

# Desenvolvimento de uma ferramenta para anotação semântica utilizando o framework JENA baseado em Ontologias de Domínio

Filipe Pacheco Quina, Evaldo de Oliveira da Silva

Curso de Bacharelado em Sistemas de Informação – Centro de Ensino Superior de Juiz de Fora (CESJF) – Campus Academia  
36016-000 – Juiz de Fora – MG– Brasil

{filipe.pacheco00, evaldo.oliveira}@gmail.com

**Abstract.** *The article shows a prototype developed as proof of concept in order to verify how the Jena framework provides the implementation of its methods for the development of semantic applications. In addition, we present the Jena framework methods that were used to create the proposed prototype that is used in the Corporate Knowledge Management Study Group (GECON), as well as its development in practical examples. In addition, this article presents concepts about ontologies and languages that can represent knowledge based on concepts of existing ontologies in the area of Computer Science. Finally, we show the results obtained from the implementation of the tool that represent the work on the representation of ontologies and the generation of semantic data using this prototype.*

**Resumo.** O artigo mostra um protótipo desenvolvido como prova de conceito, a fim de verificar como o Jena *framework* fornece a implementação dos seus métodos para o desenvolvimento de aplicações semânticas. Além disso, são apresentados os métodos do Jena *framework* que foram usados para a criação do protótipo proposto que é usado no Grupo de Estudos sobre Gestão de Conhecimento em Bases de Dados Corporativas (GECON), bem como seu desenvolvimento em exemplos práticos. Além disso, este artigo apresenta conceitos sobre ontologias e linguagens que podem representar o conhecimento baseado em conceitos de ontologias existentes na área de Ciência da Computação. Finalmente, são mostrados os resultados obtidos a partir da implementação da ferramenta que representam o trabalho sobre a representação de ontologias e a geração de dados semânticos usando este protótipo.

## 1. Introdução

Muitas organizações, independentemente do tamanho ou capacidade de produção, necessitam de informações capazes de serem utilizadas como forma de melhorar o processo de tomada de decisão ou dados que possam representar o conhecimento produzido por meio das transações realizadas. No entanto, o conhecimento dentro das organizações é mantido de forma desorganizada, por meio de arquivos de documentos, banco de dados relacionais e dados publicados na Web, e além disso, os termos utilizados para definir as regras de negócios não estão representados de forma explícita. Esse ambiente organizacional pode dificultar o compartilhamento e disseminação de informações que possam ser úteis para diversas atividades, desde a reutilização de conceitos e terminologias, até a construção de sistemas de informações com base na definição de requisitos.

É necessário que haja a gestão do conhecimento dos dados gerados dentro das organizações por meio de vocabulários e terminologias, tratando assim dos significados precisos dos elementos existentes e não somente de formatos ou sintaxes. Neste contexto, a Ciência da Computação tem utilizado ontologias como área de pesquisa para compartilhar e reutilizar conhecimento dentro de um domínio específico, a fim de descrever classes de objetos de diversas aplicações (ALMEIDA e BAX, 2003; GUARINO, 1998; GRUBER, 1993). Porém, torna-se necessário manipular as ontologias de forma computacional visando integrá-las ao ambiente computacional para que possam servir de mecanismo para compartilhamento e troca do conhecimento por ela modelado. Esta necessidade é discutida por Casanova et. al. (2005), tratando as ontologias representadas computacionalmente como um mecanismo de mediação, visando estabelecer a interoperabilidade entre aplicações que tratam conceitos distintos de fenômenos geográficos.

Desta forma, este trabalho aborda as tecnologias que são utilizadas para manipular ontologias e formatos de dados que possam servir como representações semânticas, uma vez que ontologias podem ser usadas para este fim, este artigo também apresenta estudos realizados com base nas áreas da Web Semântica e representação do conhecimento, com o objetivo de desenvolver rotinas de software para processar e visualizar ontologias representadas por linguagens computacionais. Além disso, é apresentado um protótipo que permite a geração de dados representados semanticamente a partir de ontologias que são manipuladas pelo próprio protótipo que é parte dos estudos realizados pelo Grupo de Estudos sobre Gestão de Conhecimento em Bases de Dados Corporativas (GECON), no qual participam alunos dos cursos de Engenharia de Software e Sistemas de Informação do Centro de Ensino Superior de Juiz de Fora (CESJF), sendo que ele é uma camada intermediária que recebe um arquivo de ontologia e o anota semanticamente.

O restante do artigo segue organizado em seções a serem definidas. A Seção 2 descreve os conceitos que definem as ontologias, e as tecnologias utilizadas para o desenvolvimento da proposta deste trabalho. A Seção 3 discute o *framework* Jena mantido

pela Apache (APACHE, 2017) como recurso de linguagem de programação para manipulação de ontologias e formatos de representação de dados semânticos. A Seção 4 apresenta o protótipo desenvolvido como proposta de implementação da manipulação de ontologias, e 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 em Computação**

O termo conceituação diz respeito a uma coleção de objetos, conceitos e entidades existentes em um determinado domínio e os relacionamentos entre eles. Uma conceituação é uma visão abstrata e simplificada do mundo que se deseja representar (ALMEIDA e BAX, 2003).

