título?.
}
```

Esta consulta, com os dados acima, tem uma solução:

Resultado da Consulta:

título
"Tutorial SPARQL"

Figura 9: Escrever uma Consulta Simples RDQL - Fonte W3C, (208).

Conforme visualizado na Figura 9, o autor W3C, (208) utilizou a base de conhecimento contida através da URI descrita em “Dados” para busca da “Consulta” obtendo o “Resultado da consulta”, com o título do livro pesquisado.

Segundo W3C, (208), uma consulta vem a ser uma seqüência de soluções, de acordo com a forma com que é realizada, podendo haver mais de uma solução para a mesma.

De acordo com W3C, (208) dessa forma é possível se recuperar informações básicas ou não utilizando o RDF juntamente com sua linguagem de consulta RDQL para representação das informações. Contudo, se fez trabalhoso essa idéia de criar toda essa estrutura sem nenhum apoio que facilitasse a forma de se criar o modelo RDF em conjunto com sua busca.

Será visto à frente, nas próximas seções o uso da ontologia para representar de melhor forma o uso do RDF, e também o uso de APIs que facilitam o uso do RDQL.

2.5 ONTOLOGIAS E OWL

O crescimento rápido e contínuo do volume de informações tornou cada vez mais difícil encontrar, organizar, acessar e manter informação requerida por usuários neste contexto surgiram as ontologias. A palavra Ontologia tem origem do grego: onto (ser) e logos (palavra). O termo Ontologia foi cunhado em 1613 por Rudolf Goclenius segundo A. Moreira & L. Alvarenga, (204).

Segundo Pickler, (207), o vocabulário ontologia foi introduzido no estudo da filosofia para distinguir o estudo do ser e o estudo de vários seres vivos existentes no mundo natural. tendo como objetivo fornecer sistemas de categorização para organizar a realidade.

“uma ontologia é uma especificação de uma conceituação. É designada com o propósito de habilitar o compartilhamento e reuso de conhecimentos, de forma a criar ‘compromissos ontológicos’, ou definições necessárias à criação de um vocabulário comum” (Souza & Lídia Alvarenga, 204).

De acordo com Breitman, (205), para o W3C, ontologias devem prover descrições para classes nos vários domínios de interesse, relacionamentos entre as classes e propriedades (atributos) que estas devem possuir.

Já A. Moreira & L. Alvarenga, (204) diz que o objetivo da construção de ontologias é a necessidade de um vocabulário compartilhado para troca de informações entre os membros de uma comunidade, sejam eles humanos ou agentes inteligentes.

“Dessa forma, as ontologias fornecerão o vocabulário necessário para a comunicação entre os agentes e as páginas da Web, definindo as relações entre os conceitos, como salienta” (Breitman, 205).

De acordo com Moreira & L. Alvarenga, (204), Breitman, (205) e Pickler, (207), e no que acredita o W3C sobre a função da ontologia, ela traz uma forma de se representar uma organização a ser feita para o relacionamento entre classes ou atributos que irão ajudar no reconhecimento do vocabulário, e assim tornar possível compartilhar informações entre humanos e máquinas.

Na ciência da computação se iniciou por volta dos anos 90, em projetos para grandes bases de conhecimento, de acordo com Pickler, (207).

A subárea da computação denominada Inteligência Artificial (IA) foi a primeira a utilizar ontologias para manipulação de bases de conhecimento, segundo A. Moreira & L. Alvarenga, (204).

“Ontologias são utilizadas em inteligência artificial. Web Semântica, engenharia de software, arquitetura da informação e em diversas outras áreas da Informática, como uma forma de representação de conhecimento sobre o mundo ou alguma parte deste. A sua utilização tem como objetivo estruturar de forma organizada as informações de um determinado domínio de conhecimento e refletir um entendimento semântico de situações do mundo real.”(Jardim et al. 206).

Com a evolução das ferramentas no decorrer do tempo, foi criado o OWL. Para atender de forma mais ágil e eficiente à criação de ontologias, veremos na próxima seção.

2.5.1 OWL

Neste contexto, ontologias oferecem um meio de lidar com a representação de recursos de informação: o modelo de domínio descrito por uma ontologia pode ser usado como uma estrutura unificadora para dar Semântica e uma representação comum à informação, contudo fez-se necessário a criação OWL (Web Ontology Language), que é uma linguagem criada para atender estes requisitos segundo W3C, (204e).

De acordo com W3C, (204e) para que as máquinas usem raciocínio, em cima dos documentos, é preciso mais que apenas o RDF ou XML, fazendo-se necessário o uso do OWL, que é uma linguagem criada pela W3c para representar maior poder de expressividade. Esta linguagem já possui incluídos os seguintes padrões:

- Padrão XML.
- Padrão XML-Schema: É responsável por evitar conflitos internos dentro do XML.
- Padrão RDF: É a linguagem RDF usada para trabalhar com os grafos e objetos, e as relações a fim de criar modelos para representação.
- OWL: Adiciona vocabulários, descreve as propriedades e as classes bem como as relações entre eles como disjunção, cardinalidade e igualdade sendo ela rica em propriedades e classes enumeradas.

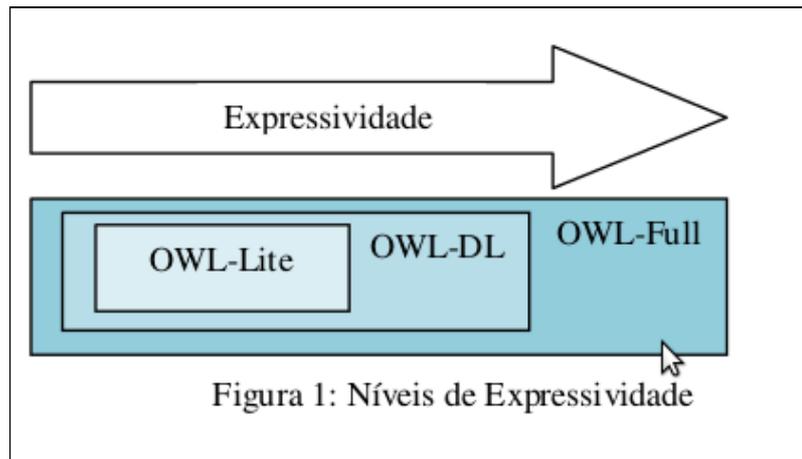


Figura 10: Nível de expressividade OWL – Fonte E.MARTINS & C.SALDIAS, (206).

O OWL possui por sua vez três sub-linguagens, conforme descrito na Figura 10 são elas:

- OWL Lite: segundo W3C, (204e), usada para tipos de restrições simples na qual os usuários não necessitam de algo muito complexo.
- OWL DL: segundo W3C, (204e), é usada para representar mais expressividade entre as relações entre as classes e suas propriedades.
- OWL FULL: segundo W3C, (204e), é usada para total expressividade e liberdade sintática do RDF, contudo não garante integração com softwares de raciocínio em cima do uso dessa aplicação tornando-a um pouco inutilizável.

De acordo com E.MARTINS & C.SALDIAS (206) o OWL Lite é utilizada quando é preciso apenas de uma modelagem de hierarquia de classes simples. O OWL DL tem como característica a lógica descritiva que possibilita automaticamente computar hierarquias de classes sendo extremamente determinística dando grande importância aos resultados computacionais e garante um retorno de tempo estabelecido, e o OWL FULL representa mais expressividade, contudo usada quando não se tem preocupação com a garantia computacional nas quais não é possível realizar inferências e é utilizada quando se tem maior dinamismo na aplicação.

Portanto, de acordo com W3C, (204e) e E.MARTINS & C.SALDIAS, (206) o OWL Lite e o OWL DL são segundo W3C, (204e), tem mais expressividade como

sub-linguagens para serem usadas deixando bem claro que isso vai de acordo com o que o usuário necessita.

É neste contexto que, segundo E.MARTINS & C.SALDIAS, surge uma ferramenta para automatizar a criação do OWL chamada Protégé, que por sua vez possibilita a criação, de maneira mais eficaz e rápida, dos OWL para redes Semânticas que pode ser visto na seção 3.1.

2.6 SISTEMA ACADÊMICO ADX

O sistema de administração acadêmico ADX, foi desenvolvido pelo Instituto Doctum de Educação e Tecnologia, para gerenciar aspectos de uma instituição de ensino superior. O gerenciamento do sistema possibilita, aos usuários, funções que vão desde a inclusão de um novo curso até o requerimento de documentos oficiais pelos alunos. (ADX, 211).

De acordo com ADX, (211) ele gerencia o fluxo de informação da diretoria, secretaria, dos professores, dos alunos, das coordenações, da biblioteca e da tesouraria. Suas principais características e vantagens são:

- Tecnologia baseada na Web, dando liberdade ao usuário de trabalhar em qualquer lugar, a qualquer hora, pois não se limita apenas ao ambiente institucional.
- Automatização de tarefas, como inscrições de candidatos no vestibular através da Web, geração de documentos escolares de acordo com dados lançados no sistema, matrícula de alunos com dependência ou regular e disponibilização de nota para alunos.
- Liberdade dos professores quanto ao lançamento de atividades avaliativas (trabalhos teóricos, testes ou atividades em classe).

- Geração automática do diário escolar, de acordo com calendário escolar e grade horária curricular.
- Geração de boletins de alunos com frequência e notas no fim do semestre letivo.
- Segurança nas informações, que são definidas pelo próprio sistema no gerenciamento de permissões para usuários e pelo banco de dados que libera a execução de acordo com o nível de acesso de cada usuário, impedindo o mesmo a outros dados.
- Módulo de tesouraria confiável, flexibilizando o controle dos boletos, a fim de facilitar o acompanhamento dos dados financeiros do aluno.
- Informações lançadas pelos usuários que são disponibilizadas imediatamente aos demais, levando em conta o perfil de cada usuário no sistema.

A linguagem de programação utilizada no sistema ADX é o PHP, que por sua vez é uma linguagem livre (código aberto), interpretada e atende muito bem, gerenciando o conteúdo do sistema de forma dinâmica e eficaz, tornando as funcionalidades do sistema ágeis e eficientes.

O banco de dados para armazenamento das informações do Sistema ADX, é o Mysql que é um sistema de gerenciamento de banco de dados SQL (Structured Query Language), que por sua vez oferece ao sistema excelente desempenho, estabilidade e portabilidade, bem como uma ótima integração a linguagem do sistema o PHP. O banco de dados Mysql garante a integridade dos dados e informações do sistema exigindo pouco recurso do hardware.

3 FERRAMENTAS

Este Capítulo é apresentado ferramentas que auxiliem no desenvolvimento de uma aplicação Semântica de acordo como fora descrito no capítulo anterior.

Neste capítulo é apresentado a Ferramenta Protégé, bem como a criação de ontologias. Pode-se verificar também sobre a API RAP (RDF API), API do PHP aplicações Semânticas capacitando integração com o RDF ou OWL e também o SPARQL.

3.1 Protégé

Para representar e indexar os documentos tem se como base ferramentas e tecnologias o uso da ferramenta open-source (Código aberto) Protégé que foi desenvolvido pelo departamento de informática médica da universidade de Stanford, segundo (Camada & Mendonça, 207). O Protégé foi desenvolvido em Java, sua linguagem prima, ele permite construir ontologias de um determinado domínio possibilitando a personalização de formulários de entrada dos dados, inserindo e editando os dados.

