

# Armazenamento e Manipulação da Representação Semântica de Dados utilizando as tecnologias Java e MongoDB

Robson Leandro dos Santos, Evaldo de Oliveira da Silva

Curso de Bacharelado em Sistemas de Informação – Centro de Ensino Superior de Juiz de Fora (CESJF) – Campus Academia  
Juiz de Fora – MG– Brasil

robsousantos@hotmail.com, evaldo.oliveira@gmail.com

***Abstract.** This article presents concepts and tools used to store and manipulate semantic data as an alternative to the management of information and knowledge guided by semantic Web theories and concepts related to it, such as Ontology and Linked data, proposing, from these, a prototype tool capable of storing and manipulating ontological structures contained in RDF (Resource Description Framework) files using Java and the MongoDB NoSQL database, exposing relevant points of these technologies and the main concepts related to them in order to present the benefits of using these concepts and technologies in the organization and management of information.*

***Resumo.** O presente artigo apresenta conceitos e ferramentas utilizados para armazenamento e manipulação de dados semânticos como alternativa a gestão de informações e conhecimentos norteado pelas teorias da Web semântica e os conceitos relacionados a ela, tais como , Ontologia e Linked data propondo, a partir destes, um protótipo de ferramenta capaz de armazenar e manipular estruturas ontológicas contidas em arquivos RDF (Resource Description Framework) utilizando Java e a base de dados NoSQL MongoDB, expondo pontos relevantes destas tecnologias e os principais conceitos relacionados a estas a fim de apresentar os benefícios do uso destes conceitos e tecnologias na organização e gestão de informações.*

## 1. Introdução.

O constante avanço no número de usuários da Web, alavancado pelas facilidades oferecidas por dispositivos e redes de dados móveis, seja para acessar redes sociais, para atividades profissionais e/ou pessoais, para comunicação, compartilhamento ou também buscas de informações, tem como consequência, o aumento no volume de conteúdo disponível na internet (MACHADO, 2017). Para organizar estes conteúdos a fim de evitar conflitos entre o que se deseja e aquilo que lhe é oferecido no acesso a internet, faz-se necessário o desenvolvimento de formas e ferramentas capazes de efetuar tal organização.

Considerando a capacidade dos seres humanos de organizar as informações que lhes são apresentadas a partir da criação de uma "memória semântica", capaz de processar ideias e conceitos que não são derivados da experiência pessoal, buscando informações na memória de longo prazo armazenada no cérebro humano, dando assim um significado a cada uma delas e possibilitando a diferenciação entre informações e a geração de conhecimento (CIÊNCIA ONLINE, 2017), avaliou-se a possibilidade de atribuir características próximas a estas a sistemas de computador, para que estes pudessem ser mais eficientes nas atividades de organização e busca de informações disponíveis na internet (BERNERS-LEE, 2001).

Para isso, grupos de estudos coordenados pela W3C (*Word Wide Web Consortium*) trabalham constantemente, desde meados da década de 1990, na ideia da Web Semântica que consiste em uma forma genérica na organização das informações disponíveis na Web.

Organizar informações e gerar conhecimento não é uma necessidade exclusiva da Web. Esta necessidade também pode ser vista em empresas que lidam com grandes volumes de informações que devem ser compreendidas, compartilhadas e acessadas por diferentes setores e em diferentes situações. Portanto, estas informações e conhecimentos devem estar armazenados de tal forma que o acesso a elas possa ser feito de forma simples por diferentes aplicações mantendo um bom desempenho nos processos de recuperação, já que hoje em dia a tecnologia é uma aliada das empresas na realização e controle de suas atividades (DE MOURA, 2011). Deste modo os conceitos de organização de dados ,oferecidos pela Web semântica também podem ser aplicados nas empresas como alternativa de organização de informações.

Este trabalho propõe a utilização dos princípios da Web semântica para o desenvolvimento de uma ferramenta capaz de armazenar e manipular informações e conhecimentos oriundos de diferentes fontes e/ou áreas de conhecimento específico. Explorando os conceitos de Ontologia aplicados na computação como forma de representação de conhecimento, valendo se do uso de Linguagens de marcação, para descrição de recursos através de arquivos RDF/XML e base de dados não estruturadas, como ferramenta de armazenamento das informações e conhecimentos contidas nestes arquivos afim de criar uma alternativa de armazenamento e manipulação de dados contidos nos arquivos RDF/XML.

A aplicação foi desenvolvida visando entre outros objetivos o ganho de desempenho nas atividades de manipulação destes arquivos e com isto tornar a organização de conteúdos mais eficiente tanto em aplicações para a Web como para aplicações empresariais.

Outro objetivo do trabalho é a compreensão das propostas do novo modelo de Web, que vem sendo desenvolvido e aprimorado pela W3C para a organização de conteúdos disponíveis na internet. A atividade proposta faz parte do projeto de pesquisa sobre Gestão de conhecimento de base de dados corporativos do Centro de ensino Superior de Juiz de fora com o objetivo de compartilhar o conhecimento gerado dentro da organização.

O restante do artigo segue organizado em seções a serem definidas. A Seção 2 descreve os princípios que definem demais conceitos sobre ontologias e as tecnologias utilizadas para o desenvolvimento da proposta deste trabalho. A Seção 3 discute características sobre o banco de dados MongoDB como proposta de armazenamento de dados representados semanticamente no formato RDF tecnologia que será apresentada nesta mesma seção. A Seção 4 apresenta o protótipo desenvolvido como proposta de implementação da manipulação de arquivos em RDF. Finalmente, a Seção 5 apresenta as considerações finais e trabalhos futuros.

## **2. Referencial Teórico**

Nesta seção serão descritos os conceitos que norteiam este trabalho, e que fundamentam as técnicas, procedimentos e implementação da solução proposta.

### **2.1. Ontologia como conceito utilizado na Web Semântica e representação do conhecimento**

Ontologia é um termo que tem origem nos estudos de Aristóteles. É possível recuperar as origens e definições sobre Ontologia como Estudo do "ser", ou estudo filosófico da natureza do ser, da existência, ou realidade, bem como as categorias básicas do ser e suas relações.

Assim, ao aprofundar nas formas de representar e obter uma Ontologia, alguns questionamentos podem ser feitos (ALMEIDA e BAX, 2003):

- O que existe ou pode ser dito que existe na realidade?
- O que é real no mundo?

Estes questionamentos discorrem de uma forma mais filosófica sobre o objetivo das ontologias que segundo Guarino e Welty (2000), tem como objetivo responder às seguintes questões: “Quais são as categorias das coisas que existem no mundo? E quais são as leis que regulam tais categorias?” para criar um sistema de categorização que explique determinada visão de mundo.

O conceito de ontologia como especificação de uma conceitualização é o mais utilizado como meio de representação de conhecimento na computação (GUARINO, 1998). O uso do conceito de ontologia tal como mencionado considera que o termo explicita que um objeto ou coisa do mundo real e que deve ser expresso formalmente de maneira clara por meio de conceitos e as relações existentes entre estes dentro de um determinado domínio (BREITMAN, 2000).

Dentre os tipos de ontologias apresentados por Almeida e Bax (2003), as ontologias de domínio têm sido amplamente utilizadas por pesquisadores das áreas de Ciência da Computação. Os trabalhos na área de computação relacionados com o uso de ontologias são motivados principalmente pelo crescente interesse pela Web Semântica (BREITMAN, 2000). A Web Semântica oferece às pessoas a capacidade de manterem dados na Web e também constroem vocabulários que representam ontologias descritas em OWL (*Ontology Web Language*) (W3C, 2017). As definições de Web Semântica e OWL serão apresentadas posteriormente visando esclarecer os fundamentos da proposta deste trabalho.

Existem várias interpretações do termo ontologia que podem ser aplicadas em outras áreas de conhecimento, porém para a Web Semântica a ontologia é usada com o intuito de criar modelos formais para a representação de conhecimento e significados existentes em um determinado domínio. Neste contexto o objetivo deste mecanismo é o registro de informações por meio de linguagens que possam ser interpretadas computacionalmente, utilizando tecnologias que possibilitam a elaboração de projetos de dados vinculados ou *Linked data* ou até mesmo que o conhecimento possa ser compartilhado a partir de representações semânticas (BREITMAN, 2000).

