



Uma Aplicação Web com Visualização de Grafos para
Documentos RDF

Por

José Olimpio Mouzinho Negrão

Trabalho de Graduação



Universidade Federal da Bahia
wiki.dcc.ufba.br/DCC/

SALVADOR, Dezembro/2018



Universidade Federal da Bahia
Departamento de Ciência da Computação

José Olimpio Mouzinho Negrão

Uma Aplicação Web com Visualização de Grafos para Documentos RDF

Trabalho apresentado ao Departamento de Ciência da Computação da Universidade Federal da Bahia como requisito parcial para obtenção do grau de bacharel em Sistemas de Informação.

Orientador: *Prof. Dr. Danilo Barbosa Coimbra*
Co-Orientador: *Prof. Dr. Frederico Araújo Durão*

SALVADOR, Dezembro/2018

*Dedico esta monografia à minha família, amigos e
professores que me deram todo o apoio necessário para
chegar até aqui.*

*Descobrir consiste em olhar para o que todo mundo está vendo e
pensar uma coisa diferente*

—ROGER VON OECH

Resumo

O advento da internet gerou uma explosão na produção de dados dos mais diversos tipos, principalmente na Web. Para utilizá-los de maneira mais adequada e organizada foi necessário a criação da Web Semântica, a qual deu origem a vários padrões. Uma vez garantida essa organização surge o desafio de como tornar mais fácil a compreensão desses dados. Uma das maneiras mais eficazes de aquisição de informação é por meio de imagens. Logo, podemos representar esses dados visualmente para acelerar a interpretação desses dados, que devido a sua natureza podem ser representados utilizando grafos. Com esses desafios como contexto, o objetivo do presente trabalho consiste no desenvolvimento de uma aplicação Web que utilizando os conceitos da Web Semântica e da Visualização de Dados, usando como base dados dispostos em padrão RDF. A visualização desenvolvida, nomeada de GraVisRDF, permite ao usuário a livre interação com o grafo gerado, suprimindo assim as limitações de outras ferramentas disponíveis na área. Ao final, foi realizada uma avaliação descritiva e exploratória entre a GraVisRDF e tais ferramentas, evidenciando que a ferramenta proposta é mais completa em termos de funcionalidades do que as ferramentas analisadas.

Palavras-chave: Visualização, Grafos, Visualização de Dados, Web Semântica.

Abstract

The advent of the Internet generated an explosion in the production of data of the most diverse types, mainly in the Web. To use them in a more adequate and organized way, it was necessary to create the Semantic Web, which gave rise to several standards. Once this organization is guaranteed, the challenge arises of how to make the data easier to understand. One of the most effective ways of acquiring information is through images. Therefore, we can represent these data visually to accelerate the interpretation of these data, which due to their nature can be represented using graphs. With these challenges as context, the objective of the present work is the development of a Web application that uses the concepts of Semantic Web and Data Visualization, based on data arranged in RDF standard. The developed visualization, named GraVisRDF, allows the user to freely interact with the generated graph, thus supplying the limitations of other tools available in the area. Lastly, a descriptive and exploratory evaluation was carried out between GraVisRDF and some existent tools, evidencing that the proposed tool is more complete in terms of functionalities than the analyzed tools.

Keywords: Visualization, Graphs, Data Visualization, Semantic Web.

Sumário

Lista de Figuras	xv
Lista de Tabelas	xvii
Lista de Acrônimos	xix
Lista de Códigos Fonte	xxi
1 Introdução	1
1.1 Motivação	2
1.2 Objetivos	2
1.3 Estrutura	3
2 Web Semântica	5
2.1 Arquitetura e Padrões	6
2.1.1 <i>Resource Definition Framework</i> (RDF)	6
2.1.2 <i>SPARQL Protocol and RDF Query Language</i> (SPARQL)	8
2.1.3 <i>Web Ontology Language</i> (OWL)	9
2.1.4 Estrutura na rede semântica	10
2.2 Linked Open Data	12
2.3 Projeto na área Web Semântica - Dbpedia	13
2.4 Sumário	15
3 Visualização de Dados	17
3.1 Pipeline de Visualização	17
3.2 Atributos Visuais	20
3.3 A Taxonomia de Keim	22
3.4 Visualização de Grafos	24
3.5 Sumário	27
4 Uma Aplicação Web com Visualização de Grafos para Documentos RDF	29
4.1 Visualização de Grafos para Documentos RDF	29
4.2 Apresentação da Aplicação	32
4.3 Avaliação	37
4.4 Discussão	38

4.5	Sumário	41
5	Conclusão	43
5.1	Contribuições	43
5.2	Pontos de Melhorias e Possíveis Incrementos	43
5.3	Sumário	44
	Referências Bibliográficas	45

Lista de Figuras

2.1	Exemplo de triplas RDF (Laufer, 2018)	7
2.2	Exemplo de uma representação RDF (Laufer, 2018)	8
2.3	Exemplo de resposta do Snorql Openlink (2013)	9
2.4	Camadas na rede semântica. Figura elaborada pelo autor de acordo com a publicação de (Horrocks <i>et al.</i> , 2005).	11
2.5	Diagrama da nuvem dos dados ligados (2014)(Andrejs Abele, 2018). . .	13
2.6	Diagrama da nuvem dos dados ligados (2018)(Andrejs Abele, 2018). . .	14
2.7	Ilustração da arquitetura do DBPédia (Lehmann <i>et al.</i> , 2015).	14
3.1	Fluxograma de Pipeline traduzido de (Ward <i>et al.</i> , 2015).	18
3.2	Fluxograma de Pipeline traduzido de Telea (2014).	19
3.3	Atributos visuais, traduzido de Bertin (1983).	21
3.4	Demonstrativo da taxonomia de keim, traduzido de Ward <i>et al.</i> (2015). .	23
3.5	Exemplo de árvore genealógica.	24
3.6	Exemplo de modelo de Grafo Espalhado, representando a densidade de informação semelhante a uma imagem térmica (Telea, 2014).	25
3.7	Projeto <i>Nabusca</i> , reprodução (Carol Cavaleiro, 2018).	26
3.8	Grafos representando como dois sistemas operam, traduzido de Holten (2006).	27
4.1	Uma visão geral da navegação gerada pelos passos anteriores.	30
4.2	Diagrama de atividades da ferramenta proposta.	31
4.3	Tela da aplicação com a visualização do recurso “ <i>Disownment</i> ”.	32
4.4	Edição de nós (1).	33
4.6	Edição de Aresta(A). Janela para edição da aresta(B).	33
4.5	Edição de nós etapas (2) e (3) - Janela contendo as opções para edição do nó.	34
4.7	Adição de nós em área vazia.	34
4.8	Exemplo de nó expandido, gerando outro grafo.	35
4.9	Agrupamento de instâncias pertencentes a mesma ontologia (“ <i>Subject</i> ”) e à direita é mostrada a Separação do agrupamento.	36
4.10	Um exemplo de nós expandidos, com o grafo inicial marcado em vermelho. .	36

Lista de Tabelas

4.1	Características de visualização de grafos existentes.	38
4.2	Comparativo entre funcionalidades.	40

Lista de Acrônimos

GraVisRDF	Graph based Visualization of RDF documents
HTTP	HyperText Transfer Protocol
HTML	Hypertext Markup Language
LOD	Linked Open Data
OWL	Ontology Web Language
RDF	Resource Description Framework
SPARQL	SPARQL Protocol and RDF Query Language
URI	Universal Resource Identifier
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

Lista de Códigos Fonte

2.1 Exemplo de consulta na linguagem SPARQL	8
---	---

1

Introdução

Tudo o que um homem pode imaginar, outros homens poderão realizar.

—JÚLIO VERNE

Dados são registros de observações do mundo real, medidas ou relatos, os mais diversos elementos podem ser classificados pela palavra “Dado”. Como dado é um fato, é a matéria prima com o qual realizamos discussões, inferências e cálculos ou seja, através desses processos, esse fato adquire relevância e sentido se tornando informação (Ackoff, 1989). A criação de informação vem evoluindo em conjunto com a humanidade, que desde o início da raça humana, trocar informações é uma das bases que explicam o sucesso da espécie. Entretanto o advento da internet, revolucionou a maneira como geramos, compartilhamos e armazenamos esses dados. Esse volume gigantesco de dados brutos possuem em alguns casos um valor real, podendo até após processamento se tornar informações valiosas.

Segundo Wellman (2013), esse grande volume de registros pode ser chamado de “Mar”, a Web Semântica entra neste contexto provendo uma maneira de organizar esses dados e informações, permitindo que os mesmos possam ser padronizados e acessados por diversos usuários e aplicações. Utilizando estes conceitos existem projetos que varrem a Web recolhendo esses dados e processando-os de maneira a torná-los úteis para inferências tanto por humanos quanto por máquinas. Seguindo esses processos dados são capturados, classificados e organizados, em especial os relacionados ao contexto da Internet.

Mas qual seria o próximo passo? Como exibir estes dados? A resposta para estas perguntas pode ser encontrada na área de Visualização de Dados, que é voltada para representação de informação através do mapeamento de dados utilizando atributos visuais

(Van Wijk, 2005). Recursos estes que nos ajudam a interpretar essa gama de informações disponíveis apresentando-as de maneira coerente Ward *et al.* (2015). Em específico, para tratar da organização dos dados na Web utilizando conceitos semânticos, faz-se necessário visualizar tais estruturas por intermédio de técnicas de Visualização de Grafos (estruturas matemáticas aqui usadas para representar dados abstratos de forma hierárquica) (Consortium *et al.*, 2014). No decorrer deste trabalho será discutido os conceitos pertinentes a esses processos e como os mesmos são utilizados para atingir o fim, uma contribuição para as áreas de Web Semântica e Visualização, em forma de uma aplicação Web.

1.1 Motivação

Devido a esse grande fluxo de informações, principalmente na Web, alinhado com uma necessidade de produção cada vez maior, se torna vital que os dados a serem processados estejam dispostos de maneiras a facilitar as manipulações e inferências realizadas com e a partir destes dados. Assim, esta matéria prima deve estar organizada de tal modo a prover informações úteis. Ao ser apresentada essa necessidade, algumas ferramentas (que possuem diversas limitações) auxiliam na manipulação e visualização destes dados, no caso, para representar dados em um padrão RDF de maneira visual que proporcione um entendimento claro e rápido dos dados originais.

1.2 Objetivos

O objetivo geral deste trabalho é o desenvolvimento de uma solução para o problema supracitado. Para isso, faz-se necessária a proposta de uma aplicação que apresente de maneira eficiente e robusta documentos que utilizem padrões da Web Semântica. Para que isso torne-se realidade, recorrer-se-á a área de Visualização para auxiliar na apresentação de um grafo visual que possua característica hierárquica. Ainda, elementos da visualização exploratória devem permitir que o usuário possa interagir e customizar os componentes e suas informações.

Dentre os objetivos específicos, destacam-se:

- Estudar como os dados são estruturados na Web Semântica.
- Entender de que maneira a visualização de grafos auxilia no entendimento dos dados.
- Investigar as limitações decorrentes da recuperação desses dados.

- Estudar as funcionalidades de bibliotecas de visualização e como respondem a tentativas de adaptação para novos conceitos *Vis* e *React*.

1.3 Estrutura

Os próximos capítulos estão organizados da seguinte maneira: o Capítulo 2 apresenta os conceitos teóricos usados neste trabalho referentes a Web Semântica. O Capítulo 3 mostra os conceitos de que embasam a visualização de dados, em especial a visualização de grafos. Posteriormente, o Capítulo 4 traz detalhes da proposta e como ela foi desenvolvida. Por fim, o Capítulo 5 faz uma tomada dos objetivos alcançados e quais possíveis melhorias poderiam ser realizadas no escopo desse projeto.

2

Web Semântica

Atualmente a rede de comunicação tem sido expandida e vem gerando um fluxo de informações imensurável. Embora a estruturação desses dados seja possível, ainda assim as informações de natureza semântica serão de difícil processamento para máquinas. Para ilustrar esse problema, podemos usar um cenário onde um usuário deseja marcar uma consulta, assim, ao buscar por um médico em uma página de uma clínica, é redirecionado para uma página pessoal do mesmo onde há um *link* para um cronograma contendo a clínica e os horários que o mesmo está atendendo. Todas essas informações podem ser encontradas separadamente na rede, mas a associação delas que dá origem a uma informação útil, e essa análise semântica se dá de forma dificultosa por uma máquina, uma vez que não há uma estrutura comum as entidades presentes nesse fluxo, tornando o processo ineficiente e improdutivo.