Além de possibilitar a inserção e edição dos dados, a criação de bases de conhecimento guiadas por uma ontologia, disponibiliza uma interface gráfica provê o acesso a cinco áreas de visualização (VIEWS ou Visões) que funcionam como módulos de navegação e edição das classes, atributos, formulários, instâncias e pesquisas na base de conhecimento, propiciando a entrada de dados e a recuperação das informações segundo Camada & Mendonça, (207) e C.Saldias & E.Martins, (206).

A Ferramenta Protégé foi escolhida inicialmente porque contém uma melhor documentação para estudo e de pesquisa, no qual se baseou esse trabalho. Abaixo conheceremos as cinco áreas de visualizações que o Protégé oferece utilizando a versão 3.x pois é uma versão mais estável atualmente de acordo com Protégé, (211). De acordo com Protégé (211), após instalado o sistema, para se iniciar um projeto basta criar um novo projeto ou abrir um já existente conforme apresentado a seguir na Figura 11:

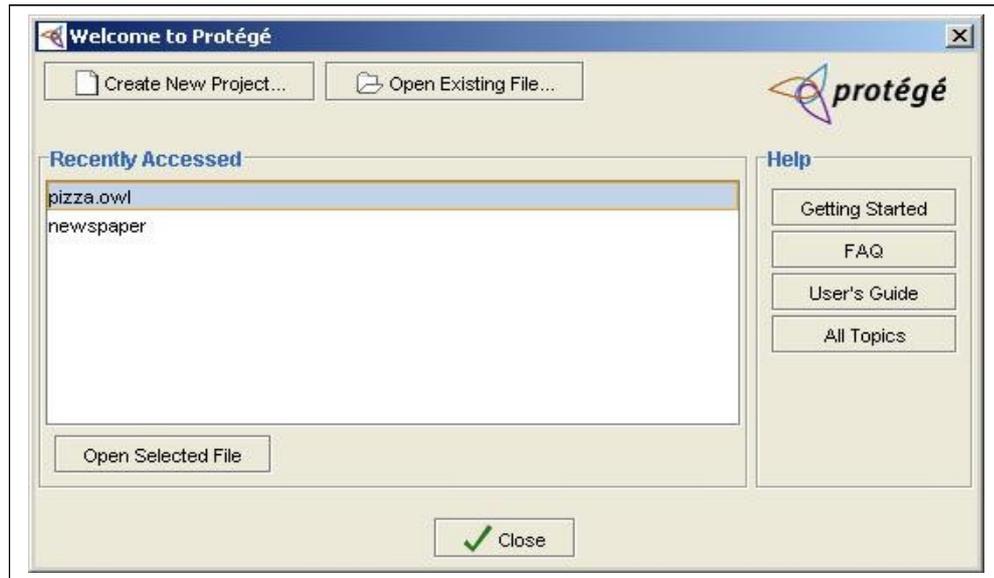


Figura 11: Novo projeto no Protégé - Fonte PROTÉGÊ, (211).

Como mostrado na Figura 11, é possível se criar um novo projeto clicando em “Create New Project”(Criar Novo projeto), ou um projeto que já esteja criado em outro diretório, basta ir em “Open Existing File”(Abrir Arquivo existente), no caso de projetos que foram criados recentemente segundo (Protégé ,211), fica disponíveis em “Recently Accessed” e guias de ajuda são disponibilizados na tela ao lado em “Help” (Ajuda).

Ao se optar pela criação de um novo modelo OWL é apresentado na Figura 12 de acordo com Protégé, (211):

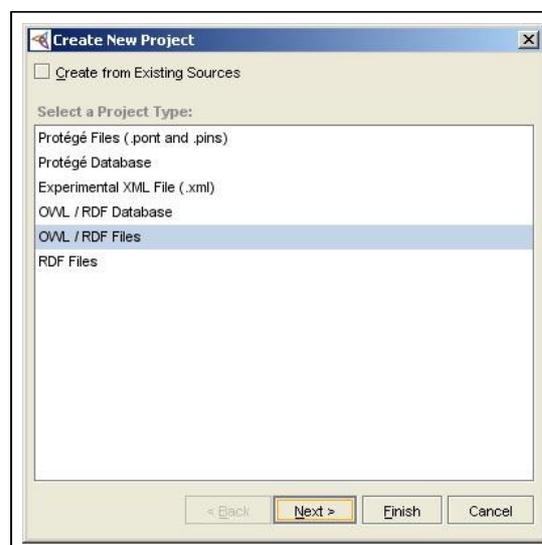


Figura 12: Escolhendo Modelo Protégé – Fonte Protégé, (211).

Nesta tela, como já havia sido explicado em seções anteriores, o Protégé dispõe as opções às quais pode-se então criar um modelo, segundo (Protégé ,211) baseado no próprio Protégé com a extensão .pont e .pins, o modelo Protégé data bases que é usada para casos de criação de ontologias diretamente no banco de dados(mais usado por desenvolvedores do Protégé) , o modelo para criação experimental do XML, o modelo para criação do OWL / RDF Data-base que já faz a integração com um banco de dados, o modelo OWL / RDF Files que cria apenas um modelo OWL / RDF e por último o RDF File que cria um modelo RDF.

Para demonstrar o autor escolhe OWL/RDF Files como modelo. Logo após, é solicitada a criação de uma URI, única do modelo criado como mostrado na Figura 13.



Figura 13: Criando URI no Protégé - Fonte Protégé, (211).

Após criada a URI, única do projeto, o autor demonstra como se define e escolhe a sub-linguagem do OWL. Abaixo a escolhida pelo autor é a OWL DL de acordo com a Figura 14.



Figura 14: Sub-Linguagem OWL - Fonte (Protégé, 211).

Após, é feita a escolha por fim é feito uma escolha entre a criação de um modelo lógico que garante mais expressividade nas relações e um modelo baseado em propriedades que são mais simples para entendimento, porém, não muito expressivo, de acordo com a Figura 15.



Figura 15: Expressividade Protégé - Fonte (Protégé, 211).

De acordo com o autor Protégé, (211), com estes passos são criadas ou carregadas as ontologias no ambiente do Protégé já definindo as características do projeto, bem como tipo do projeto, tipos de expressividade e URI a ser utilizada. Veremos agora como trabalhar com o projeto no ambiente. De acordo com (SACHS, E.206), na Figura 16 o ambiente do Protégé.

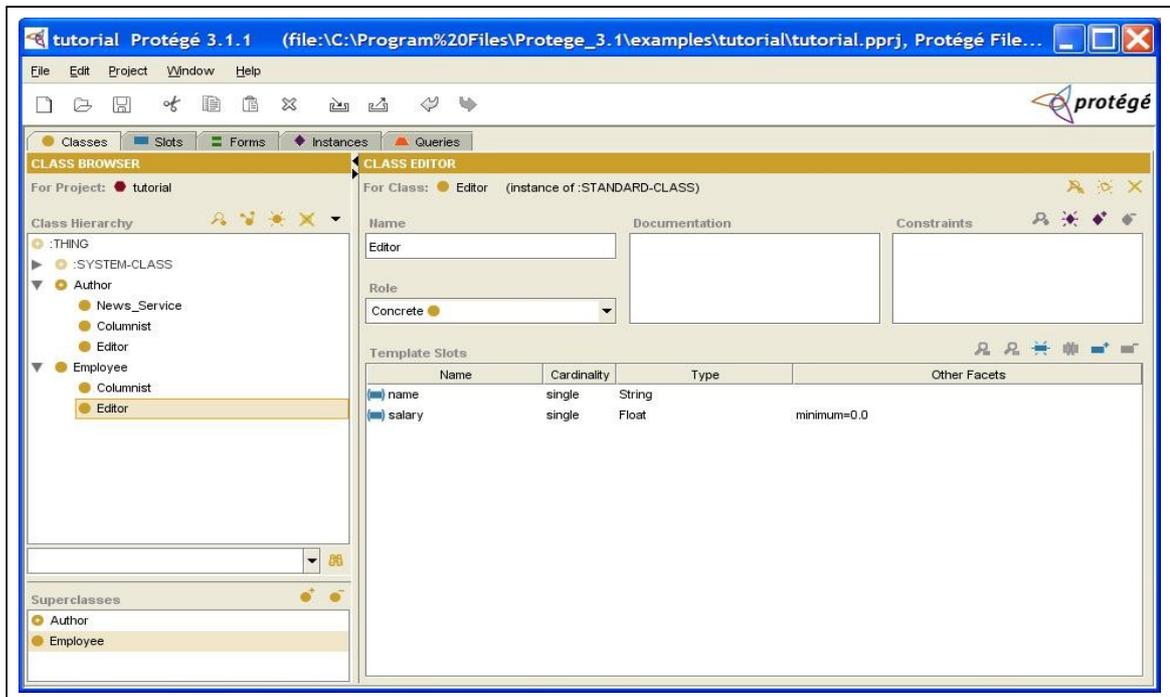


Figura 16: Ambiente Protégé Classes – Fonte SACHS, E. (206).

Na Figura 16, pode-se ver, segundo (SACHS, E.206), ao lado esquerdo, a hierarquia das classes criadas “Author” e “Employee” em CLASS BROWSER e em CLASS EDITOR as propriedades da Classe selecionada. Nesta primeira aba, pode-se visualizar abaixo em “Superclasses” as classes Mãe, ou seja a classe que origina subclasses.

Semelhante a primeira aba, a segunda aba “slots” ou “propriedades” contêm à direita todo o que estiver relacionado na ao slot selecionado. Nesta aba são criados os slots ou propriedades que serão utilizadas segundo Sachs, E. (206), devem ser definidas todas as propriedades que serão utilizadas.

Ainda segundo (SACHS, E.206), em “FORMS” são definidos os valores para as propriedades (SLOTS) criadas na aba anterior. Estes valores se transformaram

na aba “Instancies” como atributos das classes que por sua vez pode ser modificadas posteriormente.

E por fim, na Aba “QUERY”, é possível se realizar buscas simples ou mais elaboradas de acordo com o que o usuário modificar, segundo (SACHS, E.206). Contudo, não tem nenhuma utilizada para este trabalho, pois as buscas a serem feitas serão feitas em banco de dados após a inserção desta ontologia no mesmo.

Utilizando o Protégé é possível criar ontologias OWL mais elaboradas e organizadas, a fim de usar a expressividade para buscas de informações. Os processos envolvem todo o conhecimento teórico para demonstração de uma aplicação que será mostrada no capítulo de metodologia como foi realizada e no capítulo Resultados sua execução.

3.2 MYSQL

Nesta seção, não será aprofundado a pesquisa sobre banco de dados MYSQL. O MYSQL é um SGBD (sistema de gerenciamento de banco de dados) SQL (Structured Query Language ou Linguagem de Consulta Estruturada).

De acordo com MYSQL (211), o mysql tornou-se o sistema gerenciador de banco de dados open-source (código livre) mais popular do mundo, capacitando-o com desempenho e alta confiabilidade de uso. Executado em mais de 20 plataformas. O MYSQL disponibiliza acesso a manuais e à documentação em seu site oficial, como visto em MYSQL, (211b).

