



**UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA (UESB)**  
**DEPARTAMENTO DE CIÊNCIAS EXATAS (DCE)**  
**BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**SQE: UMA FERRAMENTA DE BUSCA UTILIZANDO WEB SEMÂNTICA**

**LUIZ HILÁRIO FERREIRA DAMASCENA**

**VITÓRIA DA CONQUISTA – BA**

**2010**

**LUIZ HILÁRIO FERREIRA DAMASCENA**

**SQE: UMA FERRAMENTA DE BUSCA UTILIZANDO WEB SEMÂNTICA**

Monografia de conclusão de curso apresentada ao Departamento de Ciências Exatas (DCE) da Universidade Estadual do Sudoeste da Bahia (UESB) para obtenção do título de Bacharel em Ciência da Computação.

Área de Concentração: Engenharia de Software, Engenharia Web.

Orientador: Prof. Me. Stenio Longo Araujo

**VITÓRIA DA CONQUISTA – BA**

**2010**

## AGRADECIMENTOS

Primeiramente agradeço a Deus por cumprir mais esta etapa em minha vida. Ele que sempre me acompanhou e, com certeza, possibilitará mais vitórias.

Agradeço imensamente à minha família, em especial à minha mãe e ao meu pai, que me cuidaram durante toda a vida e sempre colocaram seus filhos como prioridade. Marcus (meu irmão), meu avô e avó, minha namorada, tios e primos que estiveram ao meu lado durante minha trajetória de vida.

Agradeço, ainda, a minha Tia Raulinda (in memória) e minha avó Lindaura (in memória) que foram como mães na minha vida e que desde minha infância contribuíram para minha educação.

Ao professor Stênio Longo que gentilmente e com muita boa vontade foi orientador deste trabalho. Seu comprometimento e contribuições me ajudaram muito neste trabalho, e seu conhecimento durante toda a graduação se tornou uma referência para mim.

À Celina, secretária do colegiado de Ciência da Computação, que foi a pessoa mais dedicada e atenciosa que conheci nesses últimos anos.

Ao professor Fabrício Souza que me ajudou inicialmente com este trabalho e sempre se dispôs a ajudar.

Agradeço aos professores: Adilson, Aline, Cátia, Chico, Fábio, Máisa e Roque. Que contribuíram para a minha formação acadêmica.

Aos meus amigos de longa data com os quais vivi muitos dos melhores e mais inesquecíveis momentos da vida: Vagner, Vinícius, Guilherme, Gisele, Franklin, Douglas, Jamille, Joseph e Ricardo.

A todos os meus colegas de trabalho, em especial a Vanêide e o Prof. Manoel Antônio (UESB), e aos meus amigos do SESI: Juliana, Daniel, Adalto, Flaviana, Juscelino, Gilmendes, Santiago e Ana Lúcia. Fontes de conhecimento, amizade sincera e inesquecível.

E por último, mas não menos importantes, os meus amigos de faculdade, que foram tantos e que tanto me ajudaram, só tenho a dizer-lhes: Muito obrigado! São: Henrique, Doug, Jadson, Poções (Diogo), Hesdras, Dino, Esdras, Sinthia, Marcos, Elias, Ramon, Gabriel, Marlovich, Orlando, Hamilton, Bruno, Poly e Diego.

*“Não se pode ensinar tudo a alguém, pode-se apenas ajudá-lo a encontrar por si mesmo.”*

Galileu Galilei

## RESUMO

O presente trabalho apresenta as principais tecnologias e padrões existentes para a Web Semântica, explorando também como suas tecnologias podem funcionar em conjunto com serviços *Web*. Na investigação e percepção de como as informações estão interligadas na Web Semântica e o uso de ontologias para organizar o conhecimento de informações na Internet foi desenvolvida uma ferramenta *Web* de busca textual em redes *Web* semânticas existentes. Neste desenvolvimento é abordada a construção da aplicação em foco de notações semânticas. Os serviços *Web* serão aproveitados colaborativamente para esta aplicação enriquecendo-a. A Web Semântica em sua totalidade mostra-se ainda como alvo a ser alcançado, mas ao decorrer deste trabalho foi possível utilizá-la para recuperação de informações da *Web*.

**Palavras-chaves:** Web Semântica. Serviços Web. Busca textual. Ontologias.

## ABSTRACT

This paper shows the main technologies and existing standards for the Semantic Web, exploring how its technologies can also operate in conjunction with Web services. In research and understanding of how information is linked in the Semantic Web and the use of ontologies for organizing knowledge of information on the Internet was developed a Web tool for searching textual networks existing semantic Web. This development is discussed building the application in focus semantics notations. Web services will be used for this application collaboratively enriching it. The Semantic Web as a whole still shows up as a target to be achieved, but the course of this paper could use it to retrieve information from the Web.

**Keywords:** Semantic Web, Web Services, Text search, Ontologies.

## LISTA DE FIGURAS

Figura 1	-	Arquitetura da Web Semântica.....	15
Figura 2	-	Visão prática das camadas da Web Semântica.....	15
Figura 3	-	Exemplo de um simples grafo RDF .....	20
Figura 4	-	Interface de gerenciamento <i>Web</i> repositório Sesame .....	29
Figura 5	-	Diagrama de caso de uso do SQE .....	36
Figura 6	-	Diagrama de classes do SQE .....	38
Figura 7	-	Visualização de uma aplicação cliente de um endpoint .....	39
Figura 8	-	Notação RDF para busca em domínio geral .....	41
Figura 9	-	Notação RDF para busca em domínio específico.....	42
Figura 10	-	Busca por RDF links e outras páginas web.....	43
Figura 11	-	Notação RDF tripla com sujeito ou objeto definido .....	43
Figura 12	-	Sugestões de termos com <i>Dbpedia URI Lookup</i> .....	46
Figura 13	-	Tela inicial do sistema .....	50
Figura 14	-	Resultado de consulta do sistema (Primeira aba).....	51
Figura 15	-	Resultado de consulta do sistema (Segunda aba).....	51
Figura 16	-	Resultado de consulta do sistema (Terceira aba) .....	52
Figura 17	-	Busca por URL's.....	52
Figura 18	-	Busca por Triplas.....	53
Figura 19	-	Tela inicial do sistema (Busca por localidades) .....	54
Figura 20	-	Resultado para lugares com coordenadas geográficas.....	54
Figura 21	-	Exemplo de resultado para busca por imagens.....	55

## LISTA DE ABREVIATURAS

**API:** *Application Programming Interface*  
**CSS:** *Cascading Style Sheet*  
**DTD:** *Data Type Definition*  
**DL:** *Description Logic*  
**HTML:** *Hypertext Markup Language*  
**HTTP:** *Hypertext Transfer Protocol*  
**OWL:** *Web Ontology Language*  
**RMI :** *Remote Method Invocation*  
**RDF:** *Resource Description Framework*  
**RDFS:** *RDF Schema. RDF Vocabulary Description Language*  
**RIA:** *Rich Internet Application*  
**RQDL:** *Relational Data Query Language*  
**RuleML:** *Rule Markup Language*  
**REST:** *Representational State Transfer*  
**SeRQL:** *Sesame Query Language*  
**SGML:** *Standard Generalized Markup Language*  
**SOAP:** *Simple Object Access Protocol*  
**SPARQL:** *SPARQL Protocol and RDF Query Language*  
**SWRL:** *Semantic Web Rule Language*  
**URI:** *Universal Resource Identifier*  
**URL:** *Universal Resource Locator*  
**XHTML:** *eXtensible HyperText Markup Language*  
**XML:** *Extensible Markup Language*  
**XSD:** *XML Schema Definition*  
**W3C:** *World Wide Web Consortium*



# SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>10</b>
1.1 OBJETIVOS .....	11
1.1.1 Objetivo Geral .....	11
1.1.2 Objetivos Específicos .....	11
1.2 JUSTIFICATIVA .....	11
<b>2 WEB SEMÂNTICA.....</b>	<b>14</b>
2.1 A CAMADA BÁSICA DE DADOS .....	16
2.1.1 Unicode .....	16
2.1.2 URI .....	17
2.2 A CAMADA DE DESCRIÇÃO SINTÁTICA .....	17
2.2.1 XML .....	17
2.2.2 Espaços de nomes .....	18
2.2.3 Esquema XML .....	19
2.3 A CAMADA DE DESCRIÇÃO SEMÂNTICA .....	19
2.3.1 RDF .....	19
2.3.2 Serializações RDF .....	21
2.3.2.1 RDF/XML .....	21
2.3.2.2 Turtle .....	22
2.3.2.3 N-Triples .....	23
2.3.2.4 Esquema RDF .....	23
2.3.3 Ontologias .....	24
2.3.3.1 OWL .....	25
2.4 CAMADAS DE LÓGICA, PROVA E CONFIANÇA .....	26
2.5 BUSCA DE INFORMAÇÕES NA WEB SEMÂNTICA COM SPARQL .....	27
2.6 FERRAMENTAS SEMÂNTICAS E DE ONTOLOGIAS .....	28
2.6.1 Sesame .....	29
2.6.2 Protégé .....	30
2.7 COMPARTILHAMENTO DE INFORMAÇÕES NA WEB SEMÂNTICA .....	30
2.7.1 Linked Data .....	31
2.7.2 DBpedia .....	31
<b>3 SQE (SPARQL QUERY ENGINE).....</b>	<b>34</b>
3.1 ANÁLISE DE REQUISITOS .....	34
3.2 DIAGRAMAS DO SISTEMA .....	35
3.2.1 Diagrama de caso de uso .....	35
3.2.2 Diagrama de classe .....	36
3.3 ARQUITETURA DA APLICAÇÃO .....	38
3.4 REPRESENTAÇÃO RDF DO SISTEMA .....	40
3.5 FERRAMENTAS UTILIZADAS .....	44
3.6 INTEROPERABILIDADE DO SISTEMA COM DBPEDIA LOOKUP .....	45
3.7 INTEROPERABILIDADE COM FLICKR SERVICE .....	46
3.8 PESQUISA TEXTUAL E TÉCNICA DE ENUMERAÇÃO DOS RESULTADOS .....	47
<b>4 APRESENTAÇÃO DO SPARQL QUERY ENGINE.....</b>	<b>50</b>

4.1 INTERFACE DO SQE .....	50
<b>5 CONCLUSÃO .....</b>	<b>56</b>
5.1 CONSIDERAÇÕES INICIAIS .....	56
5.2 DIFICULDADES ENCONTRADAS.....	57
5.3 TRABALHOS FUTUROS .....	57
<b>REFERÊNCIAS.....</b>	<b>59</b>

## 1 INTRODUÇÃO

A *World Wide Web* atual é um dos principais meios de comunicação, realização de negócios e aquisição de conhecimento da nossa era. Nesta sociedade da informação em que a quantidade de conteúdo na *Web* cresce diariamente, organizar e recuperar informações relevantes não é uma tarefa simples. Segundo Berners-Lee (2001), a propriedade essencial da *Web* é a universalidade. Um *link* de hipertexto tem a força de que uma coisa pode ligar a qualquer coisa, entre milhões de documentos existentes.

Atualmente a *Web* vive a chamada *Web 2.0*. Este termo, criado pela empresa *O'Reilly Media*, serviu para designar a nova geração de serviços tendo a *Web* como plataforma, baseados na participação coletiva. O'Reilly (2005), criador do termo, define ainda que não se deve pensar em software que esteja no cliente ou servidor, mas no espaço entre eles. Sítios bastante conhecidos que estão no contexto da *Web 2.0* são sites como a *Wikipedia*, *Flickr* e as redes sociais, como Orkut, FaceBook Hi5, etc.

O conteúdo *Web* existente é ideal para o consumo humano. Informações contidas em documentos e bases de dados são praticamente apresentadas de maneira estrutural em sítios. Computadores, neste aspecto, tornam-se apenas processadores das requisições feitas pelas pessoas e exibindo resultados que somente são entendidos pelo próprio ser humano.

A *Web Semântica* é definida como uma extensão da *Web* atual em que as informações têm significados bem específicos, possibilitando a cooperação entre humanos e computadores (BERNERS-LEE et al., 2001). Ela deve ter acesso a uma coleção de dados estruturados e um conjunto de regras para poder inferir e conduzir a processos automatizados em serviços de buscas, dentre outros. Precisamente toda informação pode ser mais bem representada dentro de um domínio específico através de regras e relacionamentos. Neste sentido, a ontologia aparece para a *Web Semântica* de forma significativa ajudando a estruturar e definir significado dos termos dentro de um conjunto.

Tecnologias da *Web Semântica* começam a ser usadas por governos como o dos E.U.A para dar maior transparência aos seus dados, *Google* e *Yahoo!* agora processam conteúdo *Web* semântico em páginas da *Internet* e a *Microsoft* segue

rumos semelhantes. A Web Semântica começa a atrair pequenos desenvolvedores pela abertura de exploração do conteúdo *Web* de maneira inovadora.

## **1.1 OBJETIVOS**

### **1.1.1 Objetivo Geral**

Este trabalho propõe o desenvolvimento de uma ferramenta de busca textual multilíngue utilizando-se da arquitetura da Web Semântica para recuperar informações de uma base de dados.

### **1.1.2 Objetivos Específicos**

1. Apresentar os principais modelos de representação da informação semântica: RDF (*Resource Description Framework*), *Turtle*, RDF/XML, etc;
2. Demonstrar o uso da linguagem SPARQL para buscas em documentos RDF;
3. Identificar e utilizar API's (*Application Programming Interface*), *frameworks* semânticos e de ontologias para construção de uma aplicação Web Semântica.

## **1.2 JUSTIFICATIVA**

Pesquisas na área de Web Semântica tiveram início após 2001 com a publicação, na revista *Scientific American*, do artigo "*The Semantic Web*", escrito por Tim Bernes Lee, Ora Lassila e James Handler, especificações mais bem definidas

foram lançadas em 2004 e projetos significativos começaram em 2006. Estudos são relativamente novos e podem vir a ser o futuro não tão distante de agentes de busca, Serviços Web e demais aplicações baseadas em ontologias, entre outras.

O desenvolvimento de aplicações Web semânticas ainda é um desafio por várias razões, motivado principalmente pelo fato de grande parte das informações hoje existentes não estarem armazenadas de forma a poderem ter um significado a ser compreendido semanticamente ou inferido automaticamente, mas implementações nesta perspectiva trazem uma nova visão sobre a construção e cooperação entre aplicações na internet. Uma das áreas promissoras da Web Semântica envolve justamente recuperação de informações, já que a padronização da sua arquitetura e de documentos na *Web* permitirá que aplicações que as utilizem, pesquisem e procurem por dados de maneira ainda mais precisa.

### **1.3 METODOLOGIA**

Na primeira etapa, foi feito um estudo das bibliografias citadas posteriormente, focando nas características das camadas da Web Semântica, recursos disponíveis que operam em cada nível, além da construção de ontologias e reutilização.

A segunda fase constitui-se da análise e projeto da aplicação. Nessa etapa foram utilizados alguns dos conceitos compreendidos do estudo teórico, para verificar as ferramentas e linguagens disponíveis que melhor se adequaram ao sistema desenvolvido.

Na terceira etapa realizou-se a implementação do estudo de caso, avaliou-se a construção do sistema e finalmente foram realizadas sugestões e propostas de aperfeiçoamento para trabalhos futuros.

### **1.4 TRABALHOS RELACIONADOS**

Esta seção tem por objetivo mostrar alguns dos trabalhos científicos relacionados com esta monografia, os trabalhos discutidos a seguir foram estudados

para melhor compreensão das tecnologias presentes na Web Semântica, além de dar uma visão do que já se tem feito em sistemas que adotam a Web Semântica.