Estabelecido o objetivo da internet, ou seja, a criação e interação de dados e informações, é necessário então para fins de extensão a aplicação da Web Semântica, para que seja possível um melhor aproveitamento dos itens disponibilizados por essa incrível ferramenta. O conceito supracitado tem como função principal prover formatos para interação e integração de dados de diferentes fontes W3C (2001), assim a web tradicional preserva sua função como meio de disseminação de documentos, enquanto informações que se relacionam com objetos do mundo real são armazenadas em sua contraparte semântica. Dessa forma, retornando ao exemplo anterior, a máquina poderá anexar as informações do médico, os seus horários e lugares de atendimento, podendo encaixar a consulta no horário mais próximo disponível do usuário.

2.1 Arquitetura e Padrões

De acordo com Berners-Lee *et al.* (2001), a efetividade da Web Semântica depende da capacidade das máquinas no que diz respeito ao acesso e manipulação de estruturas de informação, regras de inferência e seus respectivos dados, uma vez que essas condições são atendidas o problema passa a ser mais complexo: como representar este conhecimento? Originalmente, o conceito foi desenvolvido com a centralização como abordagem, de forma que era requerido que as partes envolvidas compartilhassem as mesmas definições, não permitindo a mínima alteração nos conceitos comuns ou hierárquicos.

Entretanto, com a quantidade em que informação era produzida isso se tornou obsoleto, o que hoje é ainda mais notável, uma vez que atualmente a velocidade de produção de informação está em ordens de grandeza maior do que quando esse conceito foi desenvolvido. Centralizar esse conteúdo, que existe em múltiplas línguas, vem acompanhada de uma complexidade considerável. Contrastando com suas primeiras versões, a Web semântica atualmente cria linguagens para regras, as quais expressam tanto quanto necessário para que o escopo da Web amplie o seu alcance. Com esse sistema descentralizado, embora nem todas as buscas sejam satisfeitas em sua totalidade, permite que regras sejam implementadas com o objetivo de inferir sobre os dados disponíveis e assim escolher o melhor curso de ação para tentar responder com a maior riqueza possível a pergunta.

Seguindo esses ideais, o *World Wide Web Consortium (W3C)* desenvolveu uma série de padrões e formatos para a representação de informações na Web, ou seja em documentos, textos entre outros, com o objetivo de tornar a compreensão dos mesmos por máquinas possível. Para esse fim é utilizado uma miríade de tecnologias que trabalham em conjunto a fim de ampliar o máximo possível o alcance da Web Semântica, dentre elas, as que serão discutidas neste trabalho são: Resource Description Framework (RDF), SPARQL Protocol and RDF Query Language (SPARQL), Ontology Web Language (OWL).

2.1.1 *Resource Definition Framework (RDF)*

Ainda que não seja uma das especificações mais famosas da W3C, a RDF é uma das mais antigas, uma vez que um dos primeiros registros desse conceito datam de 1997. Lassila *et al.* (1998) estabeleceram os fundamentos com os quais o RDF manipularia a metadata, regras que promovem a troca de informações entre sistemas autônomos. Apesar das múltiplas interações que o framework teve, esse permanece sendo seu objetivo.

Após algumas interações, um grupo de trabalho voltado para o conceito de RDF foi criado, e em 2001, Tim Berners-Lee, Lassila e Swick publicaram um artigo contendo a seguinte proposta “A Web Semântica vai trazer uma estrutura para os conteúdos úteis das páginas, criando um ambiente onde os agentes de software vagando de página à página possam prontamente executar tarefas sofisticadas para os usuários. Um desses agentes, por exemplo, poderia varrer uma página de uma clínica não vai somente extrair as informações das *keywords* “Tratamento, remédio, físico, terapia” (que podem estar inseridas no corpo do objeto) como também vai recuperar que o Dr. Hartman trabalha nesta clínica às Segundas, Quartas e Sextas, que o script recebe informações de datas no padrão ano-mês-dia para retornar as consultas marcadas.” Berners-Lee *et al.* (2001).

Com esse foco, o framework facilita a junção de dados mesmo que seu *schema* não seja totalmente compatível, além de garantir a possibilidade de que essa evolução ocorra sem a interferência direta dos consumidores, como descrito em W3C (2014). No RDF os links são estruturados de forma estendida, utilizando os Universal Resource Identifier (URI)s para identificar as relações entre qualquer objeto, conectando assim pelo menos três itens, formando uma tripla.

I.S.B.N.	Título	Autor	Editadora	Páginas
9788532291448	Vinte Mil Léguas Submarinas	Júlio Verne	FTD	256

Figura 2.1 Exemplo de triplas RDF (Laufer, 2018)

Já a URI ou em português Localizador de Recursos Universal tem como função facilitar as buscas evitando literais que possam vir a representar atributos. Graças a isso ao utilizar a URI, o sistema pode por exemplo, isolar um predicado “Autor”, separando-o do autor do livro ou o autor de resenhas sobre o mesmo na mesma página. Assim, uma vez que os elementos da tripla estejam devidamente isolados e relacionados, podemos montar uma estrutura que expressa uma declaração semântica, estrutura essa formada por “sujeito, predicado e objeto”. Dessa maneira, é possível gerar um grafo como apresentado na Imagem 2.2, onde a elipse (**livro**, em roxo claro) é um nó que representa o “Sujeito”, acompanhado por três ligações/arestas que representa os “Predicados” (**título, isbn, autor**), os quais fazem suas respectivas conexões para os nós, ou “Objetos” (**Vinte Mil léguas Submarinas, 9788532291448, Júlio Verne**).

```

1 PREFIX owl: <http://www.w3.org/2002/07/owl#>
2 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
3 PREFIX : <http://dbpedia.org/resource/>
4 PREFIX dbpedia2: <http://dbpedia.org/property/>
5 PREFIX dbpedia: <http://dbpedia.org/>
6
7 SELECT ?musics
8 WHERE {
9 ?musics dbo:artist dbr:John_Lennon.
10 ?musics dbo:artist ?Artist
11 }LIMIT 10

```

Código Fonte 2.1 Exemplo de consulta na linguagem SPARQL

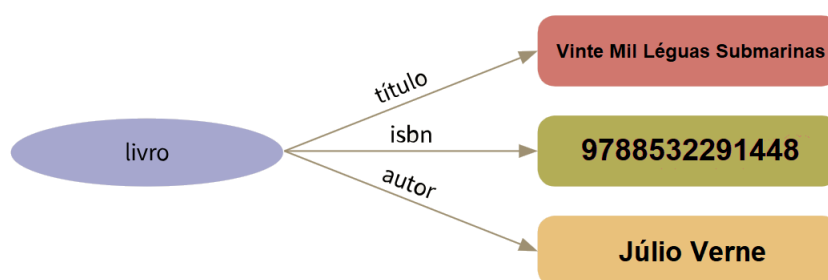


Figura 2.2 Exemplo de uma representação RDF (Laufer, 2018)

2.1.2 SPARQL Protocol and RDF Query Language (SPARQL)

SPARQL ou “SPARQL Protocol and RDF Query Language” é uma linguagem de busca para as estruturas RDF W3C (2013). Assim, pode-se criar estruturas ou *queries* para buscar ou inferir informações de diversos conjuntos de dados, desde que os mesmos estejam em triplas, permitindo-se a aplicação de filtros, limitadores e ordenadores para organizar e refinar os dados retornados. Diferente das linguagens voltadas a bancos de dados relacionais ou seja, o tipo de cada célula pode ser entendido como um objeto, sendo implicado ou definido pelo predicado informado pela URI. O exemplo do Código Fonte 2.1 demonstra a consulta de dados realizada na base de dados da DBPedia (2014) e a Figura 2.3 representa a resposta apresentada pelo trecho de Código 2.1.

SPARQL results:

Artist	musics
:John_Lennon ↗	:I_Know_(I_Know) ↗
:John_Lennon ↗	:Look_at_Me_(John_Lennon_song) ↗
:John_Lennon ↗	:How_Do_You_Sleep%3F_(John_Lennon_song) ↗
:John_Lennon ↗	:Double_Fantasy ↗
:Yoko_Ono ↗	:Double_Fantasy ↗
:John_Lennon ↗	:Shaved_Fish ↗
:John_Lennon ↗	:Walls_and_Bridges ↗
:John_Lennon ↗	:Cleanup_Time ↗
:John_Lennon ↗	:Nobody_Loves_You_(When_You're_Down_and_Out) ↗
:John_Lennon ↗	:Out_the_Blue ↗

Figura 2.3 Exemplo de resposta do Snorql Openlink (2013)

2.1.3 *Web Ontology Language (OWL)*

De acordo com W3C (2009), a “*W3C Web Ontology Language*” é uma linguagem semântica voltada para a Web, projetada para representar conhecimentos complexos sobre variados itens. Como esperado, a OWL é uma linguagem lógica criada para facilitar o entendimento das máquinas no que compete o processamento de dados, por exemplo, usada para verificar a consistência dos conhecimentos ou para inferir informação implícita em dados explícitos. Esses documentos pertencentes a essa linguagem são chamados de Ontologias. A OWL pertence ao grupo de tecnologias voltadas para a web semântica da WC3, assim como a RDF, SPARQL, entre outras, foi desenvolvida pelo atualmente descontinuado “W3C OWL Working Group”.

Basicamente, uma ontologia é uma documentação que formaliza os relacionamentos entre os itens, garantindo-os uma hierarquia semelhante a disposição de classes na programação orientada a objeto. Essa linguagem de marcação pode expressar objetos (também chamados de “indivíduos”) que pertencem a uma determinada classe (conceito), e suas respectivas propriedades, equivalências podem ser estabelecidas em classes e em propriedades. Assim, é natural que essa linguagem permita regras de inferência. A OWL estende sua aplicação a outras tecnologias, trabalhando em conjunto, por exemplo, com a RDFs (**Resource Description Framework Schema**) pode-se:

- Declarar classes como País, Pessoa, Estudante e Brasileiro;
- Definir que Estudante é uma subclasse de Pessoa;

- Definir que Brasil e Estados Unidos são ambos instâncias da classe País;
- Declarar Nacionalidade que é uma propriedade que relaciona Pessoa(domínio) e País(range);
- Definir que Idade é uma propriedade de Pessoa sendo ela seu domínio e inteiros, seu range;
- Definir que José é uma instância da classe Brasileiro, que sua idade tem o valor de 26

Com a **OWL** podemos expandir as informações das seguintes forma:

- Definir que País e Pessoa são classes disjuntas;
- Definir que Brasil e Estados Unidos são indivíduos distintos;
- Declarar “tem Cidadão” como uma propriedade inversa da “Nacionalidade”;
- Declarar a classe “Apátrida” para definir precisamente as instâncias da classe Pessoa que não tem nenhum valor para “Nacionalidade”;
- Declarar a classe “MultiplasNacionalidades” para definir os membros da classe Pessoa que tem pelo menos dois valores para a propriedade “Nacionalidade”;
- Definir que a classe Brasil existe para representar os membros da classe Pessoa que possuem “Brasil” como um valor da propriedade “Nacionalidade”;
- Definir que a propriedade “Idade” é funcional.

O breve exemplo de integração acima demonstra como a OWL é uma linguagem sofisticada. Unindo RDF/XML em suas interações e uma sintaxe de fácil leitura, essa ontologia garante o desempenho e flexibilidade que é necessária para o processamento de informações na Web semântica atualmente.(Horrocks *et al.*, 2003)

2.1.4 Estrutura na rede semântica

Para atingir seu objetivo, a Web Semântica é dividida em camadas, assim em cada uma é aplicada um dos conceitos descritos acima Berners-Lee *et al.* (2001). Essas camadas são divididas como na figura 2.4. De baixo para cima, a **Unicode / URI** fornece a interoperabilidade entre as camadas no que toca o endereçamento/nomeação

de elementos e codificação dos caracteres, no caso **Unicode** fornece uma maneira de nomear os itens sem ambiguidade, enquanto a **URI** fornece uma maneira de identificar os elementos requisitados, seguidas pela camada denominada **XML + namespace + xmlschema** que contribui em relação a sintaxe da descrição dos recursos usados na Web Semântica. A **Extensible Markup Language (XML)** e a XML “*Schema*” fornecem uma representação sintática de recursos de maneira independente da plataforma, além de restrições para as definições da mesma, no caso a “*Schema*” é uma gramática para a classes de documentos XML. Os espaços de nome (**namespaces**) são responsáveis por qualificar os nomes de elementos e atributos, quando associados a XML, os nomes são representados, qualificados e identificados por URI, assim tornando possível distinguir entre dois elementos com o mesmo nome, mas que pertencem a esquemas diferentes.