3.3 API RAP (RDF API)

Esta seção aborda sobre a API do PHP, denominada RAP, que por sua vez possibilita a integração de todas as ferramentas e linguagens descritas neste referencial teórico. Segundo C. Nepote, (2008) existem outras ferramentas para se trabalhar com Web Semântica e RDF em outras linguagens como “JavascriptRDF” para Javascript, “Wilbur” para Lisp, “RDFStore” para Perl, “SWIPT” para Python, “Ruby API Redland” para Ruby e outras mais para o PHP como “Domit” e “Saxy”, “ARC” e “SimpleRdfParser”.

A partir do teste feito, de acordo com W3C, (2002) o RAP é a segunda melhor biblioteca para manipulação de RDF com qualidade, em outras palavras com 98% de aprovação, ficando logo atrás da API JENA com 100% de aprovação. A API JENA, nativa da linguagem java que, por sua vez, foi a pioneira em manipulação de RDFs, segundo o autor. O fator que leva este trabalho ao foco em RAP se dá pelo fato do sistema acadêmico, ao qual esse trabalho pretende apoiar, é todo ele na linguagem PHP.

De acordo com C. Bizer; D. Westphal; R. Oldakowski, (2002), o RAP (RDF API) é ideal para programadores da linguagem PHP (Personal Home Page – Linguagem interpretada livre), ferramenta da Web Semântica que oferece recursos para análise; manipulação, armazenamento, consulta e serialização de RDF gráficos, Iniciado como um projeto open-source pela Universidade Freire de Berlin em 2002, e ampliado até então.

De acordo com M. K. Passos (2010), o RAP é utilizado para manipulação de modelos RDF possuindo interpretadores integrados para formatos como RDF/XML, N-triple (função para triplas no RDF) e pesquisas com SPARQL, além de armazenamento de fontes de pesquisa tanto em memória quanto em banco de dados. Para usar banco de dados segundo M. K. Passos (2010), o usuário deve carregar o modelo para um banco de dados relacional qualquer, criando, então, o modelo que passará a ser representado por tabelas no banco.

Ainda segundo M. K. Passos (210), a API, inicialmente foi lançada em 202 com poucos recursos e na medida dos anos sofrendo evoluções uma delas fora em 203 quando a API recebeu o suporte para RDQL (Linguagem de consulta RDF) e em 206 suporte a SPARQL.

No Website da API há diversos tutoriais que contem toda a documentação da API.

3.3.1 API RAP COM RDF E RDFS

Segundo C. Bizer; D. Westphal;R. OLDAKOWSKI, (202), em RAP os RDFs são representados da mesma idéia original do RDF onde são criadas instâncias da classe, que por sua vez, são os modelos, e estes modelos possuem declarações visto que cada declaração possui três nós, sendo eles o predicado, sujeito e objeto contanto que cada nó representa, então, um recurso identificado por uma URI.

“RAP inclui um pensador à frente e um para trás RDFS encadeamento e suporta alguns OWL construções. Os pensadores são implementados como extensões do in-memory armazenamento tripla, adicionando a capacidade de inferir adicionais (implícito) declarações. Assim, o processo de inferência é totalmente oculto para as interfaces de trabalhar com o in-memory armazenamento e permite acesso total dentro de todas as camadas de abstração. Isto também permite a emissão de RDQL consultas a um gráfico RDF contendo declarações inferidas, permitindo consultas mais de vinculações.”C. (BIZER; D. WESTPHAL;R. OLDAKOWSKI, 202).

De acordo com os autores C. Nepote, (208), W3C, (202), C. Bizer; D. Westphal;R. Oldakowski (202) e M. K. Passos, (210) pode-se conhecer a API RAP para manipulação do RDF e criação do modelo seja ele executado do em Memória ou em um banco de dados para, assim, realizar a inferência através de buscas utilizando o RAP com a linguagem de Consulta RDQL.

3.3.2 API RAP E ARMAZENAMENTO PERSISTENTE

De acordo com C. Bizer; D. Westphal; R. OLDAKOWSKI, (202), o núcleo do RAP no banco de dados é construído por duas classes: DbStore e DbModel. O DbStore é utilizado para definir a conexão com o banco de dados relacionais, bem como criar, armazenar, listar e recuperar os modelos RDF. O DbModel se encarrega de fornecer métodos para manipular cada modelo criado. É utilizada a biblioteca de abstração de banco de dados ADOdb(Active Data Objects data bases) para PHP que por sua vez, suporta uma grande variedade de bancos de dados relacionais diferentes como MySQL, Oracle, DB2, MS SQL-Server, MS Access.

Segundo C. Bizer; D. Westphal; R. Oldakowski, (202), a fim de otimização do banco de dados relacionais o armazenamento do RDF no RAP não é normalizado e sim desnormalizado e quando comparado em um teste de benchmark com um modelo normalizado tendo “rdf_statements”(tabela onde ficam armazenadas as triplas de dados) como tabela principal apontando apenas para outras duas tabelas “rdf_resources” e “rdf_literals” mostrou sendo a desnormalização de 2-3 vezes mais rápido.

A estrutura que será criada pelo RAP, de acordo com RAPAPI, (211) em um banco de dados é da seguinte forma:

- DataSets: relaciona cada modelo ao seu respectivo nome. Isso evita problema quando se tem mais de um modelo.
- DataSetModel: relaciona o URI de de um gráfico com seu modelo. Assim cada modelo tem seu gráfico caso ele exista. Lembrando que não é obrigatório ter um gráfico.
- Models: guarda os modelos e nome dos arquivos RDF ou Owl criados para relacionamento nas demais tabelas como dataSets e DataModel e as seguintes sendo Namespaces e Statements.
- Namespaces: Guarda todas as URIs dos namespaces, bibliotecas e caso exista URIs de ontologias importadas como por exemplo o Dublin Core.

- Statement: é nesta tabela onde os dados desnormalizados são guardados em forma de triplas, sendo cada uma com três nós portando três colunas sujeito, predicado e objeto, e claro outras referentes ao modelo como o ID do modelo por exemplo.

Assim é a estrutura das tabelas criadas pela API RAP, a fim de capacitar a integração com os demais padrões e assim possibilitar o uso correto dos mesmos. Sobre o uso destes padrões e como foram utilizados será visto na próxima seção.

4 METODOLOGIA

Buscando verificar a viabilidade de se usar a Web Semântica num sistema de gerenciamento como o ADX, criou-se este projeto que tem como objetivo, estudo para recuperação de informações acadêmicas de alunos de forma eficiente utilizando padrões da Web Semântica.

As técnicas usadas têm como base no estudo referencial dos padrões da Web Semântica possibilitando que o sistema interprete claramente o que usuário necessita retornando para o mesmo uma resposta coerente em forma de serviços e informações, tanto quanto a execução de tarefas.

Para desenvolvimento desse projeto, primeiramente foram feitas pesquisas bibliográficas descritas no capítulo “Referencial Teórico”, onde pode-se concluir que se é possível criar uma aplicação para apresentar, através de um sistema simples de busca, informações sobre alunos, como por exemplo:

- Alunos com idade maior ou igual a X.
- Alunos com idade maior que X.
- Alunos com idade menor ou igual a X.
- Alunos com idade menor que X.
- Liste todos os alunos do curso X.
- Mostre todos os alunos do curso X.

Onde “X” é um valor que pode ser alterado de acordo com a preferência do usuário.

O objetivo desta aplicação, é mostrar os resultados destas questões de forma objetiva, listando os alunos que se enquadram em cada uma das listagens, de forma que o computador saiba onde estas informações estão, para exibi-las.

Para o desenvolvimento dessa aplicação foi utilizado o Protégé na elaboração da ontologia, pois, ele agiliza e simplifica o desenvolvimento, além de ter uma ótima documentação. Foi utilizada a linguagem PHP, como linguagem de programação,

pois, além do sistema Adx ser desenvolvido nesta linguagem, é uma linguagem rápida, dinâmica e atende as necessidades da aplicação desse projeto, bem como a sua integração com a aplicação com sua API RAP que é considerada uma das melhores para manipulação de informações com RDF. A API RAP foi criada para interagir com o PHP e trabalhar em conjunto com o RDF, OWL, linguagem de consulta SPARQL.

Com o modelo ontológico no banco de dados foi possível inserir novos dados de alunos, e por sua vez, realizar consultas na linguagem SPARQL, para recuperação dos dados.

Nesta seção será visto como foi realizada a criação de uma ontologia utilizando o Protégé, bem como alguns passos importantes para a criação. Será visto como sobre a criação do banco de dados Mysql para manipular as informações bem como a recuperação de informações através da linguagem de consulta SPARQL em conjunto com a API RAP.

4.1 CRIANDO UMA ONTOLOGIA OWL USANDO Protégé

Na seção 3.1, foram definidos os passos iniciais para criação da ontologia, a fim de apresentar a ferramenta, não será mostrado o passo a passo da realização de cada tarefa, pois não é o foco deste trabalho.

4.1.1 Definindo URI da Ontologia

Ao se criar um projeto novo no Protégé, é solicitado a informação sobre qual seria a IRI do projeto na Figura 13. A ela foi dado a seguinte URI: `http://www.owl-ontologies.com/ontologiaADX.owl`.

Na Figura 17, pode-se verificar a definição da URI:

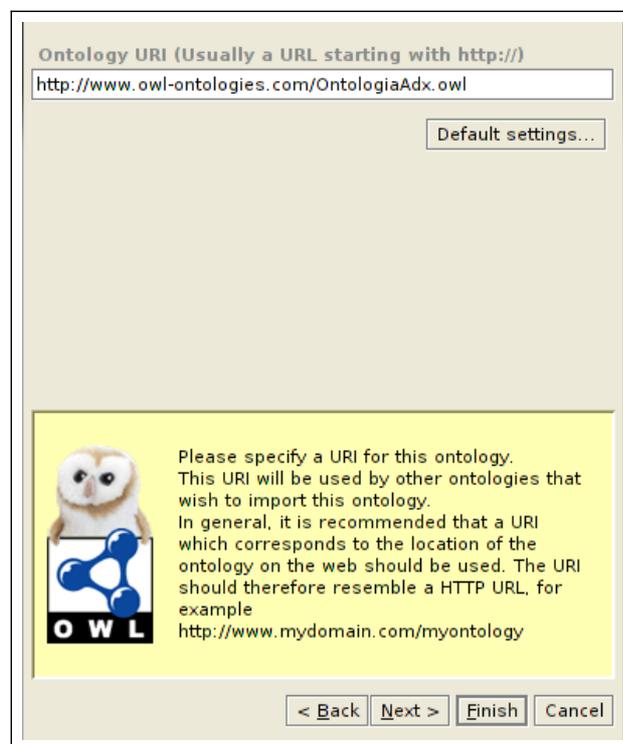


Figura 17: Definindo URI no Protégé.