Uma ontologia é uma especificação da conceituação, descrita por conceitos e relacionamentos que podem existir no mundo real. A conceituação pode ser desenvolvida contendo terminologias e vocabulários, estabelecendo propriedades e permitindo que o conhecimento seja reutilizado, evitando o retrabalho ou a redescoberta de terminologias equivalentes (GUARINO, 1998). Por exemplo, ao ser aplicada na modelagem de banco de dados, uma ontologia pode permitir que os elementos modelados possam ter seus significados definidos por uma taxonomia que descreva os objetos do banco de dados.

Gruber (1993) define uma ontologia como uma descrição de conceitos e relacionamentos que podem existir por meio de um agente ou uma comunidade de agentes. Esta definição está relacionada com o uso da ontologia como uma coleção de definições, usada inclusive na filosofia. Na área de ciência da computação há um consenso do uso de ontologias com o propósito de permitir que o conhecimento seja compartilhado e reutilizado por meio de vocabulários e coleções de conceitos.

O uso de ontologias em Inteligência Artificial (IA) como forma de compartilhamento do conhecimento permite o acesso a vocabulários para elaboração de perguntas e consultas, respeitando os conceitos e as relações existentes entre eles. Em IA são desenvolvidos agentes de software capazes de acessar a ontologia, onde os conceitos e suas relações podem ser compartilhados permitindo a troca do conhecimento (GRUBER, 1993).

### **2.2. Ontologias de Domínio**

Dentre os tipos de ontologias apresentados por Almeida e Bax (2003), as ontologias de domínio têm sido bastante usadas por pesquisadores da área de Ciência da Computação.

Nos últimos anos houve um aumento de trabalhos na área de computação relacionados com o uso de ontologias, motivados principalmente pelo crescente interesse pela Web Semântica.

O conceito de Ontologias de Domínio definido por Mizoguchi, Vanwelkenhuysen e Ikeda (1995) descreve este tipo de ontologia como um conhecimento reutilizado somente em um domínio específico. Os domínios podem estar relacionados, por exemplo, com diversas aplicações contendo conceitos existentes no espaço geográfico, no desenvolvimento de aplicações ou na definição de conceitos empresariais (SILVA, LISBOA e GONÇALVES, 2008; ARANTES, FALBO e GUIZZARDI, 2007; BERTOLLO ET AL., 2002). Elas fornecem vocabulários sobre os conceitos dentro de um domínio e seus relacionamentos, como também as propriedades e atividades desse domínio.

Arantes, Falbo e Guizzardi (2007) discutem o uso de ontologias de domínio para o gerenciamento de configuração de software (*Software Configuration Management – SCM*). Propõem ainda, uma ontologia de SCM a ser usada como referência para entendimento sobre o domínio de configuração de software e integração semântica das ferramentas de engenharia de software usadas neste processo.

Bertollo et al. (2002) apresentam o ODE (*Ontology-based software Development Environment*), que é o ambiente de desenvolvimento de software baseado em ontologias de domínio. O objetivo do ODE é prover um ambiente capaz de suportar todo o processo de desenvolvimento em Engenharia de Software, com diversas ferramentas CASE (*Computer-Aided Software Engineering*) integradas e trabalhando em conjunto. A integração dessas ferramentas pode ser facilitada por meio das ontologias, onde os conceitos envolvidos estão bem definidos. No processo de desenvolvimento de software, cada ferramenta pode funcionar independentemente das outras. Porém, caso estejam integradas em um mesmo processo de software, é necessário coordenar a ativação e a comunicação entre elas. Para que esse processo ocorra entre as ferramentas, os conceitos envolvidos no desenvolvimento de software precisam estar bem definidos e devem ser compartilhados pelas ferramentas. Esse compartilhamento pode ser alcançado no ambiente de desenvolvimento de software por meio do uso de ontologias de domínio (BERTOLLO et al., 2002).

Felicíssimo et al. (2003) mostram uma ferramenta semiautomática para geração de ontologias subsidiada pela Engenharia de Requisitos. Essa ferramenta é um plug-in para uma ferramenta de edição de Cenários e LAL (Léxico Ampliado da Linguagem), denominada C&L. A C&L é o resultado de um projeto coordenado pelo grupo de pesquisa em Engenharia de Software do departamento de informática da Pontifícia Universidade Católica do Rio de Janeiro.

Segundo os autores supracitados, uma ontologia pode ser um produto da Engenharia de Requisitos, sendo responsabilidade do engenheiro de requisitos modelá-la

durante o processo de definição do software, a partir da modelagem e análise de informações relevantes.

Na área de bancos de dados, as ontologias de domínio estão presentes na criação de mecanismos para integração e interoperabilidade de bases de dados heterogêneas (CASANOVA et al., 2005), por isto os sistemas de informação devem ser capazes de compartilhar e trocar informações, resolvendo a interoperabilidade semântica em ambientes, como a Web, ou por meio de computação distribuída.