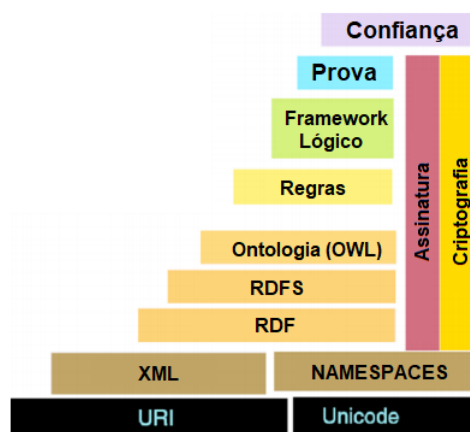


Figura 2.4 Camadas na rede semântica. Figura elaborada pelo autor de acordo com a publicação de (Horrocks *et al.*, 2005).

A camada **RDF** e **RDF Schema** entra nessa pirâmide para auxiliar na representação dos metadados dos recursos, possuindo uma sintaxe semelhante a XML e uma linguagem de definição de esquemas para vocabulários (termos). Possibilitando uma base para aplicação das ontologias, descrevendo classes e propriedades, uma vez construídas, as camadas anteriores fornecem suporte para o próximo passo nos vocabulários, processando e integrando as informações existentes independente da definição (ou falta dela) que a mesma possa vir a ter. Para isso a *Web Ontology Language (OWL)* inclui no vocabulário da *RDF schema* elementos com maior poder de relação, afim de ampliar a expressividade dos mesmo e possibilitar a inferência de informação baseada em dados existentes. Permeando essas camadas existem as camadas de segurança, no caso, o documento é **Criptografado** e caso esse documento resultante seja alterado isso será detectado pela **Assinatura**.

A Camada Lógica garante o suporte para a descrição das regras que regem as relações

apoiadas pelos conceitos da ontologia. Por fim, a camada de Prova e Confiança fornecem o apoio para a execução dessas regras, além de avaliar os resultados e a confiabilidade dessa execução, como são camadas muito novas, ainda estão em desenvolvimento e dependem da maturidade das camadas inferiores (Ribeiro, 2008).

2.2 Linked Open Data

Linked Data pode ser descrita como o uso da Web para criar ligações entre informações de diferentes fontes, que podem ser tão distintas quanto dois bancos de dados, mantidos por organizações diferentes em lugares distantes um do outro, ou simplesmente dados heterogêneos de um sistema interno (Heath and Bizer, 2011). Tecnicamente, “Dados ligados” refere-se a informações disponibilizadas na internet que podem ser lidas por máquinas, embora as unidades primárias sejam ligadas através de hipertexto (com documentos HTML), já a *Linked data* se baseia em documentos no formato RDF, entretanto em vez de se limitar em apenas conectar esses documentos a tecnologia visa dar sentido a essas conexões ligando-as arbitrariamente a coisas no mundo real Berners-Lee *et al.* (2001). De acordo Berners-Lee *et al.* (2006), algumas “regras” devem ser seguidas para publicar dados de maneira que os mesmos se tornem parte de um único espaço global de informações:

1. Usar URIs como nomes para coisas;
2. Usar HTTP URIs para que outras pessoas possam procurar esses nomes;
3. Quando alguém procurar os nomes, prover informações úteis ao usuário, utilizando os padrões (RDF, SPARQL)

Esses são conhecidos como os “Princípios da *Linked Data*”, e definem a receita básica para publicar e conectar dados usando a infraestrutura da Web, levando em consideração a sua arquitetura e padrões. Com base nesses princípios, Berners-Lee (2006) utilizando dados governamentais, desenvolveu um sistema para avaliá-los, o que iniciou em 2007, uma colaboração com o objetivo de mapear dados com licença aberta, esse grupo incluía fontes como a Wikipedia¹, Geonames², Wordnet³ entre outros. Esse projeto tem diversas interações, como ilustrado nas Figuras 2.5 e 2.6, representando o início do projeto e o estado atual das conexões, respectivamente.

¹<https://www.wikipedia.org>

²<http://www.geonames.org>

³<https://wordnet.princeton.edu>

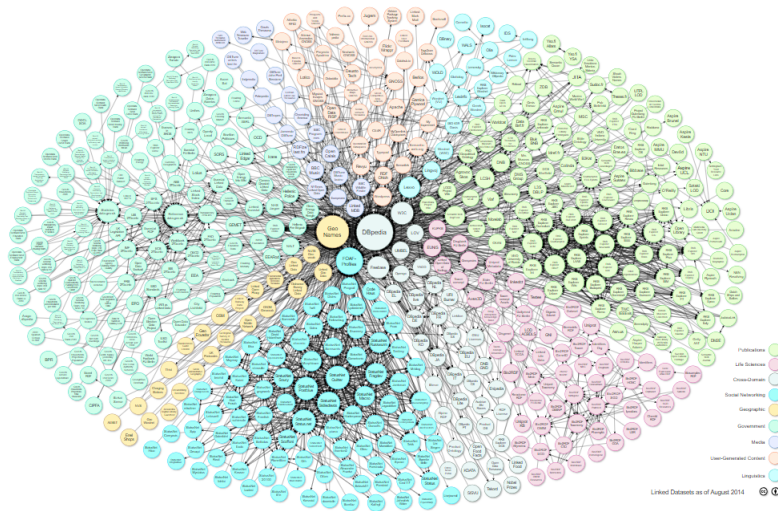


Figura 2.5 Diagrama da nuvem dos dados ligados (2014)(Andrejs Abele, 2018).

2.3 Projeto na área Web Semântica - Dbpedia

O projeto colaborativo denominado de Dbpedia (que é um neologismo de “DB” oriundo da palavra “Database” e “Pédia” derivada da palavra “Wikipédia”) tem como objetivo a extração e estruturação e disponibilização das informações contidas na Wikipédia Bizer *et al.* (2009) além de incorporar links e informações de outros *datasets*, garantindo assim o seu uso por diversas aplicações, incluindo a que será discutida neste trabalho.

A DBPédia é atualmente um dos projetos da área de Web Semântica mais importantes. Este projeto realiza um esforço contínuo na estruturação e disponibilização de dados, seguindo os princípios de *Linked Data*, e também é um núcleo nos progressos do conceito de *Linked Open Data*, já que implementa fortemente os conceitos de Web Semântica durante a estruturação e armazenamento dessas informações(utilizando RDF e SPARQL). Com isso, torna-se possível o acesso e manipulação por fontes externas Bizer *et al.* (2009), suprimindo a necessidade de haver um buscador de texto livre para recuperar informações.

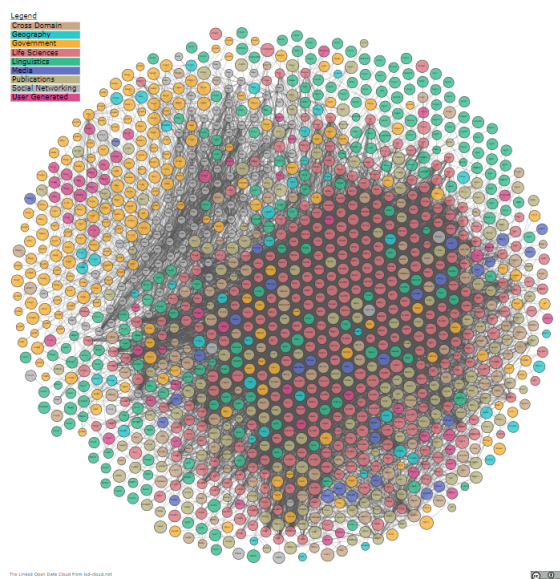


Figura 2.6 Diagrama da nuvem dos dados ligados (2018)(Andrejs Abele, 2018).

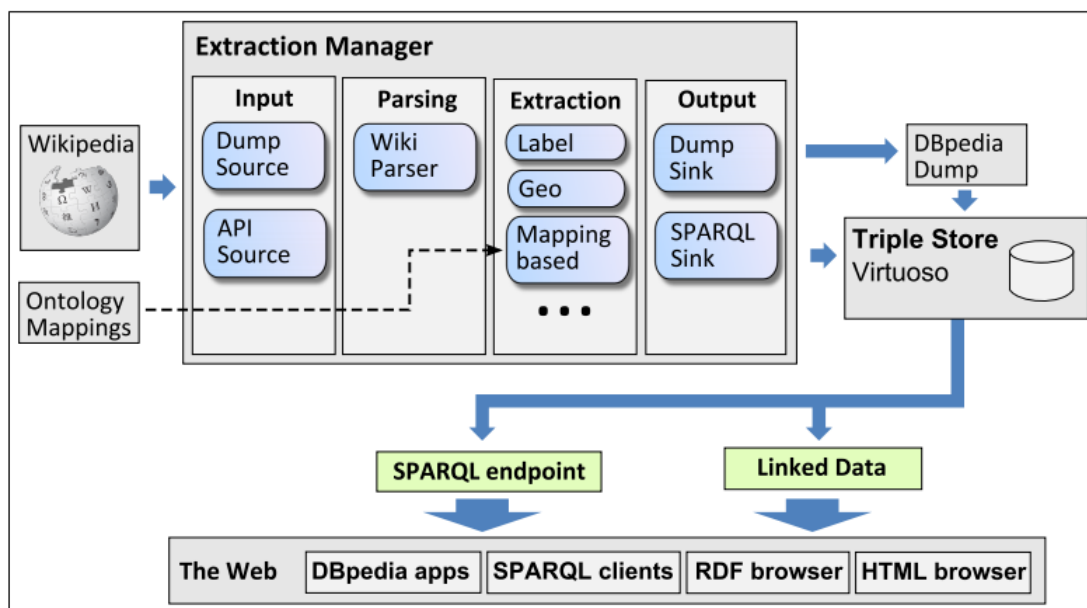


Figura 2.7 Ilustração da arquitetura do DBPédia (Lehmann *et al.*, 2015).

Como a iniciativa compõe o movimento do LOD, os dados do Dbpedia podem ser importados por aplicações *third party* utilizando a licença aberta. O projeto faz uso da plataforma do *Virtuoso Universal Server*⁴ para prover os dados de RDFs através de uma interface e um *endpoint* em SPARQL.

⁴<https://virtuoso.openlinksw.com>

2.4 Sumário

Neste capítulo foram discutidos os conceitos que baseiam a web semântica, dando ênfase aos elementos que compõem o projeto, e que vieram a ser aplicados na prática. As tecnologias que envolveram o desenvolvimento do trabalho foram descritas e brevemente explicadas e exemplificadas. Para ilustrar casos de sucesso envolvendo esses conceitos, foi descrito o trabalho na área, Dbpedia que foi utilizado como base de dados para este projeto.

3

Visualização de Dados

Imagens tem sido usadas como ferramentas de comunicação mesmo antes da formalização de linguagens escritas, o que pode ser ilustrado por achados como pinturas rupestres (Schramm, 1988). A razão para a consistência no uso deste tipo de ferramenta se dá ao fato de que a mesma, quando usada corretamente, pode expressar muitas informações que são absorvidas mais rapidamente que a escrita por se tratar de um processamento humano que é realizado simultaneamente não se limitando a coisas como entendimento sequencial e ou línguas, por exemplo independente da linguagem do usuário o entendimento aproximado de um mapa pode ser atingido apenas com imagens. Como descrito Ward *et al.* (2015) a *visualização* pode ser entendida como a transmissão de informação usando representações gráficas.

Diferente das produzidas anteriormente, as visualizações atualmente se diferenciam em ambos níveis, qualitativos e quantitativos, além de claro o objetivo de seus criadores. Com isso em mente a *Visualização Científica* tem como meta representar dados e resultados complexos, com apoio de técnicas de computação gráfica, de forma a facilitar o entendimento dos mesmos pelos usuários (Korfhage, 1995).

3.1 Pipeline de Visualização

Primariamente o objetivo da visualização é o de criar imagens que comunicam diversos tipos de informações sobre um determinado processo. Para isso é necessária uma sequência de passos a fim de preparar essas informações, manipulando-as para serem compreendidas visualmente, como todo processo, pode-se separar esses passos de forma a diminuir a complexidade dos mesmos. Devido a essa natureza modular, a decomposição do processo pode ser visto como um duto ou *Pipeline*, formado por seções, cada uma regendo uma operação de transformação de informação específica. Dessa forma a

informação é capturada e inserida nesse esquema de transformações até que seja gerada uma imagem de saída (Telea, 2014).