Após a Definição da URI é solicitado da sublinguagem do OWL de acordo como definido na seção 3.1. Neste projeto foi usada a sublinguagem OWL DL, devido seu nível de expressividade comparado com os demais apresentando melhor os resultados, conforme pode-se ver na Figura 18:

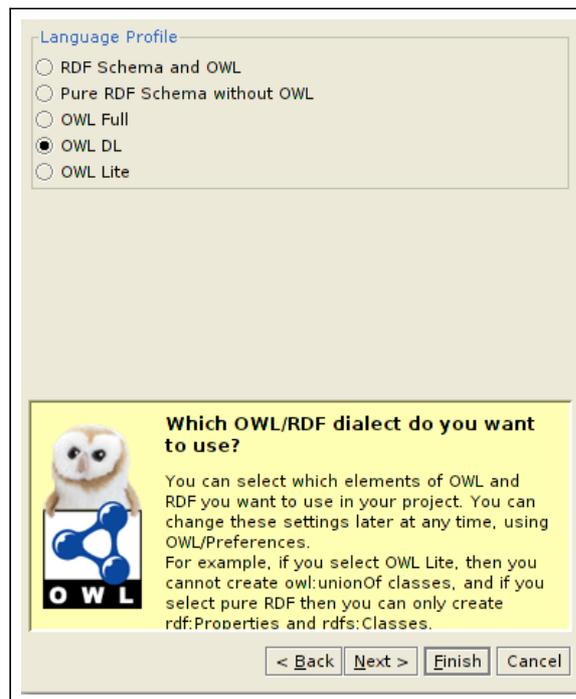


Figura 18: Definição da Sublinguagem do OWL.

Após definido a sublinguagem, o Protégé precisa ser informado sobre qual o nível de expressividade, conforme descrito na seção 3.1 e como mostra a Figura 19:

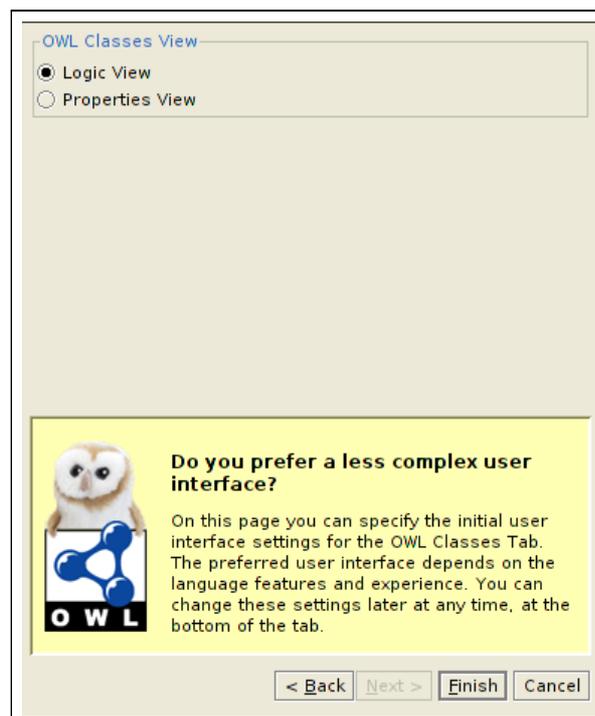


Figura 19: Definindo Nível de expressividade utilizando modelo lógico no Protégé.

Feito estas etapas iniciais, que definem o projeto, na próxima seção será apresentada a configuração do vocabulário, para que não haja problema de conflitos de namespaces na ontologia.

4.1.2 Importando vocabulário Dublin Core em Ontologia

A importação do vocabulário segue de acordo com a documentação do Protégé, (211b).

O Dublin Core, disponibiliza a seguinte URI para importação:

"<http://purl.org/dc/elements/1.1/>" "<http://purl.org/dc/elements/1.1/>" e

"<http://purl.org/dc/terms/>".

Na Figura 20 é mostrada a parte inferior da Aba "Metadata" onde, percebe-se todos os Namespaces incluídos por padrão na ontologia e o Namespaces do vocabulário Dublin Core, que também foi acrescentado, sendo ele referenciado pela sigla "dc" e "dcterms":

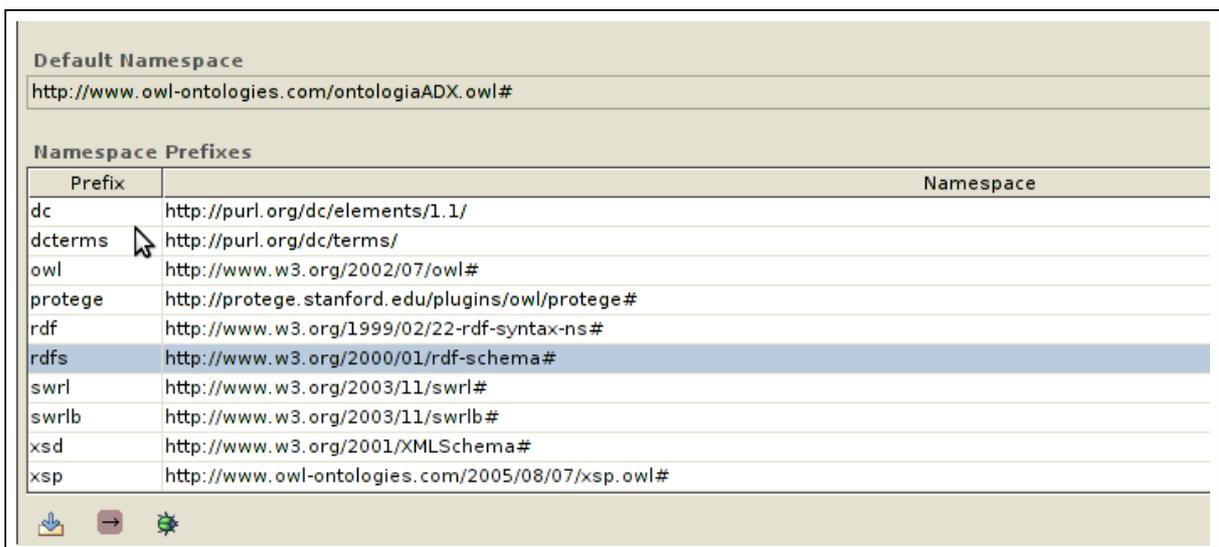


Figura 20: Importação do Padrão(Namespace) Dublin Core.

Conforme pode ser visto na Figura 17, a importação do padrão Dublin Core foi realizada e o mesmo se juntou aos Namespaces que já são incluídos por padrão pelo Protégé, a fim de garantir a sintaxe correta do XML e OWL.

4.1.3 Criando Classes

Para este projeto com o intuito demonstrativo foi criada uma classe “Alunos” contendo como propriedades Nome, Sobrenome, Idade, Curso e Nota e outra classe “Curso” contendo como propriedade o nome do curso. É possível visualizar, nas Figuras 21 e 22.

Na Figura 21, pode-se ver que, em “SUBCLASS EXPLORER”, contêm as classes criadas e em “CLASS EDITOR for Alunos”, contêm as propriedades da classe selecionada:

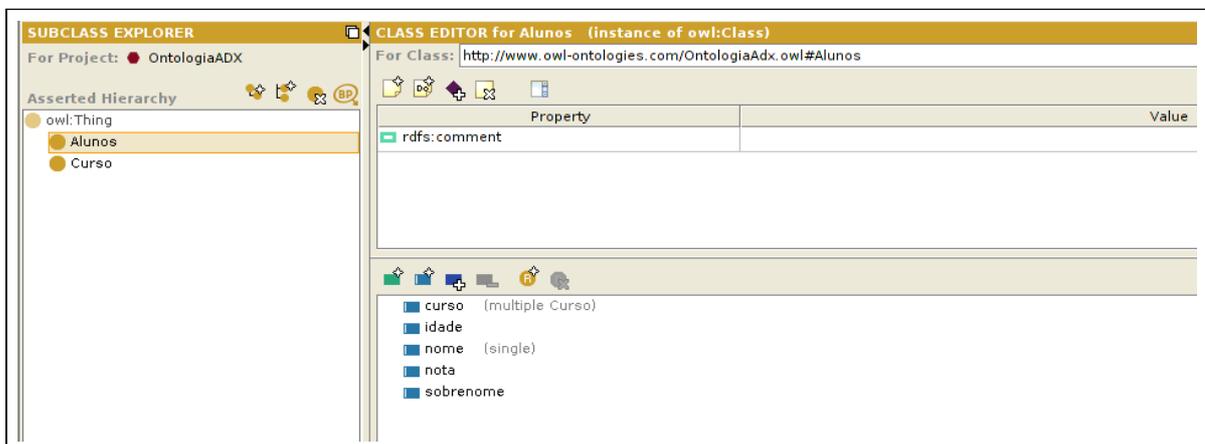


Figura 21: Definindo propriedades da classe Aluno.

O mesmo ocorre ao selecionar a Classe “curso”, como pode-se ver na Figura 22.

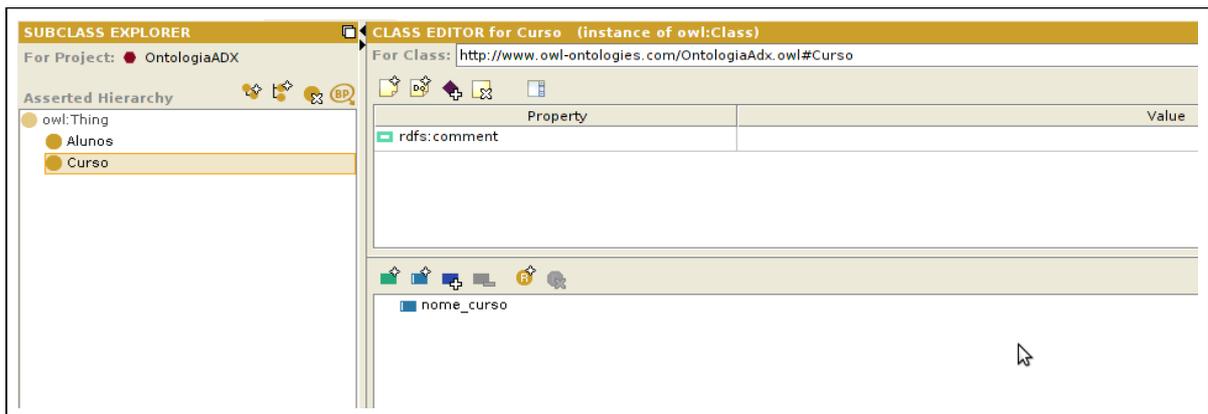


Figura 22: Definindo propriedades da classe Curso.

Para finalizar a criação das propriedades, foram definidos tipos de dados para cada propriedade na aba “propriedades”, conforme documentação do Protégé.

Logo, foi possível virtualizar na Aba instances (instâncias), uma possível definição de valores para as propriedades criadas, conforme pode-se ver nas Figuras 23 e 24.

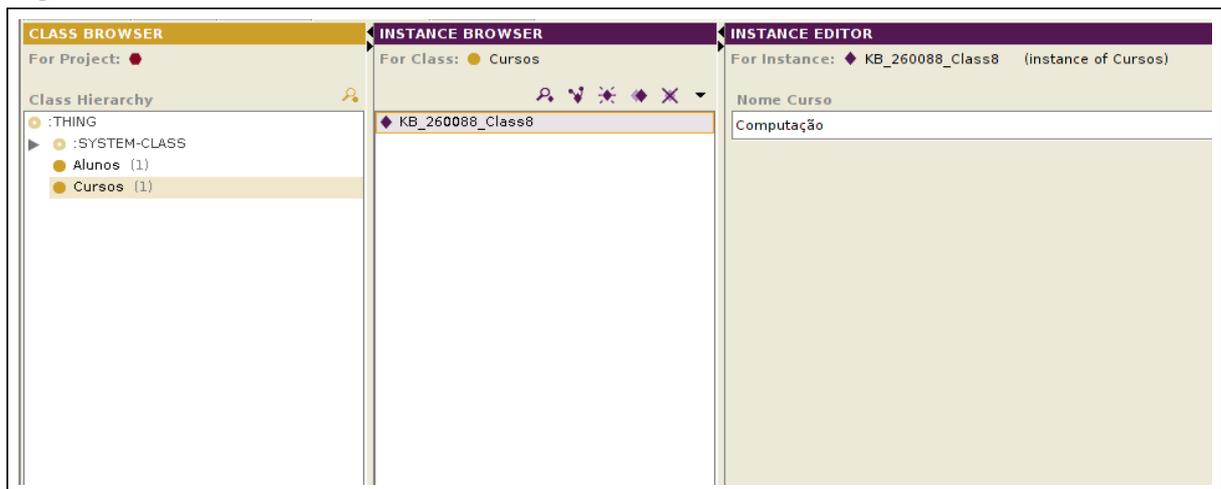


Figura 23: Inserindo Curso no Protégé.