### **2.3. Web Semântica e suas aplicações**

A Web Semântica é conceituada por Tim Berners-Lee (BERNERS-LEE, 1998; BERNERS-LEE et al, 2002) como um aperfeiçoamento da Web feita através da adição de semântica atualmente utilizada na Web para representação de dados, tornando assim a Web compreendida tanto para humanos quanto para máquinas a fim de facilitar o relacionamento e o acesso à informação.

Uma das propostas da Web Semântica permite a criação de uma comunicação mais aperfeiçoada entre homem e máquina de forma colaborativa, possibilitando que as soluções computacionais possam auxiliar de forma mais eficaz as atividades de busca de informação pelos humanos, atribuindo semântica e conhecimento às soluções que são utilizadas. Assim, é possível criar um mecanismo entre o humano e a solução computacional que organiza o conhecimento inserido na Web, permitindo maior compreensão das pesquisas realizadas e melhoria da interoperabilidade entre aplicações (W3C, 2017).

Atualmente, a Web possui um modelo predominante que 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 sem conexões entre termos no retorno das buscas por conteúdo. Desta forma, Tim Berners-Lee (BERNERS-LEE et al, 2002) propôs a Web Semântica como um aperfeiçoamento da Web visando reduzir ou eliminar os obstáculos na estrutura atual no modelo de dados na Web. Este aperfeiçoamento sugere padrões onde as informações possam ser descritas e categorizadas para eliminar possíveis ambiguidades nas buscas por conceitos na Web a partir de motores de busca. Para que esta proposta se tornasse realidade a Web Semântica passou a adotar o uso de ontologias e as tecnologias para representá-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 (*Ontology Web Language*) ou em RDF (*Resource Description Framework*), nos quais serão descritos na seção 2.4.

#### **2.4. Representação e Manipulação de Ontologias por meio de Tecnologias da Informação**

Uma ontologia necessita ser especificada e representada na computação por meio de Tecnologias da Informação (TI), com o uso de alguma linguagem formal para que seja compreendida, compartilhada e reutilizada por especialistas, usuários ou projetistas de sistemas de informação. Existem várias linguagens de especificação de ontologias, com formalismos diferentes.

Algumas destas tecnologias são baseadas em XML (*eXtensible Markup Language*) como XOL (*XML-Based Ontology Exchange Language*), OWL (MCGUINNESS e VAN HARMELEN, 2004) e RDF (*Resource Description Framework*) (BECKETT e MCBRIDE, 2004). Outras linguagens são baseadas em lógica de descrição, como por exemplo, KIF (*Knowledge Interchange Format*) (GENESERETH e FIKES, 1994). Existem também combinações da linguagem XML com lógica de descrição.

A linguagem OWL é usada para descrever ontologias. Ela foi projetada pelo *Web Ontology Working Group*, que é o grupo responsável pelo desenvolvimento de linguagens para especificação de ontologias para a Web Semântica. A OWL permite a representação do conhecimento existente em Ontologias determinando conceitos, propriedades e suas relações definidas em um domínio de aplicação. Em OWL cada conceito é descrito como uma classe, e cada classe é descrita em um frame. De forma mais detalhada, as classes são definidas por atributos (slots) e estruturadas de acordo com o relacionamento com as subclasses dentro de um grafo ou de uma taxonomia. Abaixo segue um trecho de código escrito em OWL (Figura 1).

```

1  <?xml version="1.0"?>
2  <rdf:RDF
3      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4      xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
5      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
6      xmlns:owl="http://www.w3.org/2002/07/owl#"
7      xmlns="http://www.owl-ontologies.com/unnamed.owl#"
8  xml:base="http://www.owl-ontologies.com/unnamed.owl">
9  <owl:Ontology rdf:about=""/>
10 <owl:Class rdf:ID="Clube">
11   <rdfs:subClassOf>
12     <owl:Class rdf:ID="Cultura_e_Lazer"/>
13   </rdfs:subClassOf>
14 </owl:Class>
15 <owl:Class rdf:ID="Bar_Restaurante">
16   <rdfs:subClassOf>
17     <owl:Class rdf:ID="Servicos"/>
18   </rdfs:subClassOf>
19 </owl:Class>

```

Figura 1. Exemplo de um código em OWL (SILVA, 2008).

A figura acima descreve um documento de ontologia simples que contém o cabeçalho que possui as importações necessárias para que seja tipificado este arquivo como uma ontologia, mas ele também necessita que importamos algumas definições de XML por ser uma linguagem de marcação. Na parte inferior temos alguns exemplos de como são definidas as classes com suas respectivas propriedades (a partir da linha 10). A fim de detalhar melhor o código da Figura 1, serão explicadas as seguintes linhas contidas no cabeçalho, na linha 1 é o cabeçalho padrão de um documento xml, na linha 3 é feita uma importação da sintaxe RDF, na linha 4 é uma importação do esquema XML, na linha 5 é demonstrado uma importação do esquema RDF, na linha 6 é importado a definição do OWL, finalmente na linha 7 é feita a definição da URI, ou seja, a “chave” para se ter acesso as classes da ontologia.

