Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

Criação de um Repositório de Dados Ligados para Filtragem de *Hoax*

Adriano Rodrigues Delvoux Mattos

JUIZ DE FORA
OUTUBRO, 2012

Criação de um Repositório de Dados Ligados para Filtragem de *Hoax*

Adriano Rodrigues Delvoux Mattos

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Orientador: Jairo Francisco de Souza
Co-orientador: Pablo Mendes

JUIZ DE FORA
OUTUBRO, 2012

Criação de um Repositório de Dados Ligados para Filtragem de *Hoax*

Adriano Rodrigues Delvoux Mattos

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Jairo Francisco de Souza
<<Título do Orientador>>

Alessandreia Marta de Oliveira
<<Título do Examinador 1>>

Eduardo Barrere
<<Título do Examinador 2>>

JUIZ DE FORA

22 DE OUTUBRO, 2012

Aos meus amigos e irmã.

Aos pais, pelo apoio e sustento.

Resumo

Ao oferecer informações na *Web* é importante representá-las em um formato padrão de forma que seres humanos e máquinas possam fazer uso destes dados. Este projeto exemplifica o uso de dados ligados para fornecer um mecanismo de representação e consumo de informações. Esta tecnologia visa a criação de centros de dados para vários domínios que podem interagir através de ligações entre diferentes entidades na *Web*. Desta forma surge um imenso grafo onde é possível realizar consultas detalhadas. Utilizando esta abordagem torna-se possível a criação de aplicações mais inteligentes que podem consumir e analisar estes dados. A proposta deste projeto consiste em reduzir a circulação de *spams* em serviços de e-mail através da criação de um *dataset* de dados capaz de oferecer recursos para que uma aplicação possa analisá-los. Devido à diversidade de domínios em que estes e-mails estão enquadrados será necessário limitar a análise aos e-mails relacionados ao domínio de crianças desaparecidas. Desta forma, o projeto também ajudará diversas famílias que perderam um familiar evitando que a circulação de spams atrapalhe os verdadeiros e-mails.

Palavras-chave: Dados Ligados. Filtro de mensagens. Dados abertos. *Hoax*.

Abstract

This project shows the use of linked data to provide a way to represent and consume information. This technology aims to create a knowledge base from multiple interlinked domains allowing to perform complex queries. By using linked data it is possible to create intelligent systems for consuming and analyzing semantic information. A tool was created to mark as spam messages about unmissing people on Facebook, using a database that implements the linked data principles.

Keywords: Linked data, Filter messages, Open data, *Hoax*.

Agradecimentos

A todos os meus parentes e amigos pelo encorajamento e apoio.

Ao professor Jairo Souza pela orientação, amizade e principalmente, pela paciência, sem a qual este trabalho não se realizaria.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o nosso enriquecimento pessoal e profissional.

"Lembra que o sono é sagrado e alimenta de horizontes o tempo acordado de viver".

Beto Guedes (Amor de Índio)

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

DCC	Departamento de Ciência da Computação
ONG	Organização Não Governamental
UFJF	Universidade Federal de Juiz de Fora
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RDFa	Resource Description Framework in attributes
HTML	HyperText Markup Language
HTTPS	HyperText Transfer Protocol Secure
XML	Extensible Markup Language
LOD	Linking Open Data
SPARQL	SPARQL Protocol and RDF Query Language
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
W3C	World Wide Web Consortium
PHP	Hypertext Preprocessor
CSV	Comma-separated values
JSON	JavaScript Object Notation
MQL	Metaweb Query Language
API	Application programming interface
SDK	Software Development Kit
FOAF	Friend-of-a-Friend
DBPPROP	DBPedia Property

1 Introdução

A chegada da web 2.0 trouxe mais dinâmica para a disponibilização de conteúdo, possibilitando que qualquer usuário publique conteúdo de maneira simples, sem a necessidade de conhecimento avançado em informática. As ferramentas para criação de blogs estão se tornando cada vez mais práticas e acessíveis ao usuário leigo, de forma que as pessoas possam se preocupar mais com a qualidade da informação.

As redes sociais também se destacam entre os meios de geração de conteúdo, sendo um ambiente onde as pessoas postam informações variadas e compartilham com amigos, atingindo milhares de usuários. Junto às redes sociais o número de dispositivos conectados à internet cresceu muito ao longo dos anos. Hoje, existem notebooks, smartphones, tablets e até celulares mais simples com acesso a internet, prontos para que o usuário possa interagir nas redes sociais. Com todas essas tecnologias em mãos pode-se dizer que este perfil de usuário se torna o principal criador de conteúdo na web.

1.1 Motivação

Os usuários das redes sociais convivem com a circulação de inúmeras mensagens, espalhadas facilmente em um espaço curto de tempo. Estas mensagens contêm informações diversas que são repassadas entre amigos, podendo atingir até milhares de pessoas. Porém, nem todas estas informações devem ser encaradas com seriedade. Ultimamente, o número de mensagens falsas cresceu consideravelmente. Conhecidas como *hoax*, estas mensagens apresentam supostas campanhas filantrópicas e conteúdo de apelo dramático, com a intenção de sensibilizar as pessoas e incentivar o compartilhamento (Teixeira, 2007).

Um grande problema é identificar a veracidade destas mensagens. No caso de pessoas desaparecidas, existem muitos casos de *hoax* que atrapalham as verdadeiras mensagens que poderiam ajudar uma família. Também existe a possibilidade das informações circularem entre milhares de pessoas sendo que o indivíduo já tenha sido encontrado. Um

caso que exemplifica essa situação é a de um blog[1] criado com o objetivo de avisar sobre as falsas mensagens de uma criança desaparecida que circulava nos serviços de e-mail, figura 1.1. O blog informa que após encontrar a criança a família continuou recebendo informações falsas de seu paradeiro e trotes por telefone.

Figura 1.1: Blog Maria Cecilia Encontrada

A família não é a única lesada pelos *hoaxes*, a própria pessoa que compartilha esta informação também pode vir a ter complicações por este ato. Uma matéria encontrada no Jornal Hoje em Dia, "Cibermentira - Trotes infernizam internautas", explica sobre as consequências de se transmitir mensagens de origem duvidosa. Para exemplificar, a matéria cita um caso onde um indivíduos enviou um e-mail recebido para seus contatos acreditando que tratava-se de uma informação verídica. Ele repassou automaticamente levando junto ao e-mail seus dados de assinatura como nome e telefone. Este simples ato trouxe grandes problemas, tanto para sua vida pessoal como profissional, uma vez que seus dados circulavam por toda rede.

Mesmo existindo muitos sites que possuem inúmeras informações destas crianças os usuários não acessam estas fontes para verificar, e acabam ignorando ou repassando as mensagens. Mesmo os que buscam informações, não encontram nenhum repositório de dados. As informações se encontram espalhadas em diferentes sites através de iniciativas

[1] http://mariaceciliaencontrada.blogspot.com.br

locais, e em muitos casos duplicadas e desatualizadas.

Uma base de dados de pessoas desaparecidas poderia ajudar de forma significativa na busca de informações para verificar conteúdos suspeitos na *Web*.

1.2 Objetivos

Este projeto possui como objetivo geral contribuir com a criação de um banco de dados com informações de pessoas desaparecidas capaz de ajudar a identificar possíveis *hoaxes* evitando que usuários sejam enganados.

Como objetivo específico o projeto propõe a implementação de uma abordagem para identificação automática de conteúdo sobre pessoas desaparecidas em um ambiente muito utilizado atualmente, as redes sociais. Automatizando este processo, uma aplicação criada poderá evitar que usuários destes serviços repassem mensagens falsas para sua rede de amigos.

1.3 Metodologia

A *Web*, desde seu surgimento, passou por uma série de transformações que possibilitou a criação de um ambiente cada vez mais interativo para o usuário, seja este avançado ou leigo. Porém, nem sempre ela teve esta característica.

Em sua versão 1.0, a *Web* possuía a estruturação do conteúdo sob o domínio de desenvolvedores, visto que exigia maior conhecimento do usuário. A necessidade de criar um espaço com maior interatividade foi o próximo passo.

O surgimento de novas tecnologias e ferramentas facilitou o contato com usuários leigos, possibilitando estes contribuírem com conteúdo para a nova *Web*. A geração de conteúdo dinâmico e o compartilhamento de informações marcou a transição para a *Web* 2.0. Ferramentas como Blogs, Redes Socias, sites de compartilhamento de arquivos, vídeos, músicas passaram a ser utilizados amplamente. Apesar do avanço da *Web*, um problema que existe até os dias de hoje é a falta de padronização dos dados. Até então, a *Web* é estruturada como uma rede de recursos ligados entre si, com mínimo de informações deste relacionamento. O futuro da *Web* propõe tratar este problema for-

necendo mecanismos de forma que os dados possam ser manipulados abertamente por diferentes aplicações. Atualmente, vigora a *Web* 2.5.

A *Web* Semântica é uma visão que surgiu como suporte à nova *Web*, de forma que as informações possam ser dispostas tanto para humanos quanto para as máquinas. No caso das máquinas, teria grande importância para a criação de aplicações mais sofisticadas e inteligentes.

Como consequência de um movimento mundial para a padronização dos dados surgiu o conceito de Dados Ligados cujo objetivo é agregar informações relevantes aos recursos dispostos na *Web* e interligá-los, reestruturando a atual *Web* de documentos heterogêneos.

Ao observar o atual cenário dos dados de pessoas desaparecidas na *Web*, percebe-se que não existe nenhuma padronização para exibir tais informações. Aplicando o conceito de Dados Ligados é possível obter uma disponibilização uniforme destes dados através de uma base que será criada. Dessa forma tem-se um mecanismo que possibilita a utilização de dados por pessoas e aplicações na *Web*, ajudando a combater as *hoaxes* que circulam entre as redes sociais.

1.4 Organização

Para um melhor entendimento das tecnologias utilizadas no desenvolvimento do projeto, o Capítulo 2 dedica-se a explicar os conhecimentos básicos ligados a disponibilização de Dados Abertos e ao uso das práticas de Dados Ligados. O Capítulo 3 apresenta o cenário em que este projeto está incluído, além de mostrar a arquitetura da aplicação. O Capítulo 4 mostra os passos que foram necessários, desde a coleta de dados até a ferramenta desenvolvida. Por fim, o Capítulo 5 exibe as conclusões finais deste trabalho.

2 Conceitos relacionados

Este capítulo irá tratar sobre os conceitos básicos e necessários para o desenvolvimento do projeto. A seção 2.1 apresenta o conceito de Dados Abertos e os benefícios relacionados ao seu uso, bem como as tecnologias envolvidas para sua publicação. A seção 2.3 discorre sobre o princípio de Dados Ligados além de mostrar como podemos utilizá-los para fornecer informações abertas na *Web*.

2.1 Dados Abertos

O termo transparência pública não é designado ao simples fato de expor documentos na *Web*, está relacionado à ideia de fornecer informações legíveis por humanos e que possam ser manipuladas por máquinas automaticamente (Manual dos dados abertos: desenvolvedores, 2011).

2.1.1 Definição

De acordo com a Open Definition (2009) um conjunto de informações são classificadas como dados abertos se podem ser utilizadas livremente, reutilizadas para outros fins e redistribuídas, sendo exigido, no máximo, a atribuição da fonte.