A Figura 23 mostra a inserção de um curso. No caso o curso de Computação:

Foi, portanto, criada uma instância da classe com o nome “KB_26088_Class8”, com o nome do curso.

Com o curso cadastrado, pode-se, então, cadastrar um aluno no mesmo, conforme mostrado na Figura 24.

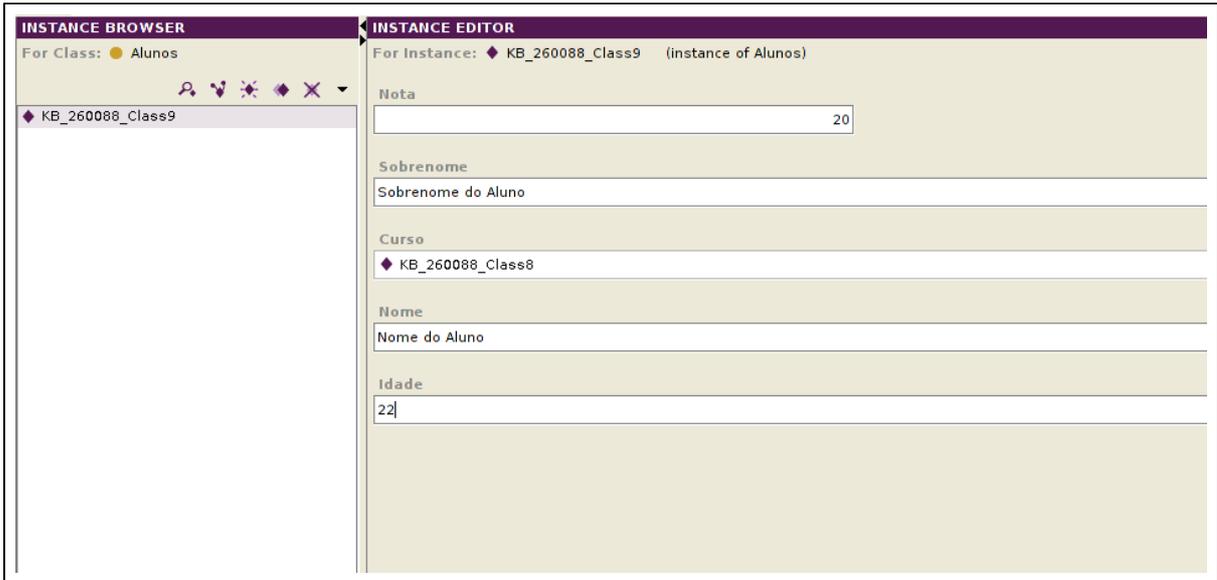


Figura 24: Inserindo Aluno no Protégé.

Na Figura 24, o aluno é cadastrado, e cria-se um instância para este referenciada com o nome “KB_26088_Class9”, que contém os dados do aluno e seu curso.

Ainda Figura 24, pode-se ver que é possível cadastrar os dados como nota, sobrenome, curso, nome e idade de um aluno. É importante ressaltar que os dados do curso do aluno são cadastrados na classe “Cursos” e, de acordo como mostrado na Figura 23, o curso computação da instância “KB_26088_Class8”, foi adicionado para o aluno, ou seja, este aluno pertence ao curso de computação.

O Protégé disponibiliza uma última aba chamada “Query”, para se fazer consultas Internas na ontologia. Esta consulta pode ser vista na Figura 25.



Figura 25: Busca Interna na Ontologia Protégé.

O objetivo da busca era, encontrar quais alunos (representado pela classe “Alunos”) eram do curso (representado pela propriedade “curso”) de computação (representado pela instância criada no ato do cadastro do curso, conforme Figura 23). Por sua vez, ele retornou corretamente, informando que só existe uma instância

de cadastro de aluno que tenha o seu curso como “Computação”, e que a instância como pode ser visto na Figura 25, é a “KB_26088_Class9”, que, neste caso, é o aluno cadastrado anteriormente na Figura 24.

Ao finalizar e salvar esta ontologia é gerado um arquivo no formato OWL, contendo, em seu interior, a linguagem XML, e, com isso, pode-se abrir esta ontologia no navegador conforme mostrado na Figura 26.

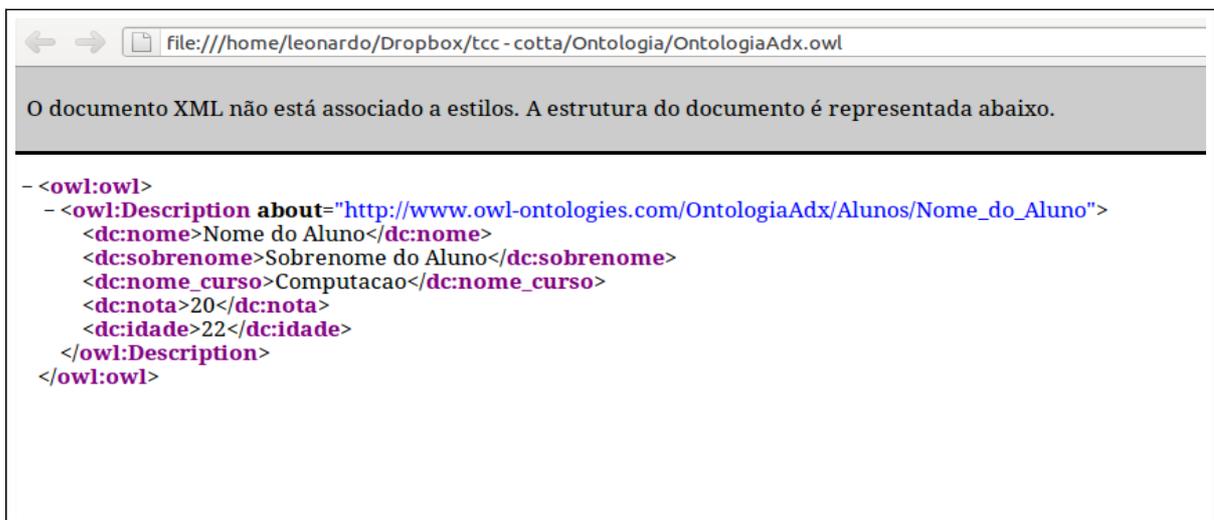


Figura : Exibindo Ontologia Criada no navegador

Finalizando a criação de um modelo ontológico, simples e de fácil interpretação, onde mostrado na Figura 26, a estrutura mostra a descrição da URI de um determinado Aluno e internamente seus módulos como “Nome”, “Sobrenome”, “nota_curso” e “idade”. Deve ser verificado que em cada módulo interno há um “dc:” que vem do vocabulário que foi incluído, e, por sua vez é responsável por não ocorrer ambigüidade na estruturada, caso esta ontologia seja usada ou importada para outras ontologias e vice-versa.

4.2 CRIANDO BASE MYSQL PARA GUARDAR A ONTOLOGIA

Foi utilizado o banco de dados MYSQL, para guardar as informações a respeito dos alunos, contudo poderiam ser outros bancos de dados visto que há suporte para todos. O MYSQL foi utilizado com o intuito de manter a compatibilidade com o sistema acadêmico Adx que utiliza o mesmo como banco de dados.

Foi criada uma base de dados chamada “BaseOntológicaAdx” onde não foram criadas nenhuma tabela. Tarefa esta que é responsabilidade da API RAP.

Na seção 4.3, é explicado como pode ser feita a criação das tabelas no banco de dados.

4.3 INSERINDO ONTOLOGIA NO MYSQL USANDO API RAP

Esta seção apresenta como foi feita para inclusão do Modelo Ontológico, criado na seção 4.1, para dentro do banco de dados usando a API RAP. Como foi descrito na seção 3.3.2, a API RAP possui o método de armazenamento persistente que visa inserção de determinada ontologia em banco de dados relacionais. Este procedimento teve como base a documentação da RAPAPI, (211). Segundo esta documentação foi desenvolvido um script de inicialização em PHP, cujo objetivo é fazer a conexão com o banco de dados MYSQL, criar um modelo com base na ontologia.

No Script PHP criado utilizando a API RAP para elaborar as tabelas no banco de dados, foram escritas as seguintes linhas na Figura 27, código para fazer tal tarefa de acordo com documentação da API.

```

1 <?php
2 $rdf_banco = ModelFactory::getDbStore('MySQL','localhost','People','root','123');
3 $rdf_banco->createTables('MySQL');
4 $base = "OntologiaAdx.rdf";
5 $memModel = ModelFactory::getDefaultModel();
6 $memModel->load($base);
7 $rdf_banco->putModel($memModel);
8 $modelURI = "OntologiaAdx.rdf";

```

Figura 26: Criando tabelas no banco de dados usando API RAP, na aplicação.

Esta parte do código é responsável por:

- Conectar com banco de dados através do “getDbStore”, a criação das tabelas no banco de dados é feita através do “createTables”.
- Definir a base ontológica representada através da variável “\$base”(a ontologia é a base).
- Criar de um modelo padrão na memória é feita com o “getDefaultModel”.
- Atribuir ao modelo na memória a “\$base” que é a ontologia ou modelo ontológico e por fim leva o modelo para o banco de dados com o “putModel”.
- Ao fim apenas a definição da URI de com nome da base que é “modelURI”.

Dessa forma os dados que antes estavam no arquivo OntologiaAdx.owl, agora encontram-se no do banco de dados, criado na seção 4.2 “BaseOntológicaAdx”.

4.4 LENDO MODELO NO BANCO DE DADOS UTILIZANDO API RAP

O RAP possui em suas funções a função “writeAsHtmlTable”, que exibe já no formato HTML, uma tabela com dados existentes no banco de dados de acordo com a documentação da API, foi criado um script que faz a leitura utilizando esta função.

Neste Script, também é feito a conexão com o banco de dados e criado o modelo na memória, processo parecido como o descrito na seção anterior.

Após a execução da função “writeAsHtmlTable”, é encerrada a conexão com o banco.

Desta forma é possível, exibir todos os dados apenas, será visto na próxima seção como realizar consultas no banco de dados.

4.5 CONSULTAS SPARQL UTILIZANDO API RAP

O SPARQL é uma linguagem de consulta RDQL (RDF Data Query Language) e por sua vez realiza consultas em arquivos RQDL.

Nesta seção será visto como utilizar a API RAP com o RDQL a fim de realizar consultas no modelo ontológico criado na seção 4.3.