A linguagem RDF é desenvolvida pelo W3C (World Wide Web Consortium) e serve para representar a informação na Web utilizando a sintaxe XML. Permite especificar também, os vocabulários usados na Web por meio de classes, propriedades, tipos, intervalos e domínios (BECKETT e MCBRIDE, 2004).

```

1 <rdf:RDF
2   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3   xmlns:owl="http://www.w3.org/2002/07/owl#"
4   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
5   xmlns="http://www.owl-ontologies.com/unnamed.owl#"
6   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
7 <owl:Ontology rdf:about="http://www.owl-ontologies.com/unnamed.owl"/>
8 <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#Avenida">
9   <rdfs:subClassOf>
10    <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#Logradouro"/>
11  </rdfs:subClassOf>
12 </owl:Class>
13 <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#Hospital">
14   <rdfs:subClassOf>
15    <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#Edificacao de Referencia"/>
16  </rdfs:subClassOf>
17 </owl:Class>
18 <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#Shopping">
19   <rdfs:subClassOf>
20    <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#Servicos"/>
21  </rdfs:subClassOf>
22 </owl:Class>

```

Figura 2. Exemplo de um código em RDF. Fonte: Elaborado pelo autor (2017).

A OWL pode se basear nas especificações do RDF, herdando suas características como a estrutura baseada em triplas e a descrição de recursos com URI, mas, por outro lado, herda a semântica descrita no esquema RDF, com isso é possível transformar uma ontologia representada por meio da linguagem OWL em RDF. A Figura 3 apresenta um exemplo de um código RDF a partir do código OWL visto na Figura 1 e Figura 2.

A Figura 3 apresenta um exemplo da classe Alameda como subclasse do conceito Logradouro descrita por meio dos modelos OWL e RDF, respectivamente.

```

1 Trecho de uma classe representada em OWL
2 <owl:Class rdf:ID="Alameda">
3   <rdfs:subClassOf>
4     <owl:Class rdf:ID="Logradouro"/>
5   </rdfs:subClassOf>
6 </owl:Class>
7
8 Trecho de uma classe representada em RDF
9 <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#Alameda">
10  <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/unnamed.owl#Logradouro"/>
11 </owl:Class>

```

Figura 3. Exemplo de um código em OWL e RDF que descrevem o conceito de Alameda como subclasse do conceito Logradouro. Fonte: Elaborado pelo autor (2017).

Alguns ambientes de desenvolvimento apoiam a documentação de ontologias, como o Protégé 2000 (GENNARI et al., 2002), Ontolândia (FARQUHAR, FIKES; e RICE, 1997), OLEd (BECHHOFFER et al., 2001) e OntoEdit (SURE et al., 2002).

A OWL pode ser vista também como um repositório, otimizando a recuperação de informações e coleções de dados por meio de ontologias escritas em RDF. Algumas soluções podem ser vistas como tecnologias utilizadas para manipulação de ontologias em OWL e RDF. A RacerPro proposta por Calvanese et. al. (2007) e de forma semelhante à RacerPro, a ferramenta QuOnto que é usada para descrever e armazenar ontologias. A QuOnto permite consultar a base das ontologias a partir de restrições existentes na base de conhecimento (CALVANESE et al., 2007).

Para que as tecnologias supracitadas possam ser aplicadas em uma ferramenta que possa tratar os dados de ontologias de OWL para RDF, se faz necessário a construção de uma ferramenta que irá contemplar todo o processo de coleta dos conceitos e anotação semântica destes conceitos, para o contexto deste trabalho será feito a demonstração da camada intermediária da aplicação desenvolvida pelo Grupo de Estudos sobre Gestão de Conhecimento em Bases de Dados Corporativas (GECON) no qual será evidenciado o *framework* Jena, no qual é desenvolvido em Java para construção de aplicações para a Web Semântica e fornece um ambiente de programação para RDF e OWL. Além disso, o *framework* Jena possibilita a persistência em bases de dados relacionais e uma linguagem de consulta, manipulação e criação de ontologias escritas em RDF e OWL (MCBRIDE, 2007).

### **3. Utilização do *framework* Jena para manipulação de Ontologias**

Jena é um *framework* para manipulação de tecnologias recomendadas pela área da Web Semântica por meio da linguagem Java, possibilitando assim a construção de soluções computacionais para manipulação de representações e dados semânticos (APACHE, 2017).

O *framework* fornece um extenso conjunto de bibliotecas Java para ajudar os desenvolvedores a criarem os códigos de aplicações que gerenciam dados gerados em RDF e OWL, de acordo com as recomendações publicadas do W3C. O *framework* Jena inclui um mecanismo de inferência baseado em regras para executar o raciocínio baseado em ontologias OWL e RDF e uma variedade de estratégias de armazenamento para armazenar triplas na memória ou diretamente em disco (APACHE, 2017).

O *framework* Jena foi originalmente desenvolvido por pesquisadores da HP Labs (*Hewlett-Packard*), começando em Bristol, no Reino Unido, no ano de 2000. Jena é um

projeto de código aberto, e tem sido amplamente utilizado em uma ampla variedade de aplicativos e demonstrativos da Web semântica. Em 2009, a HP decidiu reorientar a atividade de desenvolvimento do suporte direto ao desenvolvimento de Jena. Em 2010 o *framework* Jena passou a ser mantido pela equipe da Apache Software Foundation (APACHE, 2017).