Para uma definição mais detalhada, pode-se dizer que os dados abertos devem estar de acordo com os três itens a seguir (Manual dos dados abertos: governo, 2011):

- **Disponibilidade e acesso:** os dados devem estar disponíveis na íntegra e em um formato que facilite o acesso e que possa ser alterado.
- **Reuso e redistribuição:** as informações devem estar preparadas para serem reutilizadas e redistribuídas facilmente, além de garantir o cruzamento o cruzamento de diferentes conjuntos de dados.
- **Participação universal:** os dados devem estar livres de qualquer restrição de uso, permitindo a qualquer indivíduo redistribuir e reutilizar estas informações.

Ao mencionar sobre dados abertos é importante deixar claro o tipo de informação que se adequa a este princípio. Dados abertos não estão relacionados a dados pessoais mas sim a informações públicas, que devem estar abertas à sociedade, como os dados governamentais públicos, por exemplo Manual dos dados abertos: governo (2011).

Durante um encontro realizado em 2007, o *Open Government Working Group* (opengovdata.org, 2007) definiu alguns princípios a serem seguidos para que os dados possam ser considerados abertos. Estes princípios foram criados com base nos dados governamentais, mas podem ser aplicados a outros tipos de domínios:

- **Completos:** todos os dados públicos devem estar disponíveis e livres de qualquer restrição de privacidade, segurança ou privilégios.
- **Primários:** os dados devem ser publicados na forma como foram retirados da fonte, sem agregar informação ou alterá-los.
- **Atuais:** os dados devem ser publicados o mais rápido possível de forma que preserve o seu valor.
- **Acessíveis:** os dados devem estar disponíveis para uma grande quantidade de usuários de forma que possam ser utilizados nos mais diversos propósitos.
- **Processáveis por máquinas:** os dados devem estar disponíveis em um formato que possibilite o processamento automático por máquina.
- **Não discriminatórios:** os dados devem estar disponíveis para qualquer usuário sem a necessidade de se realizar qualquer registro.
- **Não proprietários:** os dados não devem estar sob o controle exclusivo de qualquer entidade.
- **Livres de licença:** os dados não devem estar sujeitos a qualquer copyright, marca registrada, ou regulações de segredo industrial.

Especialista em dados abertos, Eaves David (Eaves, 2009) propôs 3 leis referentes aos dados abertos governamentais, mas que podem ser aplicados também aos dados abertos em gerais, são elas:

- Se o dado não pode ser encontrado ou indexado ele não existe.
- Se ele não é oferecido em um formato aberto ou processável por máquina, não pode ser reaproveitado.
- Se ele não pode ser reaproveitado de forma legal, não é útil.

2.1.2 Benefícios

Os dados abertos são de grande importância na vida das pessoas. Uma vez que são informações públicas, podem ser acessados por qualquer indivíduo e utilizado para diversas situações, contribuindo para algumas decisões do dia-a-dia. No Manual de dados abertos (Manual dos dados abertos: governo, 2011) é possível verificar que em alguns países o impacto destes dados na vida da população é uma realidade. Na Holanda, um projeto disponibilizou um site que informa se a qualidade do ar está em um nível prejudicial. Outro projeto no Reino Unido criou um serviço capaz de ajudar usuário a encontrar um local para morar com base em diferentes características, como trajeto para o trabalho, preço do imóvel, entre outros parâmetros.

Além de contribuir na vida pessoal estes dados também podem ser aplicados amplamente no governo, aumentando a eficiência e reduzindo a redundância de informações.

Os dados abertos possibilitam o cruzamento de dados, facilitando assim a interoperabilidade. Este termo é designado à capacidade de diferentes sistemas comunicarem entre si, trabalhando com conjuntos diversos de dados. Esta interoperabilidade permite a criação de sistemas cada vez mais complexos capazes de combinar bases distintas e relacioná-las.

Com os dados publicados abertamente uma gama de aplicações podem ser criadas proporcionando uma melhor qualidade de vida para a população, transparência nos dados públicos e garantir melhorias econômicas através de ferramentas capazes de realizar planejamentos financeiro.

Existem diferentes áreas onde a disponibilidade de dados no formato aberto é importante, beneficiando um grande número de pessoas e possibilitando a inovação em diversos setores. A seguir encontram-se algumas áreas beneficiadas com o uso de dados

abertos:

- Transparência e controle democrático;
- Participação popular;
- Empoderamento dos cidadãos;
- Melhores ou novos produtos e serviços privados;
- Inovação;
- Melhora na eficiência de serviços governamentais;
- Melhora na efetividade de serviços governamentais;
- Medição do impacto das políticas;
- Conhecimento novo a partir da combinação de fontes de dados e padrões.

2.1.3 Tecnologias

A forma como os dados estão estruturados impacta em sua reutilização de maneira que, uma estrutura bem definida facilita a criação de ferramentas capazes de processar estas informações com mais confiança (Heath, 2007). O formato em que se encontra a maioria dos sites na atualidade, o HTML, foi criado para formatar a apresentação de textos e para estruturação de dados. Desta forma, tornasse complexo o papel de extração de conteúdo por aplicações, uma vez que os dados encontram-se espalhados no documento.

É importante definir o formato apropriado para a publicação dos dados. Informações que devem ser lidas por pessoas utilizam o formato (X)HTML. Para a distribuição de dados brutos tornasse necessário escolher um formato estruturado que facilite a manipulação por diferentes ferramentas e linguagens. O XML e o RDF por exemplo, podem ser acessados facilmente através de linguagens de consultas como o SPARQL e o XQuery, além de serem compatíveis com muitas linguagens de programação (Bennett and Harvey, 2009).

Formatos

A seguir, encontram-se alguns formatos mais utilizados para fornecer informações na *Web*:

- XML: O XML originou-se através da linguagem SGML, com o objetivo de se criar uma linguagem de marcação flexível capaz de ser processada facilmente por aplicações, distribuída pela internet e que fosse legível por seres humanos (Tittel, 2002). O XML não possui limitações para a criação de identificados, sendo estes criados pelo próprio desenvolvedor com base no tipo de informação a ser manipulada.

- JSON: O JSON é um formato baseado na linguagem de programação JavaScript. Trata-se de uma estrutura leve que facilita a análise por máquina, além de ser compreensível por seres humanos. É um formato de troca de informação muito utilizado por ser independente de linguagem e de rápido processamento (JSON.ORG, 1999).

- CSV: O CSV é um formato simples e compacto sendo assim muito utilizado para manipular um grande volume de dados que possuem uma mesma estrutura. Devido a sua simplicidade é necessário manter uma documentação com as características de cada coluna bem definida e respeitar a estrutura do documento para evitar que os dados sejam processados de forma errada (Manual dos dados abertos: governo, 2011).

Web Service

Web Service é um sistema criado com o objetivo de prover a interoperabilidade entre aplicações distintas fornecendo uma interface com dados em formato processável por máquina, o WSDL. Os sistemas também podem interagir com este serviço através de mensagens SOAP enviadas utilizando o protocolo HTTP em formato XML ou qualquer outro formato padrão (Booth et al, 2004).

REST é um tipo de *Web Service* que utiliza como abstração chave um recurso, que pode ser qualquer informação, oferecendo uma interface genérica para a manipulação de seus valores independente do tipo de aplicação que está fazendo a requisição.

2.2 Dados Ligados

2.2.1 Definição

De acordo com Berners-Lee (2006) a *Web* Semântica não se resume somente em colocar dados na *Web*. Sua proposta é realizar ligações entre os dados de forma que pessoas e máquinas possam reutilizá-los. Com os dados ligados entre si é possível, a partir de alguma informação, atingir outros dados relacionados.

Dados Ligados é definido por Bizer et al (2009) como uma forma de utilizar a *Web* para criar ligações entre os dados de acordo com seus tipos. Como estes dados estão publicados na *Web* podemos ter fontes de informações em bancos de dados externos em diferentes posições geográficas e legíveis por máquinas, uma vez que temos o significado dos dados explícito.

No caso da *Web* de documentos temos como unidades primárias documentos HTML com links para outros documentos. Para compor a *Web* de Dados Ligados, os recursos são representados através de um formato padrão, o RDF (Resource Description Framework), que permite interligar entidades de diferentes domínios (Bizer et al, 2009).

2.2.2 Princípios

Os quatro princípios criados por Berners-Lee (2006) que regem a publicação de dados utilizando a tecnologia de dados ligados na *Web*:

1. Utilize uma URI para identificar qualquer recurso;
2. Sempre use URIs HTTP para que seja possível encontrar estes nomes na *Web*;
3. Forneça os dados utilizando um formato padrão (RDF, SPARQL);
4. Crie ligações para outros recursos na *Web* de forma que seja possível encontrar mais informações.

Estes princípios fornecem um mecanismo para publicação e conexão entre dados usando a infra-estrutura da *Web* (Bizer et al, 2009).

2.2.3 Tecnologias

Esta sessão irá tratar sobre as tecnologias envolvidas no processo de publicação e busca de dados utilizando a tecnologia de dados ligados.

URIs

Quando queremos publicar dados na *Web* devemos primeiramente identificar o que queremos representar. Na *Web* todos os dados que possuem propriedades e relacionamentos são chamados de recursos (Cyganiak et al, 2007).

De acordo com Cyganiak et al (2007) os recursos podem ser classificados em informacionais e não-informacionais. As entidades presentes na *Web* tradicional tais como documentos, imagens e arquivos de mídia pertencem aos recursos informacionais. Por outro lado, as entidades como pessoas, lugares, produtos físicos, relacionadas a "objetos do mundo real"são classificadas como entidades não-informacionais.

Optou-se por utilizar URIs HTTP para identificar estes recursos. Primeiramente pois trata-se de uma forma simples de criar nomes únicos e globais na *Web*. Além disso o uso de URIs HTTP nos fornece um mecanismo de acesso às informações sobre um recurso na *Web*.

RDF

De acordo com Miller (2005) RDF é um modelo utilizado para descrever os recursos e suas propriedades. Para tornar possível o entendimento de máquinas o RDF utiliza triplas para representar relacionamentos entre as entidades.

As três partes que compõe esta tripla são conhecidas como sujeito, predicado e objeto. O sujeito está relacionado a um recurso na *Web* que será descrito. Para estes recursos são atribuídas características ou relacionamentos (predicado). O valor resultante entre a relação do recurso e sua propriedade é conhecido como objeto (Cyganiak et al, 2007). A imagem 2.1 ilustra um exemplo de tripla RDF:

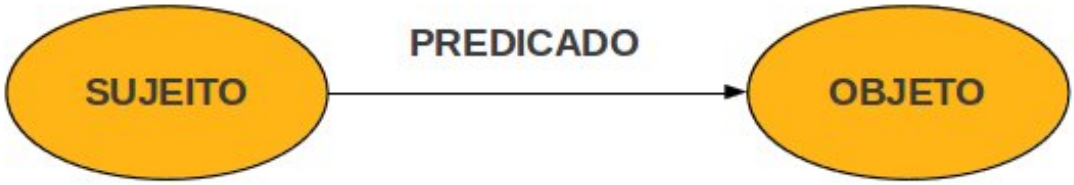

Figura 2.1: Representação da tripla RDF

RDF Schema

O RDF é um modelo de dados simples para descrever o relacionamento entre os recursos e suas propriedades. Porém ele não provê mecanismos para declarar estas propriedades e nem para o relacionamento destas com outros recursos, sendo necessário adicionar semântica ao formato, o que originou o RDF *Schema* (Brickley, 1998).