A seguir é descrito a lógica para reconhecimento da linguagem natural.

4.5.1 Lógica para reconhecer a linguagem natural

O reconhecimento de linguagem natural está fora do escopo deste trabalho, pois este visa introduzir os conceitos da Web Semântica, bem como mostrar o uso dos padrões semânticos para recuperação Semântica de informações. O tratamento da linguagem natural será proposto em trabalhos futuros.

A linguagem natural é a linguagem da forma com que é escrita, a idéia é que ela possa ser reconhecida pelo computador, ou seja, o computador interpreta o significado do que foi digitado.

Portanto, foram definidas lógicas de programação com base nas palavras digitadas pelo usuário, de forma que de acordo com a seqüência das palavras se possa formar uma pergunta ou solicitação de listagem.

Caso o usuário solicite a seguinte listagem “Liste alunos do curso de computação”, na lógica definida, as palavras desta solicitação são guardadas em uma posição de um vetor onde esta pergunta só será feita se o usuário digitar na seguinte ordem: “Liste”+”alunos”+”do”+”curso”+”de”+”computação”. Dessa foi formatado o reconhecimento das questões.

4.5.2 Consultas usando API RAP E RDQL

De acordo com documentação da RAPAPI, (211) e no na documentação SPARQL em W3C, (208), foi elaborada uma consulta que busque no banco de dados na tabela “statements” criada na seção 4.3 entre as triplas “sujeito”, “predicado” e “objeto”.

O objetivo desta consulta é buscar o “nome”, “sobrenome”, “idade”, “curso” e “nota” de acordo com os termos pesquisados definidos na lógica definida na seção 4.5.1. A consulta pode ser vista abaixo na Figura 28.

```

11 $rdql_query = '
12   SELECT *
13   FROM <OntologiaAdx.owl>
14   WHERE
15     (?x,<dc:nome>,?nome),
16     (?x,<dc:idade>,?idade),
17     (?x,<dc:sobrenome>,?sobrenome),
18     (?x,<dc:nome_curso>,?curso),
19     (?x,<dc:nota>,?nota)';
20 if($curso && !$idade)
21   $rdql_query .= ' AND ?'. $termo_pesquisa.' '.$operador.' '.$sepI.' '.$ntermo[$i].''.$sepF.'';
22 else
23   $rdql_query .= ' AND ?'. $termo_pesquisa.' '.$operador.' '.$ntermo[$i].''.$sepF.'';
24 $rdql_query .= ' USING dt for <http://foo.org#>, rdf for <http://www.w3.org/1999/02/22-rdf-syntax-ns#>';
25 $res = $dbModel->rdqlQuery($rdql_query);

```

Figura 27: Consulta RDQL criada na aplicação

Na primeira parte da consulta foi definindo o local de busca das informações sendo ele a ontologia “OntologiaAdx.owl”(linha 13), buscando por por “dc:nome” atribuindo a ele uma variável “?nome”(linha 15), “dc:idade” atribuindo a ela uma variável “?idade”(linha 16), “dc:sobrenome” atribuindo a ele a variável “?”

sobrenome”(linha 17),”dc:curso_aluno” atribuindo a este a variável “?curso”(linha 18) e por último “dc:nota” atribuindo a este a variável “?nota”(linha 19).

Desta forma são listados todos os dados disponíveis no banco de dados.

A segunda parte da consulta foi definida de acordo com os termos(palavras que definem a lógica, como fora definidos na seção 4.5.1), digitados pelo usuário, realizando as seguintes condições:

- Se na mesma frase tiver o termo “curso” e não conter o termo “idade”, será a consulta de forma apenas pegar o curso.
- Se não obedecer a essa condição entrará na próxima que aceita todas as demais condições.

É importante ressaltar que as variáveis “\$termo_pesquisa”, “\$operador”, “\$sepl”, “\$sepF” e “\$ntermo[\$i]” são definidos de acordo com lógica feita na seção 4.5.1, podendo sofrer modificações de valores de acordo com o que for solicitado pelo usuário, tornando assim a consulta uma consulta mais dinâmica.

Seguindo esta lógica, a aplicação pode reconhecer as solicitações de busca e listagem. Os resultados serão discutidos no Capítulo 5.

4.5.3 Inserindo dados fictícios de alunos

Para testar a aplicação foram inseridos dados fictícios de alunos com base no sistema ADX.

Cerca de 100 registros foram inseridas na aplicação a fim de apresentar a execução do mesmo e a listagem de informações. A execução da aplicação pode ser encontrada no Capítulo 5.

5 RESULTADOS

Utilizando padrões da Web Semântica, visando apresentar a execução de uma aplicação simples que busca informações de alunos, através do resultado pode-se verificar que é possível se obter informações utilizando padrões da Web Semântica com resultados mais precisos ao contrário de como é feito atualmente onde são listados dados providos por meio de TAGS ou seja palavras-chaves, que não buscam por informações e sim por dados e deixa aberta a possibilidade de retorno de resultados que não condizem com o solicitado.

Para aplicar esses padrões, como já dito foi desenvolvida uma aplicação simples de busca que pode ser verificada na próxima seção, onde será visto a execução da mesma.

5.1 RESULTADOS DA APLICAÇÃO DESENVOLVIDA

Primeiramente na Figura 29 é apresentada a tela inicial da aplicação, onde as listagens e pesquisas podem ser realizadas.

Carregar Banco de dados - Ler ontologia - Limpar Banco de dados

TCC - BUSCA SEMÂNTICA

Demonstração de busca de informações de alunos utilizando padrões da Web Semântica

Buscar

Figura 28: Tela da Aplicação desenvolvida para demonstração.

Na execução, uma listagem é solicitada por um usuário da seguinte forma:

- “Alunos com idade maior que 20”.

Abaixo, na Figura 30 a execução da solicitação feita, mostrando os resultados obtidos.

No.	?x	?nome	?idade	?sobrenome	?curso	?media
1.	Resource: http://foo.org/persons/030300011	Literal: NETO	Literal: 39	Literal: NETO	Literal: CIENCIAS CONTABEIS	Literal: 87
2.	Resource: http://foo.org/persons/030300139	Literal: ALENCAR	Literal: 37	Literal: ALENCAR	Literal: CIENCIAS CONTABEIS	Literal: 86
3.	Resource: http://foo.org/persons/040200100	Literal: SILVA	Literal: 29	Literal: SILVA	Literal: SERVIÇO SOCIAL	Literal: 88
4.	Resource: http://foo.org/persons/040300201	Literal: SÃ	Literal: 28	Literal: SÃ	Literal: CIENCIAS CONTABEIS	Literal: 96
5.	Resource: http://foo.org/persons/050300042	Literal: SILVA	Literal: 43	Literal: SILVA	Literal: CIENCIAS CONTABEIS	Literal: 56
6.	Resource: http://foo.org/persons/050300089	Literal: OLIVEIRA	Literal: 28	Literal: OLIVEIRA	Literal: CIENCIAS CONTABEIS	Literal: 25
7.	Resource: http://foo.org/persons/050400020	Literal: LIMA	Literal: 30	Literal: LIMA	Literal: DIREITO	Literal: 83
8.	Resource: http://foo.org/persons/050400031	Literal: RIBEIRO	Literal: 24	Literal: RIBEIRO	Literal: DIREITO	Literal: 72
9.	Resource: http://foo.org/persons/050400039	Literal: PACHECO	Literal: 37	Literal: PACHECO	Literal: DIREITO	Literal: 87
10.	Resource: http://foo.org/persons/050400081	Literal: MURTA	Literal: 25	Literal: MURTA	Literal: DIREITO	Literal: 87
11.	Resource: http://foo.org/persons/050400138	Literal: CAMINHAS	Literal: 25	Literal: CAMINHAS	Literal: DIREITO	Literal: 56
12.	Resource: http://foo.org/persons/050400163	Literal: COSTA	Literal: 29	Literal: COSTA	Literal: DIREITO	Literal: 75
13.	Resource: http://foo.org/persons/050400189	Literal: CRUZ	Literal: 27	Literal: CRUZ	Literal: DIREITO	Literal: 72
14.	Resource: http://foo.org/persons/050400215	Literal: MARCOS	Literal: 30	Literal: MARCOS	Literal: DIREITO	Literal: 92
15.	Resource: http://foo.org/persons/050400218	Literal: SANTOS	Literal: 40	Literal: SANTOS	Literal: DIREITO	Literal: 80
16.	Resource: http://foo.org/persons/060100102	Literal: MARTINS	Literal: 39	Literal: MARTINS	Literal: ENFERMAGEM	Literal: 75
17.	Resource: http://foo.org/persons/060300005	Literal: PARREIRA	Literal: 34	Literal: PARREIRA	Literal: CIENCIAS CONTABEIS	Literal: 82
18.	Resource: http://foo.org/persons/060400022	Literal:	Literal: 24	Literal:	Literal: DIREITO	Literal: 0
19.	Resource: http://foo.org/persons/060400034	Literal: PAIXÃFO	Literal: 26	Literal: PAIXÃFO	Literal: DIREITO	Literal: 70
20.	Resource: http://foo.org/persons/060400046	Literal: MORAES	Literal: 25	Literal: MORAES	Literal: DIREITO	Literal: 0

Figura 29: Apenas resultados de alunos com idade maior que 20 anos

Como se pode ver na Figura 30, apenas alunos com idade maior que 20 anos foram listados como resultados.

Outra demonstração dentre as que foram listadas como objetivo é caso o usuário solicite mostrar alunos de um determinado curso como “Mostre, alunos do curso de Psicologia” ou “Liste alunos do curso de Psicologia”, na Figura 30 pode-se visualizar outros resultados.