### 3.1. Manipulação de Ontologias utilizando o *framework* Jena

Os métodos que serão descritos neste tópico foram usados para a construção do protótipo proposto, no intuito de orientar qual foi a estratégia de uso de cada um deles.

Dentre os principais métodos utilizados pelo *framework* Jena, para manipulação de ontologias, é importante termos que salvar o código escrito em OWL na memória, uma vez salvo, o acesso aos métodos à alguma ontologia utiliza-se o método `createOntologyModel(model)`, de acordo com o exemplo visto abaixo:

```
OntModel OntMod = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM);
```

Figura 4. Exemplo de uso do método `createOntologyModel`. Fonte: Elaborado pelo autor (2017).

Conforme a necessidade deste trabalho foi utilizada no parâmetro o `OntModelSpec.OWL_MEM` (APACHE 2017), como forma de acessar toda a especificação do modelo descrito pela linguagem OWL.

Além das instruções de acesso e armazenamento da descrição de uma ontologia na memória, existem métodos que se torna possível a leitura desta descrição. São várias instruções utilizadas para leitura de ontologias, que manipulam diferenças na fonte do documento (a serem lidas a partir de um URI, a partir das instruções encontradas no documento OWL), a partir de uma URI associada ao documento OWL lido. Resumindo, os métodos podem ser utilizados pelo método `read` conforme exemplo abaixo:

```
OntMod.read(in, "");
```

Figura 5. Exemplo de uso do método `read`. Fonte: Elaborado pelo autor (2017).

Sendo que a variável `OntMod` é uma instância da interface `OntModel` citado acima e a variável `in` é o arquivo recebido, o segundo parâmetro não foi definido pois seria necessário definir um linguagem (APACHE, 2017).

Para que seja possível percorrer as classes de uma ontologia descrita em OWL, o *framework* Jena disponibiliza uma interface com base no padrão de projeto chamado *Iterator*, para que se possa navegar entre todas as classes, permitindo assim a leitura, escrita e exclusão das mesmas. Abaixo segue um exemplo da instanciação da interface `ExtendedIterator` que permite acesso às classes de uma ontologia (APACHE, 2017):

```

OntModel OntMod = gerarModelo();
ExtendedIterator classes = OntMod.listClasses();

while (classes.hasNext()) {
    OntClass ontClass = (OntClass) classes.next();
}

```

Figura 6. Exemplo de uso dos métodos `listClasses`, `hasNext` e `next`, e também da instanciação da interface `ExtendedIterator`. Fonte: Elaborado pelo autor (2017).

Ao instanciar o `ExtendedIterator` é possível acessar as informações de todas as classes por meio do método `listClasses` da interface `OntModel` conforme visto acima, após instanciar a lista de classes pelo método anterior e com o auxílio de uma estrutura de repetição da própria linguagem Java, pode-se percorrer o modelo de uma ontologia usando método `next` e o `hasNext` que é descrito pela interface `ExtendedIterator`.

Para a captura da URI associada à classe da ontologia, é possível utilizar o método `getNsPrefixURI` da interface `OntModel`, entre várias utilidades a este método, se destaca o uso para captura automática da URI, evitando que o usuário ou programador cometa enganos ao passar ela ao sistema, além do fato de simplificar a parametrização do sistema, o uso deste método é descrito no parâmetro abaixo:

```

URI = OntMod.getNsPrefixURI("");

```

Figura 7. Exemplo de uso do método `getNsPrefixURI`. Fonte: Elaborado pelo autor (2017).

Dependendo do contexto que irá se fazer o uso do *framework* Jena, é pertinente saber se uma determinada classe possui subclasses, caso possua, que seja possível listá-las, de uma forma análoga como foi descrito anteriormente podemos fazer um laço para capturar as subclasses de uma determinada classe conforme no exemplo abaixo:

```

for (Iterator i = ontScla.listSubClasses(); i.hasNext();) {
    OntClass c = (OntClass) i.next();
    String VSubClasses = c.getLocalName();
    String comentarioSubClasse = OntMod.getOntClass(URI + VSubClasses).getComment(null);
}

```

Figura 8. Exemplo de uso do método `listSubClasses`. Fonte: Elaborado pelo autor (2017).

Para determinados casos de varredura de um modelo de ontologia, é possível coletar informações de uma classe, tais como: `getLocalName` e o `getComment(String lang)`, ambos da classe `OntModel` APACHE (2017).

```
String comentario = OntMod.getOntClass(URI + classe).getComment(null);
```

Figura 9. Exemplo de uso do método `getComment`. Fonte: Elaborado pelo autor (2017).