Essa modularização varia de autor para autor, como por exemplo as Figuras 3.1 e 3.2. Segundo o pipeline de Ward *et al.* (2015), as etapas de visualização são compreendidas em:

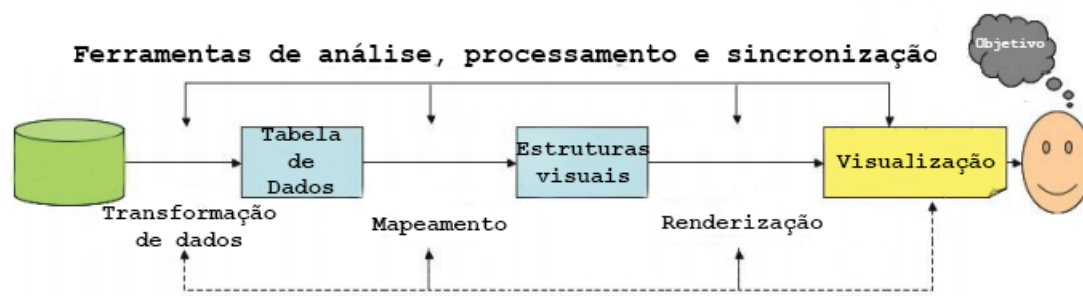


Figura 3.1 Fluxograma de Pipeline traduzido de (Ward *et al.*, 2015).

- **Processamento e Transformação de dados**

O ponto de início da representação das informações, capturando e transformando o dado base de forma que o mesmo possa ser utilizado nos sistemas de visualização. Primeiro é necessário verificar se os dados podem ser mapeados para os atributos visuais. Em seguida é necessário filtrar os principais problemas presentes nesse processo por exemplo, dados faltantes, incompletos, sujos ou mesmo grandes demais para serem processados. Por exemplo, um dado que precise de mais espaço de memória para ser carregado do que há disponível para o sistema.

- **Mapeamento de visualizações**

Uma vez que os dados estão disponíveis, é possível decidir qual tipo de técnica de visualização será utilizada, mapeando esses dados para representações geométricas ou ainda a cores e sons. É necessário um cuidado considerável nessa parte, uma vez que desenvolver visualizações sem sentido é muito fácil, ou seja, desenvolver visualizações que não expressam nenhuma informação coerente ou útil. Aqui ficam claros os princípios das técnicas, Expressividade e Efetividade, para apoiar isso é necessário o desenvolvimento de padrões e métricas.

- **Renderização**

Este é o estágio final do preparo da visualização, o mapeamento direto de dados

filtrados, utilizando as técnicas decididas no passo anterior através de interfaces computacionais. Após isso ocorrem os ajustes de parâmetros de visualização.

Ainda podemos observar o mesmo processo mas de uma outra maneira, apresentada pelo autor Telea (Telea, 2014) na Figura 3.2, onde o processo é decomposto, em mais passos, de outro modo:

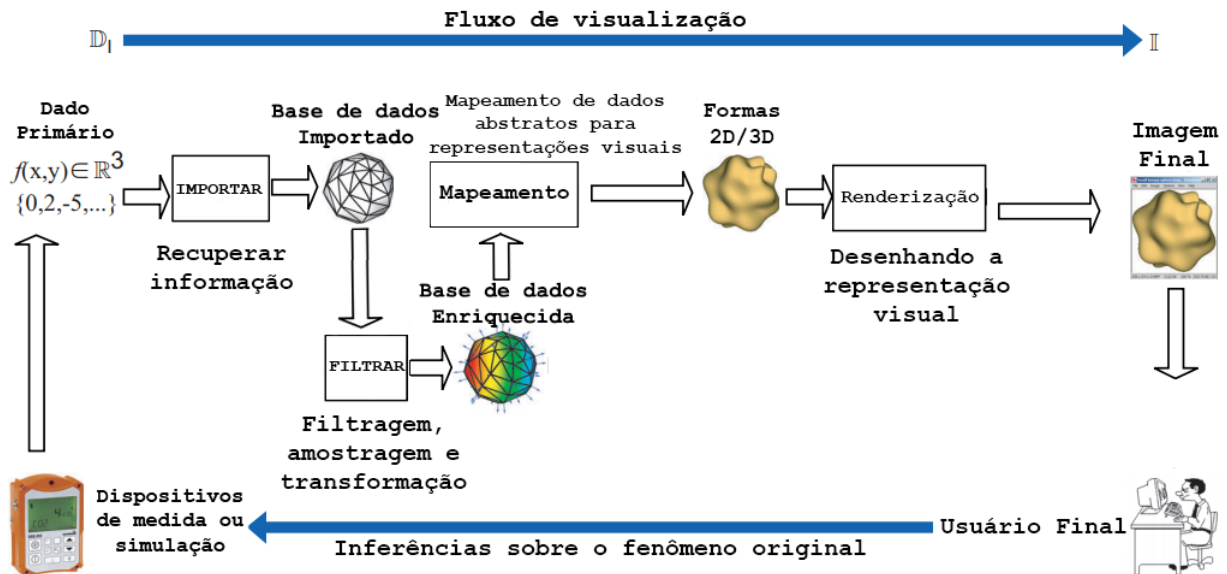


Figura 3.2 Fluxograma de Pipeline traduzido de Telea (2014).

- **Importação**

Primeiro capturar os dados originais, é necessário um cuidado maior aqui pois a definição dos dados escolhidos afeta diretamente os próximos passos nesse modelo, então ao adquirir essas informações. Verificam-se as incongruências da base dados a fim de garantir que a imagem resultante seja uma reflexão fiel desses dados.

- **Filtragem e enriquecimento**

Uma vez a base de dados capturada, é necessário separar as informações mais importantes, confirmar lacunas entre informações e tentar realizar simulações ou amostragens. Essa etapa visa entregar a próxima um conjunto de dados o mais completo e preciso possível.

- **Mapeamento**

Com a base de dados enriquecida, é iniciado o mapeamento dos dados abstratos para

uma representação visual rudimentar, focando nos dados que tem características não visuais e convertendo-os para o visual.

- **Renderização:**

Neste último passo, a visualização rudimentar é gerada pelo mapeamento e aplica os parâmetros de visualização específicos para usuários, por exemplo: iluminação e ponto de vista. Esses pontos são levados em consideração pois para aplicações de visualização esses parâmetros são fundamentais para facilitar a interação dos usuários com o objeto final do processo.

Como pode-se observar, a abordagem de Telea (2014) aparenta ser mais completa que a de Ward *et al.* (2015), uma vez que a primeira possui passos extras o que possibilita uma melhor modularização entre os passos, por isso o modelo de Telea (2014) foi considerado a melhor escolha e foi utilizado como inspiração para a aplicação desenvolvida.

3.2 Atributos Visuais

Informações podem ser apresentadas de várias formas, devido a maneira como o ser humano observa esses dados. Importante considerar as diversas formas de interpretação, para isso existem alguns atributos visuais que regem o mapeamento dos dados na visualização de informação. Segundo Ward *et al.* (2015), os principais atributos visuais são:

- **Posição**

O posicionamento pode ser considerado o atributo visual mais importante, pois ele quem dita a densidade das informações e influencia a maneira como o conteúdo vai ser absorvido e interpretado.

- **Marcas**

Marcas e símbolos são formas primitivas que representam dados individuais. Esses recursos têm como características serem distintos uns dos outros, isso independente de tamanho, intensidade ou cores (outros atributos descritos a seguir), para evitar ambiguidade no entendimento.

- **Tamanho (Comprimento, Área e Volume)**

Visualização consiste basicamente de posicionamento e Marcas, os próximos atributos são definições que modificam diretamente um deles ou ambos. Tamanho é o primeiro atributo a alterar graficamente os atributos anteriores.







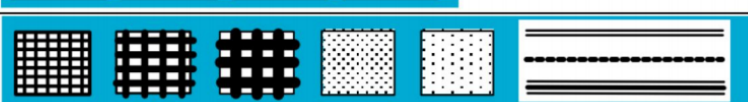
Atributos Visuais de Bertin	
POSIÇÃO	
TAMANHO	
MARCAS	
LUMINOSIDADE	
COR	
ORIENTAÇÃO	
TEXTURA	

Figura 3.3 Atributos visuais, traduzido de Bertin (1983).

- **Luminosidade ou Brilho**

Atributo usado para mapear informações adicionais nas marcas. Importante atentar que embora o atributo possa ser quantificado em uma escala numérica a percepção humana é limitada para definir valores de luminosidade muito próximos, logo, esta variável é usada para representar diferenças consideráveis em intervalos de dados comparáveis ou informações contínuas.

- **Cor**

Cores funcionam como o atributo de Tamanho, adicionando ainda mais informações as marcas, sendo necessário mapear essas informações diretamente as cores usadas na representação. Além de luminosidade, as cores possuem matiz (que define qual a faixa de onda dominante) e a saturação que define a quantidade de cinza em relação a matiz, definindo assim a pureza. Necessário ressaltar que devido a percepção humana, este atributo é um dos mais eficazes na diferenciação de dados distintos.

- **Orientação ou Direção**

Atributo visual modificador de marcas e símbolos focado no recurso da pre-atenção que segundo Janiszewski (1993), é a absorção inconsciente de informações através

de um processamento em paralelo a consciência que utiliza todos os órgãos sensoriais disponíveis. Importante ressaltar que não é aplicável a todas as marcas, por exemplo, a orientação de um círculo é irrelevante.

- **Textura**

Textura é uma combinação dos atributos citados anteriormente, Marcas, Cores e Orientação e assim como as outras é usada para mapear ainda mais informações na visualização.

- **Movimento**

Pode ser associado a qualquer um dos atributos visuais acima, uma vez que a maneira, velocidade, e intensidade do movimento na visualização pode indicar variações e modificações nos dados assim comunicando ainda mais informações na mesma visualização.

3.3 A Taxonomia de Keim

Taxonomia é definida como estudo científico responsável por determinar a classificação sistemática de diferentes coisas em categorias. Logo, existem várias formas de classificar visualizações, para este trabalho é importante entender a de definição de Keim (2002) na Figura 3.4, este modelo é dividido em três eixos, a saber: tipos de dados visualizados, a interação entre eles e o usuário, as técnicas de distorção que estão voltadas aos detalhes enquanto mantém a visão geral dos dados e por fim as técnicas de visualização que tem como objetivo caracterizar a maneira como as informações são mostradas.

Alguns exemplos de cada parte que integra o demonstrativo dessa taxonomia, traduzidos de (Ward *et al.*, 2015):

Tipos de dados a serem visualizados

- **Unidimensional:** dados sobre tempo, ações da bolsa, documentos de texto.
- **bidimensional:** mapas e plantas baixas.
- **Multidimensional:** tabelas e base de dados relacionadas.
- **Texto / Web:** notícias e hipertextos em páginas Web.
- **Hierarquias/Grafos:** números de telefone, fluxo de rede e modelos dinâmicos de sistemas.

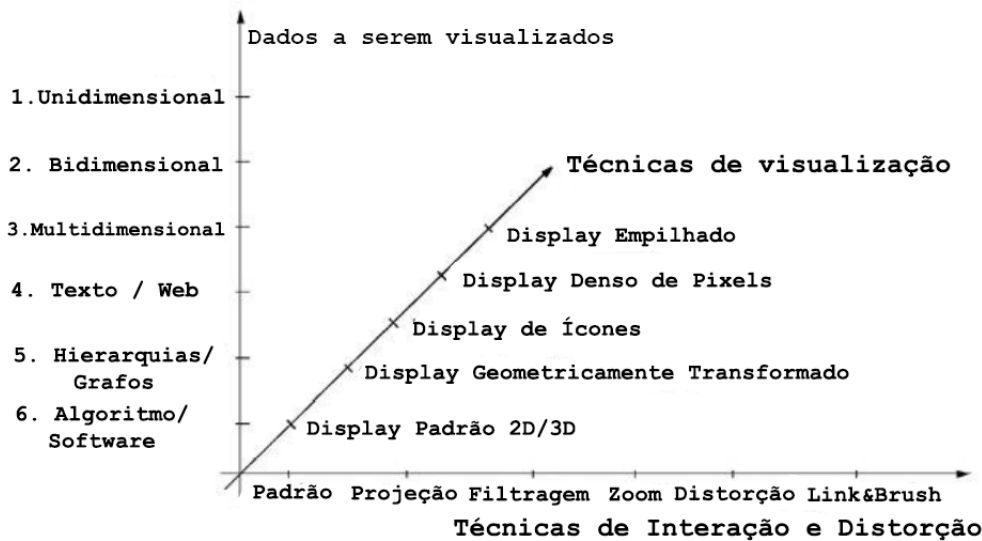


Figura 3.4 Demonstrativo da taxonomia de keim, traduzido de Ward *et al.* (2015).