## **2.2. Web Semântica e *Linked Data***

A Web Semântica é definida por Tim Berners-Lee como uma extensão da Web obtida através da adição de semântica ao atual formato de representação de dados, tornando assim, a Web compreensível tanto para humanos quanto para máquinas, facilitando o relacionamento e o acesso à informação (BERNERS-LEE, 2009).

A Web Semântica permite a criação de uma comunicação mais colaborativa entre homem e máquina por meio do uso de tecnologias que ao aplicadas podem ser utilizadas para oferecer significados aos dados que são processados pela Web. Além disso, permite que softwares possam auxiliar de forma mais eficiente às atividades de busca de informação pelos humanos, atribuindo semântica e conhecimento aos dados processados pelas aplicações de software. Assim torna-se possível organizar e compreender as relações entre os termos processados, minimizando as perdas durante pesquisas (W3C, 2017).

O modelo de Web predominante atualmente proporciona uma publicação descontrolada de conteúdos e informações na rede nas quais em sua maioria as informações contidas nas publicações não são descritas de maneira correta, o que causa uma sobrecarga de

informações descontextualizadas no retorno das buscas por conteúdo. Tim Berners-Lee (2001) propôs a Web Semântica como uma extensão da Web anteriormente proposta por ele para reduzir ou eliminar os problemas estruturais do atual modelo de disponibilidade de dados na Web sugerindo padrões onde as informações pudessem ser descritas e categorizadas para eliminar possíveis ambiguidades nas buscas por conceitos na Web a partir de motores de busca (ou *search engines*). Para que essa ideia se tornasse realidade a Web Semântica passou a adotar o uso de ontologias e as tecnologias para representa-las computacionalmente, possibilitando o surgimento de outros conceitos da Web Semântica, como por exemplo, os dados vinculados (ou *linked data*) semanticamente.

*Linked data* é um conjunto de práticas para aprimoramento da "Web de dados" na elaboração da Web Semântica, para criar relacionamentos entre os dados disponíveis na rede de dados. A Web Semântica não consiste apenas em colocar dados na Web, trata-se de fazer links, para que uma pessoa ou máquina possa explorar a rede de dados. Assim, com os dados vinculados, torna-se possível encontrar de forma mais facilitada os dados relacionados e com mesmos significados.

Na "Web de dados" este relacionamento é feito através dos links (ou hipertextos) em HTML que ligam os dados através de um documento HTML a outro. Por outro lado, na Web Semântica existe uma ligação entre os dados por meio de uma estrutura que define como o dado deve estar relacionado, que são feitas através do uso de URIs (*Uniform Resource Identifier* - identificador uniforme de recursos). Nestas estruturas são encontradas as descrições de tais recursos, possibilitando assim a compreensão de seu significado descrito em OWL ou em RDF. Para isto Tim Berners-Lee também sugere quatro regras a elaboração de projetos Semânticos para a Web, são elas:

- Uso de URIs.
- Uso de HTTP URIs para que as pessoas possam procurar esses nomes.
- Por meio de uma URI são fornecidas informações úteis usando os padrões (RDF).
- Inclusão de links para outros URIs para que possam descobrir mais coisas.

### **2.3. Representação Semântica de Dados usando RDF**

A representação semântica de dados pode ser feita através da utilização de diversas tecnologias que vem sendo desenvolvidas tais como OWL e RDF entre outras. Neste trabalho abordaremos a utilização de arquivos RDF/XML com base nas características destas tecnologias.

Desenvolvido a partir de 1996 pelo XML (*eXtensible markup language*) Working Group patrocinado pela W3C, o XML é uma metalinguagem, ou seja, ela oferece recursos para a definição de gramáticas que caracterizam linguagens para classes de documentos específicos, com conjunto de elementos, atributos e regras de composição bem determinados (BRAY, 1997).

O XML pode ser definido também como uma linguagem de metamarcação como o HTML, porém diferentemente do HTML que trata se de uma linguagem que tem por objetivo a apresentação de conteúdos, o XML, busca a descrição de estruturas relacionando os marcadores ao significado dos conteúdos delimitados por eles. Com isto, torna se possível armazenar e recuperar dados em arquivos XML, arquivos estes que podem ser elaborados em qualquer editor de texto, já que o XML é uma linguagem baseada em texto (DE LIMA, 2005).

Além disso, o XML pode ser utilizado também para descrever imagens, gráficos e qualquer outro tipo de dado. Com isto, o XML está sendo cada vez mais utilizada, seja para a

construção de arquivos de configuração, seja para o intercâmbio de dados entre aplicações na Web ou estruturação e armazenamento de dados (DE LIMA,2005).

Em sua sintaxe o XML descreve os conteúdos e informações de forma estruturada e hierárquica de tal forma que um documento XML possui um único elemento raiz ao qual contém todos os demais elementos que são chamados de elementos filhos (BRAY, 1997). Seguindo esta estrutura, o XML permite com que seus usuários definam a sua própria linguagem de marcação adaptados a seus requisitos (DE LIMA,2005).

Os elementos são os componentes básicos do XML. Eles são criados pelo autor do documento XML, que deve atribuir a cada elemento um nome, que é representado no documento XML, delimitado pelas marcações < Nome\_do\_Elemento >, a qual indica o início do elemento e o < /Nome\_do\_Elemento > que indica o fim do mesmo (BRAY, 1997). Para cada elemento deve existir um marcador de início e um marcador de fim, e tudo que se encontra entre estes marcadores são características deste elemento. Quando um elemento carrega elementos filhos, estes filhos são considerados marcadores e não características do elemento. Isto se deve ao fato de um elemento poder armazenar um texto simples ou outro elemento dentre suas características, como pode ser visto na Figura 1.

```
1  <? xml version = "1.0" encodetype="Utf-8"?>
2
3  <obra>
4
5  <musica>
6      <faixa>01</faixa>
7      <titulo>Canção da America</titulo>
8      <compositor>Milton Nascimento</compositor>
9      <genero>MPB</genero>
10     <comentario> Musica escrita em parceria com <compositor> Fernando Brant</compositor>.</comentario>
11 </musica>
12
13 </obra>
```

Figura 1. Exemplo de documento XML - Fonte: Elaborado pelo autor (2017).

Na Figura 1 é apresentado um exemplo de documento XML onde se pode observar na primeira linha a descrição do tipo de documento que esta sendo criado no caso XML versão 1.0 e a codificação a ser utilizada na apresentação dos caracteres, a UTF-8. Esta linha deve estar obrigatoriamente contida em todos os documentos XML, pois é ela que possui as indicações das principais definições do documento XML. Na linha 3 (Figura 1) está declarado o elemento raiz da estrutura que esta finalizado na linha 13. No intervalo do elemento raiz encontramos o elemento "musica" que possui como sua descrição os elementos "Faixa", "titulo", "compositor", "gênero" e "comentário" elemento este que contém além de um texto outro elemento "compositor" que se refere a um coautor da música em sua descrição. Com este exemplo fica mais fácil a compreensão da sintaxe de um documento XML.

Apesar da possibilidade do uso de outras tecnologias, o modelo de descrição de recursos em RDF utiliza a linguagem XML como complemento da tecnologia para que se possam criar especificações de semântica de maneira padronizada. Também a linguagem XML é uma metalinguagem que define uma sintaxe para ser utilizada na criação de outras linguagens de marcação para um domínio específico, com estrutura e semântica próprias (SILVA e LIMA, 2002), podendo ser facilmente compreendida por seres humanos. É importante considerar o fato de que os documentos XML são menos restritos em sua elaboração, já que seus marcadores podem ser definidos de acordo com a necessidade do

usuário por serem mais flexíveis se comparados a outras metalinguagens como o HTML e também podem ser mais facilmente manipulados por máquinas (SILVA e LIMA, 2002).