Ressaltando o uso do método `getLocalName`, destaca-se sua utilidade para obtenção do *namespace* da classe com o auxílio do método `getOntClass`, sendo um método da interface `OntClass` como estão descritos nos exemplos abaixo:

```
String subClasses = c.getLocalName();
```

Figura 10. Exemplo de uso do método `getLocalName`. Fonte: Elaborado pelo autor (2017).

```
OntClass ontSCla = OntMod.getOntClass(URI + classe);
```

Figura 11. Exemplo de uso do método `getOntClass`. Fonte: Elaborado pelo autor (2017).

De acordo com o exemplo acima caso o desenvolvedor tivesse somente o nome da classe, seria possível recuperar o *namespace* da mesma, e assim poderia ser feito quaisquer operações que a interface `OntClass` oferecer.

Finalmente, para escrever o modelo que carregamos na memória, usamos o método `write` para ser exportado para uma variável APACHE (2017). Abaixo segue um exemplo do uso do método mencionado:

```
OntMod.write(sw, "RDF/XML-ABBREV");
```

Figura 12. Exemplo de uso do método `write`. Fonte: Elaborado pelo autor (2017).

No exemplo acima a variável `sw` receberá o modelo que está gravado na memória, no segundo parâmetro é a definição da forma como a ontologia será gravada.

#### **4. Protótipo de uma aplicação para anotação semântica utilizando o *framework* Jena**

Como citado anteriormente, o protótipo utiliza o *framework* Jena, então a linguagem escolhida para a construção do protótipo foi o Java Web, pois no back-end usamos os

métodos do framework Jena e na parte Web (View) temos uma representação da ontologia que será gravada, o protótipo tem como objetivo intermediar um processo que produz um arquivo de ontologia em OWL para outro processo que armazena as estruturas contidas no RDF que foi emitido pelo protótipo proposto a este trabalho.

A aplicação apresentada possui funcionalidades, tais como: a) a leitura de arquivos de ontologias descritas em OWL com o auxílio de um formulário da web; b) a visualização da hierarquia disposta em forma de árvore dos conceitos existentes na ontologia selecionada na primeira funcionalidade; e c) a geração de um documento que representa os significados presentes na ontologia selecionada.

**Visualizador de Hierarquia Ontológica**

Insira o arquivo de Ontologia: 1 Selecionar arquivo... Nenhum arquivo selecionado.

Insira a URI: 2

(Só será necessário caso dê problemas na Hora de exibição, pelo fato que existem tipos de arquivos de ontologia escrito de formas diferentes.)

3 Visualizar Hierarquia e Anotar Semanticamente

Figura 13. Imagem do formulário para leitura de uma ontologia. Fonte: Elaborado pelo autor (2017).

Primeiramente a funcionalidade é representada por um formulário conforme apresentado na Figura 13. Neste formulário a escolha da ontologia é mandatória (opção 1), por meio da seleção de um arquivo obrigatoriamente com extensão OWL. Neste formulário, a inserção da URI é opcional (opção 2), e finalmente é pressionado o botão “Visualizar Hierarquia e Anotar Semanticamente” (opção 3).

Uma vez feito o upload do arquivo de ontologia, será apresentado a manipulação de uma ontologia para conceituar o conjunto de conceitos e suas relações dentro de um domínio de recursos computacionais. A ontologia foi desenvolvida pelo Grupo de Estudos em Gestão de Conhecimento em Bases de Dados Corporativas (GECON), como forma de aplicar a geração de dados semânticos no formato RDF. Esta ontologia descreve basicamente os conceitos das “coisas” que fazem parte de um ambiente de tecnologia da

informação e foi construída a partir da ferramenta Protégé (PROTÉGÉ, 2017). A Figura 14 apresenta os conceitos existentes na ontologia de recursos computacionais e suas relações.

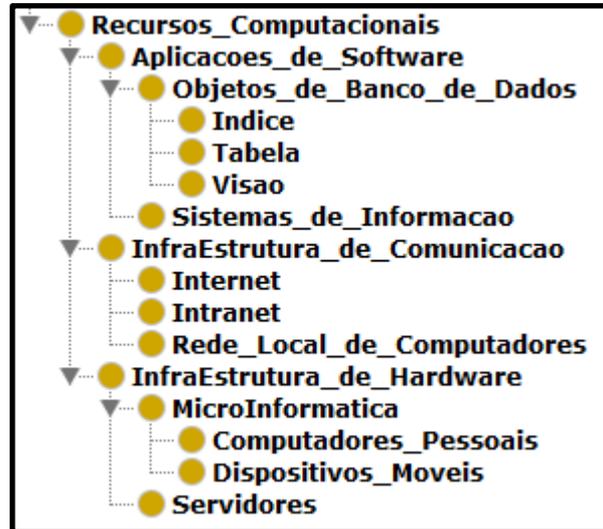


Figura 14. Ontologia de Recursos Computacionais. Fonte: PROTÉGÉ (2017).

Uma vez a ontologia armazenada em memória pelas rotinas desenvolvidas no protótipo, a mesma pode ser apresentada pela funcionalidade de visualização de ontologias, apresentando o estado atual das classes e suas interações umas com as outras e além de mostrar os comentários de cada classe indicada (Figura 15).



Figura 15. Imagem da leitura da ontologia de recursos computacionais realizada pelo protótipo. Fonte: Elaborado pelo autor (2017).