Dias (2009), faz uma explanação geral dos principais conceitos envolvidos na Web Semântica, aborda os problemas de estruturação de documentos na Web e principalmente as questões de interoperabilidade de informações na *Web*, envolve alguns temas abordados no desenvolvimento da aplicação que são relacionadas à busca e recuperação da informação e descrição das informações com metadados. Discute o papel das ontologias para orientar e organizar o conhecimento humano, porém nesta questão o trabalho torna-se ineficiente pelo fato de não abordar os atuais padrões da W3C nesta área.

Gaspar (2008), é feita a implementação de um motor de busca com recuperação das informações com o domínio relativo a futebol. Faz uma descrição sucinta da arquitetura do servidor e base de dados com informações gerenciadas por uma ontologia, descreve também a arquitetura do cliente e define as operações sobre o sistema.

Trucolo (2008), faz a proposta de construção de um agente inteligente simples que às consultas semânticas que serão implementados e discutidos posteriormente. Recupere informação na *Web* baseada nas tecnologias da Web Semântica. É feita uma discussão teórica de um modelo baseado em vetor de termos com modelo booleano para buscas textuais. Tal modelo usará conceitos de sinonímia e antonímia.

Erling (2009) aborda os principais problemas e funcionalidades encontradas na construção de um serviço que utiliza base de dados remota armazenada sobre o *Virtuoso Universal Server*, comenta e mostra exemplos de consultas que combinem RDF com pesquisas de texto e demonstram como agregar métodos de enumeração de resultados às consultas semânticas que serão implementados e discutidos posteriormente.

## 2 WEB SEMÂNTICA

O desenvolvimento de aplicações no contexto da Web Semântica é feito observando sua estrutura de camadas, sendo uma camada construída com base em definições da inferior. A justificativa para esta abordagem é padronizar o modelo que promove o desenvolvimento de aplicações, sendo que muitos grupos de cientistas e companhias podem vir a adotá-las futuramente. Além do que é melhor tentar resolver problemas maiores subdividindo-os. Por ser um fato novo, a Web Semântica não deve esperar até que todo o conteúdo esteja padronizado para seu uso de maneira mais efetiva. Para tanto, deve-se prover de mecanismos que atendam os seguintes princípios:

- Compatibilidade total com a camada inferior: Agentes plenamente conscientes de que uma camada também deve ser capaz de interpretar e utilizar informação escrita em níveis inferiores. Por exemplo, os agentes de conhecimento da semântica da OWL podem tirar pleno partido das informações escritas em RDF e Esquema RDF (ANTONIOU; VAN HARMELEN, 2008);
- Compreensão parcial da camada superior: O modelo propõe que os agentes plenamente conscientes de uma camada devem ser capazes de tomar, pelo menos parcial, vantagem de informações em níveis mais elevados. Por exemplo, um agente de conhecimento apenas de RDF e Esquema RDF pode interpretar escritas de conhecimentos em OWL, em parte, por desrespeitar os elementos que vão além do RDF e Esquema RDF. Não há nenhuma exigência para todas as ferramentas para fornecerem essa funcionalidade, embora esta deva ser adotada (ANTONIOU; VAN HARMELEN, 2008).

Essas são as idéias principais, servem como base da orientação para o desenvolvimento da Web Semântica. A figura 1 ilustra as camadas da arquitetura, que é dividida da seguinte forma:

- Camada Básica de Dados: *Unicode* e *URI(Universal Resource Identifier)*;
- Camada de Descrição Sintática: *XML (Extensible Markup Language)*, Espaço de nomes e Esquema XML;
- Camada de Descrição Semântica: RDF, Esquema RDF e Ontologias.
- Camada Lógica;

- Camada de Prova;
- Camada de Confiança.

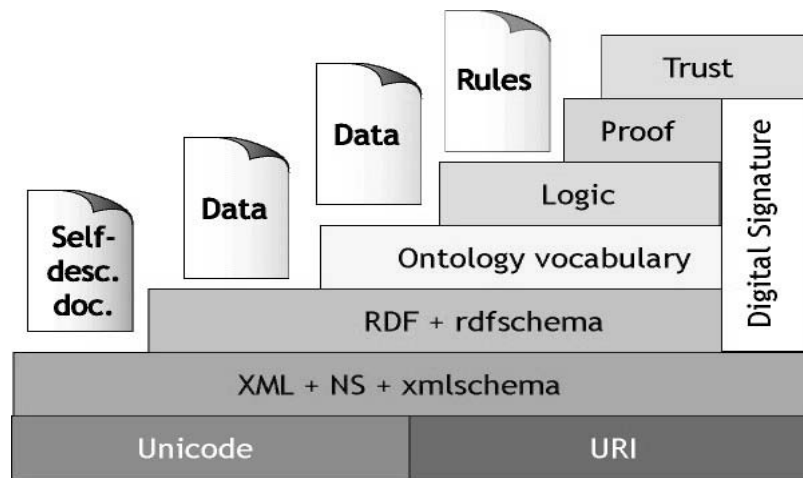


Fig. 1: Arquitetura da Web Semântica. Antoniou and van Harmelen, (2008).

Toda a arquitetura da Web Semântica ainda é muito nova, padrões podem ser redefinidos para o melhor desenvolvimento, compatibilidade e integração entre as camadas. Segaran, Evans e Taylor apresentam uma visão mais prática das tecnologias presentes e seu estágio de desenvolvimento nas aplicações:

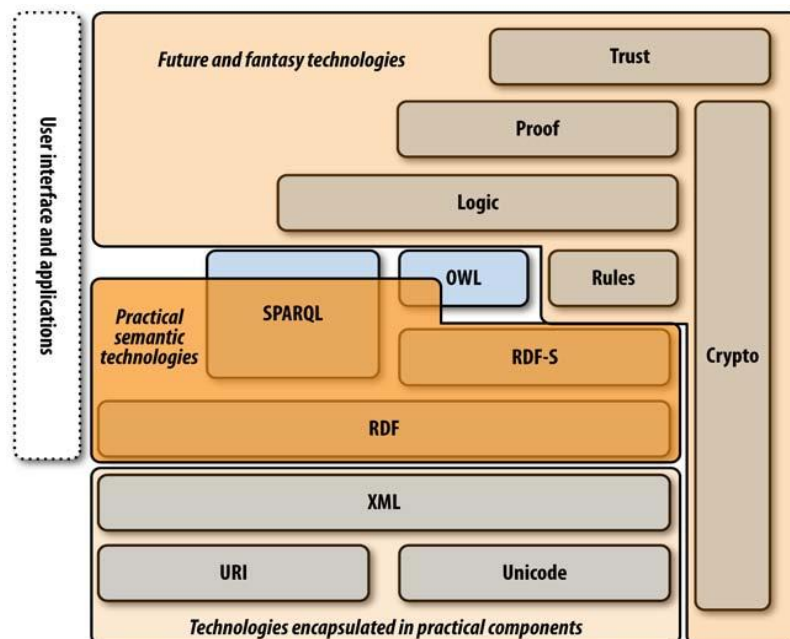


Fig. 2: Visão prática das camadas da Web Semântica. Segaran, Evans e Taylor (2009).

O nível mais baixo (URI, *Unicode* e XML) corresponde aos padrões já conhecidos em outras arquiteturas e que estão encapsulados sobre as tecnologias



atuais usadas pela Web Semântica. O nível intermediário que compreende RDF, RDF-S, SPARQL e parte da OWL (*Web Ontology Language*) são os conceitos mais utilizados para o desenvolvimento, projetados com fim de utilização em aplicações semânticas. O nível mais alto (Lógica, Prova e Confiança) reúne as camadas que englobam as tecnologias futuras que permitirão construir as aplicações da visão originalmente proposta na concepção da Web Semântica (SEGARAN; EVANS; TAYLOR, 2009).

As seguintes seções têm como objetivo explicar os componentes que envolvem cada nível da arquitetura da Web Semântica original, com foco principal nas camadas que englobam conceitos práticos que atualmente fornecem capacidade para desenvolvimento de implementações, mostrando os padrões presentes, funções e relações com as camadas mais próximas.

## **2.1 A CAMADA BÁSICA DE DADOS**

A camada mais inferior é a camada básica de dados suportando o padrão *Unicode*, um conjunto de caracteres identificados por um número único para cada caractere, independente de idioma, plataforma ou aplicação. Junto a ele está a URI, um conjunto de símbolos utilizados para denominar ou identificar um recurso na *Web*. Tais características validam a codificação e endereçamento de recurso na Web Semântica.

### **2.1.1 Unicode**

O *Unicode* fornece um único número para cada caractere, independente da máquina ou plataforma operacional. Além do XML, o padrão *Unicode* é necessário para padrões modernos como o Java, *JavaScript*, WML, etc. Seu suporte a diversos sistemas operacionais e todos os navegadores atualmente disponíveis tornam fato importante para adoção de padrão na Web Semântica pela W3C.

### 2.1.2 URI

Recursos na Web Semântica devem ser identificáveis, para que estes possam ser acessados, o termo URI sugere o ideal de um identificador único para o recurso em questão, de modo que para cada recurso existente haja um único identificador, cada URI é composta por diversos caracteres *Unicode*.

## 2.2 A CAMADA DE DESCRIÇÃO SINTÁTICA

Conforme a seção anterior, esta camada refere-se à modelagem de sintaxe para descrever um recurso. Segundo Prazeres (2009):

“É importante notar que a camada de descrição sintática (espaço de nomes XML, XML e esquema XML) utiliza os padrões da camada inferior da arquitetura da Web Semântica, que incluem URI e o padrão Unicode. Todos os recursos contidos em um documento XML são endereçados através de uma URI e os documentos são texto puro com codificação Unicode.”

### 2.2.1 XML

Atualmente o HTML (*Hyper Text Markup Language*) é o principal padrão em que as páginas *Web* são escritas. Este foi derivado do SGML (*Standard Generalized Markup Language*), um padrão internacional (ISO 8879) para definição de um dispositivo para representação da informação, simples de entendimento tanto para pessoas quanto para computadores (ANTONIOU; VAN HARMELEN, 2008).

A definição como padrão foi importante, pois permite eficiente comunicação, suporte tecnológico e suporte a ambientes colaborativos. Linguagens em conformidade com o SGML são aplicadas, já que o SGML foi considerado muito complexo para o propósito da *Web*. HTML e XML são algumas de suas aplicações. Sendo assim, o XML é uma recomendação da W3C. XML também pode ser definido como uma meta-linguagem de marcação que determina uma sintaxe usada para

definir outras linguagens de marcação para domínios específicos.

Os elementos XML adicionam estrutura ao conteúdo dos documentos, não se atendo às questões de formatação e apresentação desses documentos (PRAZERES, 2009). Permite, assim, que cada pedaço da informação seja descrita, sendo que relações podem ser identificadas por meio da estruturação de seus elementos. Por exemplo, uma *tag* **<autor>** que apareça dentro de um **<livro>** pode ser entendida como pertencente ao livro em questão, facilitando o processamento de uma aplicação por máquinas.

Além da permissão encontrada no XML de definir suas próprias *tags* e a possibilidade de deduzir alguns relacionamentos através da proximidade das marcações, outra vantagem é permitir restrições quanto ao valor de constantes. Assim: um valor numérico pode ser restringido a um valor máximo e/ou mínimo com uma quantidade fixada de dígitos.

### **2.2.2 Espaços de nomes**

Uma das vantagens do XML é que informações de vários recursos podem ser acessadas. Tecnicamente, um documento XML pode usar mais de um DTD (*Data Type Definition*) ou esquema. Esta flexibilidade pode causar o aparecimento de ambiguidades. O uso de nomes globais é de grande importância porque significa que as declarações possam sempre sofrer uniões sem que sejam necessárias modificações nos nomes. Desde que cada declaração que constitui um grafo possa ser usada sem alterações, então grafos inteiros podem ser transportados e combinados sem modificações, o que é uma grande vantagem quando se trata da troca de informação (HEBELER et al. 2009).

Por exemplo, o nome “autor” pode se referir a diferentes significados (autor de livro, autor de obras de arte, autor de música, etc.). Para evitar este problema associa-se um espaço de nomes a um URI, dessa forma mesmo que elementos de mesmo nome com semânticas diferentes ocorram em documentos diferentes, estes devem ser únicos em seus espaços de nomes.

### 2.2.3 Esquema XML

Os esquemas XML são documentos que são usados para definir e validar o conteúdo e estrutura de dados XML, da mesma forma que um banco de dados relacional define e valida tabelas, colunas, e tipos .

Um esquema XML define e descreve certos tipos de dados XML usando o idioma de definição do esquema XML (XSD). Elementos do esquema XML (elementos, atributos, tipos, e grupos) são usados para definir a estrutura válida, conteúdo dos dados válidos e relacionamentos de certos tipos de dados XML. Esquemas XML também podem fornecer valores padrões para atributos e elementos.

## 2.3 A CAMADA DE DESCRIÇÃO SEMÂNTICA

Esta seção descreve os principais padrões presentes na camada, discutindo sobre as principais formas de serialização do RDF. Será abordada também a linguagem de descrição de vocabulário, chamada de Esquema RDF e a OWL e sua capacidade de melhor prover domínio em aplicações através dela.

### 2.3.1 RDF

Recomendado pela W3C, o RDF descreve simples afirmações sobre objetos *Web*. O RDF não necessariamente depende do XML, mas baseia-se na sintaxe deste para sua construção (ANTONIOU; VAN HARMELEN, 2008). Sendo assim, seu principal objetivo é o de representar toda forma de recurso possível na *Web* em meta-dados, tais como: foto, áudio, vídeo, documentos HTML ou até mesmo pessoas.

A especificação de RDF define um modelo de dados para representação de informação sobre recursos na *Web* (W3C, 2004). Segundo Hong (2007), RDF é para

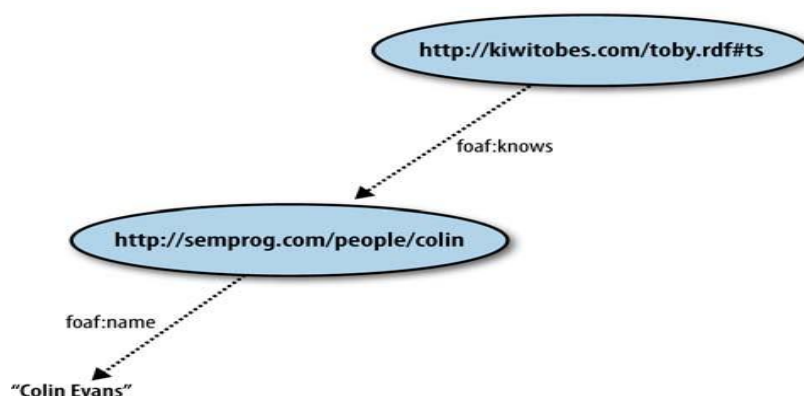
a Web Semântica o que o HTML foi para a *Web*.

Yu (2007) descreve as principais características em se utilizar RDF:

- RDF é o modelo para descrição de metadados recomendada pela W3C para a Web Semântica;
- RDF é capaz de descrever qualquer recurso independente do seu domínio;
- RDF fornece a base para codificação, troca e reutilização estruturada de metadados;
- Por ser estruturado, o RDF permite ser compreendido por máquinas. Podendo assim fazer operações úteis com o conhecimento expresso em RDF;
- Permite a interoperabilidade entre aplicativos na *Web*.