No modelo RDFS os recursos podem ser instanciados por uma ou mais classes, indicado pela propriedade *rdf:type*. Também é possível apresentar subclasses utilizando a representação *rdfssbClassOf*.

RDFa

As páginas disponiblizadas para usuários em HTML poderiam ser mais úteis se fosse atribuído mais significado entre seus dados. Afim de suprir esta funcionalidade surgiu o RDFa. Este formato possibilita a inserção de atributos entre as TAGS HTML, facilitando o entendimento do conteúdo por aplicações automatizadas. Esta marcação pode incluir tanto indicações mais simples como o título de artigo, quanto descrições mais complexas como dados completos de usuários de uma rede social (Adida et al, 2012).

A listagem 2.1 exemplifica o uso do RDFa em uma página HTML simples:

Listing 2.1: Exemplo de aplicação do RDFa

```
1 <html>
2 <head> [...] </head>
3 <body>
4   <h2 property="http://purl.org/dc/terms/title">The Trouble with
      Bob</h2>
5   <p>Date: <span property="http://purl.org/dc/terms/created">
      2011-09-10</span></p>
6 </body>
```

SPARQL

SPARQL é uma linguagem de consulta utilizada para processar dados representados no formato RDF. Ele é capaz de realizar consultas em diferentes bases de dados em RDF e obtém um conjunto de resultados ou um grafo RDF Prud'hommeaux and Seaborne (2008).

Uma consulta simples utilizando SPARQL é composta por uma cláusula *SELECT* responsável por definir quais variáveis devem aparecer no resultado, e uma cláusula *WHERE* com o modelo padrão de grafo que deve ser buscado. O exemplo a seguir utiliza como padrão de busca uma única tripla onde o resultado que será mostrado é o conteúdo da variável *?titulo* que corresponde ao objeto da tripla, veja o exemplo na listagem 2.2:

Listing 2.2: Exemplo de uma consulta simples com SPARQL

```
1 SELECT ?title
2 WHERE
3 {
4   <http://example.org/book/book1> <http://purl.org/dc/elements
       /1.1/title> ?title .
5 }
```

URIs HTTP Dereferenciáveis

Toda URI deve ser dereferenciável, ou seja, o cliente pode utilizar um protocolo HTTP e obter informações relacionadas ao recurso identificado pela URI. A informação retornada depende do cliente que fez a requisição, para seres humanos uma representação HTML seria mais apropriada enquanto para aplicações que consomem dados o uso do padrão RDF é indicado (Heath, 2007).

É importante saber distinguir a diferença entre uma URI responsável por descreve um objeto da vida real de uma URI que representa o próprio objeto. Uma prática usual é utilizar URIs diferenciadas para cada tipo de retorno.

A definição do tipo de conteúdo a ser enviado para o cliente é estabelecida através da negociação de conteúdo, onde o cliente faz uma requisição através de uma URI HTTP, informando no cabeçalho o tipo de retorno. O servidor deve ser capaz de identificar o formato requisitado e responder com a preferência do cliente, um documento HTML ou RDF. Existem dois mecanismos conhecidos para este processo, o 303 URIs e hash URIs.

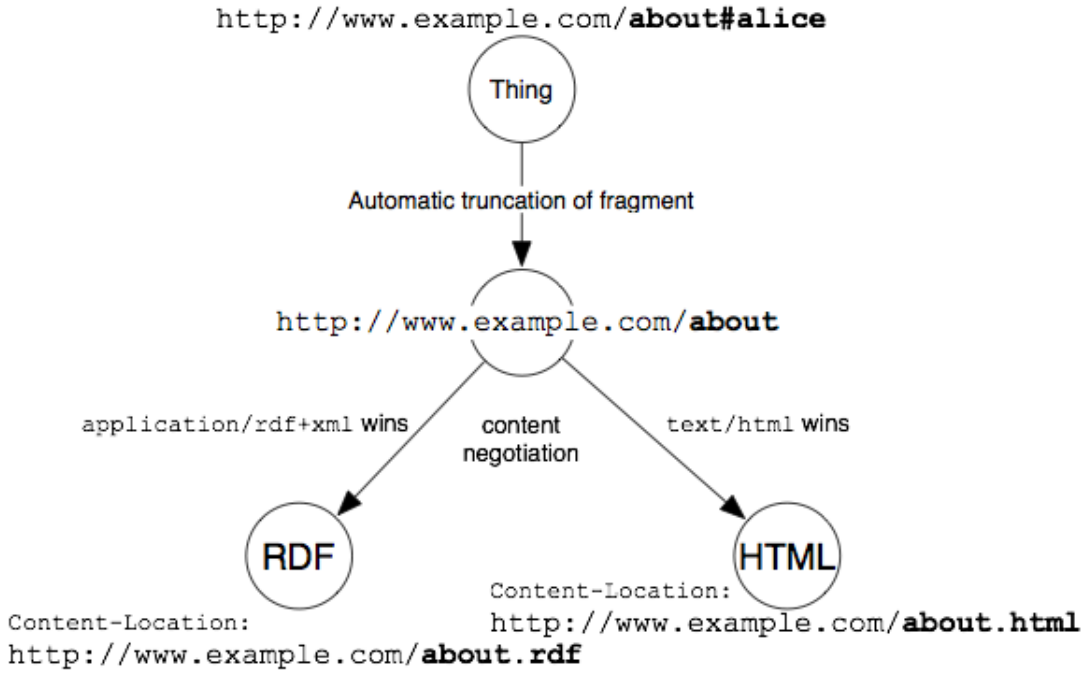

Figura 2.2: Negociação de conteúdo utilizando o mecanismo Hash URI

303 URIs

Quando o cliente quer obter informações de um recurso não informacional ele faz uma requisição HTTP GET através da URI do recurso com uma mensagem Accept no cabeçalho informando a preferência para o formato de documento a ser recebido. O servidor verifica que a requisição corresponde a um recurso não informacional e analisa a preferência do cliente enviando uma URI de um documento que descreve aquele recurso. Após receber a mensagem do servidor o cliente envia outra requisição GET para o documento que descreve o recurso não informacional. Por fim, o servidor envia o documento solicitado pelo cliente (Cyganiak et al, 2007).

Hash URIs

Uma hash URI contém um fragmento separado por um símbolo #. Para o cliente recuperar uma informação através de uma hash URI utilizando o protocolo HTTP ele deve retirar este fragmento antes de fazer a requisição para o servidor. Por fim, o cliente receberá um documento no formato desejado contendo descrições de diversos recursos e poderá utilizar a *hash* URI para obter a informação do recurso requerido (Cyganiak, 2008).

A figura 2.2 exemplifica o mecanismo de hash URI e a negociação de conteúdo:

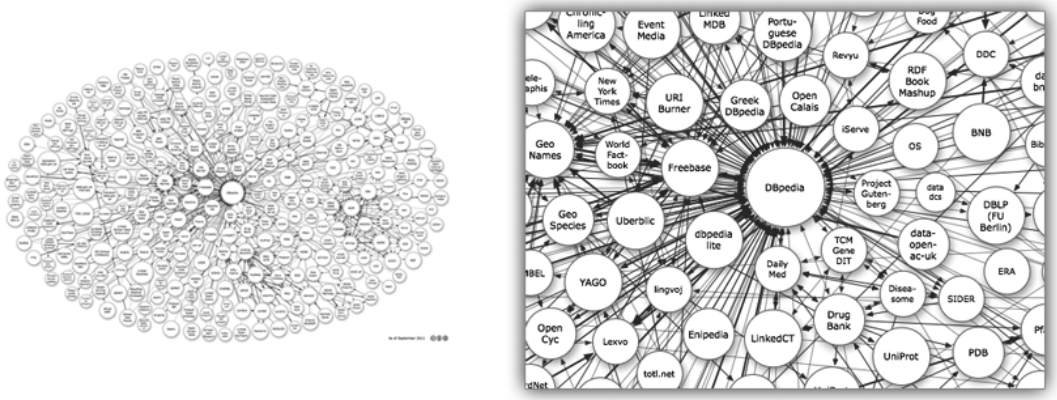

Figura 2.3: Diagrama de representação do projeto LOD até o dia 19 de Setembro de 2011

2.2.4 Projeto: *Linking Open Data*

Atualmente, existem vários projetos como o *Wikipedia*, *Wikibooks*, *Geonames*, que visão a disponibilização de conteúdo livre na *Web*. O projeto *Linking Open Data*, criado pela W3C SWEO, surgiu para incentivar a publicação de *datasets* em RDF, atribuindo ligações entre dados de diferentes fontes (W3C SWEO Community Project, 2011).

Inicialmente o projeto contou com o apoio de pesquisadores , desenvolvedores e pequenas empresas. Com o tempo o projeto ganhou proporções maiores e atingiu grandes organizações principalmente pelo seu caráter público, onde qualquer pessoa pode contribuir fornecendo um conjunto de dados seguindo os princípios de dados ligados (Bizer et al, 2009).

A imagem 2.3 mostra uma representação dos *datasets* criados e as respectivas ligações entre eles:

2.2.5 Aplicações

O crescente volume de dados publicados na *Web* está motivando a criação de aplicações que fazem uso dessas informações. Dentre estas aplicações podemos destacar os navegadores de dados ligados, mecanismos de busca sobre dados ligados e aplicações para domínios específicos Bizer et al (2009).

Navegadores de dados ligados

Os navegadores de dados ligados são aplicações voltadas para seres humanos onde é possível navegar entre dados ligados de fontes diferentes, esta é a principal diferença dos navegadores tradicionais da *Web* que são voltados para a navegação entre documentos utilizando hiperlinks.

Mecanismos de busca

Existem mecanismos de busca que rastreiam recursos na *Web* através dos links nos documentos RDF. Essas aplicações apresentam uma interface para a realização da consulta, podendo ser orientado a seres humanos e orientado a indexação. No primeiro caso temos uma interface onde um usuário pode escrever palavras chaves e o motor de busca irá encontrar os recursos relacionados e exibi-los com suas descrições. No caso dos motores de busca orientados a indexação são utilizados por aplicações que necessitam de realizar buscas e não possuem uma estrutura própria para isso. Os motores fornecem APIs que são utilizadas pelas aplicações para realizar as buscas e estas recebem uma referência dos documentos relevantes encontrados.

Aplicações de domínios específicos

Além dos navegadores e dos motores de busca temos também uma classe de aplicações que atuam em domínios específicos.

2.3 Conclusões

3 Análise do projeto para criação do dataset

Neste capítulo será apresentado o problema que gerou a criação deste projeto e as soluções encontradas. A seção 3.1 mostra o cenário do problema a ser tratado. A seção 3.2 se dedica a apresentar as soluções utilizadas para resolução do problema. A seção 3.3 exibe a arquitetura do projeto explicando como será a interação dos módulos. Por fim, a seção 3.4 é reservada às conclusões finais do projeto.