Finalmente, conforme já apresentado o protótipo faz a definição de elementos definidos no modelo OWL gerando um arquivo RDF. O arquivo RDF é gravado em uma

pasta pré-determinada. Um exemplo da geração do arquivo RDF é apresentado na Figura 16, exibindo o formato em RDF gerado a partir de uma ontologia.

```
1 <rdf:RDF
2   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3   xmlns:owl="http://www.w3.org/2002/07/owl#"
4   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
5   xmlns="http://www.owl-ontologies.com/unnamed.owl#"
6   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
7 <owl:Ontology rdf:about="http://www.owl-ontologies.com/unnamed.owl"/>
8 <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#Avenida">
9   <rdfs:subClassOf>
10    <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#Logradouro"/>
11  </rdfs:subClassOf>
12 </owl:Class>
13 <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#Hospital">
14   <rdfs:subClassOf>
15    <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#Edificacao_de_Referencia"/>
16  </rdfs:subClassOf>
17 </owl:Class>
18 <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#Shopping">
19   <rdfs:subClassOf>
20    <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#Servicos"/>
21  </rdfs:subClassOf>
22 </owl:Class>
23 <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#Complemento_Endereco">
24   <rdfs:subClassOf>
25    <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#Endereco"/>
26  </rdfs:subClassOf>
27 </owl:Class>
28 <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#Linha">
29   <rdfs:subClassOf>
30    <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#Geometria"/>
31  </rdfs:subClassOf>
32 <rdfs:comment>Par ordenado (x,y) de coordenadas espaciais.</rdfs:comment>
33 </owl:Class>
34 <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#SubDistrito">
35   <rdfs:subClassOf>
36    <owl:Class rdf:about="http://www.owl-ontologies.com/unnamed.owl#Unidade_Administrativa"/>
37  </rdfs:subClassOf>
38 <rdfs:comment>Unidade administrativa de um município</rdfs:comment>
```

Figura 16. Trecho do arquivo RDF gerado a partir da leitura da ontologia de recursos computacionais. Fonte: Elaborado pelo autor (2017).

## 5. Considerações Finais

Gerar conhecimento a partir de estruturas já existentes é uma tarefa que exige muito esforço e tempo para que seja desenvolvida uma solução em forma de ferramenta, por isto este protótipo foi desenvolvido para que ele possa flexibilizar a estrutura de arquivos OWL e passá-la para RDF para que possamos futuramente com o avanço do grupo de estudos

aprimorar as ligações entre os recursos já existentes e que possamos adicionar novos tipos de ligação, no caso da interface ela foi desenvolvida para demonstração dos processos que ela faz, pois futuramente a parte visual poderá ser eliminada ou aprimorada para uma compreensão macro do conceito que a ontologia carrega.

Apesar das limitações que o protótipo possui atualmente, o trabalho teve uma contribuição no quesito de apresentar o *framework* JENA para usuários que nunca tiveram contato com esta ferramenta e também puderam ver um exemplo prático do uso desta biblioteca.

## Referências

- APACHE JENA. **What is Jena?** Disponível em: [https://jena.apache.org/about\\_jena/about.html](https://jena.apache.org/about_jena/about.html). Acesso em 20 de Setembro 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.
- ARANTES, L. O.; FALBO, R. A.; GUIZZARDI, G. **Evolving a Software Configuration Ontology. In: Proceedings of the Second Brazilian Workshop on Ontologies and Metamodels for Software and Data Engineering (WOMSDE'07), 22nd Brazilian Symposium on Databases (SBBD)/21st Brazilian Symposium on Software Engineering (SBES),** João Pessoa, Brazil, 2007.
- BECHHOFER, S.; HORROCKS, I.; GOBLE, C.; STEVENS, R. **OilEd: a Reason-able Ontology Editor for the Semantic Web.** DL2001, 14th International Workshop on Description Logics, Stanford, USA, August 2001.
- BECKETT, D.; MCBRIDE, B. **RDF/XML Syntax Specification (Revised),** 2004. Overview. Disponível em <http://www.w3.org/TR/rdf-syntax-grammar/> . Acesso em : 20 de julho de 2017.
- BERNERS-LEE, Tim. **Chapter 30: “Realizing the potential of the Web”, Web-Weaving by Peter Lloyd and Paula Boyle,** Butterworth-Heinemann, United Kingdom, 1998.
- BERNERS-LEE, Tim et al. **forward: Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential by Dieter Fensel (Editor), Wolfgang Wahlster, Henry Lieberman, James Hendler,** MIT Press, 2002.
- BERTOLLO, G.; RUY, F. B. ; MIAN, P. G.; PEZZIN, J.; SCHWAMBACK, M.; NATALI, A. C. C.; FALBO, R. A . **ODE - Um Ambiente de Desenvolvimento de Software baseado em Ontologias. In: Simpósio Brasileiro de Engenharia de**