No.	?x	?nome	?idade	?sobrenome	?curso	?media
1.	Resource: http://foo.org/persons/070900004	Literal: SILVA	Literal: 27	Literal: SILVA	Literal: PSICOLOGIA	Literal: 91
2.	Resource: http://foo.org/persons/070900005	Literal: MIRANDA	Literal: 24	Literal: MIRANDA	Literal: PSICOLOGIA	Literal: 82
3.	Resource: http://foo.org/persons/070900006	Literal: JUNIOR	Literal: 56	Literal: JUNIOR	Literal: PSICOLOGIA	Literal: 93
4.	Resource: http://foo.org/persons/070900008	Literal: VIANA	Literal: 24	Literal: VIANA	Literal: PSICOLOGIA	Literal: 89
5.	Resource: http://foo.org/persons/070900009	Literal: CAMINHAS	Literal: 24	Literal: CAMINHAS	Literal: PSICOLOGIA	Literal: 0
6.	Resource: http://foo.org/persons/070900011	Literal: MARQUES	Literal: 22	Literal: MARQUES	Literal: PSICOLOGIA	Literal: 86
7.	Resource: http://foo.org/persons/070900013	Literal: SILVA	Literal: 33	Literal: SILVA	Literal: PSICOLOGIA	Literal: 89
8.	Resource: http://foo.org/persons/070900016	Literal: ALVES	Literal: 35	Literal: ALVES	Literal: PSICOLOGIA	Literal: 89
9.	Resource: http://foo.org/persons/070900017	Literal: OLIVEIRA	Literal: 45	Literal: OLIVEIRA	Literal: PSICOLOGIA	Literal: 85
10.	Resource: http://foo.org/persons/070900019	Literal: CARVALHO	Literal: 46	Literal: CARVALHO	Literal: PSICOLOGIA	Literal: 92
11.	Resource: http://foo.org/persons/070900020	Literal: TEIXEIRA	Literal: 48	Literal: TEIXEIRA	Literal: PSICOLOGIA	Literal: 89
12.	Resource: http://foo.org/persons/070900022	Literal: FERNANDES	Literal: 29	Literal: FERNANDES	Literal: PSICOLOGIA	Literal: 87
13.	Resource: http://foo.org/persons/070900023	Literal: BENICIO	Literal: 42	Literal: BENICIO	Literal: PSICOLOGIA	Literal: 86
14.	Resource: http://foo.org/persons/070900024	Literal: MARQUES	Literal: 41	Literal: MARQUES	Literal: PSICOLOGIA	Literal: 90
15.	Resource: http://foo.org/persons/070900025	Literal: PEREIRA	Literal: 37	Literal: PEREIRA	Literal: PSICOLOGIA	Literal: 90
16.	Resource: http://foo.org/persons/070900026	Literal: MOTA	Literal: 47	Literal: MOTA	Literal: PSICOLOGIA	Literal: 84
17.	Resource: http://foo.org/persons/070900027	Literal: SANTOS	Literal: 27	Literal: SANTOS	Literal: PSICOLOGIA	Literal: 81
18.	Resource: http://foo.org/persons/070900029	Literal: HIRLE	Literal: 22	Literal: HIRLE	Literal: PSICOLOGIA	Literal: 90
19.	Resource: http://foo.org/persons/070900031	Literal: AGUILAR	Literal: 37	Literal: AGUILAR	Literal: PSICOLOGIA	Literal: 86
20.	Resource: http://foo.org/persons/070900033	Literal: GUSMAO	Literal: 34	Literal: GUSMAO	Literal: PSICOLOGIA	Literal: 87

Figura 30: Apenas alunos do curso de psicologia

Conforme mostrado na Figura 30 e 31, dados como idade e curso foram exibidos corretamente, é importante ressaltar que solicitações de listagens abaixo também obtiveram resultado positivo:

- Alunos com idade maior ou igual que X.
- Alunos com idade maior que X.
- Alunos com idade menor ou igual que X.
- Alunos com idade menor que X.
- Liste todos os alunos do curso X.
- Mostre todos os alunos do curso X.

Onde “X” é um valor que varia de acordo com a preferência do usuário.

Com isso, conforme se pode verificar, foi possível realizar a busca de informações semanticamente, uma vez que o mesmo conhece o caminho dos dados, exibindo-os corretamente para o usuário final.

6 CONCLUSÃO

Através da Web Semântica é possível tornar um sistema ou uma página interpretável por agentes de softwares encarregados de extrair as informações necessárias corretamente.

O motivo pelo qual ainda não se podem realizar buscas como a mostrada neste trabalho, em toda a extensão da Web é porque as páginas, sistemas e documentos ainda não estão preparados para usar a esta tecnologia. Para que se torne realidade, o uso da Web Semântica em todos os seus aspectos, seria necessário uma mudança no paradigma de desenvolvimento de sistemas para a web adotando conceitos e padrões ainda pouco usados. Este processo leva tempo mas algo semelhante já aconteceu anteriormente com a gradativa substituição da web de conteúdo estático para a web de conteúdo dinâmico.

Pode-se notar que muitos dos serviços disponíveis na Web atualmente poderiam ser melhores utilizando os padrões da Web Semântica e por sua vez a qualidade dos serviços seria mais bem capacitada para atender demandas, auxiliando empresas e usuários a encontrem ou se beneficiar de algum serviço.

Foi verificada a importância de linguagens Semânticas como XML e RDF, que identificam onde estão as informações juntamente com o uso de ontologia para se criar um domínio de conhecimento sobre determinada área ou assunto para o compartilhamento destas informações caso necessário. Também foi verificado que ainda existem algumas melhorias que podem ser feitas como o uso de linguagem natural e o vínculo com banco de dados relacionais existentes.

O uso de padrões da web semântica juntamente com a linguagem natural (proposto para trabalhos futuros), quando empregadas juntas, fazem a diferença nos sistemas atuais, pois, com a linguagem natural a interpretação do que o usuário quer é feita e os padrões se encarregam de buscar tal informação, no caso de um sistema de busca por exemplo. Diferentemente do que se utiliza atualmente, onde as consultas as

informações em sistemas online são feitas de forma fixa deixando com que usuarios busquem apenas o que já fora programado.

Com este trabalho foi possível demonstrar o uso de alguns dos principais padrões da Web semântica empregados em sistema simples de busca de informações acadêmicas de alunos no qual pode-se verificar que é possível se ter resultados objetivos e claros para possivelmente se empregar essa metodologia em aplicações como o sistema Adx e por que não em sistemas na Web como paginas e serviços.

7 TRABALHOS FUTUROS

Com base nos resultados obtidos neste trabalho pode-se aferir que é viável a construção de sistemas que tiram proveito dos benefícios propostos pela web semântica. O escopo do trabalho não leva em conta outros aspectos que podem ilustrar bem os benefícios que a aplicação da web semântica. Alguns destes aspectos são propostos aqui como trabalhos futuros na área.

O uso de agentes inteligentes para trabalhar juntamente com a linguagem natural para interpretação clara da linguagem, a fim de facilitar a compreensão entre homem e máquina.

Também é indicado um trabalho específico para a conversão de um banco de dados existentes em uma ontologia, assim eliminando a etapa da criação da ontologia. A ideia é o desenvolvimento de um framework (junção de tecnologias, trabalhando junto para automatizar tarefas) para converter a estrutura de um banco de dados em uma ontologia de acordo com os padrões RDF ou OWL, para automatizar a migração de um sistema que já existe, dessa forma contribuindo para a expansão da Web Semântica.

Outra proposta para trabalhos relacionados à área é um estudo inserindo a linguagem natural para interpretação computacional de modo que não mais se use uma lógica definida apenas para algumas questões, fazendo com que o computador realmente possa entender o significado deste trabalho.

7.1 LINGUAGEM NATURAL

A proposta para este trabalho futuro é baseada apenas em técnicas de processamento de linguagem natural. Para dar continuidade a esta linha de pesquisa.

A idéia da linguagem natural é capacitar o computador a realizar operações com base no que for solicitado pelo usuário, utilizando métodos e técnicas para interpretar de forma natural a linguagem seja ela qual for, ou seja, passar para um agente de software o sentido e o real significado do que se busca ou de qual tarefa se deseja executar.

7.2 FRAMEWORK PARA CRIAÇÃO DE ONTOLOGIAS A PARTIR DE BANCO DE DADOS EXISTENTES

A proposta para o desenvolvimento Framework para criação de ontologias a partir de banco de dados existentes, se baseia na tese de que sistemas como o ADX, que já estão em funcionamento e já tem sua base de dados bem estruturada, com seus relacionamentos possam ter uma ontologia que englobe não só algumas partes do sistema, mas sim, o sistema como um todo.

Com esse framework será possível tornar um sistema mais inteligente, pois ele terá a estrutura Semântica representado pela ontologia e por sua vez, as informações dele extraídas serão tão quanto precisas e de acordo com for feitas às solicitações.

7.3 DESENVOLVIMENTO DE FRAMEWORK COMPLETO DA WEB SEMÂNTICA

Por fim a idéia pra esse outro trabalho é a junção dos dois trabalhos citados nas seções 7.2 e 7.3, para desenvolvimento de um framework completo, junramente com um estudo de caso sobre a segurança das informações e serviços na Web semântica.

A proposta é a criação de um framework para unir a linguagem natura e a integração com banco de dados relacionais, para assim concretizar em um único framework completo de desenvolvimento de acordo com padrões da Web Semântica.

Por fim, um estudo a fim de apresentar como é feita a camada de criptografia da Web Semântica, bem como a segurança de informações e integração entre serviços a fim de mostrar seus pontos positivos e negativos, e o que pode ser melhorado para atender se necessário os requisitos de segurança informando quais são e porque.

8 REFERÊNCIAS

ADX. **Sistema de gerenciamento acadêmico ADX**. Disponível em:

<<http://www.doctumtec.com.br:9673/doctumtec/produtos/desenvolvimento?abrir=adx>>.

Acesso em 22 de nov. de 2011.

ALMEIDA, Iêda *et al.* **Automação de bibliotecas e centros de documentação: O Processo de avaliação e seleção de softwares**. 1999.

ALVARENGA, Lidia; ROCHA, Renato; SOUZA. **A Web Semântica e suas contribuições para a ciência da informação**. Ciência da informação. 2004. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19652004000100016&lng=en&nrm=iso>. Acesso em 20 out. 2011.

BERNERS-LEE, Tim *et al.* **The semantic Web: a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities**. scientific american magazine, new york, mai. 2001- Berners-Lee, Tim; Hendler, James; Lassila, ora, 2001. Disponível em: <<http://www.scientificamerican.com/article.cfm?id=the-semantic-Web>> .Acesso em: 02 ago. 2011.

BERNERS-LEE, Tim *et. al.* **Uniform Resource Identifier (URI), 2004**. Disponível em: <<http://dret.net/biblio/reference/rfc3986>>. Acesso em 14 de nov. de 2011.

BERNERS-LEE, Tim; HENDLER, Jame; LASSILA, Ora. **The Semantic Web**. Scientific American, 2001. Disponível em: <<http://www.sciam.com/article.cfm?articleID=00048144-10D2-C70-84A9809EC588EF21>>. Acesso em 21 ago. 2011.

BERNERS-LEE, Tim. **A Proposal , 1989**. Disponível em:

<<http://www.w3.org/History/1989/proposal.html>>. Acesso em: 29 Out. de 2011.

BERNERS-LEE, Tim. **A PROPOSAl. (May 1990)**. Disponível em:

<<http://www.uazuay.edu.ec/bibliotecas/conectividad/pdf/Information%20Management%20A%20Proposal%20Original%20Tim%20Bernes-Lee.pdf>>. Acesso em 22 de jun 2011.

BIZER, Christian; WESTPHAL, Daniel; OLDAKOWSKI, Radoslaw *et al.* **RAP: Rdf Api for PHP**. Universidade de Berlin contributions from the semantic Web community, 2002.

Disponível em: <www.semanticscripting.org/SFSW2005/papers/Oldakowski-RAP.pdf>. Acesso em: 19 de nov. De 2011.

BRANDÃO, Anarosa A. F; LUCENA, Carlos José P. **Uma introdução à engenharia de ontologias no contexto da Web Semântica**. Departamento de informática puc do rio de janeiro. rio de janeiro. puc-rioinf. mcc29/02 nov. Disponível em: <http://www.dbd.puc-rio.br/depto_informatica/02_29_brandwww.dbd.puc-rio.br/depto_informatica/02_29_brandao.pdf>. Acesso em 25 de jul. de 2011.

BRAVO, Carlos D. O. **Geração Automática de Ontologias para Web Semântica, 2010. Universidade de Brasília – UNB**. Disponível em:<<http://monografias.cic.unb.br/dspace/handle/123456789/63>>. Acesso em 28 de jul. de 2011.

BREITMAN, Karin. **Web Semântica, 2005**. Rio de Janeiro. Disponível em: <<http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Web+Semântica#0>>. Acesso em 25 de ago. de 2011.