O RDF representa a informação como um grafo de declarações relacionadas. A declaração é composta por três elementos: sujeito, predicado e objeto. O sujeito é sempre um recurso, por isso sempre tem um URI associado, o predicado sempre é também um recurso e tem uma URI ligada a ele, por último o objeto é o valor da propriedade do sujeito podendo ser outro recurso ou um valor literal.

A tripla por si só é uma poderosa ferramenta para integração de informação. Triplas são apenas coleções de URIs e literais, sendo que cada um desses possuem um escopo global (HEBELER et al. 2009).



**Fig. 3:** Exemplo de um simples grafo RDF. Segaran, Evans e Taylor (2009)

Grafos não possuem raiz. Algumas outras representações, por exemplo, XML, possuem uma estrutura em árvore. Em um documento XML, o elemento *root* da árvore tem um significado especial porque todos os outros elementos são orientados a respeito do *root*. Ao tentar fazer a junção de duas árvores XML, pode ser difícil

determinar qual deve ser o *root*. Em um grafo RDF, ao contrário, nenhum recurso individual possui qualquer significância quando comparado a qualquer outro.

O uso de nomes globais é de grande importância porque significa que grafos podem sempre sofrer uniões sem que sejam necessárias modificações nos nomes. Já que as declarações RDF não precisam ser modificadas quando movidas de um sistema para outro, elas são válidas em qualquer contexto.

Em um grafo é possível encontrar com um recurso ao qual não temos um URI relacionado a ele. Isto é, um problema, que é tratado em RDF com a inserção dos chamados “nodos brancos”, estes nodos são conjunto de URI que não possuem Espaço de nomes e referenciam recursos não acessíveis. Nodos são de contexto local, ou seja, não podem ser referenciados fora do local em que são criados.

### 2.3.2 Serializações RDF

RDF possibilita um meio de representar a informação de forma entendível ao ser humano, porém de maneira abstrata. Para prover entendimento conciso a máquinas sem presença de ambiguidade é necessário que esta seja serializável. O processo de serialização fornece uma maneira de converter o modelo abstrato para um modelo concreto. Podendo ser um arquivo ou um *stream* de *bytes*, por exemplo. (HEBELER *et al.* 2009). Há vários formatos de serialização, sendo os mais conhecidos e utilizados o RDF/XML, Turtle e N-Triples.

Desde que grafos tenham a mesma estrutura, cada formato de serialização deve representar os mesmos construtores das asserções: URI e literais. Percebe-se que sua principal diferença são as possibilidades sintáticas de descrever a mesma informação.

#### 2.3.2.1 RDF/XML

Como descreve o nome, sua sintaxe é baseada no XML, é a serialização padrão para RDF recomendada pela W3C. Uma aplicação Web Semântica deve

fornecer suporte para este formato. (SEGARAN; EVANS; TAYLOR, 2009).

Conceitualmente RDF/XML é descrito com um conjunto de pequenas descrições, cada qual percorre um caminho em um grafo RDF representando sujeito, predicado e objeto (SEGARAN; EVANS; TAYLOR, 2009).

Todo o conteúdo RDF está na tag **rdf:RDF**, a qual contém uma série de elementos **rdf:Description**. Também é importante notar a declaração do espaço de nome XML na tag **rdf:RDF**. Esse documento é definido de acordo ao seguinte recurso: <http://www.w3.org/1999/02/22-rdf-syntax-ns>. Por convenção, esse espaço de nomes é sempre abreviado para **rdf**.

Declarações em RDF/XML são agrupadas nos elementos **<rdf:Description>** anteriormente descritos. Cada um destes elementos tem um atributo **rdf:about** que descreve o sujeito que relaciona-se com as declarações presentes na descrição. Cada um destes elementos, então, define o predicado e objeto das declarações.

#### 2.3.2.2 Turtle

O *Turtle* é outra serialização da sintaxe para RDF. Comparada a outras serializações, *Turtle* é mais amigável ao entendimento humano e de sintaxe mais legível. Ela não é uma linguagem baseada em XML, foi projetada especificamente para RDF. Pelo fato de não ter de representar grafos como uma árvore, ela pode ser mais concisa e de leitura fácil. (HEBELER et al. 2009).

*Turtle* usa um simples formato para descrever uma tripla. O sujeito, predicado e objeto são escritos em apenas uma linha, separados por um espaço em branco e terminados com um ponto. O uso do ponto-e-vírgula pode agrupar vários predicados e objetos para um mesmo sujeito. Vírgulas também podem ser usadas para indicar que determinado predicado e objeto têm o mesmo sujeito da declaração anterior, podendo assim ser usadas repetidamente.

Os recursos podem ser escritos de duas maneiras. As URIs aparecem em sua totalidade e entre “<” “>” (menor que e maior que) ou com um prefixo pré-definido. Literais em *Turtle* ficam em aspas duplas.

### 2.3.2.3 N-Triplas

N-Triplas foi usada pelo *W3C Core Working Group* em diversos casos de teste enquanto atualizações nas especificações RDF eram desenvolvidas. Pela facilidade de uso, ela é indicada principalmente para aplicações em processo de teste ou *debug* (SEGARAN; EVANS; TAYLOR, 2009).

Esta notação é considerada a maneira mais simples de escrever declarações RDF. Ela usa a mesma sintaxe do Turtle, mas impõe algumas restrições para simplificação. N-Triplas, por exemplo, não suporta a notação @prefix ou vírgula, nem, tampouco, ponto-e-vírgula. Uma declaração é representada em apenas uma linha contendo apenas o símbolo do ponto para indicar o fim da declaração. Sua simplicidade também é atrativa para aplicações que precisam que a informação, além de ser serializável, também seja transmitida em um fluxo de dados (HEBELER et al. 2009).

### 2.3.2.4 Esquema RDF

RDF fornece um modelo virtualmente ilimitado para descrever informação. Entretanto, se usada sozinha, RDF não representa o significado, ou semântica, por trás das descrições. Na Web Semântica, essa capacidade de descrever semântica é fornecida por Esquema RDF (RDFS) e Linguagem de Ontologias para Web (OWL) (HEBELER et al., 2009).

Expressar significado implica ter um vocabulário comum, ou em termos de RDF é ter uma coleção de recursos que permitem descrever outros recursos. O RDFS possibilita especificar ontologias limitadas através de hierarquia de classes e de propriedades padronizadas. As principais contribuições de RDFS são a adição formal de conceitos semânticos de classes, propriedades, generalizações, domínio e restrições a propriedades (PRAZERES, 2009).

O padrão RDFS, no entanto, é limitado para a expressividade desejada da Web Semântica. Assim sendo, para prover mais expressividade e ampliar a capacidade de inferência, conceitos semânticos foram incorporados em uma camada



acima do RDF e RDFS. Antoniou e Van Harmelen (2003) citam alguns aspectos que são oferecidos por RDFS. São eles:

- Propriedades com escopo local: em RDFS não se pode declarar restrições a espaços de valores para apenas algumas classes;
- Classes disjuntas: não existe relação de disjunção entre classes em RDFS. Por exemplo, pessoas do sexo masculino e feminino são classes disjuntas;
- Combinações booleanas de classes: com RDFS não é possível criar novas classes a partir de combinações de outras classes, tais como união, intersecção e complemento;
- Restrições de cardinalidade: é impossível expressar restrições de cardinalidade em propriedades com RDFS, ou seja, não dá para expressar, por exemplo, que uma pessoa tem exatamente uma mãe biológica.

### 2.3.3 Ontologias

Ontologias são utilizadas para capturar conhecimento sobre um domínio de interesse. Uma ontologia descreve os conceitos de um domínio e também as relações que existem entre esses conceitos. As diversas linguagens para construção de ontologias fornecem diferentes funcionalidades. O padrão mais recente de linguagens para ontologias é o OWL, desenvolvido pelo W3C (HORRIDGE et al., 2008).

Borst (1997) define ontologias como uma distribuição formal e explícita de uma conceitualização compartilhada.

Em computação de modo geral o termo ontologias é muito utilizado para englobar um conjunto de definições de conceitos, propriedades, relações, restrições, axiomas, processos e eventos que descrevem o domínio do universo em disucrsão. Tais definições podem então ser utilizadas por aplicações e agentes de software a utilizarem semântica formal, precisa e clara para que processar a informação descrita pela ontologia e usar esta informações em aplicações inteligentes (PRAZERES, 2009).

### 2.3.3.1 OWL

Recursos na *Web* são inerentemente distribuídos e como resultado as declarações de recursos contidos na Web Semântica são também distribuídos. OWL suporta este tipo de conhecimento distribuído, pois é construído sobre RDF que permite declarar e descrever recursos localmente ou consultá-los remotamente. OWL também fornece um mecanismo para a importação e reutilização de ontologias em um ambiente distribuído (HEBELER et al., 2009).

A OWL amplia o vocabulário RDFS com adição de recursos adicionais que podem ser usados para construir ontologias mais expressivas para *Web*. OWL introduz restrições adicionais sobre estrutura e conteúdo de documentos RDF, a fim de fazer processamento e raciocínio decidíveis e computável. OWL usa o RDF e RDFS, Esquema XML e espaços de nomes.

Ontologias OWL são normalmente armazenadas como documentos na *Web*. Cada documento consiste em um cabeçalho opcional da ontologia, anotações de classe e das definições de propriedades (mais formalmente conhecido como axiomas), os fatos sobre os indivíduos e definições de tipo de dados (HEBELER et al., 2009).

Por ser OWL baseada no modelo RDF, não há distinção explícita entre a ontologia e os dados da ontologia que ela descreve. Uma maneira menos formal de dizer isso é que não há separação oficial entre ontologias e instâncias. A divisão é arbitrária e não afetam o significado das informações. No entanto, é uma prática comum em ontologias mantê-las separadamente dos dados que descrevem. Ontologias são compostas basicamente de três elementos: classes, indivíduos e propriedades.

Uma classe OWL é uma espécie de recurso especial que representa um conjunto de recursos que compartilham características comuns ou similares, de alguma forma. Um recurso que é um membro de uma classe é chamado de um indivíduo e representa uma instância de dessa classe.

A propriedade em OWL é um recurso que é usado como um predicado em declarações que indivíduos descrevem. Existem dois principais tipos de propriedades em OWL: propriedades entre objetos que ligam indivíduos a outros indivíduos e as propriedades entre tipos de dados que os indivíduos estão

associados a valores literais.

As ontologias OWL podem ser classificadas em três espécies, de acordo com a sub-linguagem utilizada: *OWL-Lite*, *OWL-DL (Description Logic)* e *OWL-Full*. A característica principal de cada sub-linguagem é a sua expressividade: a *OWL-Lite* é a menos expressiva; a *OWL-Full* é a mais expressiva; a expressividade da *OWL-DL* está entre as duas, entre a *OWL-Lite* e a *OWL-Full*. Horridge (2005) descreve-as como:

- A *OWL-Lite* é a sub-linguagem sintaticamente mais simples. Destina-se a situações em que apenas são necessárias restrições e uma hierarquia de classe simples;
- A *OWL-DL* é mais expressiva que a *OWL-Lite* e baseia-se em lógica descritiva, um fragmento de Lógica de Primeira Ordem, passível, portanto, de raciocínio automático. É possível assim computar automaticamente a hierarquia de classes e verificar inconsistências na ontologia;
- A *OWL-Full* é a sub-linguagem OWL mais expressiva. Destina-se a situações onde alta expressividade é mais importante do que garantir decidibilidade ou completeza da linguagem. Não é possível efetuar inferências em ontologias *OWL-Full*.

## 2.4 CAMADAS DE LÓGICA, PROVA E CONFIANÇA

Embora OWL permita modelar a maioria dos conceitos, há algumas regras que não são possíveis de expressar, por exemplo, a seguinte afirmação: “se um cliente comprar em uma mesma loja (cativo) e tem mais de 60 anos ele deve receber desconto nessa loja”. Pode ser representado na seguinte proposição lógica:

$$\text{clienteCativo}(X) \wedge \text{idade}(X) > 60 \rightarrow \text{desconto}(X).$$

Esse tipo de declaração não é possível de ser representada com OWL. Assim, existem estudos na direção de estender a OWL com regras SWRL (*Semantic Web Rule Language*). O SWRL é uma linguagem baseada na combinação de OWL com a linguagem RuleML (*Rule Markup Language*) para representação de fatos e regras. RuleML utiliza um subconjunto da linguagem Prolog (PRAZERES, 2009).

Segundo Prazeres (2009), as camadas de Prova e Confiança ainda não possuem implementações concretas e Swartz (2006), diz que nem todas as fontes e recursos na Web Semântica deverão ser confiáveis assim como não é na *Web* atual.

A *Web* só alcançará todo o seu potencial quando os usuários tiverem a confiança em suas operações (segurança) e na qualidade das informações fornecidas (ANTONIOU; VAN HARMELEN, 2008).

Assim, é necessária uma forma de garantir que essas informações sejam confiáveis. Segundo Swartz (2006), isto deverá ser realizado com o uso de assinaturas digitais. Todas as declarações RDF, por exemplo, deverão ser assinadas digitalmente e cada pessoa poderá ter certeza de quem criou aquela declaração e ainda informar aos seus programas quais assinaturas são confiáveis e o computador pode decidir, com base nas assinaturas, se a informação que está processando é originada de uma fonte confiável.

## **2.5 BUSCA DE INFORMAÇÕES NA WEB SEMÂNTICA COM SPARQL**

Fazer consultas na Web Semântica requer uma linguagem que reconheça RDF como sintaxe fundamental. Então, pesquisar em linguagens baseadas em RDF como OWL não exige procedimentos especiais ou recursos da linguagem (HEBELER, *et al.* 2009). SPARQL é um acrônimo recursivo para *SPARQL Protocol RDF Query Language*, uma recomendação da W3C. Existem outras linguagens de consulta RDF como RDQL (*RDF Data Query Language*) e SeRQL (*Sesame RDF Query Language*). Embora estas diferentes linguagens possuam recursos semelhantes, a padronização pela W3C do SPARQL como linguagem de busca para a Web Semântica e a grande quantidade de *endpoints* públicos que são capazes de interpretá-la, fazem desta o principal mecanismo de busca da Web Semântica. Um *endpoint* é como um serviço que aceita e processa consultas SPARQL e retorna em diferentes formatos dependendo da consulta. Esses *endpoints* são disponíveis via HTTP (*Hypertext Transfer Protocol*) e devem seguir o protocolo SPARQL.

Nota-se que SPARQL é uma linguagem e protocolo. Várias bibliotecas permitem que programadores foquem somente na sintaxe da linguagem. O protocolo é utilizado para descrever como um cliente faz requisições a um *endpoint* SPARQL.

Em termos de sintaxe, são possíveis de fazer quatro tipos de consultas em SPARQL:

- Consultas *SELECT*: realizadas através da linguagem por meio da cláusula *SELECT* (palavra-chave) permite trazer informações que combinam com determinado sujeitos, predicados e objetos. Contém ao menos uma cláusula *WHERE* associada e outros filtros podem lhe ser adicionados.
- Consultas *CONSTRUCT*: Permite reformular variáveis ligadas em qualquer grafo desde que contenha somente triplas válidas. Este recurso possibilita uma fácil maneira de converter grafos RDF em ontologias OWL e outras formas. É realizada com a cláusula *CONSTRUCT*.
- Consultas de *ASK*: Torna capaz saber se uma determinada condição existe em um grafo. Retornando somente os valores verdadeiro ou falso. Pode evitar que grandes consultas com *SELECT* ou *CONSTRUCT* sejam submetidas. Realizada com a cláusula *ASK*.
- Consultas de *DESCRIBE*: Permite que um cliente saiba da estrutura do grafo sem ter conhecimento total sobre o mesmo. Em termos mais simples, o cliente pode utilizar o que sabe sobre os dados para descobrir relacionamentos presentes no grafo.