- Software** - Sessão de Ferramentas, Gramado, 2002. Anais do XVI Simpósio Brasileiro de Engenharia de Software, 2002, p. 438-443.
- BOOTSTRAP, Bootstrap · The most popular HTML, CSS, and JS library in the world.** Disponível em: <https://getbootstrap.com/>. Acesso em 30 de Novembro de 2017.
- BRITISH COLUMBIA. DATABASE DESIGN (PHYSICAL) MODELING STANDARDS AND GUIDELINES** (2017). Disponível em [https://www2.gov.bc.ca/assets/gov/british-columbians-our-governments/services-policies-for-government/information-technology/standards/economy-sector/database\\_design\\_physical\\_modelling\\_standards\\_and\\_guidelines.pdf](https://www2.gov.bc.ca/assets/gov/british-columbians-our-governments/services-policies-for-government/information-technology/standards/economy-sector/database_design_physical_modelling_standards_and_guidelines.pdf). Acesso em 07 de Novembro de 2017.
- CALVANESE, Diego et al. **Software Tools for Ontology Design and Maintenance.** Deliverable TONES-D15, 2007.
- CASANOVA, M. A. ; BRAUNER, D. F. ; CÂMARA, G. ; LIMA Jr., P. O. . **Integração e interoperabilidade entre fontes de dados geográficos.** In: Marco Antônio Casanova; Gilberto Câmara; Clodoveu Augusto Davis Junior; Lúbia Vinhas; Gilberto Ribeiro de Queiroz. (Org.). Bancos de dados geográficos. Curitiba: MundoGEO, 2005, p. 317-352.
- DELFMANN, Patrick; HERWIG, Sebastian; LIS, Lukasz. Unified enterprise knowledge representation with conceptual models-Capturing corporate language in naming conventions. ICIS 2009 proceedings, p. 45, 2009.
- FARQUHAR, A.; FIKES, R.; RICE, J. Tools for Assembling Modular Ontologies in Ontolingua. Knowledge Systems, AI Laboratory, 1997. Disponível em : <http://www.cs.umbc.edu/771/papers/KSL-97-03.pdf> . Acesso em : 05 de Outubro de 2017.
- FELICÍSSIMO, C. H. ; LEITE, J. C. P. ; BREITMAN, K. K. ; SILVA, L. F. **Geração de Ontologias subsidiada pela Engenharia de Requisitos.** In: VI Workshop em Engenharia de Requisitos (WER-03), São Paulo, 2003.
- GENESERETH, M. R.; FIKES, R. E. Knowledge Interchange Format, version 3.0 Reference Manual, Logic Group Report Logic-92-1 Computer Science Department, Stanford University, Stanford, California, USA, 1994. Disponível em <http://logic.stanford.edu/kif/Hypertext/kif-manual.html>. Acesso em Agosto de 2017.
- GENNARI, J.; MUSEN, M. A.; FERGERSON, R. W.; GROSSO, W. E.; CRUBEZY, M.; ERIKSSON, H.; NOY, N. F. **The Evolution of Protégé: An Environment for Knowledge-Based Systems Development.** 2002. Disponível em: <http://portal.acm.org/citation.cfm?id=766321>. Acesso em: 20 de Agosto 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.
- GRUBER, T. R. **A Translation Approach to Portable Ontology Specifications**. 1993. Disponível em <[http://www-ksl.stanford.edu/KSL\\_Abstracts/KSL-92-71.html](http://www-ksl.stanford.edu/KSL_Abstracts/KSL-92-71.html)>. Acesso em 13 de Setembro de 2017.
- MCBRIDE, B. An Introduction to RDF and the Jena RDF API, 2007. Disponível on-line em: [http://jena.sourceforge.net/tutorial/RDF\\_API/index.html](http://jena.sourceforge.net/tutorial/RDF_API/index.html). Acesso em : 19 de Setembro de 2017.
- MCGUINNESS, D. L.; VAN HARMELEN, F.. OWL Web Ontology Language – Overview, 2004. Disponível em <http://www.w3.org/TR/owl-features/> . Acesso em : 20 de Agosto de 2017.
- MIZOGUCHI, R.; VANWELKENHUYSEN, J.; IKEDA, M. Task ontology for reuse of problem solving knowledge. In: Proceedings OF ECAI'94 TOWARDS VERY LARGE KNOWLEDGE BASES, 1995, Amsterdam: IOS Press, 1995. p. 46-59.
- PROTEGÉ. **A free, open-source ontology editor and framework for building intelligent systems**. Disponível em: <https://protege.stanford.edu/>. Acesso em 09 de novembro de 2017.
- SILVA, Evaldo De Oliveira; LISBOA FILHO, Jugurta; GONÇALVES, Gabriel. **Improving Analysis Patterns in the Geographic Domain Using Ontological Meta-Properties**. In: ICEIS (3-1). 2008. p. 256-261.
- SURE, Y.; ERDMANN, M.; ANGELE, J.; STAAB, S.; STUDER, R.; WENKE, D. Ontoedit: Collaborative Ontology Development for the Semantic Web. In: Proceedings of the 1st International Semantic Web Conference, 2002. vol. 2342. p. 348-363.
- W3C. **Semantic Web**. Disponível em: <https://www.w3.org/standards/semanticweb/>. Acesso em 09 de novembro de 2017.