A tecnologia *Resource Description Framework* (RDF) foi elaborada para possibilitar interoperabilidade entre sistemas que trocam informações a partir do processamento de metadados com base na linguagem XML (SILVA e LIMA, 2002). O objetivo geral do RDF é definir um mecanismo para descrever recursos sobre um domínio de aplicativo específico. A definição do mecanismo deve ser neutra ao domínio, contudo o mecanismo deve ser apropriado para descrever a informação sobre qualquer domínio (BERNERS-LEE, HENDLER e LASSILA, 2001). Com isso, essa tecnologia pode ser aplicada em diferentes ambientes adequando-se a atividade a qual ela é destinada a partir do desenvolvimento de softwares específicos para a mesma, o que levou a escolha desta tecnologia como parâmetro a ser utilizado neste trabalho.

Utilizando os princípios das ontologias que propõe a categorização das “coisas” existentes em um domínio, o RDF trata da relação entre essas coisas que são descritas como recursos que por sua vez devem ser sempre identificados por uma URI (Uniform Resource Identifier - Identificador Uniforme de Recurso), seguido opcionalmente por um Id de âncora que é um tipo de hiperlink usado para direcionar um usuário ao recurso que o utiliza.

Um recurso pode ser todo e qualquer elemento que esteja sendo descrita por uma expressão RDF, seja ela uma página Web, um arquivo ou até mesmo algo que não seja acessível por um sistema Web, como por exemplo, um CD de músicas, um livro, um carro, etc. Além dos recursos, o RDF estabelece em seu modelo básico de descrição de dados outros dois objetos que são: as propriedades, que podem ser uma característica, um atributo ou uma relação utilizada para descrever o recurso; e as declarações, que são o conjunto de um recurso específico com uma propriedade definida acrescido do valor dessa propriedade. Estas três partes individuais são chamadas respectivamente de sujeito, predicado e objeto e formam uma tripla RDF (Figura 2).



Figura 2. Modelo de representação de RDF em grafo. Fonte: Elaborado pelo autor (2017).

Um recurso RDF pode ser representado graficamente através do sistema de representação conhecido como diagrama de nós e arcos ou grafo rotulado onde os nós, representam os recursos e os arcos representam suas respectivas propriedades (Figura 2). É importante ressaltar que a direção do arco é relevante ao contexto do grafo, pois, a seta sempre aponta do sujeito para o objeto da declaração.

Para representar computacionalmente os recursos, o modelo de dados definido pelo RDF fornece uma estrutura abstrata e conceitual para definir e usar metadados. Uma sintaxe concreta também é necessária para criar e trocar esses metadados. Essa especificação usa a codificação XML como sua sintaxe de intercâmbio (BERNERS-LEE, HENDLER e LASSILA, 2001). A partir do uso do XML, o W3C define duas sintaxes para a codificação de modelos de dados RDF.

- Sintaxe de serialização básica: que apresenta uma estrutura mais clara para o documento RDF (Figura 3).
- Sintaxe de serialização abreviada que apresenta uma estrutura mais compacta de criação de documentos RDF (Figura 4).

Os recursos são descritos entre os marcadores `<RDF:Description>`, `</RDF:Description>`, estes marcadores podem possuir o parâmetro `about` (Figura 3, linha 7) que se refere a um URI ou um recurso existente, bem com o parâmetro `ID` (Figura 4, linha 4) que indica a criação de um novo recurso.

```

1 <rdf:RDF
2   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3   xmlns:s="http://description.org/schema/"
4
5   <rdf:Description about="http://www.cesjf.br">
6     <s:Orientador>
7       <rdf:Description about="http://www.cesjf.br/staffid/395">
8         <rdf:type resource="http://description.org/schema/Pessoa"/>
9         <v:Nome>Evaldo de oliveira</v:Nome>
10        <v:Email>evaldo.oliveira@cesjf.br</v:Email>
11      </rdf:Description>
12    </s:Orientador>
13  </rdf:Description>
14 </rdf:RDF>

```

Figura 3. Exemplo de documento RDF utilizando a serialização básica. fonte: Elaborado pelo autor (2017).

```

1 <rdf:RDF
2   <rdf:Description about="http://cesjf.br">
3     <s:Orientador>
4       <rdf:s:Pessoa id="http://www.cesjf.br/staffid/395">
5         <v:nome>Evaldo de Oliveira</v:nome>
6         <v:email>evaldo..oliveira@cesjf.br</v:email>
7       </s:Pessoa>
8     </s:Orientador>
9   </rdf:Description>
10 </rdf:RDF>

```

Figura 4. Exemplo de documento RDF utilizando a serialização abreviada. Fonte: Elaborado pelo autor (2017).

Em ambos os exemplos apresentados anteriormente (Figura 3 e Figura 4), o prefixo “s” indica o Namespace XML específico definido na declaração do documento. Um namespace XML é uma coleção de nomes que podem ser usados como nomes de atributo ou elemento em um documento XML (MICROSOFT, 2017).

Além das serializações apresentadas anteriormente outras sintaxes se destacam na representação de modelos RDF tais como Turtle, N-Triples, N-Quads, entre outras (DAVIS, STEINER e HORS, 2013).

Outro conceito importante a ser apresentado é o de Esquemas RDF (ou RDF-s). Os esquemas RDF são uma espécie de dicionário que define os termos que serão usados em declarações RDF e do significado específico para eles (BERNERS-LEE, HENDLER e LASSILA, 2001). O esquema RDF tem também por objetivo documentar as definições e restrições de uso de propriedades para evitar ambiguidade entre definições independentes e/ou conflitantes do mesmo termo, ou seja, em um RDF-s são descritas as diferenciações entre recursos utilizados em documentos RDF. Por exemplo, o termo "Oliveira" que pode ser uma

árvore bem como um sobrenome, então, esta diferenciação estaria descrita no RDF-s onde cada um dos termos portaria um *Namespace* diferente para identificá-lo.

Na Figura 5 pode-se observar o contexto do exemplo sugerido anteriormente no texto onde no grafo 01 o termo "Oliveira" indica o sobrenome do professor cujo nome é "Evaldo" e tem por URI "www.cesjf.br/Pessoa/professor/id/4321" professor este que é indicado como o Orientador do Aluno que é identificado pelo URI "www.cesjf.br/Pessoa/aluno/id/1234". Já no Grafo 02 representado nesta mesma Figura 5 temos a Espécie de árvore cujo URI é "www.cesjf.br/arvore/id/0001", nome científico é "Olea europaea" e nome popular é "Oliveira" espécie que pertence ao grupo de árvores existentes no domínio "www.cesjf.br/arvore". Note que os URIs utilizados na representação dos termos são diferentes, o que indica que eles trabalham com *Namespaces* diferentes, enquanto no Grafo 01 o URI trata de pessoas no Grafo 02 podemos notar a indicação de árvore.

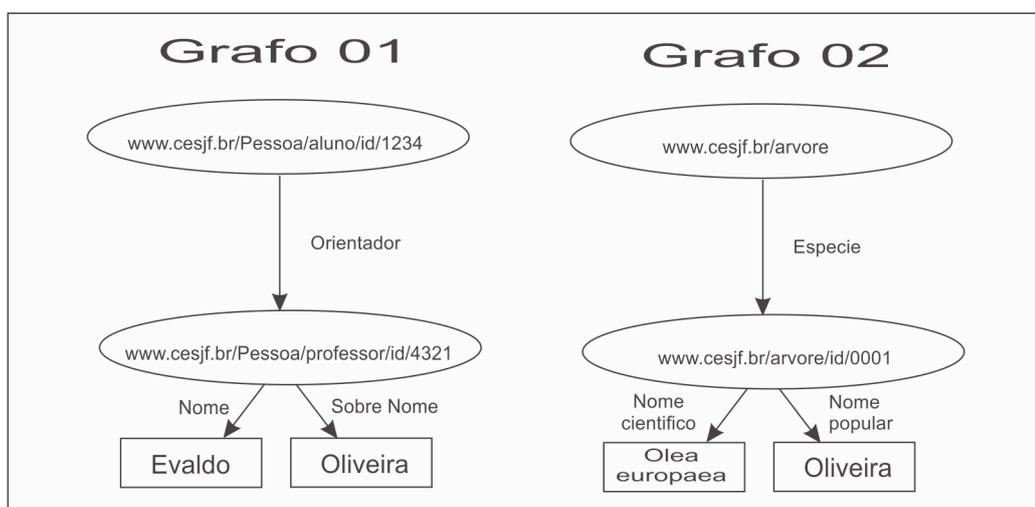


Figura 5. Representações do termo "Oliveira" em grafo. Fonte: Elaborado pelo autor (2017).