3.1 Cenário

No Brasil as Organizações Não Governamentais são as principais atuantes na busca por pessoas desaparecidas junto às famílias. As ONGs utilizam amplamente a Internet como um meio para a divulgação de casos de desaparecimentos por atingir um grande número de pessoas. Para estes casos, as informações estão limitadas a documentos HTML simples e de fácil visualização para qualquer pessoa, mas que dificultam o processamento por máquina. Uma vez que os dados não se encontram estruturados, torna-se difícil para qualquer aplicação extrair conteúdos pertinentes das páginas. Outro aspecto importante é que os sites que tratam destes assuntos agem de forma independente, cada um com sua própria base, sendo possível a ocorrência de informações duplicadas.

Como as ONGs são entidades filantrópicas, em muitos dos casos faltam recursos para manter uma equipe capaz de trabalhar com a manutenção dos dados. As informações na *Web* nem sempre são atualizadas. Para este projeto, por exemplo, tentativas para entrar em contato com algumas ONGs não obtiveram sucesso. Poucos foram os casos em que tivemos apoio para a utilização dos dados.

Além do conteúdo presente em sites também é possível encontrar dados de pessoas desaparecidas em serviços de e-mails através de mensagens que procuram sensibilizar os usuários a ajudarem repassando o e-mail para seus contatos. Como consequência temos uma grande quantidade de informações sendo propagadas que acabam se transformando em correntes de e-mail, ou *hoax*. Quando uma pessoa é encontrada, as mensagens repas-

sadas continuam sendo transmitidas para outros usuários tornando-se um incômodo para os clientes de e-mail e para a própria família que recebe informações falsas.

A *Web* está evoluindo para que as informações possam ser extraídas e manipuladas através de aplicações, não se limitando somente a documentos. A idéia de assimilar significado aos dados, princípio da *Web* semântica, requer também que estes estejam conectados, formando assim um espaço global de dados. O projeto Linking Open Data visa disponibilizar dados de forma aberta na internet utilizando o princípio de Dados Ligados, onde as informações são oferecidas em um formato padrão, o RDF, e os dados possuem ligações entre si. Desta forma os links não são somente endereços para outros documentos, mas sim um relacionamento entre duas entidades onde é possível ressaltar a natureza desta relação. Como este é um projeto recente, muito trabalho ainda é necessário para que os dados estejam no formato mencionado. Além disso, cada domínio específico deve possuir uma estrutura que o defina. Voltando ao caso das pessoas desaparecidas seria interessante a disponibilização aberta destas informações, porém inicialmente não existe nenhum modelo padronizado que descreva uma *hoax*.

Atualmente não existe um modelo padronizado para descrever *hoax*.

3.2 Soluções

Este projeto tem como objetivo geral oferecer um mecanismo que possa contribuir com a redução de *spams* que circulam entre os serviços de e-mails. Porém, é difícil manipular dados em sites de ONGs devido a falta de padronização no fornecimento destas informações.

Como o problema principal é a falta de um repositório de dados abertos e prontos para o consumo, este projeto propõe a criação de um *dataset* em formato RDF com informações de pessoas desaparecidas. O *dataset* será integrado ao projeto *linking open data* seguindo os princípios de dados ligados definidos na seção 2.2.2.

Tratando-se de dados públicos deve ser oferecido uma interface HTML para que qualquer pessoa interessada pelos dados possa acessá-los. Para isso, a página que oferece estes dados deve identificar a requisição de um usuário normal e exibir o conteúdo no formato apropriado.

A finalidade de utilizar dados ligados está na capacidade de realizar consultas entre domínios relacionados, possível através das conexões entre os recursos. Para que desenvolvedores tenham acesso aos dados do *dataset* é fornecida uma interface de consulta. Através da interface as aplicações podem processar as informações recebidas em um formato padronizado.

Para oferecer estes dados é necessário criar uma forma de descrevê-los. Como não existe um modelo para descrever *hoax* será necessário criar uma ontologia específica para o domínio. Ainda existem propriedades que podem ser comuns em algumas ontologias já existentes que podem ser aproveitadas para descrever *hoax*.

Em relação ao problema de classificar e-mails de pessoas desaparecidas, surgiu a possibilidade de criar uma aplicação *Web* capaz de contribuir com a análise destes dados. Esta aplicação analisa os dados do e-mail e verifica a existência da pessoa desaparecida de acordo com as informações contidas no *dataset*.

A automatização do processo de validação e o uso de uma base da dados autentica oferece um mecanismo para que os usuários de serviços de e-mail tenham maior motivação para verificar os e-mails recebidos uma vez que terão o apoio de um software, não sendo necessário nenhum trabalho de pesquisas na *Web*, e poderá exercer um papel social.

3.3 Arquitetura

O modelo arquitetural do projeto consiste em um conjunto de módulos que trocam informações entre si. O modelo pode ser visto na figura 3.1.

Como não existem informações em um formato aberto o primeiro passo é obter estes dados em sites de ONGs. O método utilizado neste caso é chamado de raspagem de dados. Através deste mecanismo, um script conhecido como *web crawler* acessa o HTML das páginas e retira o conteúdo de acordo com as especificações do desenvolvedor.

Os dados obtidos são armazenados em um banco relacional e mapeados para um formato aberto e disponível para consultas. Uma interface será oferecida para a realização de consultas pelas aplicações.

O site é responsável pela publicação dos dados na *Web* oferecendo a interface HTML, visível pelo usuário. Ao identificar uma requisição de uma aplicação o site deve

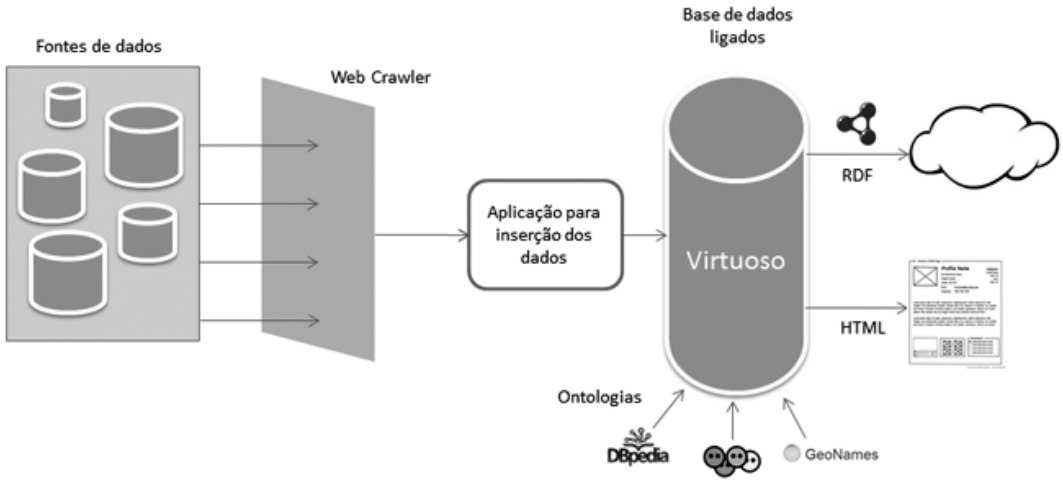

Figura 3.1: Arquitetura do projeto

fornecer os dados em um formato padrão adicionando ligações para outros recursos.

Para exemplificar o uso de *datasets* em aplicações, será criado neste projeto um programa capaz de analisar o conteúdo de mensagens em redes sociais, identificando informações de pessoas desaparecidas e buscando dados relacionadas na base gerada para o projeto. A aplicação consiste em uma aplicação *Web mobile*, acessível via *browser* em qualquer *smartphone*. O protótipo deverá capturar as informações presentes nas mensagens e, utilizando a interface de consulta SPARQL, poderá receber uma resposta mais precisa sobre a situação atual da pessoa desaparecida, transmitindo para o usuário final.

3.4 Conclusões

Este capítulo apresentou brevemente como será tratado cada parte do desenvolvimento do projeto. É importante salientar que a escolha de se trabalhar com dados de pessoas desaparecidas não é restritiva. Este protótipo pode ser adaptado de forma a tratar outros tipos de *hoaxes*.

Cada módulo é composto por tecnologias que serão descritas mais detalhadamente no capítulo a seguir.

4 Implementação

Este capítulo mostra como foi realizada a extração e disponibilização de dados, bem como as tecnologias utilizadas, além de exemplificar o uso de dados abertos em uma aplicação prática. A seção 4.1 discorre sobre a técnica de raspagem de dados. A seção 4.2 traz a ferramenta utilizada para o armazenamento dos dados. Na seção 4.3 é apresentado o site que traz uma interface de visualização para o usuário, além de disponibilizar os dados no formato RDF. A seção 4.4 apresenta um protótipo de aplicação para rede social. Por fim, a seção 4.5 apresenta as conclusões do projeto.

4.1 Coleta de dados

Para formar um dataset de pessoas desaparecidas foi necessário inicialmente recolher dados na *web*. Como não existe informação em um formato aberto para o domínio em questão, o primeiro passo foi obter estes dados em sites de ONGs. O processo de raspagem de dados, também conhecido como *screen scraping*, consiste em extrair dados em páginas *Web* através de *scripts* que analisam o HTML, em busca de padrões pré-definidos, e armazenam em um banco de dados Vlist et al (2007). Atualmente, existem milhares de *scripts* que realização esta função. Como as linguagens procuram por padrões em elementos HTML o uso de expressão regular é altamente recomendável para esta tarefa. A complexidade em extrair dados varia de acordo com o código HTML obtido. Em site que apresentam padrões como TAGS para título, subtítulo, tabelas, além do uso de classes e identificadores, torna-se fácil a construção de um programa para tal processo. Uma vez que, muitos dos sites disponíveis não aplicam estas práticas, foi necessário a busca por uma ferramenta que facilita-se a tarefa de raspagem de dados. Utilizou-se um serviço disponibilizado gratuitamente chamado ScraperWiki[1].

O ScraperWiki é uma ferramenta colaborativa que permite a criação de *scripts* em diferentes linguagens como PHP, Ruby e Python, para capturar dados na *Web*. É possível

[1]https://scraperwiki.com

criar uma conta on-line e armazenar o histórico de seus códigos. O sistema também possui um banco de dados para armazenar as informações obtidas e mecanismos para realizar consultas sobre o banco.