## 2.6 FERRAMENTAS SEMÂNTICAS E DE ONTOLOGIAS

Esta seção descreve algumas das ferramentas que possibilitam a construção de aplicações baseadas na arquitetura da Web Semântica, foram escolhidas as principais ferramentas semânticas amplamente utilizadas pela comunidade de desenvolvedores, com seus projetos ativos e em constante desenvolvimento. Importante lembrar que outras soluções de software existem, mas as apresentadas adiante formam um consenso pela comunidade Web Semântica quanto um padrão para desenvolvimento e foram também utilizadas na construção do sistema a ser apresentado em capítulos seguintes.

### 2.6.1 Sesame

*Sesame* é um *framework* de código aberto com suporte a RDFS inferência e realização de buscas. Foi originalmente desenvolvido pela *Aduna* (conhecida antes como *Administrator*) como um protótipo para o grupo de pesquisa *On-To-Knowledge*. Agora ele é continuamente desenvolvido pela *Aduna* em cooperação com a Fundação *NLnet*, desenvolvedores da *OntoText*, e um grupo de desenvolvedores voluntários que contribuem com ideias reportam problemas e soluções.

*Sesame* foi desenvolvido de maneira a priorizar a flexibilidade. Ele pode atuar em uma das camadas acima de uma variedade de sistemas de armazenamento: Banco de dados relacionais (MySQL e PostgreSQL), armazenamento em memória, sistemas de arquivos, dentre outros; e fornece suporte ao seu gerenciamento, através de um interface *Web* que precisa estar rodando sobre um servidor *Java Web* (ver figura 4) e também através da *Sesame Java API* que suporta acessos locais e remotos através de HTTP e RMI (*Remote Method Invocation*) respectivamente, está em conformidade com as mais conhecidas linguagens de busca, como SPARQL. A última versão do *Sesame* é a 2.3.2 (lançada em dezesseis de setembro de 2010) a compatibilidade com projetos que usam versões 1.x não são garantidas. Existem dezenas de *plugins* que trabalham junto com o *Sesame* para estender suas funcionalidades. Entre estes o mais conhecido é o *Elmo* que funciona como um *JavaBean* para persistência de elementos de uma ontologia, permite criar aplicações em que desenvolvedores trabalhem com bases de dados RDF/OWL semelhante a orientação com objetos.

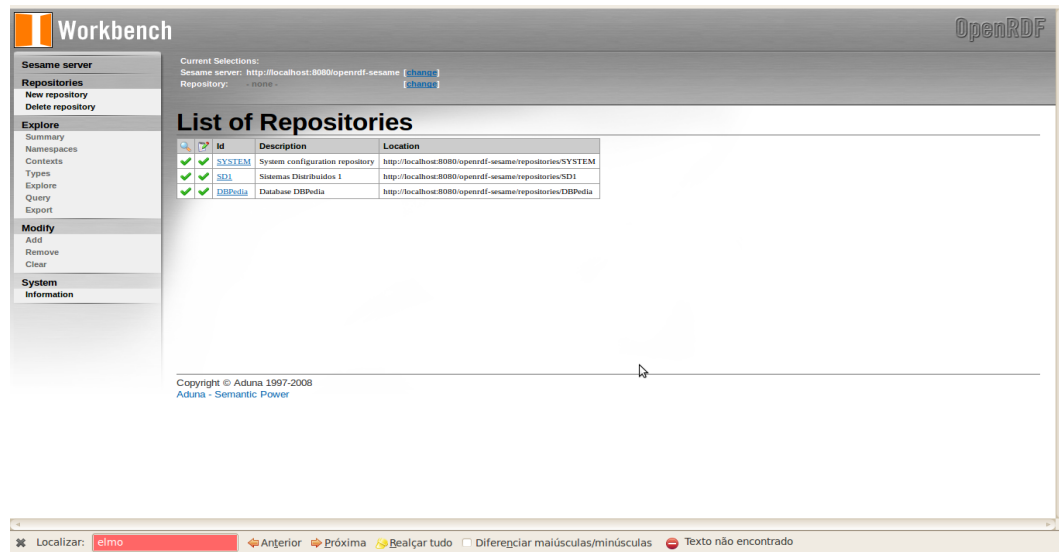


Fig. 4: Interface de gerenciamento web repositório Sesame.

## 2.6.2 Protégé

Desenvolvida pela Universidade de *Stanford*, esta ferramenta possibilita a criação de ontologias em diversas linguagens, incluindo OWL. *Protégé* é construído em Java e possui código aberto. Possui uma grande quantidade de plugins que possibilita integração com outras ferramentas como o *Jena* e *Sesame*, além de linguagens como SPARQL.

## 2.7 COMPARTILHAMENTO DE INFORMAÇÕES NA WEB SEMÂNTICA

Esta seção busca explicar os mais importantes princípios que orientam a publicação e compartilhamento de dados na Web Semântica. De modo que a aplicação que será descrita no capítulo seguinte utiliza parte dos dados compartilhados na base de dados DBpedia que integra o *Linked Data*.

### 2.7.1 Linked Data

O termo *Linked Data* é hoje um projeto para ligar dados que não estavam previamente ligados, usando outros métodos diferentemente dos encontrados na *Web 2.0*. Pode ser definido mais precisamente como um conjunto de melhores práticas para expor, compartilhar e conectar pedaços de dados, informações e conhecimento na Web Semântica, usando URI's e RDF.

O projeto foi sucintamente escrito por Tim Berners-Lee, este apresenta as quatro principais regras descritas abaixo para o *Linked Data*:

1. Utilizar URI's como identificador para todas as coisas;
2. Usar URI's HTTP para que as pessoas possam procurá-las.
3. Quando alguém procurar uma URI, fornecer informações úteis.
4. Incluir links para outras URI's, de forma que possam ser descobertas mais coisas.

### 2.7.2 DBpedia

*DBpedia* é um projeto para extração de informação estruturada da *Wikipedia* integrando-a com outras bases de dados semânticas disponíveis na Web. *DBpedia* permite realizar consultas personalizadas dentre os dados que contém. Seu desenvolvimento é creditado na concepção de que a imensa quantidade de dados presentes em enciclopédias virtuais como a *Wikipedia* pode ser usado de maneiras novas e interessantes, e que pode inspirar a novos mecanismos de navegação, ligando e melhorando a enciclopédia em si. É uma base que está sobre a licença GNU, ideal para um projeto que permite investigação e estudos da base de dados.

O conteúdo atual da *DBpedia* (ver quadro 1) consiste da descrição de aproximadamente três milhões e quatrocentos mil recursos, dos quais a metade está classificada dentro em uma consistente ontologia.



Classificação dos recursos	Número de artigos
Lugares	413000
Pessoas	312000
Organizações	140000
Espécies	146000
Músicas e álbuns	94000
Filmes	49000

**Quadro 1:** Quantidade de artigos por classes.

Fonte: DBpedia, 2010.

*DBpedia* usa RDF como modelo para representação de dados e publicação na *Web*. A base de conhecimento consiste em mais de um bilhão de triplas RDF. Esses dados são resultados de recursos que são modelados conforme uma ontologia, que abrange todas as categorias descritas acima, entre outras com propriedades específicas e generalizadas de classes. Todos os dados são armazenados sobre o BD *OpenLinkVirtuoso*. O acesso a este conjunto de informação pode ser feito através de *dumps* feitos da base de dados no formato N-triplas ou *Turtle* em mais de 92 idiomas, ou através de acesso a um *endpoint* que receba consultas SPARQL e devolve triplas de informação RDF em diferentes formatos. A ontologia da *DBpedia* é definida como uma ontologia de multi-domínio se comparada a outras ontologias que cobrem domínios específicos.

Todos os recursos da *DBpedia* são identificados na forma:

`http://dbpedia.org/resource/Nome_do_recurso`

onde a URL (*Universal Resource Locator*) é equivalente há termos retirados dos artigos em inglês da Wikipédia da seguinte forma:

`http://en.wikipedia.org/wiki/Nome`

Todos os recursos podem ser descritos por diversas propriedades definidas na ontologia. As propriedades mais básicas são:

- *label* (rótulos): Sequência de caractere que identifica um nome para o recurso;
- *short abstract* (pequenos resumos): Descrição abreviada sobre o recurso;
- *long abstract* (resumos estendidos): Descrição mais detalhada dos recursos. Geralmente são *short abstracts*, acrescidos de maiores informação;
- *depiction* (representação de imagem): *link* para uma imagem que

representa o recurso. Nem sempre é possível obter URI relacionada a esta propriedade.

O quadro 2 apenas mostra a quantidade de resumos existentes por idioma, percebe-se que o idioma textos em inglês possui a maior quantidade de resumos.

<b>Inglês</b>	3144000	<b>Espanhol</b>	362000
<b>Alemão</b>	545000	<b>Japonês</b>	275000
<b>Francês</b>	503000	<b>Português</b>	367000
<b>Polonês</b>	430000	<b>Sueco</b>	213000
<b>Holandês</b>	392000	<b>Chinês</b>	179000
<b>Italiano</b>	381000		

**Quadro 2:** Quantidade de resumos por idioma.  
Fonte: DBpedia, 2010.

### 3 SQE (SPARQL Query Engine)

Devido à grande cobertura de domínios que é dada pela ontologia, foi proposto o desenvolvimento de um sistema que fosse capaz de trazer tanto informações específicas de um conceito definido como propriedades gerais compartilhadas a todos os recursos da base de dados da *DBpedia*. Foram definidos requisitos a serem atendidos e diagramas a serem especificados para que a realização do sistema cumprisse com os objetivos descritos anteriormente, focando como podem ser implementados sistemas com emprego de padrões e tecnologias da Web Semântica para recuperação de informações.

O sistema desenvolvido foi chamado de SQE (**SPARQL Query Engine**), pois grande parte dele é construída sobre a linguagem SPARQL.

#### 3.1 ANÁLISE DE REQUISITOS

A análise de requisitos captura as intenções e necessidades do usuário sobre o sistema a ser desenvolvido. Para o sistema em questão, como sua principal função é a de recuperar informações a partir de texto digitado pelo usuário, elencou-se um conjunto de requisitos divididos entre:

a) **Funcionais:**

- Permitir que o usuário escolha o idioma em que será feita a pesquisa;
- Fornecer ao usuário sugestões de busca em tempo real, de acordo a entrada de dados em questão;
- Deve ser capaz de descobrir novas informações a partir dos resultados obtidos.

b) **Não funcionais:**

- O sistema deve processar e devolver as informações em um tempo que seja considerado satisfatório ao usuário;
- Ser um sistema *Web*;
- Ser portátil, ou seja, funcionar independente de navegadores.

De acordo com essas principais necessidades, o sistema foi desenvolvido utilizando as tecnologias e conceitos presentes no capítulo anterior. Na seção a seguir serão discutidos os diagramas que ajudam a melhor descrever o sistema.

Os idiomas escolhidos para esta aplicação foram o inglês, alemão e português. A escolha dos dois primeiros foi o fato destes apresentarem a maior quantidade de resumos na base de dados associados as URI's, conseqüentemente podem dar um maior número de respostas às consultas. O português logicamente por ser o idioma oficial do nosso país e nono em quantidade de resumos na base de dados, entra como terceira opção.

## 3.2 DIAGRAMAS DO SISTEMA

### 3.2.1 Diagrama de caso de uso

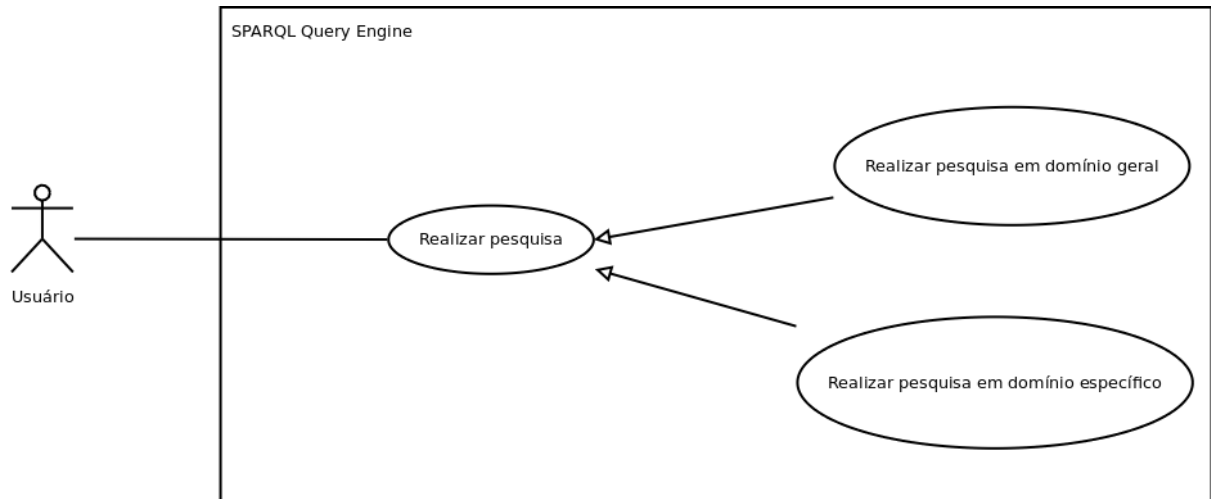
A figura 5 ilustra o caso de uso que representa o sistema. Este tem somente um ator que é responsável por toda informação de entrada no sistema. O usuário geralmente é representado na figura de uma pessoa. Descrição dos casos de uso:

**Realizar pesquisa:** O usuário entrará com os parâmetros com o qual deseja fazer a busca, para que o sistema tente encontrar resultados de acordo estes critérios. Para que isso aconteça, o usuário deverá inserir as palavras-chaves e o idioma de busca. Os resultados são apresentados na mesma página em que o usuário faz a pesquisa.

É importante lembrar que, devido o sistema trabalhar com base de dados remotas e serviços web, erros de inviabilidade de conexão podem ocorrer momentaneamente ou até indisponibilidade dos serviços externos por tempo mais prolongado, nestas condições pesquisas não podem ser feitas interrompendo o fluxo entre os casos de uso. Para o usuário é informada sobre a impossibilidade de realizar as operações.

**Realizar pesquisa em multi-domínio:** Reflete a pesquisa em todos os recursos que possuem similaridade no rótulo com os termos passados pelo caso de uso acima. Sem restrições à ontologia.

**Pesquisar em domínio específico:** O sistema pesquisa de acordo os termos passados, por localidades (cidades, locais históricos, etc), baseando-se no rótulo dos recursos, mostrando valores de propriedades dos recursos que pertencem apenas a classe Lugares (owl:Places) ou de coordenadas geográficas.



**Fig. 5:** Diagrama de caso de uso do SQE.

### 3.2.2 Diagrama de classe

O diagrama da figura 6 representa as principais classes do sistema e seus relacionamentos, que assim podem ser descritos:

**Grafo:** Classe que implementa a conexão com *endpoints* remotos. Esta classe é responsável por gerenciar o acesso à base de dados *DBpedia*, podendo também ser utilizada para conexão com qualquer outro *endpoint* que trabalhe com bases sobre notação OWL e RDF.

**Consulta:** Esta classe é responsável por formular as consultas e executá-las contra uma base de dados representada em um objeto Grafo. Traduções da representação de grafo RDF para a linguagem SPARQL são feitas dentro desta classe, além de verificação de validade e tempo de execução das consultas.