- **Algoritmo/Software:** programas, descargas de memória e relatórios de execução.

Técnicas de visualização

- **Padrão 2D/3D:** planos cartesianos, Gráficos em geral (barras, pizza, por exemplo).
- **Geometricamente Transformado:** matrizes de *Scatterplot*, projeções visuais.
- **Baseado em Ícones:** Faces de Chernoff, Ícones agulha, Bonecos palito.
- **Denso em Pixels:** Grafos, padrões recursivos e segmentos circulares.
- **Empilhado:** Empilhamento dimensional, Eixos hierárquico e Árvores cônicas.

Técnicas de Interação e distorção

- **Projeção:** XGobi(Ferramenta de software estatístico).
- **Filtragem:** Queries dinâmicas, usadas para separar dados e interações relevantes de não relevantes.
- **Zoom:** IVEE/Spotfire (Ferramenta de estatística voltada a Inteligência de Negócios).
- **Distorção:** Distorção visual: Olho de peixe, visualização hiperbólica.
- **Link & Brush:** XmdvTool(Integração de múltiplas técnicas de visualização em múltiplos tipos de dados).

3.4 Visualização de Grafos

Grafos são estruturas simples que consistem de um conjunto de vértices e arestas. Telea (2014) apresenta uma definição formal de grafos pela fórmula $G=(V,A)$, onde um conjunto de vértices $V = v_i$ e um conjunto de arestas $A = a_i$, onde toda aresta $a_i = (v_j, v_k)$ é um par de vértices v_j e $v_k \in V$.

Este modelo matemático visa representar objetos concretos e abstratos utilizando os vértices e suas relações utilizando as arestas. Para isso, recorremos a técnicas de visualização para expor esses dados de modo coerente e útil, propriedades necessárias para representar arquivos RDF, como descrito em Consortium *et al.* (2014). Nessa estrutura pode-se associar atributos aos vértices e arestas, assim como em uma tabela, esses atributos são independentemente relacionados. De acordo com Telea (2014), é importante notar a quantidade de vértices (ou nós) conectados à estrutura, uma vez que pelo menos dois nós devem estar conectados através de uma aresta, o que forma uma tripla onde dois elementos possuem uma relação. Podemos então classificar os grafos em diversas visualizações já existentes, abaixo uma seleção dos métodos mais populares (Telea, 2014).

- **Visualização hierárquica de grafos**

Grafos que não possuem ciclos são denominados **árvores**. Cada nó da árvore tem um nó pai, um exemplo dessa estrutura é um esquema de hierarquia de cargos, onde cada funcionário tem um superior, exceto as folhas da árvore que não são superiores a nenhum outro nó, outro exemplo é uma árvore genealógica que segue o mesmo princípio como mostrado na Figura 3.5, podemos notar que uma das características desse modelo é que os nós são separados por camadas, definidas pela semântica que o grafo expressa (Lima, 2013).

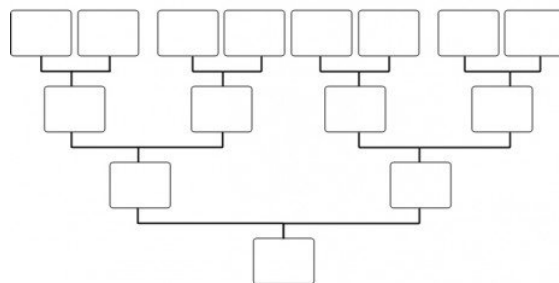


Figura 3.5 Exemplo de árvore genealógica.

- **Grafo Espalhado**

Embora a visualização de dados com uma hierarquia definida seja eficiente, a mesma possui limitações quanto a sua escalabilidade, quanto maior a massa de dados mais difícil (e confusa) se torna a leitura dos modelos hierárquicos. Por outro lado Grafo Espalhado (em tradução livre de *Graph Splatting* van Liere and de Leeuw (2003)) é uma outra abordagem que visa uma exposição eficiente focando na percepção humana que pode não captar variações em valores numéricos muito próximos, para isso utilizamos dois métodos, simplificação das informações originais a fim de torná-las menos densas, produzindo assim imagens de mais fácil entendimento, ou preservar os dados originais e aplicar técnicas que alterem o esquema de visualização dos dados a fim de modificar o nível de detalhes do mapeamento produzindo algo coerente que reflita os dados originais, um exemplo dessa técnica é o mapeamento contínuo de informação ilustrado na Figura 3.6.

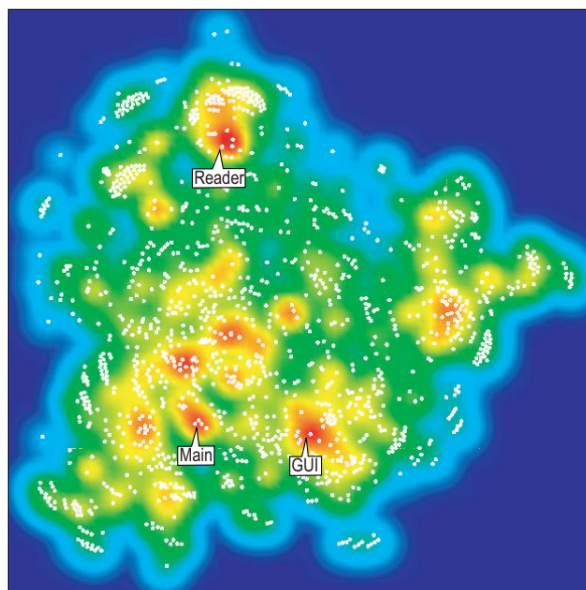


Figura 3.6 Exemplo de modelo de Grafo Espalhado, representando a densidade de informação semelhante a uma imagem térmica (Telea, 2014).

Grafos podem ser dinamicamente desenhados, ou seja, alguns são produzidos baseados em informações previamente preparadas ou utilizando informações em tempo de execução. Essas visualizações são divididas em *Streaming*, ou seja onde vértices e arestas existem durante um intervalo de tempo e *Sequential*, onde o grafo expressa um recorte de uma determinada base dados contínua. Essas duas técnicas possuem ainda variantes *Offline* e *Online*, ou seja, alguns grafos podem ter seus dados previamente analisados

antes de mostrados para usuários. Ao contrário desse modelo, outros são feitos “ao vivo” onde os dados são mostrados quando disponibilizados, ou seja não há um controle nas futuras modificações que o mesmo possa sofrer. Para isso recursos de **animação** podem ser usados para mostrar transições e afins, recursos abordados nesse trabalho, a fim de demonstrar a adição e navegação entre nós.

Um exemplo mais atual de visualização de grafos é o projeto *Nabusca*¹ exibe, de maneira interativa, informações e dados que o Google coleta sobre os candidatos à presidência. Ele que reúne informações sobre as diversas campanhas presidenciais sob a ótica das buscas feitas pelos/ usuários, com dados atualizados em tempo real sobre o que eles buscam em temas relevantes ao assunto, exemplificado na Figura 3.7.

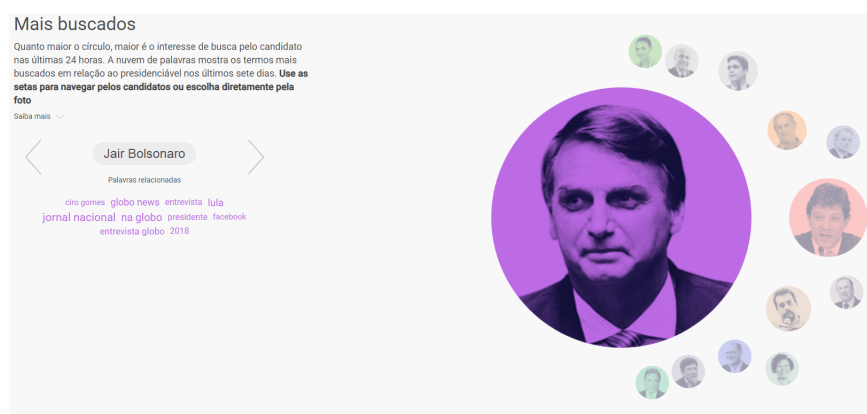


Figura 3.7 Projeto *Nabusca*, reprodução (Carol Cavaleiro, 2018).

Como exemplo de grafos em visualização de software, podemos citar o trabalho de Holten (2006), que utiliza um grafo com três tipos de nós (métodos, classes e arquivos) e dois tipos de relações (chamada e contenção). Dispondo o grafo em círculos concêntricos separando o anel interno representa as funções, no meio as classes e na borda os contêineres. Na Figura 3.8 pode-se observar que os métodos (em verde) estão todos contidos em uma única classe (em azul) que está contida em um arquivo (em amarelo), devido a essa disposição de nós e relações é possível extrair várias informações sobre esses sistemas, por exemplo, a disposição dos componentes nos grafos indicam que o sistema representado pela imagem da esquerda é mais modular que o representado pela imagem à direita.

¹Disponível em <http://www.nabuscadocandidato.com.br>. Acessado em: 2 de novembro 2018)

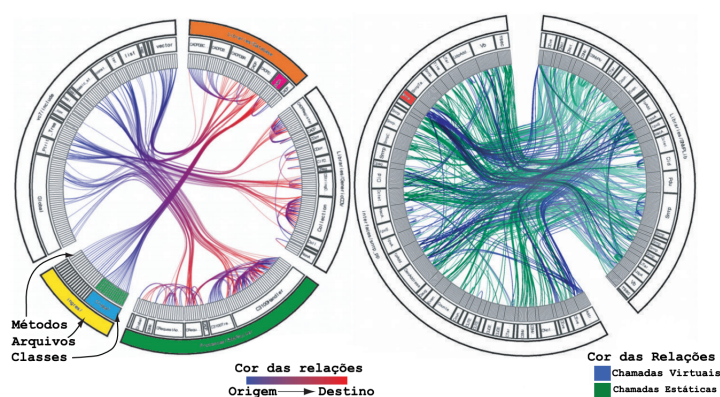


Figura 3.8 Grafos representando como dois sistemas operam, traduzido de Holten (2006).

3.5 Sumário

Neste capítulo foi apresentada uma breve introdução sobre visualização e sua importância, foram introduzidos os principais conceitos de visualização responsáveis pelo mapeamento de informações em imagens. Assim como foi discutido também a taxonomia de Keim (2002) que foi escolhida para demonstrar como podem ser categorizadas as técnicas de visualização.

Após listar seus componentes que formam uma imagem com dados mapeados, foram introduzidos conceitos de grafos e apresentadas alguns tipos dessas estruturas. Por fim, exemplos práticos que demonstram a versatilidade para solução de problemas reais, como a eleição presidencial que teve tanto impacto na sociedade atual, como também de sua utilidade para problemas específicos (como aplicado em uma análise de software) e, em ambos os casos, provando sua eficiência como método de visualização de dados.

4

Uma Aplicação Web com Visualização de Grafos para Documentos RDF

Este capítulo descreve os resultados deste trabalho, o qual é composto de uma aplicação Web com visualizações de dados em grafos que utiliza documentos RDF da base da Dbpedia. Para tal é apresentado o funcionamento da ferramenta, descrevendo como os dados são capturados, processados e apresentados. Então, é apresentada a aplicação e sua metáfora visual, assim como suas funcionalidades. Posteriormente é realizada uma avaliação descritiva e exploratória com ferramentas existentes. Ao término, é feita uma discussão e, por fim, são apresentadas as considerações finais do capítulo.

4.1 Visualização de Grafos para Documentos RDF

Como descrito no capítulo anterior, para representar informações em grafos é necessário tratar os dados, escolher uma técnica de visualização e então mapear os dados na imagem. Seguindo essa lógica, podemos aplicar o conceito de Pipeline na abordagem do problema, ou seja, a representação visual utilizando grafos de documentos RDF. Como previamente descrito, existem algumas abordagens que descrevem o fluxo de dados e transformação em imagens ou animações. Baseado nos pontos discutidos, podemos fazer uma analogia ao modelo proposto por Telea (2014), o qual foi apresentado e discutido no Capítulo 3 e ilustrado na Figura 3.2.