Assim como nos documentos XML, o conteúdo dos documentos RDF são armazenados no próprio arquivo, por isto, este artigo propõe o armazenamento do conteúdo e das estruturas descritas no documento RDF em uma base de dados como um meio de prevenção e alternativa de redundância a segurança das informações contidas no documento RDF, bem como, uma forma de ganho de desempenho na manipulação das informações contidas no arquivo, já que em todas as aplicações que manipulam arquivos RDF encontradas durante os estudos para a elaboração deste trabalho tais como o Apache Jena entre outros que serão apresentadas em seção posterior, utilizam a memória RAM (*Random Access Memory*) para armazenar o conteúdo do arquivo durante sua manipulação, o que pode ocasionar lentidão e travamentos em casos de manipulação arquivos que possuam um grande volume de informação, e para este sugere a utilização de um Banco de dados NoSQL.

## 2.4. Banco de dados NoSQL

Os bancos de dados NoSQL surgiram como uma alternativa aos Sistemas Gerenciadores de Banco de Dados Relacionais (SGBDR) com o objetivo de suprir as lacunas deixadas por este tipo de tecnologia de armazenamento de dados tais como a dificuldade de bases de dados relacionais de lidar com grande volume de dados (BOAGLIO,2015) o que vem sendo uma consequência da expansão da internet conforme mencionado na seção 1 deste artigo..

Os dados não estruturados podem ser entendidos como: “qualquer dado não gerenciado pela forma padronizada oferecida pelos padrões estabelecidos pelos sistemas gerenciadores de banco de dados” (IMON, 2014). O uso de SGBDR para armazenamento de

dados não estruturados oferece grande dificuldade para mantê-los com grandes volumes, principalmente quando se trata de dados oriundos da Web com o grande número de usuários manipulando vários tipos de informações estruturadas de forma distintas (AMAZON, 2017).

Com o crescente no volume de informações armazenadas e também na quantidade de acessos e de buscas simultâneas nas bases de dados o desempenho destas bases tendem a ser reduzidos e para solucionar este tipo de problema tem-se como alternativa o escalonamento das bases de dados que pode ser feito das seguintes formas (CARDOSO, 2012):

- Vertical: que consiste na substituição de hardwares nos servidores onde os dados são armazenados para aumentar seu desempenho, o que vem se tornando uma alternativa muito cara já que hardwares de auto desempenho tem um preço muito elevado.
- Horizontal: que consiste na distribuição das bases de dados em múltiplas instâncias, o que ocasiona um aumento na complexidade dos relacionamentos entre as tabelas podendo culminar na necessidade da contratação de novos DBA's (*Database Administrator*) para gerir todo este aumento de complexidade.

Os bancos de dados NoSQL implementam com facilidade a distribuição horizontal, com isto é possível discriminar suas informações em vários servidores que não necessariamente devem ser de alta performance consequentemente gerando um menor custo e também os bancos de dados NoSQL são bastante tolerantes a erros (PPTEC,2017).

Além dos problemas de escalabilidade outra lacuna deixada pelos bancos de dados relacionais é o armazenamento de objetos. Os bancos de dados relacionais nativamente não suportam esse tipo de dado tornando assim necessário programar o mapeamento dos objetos da aplicação para a estrutura da base de dados, já em bancos de dados NoSQL existem tipos de bases de dados que podem se adequar a estas necessidades de armazenamento de objetos. Os tipos de base de dados NoSQL existentes são (PPTEC,2017):

- Par chave / valor: este tipo de base de dados tem uma definição relativamente simples, onde para cada chave atribui-se um valor. Este tipo de base de dados NoSQL é considerado o mais importante dos tipos sendo que dele se derivam os demais tipos existentes. Alguns exemplos são:
  - Project Voldemort (VOLDEMORT, 2017)
  - Amazon SimpleDB (SIMPLEDB, 2017)
  - Oracle Berkeley DB (ORACLE BERKELEY, 2017)
- Colunas: uma evolução do modelo chave/valor que sugere que os pares chave valor sejam agrupados em linhas para que haja um mínimo de organização no armazenamento das informações. Alguns exemplos são:
  - Cassandra (CASSANDRA, 2017)
  - HiperTable (HYPERTABLE, 2017)
- Documentos: para este modelo de base de dados os armazenamentos são feitos a partir de documentos. Estes documentos normalmente são divididos em chave e valor onde nestes valores podem ser inseridos mais identificadores seguidos de seus respectivos valores. Alguns exemplos destes são:
  - CouchDB (COUCHDB, 2017)
  - RavenDB (RAVENDB, 2017)
  - MongoDB (MONGODB, 2017)

- Grafo: baseado na teoria de grafos, estas bases de dados armazenam objetos e não registros como é feito nos demais tipos de bases de dados, Estes possuem uma complexidade maior sustentada por algoritmo que percorrem os grafos em busca de informações de forma bem otimizada, que também proporcionam modificações na estrutura destes grafos sem grandes conflitos. Alguns exemplos são:
  - Neo4J (NEO4J, 2017)
  - InfoGrid (INFOGRID, 2017)
  - AllegroGraph (ALLEGROGRAPH, 2017)

Para o desenvolvimento do protótipo o MongoDB foi a base de dados escolhida para o armazenamento dos conteúdos oriundos dos arquivos RDF, isto foi motivado pelas facilidades encontradas em seus processos de instalação e uso, suas possibilidades de armazenamento de diferentes tipos de dados e a fácil compreensão da linguagem utilizada por ele. Algumas características do MongoDB e vantagens obtidas a partir do uso desta tecnologia para a implementação do protótipo serão descritas na seção a seguir.

#### **2.4.1. Características de armazenamento e manipulação de Dados não estruturados em MongoDB**

Os documentos RDF são utilizados como tecnologias para armazenamento da representação de dados gerados a partir dos modelos de ontologias e podem ser considerados como dados não estruturados. Além disso, é possível classificá-los como dados semânticos e são armazenados em documentos com a extensão RDF (SANTOS E CARVALHO, 2007).

Considerando que estes dados podem ser armazenados em arquivos no formato de texto, a integridade deste tipo de dado pode ser ameaçada, já que o documento pode ser corrompido durante seu processo de manipulação e armazenamento. Além disso, o processamento e indexação de dados por meio de sistemas de arquivo podem ser dificultados, criando um obstáculo para recuperação de dados e informações sobre os dados no formato RDF.

Com base nestas características, este trabalho também propõe que o gerenciamento de dados no formato RDF seja feito em um sistema gerenciador de dados não estruturados, a partir do banco de dados MongoDB, para reduzir os riscos gerados pela ameaça a integridade do arquivo durante sua manipulação e ganho de desempenho também no processo de manipulação, já que, o uso da memória RAM durante a manipulação de arquivos RDF podem ocasionar na redução do desempenho do computador no qual este processo esta sendo aplicado, considerando o fato de que o carga do arquivo para está memória RAM ocupará espaço que poderia ser utilizado por outros processos de manipulação.

As bases de dados NoSQL trabalham com um tipo de estrutura chamado *Schema free* no qual as bases de dados não têm uma estrutura fixa, isto é, não são obrigados a obedecer a rigidez de uma modelagem definida anteriormente a partir de uma normatização de base de dados. Isto gera uma base de dados mais flexível na qual uma mudança pode ser mais facilmente implementada em sua estrutura sem a necessidade de grandes alterações em estruturas ou até mesmo de reestruturação de todo o projeto como é necessário em muitas ocasiões em que se utiliza bancos de dados relacionais (CARDOSO, 2012). Com isto, o MongoDB proporciona o armazenamento de diferentes tipos de estruturas, o que facilita o armazenamento das estruturas ontológicas contidas nos arquivos RDF, que são manipulados pelo protótipo desenvolvido durante este estudo.