**Resultado e Lugares:** Retorno das consultas é convertido para objetos Java para que possam ser mais bem manipulados para a exibição na interface com usuário, foi feito o mapeamento das propriedades envolvidas nas consultas para atributos de classes Java, como resultado foram obtidas duas classes: Resultado e Lugares. Sendo que a segunda é uma especialização da primeira, de maneira a

possibilitar o armazenamento de dados de uma pesquisa mais refinada. Ambas podem também ser entendidas apenas como escopo de objeto do conceito de triplas RDF.

**Triplas:** Implementa mais precisamente o conceito de triplas RDF (sujeito, predicado e objeto). Esta classe também tem a mesma função das duas classes acima, porém sua principal diferença é ser genérica quanto aos possíveis tipos de triplas que podem retornar de uma pesquisa na base de dados. Esta classe foi usada para consultas que não se sabe com certeza o tipo de dados que pode retornar.

**Util:** Esta classe contém um conjunto de métodos estáticos auxiliares para formatar o texto de busca para que possam compor as consultas em SPARQL, como colocar informações sobre idiomas a qual se referem os termos, formatar visualização dos dados, dentre outras.

**BeanControlador:** Classe responsável por processar as informações entradas pelo usuário através da interface e devolvê-las. Esta classe também compreende a chamada de um serviço web para sugestões de texto, inicializar Grafo para requisições e chamar a execução de métodos de consulta e busca de imagens.

**FlickrRest:** Classe que é responsável pela chamada de requisições a serviços do Flickr para obter imagens. Implementa alguns dos métodos da API do *Flickr*®, que foram necessárias para obter a URL das imagens. Todos os métodos retornam resultados diante da análise de documentos XML.

**Imagem:** Esta classe é utilizada para criação de objetos em que se armazena resultados obtidos de busca para imagens. Cada objeto contém especificamente a URL da imagem e outra para o álbum a qual pertence.

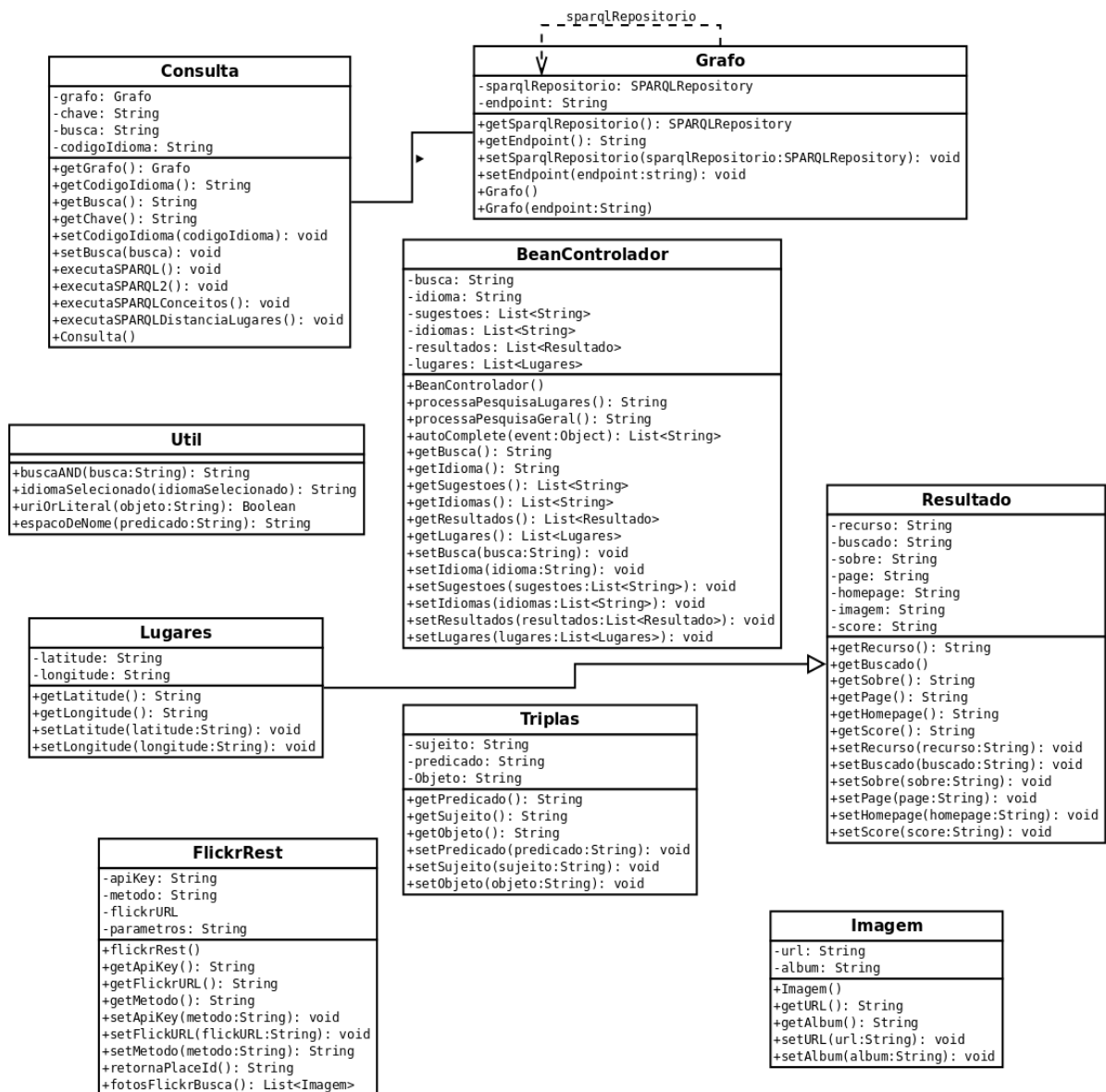


Fig. 6: Diagrama de classes SQE.

### 3.3 ARQUITETURA DA APLICAÇÃO

A figura 7 mostra como a aplicação torna-se um cliente, acessando a estrutura de dados que está organizada diante dos relacionamentos da ontologia.

1 – Gerenciamento de anotações – Pode ser dividida em duas partes: Uma para mecanismos de persistência e outra acesso da API. Em uma aplicação distribuída, a base de dados deve prover um SPARQL *endpoint* para receber

pesquisas ou mudanças feitas do cliente. A camada de acesso por API fornece um mecanismo para que aplicações clientes recebam os dados.

Para este projeto o *endpoint* é conectado apenas através de uma *string* que identifica a URL do mesmo.

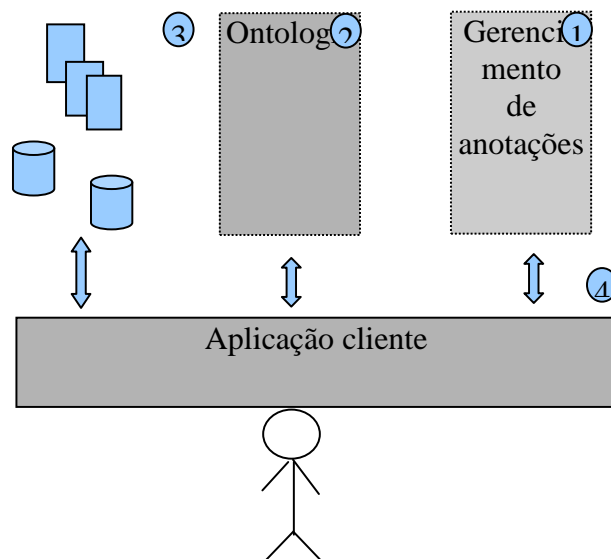
2 – Gerenciamento de ontologia: Provê mecanismos para que as ontologias possam ser usadas para pesquisa das informações necessárias ou para estendê-las, caso necessário. O aspecto importante deste gerenciamento é que ele provê à aplicação desenvolvida uma série de conceitos e relacionamentos que podem ser usados para gerar informação.

Neste trabalho foi utilizado a ontologia em formato OWL da *DBpedia* para estudar as relações entre todas as propriedades utilizadas. Foram analisados seus Espaços de nomes comumente utilizados além de seus possíveis sujeitos e objetos presentes em triplas que as envolvam.

3 – Dados sem estruturação: Os dados sem estruturação devem fazer parte de conceitos e propriedades associados para que uma API possa acessar os recursos de forma estruturada.

4 – Aplicação cliente: Deve ser capaz de buscar e receber dados através de buscas semânticas para específicos tipos de anotações ou conceitos específicos associados a estas anotações.

O sistema utiliza SPARQL para consultar os grafos criados sobre a notação de RDF, trazendo as informações definidas com base nos relacionamentos expostos da ontologia.



**Fig. 7:** Visualização de uma aplicação cliente de um *endpoint*.



### 3.4 REPRESENTAÇÃO RDF DO SISTEMA

A representação das informações que provêm da base de dados pode ser vista em notação de grafo RDF. Utilizar este conceito para visualizar resultados de consultas contribui não somente para entender melhor como funciona o sistema, mas também quais são as propriedades abordadas e quais informações pretende-se extrair.

Aqui serão apresentadas as principais visões em RDF do sistema, em que cada busca é definida em termos de conjunto de triplas RDF.

Os espaços de nomes utilizados:

```
foaf: <http://xmlns.com/foaf/0.1/>
dbprop: <http://dbpedia.org/property/>
owl: <http://www.w3.org/2002/07/owl#>
dbpedia: <http://dbpedia.org/ontology/>
rdf: <http://www.w3.org/1999/02/22-rdf-syntax-
ns#>
```

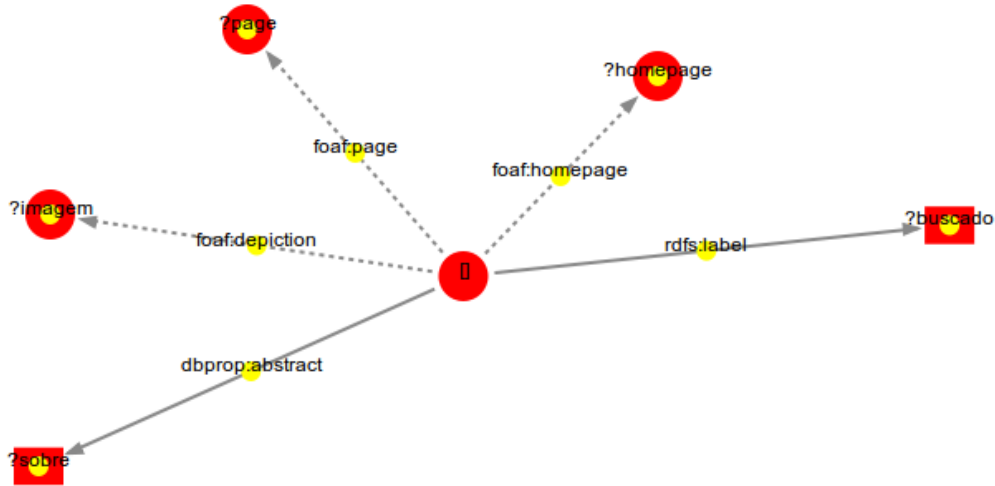
A figura 8 representa a busca em domínio geral. Esta visualização permite conceitualmente determinar que:

Cada busca a ser efetuada retornará duas informações com valor literal. Ambas são de caráter obrigatório.

A variável **?buscado** é a representação da propriedade ***rdfs:label***.

- O valor da variável **?sobre** contém o valor da propriedade **dbprop:abstract**;
- A busca retornará um URI representada pela variável “[ ]” (dois colchetes) que é o sujeito de todas as declarações feitas na consulta. A busca pode agregar resultados opcionais de URI's que descrevem outras propriedades dos recursos;
- A variável **?page** é utilizada para descrever a URI que identifica uma página na internet como página oficial do recurso;
- O valor de **?imagem** armazena o valor da propriedade **foaf:depiction**. Uma URI para um arquivo de imagem;
- A variável **?homepage** contém uma URI que é um artigo da Wikipedia.

Refere-se ao valor da propriedade **foaf:homepage**.



**Fig. 8:** Notação RDF para busca em domínio geral.

Em uma busca de domínio específico pode se procurar por novos atributos da ontologia que possibilitem descrever o recurso mais detalhadamente. Desta, foram utilizados novos relacionamentos entre recursos, propriedades e que estendam o grafo RDF anterior.

O grafo da figura 9 em relação a 8 não remove nenhuma tripla. No entanto, outras foram adicionadas para que, desta forma, seja possível obter outras informações mais específicas dos recursos que atendem ao valor da propriedade **rdf:type**. Esses relacionamentos, uma vez adicionados, foram associados como obrigatórios, assim mesmo que uma URI possa ser identificada como do tipo de uma coordenada geográfica, ou um lugar (**dbpedia:Places**) ela só será parte do resultado de busca caso contenha valores para as triplas que envolvem latitude e longitude. Outro fator que advém da obrigatoriedade de valores das novas propriedades é a menor quantidade de recursos a serem pesquisados já que somente uma parte dos recursos atende aos quatro novos requisitos.

- A latitude e longitude de um recurso são definidos para as variáveis **?latitude** e **?longitude**, respectivamente. Estes valores na ordem em que aparecem são definições das propriedades:

- `<http://www.w3.org/2003/01/geo/wgs84_pos#lat>`

- `<http://www.w3.org/2003/01/geo/wgs84_pos#long>`

A classe dentro da ontologia para representar pontos com coordenadas geográficas é representada na definição da URI:

`<http://www.opengis.net/gml/_Feature>`

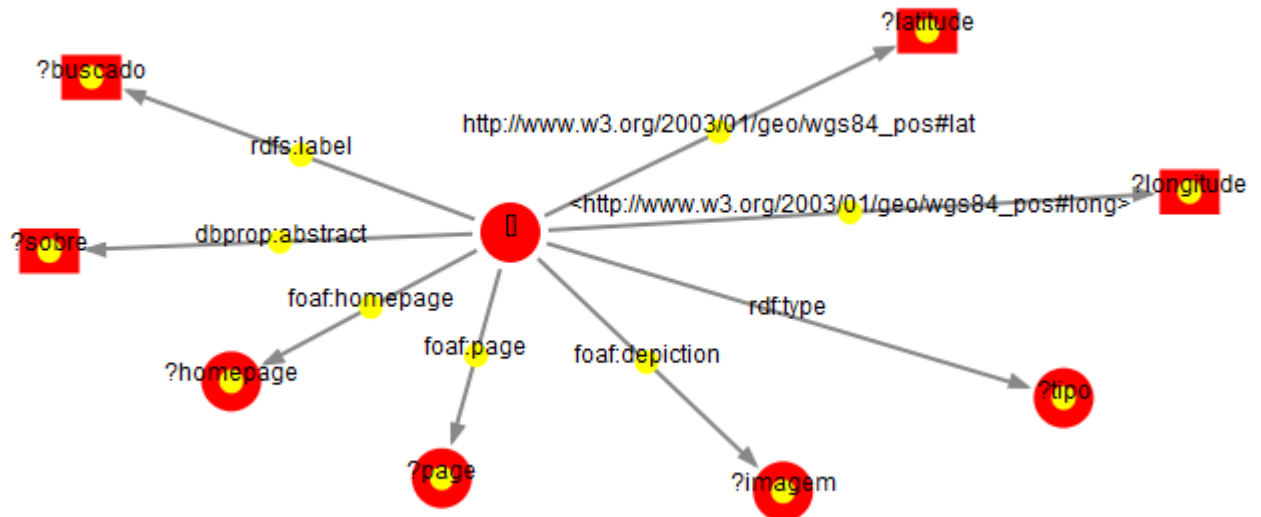
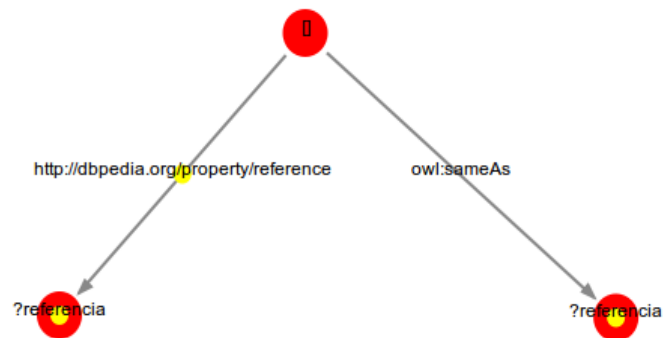


Fig. 9: Notação RDF para busca em domínio específico.