Um dos principais fatores que torna a ferramenta atrativa é a qualidade da biblioteca que esta oferece. Através de uma interface *Web* é possível executar *scripts* diretamente no servidor do ScraperWiki como visto na imagem 4.1:

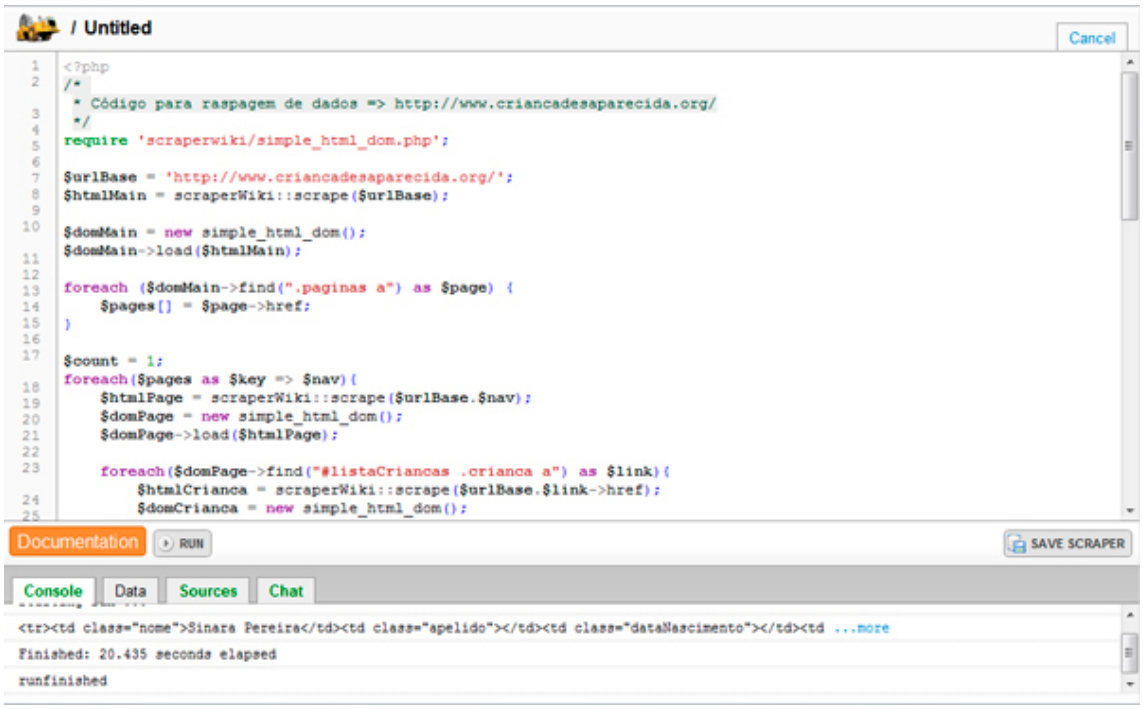

Figura 4.1: Interface de execução de scripts do ScraperWiki

Devido a familiaridade com a linguagem, foi escolhido o PHP. O uso desta biblioteca ajudou consideravelmente durante a coleta dos dados. Um objeto é criado a partir da URL base que será analisada. Este objeto possui uma representação dos elementos HTML que podem ser acessados facilmente, dependendo da estrutura criada pelo desenvolvedor do site. É possível acessar o conteúdo de tags HTML através da busca de padrões entre tags, classes e identificadores.

Listing 4.1: Exemplo de código para extração de dados

```
1 <?php
2     require 'scraperwiki/simple_html_dom.php';
3
4     $urlBase = 'http://www.criancadesaparecida.org/';
5     $htmlMain = scraperWiki::scrape($urlBase);
6
7     $domMain = new simple_html_dom();
```

```
8       $domMain->load($htmlMain);
9
10      foreach ($domMain->find(".paginas a") as $page) {
11          $pages[] = $page->href;
12      }
13  ?>
```

O trecho de código exemplificado na listagem 4.1 mostra como os dados são coletados. Em uma estrutura de controle podemos atribuir o resultado da busca em uma variável. O método *find* analisa o objeto $domMain e busca por TAGS "a", que se encontram dentro de outras TAGS que possuem a classe "paginas". A iteração irá se repetir para todos os padrões encontrados. Através da variável é possível acessar os atributos e valores, caso existam.

Através de uma pesquisa foi possível encontrar alguns sites com informações relevantes para alimentar o banco de dados. Existem diversos estados que disponibilizam sites para mostrar dados de pessoas desaparecidas, Santa Catarina[2], Minas Gerais[3], São Paulo[4], Rio de Janeiro[5], entre outros. Utilizou-se também algumas páginas que procuram mostrar as fraldes na internet, a página do facebook chamada "Lendas da internet (Pessoas Desaparecidas)"[6].

Um aspecto que deve ser levado em consideração é o fato deste método de captura de informação ser totalmente dependente do layout do site. Qualquer alteração na estrutura do HTML deverá refletir em alterações no código da raspagem.

4.2 Armazenamento dos dados

Após a coleta de dados foi necessário um banco de dados para armazená-los. Como estas informações deverão ser disponibilizadas em um formato aberto e de fácil acesso, optou-se por utilizar uma ferramenta própria para este processo, o Virtuoso, que será apresentado a seguir.

[2]http://www.criancadesaparecida.org/
[3]http://www.desaparecidos.mg.gov.br
[4]http://www2.policiacivil.sp.gov.br
[5]http://www.fia.rj.gov.br
[6]https://www.facebook.com/pages/Lendas-da-internet-Pessoas-Desaparecidas/166504023443529?fref=ts

4.2.1 OpenLink Virtuoso

O Virtuoso[7] é um software criado para atuar como um servidor universal que comporte as mais variadas tecnologias incluindo servidor *Web*, servidor de arquivos, banco de dados e armazenamento de XML nativo, todas estas funcionalidades disponíveis em uma única ferramenta. O Virtuoso possibilita trabalhar com o gerenciamento de bancos de dados relacionais e de dados no formato RDF e XML, como mencionado anteriormente possui um servidor *Web* além de um servidor de Dados Ligados e de aplicações *Web*. Outra característica importante é a possibilidade de trabalhar com Serviços *Web*, SOAP ou REST OpenLink Software Documentation Team (2007)).

O Virtuoso apresenta compatibilidade entre os mais conhecidos sistemas operacionais como Linux, Windows, Mac e suporta também uma grande variedade de banco de dados, tais como SQL Server, MySQL, Oracle, PostgreSQL, entre outros. A versão open source do virtuoso foi desenvolvida pela OpenLink Software e é conhecida como OpenLink Virtuoso.

Interface administrativa

Logo após a instalação o Virtuoso oferece uma interface administrativa acessível via *browser*. A imagem 4.2 mostra a tela principal onde é possível observar os atalhos para os principais recursos disponíveis.

Interface de consulta SPARQL

O Virtuoso oferece suporte a uma linguagem de consulta para o processamento de dados RDF, o SPARQL. A interface de consulta pode ser acessada utilizando o caminho "/sparql/", ou seja, se existe uma instalação local onde as solicitações HTTP são feitas na porta 8080 basta utilizar a URL http://example.com:8080/sparql/, veja na figura 4.3. O endereço por si só garante o acesso a uma interface *Web* onde o usuário pode definir o grafo, a consulta SPARQL, o formato de retorno, entre outras configurações Virtuoso Wiki (2009).

A mesma URL aceita parâmetros e pode ser utilizada para fazer requisições GET

[7]http://virtuoso.openlinksw.com

Figura 4.2: Interface administrativa do OpenLink Virtuoso

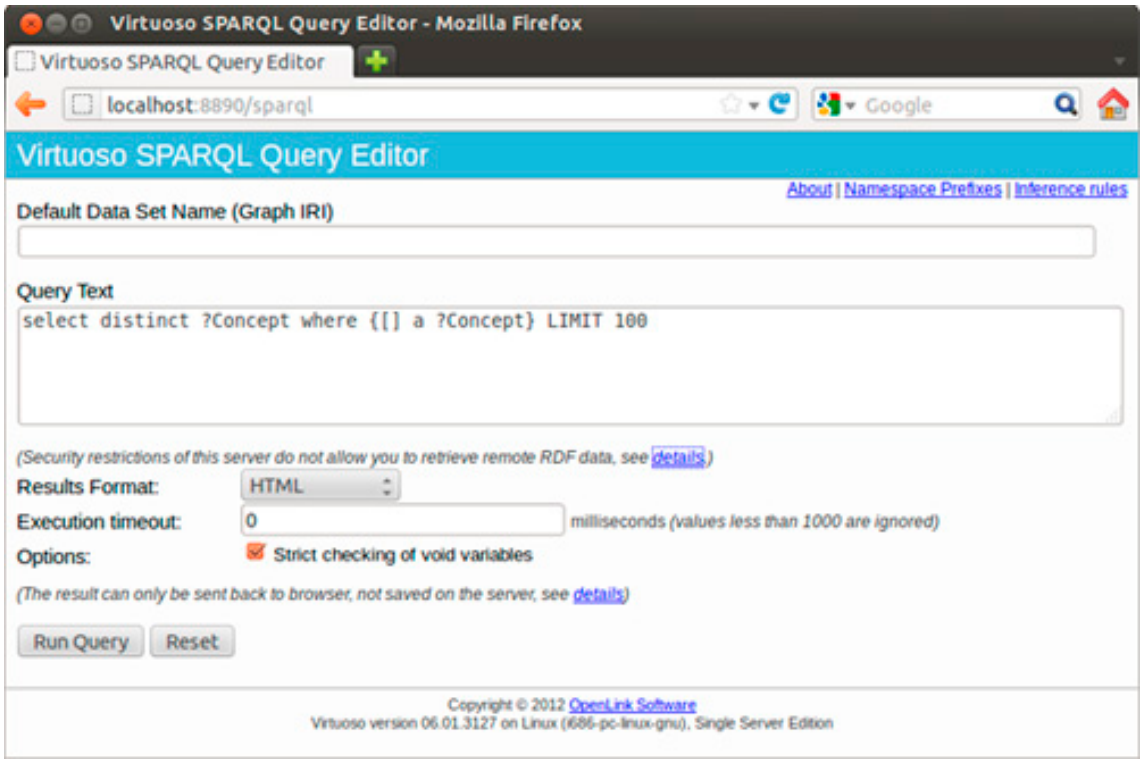

Figura 4.3: Interface de consulta SPARQL

e POST, dependendo da necessidade do desenvolvedor. Um dos parâmetros está relacionado ao tipo de retorno da consulta, suportando diferentes formatos utilizados para

disponibilizar dados abertos como XML, CSV, JSON e RDF, além do formato HTML.

Além da cláusula SELECT o endpoint SPARQL possibilita a inserção e remoção de triplas RDF sendo necessário atribuir permissão ao usuário que possui acesso à interface de consulta.

RDF Views

Grande parte das informações presentes na *Web* atual encontram-se armazenadas em banco de dados relacionais. Para facilitar a disponibilização destes dados no formato RDF o Virtuoso disponibiliza um módulo capaz de mapear dados de bancos relacionais para representações no formato RDF. Uma vez que o mapeamento é realizado dinamicamente qualquer alteração dos dados na base relacional também reflete nas visões RDF.

O Virtuoso oferece um ajudante para a criação de mapeamentos simplificados de banco de dados, porém existe a possibilidade de customizar este processo.