O MongoDB é um banco de dados que armazena documentos com informações no formato JSON (*JavaScript Object Notation*, Notação de objetos JavaScript), sendo um formato amplamente utilizado atualmente para troca de informações por meio da Web, sem a necessidade do uso de adaptadores (*wrappers*, *drivers* ou *plug-ins*) (BOAGLIO, 2015). O MongoDB tem sido utilizado como gerenciador de dados pelos autores destes trabalhos em grupos de pesquisa que tratam da extração de dados não estruturados, o que facilitou a escolha por esta tecnologia.

Por meio do JSON o MongoDB suporta os seguintes tipos de dados:

- Null - Vazio.
- Boolean - True ou False.
- Number - Número com sinal que pode ter uma notação com E Exponencial.
- String - Uma sequência com um ou mais caracteres.
- Object - Array não ordenado com itens do tipo chave valor, onde todas as chaves devem ser Strings distintos no mesmo objeto.
- Array - Lista ordenada de qualquer tipo, inteira entre colchetes e com elementos separados por vírgula.

Porém o JSON não implementa uma padronização para o formato de data nem como trabalhar com dados binários. Por este motivo o MongoDB introduz o BSON (Binary JSON) que é uma extensão ao JSON que além dos formatos de dados suportados por este também comporta:

- MinKey, MaxKey, TimeStamp - Tipos utilizados internamente no MongoDB.
- BinData - array de bytes para dados binários.
- ObjectId - Identificador único de registros do MongoDB.
- Date - Representação de data.
- Expressões regulares.

Com essa variedade de tipo de dados suportados pelo MongoDB e considerando a variedade de dados possíveis em diversos domínios do mundo real a sugestão do MongoDB como base de dados oficial da aplicação convém com as necessidades requisitadas na mesma que são:

- Armazenar estruturas contidas em ontologias RDF/XML.
- Permitir a revisão das estruturas contidas em uma Ontologia RDF/XML.
- Compartilhar o conhecimento armazenado na base de dados com outras aplicações a fim de mineração de dados para estudos de ciência de dados.

A partir das necessidades apresentadas foi criado um protótipo de aplicação em Java para manipulação de dados RDF armazenados no formato JSON pelo MongoDB. Esta aplicação se baseia nos conceitos apresentados neste trabalho para criação de uma ferramenta de gestão de representações semânticas de dados, será descrita posteriormente na Seção 4. As estruturas abaixo são aplicadas no protótipo para a compreensão de como os dados representados em RDF podem ser armazenados pelo MongoDB:

- Documento / Document - Estrutura onde são armazenadas as informações. Em um documento pode existir um valor simples ou uma lista de valores. Esta estrutura pode ser comparada as linhas e colunas contidas em uma tabela de uma base de dados relacional.
- Coleção / collection - Estrutura onde são agrupados os documentos criados. Esta estrutura pode ser comparada a uma tabela de uma base de dados relacional.
- Base de dados / database - Estrutura onde são armazenadas um conjunto de coleções. Estes se comparam aos bancos de dados relacionais.

O MongoDB disponibiliza uma aplicação de comunicação entre aplicações Java e seu banco de dados, trata-se do Mongo-Java-Driver que permite operações de manipulação dos conteúdos armazenados no banco, por exemplo, inserção, edição, exclusão e consulta de conteúdos nas coleções e documentos( MAGNITUDE,2017). Com isto, torna-se possível a recuperação e manipulação dos *Schemas* RDF (RDF-s) armazenados a partir da aplicação construída.

## 2.5. Manipulação de Dados com Representação Semântica

No intuito de apoiar as atividades referentes à geração e manipulação de dados semânticos, diversas ferramentas vêm sendo desenvolvidas, a seguir serão listadas algumas destas ferramentas já produzidas e suas respectivas funcionalidades dentro da área da Web Semântica.

- Apache Any23. Uma ferramenta que pode ser utilizada como uma biblioteca por aplicações Java que processam dados da Web, como uma ferramenta de linhas de comando a partir de seu modulo Apache Any23 Cli que permite a extração e a conversão de Documentos RDF em diferentes formatos e como uma API online para a conversão de Documentos RDF (APACHE ANY23 , 2017) .
- Sesame / RDF4J. Uma biblioteca escrita em Java e de código aberto, usada para processamento de dados RDF, e inclui a análise, armazenamento, inferência e consultas de dados em RDF. A ferramenta oferece dois tipos de armazenamento de dados que pode ser feito na memória e a loja nativa que permite o armazenamento em disco, com suporte a linguagem SPARQL para consultas Sezane. Também é compatível com diversos formatos RDF como RDF/XML, turtle, N-Triples entre outros ( RDF4J , 2017).
- Apache Jena. É *framework* desenvolvido em Java para a construção de aplicações de Web Semântica. Fornece um ambiente de desenvolvimento de rotinas para manipulação de dados no formato RDFs e OWL, SPARQL (*Protocol and RDF Query Language*), GRDDL (*Gleaning Resource Descriptions from Dialects of Languages*) e inclui um mecanismo de inferências (APACHE JENA, 2017).

As ferramentas apresentadas são livres e oferecem suporte para aplicação de sistemas de informação para manipulação de dados semânticos. Em especial, o *framework* Jena é amplamente utilizado devido ao volume de material de pesquisa e aplicações que o utilizam desde sua criação e manutenção feita pela Apache (APACHE JENA, 2017).

A próxima seção apresenta um protótipo de aplicação desenvolvida em Java para manipulação de dados no formato RDF utilizando o *framework* Jena e o banco de dados MongoDB.

### 3. Mongo/RDF: Aplicação para armazenamento e manipulação de dados semânticos em NoSQL

O objetivo do protótipo Mongo/RDF é armazenar e manipular dados semânticos que são extraídos de arquivos no formato RDF/XML utilizando como sistema gerenciador de dados o MongoDB onde os dados ficarão armazenados, possibilitando consultas e modificações a partir da aplicação que permite também o compartilhamento do conteúdo semântico extraído dos dados em RDF.

O protótipo tem como característica a manipulação dos conteúdos ontológicos descritos nos arquivos RDF, sendo feitas a partir do banco de dados MongoDB, o que minimiza o consumo de memória RAM, e com isto, gera um ganho de desempenho na execução destas atividades já que com o uso da aplicação não é necessária a carga do arquivo RDF para a memória RAM.

A aplicação foi desenvolvida a partir de um projeto de aplicação Java Web utilizando modelo MVC (*Model, View, Controller*). Foi necessário a utilização do *framework* Apache Jena 3.3.0, para tornar possível através da aplicação Java web a extração das classes ontológicas contidas no documento RDF.

Ao acessar o protótipo Mongo/RDF o usuário é levado a sua tela inicial ( Figura 06), nesta tela são encontradas todas as funcionalidades do Protótipo Mongo/RDF, que podem ser acessadas através dos menus no centro da tela com, exceção a funcionalidade "Excluir Schema" que esta presente apenas no menu lateral ou dos menus laterais que estão presente em todas as telas exibidas no protótipo.



Figura 6. Página inicial do protótipo Mongo/RDF. Fonte: Elaborado pelo autor (2017).

O procedimento de armazenamento dos dados é implementado na funcionalidade "Carregar RDF" da ferramenta Mongo/RDF conforme apresentado na Figura 7. Nesta funcionalidade, ao clicar no botão "Procurar" , o usuário pode localizar um arquivo RDF que terá o conteúdo de sua estrutura ontológica extraído e armazenado na base de dados criada no MongoDB. O conteúdo extraído do documento RDF é armazenado em uma *collection* que é criada com o mesmo nome do documento RDF dentro da base de dados previamente criada no MongoDB. Na *collection* criada, cada classe ontológica é descrita em um *document*, com isto é possível agrupar o conteúdo da estrutura do documento RDF em uma coleção, armazenando diferentes coleções em um mesmo banco de dados.



Figura 7. Funcionalidade “Carregar RDF” implementada no protótipo Mongo/RDF. Fonte: Elaborado pelo autor (2017).

Uma vez armazenado no MongoDB é possível visualizar as classes ontológicas extraídas do arquivo RDF a partir do Menu "Visualizar Schema", onde são listados os *schemas* do formato RDF armazenados no MongoDB. Esta listagem permite a seleção de um schema a ser visualizado conforme visto na Figura 8.