O grafo apresentado na figura 10 mostra uma outra característica do sistema que é a possibilidade de resgatar *links* RDF de outras bases semânticas e URL's de páginas na internet que contenham mais informações sobre um resultado de busca. Estes são resgatados através da união de propriedades **dbprop:reference** e **owl:sameAs** que possibilitem identificar tais características em uma única variável de consulta.

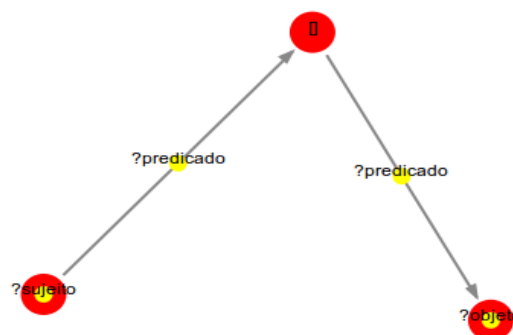
Esta consulta utiliza-se do conceito de união de conjuntos para unir dois resultados distintos em uma só resposta, desta forma deve ser dada a mesma variável aos objetos das triplas envolvidas, a única restrição neste conjunto é que todos os resultados sejam URI's.



**Fig 10:** Busca por RDF links e outras páginas *Web*.

O grafo da figura 11 apresenta uma consulta em que a URI, retornada como resposta, é ao mesmo tempo sujeito e objeto da busca. Isso produz o efeito de achar toda e qualquer literal ou URL relacionada a ela de maneira que o usuário saiba todas as informações da ontologia que de alguma maneira está ligada ao resultado apresentado. Formalmente o grafo abaixo nos mostra que:

- A URI representada por “[ ]” é um dos resultados da busca, sofre a ação de ser sujeito e objeto da consulta. Então os resultados são unidos de forma a trazer todas as URI's e literais que sejam objeto ou sujeito de uma URI da busca.
- A variável representada por **?sujeito** gera respostas quando a URI é o objeto. O valor da variável **?objeto** está ligada quando a URI é o sujeito da tripla.



**Fig. 11:** Notação RDF tripla com sujeito ou objeto definido.

### 3.5 FERRAMENTAS UTILIZADAS

O sistema foi desenvolvido utilizando *Sesame Framework* com a API *Elmo* integrando-as à tecnologia *Java Server Faces* com *Richfaces* e com o Tomcat 6.0, além dos recursos que a linguagem Java oferece. A aplicação em sua essencialidade torna-se um cliente da base *DBpedia* e ao mesmo tempo consome serviços *Web*.

A integração destas tecnologias foi combinada de maneira que trabalhasse remotamente com servidores de dados através de *endpoints* públicos e retornasse informações definidas nos grafos da seções anteriores. Abaixo, resume-se o papel de cada *framework* na construção do sistema:

- *Sesame Framework*: No desenvolvimento foi utilizado um conjunto de funções da sua API, para validar e realizar consultas em SPARQL, converter valores RDF para atributos de classes Java;
- *Elmo*: Embora, como descrito no capítulo anterior deste trabalho, sua principal funcionalidade seja persistir modelos RDF para objetos Java, seu principal uso para este trabalho foi utilizar sua capacidade de conectar a um repositório SPARQL. Isso é devido à implementação na biblioteca de uma extensão de um repositório *Sesame* HTTP comum. Com isso é possível trabalhar com todos os métodos válidos a um repositório *Sesame* local;
- *Tomcat*: É um servidor *Web* para Java, licenciado como software livre. Pode-se dizer resumidamente que é um *container* de *servelets*. A aplicação cliente desenvolvida ficou hospedada em um servidor destes;
- *Java Server Faces* e *Richfaces*: Foram utilizadas essas tecnologias na camada de visualização devido ao rápido desenvolvimento oferecido por estas aplicações *Web* atuais. Outro fator importante é que o uso do *Richfaces* oferece criar aplicações RIA (*Rich Internet Application*), aplicações ricas na interface, em que o processamento da interface fica em grande parte para o cliente. Outra característica importante é a facilidade de uso de Ajax com estas tecnologias;
- XHTML (*eXtensible Hypertext Markup Language*): As páginas *Web* foram desenvolvidas utilizando-se esta linguagem. Sua principal característica é forte

padronização adquirida do XML ao desenvolvimento de páginas, aprimorando o HTML. Para algumas formatações de apresentação de conteúdo foi utilizado também CSS (*Cascading Style Sheet*);

- SVN: É um software de versionamento e controle de revisão. Utilizado muitas vezes para que os programadores mantenham cópias e um histórico de alterações do código fonte. O sistema consta com algumas revisões que foram disponibilizadas por SVN através desta ferramenta. Disponível em: <http://dbpediasparql.sourceforge.net/>.

### 3.6 INTEROPERABILIDADE DO SISTEMA COM DBPEDIA LOOKUP

Serviços *Web* são identificados por uma URI, descritos e definidos usando XML. Um dos motivos que tornam serviços *Web* atrativos é o fato deste modelo ser baseado em tecnologias padrões, tais como XML e HTTP. Eles são usados para disponibilizar serviços interativos na Web, que podem ser acessados por outras operações. SOAP (*Simple Object Access Protocol*) é um padrão W3C para a troca de mensagens entre aplicações e serviços, já que é uma tecnologia construída com base em XML e HTTP.

Há uma grande quantidade de palavras-chaves que podem ser usadas para identificar os recursos da Dbpedia. No entanto, determinar URI's a partir de palavras-chaves que possam trazer alguma informação da base de dados pode trazer algumas dificuldades, ressaltadas pelos termos de busca nesta aplicação serem referentes apenas a palavras contidas no valor da propriedade **rdfs:label** dos recursos.

Para melhorar esta lacuna foi usada a integração do sistema com o *DBpedia Lookup*. Um serviço web que encontra a mais provável URI para uma determinada palavra-chave. Segundo seu criador, George Kobarov, seu algoritmo enumera os recursos *DBpedia* de acordo a sua relevância na *Wikipedia*. Este serviço oferece a chamada de dois métodos:

- **keywordSearch**: método em que se retornam entidades que contêm todas as palavras passadas como argumento.

- **prefixSearch**: método para retornar uma lista de entidades que começam com determinada sequência de caracteres.

O método utilizado para o desenvolvimento da aplicação foi o *prefixSearch*, já que foi preciso fornecer ao usuário sugestões em tempo real do que aquele digita até o momento.

Depois de configurar a aplicação para operar com o serviço web, seu acesso através da interface assemelha-se a este:

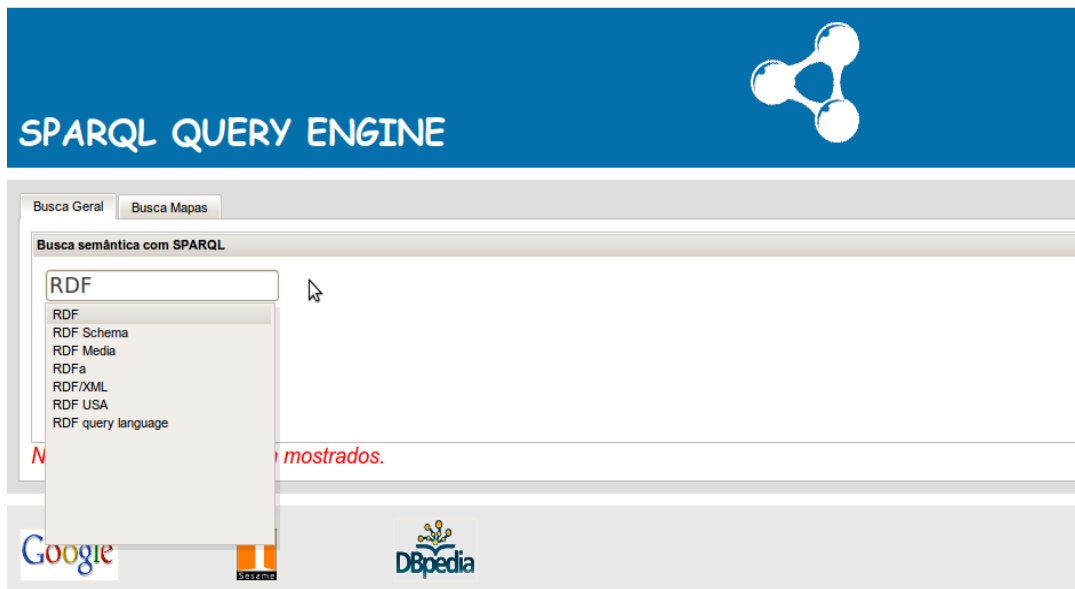


Fig. 12: Sugestões de termos com *DBpedia URI Lookup*.

Neste exemplo percebe-se que sugestões de busca são mostradas de forma explicitamente relevantes ao conteúdo digitado.

### 3.7 INTEROPERABILIDADE COM FLICKR SERVICE

Um outro serviço *Web* incorporado a aplicação foi o *Flickr Service*, *Flickr*® é um sítio que hospeda e compartilha imagens fotográficas, desenhos e ilustrações. Uma característica interessante é que *Flickr*® utiliza de marcações para poder categorizar as fotos de seus usuário.

No desenvolvimento da aplicação na parte da pesquisa que devolve coordenadas geográficas, foi realizada a opção de recuperação de imagens que retratam lugares através de informações extraídas primeiramente de pesquisas

feitas na *DBpedia* e posteriormente estas respostas são encapsuladas e com elas são feitas requisições a base de dados do Flickr® sobre o padrão REST (*Representational State Transfer*) que, desta vez, devolve em XML tags que descrevem propriedades das imagens. As únicas informações tratadas são aquelas que estão associadas a URL da imagem e o álbum de imagens a qual pertence.

A recuperação de imagens se dá baseada em valores dos predicados **rdfs:label**, **geo:lat**, **geo:long**. É formulada uma requisição em que se leva em conta estes três parâmetros nos quais os dois últimos são utilizados para construir uma área mínima em que fotos geo-referenciadas se encaixem dentro desta área. O rótulo também é utilizado como parâmetro opcional para refinar ainda mais a pesquisa.

As imagens utilizadas foram apenas aquelas que são consideradas públicas pelos seus usuários. Elas são ordenadas pela distância em que estão da área geográfica e são logo em seguida tratadas pelo critério de relevância quanto ao texto atribuído na consulta em relação a descrições e títulos das imagens e por quaisquer outros parâmetros que possam ser passados. Atribuiu-se a quantidade máxima de vinte imagens como quantidade máxima de resposta.

Isto leva a uma visão de como os dados podem ser extraídos da Web Semântica e utilizados posteriormente não só para visualização do usuário, mas também como informações a serem processadas por outras aplicações para gerar novas informações.

### **3.8 PESQUISA TEXTUAL E TÉCNICA DE ENUMERAÇÃO DOS RESULTADOS**

Para trabalhar com pesquisa textual em valores das propriedades RDF. Primeiramente foi necessário levantar algumas observações:

- Qual predicado dentre os formulados em grafo RDF são considerados relevantes para uma pesquisa textual e podem melhor definir um termo?
- A influência em tempo de resposta das requisições serão afetadas pelo tamanho e quantidade das literais a serem envolvidas na pesquisa?
- Qual a quantidade de resultados dentro da base da ontologia pode ser devolvida de forma a não sobrecarregar consultas e servidores?



- Depois de escolhido o predicado há outros métodos que podem ajudar a enumerar os resultados mais relevantes?

Para o desenvolvimento do sistema tais questionamentos deveriam ser estudados, avaliados e testados o quanto possível.

A resposta da primeira indagação veio da observação de alguns serviços de busca baseados sobre dados armazenados sobre a mesma versão do servidor, neste caso algumas implementações baseiam-se na busca por rótulos (*label*) dos recursos, que pode ser uma propriedade envolvida com tripla de um grafo RDF. Outras implementações fazem busca em vários predicados com objetos textuais, mas exigem um consumo de hardware por parte do servidor de dados muito maior.

Respostas de consultas SPARQL são geralmente interrompidas devido ao fato de *endpoints* públicos permitirem um tempo bastante curto na aplicação desenvolvida. Consultas que ocupem o processamento por mais de 1000 milissegundos na base DBpedia não são processadas, desta forma valores literais muito grandes como objetos do predicado **dbprop:abstract** podem ser facilmente interrompidos a depender da complexidade da consulta e sobrecarga da rede.

Um conjunto de testes foi executado para saber quantas respostas podem ser requisitadas por vez ao servidor. Depois, e de acordo ao nível de elaboração da consulta, percebeu-se que 10 resultados por requisições foi a média estabelecida para cada consulta sem extrapolar tempos limites e trazer respostas satisfatórias.

SPARQL ainda não provê um método padrão para pesquisa textual em dados RDF e tão pouco otimizada, por isso foi utilizada funções extensão de SPARQL providas pelo *OpenLinkVirtuoso*. Essas funções podem ser inseridas dentro de um consulta SPARQL em que implementações específicas de compiladores SPARQL as processam junto com o código da consulta. Será explicada uma visão geral do uso destas funções.

A função **bif:contains** olha por frases ou palavras separadas, independente da ordem e pode normalizar palavras que usam caracteres fora do padrão Unicode. A função usa dois argumentos: Uma variável que representa objeto de triplas RDF (ou uma variável que represente um grupo de objetos) em que procura-se a palavra ou os termos a serem pesquisados. Esta função atribui certa pontuação a cada ocorrência de palavras no objeto, recebendo maior pontuação o quão mais à esquerda aparecem do mesmo. As palavras devem ser formatadas de forma que estejam vinculadas sobre uma sequencia lógica de operadores *AND* ou *OR*, isto

implica em dizer que, ou se procura por todas as palavras-chaves ou por pelo menos uma delas. Quando as palavras estão sobre operadores *OR*, estas recebem valores mais baixos, podem trazer mais resultados, porém com menor significância. Neste projeto foram utilizados apenas operadores *AND*, que melhor filtraram os resultados.

Com isso pode-se atribuir este valor a uma variável SPARQL para que, por exemplo, possam ser ordenadas a partir da que contenha maior pontuação. Um ponto mínimo de corte pode ser estabelecido para filtrar dados que não atingissem a média, mas como depende da base de dados estabelecer um padrão para cada ocorrência de palavras preferiu-se não adotar esta parte da técnica.

Para agregar mais confiabilidade a possíveis respostas, fez-se a inclusão de índices de relevância das URI junto às consultas. A base de dados periodicamente atualiza o índice de relevância das entidades, cada URI tem maior pontuação a depender da quantidade de itens em que ela é referenciada (sujeito ou objeto) de uma URI. Desta maneira, o próprio servidor, através de outra função ***sum\_rank***, *efetua* o cálculo levando em conta estes dois valores apresentados acima e os argumentos da função ***bif:contains***. Esta técnica é sugerida em Erling (2009), que trata em parte da combinação de pesquisa textual com RDF sobre um banco de dados *OpenLinkVirtuoso*, pois implementações semelhantes são utilizadas em outros serviços de busca que usam a mesma ou semelhante base de dados.