Nesse sentido, a Figura 4.1 representa o fluxo de dados da aplicação proposta, o qual foi inspirado e adaptado do modelo do Telea (2014)

Considerando o fluxo supracitado (Figura 4.1), a primeira etapa é composta da base de dados, a qual é composta de documentos RDF do Dbpedia. Seguindo o fluxo, é efetuado uma requisição por meio de *Request HTTP* utilizando string, no ponto de acesso

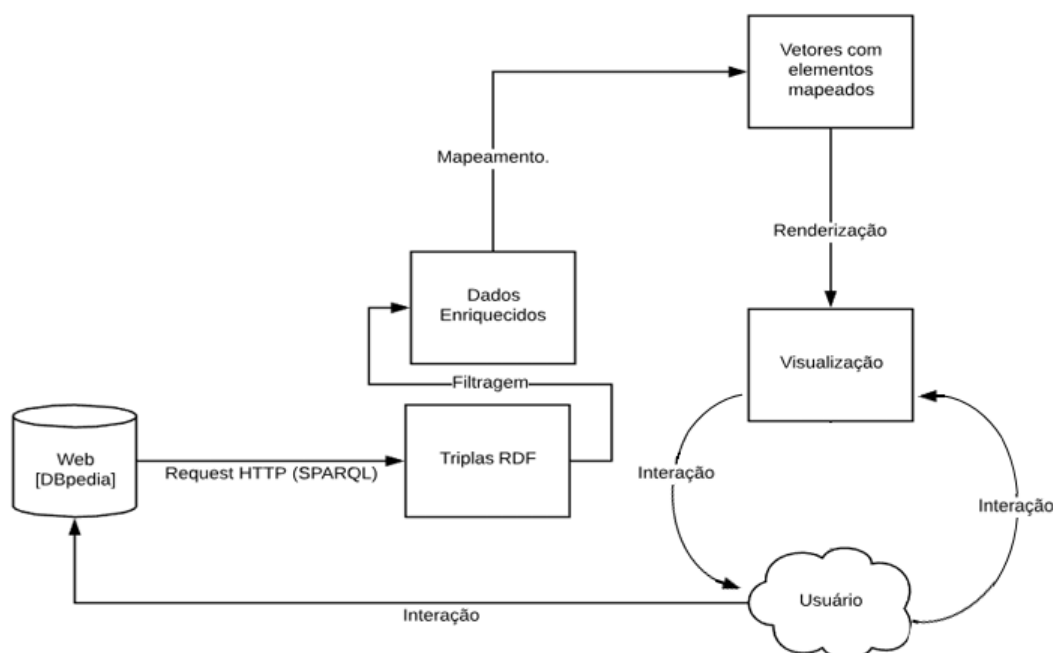


Figura 4.1 Uma visão geral da navegação gerada pelos passos anteriores.

*Endpoint*¹ que pertence a Dbpedia e permite a execução de consultas Sparql, através de um *HTTP Request*. O resultado da busca provê informações textuais em diversos formatos (sendo eles: *JSON*, *XML* e *XML-XSLT*), no caso da aplicação proposta, o formato escolhido para recebimento foi o *JSON*, pois o mesmo é melhor integrado com a linguagem utilizada, ou seja os dados podem ser enviados diretamente para um objeto para serem tratados. A próxima etapa é **Filtragem e enriquecimento**, que é feita durante a consulta é limitando o idioma dos dados que serão retornados. **Mapeamento** é a etapa que os dados abstratos são novamente separados entre nós e arestas. A definição de nós é feita usando a estrutura de triplas na qual os dados originais foram organizados permite a aquisição de informações quanto a ligação entre nós dos arquétipos “Sujeito” e “Objeto”, utilizando as ontologias oriundas da Dbpedia, e suas classes definidas nas ontologias como definições para as arestas. Ao definir os pontos de origem e destino das ligações entre nós, essas informações são alimentadas no grafo, usando uma modificação da biblioteca Javascript *Vis.js*² que modulariza os recursos para que o carregamento seja realizado em apenas uma parte da página, *Vis.js React*³. Ao realizar esse mapeamento,

¹<http://dbpedia.org/snorql/>

²<http://visjs.org>

³<https://github.com/crubier/react-graph-vis>

4.1. VISUALIZAÇÃO DE GRAFOS PARA DOCUMENTOS RDF

é iniciada a etapa final deste modelo, **Renderização** para isso, são adicionados a um objeto chamado **Estado**, o vetor de nós. Então são adicionadas as arestas, que contém a estrutura [Identificador do nó de origem][Rótulo][Identificador do nó destino], ao concluir a inserção de informações neste objeto, é acionada a renderização, que desenha na tela um Grafo Espalhado, como discutido na Seção 3.4 do Capítulo 3. Com a renderização é gerada a visualização de grafos, a qual permite ao usuário mecanismos de interação/navegação para que o mesmo possa criar diferentes cenários a fim de extrair o máximo de conhecimento possível. Além do usuário interagir com a visualização de grafos, ele pode requisitar uma nova informação da aplicação na base Dbpedia.

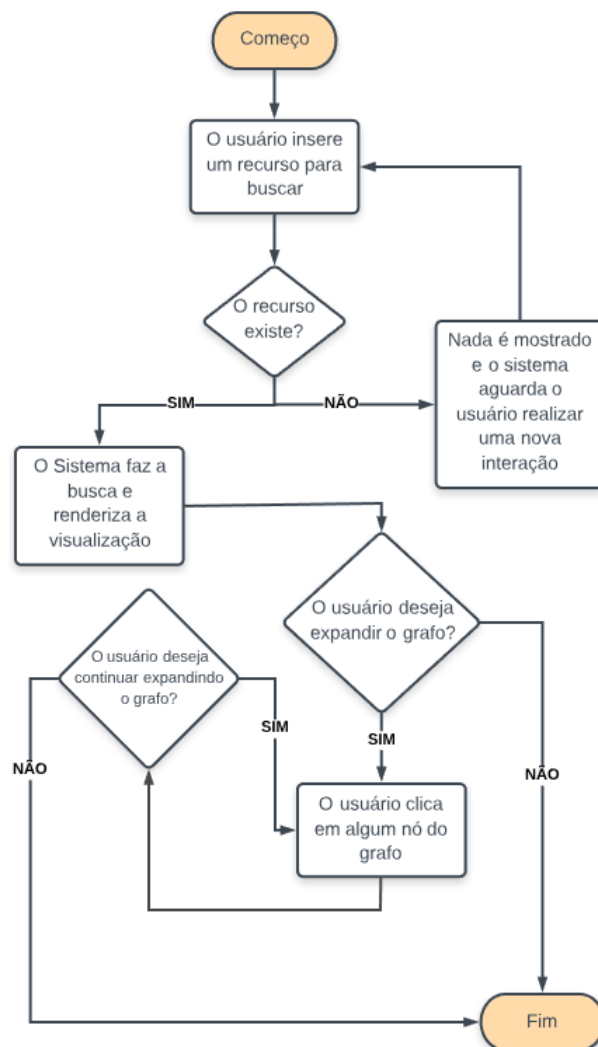


Figura 4.2 Diagrama de atividades da ferramenta proposta.

4.2 Apresentação da Aplicação

Seguindo os conceitos propostos nas seções anteriores foi criado o diagrama de atividades descrito na Figura 4.2, a seguir será exemplificado o funcionamento da ferramenta (nomeada “*Graph based Visualization of RDF documents*” – GraVisRDF). O fluxo do sistema começa com o usuário inserindo o recurso desejado nos campos representados na Figura 4.3. Uma vez inserido o recurso “*Disownment*” no campo de busca e o botão [Carregar] é pressionado, essas informações são carregadas na aplicação e apresentadas com visualização, em específico visualização de grafos. O botão [Limpar] ao lado de [Carregar] serve para apagar todas as entradas na página, eliminando os elementos.



Figura 4.3 Tela da aplicação com a visualização do recurso “*Disownment*”.

Uma vez gerada a visualização de grafo para o recurso “*Disownment*”, é possível realizar algumas interações na visualização. No tocante à edição de nós, é possível destacar três etapas, descritas a seguir. Na primeira etapa (1) ilustrada pela Figura 4.4 ocorre a edição do nó de nome “*Band*”. Ao clicar sobre nó contendo o rótulo “*Band*” com o botão direito do mouse, é mostrado um menu com duas opções. A primeira permite editar o nó previamente selecionado e a segunda possibilita excluir esse nó.

Ao selecionar a opção de editar, a janela de edição é apresentada, representada na

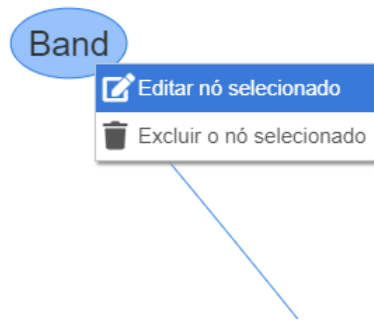


Figura 4.4 Edição de nós (1).

Figura 4.5 (A). Esta figura contém o campo **Label** que descreve o rótulo do nó, a opção **Cor** que permite ao usuário escolher uma cor de uma paleta de cores (representada na Figura 4.5 (B)), além disso duas áreas são reservadas para o usuário selecionar quais nós serão conectados ao nó que está sendo editado, assim como definir com quais nós irá conectar-se. Por fim, dois botões podem ser utilizados abaixo caso o usuário deseje manter essa alteração ele deve clicar em [Editar], caso contrário o botão [Cancelar] irá descartar as alterações realizadas na janela.



Figura 4.6 Edição de Aresta(A). Janela para edição da aresta(B).

Para manipular as informações das arestas primeiramente selecionamos com um clique, para sinalizar que aquela aresta foi selecionada o sistema aumenta a saturação da mesma, tornando-a mais visível, e a partir deste momento o usuário pode clicar com o botão direito na aresta selecionada anteriormente que gera um menu contendo a opção de editá-la (Figura 4.6 (A)), abrindo uma janela descrita na imagem (B) contendo o “Label”

CAPÍTULO 4. UMA APLICAÇÃO WEB COM VISUALIZAÇÃO DE GRAFOS PARA DOCUMENTOS RDF

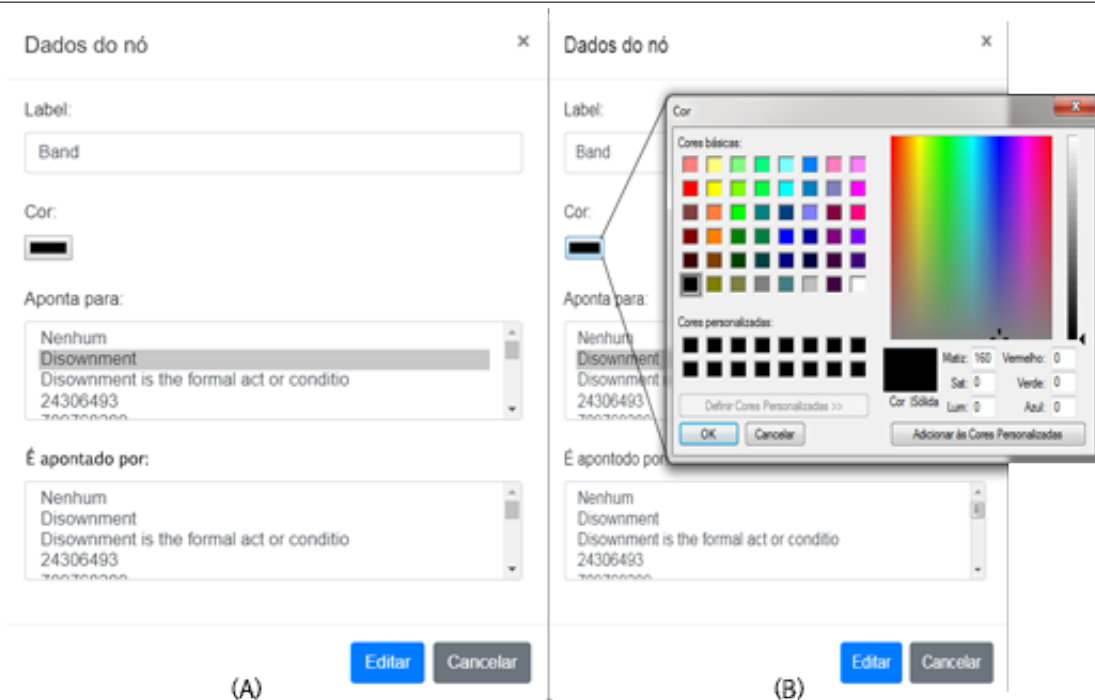


Figura 4.5 Edição de nós etapas (2) e (3) - Janela contendo as opções para edição do nó.

da aresta e a posição do rótulo em relação a linha, podendo ser escolhida uma de três opções, *Top* que deixa o texto linha da linha, *Middle* que coloca o texto dentro da linha, *Bottom* que coloca o texto abaixo da linha, novamente os botões [Editar] e [Cancelar] realizam as mesmas funções descritas anteriormente.