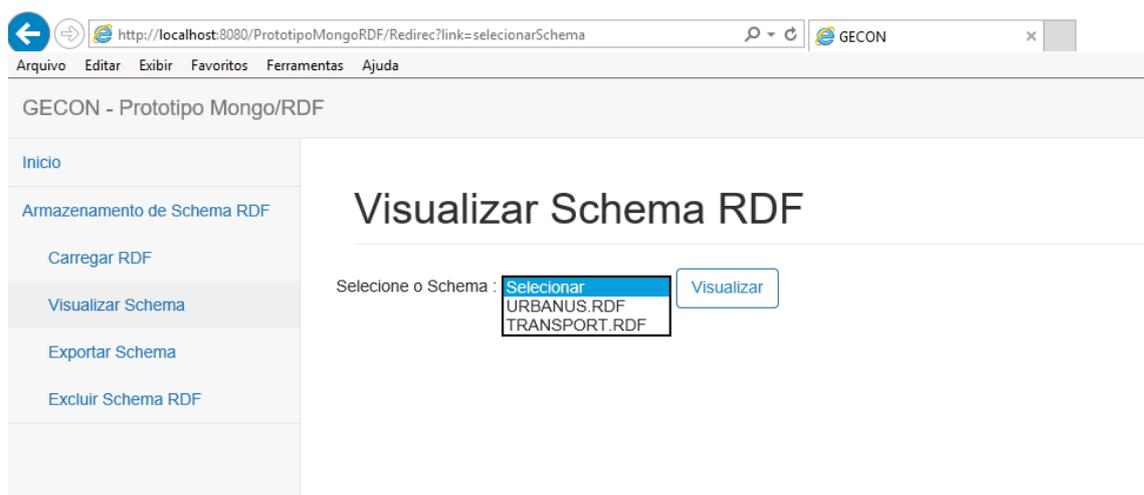


Figura 8. Funcionalidade “Visualizar Schema RDF” implementada no protótipo Mongo/RDF. Fonte: Elaborado pelo autor (2017).

Ao selecionar um *schema* e clicar no Botão Visualizar será exibido o conteúdo armazenado, a partir dessa visualização é possível fazer modificações na classe ontológica desejada, como edição do nome da classe, a URI e o comentário que descreve a classe selecionada clicando no botão "Editar registro" que permite a edição dos campos mencionados anteriormente para cada classe de ontologia armazenada no banco de dados. A Figura 9 apresenta a visualização de conteúdo extraído do arquivo RDF e também o botão Editar Registro que é utilizado para efetuar modificações no conteúdo extraído do arquivo RDF.

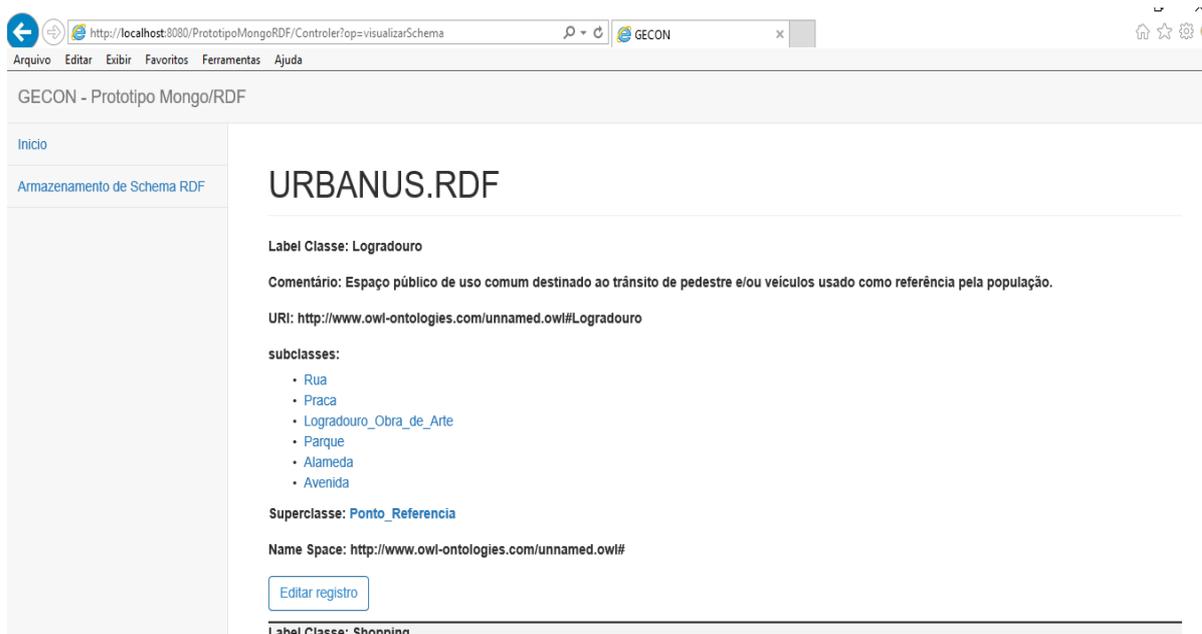


Figura 9. Apresentação do conteúdo armazenado no banco de dados. Fonte: Elaborado pelo autor (2017).

Além da visualização e modificação do conteúdo armazenado no MongoDB foi implementado também no protótipo a exclusão de *Schemas* armazenados na base de dados (Figura 10) e também a exportação de um *Schema* que pode ser feita para um novo arquivo RDF/XML ou um arquivo JSON como ilustrado na Figura 11. Esta Exportação permite que o usuário obtenha versões atualizadas de um arquivo RDF/XML, bem como a exportação das estruturas armazenadas no MongoDB para outras bases de dados ou até mesmo o compartilhamento destas estruturas armazenadas com outros sistemas.

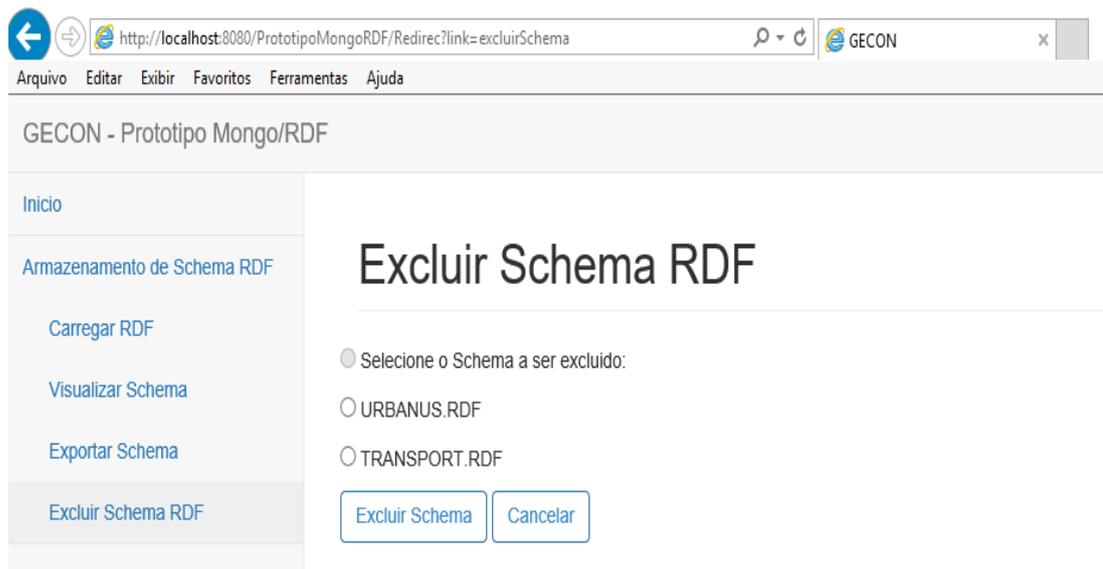


Figura 10. Funcionalidade Excluir Schema RDF implementada no protótipo Mongo/RDF. Fonte: Elaborado pelo autor (2017).