DDUBLIN CORE. **Utilizando Dublin Core, 2005**. Disponível em: <<http://dublinCore.org/documents/usageguide/>>. Acesso em 01 de nov. de 2011

DECKER, Stefan *et al.* **The Semantic Web-on The respective roles of xml and rdf**. IEEE internet computing, 2000. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.37.1151&rep=rep1&type=pdf>>. Acesso em 30 de out. de 2011.

DUBLIN CORE. **Dublin Core Metadata Element Set, Version 1.1, 2011**. Disponível em: <<http://dublincore.org/documents/dces/>> . Acesso em 12 de nov. de 2011.

ETSI. **XML Namespace URI, 2011**. Disponível em: <<http://portal.etsi.org/pnns/xml.asp>>. Acesso em 02 de nov. De 2011.

GAUTHIER, Fernando. **RDF E RDFSHEMA**. Universidade Federal de Santa Catarina. Disponível em: <<http://www.google.com/url?sa=t&rct=j&q=linguagem%20rdf&source=Web&cd=3&ved=0CD0QFjAC&url=http%3A%2F%2Fwww.inf.ufsc.br%2F~gauthier%2FEGC6006%2Fmaterial%2FAula%25202%2FRDF%2520e%2520RDFSchema.doc&ctbs=lr>>

%3Alang_1pt&ei=A1jBTqbVK4b00gHepri2BA&usg=AFQjCNHXHPWmnmZaYbYmbZx_hZg1ykXjOw&sig2=_yNkQwr0TdFUhw05ftaSOg&cad=rja>. Acesso em 03 de ago. 2011.

HORROCKS, Ian; ROAD, Oxford. **DAML+OIL: A Description Logic for The Semantic Web**. Computer, 2002. Acesso em: Disponível em: <<http://www.cs.ox.ac.uk/ian.horrocks/Publications/download/2002/ieeede2002.pdf>>. Acesso em 04 de nov. 2011.

IANNELLA, Renato. **Metadata: Enabling the Internet. The Information Professions and the Information**, 1997. Disponível em: <<http://archive.ifla.org/documents/libraries/cataloging/metadata/ianr1.pdf>>. Acesso em 10 de jul. 2011.

ISKOLD, Alex. **Semantic Web Patterns: A Guide To Semantic Technologies**. 2008. Disponível em: <http://www.readwriteweb.com/archives/semantic_web_patterns_a_guide_redux.php> . Acesso em: 29 Out. 2011.

LANGER, J. **Extensible Markup Language (XML)**. Annals of physics, (XML), 1969. Disponível em: <<http://www.mendeley.com/research/no-title-avail/>>. Acesso em 12 de out. 2011.

MARTINS, Eunice; SALDIAS, Cláudio . Protégé. **Programa de Pós-graduação em Engenharia de Automação e Sistemas , Programa de pós-graduação em Engenharia Mecânica. Universidade Federal de Santa Catarina (UFSC) - Florianópolis, SC, Brasil, 2006**. Disponível em: <http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=Web&cd=4&ved=0CDsQFjAD&url=http%3A%2F%2Fwww.das.ufsc.br%2F~gb%2Fpg-ia%2FProtege08%2Fclaudio_euclides_artigo_protege_rev5.pdf&ei=8B3FTqrsJ7OonsAKu9JWxCw&usg=AFQjCNE670xV6PkO2cExaXoRQsdyNQsjoQ&sig2=6Do_FSoWvW982Mqc hh3Zig>. Acesso em 02 de nov. de 2011.

MOREIRA, Alexandra; ALVARENGA, Lídia; OLIVEIRA, Alcione de Paula *et al.* **O Nível do conhecimento e os instrumentos de representação: Tesouros e ontologias**. Disponível em: <http://www.dgz.org.br/dez04/art_01.htm>. Acesso em 10 de jul. 2011.

MYSQL, 2011b. **MYSQL Documentation: mysql reference manuals**. Disponível em: <<http://dev.mysql.com/doc/>>. Acesso em 22 de nov. de 2011.

MYSQL. **WHY MYSQL?**, 2011. Disponível em: <<http://www.mysql.com/>>. Acesso em: 22 de nov. de 2011.

NEPOTE, Charles. **Web Sémantique**, 2008. Disponível em: <<http://Websemantique.org/AnalyseurSyntaxique>> . Acesso em 19 de nov. de 2011.

OBITKO, Marek. **Traduções entre ontologias em sistemas multi-agente. Dissertação, de engenharia elétrica, universidade técnica checa, em praga, 2007.** Disponível em: <<http://www.obitko.com/tutorials/ontologies-semantic-Web/rdf-query-language-sparql.html>>. Acesso em 22 de nov. de 2011.

OLIVEIRA, Fábio. **Estudo da Iniciativa Web Semântica da W3C, e do RDF/XML, 2004.** Disponível em: <http://www.linux.ime.usp.br/~cef/mac499-04/monografias/fbdo/node8.html>>. Acesso em 2 de nov. 2011.

PALMER, Sean B. **The Early History of**, 2004. Disponível em: <<http://infomesh.net/html/history/early/>>. Acesso em 13 de out. de 2011.

PASSOS, Marcel. **SPARQL NA IMPLEMENTAÇÃO DO QUIZ ONTOMÚSICA. Monografia. 2010. Universidade Feevale**, Novo Hamburgo. Disponível em: <http://tconline.feevale.br/tc/files/0001_2499.pdf>. Acesso em: 19 de nov. de 2011.

PICKLER, Maria Elisa. **Web Semântica: ontologias como ferramentas de representação do conhecimento. Perspectivas em Ciência da Informação. SciELO Brasil, 2007.** Disponível em: <<http://www.mendeley.com/research/no-title-avail/>> . Acesso em 23 de nov. de 2011.

PIRES, Roberta. **Documentação de Estudos em Linguística Teórica e Aplicada.** Disponível em: <http://www.scielo.br/scielo.php?pid=S0102-44502002000200008&script=sci_arttext>. Acesso em 28 de nov. de 2011.

Protégé, 2011. **The Protégé Ontology Editor and Knowledge Acquisition System.** Disponível em <http://protege.stanford.edu/> . Acesso em 18 de nov. de 2011.

Protégé, 2011b. **Documentation Protégé: Documentação do Protégé.** Disponível em: <<http://protege.stanford.edu/doc/owl/getting-started.html>>. Acesso em 18 de nov. de 2011.

Protégé, 2011c **Tutorial Owl**. Disponível em:

<protege.stanford.edu/conference/2005/slides/T2_OWLTutorialI_Drummond_final.pdf>.

Acesso em 19 de out. de 2011.

RAPAPI, 2011. **Documentation AP API FOR PHP, 2011**. Disponível em:

<<http://www4.wiwiwiss.fu-berlin.de/bizer/rdfapi/tests.html>>. Acesso em 19 de jul. de 2011.

SACHS, Eliza. **Noções básicas para uso do Protege, 2006**. Disponível em:

<http://mba.eci.ufmg.br/onto_frames/> . Acesso em 17 de nov. 2011.

SEMPREBOM, Tiago; CAMADA, Marcos; MENDONÇA, Igor. **Ontologias e**

Protégé. Programa de Pós-Graduação em Engenharia de Automação e Sistemas, 2007.

Disponível em: <www.das.ufsc.br/~gb/pg-ia/Protege07/ontologia_protege.pdf>. Acesso em 20 de jul. de 2011.

SOUZA, Márcia *et al.* **Informação para internet: Uso de metadados e o padrão dublin**

Core para catalogação de recursos eletrônicos na empresa. Computer, 2000. Disponível

em: <<http://orton.catie.ac.cr/cgi-bin/wxis.exe/?>

IsisScript=ACERVO.xis&method=post&formato=2&cantidad=1&expression=mfn=035510>. Acesso em 16 de nov. de 2011.

TAMBA-MECZ, Irène. **A Semântica, 2011**. Disponível em:

<<http://www.parabolaeditorial.com.br/DIAGSEMANTICA.pdf>> . Acesso em 15 de jun.

2011.

TOLENTINO, Ricardo José Vaz. **Aplicações Web em XML: Estágio atual e tendências**

futuras. Belo Horizonte: FACE-FUMEC, 2004. 150p.

W3C, 2004 d. **RDF / XML Specification Syntax**. Disponível em:

<http://translate.googleusercontent.com/translate_c?hl=pt-BR&langpair=en

[%7Cpt&rurl=translate.google.com&twu=1&u=http://www.w3.org/TR/2004/REC-rdf-](http://translate.google.com&twu=1&u=http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/&usg=ALkJrhj1Gbm01MdscMr9aaOwigl5Usf6Q)

[syntax-grammar-20040210/&usg=ALkJrhj1Gbm01MdscMr9aaOwigl5Usf6Q](http://translate.google.com&twu=1&u=http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/&usg=ALkJrhj1Gbm01MdscMr9aaOwigl5Usf6Q)>. Acesso

em: 14 de nov. de 2011.

W3C, 2004a. **RDF/XML Syntax Specification (Revised)**. Disponível em:

<<http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/>>. Acesso em 2 de

nov. de 2011.

W3C, 2004b. **RDF Resource Description Framework (RDF)**. Disponível em: <<http://www.w3.org/RDF/>>. Acesso em 14 de nov. de 2011.

W3C, 2004c. **W3C Recommendation 10 February 2004**. Disponível em: <<http://www.w3.org/TR/rdf-concepts/>>. Acesso em 14 de nov. 2011.

W3C, 2004e. **OWL Web Ontology Language 10 February 2004**. Disponível em: <<http://www.w3.org/TR/2004/REC-owl-features-20040210/>>. Acesso em 15 de nov. de 2011.

W3C. **Namespaces in XML, 1997**. Disponível em: <<http://www.w3.org/TR/1999/REC-xml-names-19990114/>>. Acesso em 2 de nov. de 2011.

W3C. **Platform for Internet Content Selection (PICS), 2009**. Disponível em: <<http://www.w3.org/PICS/>>. Acesso em: 15 Set. de 2011.

W3C. **SPARQL Query Language para RDF, 2008** . Disponível em: <<http://www.w3.org/TR/rdf-sparql-query/>>. Acesso em 13 de jun. 2011.

W3C. **Teste RDF Resultados Núcleo, 2002** .Disponível em: <<http://www.w3.org/2003/11/results/rdf-Core-tests>>. Acessado em: 19 de nov. de 2011.

W3C. **W3C Semantic Web Activity, 2010**. Disponível em: <<http://www.w3.org/2001/sw/>>. Acesso em: 2 set. de 2011.

W3C. **XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition).A Reformulation of HTML 4 in XML 1.0. W3C Recommendation 26 January 2000, revised 1 August 202, 2011**. Disponível em: <<http://www.w3.org/TR/202/REC-xhtml1-20281/>>. Acesso em 12 de nov. de 2011.

W3SCHOOLS. **Introdução à RDF, 2011**. Disponível em: <http://www.w3schools.com/RDF/rdf_intro.asp>. Acesso em 14 de nov. de 2011.

WEIBEL, Stuart. **The Dublin Core**, . Disponível em: <<http://dublinCore.org/resources/bibliography/>>. Acesso em 12 de nov. de 2011.