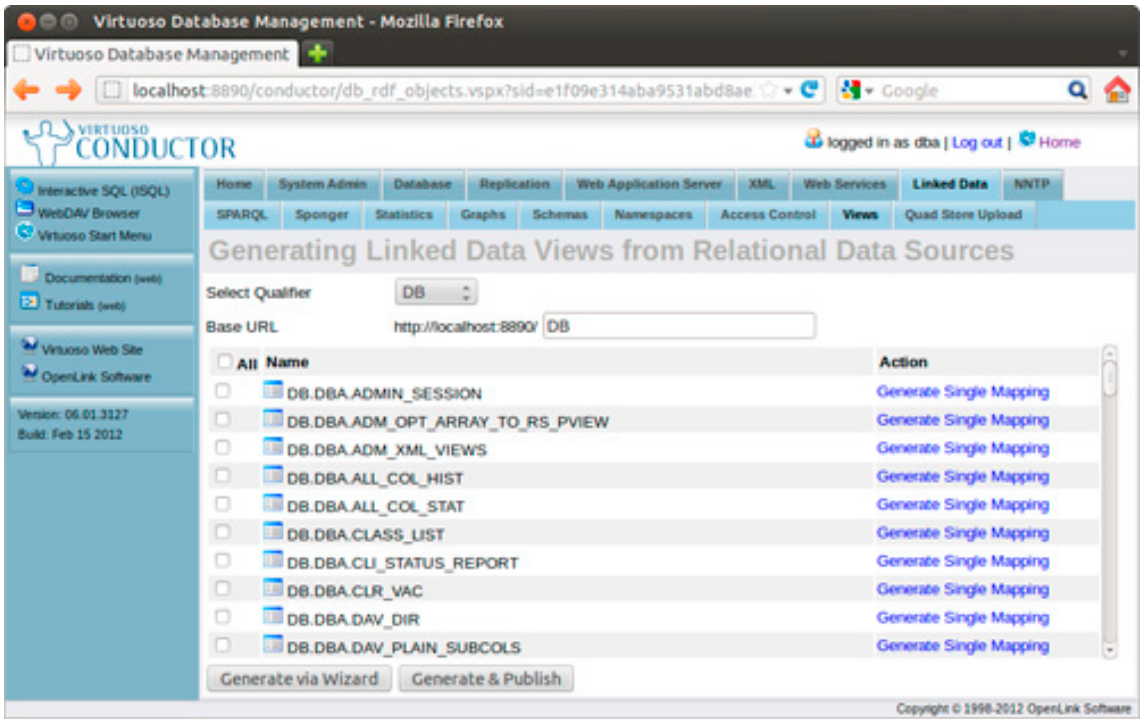

Figura 4.4: Interface para gerar mapeamentos de dados de bancos relacionais para visões RDF

Ao criar um novo mapeamento customizado é importante conhecer como os dados se relacionam de um modelo para o outro. O modelo entidade relacionamento é formado por tabelas e seus campos. Um recurso é representado por um registro de uma tabela e identificado por algum identificador único. No modelo RDF uma entidade é descrita

através de um conjunto de triplas, que representam o sujeito, predicado e objeto. A tabela representa o tipo de informação que ela armazena, sendo assim é definida por uma classe. Os campos da tabela são os atributos da classe e correspondem ao predicado nas triplas. Os valores dos campos estão associados ao objeto das triplas contendo um valor atômico ou uma referência a outro recurso, no caso da existência de chave estrangeira no banco. Por fim, da mesma forma que o *id* identifica um registro único da tabela o sujeito da tripla também referencia um recurso único.

Para este projeto temos um modelo relacional simples composto por uma única tabela sem chave estrangeira. Veja a representação da tabela e seus campos na figura 4.5.

Figura 4.5: Tabela do modelo relacional contando dados de pessoas desaparecidas

4.3 Disponibilização dos dados

Os dados das pessoas desaparecidas possuem um identificador gerado de acordo com os princípios de dados ligados, uma URI HTTP. Para ser único, cada indivíduo possui um *id* numérico incrementado automaticamente pelo sistema durante a inserção. Dependendo do usuário que irá solicitar os dados temos dois tipos de representações diferentes: o HTML, quando for requisitado por um browser, ou o RDF, para a solicitação de aplicações. A negociação de conteúdo foi realizada utilizando um script PHP chamado *EasyPub*[8], que analisa o cabeçalho da requisição de um recurso não-informacional e retorna o conteúdo adequado.

[8]http://buzzword.org.uk/2009/easypub/

Após a inserção dos dados no Virtuoso, desenvolvedores podem acessar todo conteúdo abertamente através de uma interface de consulta SPAQL. Utilizando este mecanismo de consumo foi criado um site para a exibição dos dados no formato HTML para a visualização de usuários e no formato RDF para que possam ser acessados por aplicações. O site também apresenta uma interface para pesquisa como pode ser visto na imagem 4.6:

Figura 4.6: Página principal do site http://desaparecidos.ice.ufjf.br/

A página inicial apresenta um sistema de busca onde o usuário pode pesquisar por pessoas desaparecidas selecionando os critérios desejados. Ao selecionar um indivíduo, é possível visualizar uma descrição mais completa do caso de desaparecimento, como visto na imagem 4.7. Na mesma tela encontra-se a opção de download dos dados em formato RDF.

Figura 4.7: Interface de busca.

4.3.1 Representação RDF

Para a representação RDF reaproveitou-se algumas ontologias como Foaf[9] para descrição de pessoas, Geonames[10] para dados de localização, e dbpprop[11] para descrever características físicas. Algumas propriedades relacionadas ao domínio de pessoas desaparecidas como a data de localização e a situação do indivíduo foram criadas originando uma nova ontologia chamada des. A figura 4.8 exemplifica a ligação entre os *datasets*.

Um dos princípios dos dados ligados requer que as informações estejam conectadas

[9] http://www.foaf-project.org
[10] http://www.geonames.org
[11] http://dbpedia.org

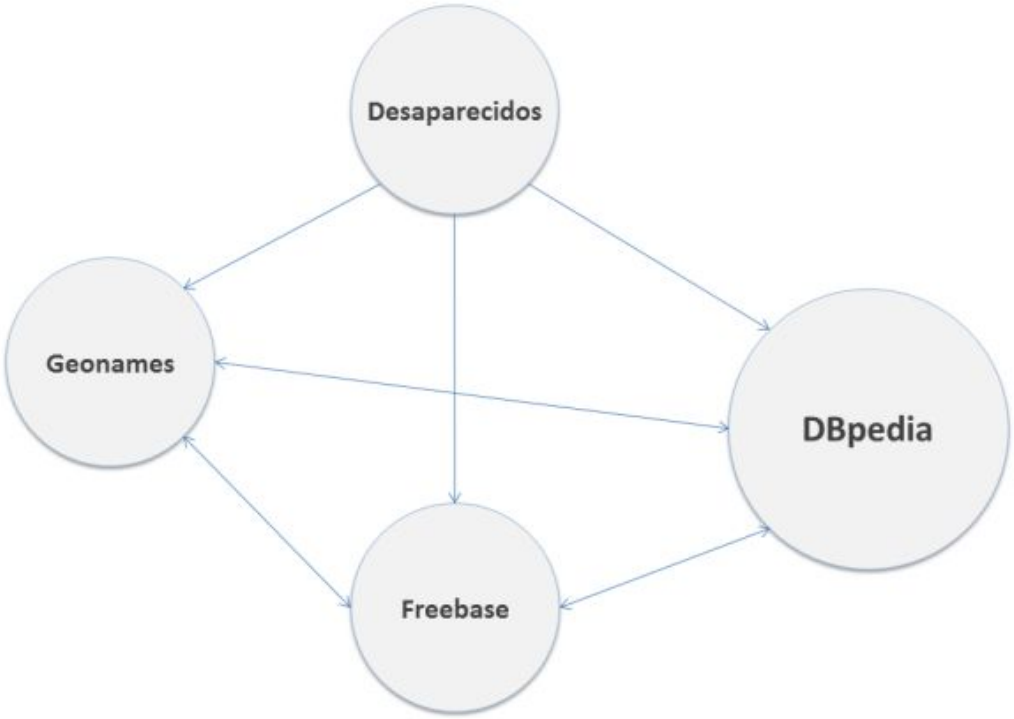

Figura 4.8: Ligações entre o *dataset* Desaparecidos e os *datasets* pertencentes ao LOD

entre si. Com o objetivo de satisfazer esta condição foi necessário estabelecer ligações entre os dados de desaparecidos com outras bases existentes. Para este caso, utilizou-se dados de localização e a fonte em que a informação foi retirada. Os dados de localização foram obtidos através do *Freebase*[12]. Este serviço é composto por um vasto banco de dados com informações recolhidas de diferentes fontes, e as disponibilizam para consulta. Além das buscas oferecidas para usuários comuns, o Freebase fornece uma API que possibilita o consumo de seus dados por aplicações. Com o auxílio desta ferramenta, foi possível buscar informações de cidades que possam compor os links para o RDF gerado.

Para acessar os dados utilizou-se uma linguagem de consulta própria do *Freebase*, o MQL (*Metaweb query language*). Durante o processo de consumo de dados a aplicação solicita um *HTTP Request*, informando a query a ser processada, e recebe um *HTTP Response* com o resultado obtido. O desenvolvedor pode optar por utilizar o método POST ou GET para fazer a solicitação.

Um exemplo de RDF gerado pode ser visto na listagem 4.2:

Listing 4.2: Modelo de RDF gerado

[12]http://www.freebase.com/

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dbpprop="http://dbpedia.org/property/"
  xmlns:being="http://purl.org/ontomedia/ext/common/being#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:des="http://www.desaparecidos.com.br/rdf/">