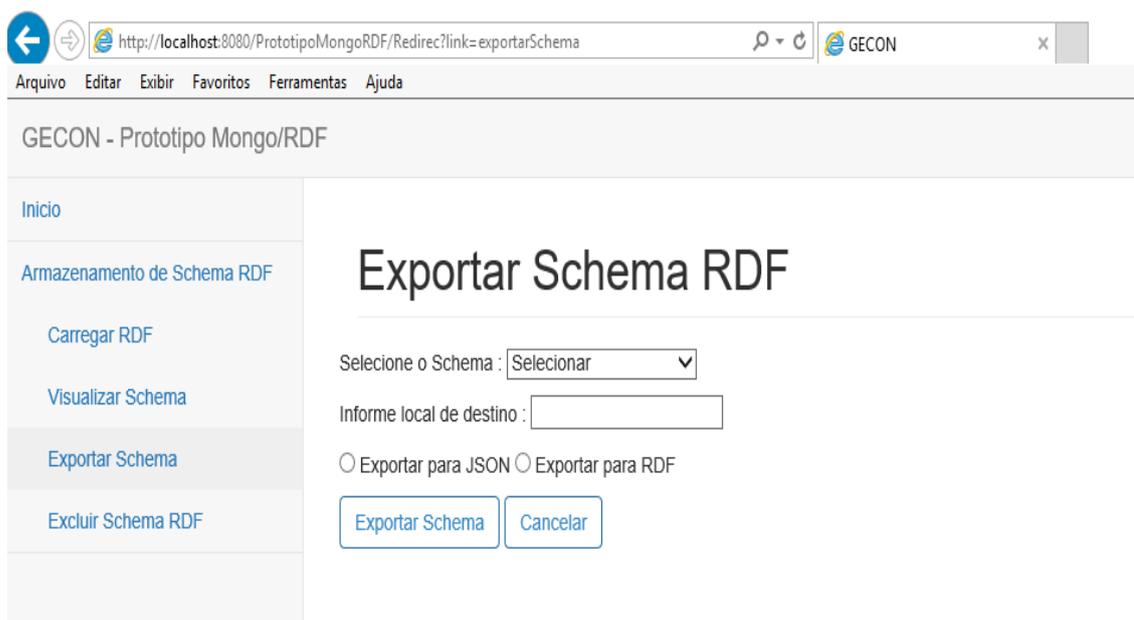


Figura 11. Funcionalidade Exportar Schema RDF implementada no protótipo Mongo/RDF.  
Fonte: Elaborado pelo autor (2017).

O acesso a base de dados MongoDB pode ser feito com o uso da classe MongoClient que permite a criação de um objeto do tipo cliente de conexão pelo método MongoClient() que acessa o serviço do MongoDB no servidor. Por padrão o objeto utiliza o endereço de servidor local (*localhost*) e a porta 27017 como caminho do serviço do MongoDB. Porém, este caminho pode ser indicado pelo desenvolvedor da aplicação caso necessário.

Com a conexão estabelecida, é possível se conectar a um banco de dados criando um objeto do tipo MongoDBDatabase atribuindo a ele o valor obtido no método getDatabase() do objeto criado anteriormente para se conectar ao serviço do MongoDB.

Para acessar uma coleção armazenada na base de dados cria-se um objeto do tipo MongoCollection e atribui a ele o resultado do método getCollection() disponível nos objetos do tipo MongoDBDatabase. É importante ressaltar que deve ser indicado o nome da coleção desejada de forma correta já que o MongoDB é uma aplicação "*case sensitive*", ou seja, diferencia caracteres maiúsculos e minúsculos entre outras especificações de caracteres alfa numéricos.

A partir do acesso a coleção é possível a edição de documento como o método updateOne() que altera apenas a primeira linha obtida como resultado de busca no MongoDB ou updateMany() atualizando todos os documentos que tiverem as condicionais indicadas na consulta. O MongoDB utiliza estas duas formas como uma medida preventiva a modificações equivocadas no banco de dados, esta característica também é encontrada nos métodos de exclusão deleteOne() e deleteMany().

O método Find() busca todos os documentos contidos na coleção e atribui o resultado desta busca a um objeto do tipo FindIterable conforme apresentado na Figura 12.

```

79 public FindIterable buscarSchema(String nomeColecao){
80     MongoCollection col = banco.getCollection(nomeColecao);
81
82     FindIterable retorno = col.find();
83     Document dc = new Document();
84
85     return retorno;
86 }

```

Figura 12: Implementação do método Find(). Fonte: Elaborado pelo autor(2017)

Para percorrer estes registros é necessário o uso do método Iterator() do objeto do tipo FindIterable (Figura 12) atribuindo o retorno deste método a um objeto do tipo MongoCursor que armazena os itens contidos no objeto FindIterable criando um laço de iteração sobre este. Veja um exemplo na Figura 13 a seguir.

```

158 public List<Document> criarSchema(String nomeColecao){
159     MongoCursor<Document> cursor = dao.buscarSchema(nomeColecao).iterator();
160     List<Document> Schema = new ArrayList<>();
161     while(cursor.hasNext()){
162
163         Document colecao = cursor.next();
164
165         Schema.add(colecao);
166     }
167     return Schema;
168 }
169 }

```

Figura 13: Exemplo de Iteração sobre MongoCursor. fonte: Elaborado pelo autor (2017).

As facilidades proporcionadas pelo MongoDB, tais como, a fácil conectividade entre a base de dados e a aplicação Java a partir do uso do Mongo-java-Driver bem como a fácil compreensão e uso dos métodos, propriedades e objetos disponíveis em sua biblioteca além da documentação que descreve com qualidade os recursos do MongoDB e seu modo de uso em aplicações facilita e motiva o uso desta tecnologia. Isto, tornou o desenvolvimento do protótipo mais eficiente e as funcionalidades implementadas coerentes com o objetivo do protótipo.

#### 4. Considerações Finais e Trabalhos Futuros

Por meio da abordagem utilizada neste trabalho, foi mostrado que, é possível aplicar as tecnologias da Web semântica como uma alternativa para gestão de informações e conhecimentos, tornando a organização destes conteúdos mais eficiente, de tal forma que, a busca por conteúdos obtenhas resultados mais coniventes com a pesquisa realizada. Para isto conceitos como o de Ontologia e Linked data são utilizados na elaboração de projetos semânticos.

Os métodos apresentados na seção anterior foram utilizados no protótipo desenvolvido a fim de manipular os dados armazenados na base de dados criada no MongoDB. Este protótipo foi criado para oferecer uma alternativa para o armazenamento e a manipulação de dados semânticos conforme a proposta inicial do estudo exemplificando tal atividade. A construção do protótipo mostrou que é possível manipular estes tipos de dados. Porém, é necessário compreender e participar da evolução dos conceitos apresentados neste trabalho, a fim de implementar novas funcionalidades ao

protótipo apresentado que abranjam além do armazenamento e manipulação de representações semânticas, o desenvolvimento de algoritmos que possam realizar buscas semânticas.

A partir do desenvolvimento do protótipo, trabalhos futuros, visando a avaliação de desempenho da aplicação desenvolvida nas atividades de manipulação de dados semânticos, comparando os resultados com ferramentas já existentes a fim de comprovar a eficiência do uso desta forma de armazenamento e manipulação de representação de dados semânticos utilizando as tecnologias Java e MongoDB podem ser desenvolvidos. Contribuindo com a evolução dos novos conceitos de Web quem vem sendo trabalhado pela W3C e outras instituições. Assim, a implementação de triplas RDF bem como a realização de buscas semânticas a partir dos *schemas* armazenados no MongoDB, são também alguns dos objetivos de trabalhos futuros.

Além destes o aprofundamento nas possibilidades de uso de tecnologias apresentadas na seção 2.5. e também nos conceitos apresentados nas sessões anteriores a citada e na base de dados MongoDB para fim de elaboração de estudos e ferramentas pertinentes a área de ciência e mineração de dados juntamente com o grupo de estudos de ciência de dados existente no Centro de Ensino Superior de Juiz de fora, Grupo este que em paralelo a este trabalho desenvolve uma aplicação no intuito de validar Ontologias e gerar arquivos OWL e/ou RDF com estas, com o objetivo de futuramente integrar estas partes (protótipos) desenvolvidas em uma única aplicação capaz de gerir informações oriundas de ambientes empresariais, contribuindo com a compreensão dos temas aqui apresentados e sua utilidade na organização de informações e gestão do conhecimento gerado por estas.