## 4 APRESENTAÇÃO DO SPARQL QUERY ENGINE

Neste capítulo serão apresentadas as principais telas do SQE. Cada tela terá seu comportamento explicado de forma a exemplificar os relacionamentos envolvidos, focando principalmente nas propriedades da ontologia envolvidas e na visão em RDF das respostas. Também será apresentada considerações sobre o desenvolvimento deste trabalho sobre a Web Semântica.

### 4.1 INTERFACE DO SQE

A figura 13 mostra a tela inicial do sistema em que o usuário pode inicialmente interagir com o sistema. É carregada inicialmente a aba chamada de “Busca Geral”, nesta parte pode-se fazer a pesquisa de uma maneira mais abrangente sobre os dados na ontologia. As informações necessárias para a pesquisa são simplesmente passar as palavras-chaves juntamente com o idioma em que se deve buscar a resposta.

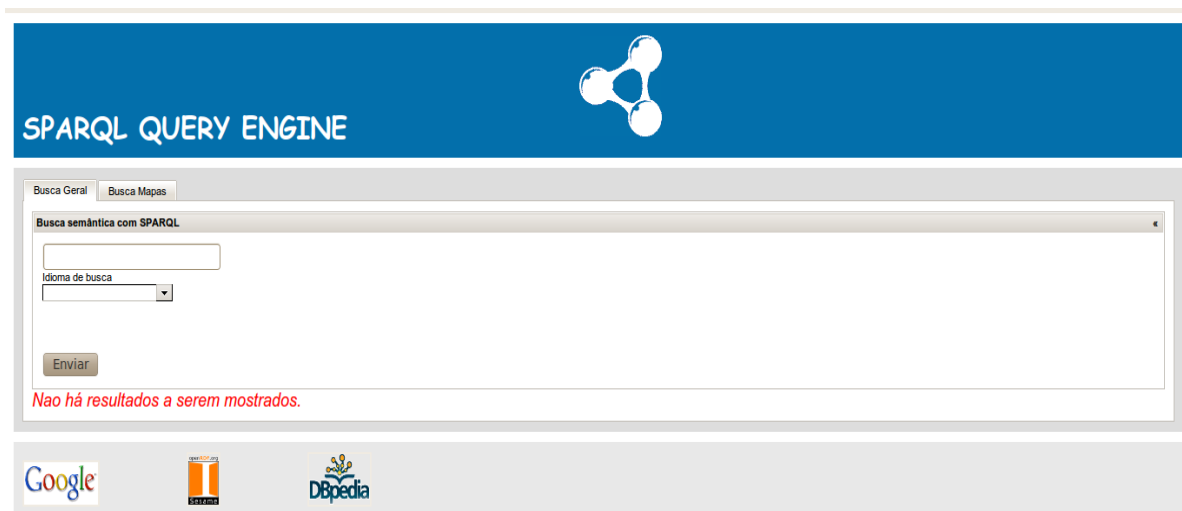


Fig. 13: Tela inicial do sistema.

Um resultado exibido é a geração das propriedades referentes a um recurso envolvido no grafo da figura 8, a URL a qual se refere ao termo “Recurso DBpedia” é a URI do nodo central do grafo. “Refere-se” é o valor da propriedade **rdfs:label**,

“sobre” tem o valor da propriedade **dbprop:abstract**. Os predicados opcionais são incluídos logo após, “*Homepage*” representando **foaf:homepage**, “Página Wikipedia” é o valor objeto **foaf:page** e por último “Imagem” para **foaf:depiction**.

As figuras 14, 15 e 16 mostram todas as guias de navegação geradas da consulta com os seguintes termos: “euclides”, “da”, “cunha”. Tais termos foram escolhidos arbitrariamente e digitados em diferentes sequências, mas produzindo sempre o mesmo resultado. Usando a técnica de classificação demonstrada no capítulo anterior, obteve-se:

The screenshot shows the SPARQL Query Engine interface. At the top, there is a blue header with the text "SPARQL QUERY ENGINE" and a logo. Below the header, there are two tabs: "Busca Geral" and "Busca Mapas". The main content area is titled "Busca semântica com SPARQL" and displays three search results. The first result is for "Euclides da Cunha" with the following details:


- Recurso DBPedia:** [http://dbpedia.org/resource/Euclides\\_da\\_Cunha](http://dbpedia.org/resource/Euclides_da_Cunha)
- Refere-se a:** **Euclides da Cunha**
- Sobre:** Euclides Rodrigues Pimenta da Cunha foi um escritor, sociólogo, repórter jornalístico, historiador, geógrafo, poeta e engenheiro brasileiro.
- Homepage:** [pagina nao encontrada](#)
- Página Wikipedia:** [http://en.wikipedia.org/wiki/Euclides\\_da\\_Cunha](http://en.wikipedia.org/wiki/Euclides_da_Cunha)
- Imagem:** 

Fig. 14: Resultado de consulta do sistema (Primeira aba).

The screenshot shows the SPARQL Query Engine interface. At the top, there is a blue header with the text "SPARQL QUERY ENGINE" and a logo. Below the header, there are two tabs: "Busca Geral" and "Busca Mapas". The main content area is titled "Busca semântica com SPARQL" and displays three search results. The second result is for "Euclides da Cunha (Bahia)" with the following details:

- Recurso DBPedia:** [http://dbpedia.org/resource/Euclides\\_da\\_Cunha%2C\\_Bahia](http://dbpedia.org/resource/Euclides_da_Cunha%2C_Bahia)
- Refere-se a:** **Euclides da Cunha (Bahia)**
- Sobre:** Euclides da Cunha é um município brasileiro do estado da Bahia. Localiza-se a uma latitude 10°00'27" sul e a uma longitude 39°00'57" oeste, estando a uma altitude de 472 metros. Sua população estimada em 2004 era de 54.949 habitantes. Possui uma área de 2.324,965 km².
- Homepage:** [pagina nao encontrada](#)
- Página Wikipedia:** [http://en.wikipedia.org/wiki/Euclides\\_da\\_Cunha%2C\\_Bahia](http://en.wikipedia.org/wiki/Euclides_da_Cunha%2C_Bahia)
- Imagem:**

At the bottom of the interface, there is a search bar with the text "Localizar: llabel" and several navigation buttons: "Anterior", "Próxima", "Realçar tudo", "Diferenciar maiúsculas/minúsculas", and "Fim da página atingido, continuando do início".

Fig. 15: Resultado de consulta do sistema (Segunda aba).

The screenshot shows the SPARQL Query Engine interface. At the top, there is a blue header with the text "SPARQL QUERY ENGINE" and a logo. Below the header, there are tabs for "Busca Geral" and "Busca Mapas". The main search area contains a text input field with "Euclides da Cunha" and a dropdown menu for "Idioma de busca" set to "português". An "Enviar" button is below the input field. Below the search area, there are three resource URIs: [http://dbpedia.org/resource/Euclides\\_da\\_Cunha](http://dbpedia.org/resource/Euclides_da_Cunha), [http://dbpedia.org/resource/Euclides\\_da\\_Cunha%2C\\_Bahia](http://dbpedia.org/resource/Euclides_da_Cunha%2C_Bahia), and [http://dbpedia.org/resource/Euclides\\_da\\_Cunha\\_Paulista](http://dbpedia.org/resource/Euclides_da_Cunha_Paulista). The main content area displays the following information:

- Recurso DBpedia:** [http://dbpedia.org/resource/Euclides\\_da\\_Cunha\\_Paulista](http://dbpedia.org/resource/Euclides_da_Cunha_Paulista)
- Refere-se a:** **Euclides da Cunha Paulista**
- Sobre:** Euclides da Cunha Paulista é um município brasileiro do estado de São Paulo. Localiza-se a uma latitude 22°33'41" sul e a uma longitude 52°35'29" oeste, estando a uma altitude de 265 metros. Sua população estimada em 2004 era de 10.547 habitantes. Possui uma área de 577,122 km².
- Homepage:** Página não encontrada
- Página Wikipedia:** [http://en.wikipedia.org/wiki/Euclides\\_da\\_Cunha\\_Paulista](http://en.wikipedia.org/wiki/Euclides_da_Cunha_Paulista)
- Imagem:**

At the bottom of the interface, there are logos for Google, a small logo, and DBpedia.

Fig. 16: Resultado de consulta do sistema (Terceira aba).

A figura 17 mostra um exemplo de como o SQE encontra URL's externas. Esta consulta corresponde ao grafo da figura 10. Neste exemplo, é recuperado URLs de sítios como *Youtube* e uma outra base semântica como *Freebase*.

The screenshot shows the SPARQL Query Engine interface with search results for "Chico Buarque". The main content area displays the following information:

- Recurso DBpedia:** [http://dbpedia.org/resource/Chico\\_Buarque](http://dbpedia.org/resource/Chico_Buarque)
- Refere-se a:** **Chico Buarque**
- Sobre:** Chico Buarque, nome artístico de Francisco Buarque de Hollanda, é um músico, dramaturgo e escritor brasileiro. Filho do historiador Sérgio Buarque de Hollanda, iniciou sua carreira na década de 1960, destacando-se em 1966, quando venceu, com a canção A Banda, o Festival de Música Popular Brasileira. Em 1969, com o crescente repressão da Ditadura Militar no Brasil, se auto-exilou na Itália, tornando-se, ao retornar, um dos artistas mais ativos na crítica política e na luta pela democratização do Brasil. Na carreira literária, foi ganhador do Prêmio Jabuti, pelo livro Budapeste, lançado em 2004. Casou-se com e separou-se da atriz Marieta Severo, com quem teve três filhas: Sílvia, que é atriz e casada com Chico Diaz, Helena, casada com o percussionista Carlinhos Brown e Luísa. É irmão das cantoras Miúcha, Ana de Hollanda e Cristina. Ao contrário do que tem sido propagado na internet, Aurélio Buarque era apenas um primo distante do pai de Chico.
- Homepage:** <http://www.chicobuarque.com.br>
- Página Wikipedia:** [http://en.wikipedia.org/wiki/Chico\\_Buarque](http://en.wikipedia.org/wiki/Chico_Buarque)
- Imagem:**

On the right side, there is a box titled "Referências encontradas" with a "Fechar" button. It contains a list of external URLs:

- <http://ztlgist.com/music/artist/3a5d1cc7-627e-48ea-aba3-fdd204782e33>
- <http://rdf.freebase.com/m/guid.9202a8e04000641f800000003b2be50>
- <http://www.youtube.com/watch?v=kMZJyafvcx8>
- <http://www.slipcue.com/music/brazil/buarque.html>
- [http://umbel.org/umbel/ne/wikipedia/Chico\\_Buarque](http://umbel.org/umbel/ne/wikipedia/Chico_Buarque)
- <http://www.chicobuarque.com.br>
- [http://www.youtube.com/watch?v=9A\\_JrsJF6mM](http://www.youtube.com/watch?v=9A_JrsJF6mM)

At the bottom of the interface, there are logos for Google, a small logo, and DBpedia.

Fig. 17: Busca por URL's.

A figura 18 mostra relação do URI resultante de uma pesquisa em outras propriedades da ontologia, ligando a termos que são relacionados a outras URI's ou

valores literais, essa parte do sistema permite que o usuário descubra ainda mais informações relacionadas ao resultado. São exibidas sempre 3 colunas por linha, a coluna do meio é sempre a propriedade que liga o resultado obtido na tela principal do sistema a outras URL's ou valores literais que estarão na primeira ou última coluna. Sempre haverá um espaço em branco em cada linha que seria a URI do resultado já obtida anteriormente.

Neste exemplo é possível obter algumas asserções dentre outras sobre Chico Buarque como:

- É o compositor musical de Bye Bye Brasil;
- É uma pessoa que vive no Rio de Janeiro;
- Cantor de samba;
- Cantor da Bossa Nova.

The screenshot shows a web interface for a SPARQL search. The search term 'Chico Buarque' is entered in the search box. The results are displayed in a table with three columns: 'Sujeito', 'Predicado', and 'Objeto'. The table contains various RDF triples related to Chico Buarque, such as his role as a composer, his location, and his relationships with other people and works.

Sujeito	Predicado	Objeto
<a href="http://dbpedia.org/resource/Chico_Buarque">http://dbpedia.org/resource/Chico_Buarque</a>	dbprop:disambiguates	
<a href="http://dbpedia.org/resource/Chico_Buarque">http://dbpedia.org/resource/Chico_Buarque</a>	dbprop:disambiguates	
<a href="http://dbpedia.org/resource/Maria_Beth_%C3%A2nia:_Music_is_Perfume">http://dbpedia.org/resource/Maria_Beth_%C3%A2nia:_Music_is_Perfume</a>	dbprop-owl:ontology:starring	
<a href="http://dbpedia.org/resource/Francisco_Buarque_de_Holanda">http://dbpedia.org/resource/Francisco_Buarque_de_Holanda</a>	dbprop:redirect	
<a href="http://dbpedia.org/resource/Francisco_Buarque_de_Holanda">http://dbpedia.org/resource/Francisco_Buarque_de_Holanda</a>	dbprop:redirect	
<a href="http://dbpedia.org/resource/Francisco_Buarque_de_Holanda">http://dbpedia.org/resource/Francisco_Buarque_de_Holanda</a>	dbprop:redirect	
<a href="http://dbpedia.org/resource/Bye_Bye_Brasil">http://dbpedia.org/resource/Bye_Bye_Brasil</a>	dbprop-owl:ontology:musicComposer	
<a href="http://dbpedia.org/resource/Four_Days_in_September">http://dbpedia.org/resource/Four_Days_in_September</a>	dbprop-owl:ontology:musicComposer	
<a href="http://dbpedia.org/resource/Dona_Flor_and_Her_Two_Husbands">http://dbpedia.org/resource/Dona_Flor_and_Her_Two_Husbands</a>	dbprop-owl:ontology:musicComposer	
<a href="http://en.wikipedia.org/wiki/Chico_Buarque">http://en.wikipedia.org/wiki/Chico_Buarque</a>	foaf:primaryTopic	
	rdf:type	<a href="http://xmlns.com/foaf0.1/Person">http://xmlns.com/foaf0.1/Person</a>
	rdf:type	<a href="http://dbpedia.org/class/yago/BrazilianSingers">http://dbpedia.org/class/yago/BrazilianSingers</a>
	rdf:type	<a href="http://dbpedia.org/class/yago/BrazilianWriters">http://dbpedia.org/class/yago/BrazilianWriters</a>
	rdf:type	<a href="http://dbpedia.org/class/yago/BrazilianMusicians">http://dbpedia.org/class/yago/BrazilianMusicians</a>
	rdf:type	<a href="http://www.w3.org/2002/07/owl#Thing">http://www.w3.org/2002/07/owl#Thing</a>
	rdf:type	<a href="http://dbpedia.org/class/yago/Writer110794014">http://dbpedia.org/class/yago/Writer110794014</a>
	rdf:type	<a href="http://dbpedia.org/class/yago/Artist109812338">http://dbpedia.org/class/yago/Artist109812338</a>
	rdf:type	<a href="http://dbpedia.org/class/yago/Dramatist110030277">http://dbpedia.org/class/yago/Dramatist110030277</a>
	rdf:type	<a href="http://dbpedia.org/class/yago/LivingPeople">http://dbpedia.org/class/yago/LivingPeople</a>
	rdf:type	<a href="http://dbpedia.org/class/yago/Singer110599806">http://dbpedia.org/class/yago/Singer110599806</a>
	rdf:type	<a href="http://dbpedia.org/class/yago/Songwriter110624540">http://dbpedia.org/class/yago/Songwriter110624540</a>
	rdf:type	<a href="http://dbpedia.org/class/yago/PeopleFromRioDeJaneiro(city)">http://dbpedia.org/class/yago/PeopleFromRioDeJaneiro(city)</a>
	rdf:type	<a href="http://dbpedia.org/class/yago/SambaMusicians">http://dbpedia.org/class/yago/SambaMusicians</a>
	rdf:type	<a href="http://dbpedia.org/class/yago/BrazilianMaleSingers">http://dbpedia.org/class/yago/BrazilianMaleSingers</a>
	rdf:type	<a href="http://dbpedia.org/class/yago/BossaNovaMusicians">http://dbpedia.org/class/yago/BossaNovaMusicians</a>
	rdf:type	<a href="http://dbpedia.org/ontology/Person">http://dbpedia.org/ontology/Person</a>

Fig. 18: Busca por triplas.