Figura 4.7 Adição de nós em área vazia.

O usuário também tem a opção de criar um nó a qualquer momento e em qualquer lugar do grafo, para isso é necessário que um clique com o botão direito em uma área em branco, como ilustrado na Figura 4.7, ao realizar a operação um menu é mostrado

com a opção para adicionar um novo nó, que uma vez selecionada abre uma janela idêntica a mostrada na Figura 4.5 com as diferenças de não haver rótulo preenchido, nem apontamentos já selecionados.

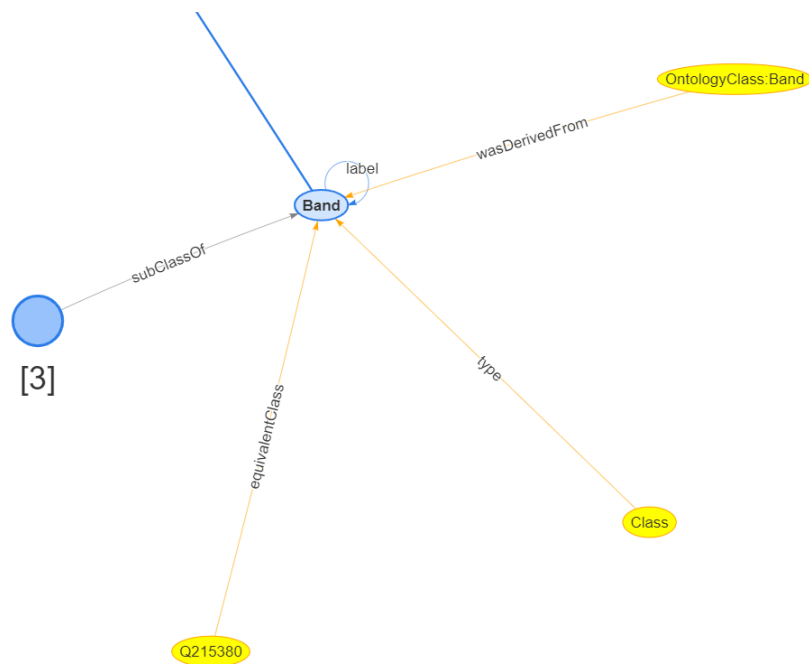


Figura 4.8 Exemplo de nó expandido, gerando outro grafo.

Outra função que o usuário pode interagir é a de **Agrupamento e Separação**. Ao carregar um grafo, as ontologias que possuem múltiplos elementos são carregadas agrupadas, como descrito na imagem da esquerda na Figura 4.9, que representa a classe “*Subject*” e possui sete elementos. Quando clicada com o botão esquerdo é separada em instâncias individuais como observada na imagem à direita (com suas arestas marcadas em vermelho).

CAPÍTULO 4. UMA APLICAÇÃO WEB COM VISUALIZAÇÃO DE GRAFOS PARA DOCUMENTOS RDF

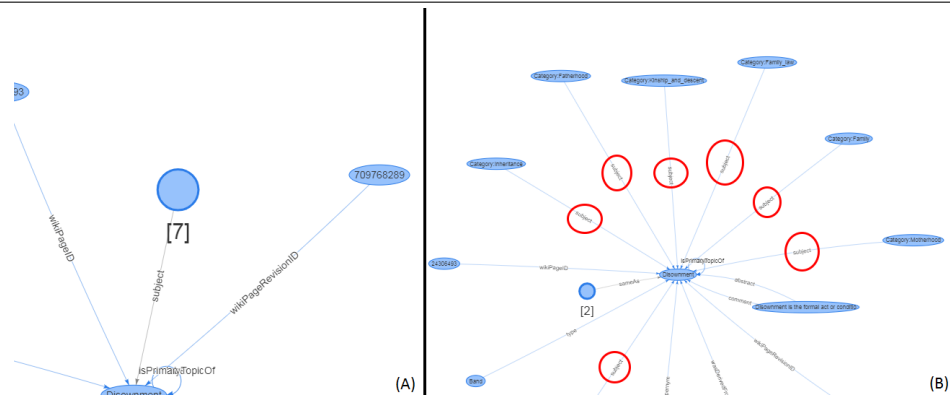


Figura 4.9 Agrupamento de instâncias pertencentes a mesma ontologia (“Subject”) e à direita é mostrada a Separação do agrupamento.

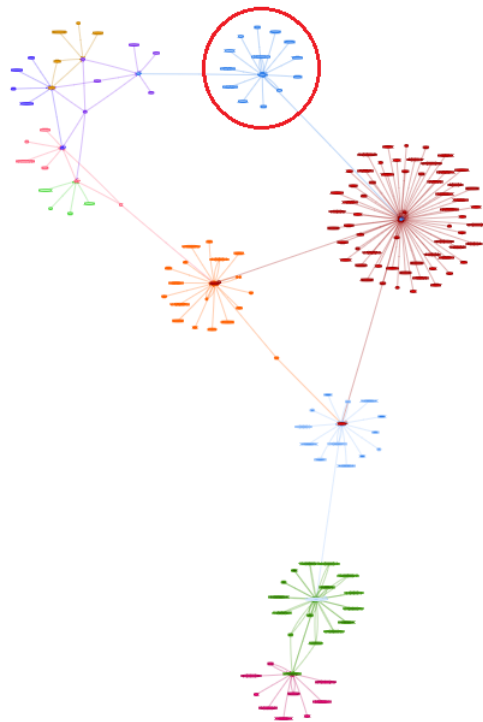


Figura 4.10 Um exemplo de nós expandidos, com o grafo inicial marcado em vermelho.

Retornando ao nó “*Band*” que foi utilizado como exemplo nas descrições anteriores, podemos visualizar a função de navegação, ou seja, ao clicar duas vezes em um determinado nó, caso o mesmo esteja na base de dados da Dbpedia, é possível expandir o grafo, agora utilizando o nó clicado como sujeito como mostrado na Imagem 4.8, importante notar que a cada expansão de nós uma nova cor é definida para diferenciar as interações,

tendo sempre a cor azul como inicial.

Assim são concluídas as descrições das funcionalidades da aplicação desenvolvida que gera uma visualização escalável, ou seja enquanto houver itens que possuam entradas no servidor da Dbpedia o grafo pode ser expandido e alterado como mostrado na Figura 4.10, vale ressaltar que apenas a visualização é alterada e não os arquivos RDF que deram origem a mesma.

4.3 Avaliação

Para melhor elucidar as principais contribuições deste trabalho, nesta sessão serão apresentadas algumas aplicações que apresentam documentos RDF e possuem funcionalidade semelhantes a aplicação desenvolvida para este trabalho. Em outras palavras, sistemas que tem como objetivo apresentar informações do banco de dados da Dbpedia utilizando visualizações de grafos. A seguir, são apresentadas e descritas sucintamente tais aplicações:

- **LodLive**⁴: Um projeto que visa demonstrar o uso dos padrões de dados ligados (RDF, SPARQL) para navegar entre os recursos, tendo como objetivo popularizar esses conceitos de maneira simples e com uma interface amigável.
- **Graphviz**⁵: Um software de visualização com código aberto, voltado a representação de informação estruturada como diagramas e redes. O principal foco do sistema é a saída em múltiplos formatos.
- **SZTAKI LODmilla**⁶: Projeto desenvolvido pelo DSD (Departamento de Sistemas Distribuídos) do Instituto de Ciência de Computação e Controle da Academia Húngara de Ciências, é um navegador visual de base de dados RDF e LOD.
- **Ontology-visualization**⁷: Projeto desenvolvido por Xi Jin para a Universidade da Califórnia do Sul, utiliza Python para gerar grafos a partir de documentos RDF.

Para fins de comparação entre as abordagens de visualização de grafos existentes, a Tabela 4.1 apresenta algumas características que as ferramentas possuem em comum, bem como a aplicação desenvolvida, para que seja efetuada uma comparação. Tendo como base os seguintes critérios:

⁴Disponível em: <http://en.lodlive.it/credits>. Acessado em: 01/12/2018

⁵Disponível em: <http://www.webgraphviz.com> Acessado em: 01/12/2018

⁶Disponível em: <http://lodmilla.sztaki.hu/lodmilla> . Acessado em: 01/12/2018

⁷Disponível em: <https://github.com/usc-isi-i2/ontology-visualization> Acessado em: 01/12/2018

CAPÍTULO 4. UMA APLICAÇÃO WEB COM VISUALIZAÇÃO DE GRAFOS PARA DOCUMENTOS RDF

- **Código aberto:** Se o software teve o código disponibilizado para a comunidade de forma livre, para melhorias ou modificações;
- **Aplicação Web:** A ferramenta deve estar disponibilizada em uma página web, e é funcional (totalmente ou parcialmente), ou seja a captura e geração da visualização é completamente feita sem necessidade de instalação de arquivo;
- **Dinâmico:** A possibilidade de alteração de elementos no grafo de forma dinâmica;
- **Navegação hierárquica:** A possibilidade de criação de novos grafos ligando o atual de forma responsiva.

Ferramenta	Código aberto	Aplicação Web	Dinâmico	Navegação hierárquica
LodLive	Sim	Sim	Sim	Sim
Graphviz	Sim	Sim	Não	Não
SZTAKI LODmilla	Não	Sim	Sim	Sim
Ontology visualization	Sim	Sim	Não	Não
Aplicação desenvolvida	Sim	Sim	Sim	Sim

Tabela 4.1 Características de visualização de grafos existentes.

4.4 Discussão

Como podemos observar usando essas informações da seção anterior que apesar dessas ferramentas serem diferentes, tanto quanto a tecnologias quanto a objetivos, os pontos em comuns são claros. A maioria dos projeto possuem código aberto, assim como a maioria tem pelo menos uma parte voltada para Web, salvo a **Ontology-visualization**, entretanto a liberdade do usuário para alterar os valores dos campos gerados pela visualização é limitada em todas, basicamente o usuário fica restrito a apenas ocultar informações.

A ausência destas funções motivou uma comparação entre funcionalidades presentes nesta aplicação com o intuito comparar a implementação desenvolvida. Para isso, primeiramente foram definidas algumas funcionalidades, sendo estas descritas a baixo.

1. **Editar:** O usuário tem maneiras de editar os nós.
2. **Adicionar:** É possível adicionar novos nós e arestas.

3. **Excluir:** Nós e Arestas existentes podem ser removidos.
4. **Mostra Rótulos:** Todos os item são rotulados.
5. **Agrupamento:** nós com características semelhante são condensados em um único nó.
6. **Separação:** Nós agrupados são separados em instâncias únicas.
7. **Interações:** Possíveis interações que os usuários podem ter com o grafo.
Sendo algumas delas, “Mouse Over” informações extras são mostradas quando o usuário passa o mouse por cima do elemento, ocorre algum evento quando o é clicado (singular, duplamente ou com o direito) e por fim o clicar e arrastar elementos pela tela (claro se o gráfico responde as alterações)
8. **Responsividade:** A aplicação possui maneiras de se redimensionar de acordo com o dispositivo na qual está sendo acessada.
9. **Zoom (in/out):** O usuário pode aumentar ou diminuir a magnificação da página, tendo uma visão aproximada ou abrangente dos elementos
10. **Exportação:** Maneiras que informações que podem ser geradas a partir das interações com os grafos.

A partir dessas informações foi criada a Tabela 4.2 a fim de comparar essas funcionalidades em todas as ferramentas. Com essas informações, podemos iniciar uma discussão levando em consideração alguns dos pontos mostrados na tabela que necessitam de um detalhamento mais aprofundado. Começando pelas funções mais comuns e suas diferenças entre as aplicações verificadas, tomando como exemplo o tópico “Mostrar Rótulo”, mesmo que seja óbvio que todos os mecanismos que visualizem dados precisem marcá-los de alguma forma, no caso texto, porém nenhuma das aplicações possibilitam a livre manipulação dessas informações, salvo a **LODmilla**, que permite ao usuário “excluir” as informações (artifício atingido não necessariamente deletando os itens e sim ocultando-os), quanto a aplicação desenvolvida elimina as entidades totalmente da visualização.