## Referências

- ALEGROGRAPH. **Allegro Graph**. Disponível em <https://franz.com/agraph/allegrograph/>. Acesso em 03 de Novembro de 2017.
- ALMEIDA, M. B.; BAX, M. P. **Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção**. Ciência da Informação, Brasília DF, v. 32, n. 3, p. 7-20, 2003.
- AMAZON. **O que é NoSQL**. Disponível em: <https://aws.amazon.com/pt/nosql/>. Acesso em 03 de Novembro de 2017.
- APACHE ANY23. **Introduction to Apache Any23**. Disponível em: <https://any23.apache.org/index.html>. Acesso em 04 de novembro de 2017.
- APACHE CASSANDRA. **What is Cassandra?**. Disponível em: <http://cassandra.apache.org/>. Acesso em 03 de Novembro de 2017.
- APACHE JENA. **Jena a Java RDF API and Toolkit**. Disponível em: [https://www.w3.org/2001/sw/wiki/Apache\\_Jena](https://www.w3.org/2001/sw/wiki/Apache_Jena). Acesso em: 04 de novembro de 2017.
- BERNERS-LEE, Tim. **Linked Data** (2009). Disponível em: <https://www.w3.org/DesignIssues/LinkedData.html>. Acesso em 13 de Setembro de 2017.
- BERNERS-LEE, Tim, James Hendler, and Ora Lassila. "The semantic web." Scientific American 284.5 (2001): 28-37.
- BERNERS-LEE, Timothy J. **Information management: A proposal**. No. CERN-DD-89-001-OC. 1989.
- BRAY, Tim et al. **Extensible markup language (XML)**. World Wide Web Journal, v. 2, n. 4, p. 27-66, 1997.

- BREITMAN, Karin Koogan. **Web semântica: a internet do futuro**. Grupo Gen-LTC, 2000.
- BOAGLIO, Fernando. **MongoDB. Construa novas aplicações com novas tecnologias**. Casa do código, Vila Mariana São Paulo, 2015.
- CARDOSO. Ricardo Manoel Fonseca. **Base de dados NoSQL**. Instituto Superior de Engenharia do porto. Porto - Portugal Novembro de 2012.
- COUCHDB. **CouchDB**. Disponível em <http://couchdb.apache.org/>. Acesso em 03 de Novembro de 2017.
- CIÊNCIA ONLINE. **Memória semântica: definição e exemplo**. Disponível em: <http://web.tecnico.ulisboa.pt/~ist425696/wordpress/ciencia-e-tecnologia/a-memoria-humana-como-funciona/>. Acessado em: 30 de outubro de 2017.
- DAVIS, Ian; STEINER, Thomas; HORS, A. L. **Rdf 1.1 json alternate serialization (rdf/json)**. W3C Working Group Note. W3C, v. 7, 2013.
- DE LIMA. Júnior César e Cedric Luis de Carvalho. **Extensible Markup Language (XML)**. Instituto de informática Universidade Federal de Goiás. junho de 2005.
- DE MOURA. Artur Alves e Tarcisio Campanholo. **Tecnologia da informação aliada a gestão do conhecimento na melhoria da estratégia e desempenho organizacional**. Revista da Católica v.3 n.5 Jan/Jul de 2011.
- HIPERTABLE. **Hypertable**. Disponível em: <http://www.hypertable.org/>. Acesso em 03 de Novembro de 2017.
- INFOGRID. **The Web Graph Database**. Disponível em <http://infogrid.org/trac/>. Acesso em 03 de Novembro de 2017.
- GUARINO, Nicola. **Formal Ontology and Information Systems**. In: Proceedings of the First Int. Conference on Formal Ontology in Information Systems, Trento, Italy, Junho 1998.
- GUARINO, N.; WELTY, C. 2000. **A formal ontology of properties**. In R. Dieng, Ed., Proceedings of 12th Int. Conf. On Knowledge Engineering and Knowledge Management, Springer Verlag, 2000.
- MACHADO. André. **Estudo da EMC prevê que volume de dados virtuais armazenados será seis vezes maior em 2020**. O globo 10 de abril de 2014. Disponível em: <https://oglobo.globo.com/sociedade/tecnologia/estudo-da-emc-preve-que-volume-de-dados-virtuais-armazenados-sera-seis-vezes-maior-em-2020-12147682>. acessado em: 03 de novembro de 2017.
- MAGNITUDE Simba. **MongoDB ODBC e JDBC Driver com SQL connection**. Disponível em: <https://www.simba.com/drivers/mongodb-odbc-jdbc/>. Acessado em: 03 de novembro de 2017.
- MICROSOFT. **Usar Namespace XML**. Disponível em: [https://msdn.microsoft.com/pt-br/library/ms754539\(v=vs.85\).aspx](https://msdn.microsoft.com/pt-br/library/ms754539(v=vs.85).aspx). Acessado em: 30 de outubro de 2017.
- MONGODB. **MongoDB Atlas Database as a Service**. Disponível em: <https://www.mongodb.com/>. Acesso em 03 de Novembro de 2017.
- IMON, W.H. **Untangling the Definition of Unstructured Data**. 2014. Disponível em: <http://www.ibmbigdatahub.com/blog/untangling-definition-unstructured-data>. Acesso em 08 de Novembro de 2017.

- NEO4J. **Introducing the Neo4j Graph Platform**. Disponível em: <https://neo4j.com/>. Acesso em 03 de Novembro de 2017.
- ORACLE BERKELEY. **Oracle Berkeley DB**. Disponível em: <http://www.oracle.com/technetwork/database/database-technologies/berkeleydb/downloads/index.html>. Acesso em: 03 de Novembro de 2017.
- PPTEC. **Banco de dados não relacional - NOSQL**. Disponível em: <http://www.pptec.com.br/banco-de-dados-nao-relacionais-nosql/>. Acessado em: 01 de novembro de 2017.
- PYPL. **PYPL PopularitY of Programming Language**. Disponível em: <https://pypl.github.io/PYPL.html>. acessado em: 01 de novembro de 2017.
- RAMALHO, Rogério Aparecido de Sá. **Web semântica: Aspectos interdisciplinares da gestão de recursos informacionais no âmbito da ciência da computação**. Marília - SP. 2006.
- RAVENDB. **RavenDB**. Disponível em <https://ravendb.net/>. Acesso em 03 de Novembro de 2017.
- RDF4J. **The Eclipse framework**. Disponível em: <http://rdf4j.org/about/>. Acesso: 04 de novembro de 2017.
- SANTOS, Fernando chagas. CARVALHO, Cedric Luis. **Aplicações de suporte a Web Semântica**. Instituto de Informática - Universidade Federal de Goiás. 2007.
- SILVA, Geiza Cristina da; LIMA, Tarcísio de Souza. "**RDF e RDFS na Infra-estrutura de Suporte à Web Semântica**." Revista Eletrônica de Iniciação Científica da Sociedade Brasileira de Computação. Porto Alegre, Ano II 2 (2002).
- SIMPLEDB. **Amazon SimpleDB**. Disponível em: <https://aws.amazon.com/pt/simpledb/>. Acesso em 03 de Novembro de 2017.
- TIOBE INDEX. **TIOBE index for october 2017> October Headline: Swift is losing popularity**. Disponível em: <https://www.tiobe.com/tiobe-index/>. Acessado em: 01 de novembro de 2017.
- W3C. **Web Semântica**. Disponível em: <http://www.w3c.br/Padroes/WebSemantica> Acesso em 13 de Setembro de 2017.
- VIEIRA, Marcos Rodrigues et al. **Bancos de Dados NoSQL: conceitos, ferramentas, linguagens e estudos de casos no contexto de Big Data**. Simpósio Brasileiro de Bancos de Dados, 2012.
- VOLDEMORT. **Project Voldemort**. Disponível em: <http://www.project-voldemort.com/voldemort/>. Acesso em 03 de Novembro de 2017.