  <rdf:description rdf:about="http://www.desaparecidos.ufjf.br/
     desaparecidos/3">
    <foaf:name>José da Silva</foaf:name>
     <foaf:birthday>10/10/2000</foaf:birthday>
     <foaf:gender>Masculino</foaf:gender>
     <foaf:img>http://desaparecidos.ice.ufjf.br/123456789.jpg</
        foaf:img>
     <des:cityDes>Juiz de Fora</des:cityDes>
     <des:cityDes rdf:resource="http://rdf.freebase.com/ns/en.
        juiz_de_fora" />
     <des:cityDes rdf:resource="http://dbpedia.org/resource/
        Juiz_de_Fora" />
     <des:stateDes>Minas Gerais</des:stateDes>
     <dbpprop:height>1.70</dbpprop:height>
     <dbpprop:weight>65</dbpprop:weight>
     [...]
     <des:status>Desaparecido</des:status>
     <des:source>http://desaparecidos.ice.ufjf.br/123456</
        des:source>
  </rdf:description>
</rdf:RDF>
```

4.4 Aplicação para redes socias

Com um repositório de informações abertas a disposição, torna-se mais fácil a criação de aplicações. O problema identificado é a propagação de *hoax* nas redes sociais e a falta de praticidade em descobrir a validade destas mensagens. Para este cenário, uma aplicação poderia contribuir reduzindo os compartilhamentos desnecessários.

4.4.1 Visão geral

Para este projeto sugeriu-se o desenvolvimento de uma aplicação *Web* capaz de acessar dados no mural de usuários do Facebook em busca de informações de pessoas desaparecidas e, através de uma interface simples, retornar a situação de um indivíduo e meios para contribuir com a sua localização. A imagem 4.9 ilustra a arquitetura do projeto.

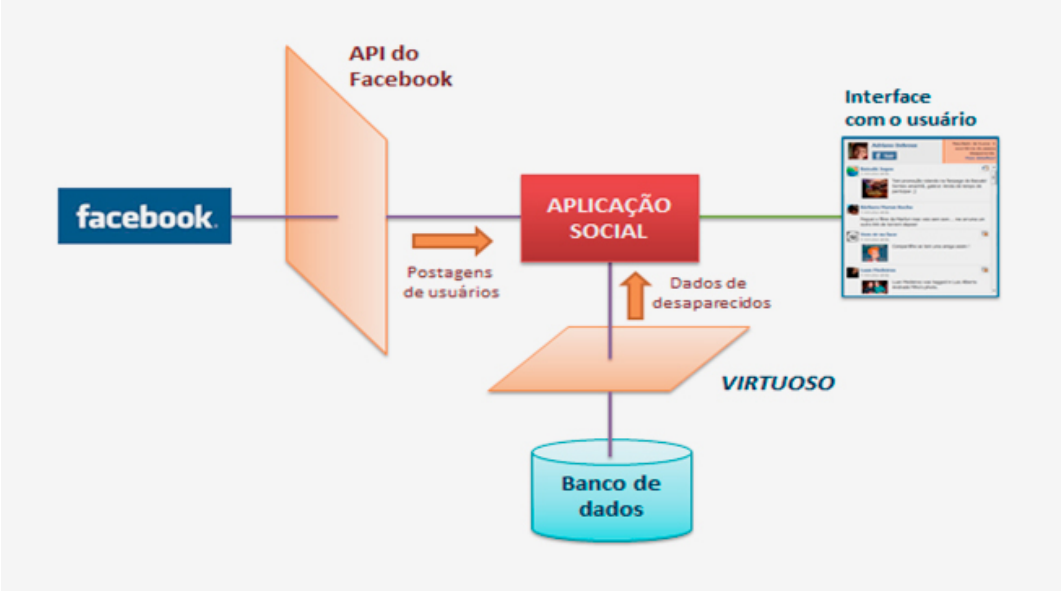

Figura 4.9: Arquitetura da aplicação

O Facebook disponibiliza uma API para que os desenvolvedores possam criar aplicações e buscar dados de usuários na rede. Ao acessar as atualizações do mural é possível utilizar técnicas para identificar possíveis mensagens de pessoas desaparecidas. Neste caso foi implementado um *array de tokens*. Para melhorar a precisão da busca utilizou-se o prefixo de palavras que possuem maior ocorrência entre postagens de pessoas desaparecidas.

Ao identificar uma postagem como sendo de um desaparecido a aplicação deverá verificar a ocorrência desta pessoa na base. A verificação é realizada através de uma interface fornecida pelo Virtuoso que permite realizar consultas SPARQL.

Antes de realizar são retirados termos irrelevantes através da técnica conhecida como *stopwords* a fim de simplificar o conteúdo analisado. Em seguida a mensagem é separada em combinações sequenciais de 3 palavras, técnica conhecida com n-gram. Todas as combinações são agrupadas formando uma única consulta SPARQL.

Caso a pessoa seja encontrada na base o sistema irá exibir seu status para o usuário. Se ela não existir a pessoa poderá cadastrá-la de uma forma simples, permitindo assim ampliar o banco de dados, sem que as informações fiquem atreladas somente aos dados de ONGs.

4.4.2 Acessando os dados do Facebook

Através do Facebook é possível criar aplicações *Web* integradas à rede social de forma que o desenvolvedor possa reaproveitar dados dos usuários e trabalhar com tais informações. O acesso das informações é realizado através da *Graph API* onde os dados são tratados como objetos, tais como páginas, pessoas, fotos, e estão relacionados entre si.

Cada objeto possui um identificador único e pode ser acessado através de uma requisição HTTPS com o seguinte formato:

https://graph.facebook.com/ID

O resultado é oferecido no formato JSON. A listagem 4.3 mostra um exemplo de requisição utilizando o *id* da UFJF:

Listing 4.3: Exemplo de requisição utilizando a *GRAPH API*

```
1  {
2     "name": "Universidade Federal de Juiz de Fora (UFJF)",
3     "is_published": true,
4     "website": "http://www.ufjf.br/",
5     "username": "souUFJF",
6     "founded": "23/12/1960",
7     "description": "A UFJF posiciona-se como polo cient\u00edfico
         e cultural de uma [...]",
8     "about": "UFJF - Universidade Federal de Juiz de Fora: uma das
          melhores [...]",
9     "location": {
10       "street": "Rua Jos\u00e9 Louren\u00e7o Kelmer, [...] Bairro
             S\u00e3o Pedro",
11       "city": "Juiz de Fora",
12       "country": "Brazil",
13       "zip": "36036900"
14    },
15    "parking": {
16       "street": 0,
17       "lot": 1,
18       "valet": 0
19    },
20    "phone": "(32) 2102-3978, 2102-3911 - Central de Atendimento",
21    [...]
22 }
```

Através desta requisição o Facebook retorna apenas os dados públicos disponíveis. Consumindo dados de usuários, este detalhe é mais notório. Os usuários da rede social podem escolher o que deve ou não ser exibido publicamente. Neste caso, o resultado fica restrito a poucas informações, o que pode não ser suficiente para uma determinada

aplicação.

Para que desenvolvedores acessem dados restritos é necessário a permissão do usuário. O Facebook oferece um conjunto de permissões capazes de buscar informações, postar fotos e até mensagens no mural de uma determinada conta. Ao acessar uma aplicação o usuário deve autorizar ou não o uso de seus dados. Com a autorização concedida o desenvolvedor recebe um token de acesso e pode utilizá-lo para buscar a informação requerida.

Com o objetivo de facilitar a criação de novos aplicativos o Facebook também disponibiliza um conjunto de SDKs com métodos de acesso aos dados para aplicações *Web* e mobile.

Com o objetivo de facilitar a criação de novos aplicativos o Facebook também disponibiliza um conjunto de SDKs com métodos de acesso aos dados para aplicações *Web* e mobile. O Facebook disponibiliza uma página direcionada aos desenvolvedores[13] com todas as informações necessárias para criar ferramentas capazes de interagir com a rede social.

4.4.3 Aplicação social

Aproveitando a popularidade dos smartphones nos últimos anos optou-se por criar um protótipo de aplicação *mobile* onde o usuário possa acompanhar as atualizações de seu mural no Facebook, assim como outras aplicações do gênero, porém com um diferencial que será o analisador de *hoax* para pessoas desaparecidas.

A fim de abranger uma quantidade maior de aparelhos criou-se uma aplicação *Web mobile* através do Facebook. Este tipo de aplicação não é direcionada para uma só plataforma. Ela é construída com XHTML, JavaScript e CSS, acessível via *browser*, permitindo seu uso através de Iphone, Ipad, Android, entre outros.

Para criar a aplicação mencionada acima basta selecionar a opção *Mobile Web* ao criar uma nova aplicação no Facebook e informar o endereço onde será hospedado, como exemplificado na figura 4.10.

[13]https://developers.facebook.com/

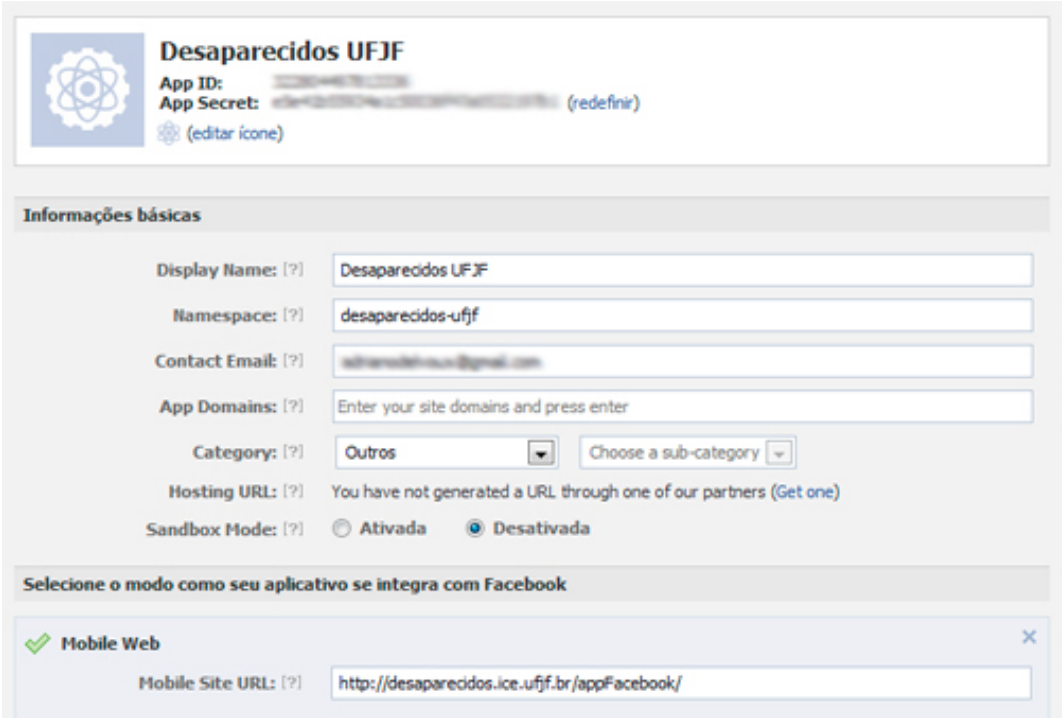

Figura 4.10: Criando uma nova aplicação no Facebook.

4.4.4 Funcionamento da aplicação

A imagem 4.11 apresenta a interface da aplicação em três situações possíveis. A primeira tela é exibida para usuários que não estão logados na rede social. Através dela é possível acessar a uma conta do Facebook e autenticar a aplicação. A segunda tela exemplifica a funcionalidade básica da aplicação, onde o usuário pode acompanhar as atualizações do seu mural. O diferencial desta aplicação se encontra na terceira tela. As mensagens que apresentam qualquer conteúdo de pessoas desaparecidas são marcadas automaticamente e analisadas de acordo com as informações presentes na base. Se for encontrada qualquer informação relevante o usuário será informado imediatamente e ainda terá a possibilidade de acessar o site e verificar outros dados relacionados a esta pessoa clicando no link “Mais informações”, dentro da caixa azul. A aplicação também informa o número de ocorrências de mensagens encontradas logo na tela principal.

Para evitar consultas desnecessárias na base, as mensagens são processadas inicialmente em busca de termos relevantes capazes de identificar prováveis postagens de pessoas desaparecidas. Neste caso, optou-se por utilizar um *array de tokens*. A escolha dos melhores *tokens* foi realizada com o apoio de uma técnica chamada *stemming*, que propõe um conjunto de etapas para chegar a um termo comum, eliminando os fatores que

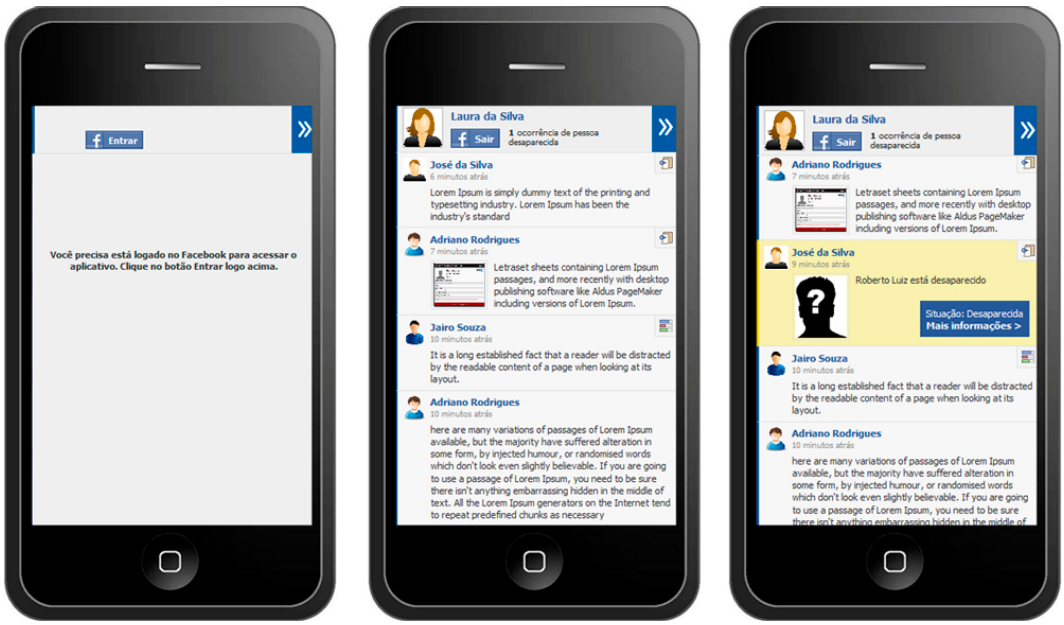

Figura 4.11: Interface mobile da aplicação.

geram variações Orengo and Huyck (2001). Um exemplo é a palavra “desaparecido”, que pode aparecer em diversas postagens sofrendo variações. Aplicando o algoritmo *stemming*, o *token* para busca passa a ser somente “desaparec”.

Para reduzir o conteúdo a ser processado utilizou-se uma técnica conhecida como stop words. Comum entre os mecanismos de buscas esta técnica propõe a remoção de palavras não relevantes em uma pesquisa, com o objetivo de simplificar a consulta e reduzir o tempo de resposta Rouse (2005). As stop words são artigos, preposições, pronomes, entre outras. A lista de stop words utilizada para o projeto encontra-se acessível em http://snowball.tartarus.org/algorithms/portuguese/stop.txt.

A sequência de termos resultantes, agora simplificada, é analisada para retirar possíveis nomes próprios. Como os nomes são encontrados entre os textos iniciando, na maioria das vezes, com a letra maiúscula, criou-se um algoritmo que fizesse esta verificação e guardasse tais nomes. Por fim, os nomes encontrados são utilizados na consulta realizada na base criada.

O resultado obtido até então é um array com os possíveis nomes encontrados. Estes dados devem agora ser verificados na base, utilizando a interface SPARQL do Virtuoso. Para evitar mais de uma requisição REST optou-se por fazer a busca em apenas uma consulta utilizando o operador OU.

A listagem 4.4 exibe um exemplo de consulta realizada na base do Virtuoso.

Listing 4.4: Exemplo de consulta no Virtuoso