A figura 19 exibe a aba responsável pela pesquisa em domínio específico, inicialmente sem nenhuma resposta. O preenchimento dos dados é exatamente o mesmo que ocorre para a figura 4.1. A diferença das abas é o resultado utilizando visualização no *Google Maps* incorporado na página.

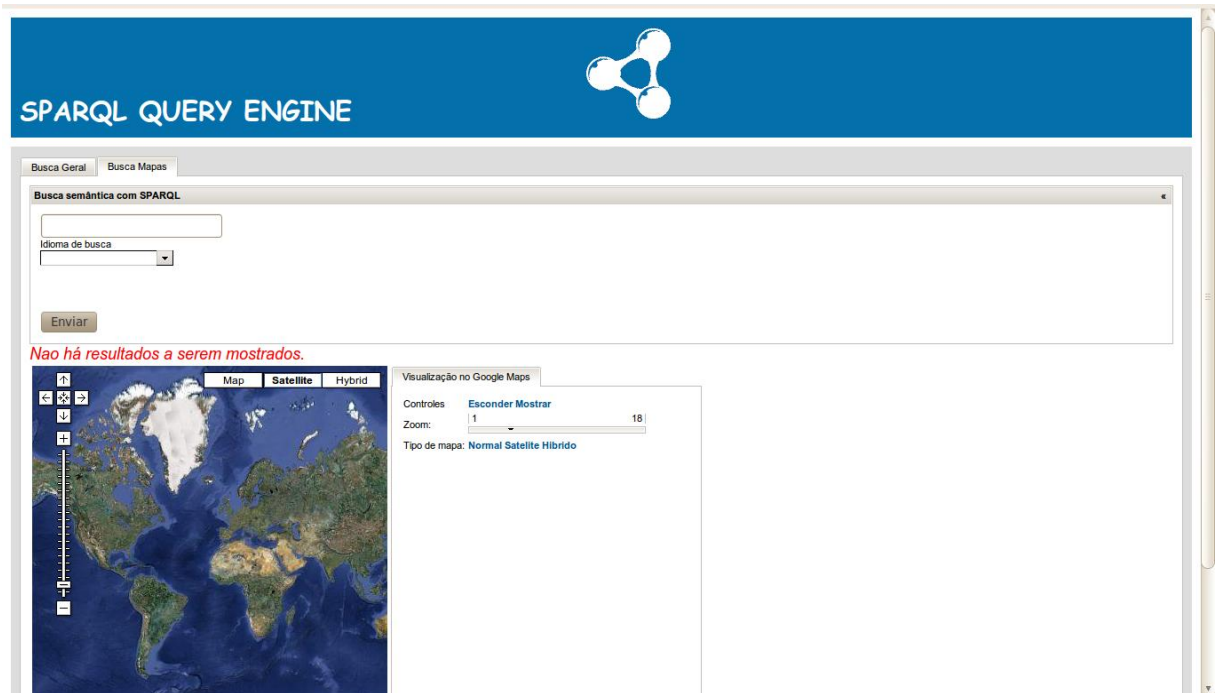


Fig. 19: Tela inicial do sistema (Busca por localidades)

A figura 20 exibe um resultado gerado para as palavras: “Cristo” e “Redentor”. O resultado visualmente é agora a representação do grafo da figura 9. A figura permite observar a adição de novos termos em função dos valores dos predicados **geo:lat** e **geo:long**. Com estes novos dados o sistema adquire maior interatividade sobre os resultados. O link de visualização no mapa simplesmente atribui esses valores a uma requisição e faz com que o mapa seja renderizado com os novos atributos mostrando a localização geográfica do ponto.

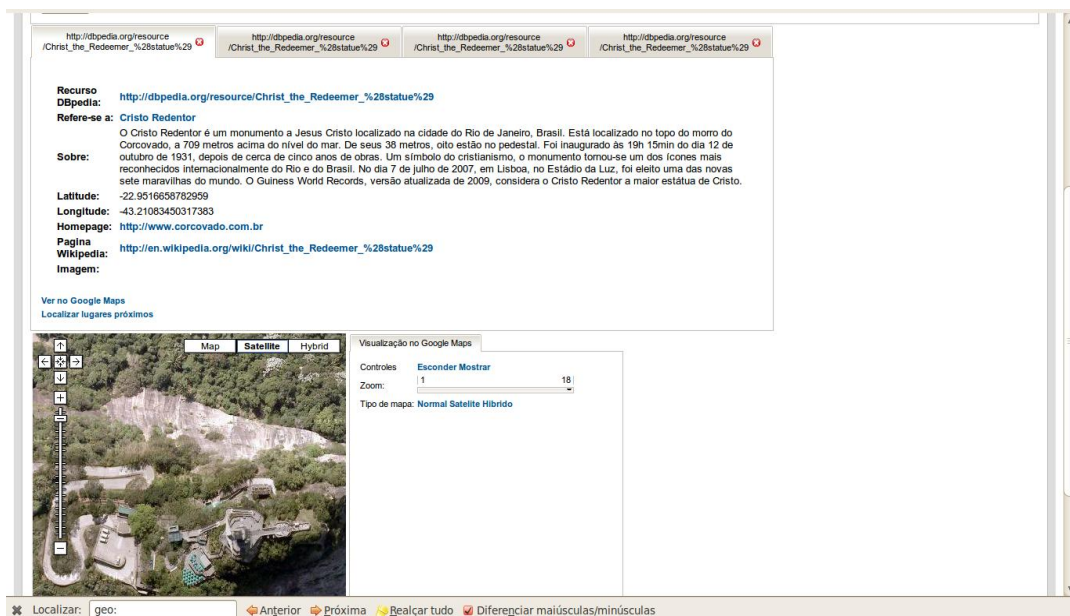
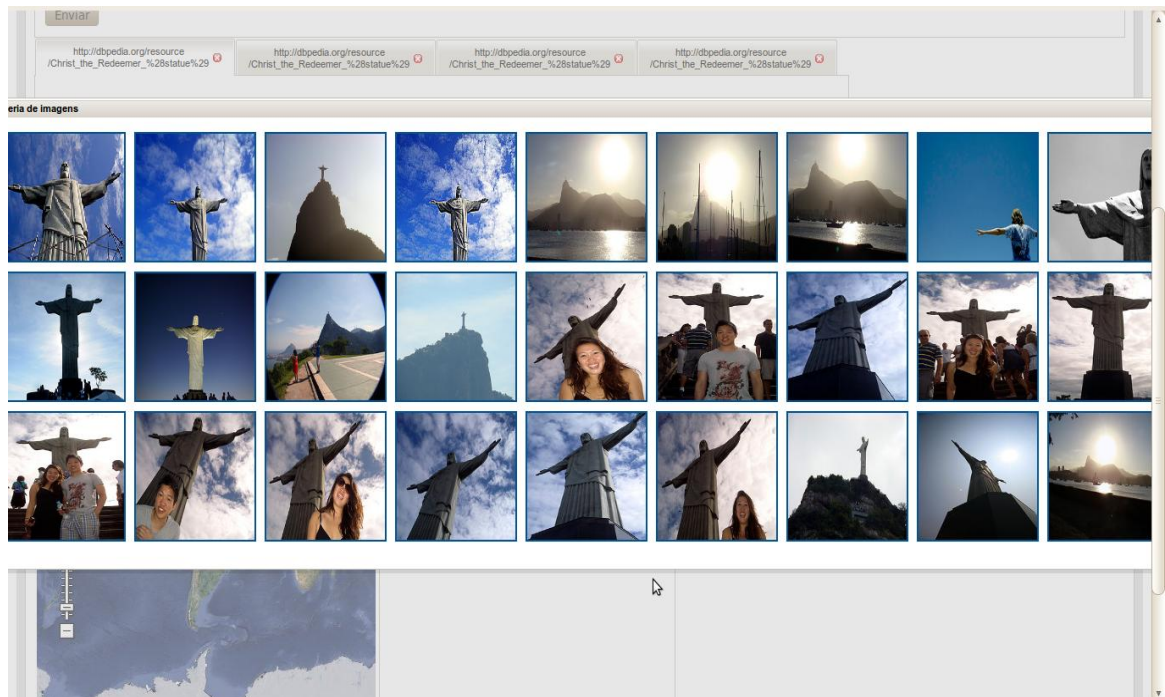


Fig. 20: Resultado para lugares com coordenadas geográficas.

A figura 21 exibe uma busca geo-referenciada por imagens, a galeria de imagens apresentada é construída do resultado apresentado na imagem 21.



**Fig. 21:** Exemplo de resultado para busca por imagens



## 5 CONCLUSÃO

### 5.1 CONSIDERAÇÕES INICIAIS

Neste trabalho foram apresentados os principais conceitos relacionados a Web Semântica, e como estes podem ser utilizados para recuperar informações.

Foi desenvolvida uma aplicação de busca textual utilizando muitos dos padrões propostos pela W3C para Web Semântica, que ao mesmo tempo combinou técnicas comuns em outros serviços de busca. O SQE teve um escopo razoavelmente amplo de busca considerando a grande quantidade de informações disponíveis na base de dados, contudo a ferramenta buscou reduzir a quantidade de resultados trazidos ao usuário.

A utilização dos princípios e conceitos da Web Semântica no desenvolvimento da aplicação reduziu consideravelmente o tempo de produção do software, pois permite aos desenvolvedores focarem que informações devem ser lidas pelo usuário quando tratam de pesquisas na *Web*.

Outro fator importante foi descobrir novas informações a partir de resultados já obtidos e utilizar estes mesmos dados para trabalharem com outras bases de dados.

Por utilizar padrões comuns hoje utilizados por aplicações semânticas, este sistema poderia ser implementado com pequenas mudanças para diversos outros fins. Idiomas, por exemplo, poderiam ser facilmente adicionados e outras variáveis de informação poderiam ser visualizadas com pouquíssima mudança no código da aplicação.

É muito importante citar que durante o desenvolvimento deste trabalho foi possível pensar de maneira diferente da habitual na construção de sistemas *Web*. Apesar deste projeto ter focado em uma ferramenta de busca, pode-se expandi-las para muitos outros domínios, construção de reutilização de ontologias e base de dados para interesses mais específicos. Deve-se ser citado também que pensar nos sistemas como conjuntos de expressões em OWL e RDF possui características diferentes daquelas de orientação a objetos ou outros paradigmas, entretanto podem trabalhar conjuntamente. Contudo, a principal contribuição deste trabalho é mostrar

que podem ser construídos sistemas que utilizem os recursos e ferramentas disponíveis atualmente da Web Semântica.

O ideal de uma *Web* estruturada e totalmente ligada ainda está longe de se tornar uma realidade. Entretanto, esforços têm sido cada vez maiores nesse sentido e está se tornando uma realidade dentro da atual *Web 2.0*.

## 5.2 DIFICULDADES ENCONTRADAS

Ao decorrer do trabalho algumas dificuldades foram encontradas, a principal foi que os dados da ontologia da *DBpedia* não estarem ainda totalmente estruturados. Por ser uma base aberta, algumas informações possuem um certo nível de dados ainda incompletos, portanto algumas informações existem para um tipo de idioma e por vezes faltam em outro (informações literais, com anotação semântica de idiomas), influenciando por vezes nas consultas.

Um outro problema é que, ao se trabalhar com esta base de dados, as consultas devem ser bem estruturadas e formuladas para que sejam ao mínimo impactadas pela restrição do servidor. De modo que as consultas foram elaboradas levando em consideração este fator.

Importante ser citado é que, apesar de SPARQL ser para a Web Semântica a principal linguagem para comunicação entre os dados interligados, ainda falta-lhe implementações que permitam usar mais eficientemente busca textuais em documentos RDF. Embora extensões da linguagem venham a amenizar este problema.

## 5.3 TRABALHOS FUTUROS

Em trabalhos futuros o sistema pode ser melhorado para que busque mais informações em outras bases de dados, já que a maioria delas utilizam *endpoints* para acesso, uma possibilidade é utilizar outras bases de dados da Web Semântica para receber mais informações geográficas de recursos que falem sobre isto na

*DBpedia*. Outra melhoria significativa é pesquisar termos de buscas em mais literais que não sejam somente rótulos de recursos.

## REFERÊNCIAS

ANTONIOU, G.; HARMELEN, F. V. **A Semantic Web Primer**. 2. ed. Londres: The Mit Press, 2008.

BORST, W. N. **Construction of Engineering Ontologies for Knowledge Sharing and Reuse**: Tese (Pós-Doutorado), Universidade de Twente, 1997.

BERNERS-LEE, T. **Linked Data**. Disponível em: <<http://www.w3.org/DesignIssues/LinkedData.html>>. Acesso em: 20 de out. 2010.

BERNERS-LEE, T.; LASSILA, O.; HANDLER, J. **The Semantic Web**: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, Nova Iorque, p.22-43, 01 maio 2001.

DIAS, T.; SANTOS, N. **Web Semântica**: Conceitos Básicos e Tecnologias Associadas. Disponível em: <<http://magnum.ime.uerj.br/cadernos/cadinf/vol14/7-neide.pdf>>. Acesso em: 11 jul. 2010.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 4. ed. Rio de Janeiro: Atlas, 2002.

HEATH, T. **Linked Data – Connect Distributed Data Across the Web**. Disponível em: <<http://linkeddata.org/>>. Acesso em: 25 nov. 2010.

HEBELER, T.; et. al. **Semantic Web Programming**. Estados Unidos: Wiley, 2009. p. 585.

O' REILLY, Tim . **What Is Web 2.0**. Disponível em: <<http://oreilly.com/web2/archive/what-is-web-20.html>>. Acesso em: 25 nov. 2010.

SEGARAN, T.; EVANS, C.; TAYLOR, J. **Programming the Web Semantic**. Estados Unidos: O'Reilly, 2009. 271 p.

PRAZERES, Cássio Vinícius. **Serviços Web Semânticos**: da modelagem à composição. 2009. 389 f. Tese (Doutorado) - USP, São Carlos, 2009.

SWARTZ, A. **The Semantic in Breadth**. Disponível em: <<http://logicerror.com/semanticWeb-long>>. Acesso em: 19 abr. 2010.

TRUCOLO, F.; COELLO, J. Recuperação de informação na Web Semântica. In: XIII ENCONTRO DE INICIAÇÃO CIENTÍFICA. **Anais...** Campinas: Pontifícia Universidade Católica. 2001.

W3C. **Semantic Web**. Disponível em: <<http://www.w3.org/standards/semanticweb/>>. Acesso em: 02 abr. 2010.

W3C. **RDF Syntax**. Disponível em: <<http://www.w3.org/1999/02/22-rdf-syntax-ns>>. Acesso em: 06 abr. 2010.

YU, Lyang.. **Introduction to the Semantic Web and Semantic Web Services**. Estados Unidos: Chapman & Hall/CRC, 2007. 341p.