Uma vez comentado sobre a função mais comum, partimos para as funções menos presentes nas ferramentas, ou seja, a possibilidade de condensar elementos com as mesmas características, descrita pelas funcionalidades de AGRUPAMENTO e SEPARAÇÃO, que é realizada somente pela ferramenta **LodLive**, que realiza algo parecido com a

CAPÍTULO 4. UMA APLICAÇÃO WEB COM VISUALIZAÇÃO DE GRAFOS PARA DOCUMENTOS RDF

Ferramentas					
Função	LodLive	Graphviz	LODmilla	Ontology visualization	Aplicação Desenvolvida
Editar	Não	Não	Não	Não	Sim
Adicionar	Não	Não	Sim	Não	Sim
Excluir	Não	Não	Sim	Não	Sim
Agrupamento	Não	Não	Não	Não	Sim
Separação	Sim	Não	Não	Não	Sim
Mostra Rótulos	Sim	Sim	Sim	Sim	Sim
Importação	Página do Recurso	Entrada de texto RDF	Página do Recurso	Entrada de texto RDF	Página do Recurso
Interações	Mouse over Duplo clique Clicar e Arrastar	Nenhuma	Mouse over Duplo clique Clicar e Arrastar	Nenhuma	Mouse over Duplo clique Clicar e Arrastar
Responsividade	Sim	Não	Sim	Não	Sim
Zoom	Não	Não	Sim	Não	Sim
Exportação	Nenhuma	Gera Imagem	Gera um link	Gera Imagem	Nenhuma

Tabela 4.2 Comparativo entre funcionalidades.

ideia proposta na nossa aplicação, mas efetivamente funciona de maneira distinta. Os nós são carregados e nós subsequentes ficam para ser carregados quando requisitados, ou seja até que o usuário ativamente busque-os, eles não estarão carregados na tela, a visualização gerada pela ferramenta também não disponibiliza nenhuma informação sobre esses agrupamentos. A aplicação desenvolvida trata os agrupamentos como entidades que possuem também suas informações mapeadas (quantos elementos estão agrupados e qual aresta esses itens compartilham).

A maneira como esses dados são alimentados também varia de ferramenta para ferramenta, mas podem ser classificados em apenas duas categorias, “Página do recurso” e “Entrada de texto RDF”. A alimentação via página de recurso implica que o usuário dá entrada no sistema com um endereço da página do recurso na base da Dbpedia, já o método de “Entrada de texto RDF” envolve a inserção manual de um documento RDF não necessariamente proveniente da base supracitada.

Outra função importante são as maneiras como todos os sistemas exportam as informações geradas, após uma investigação sobre isto, foi concluído que apenas três das cinco ferramentas avaliadas possuíam maneiras de exportar dados acessíveis para os usuários, nos casos de **Graphviz** e **Ontology-visualization** permite aos operadores salvar as visualizações geradas em imagens. Quanto a aplicação **LODmilla**, a exportação é composta de um *link* que possui uma sequencia de texto composta de variáveis que a

ferramenta utiliza para reconstruir o grafo original.

Podemos então após essa breve avaliação é possível constatar que embora nenhuma das aplicações possuam todas as funções listadas acima, a Tabela 4.2 indica que a **Graphviz** é a ferramenta mais limitada entre as avaliadas, assim como a mais completa segundo essa análise é a aplicação desenvolvida. Desse modo, as implementações dessas funcionalidades compõem as principais contribuições do trabalho, juntamente com a função hierárquica elencada anteriormente e as avaliações comparativas.

4.5 Sumário

Neste capítulo foi discutida a aplicação das técnicas de visualização de forma prática a dados reais, para mapear informações e então uma demonstração dos conceitos de Web Semântica para registro e captura de dados. Utilizando a base dados do projeto Dbpedia foi mostrado como esses elementos são tratados e mapeados utilizando as técnicas comentadas e embasadas no capítulo de visualização.

Uma vez criada a imagem, a ferramenta proporciona ao usuário a liberdade de interagir com os elementos, arrastando-os, editando ou navegando através de seus nós. Válido ressaltar que originalmente as bibliotecas que dão suporte a renderização dos itens na tela não os faziam de forma dinâmica e modularizada, foram feitas alterações no código fonte original para permitir que os aglomerados de nós mantivessem seus devidos rótulos após colapsados.

Mostrando além disto, que há uma necessidade de maior variedade quanto a ferramentas de apoio para a Web Semântica, ferramentas com fácil acesso e disponíveis como aplicação Web, e com isso utilizando alguns programas semelhantes a ideia central do trabalho para uma comparação. Demonstrou-se que a ferramenta GraVisRDF se destaca entre as outras no que toca funcionalidades, e é um exemplo do poder que os conceitos de dados ligados têm com a quantidade limitada de informação indexada atualmente.

5

Conclusão

Neste capítulo será apresentado um resumo geral do material produzido, uma breve análise dos resultados obtidos pela aplicação, contribuições deste trabalho e propostas para melhorias futuras. Este trabalho teve como proposta uma melhoria na visualização de grafos para documentos RDF provenientes da Dbpedia, com foco na interação do usuário. Os resultados finais mostram que a ferramenta desenvolvida possui mais funcionalidades que as ferramentas apresentadas.

5.1 Contribuições

A aplicação GraVisRDF foi desenvolvida a partir dos conceitos da Web Semântica e Visualização de Dados com grafos para suprir um nicho de usuários que precisam visualizar informações e ter a liberdade de alterá-las, com um adicional da aplicação ser totalmente Web. Importante salientar que embora existam algumas soluções disponíveis, dentre as quais citadas nos capítulos anteriores, a nossa ferramenta possui mais funções como por exemplo **EDITAR**, que é aplicada em todos os elementos que formam o grafo, permitindo assim maior flexibilidade para os usuários e maior controle sobre as informações geradas dessas manipulações, enquanto mantém-se padrões estabelecidos pela Web Semântica e se mantendo de acordo com os atributos visuais da Visualização de dados. Além disto, o código fonte é livre e está disponível no endereço: <https://github.com/joseolimpio/navegadorDbpedia>

5.2 Pontos de Melhorias e Possíveis Incrementos

A representação dos elementos que compõe o gráfico pode ser alterado para se adequar aos padrões de representação estabelecidos pelo W3C. Outro ponto de melhoria

é que o sistema poderia ser integrado com algum algoritmo de inteligência artificial que pudesse identificar o menor caminho entre dois nós, a fim de gerar uma relação visual entre dois pontos. Dessa forma ficaria claro para o usuário quais foram os elementos que ligam dois nós. Por fim, a proposta de geração de saídas para o sistema também é interessante, afim de produzir materiais científicos, tanto como imagens ou documentos de texto contendo as estruturas geradas pela ferramenta. A limitação quanto a base de dados também deve ser explorada, uma vez que o projeto foi modularizado para receber um padrão de informação derivada da Dbpedia (em *JSON*), e embora em virtude de limitações temporais não tenha sido testado, qualquer documento possa ser recebido pelo mesmo, claro, com seus devidos tratamentos para se adequar aos padrões aceitos. Por fim, o sistema não possui nenhuma forma de exportar as visualizações geradas, um exemplo disso seria a captura de todos os nós em uma mesma imagem de alta resolução, ou uma saída que gere um documento RDF para reprodução posterior.

5.3 Sumário

Neste capítulo foi discutido a aplicação GraVisRDF e suas contribuições para as áreas pertinentes ao trabalho, além disso as possíveis ramificações que venham a criar melhorias e impactos no futuro. Foi comentado também a importância do projeto e onde o mesmo se encontra disponível.

Referências Bibliográficas

- Ackoff, R. L. (1989). From data to wisdom. *Journal of applied systems analysis*, **16**(1), 3–9.
- Andrejs Abele, J. M. (2018). The linking open data cloud diagram. Disponível em: <http://lod-cloud.net/>. [Último acesso em 30 de outubro de 2018].
- Berners-Lee, T. (2006). Linked data. Disponível em: <https://www.w3.org/DesignIssues/LinkedData.html>. [Último acesso em 03 de Novembro de 2018].
- Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, **284**(5), 34–43.
- Berners-Lee, T., Chen, Y., Chilton, L., Connolly, D., Dhanaraj, R., Hollenbach, J., Lerer, A., and Sheets, D. (2006). Tabulator: Exploring and analyzing linked data on the semantic web. In *Proceedings of the 3rd international semantic web user interaction workshop*, volume 2006, page 159. Citeseer.
- Bertin, J. (1983). *Semiology of Graphics*. University of Wisconsin Press.
- Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009). Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web*, **7**(3), 154–165.
- Carol Cavaleiro, Thais Viana, T. S. M. T. P. S. R. A. C. (2018). Na busca do candidato. Disponível em: <http://www.nabuscadocandidato.com.br>. [Último acesso em 20 de novembro de 2018].
- Consortium, W. W. W. *et al.* (2014). Rdf 1.1 concepts and abstract syntax.
- DBPedia (2014). Dbpedia 2014 data set statistics. Disponível em: <http://wiki.dbpedia.org/services-resources/datasets/dataset-statistics>. [Último acesso em 03 de Novembro de 2017].
- Heath, T. and Bizer, C. (2011). Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, **1**(1), 1–136.
- Holten, D. (2006). Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, **12**(5), 741–748.

REFERÊNCIAS BIBLIOGRÁFICAS

- Horrocks, I., Patel-Schneider, P. F., and van Harmelen, F. (2003). From shiq and rdf to owl: the making of a web ontology language. *Web Semantics: Science, Services and Agents on the World Wide Web*, **1**(1), 7 – 26.
- Horrocks, I., Parsia, B., Patel-Schneider, P., and Hendler, J. (2005). Semantic web architecture: Stack or two towers? In F. Fages and S. Soliman, editors, *Principles and Practice of Semantic Web Reasoning*, pages 37–41, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Janiszewski, C. (1993). Preattentive mere exposure effects. *Journal of Consumer Research*, **20**(3), 376–392.
- Keim, D. A. (2002). Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, **8**(1), 1–8.
- Korfhage, R. R. (1995). Visualization of natural phenomena. *Information Processing Management*, **v31**(n1), p143(2).
- Lassila, O., Swick, R. R., *et al.* (1998). Resource description framework (rdf) model and syntax specification.
- Laufer, C. (2018). Guia : Web semântica, capítulo 4. Disponível em: <http://ceweb.br/guias/web-semantica//capitulo-4/>. [Último acesso em 27 de Outubro de 2018].
- Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., *et al.* (2015). Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, **6**(2), 167–195.
- Lima, D. M. d. (2013). *Análise visual de dados relacionais: uma abordagem interativa suportada por teoria dos grafos*. Ph.D. thesis, Universidade de São Paulo.
- Openlink, D. . (2013). Sparql explorer for dbpedia. Disponível em: <http://dbpedia.org/snorql/>. [Último acesso em 27 de outubro de 2018].
- Ribeiro, A. L. (2008). As camadas da arquitetura da web semântica. Disponível em: <http://adagenor.blogspot.com/2008/03/as-camadas-da-arquitetura-da-web.html>. [Último acesso em 27 de Outubro de 2018].

- Schramm, W. L. (1988). *The story of human communication: Cave painting to microchip*. Harpercollins College Division.
- Telea, A. C. (2014). *Data Visualization: Principles and Practice, Second Edition*. A. K. Peters, Ltd., Natick, MA, USA, 2nd edition.
- van Liere, R. and de Leeuw, W. C. (2003). Graphsplatting: Visualizing graphs as continuous fields. *IEEE Trans. Vis. Comput. Graph.*, **9**(2), 206–212.
- Van Wijk, J. J. (2005). The value of visualization. In *Visualization, 2005. VIS 05. IEEE*, pages 79–86. IEEE.
- W3C (2001). W3c semantic web activity. Disponível em: <https://www.w3.org/2001/sw/>. [Último acesso em 02 de Novembro de 2018].
- W3C (2009). Owl web ontology language guide. Disponível em: <https://www.w3.org/TR/owl-guide/>. [Último acesso em 28 de outubro de 2018].
- W3C (2013). Sparql query language for rdf. Disponível em: <https://www.w3.org/TR/rdf-sparql-query/>. [Último acesso em 27 de outubro de 2018].
- W3C (2014). Rdf. Disponível em: <https://www.w3.org/RDF/>. [Último acesso em 26 de Outubro de 2018].
- Ward, M. O., Grinstein, G., and Keim, D. (2015). *Interactive Data Visualization: Foundations, Techniques, and Applications, Second Edition - 360 Degree Business*. A. K. Peters, Ltd., Natick, MA, USA, 2nd edition.
- Wellman, D. (2013). What is big data? Disponível em: <https://www.slideshare.net/dwellman/what-is-big-data-24401517>. [Último acesso em 02 de Novembro de 2018].