```
1  PREFIX foaf: <http://xmlns.com/foaf/0.1/>
2  PREFIX des: <http://desaparecidos.ice.ufjf.br/desaparecido/>
3  SELECT  ?id ?nome ?situacao ?data_localizacao
4  WHERE {
5       ?recurso des:id ?id
6       FILTER regex(?nome, "Leticia", "i")
7       OPTIONAL {
8            ?recurso foaf:name ?nome
9       }.
10      OPTIONAL {
11           ?recurso des:dateLocation ?data_localizacao
12      }.
13      OPTIONAL {
14           ?recurso des:status ?situacao
15      }.
16 }
17 ORDER BY ?nome
```

O retorno da consulta será verificado e transmitido ao usuário final da aplicação.

4.5 Conclusões

A etapa inicial deste projeto trouxe um primeiro obstáculo que mostrou a importância que os dados abertos possuem para o consumo de aplicações. O método de raspagem de dados foi uma solução para obter os dados iniciais, uma vez que não foi possível encontrar mecanismos facilitadores para a obtenção de informações de pessoas desaparecidas. É importante salientar a necessidade de constantes atualizações nos scripts caso algum site passe por mudanças na exibição de seu conteúdo.

Com um conjunto de dados satisfatórios foi possível criar uma base de dados e um mecanismo de consulta. Podemos dizer que foram atendidos os princípios básicos que regem a disponibilização de dados abertos. As informações não se encontram presas a cadastros e não possuem restrição de privacidade. Os dados são oferecidos com o mesmo conteúdo da fonte onde foram retirados, sem alterá-los. Encontram-se acessíveis para qualquer usuário, e inclusive, para o processamento de aplicações. Como são informações de utilidade pública não estão sujeitas a licenças.

Para adequar as informações aos princípios dos dados ligados criou-se um identificador para cada pessoa desaparecida, ou recurso, sendo este acessível através de uma

URL HTTP. Os dados também foram disponibilizados no formato RDF. Por fim, foram adicionadas ligações para outros recursos, a fim de relacionar outras informações.

A etapa final deste projeto consistia em implementar uma aplicação que demonstrasse o uso de dados ligados para um exemplo prático. A ferramenta de análise de *hoax* é um protótipo com grande potencial de crescimento, é de grande utilidade para o cenário das redes sociais, onde milhares de mensagens são transmitidas de forma desnecessária por usuários sem conhecimento.

5 Conclusão

Através deste trabalho foi possível verificar as infinitas possibilidades disponíveis ao se trabalhar com dados abertos. Muito tem sido falado sobre transparência no setor público, e é fácil chegar a uma conclusão de que, a aplicação de dados abertos para este fim é altamente aconselhável, trazendo diversos benefícios ao governo, como já foi discutido no capítulo 2.

O avanço da tecnologia e o consequente surgimento de novos dispositivos conectados estão fazendo com que a *Web* faça parte, cada vez mais, da rotina das pessoas, de forma que estas tomem decisões baseadas em informações coletadas, seja nas redes sociais como também em aplicativos específicos. O uso de dados abertos para a construção de novas aplicações facilita o consumo de diferentes dados em um único serviço, favorecendo o crescimento dos *marshups*.

Através do projeto também foi possível verificar a importância dos princípios de dados ligados para a disponibilização e coleta de informações na *Web*. Tal prática influencia na forma como os dados são analisados pelo computador, proporcionando o crescimento da *Web* semântica. A conexão entre *datasets*, característica destes dados, facilita a interoperabilidade entre aplicações, o que é uma grande vantagem para os diversos serviços.

Porém, todas estas vantagens ainda não podem ser aproveitadas em sua totalidade. O Brasil ainda possui poucos *datasets* disponíveis no projeto *Linking Open Data*. Apesar de muitos projetos terem contribuído para a disponibilização neste formato ainda é necessário muito esforço para tornar isso uma realidade entre os desenvolvedores.

Através deste projeto foi possível acompanhar as etapas que envolvem a manutenção de dados abertos utilizando a tecnologia de dados ligados. Com informações disponíveis, a proposta de uma aplicação contribuiu para exemplificar uma das diversas ferramentas inteligentes que podem ser criadas com estes dados. O serviço é alimentado com informações obtidas através de raspagem de dados. Porém, projetos futuros podem criar mecanismos de inserção mais sofisticados para a alimentação da base.

O Virtuoso é uma ferramenta de grande potencial e com muitas funcionalidades a serem exploradas. Neste projeto utilizou-se o mapeamento RDF que possibilita disponibilizar dados neste formato, e também oferece uma interface para consultas SPARQL. Propõe-se em próximos trabalhos, a criação de uma base composta por triplas RDF de forma que a aplicação possa além de coletar informações participar do processo de atualização, contribuindo com o crescimento do banco de dados.

Optou-se por trabalhar com as redes sociais devido a popularidade deste serviço nos últimos tempos, e por se tratar de um meio onde o excesso de informações irrelevantes podem se tornar um incômodo para os usuários. Outro cenário em que estes dados podem ser aplicados são os serviços de e-mail, que são atacados constantemente por correntes que circulam a anos.

Existe também a possiblidade de ampliar a base de dados, alimentando-a com informações para identificar outros tipos de *hoaxes*. O caso das pessoas desaparecidas é apenas um dos vários existentes. Uma ontologia poderia ser criada para a melhor descrição deste tipo de informação.

Uma das dificuldades enfrentadas durante a busca por dados foi a falta de padronização nas páginas HTML, muitos códigos encontrados tornaram esta tarefa quase impossível. Uma opção para os sites que fornecem informações úteis para usuários seria a utilização do formato RDFa, assegurando legibilidade destes dados por máquina através da adição de semântica ao conteúdo.

Referências Bibliográficas

Adida, B.; Herman, I.; Sporny, M. ; Birbeck, M. **Rdfa 1.1 primer - rich structured data markup for web documents**, 2012.

Bennett, D.; Harvey, A. **Publishing open government data**, 2009. Disponível em: http://www.w3.org/TR/gov-data [Online; acessado em 02-outubro-2012].

Berners-Lee, T. **Linked data**, 2006. [Online; acessado em 31-Outubro-2011].

Bizer, C.; Heath, T. ; Berners-Lee, T. **Linked data - the story so far**, 2009.

Booth, D.; Haas, H.; McCabe, F.; Newcomer, E.; Champion, M.; Ferris, C. ; Orchard, D. **Web services architecture**, 2004.

Brickley, D.; Guha, R. **Resource description framework (rdf) schema specification**, 1998. Disponível em: http://www.w3.org/TR/1998/WD-rdf-schema [Online; acessado em 31-Outubro-2011].

Cyganiak, R.; Sauermann, L. **Cool uris for the semantic web**. W3c note.

Bizer, C.; Cyganiak, R. ; Heath, T. **How to publish linked data on the web**, 2007.

Eaves, D. **The three laws of open government data**, 2009. Disponível em: http://eaves.ca/2009/09/30/three-law-of-open-government-data [Online; acessado em 02-junho-2012].

Heath, T.; Bizer, C. **Linked Data: Evolving the Web into a Global Data Space**. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan and Claypool, 2011.

JSON.ORG. 1999.

Brasil, W. **Manual dos dados abertos: desenvolvedores**, 2011.

Brasil, W. **Manual dos dados abertos: governo**, 2011.

Miller, E. **An introduction to the resource description framework**, 2005. Disponível em: http://onlinelibrary.wiley.com/doi/10.1002/bult.105/full [Online; acessado em 31-Outubro-2011].

Definition, O. **Defining the open in open data, open content and open services**, 2009. Disponível em: http://opendefinition.org [Online; acessado em 31-Outubro-2011].

Team, O. S. D. **OpenLink Virtuoso Universal Server: Documentation**, 2009.

Orengo, V. M.; Huyck, C. **A stemming algorithm for portuguese language**. In: Proc. of Eigth Symposium on String Processing and Information Retrieval (SPIRE 2001) - Chile, p. 186–193, 2001.

Prud'hommeaux, E.; Seaborne, A. **Sparql query language for rdf**, 2008.

Rouse, M. **Definition: Stop word**, 2005. Disponível em: http://searchsoa.techtarget.com/definition/stop-word [Online; acessado em 02-outubro-2012].

Teixeira, R. C. **Boatos (hoax)**, 2007.

Tittel, E. 2002.

Wiki, V. O.-S. **Virtuoso open-source wiki**, 2009.

van der Vlist, E.; Ayers, D.; Bruchez, E.; Fawcett, J. ; Vernet, A. 2007.

Project, W. S. C. **Linking open data - w3c sweo community project**, 2011.

Group, O. G. W. **Open government data**, 2007. Disponível em: http://www.opengovdata.org [Online; acessado em 31-Outubro-2011].