



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

## Modelagem computacional de conhecimento e apoio ao ensino do questionamento no quizWiki

Victor Nogueira Guimarães

Monografia apresentada como requisito parcial  
para conclusão do Bacharelado em Ciência da Computação

Orientadora  
Prof.<sup>a</sup> Dr.<sup>a</sup> Germana Menezes da Nóbrega

Brasília  
2015

Universidade de Brasília — UnB  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação  
Bacharelado em Ciência da Computação

Coordenador: Prof. Dr. Homero Luiz Piccolo

Banca examinadora composta por:

Prof.<sup>a</sup> Dr.<sup>a</sup> Germana Menezes da Nóbrega (Orientadora) — CIC/UnB  
Prof.<sup>a</sup> Dr.<sup>a</sup> Fernanda Lima — CIC/UnB  
Prof.<sup>a</sup> Dr.<sup>a</sup> Maria Emilia Machado Telles Walter — CIC/UnB

### **CIP — Catalogação Internacional na Publicação**

Guimarães, Victor Nogueira.

Modelagem computacional de conhecimento e apoio ao ensino do questionamento no quizWiki / Victor Nogueira Guimarães. Brasília : UnB, 2015.

245 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2015.

1. quizWiki, 2. ontologia, 3. OWL, 4. Protégé, 5. perguntas, 6. questionamento, 7. MediaWiki, 8. Semantic MediaWiki, 9. aprendizado.

CDU 004.4

Endereço: Universidade de Brasília  
Campus Universitário Darcy Ribeiro — Asa Norte  
CEP 70910-900  
Brasília-DF — Brasil



# Dedicatória

Dedico este trabalho a Deus.

# Agradecimentos

Agradeço a Deus, pois não foi fácil chegar até aqui, a meus pais e irmãos, que me ajudaram em todos esses anos de graduação.

Agradeço também a professora Germana, que gentilmente me aceitou como orientando e direcionou o trabalho.

Não poderia deixar de agradecer, a todos as pessoas com quem trabalhei, na condição de estagiário, pois meu aprendizado, tanto como profissional quanto como pessoa, foi extremamente valioso.

Agradeço também a todos os colegas que ingressaram comigo na universidade, mesmo que mais da metade tenha desistido, e uma boa parte ainda esteja por formar, tenho certeza que cada um me ensinou alguma coisa pessoal ou academicamente, que me ajudou a efetuar este trabalho.

Como foi expressamente solicitado no site oficial da ferramenta *Protégé*, pontuo nos agradecimentos, que: este trabalho foi realizado com o recurso *Protégé*, que é apoiado pela concessão *GM10331601 do National Institute of General Medical Sciences of the United States National Institutes of Health*.

# Resumo

Com o desenvolvimento tecnológico, nos deparamos com a crescente imersão da sociedade e das pessoas dentro do mundo virtual, através de vários dispositivos, como computadores pessoais, celulares, *tablets* e andando em direção a tecnologias vestíveis (*wearable*). A presença de tanta tecnologia, infelizmente, não tem sido acompanhada dentro das salas de aula, pelo menos não na mesma escala. Para isso é necessário, cada vez mais, o desenvolvimento de conhecimento na área que permeia a Informática na Educação. Neste sentido situamos o nosso trabalho, baseado no projeto *quizWiki* que auxilia no aprendizado através do ensino do questionamento, nota-se a dificuldade dos alunos para formular perguntas. Esta monografia versa sobre a criação de uma ontologia sobre perguntas, sua inserção no ambiente wiki e a sua compatibilização com o projeto *quizWiki*, a fim de auxiliar os alunos a aprenderem, através do processo de elaboração e avaliação de perguntas.

**Palavras-chave:** quizWiki, ontologia, OWL, Protégé, perguntas, questionamento, MediaWiki, Semantic MediaWiki, aprendizado.

# Abstract

With technological development, we face the increasing immersion of society and people, inside the virtual world, through various devices such as personal computers, mobile phones, tablets and walking toward wearable technologies. The presence of so much technology, unfortunately, has not been followed in the classroom, at least not on the same scale. It requires, ever more, the development of knowledge in the field that permeates the information technology in education. In this sense we situate our work, based on quizWiki project that aids learning through questioning learning, we notice a difficulty of students to formulate questions.

This work deals with the creation of an ontology of questions, the inclusion in the wiki environment and the compatibilization with the *quizWiki* project, in order to help students, to learn through the process of elaboration and evaluation of questions.

**Keywords:** quizWiki, ontology, OWL, Protégé, questions, MediaWiki, Semantic MediaWiki, learning.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contextualização . . . . .	1
1.1.1	Problema . . . . .	2
1.1.2	Suposição . . . . .	2
1.1.3	Objetivos . . . . .	3
1.2	Organização do documento . . . . .	4
<b>2</b>	<b>Trabalhos relacionados (revisão de literatura) : suporte computacional ao ensino do questionamento</b>	<b>5</b>
2.1	O quizWiki . . . . .	5
2.2	Outros trabalhos . . . . .	7
<b>3</b>	<b>Referencial Teórico</b>	<b>8</b>
3.1	Questionamento . . . . .	8
3.1.1	O que é uma pergunta . . . . .	8
3.1.2	Questionamento dentro do contexto educacional . . . . .	8
3.1.3	O questionamento 1 para 1 . . . . .	9
3.1.4	Ambientes Virtuais de Aprendizagem . . . . .	9
3.1.5	Uso neste contexto . . . . .	11
3.2	Ontologia . . . . .	11
3.2.1	Ontologia para a Filosofia . . . . .	11
3.2.2	Ontologia para a Ciência da Computação . . . . .	11
3.2.3	OWL . . . . .	14
3.2.4	Ferramentas . . . . .	15
3.3	O <i>quizWiki</i> . . . . .	19
3.3.1	O ambiente wiki no contexto pedagógico . . . . .	19
3.3.2	<i>QuizWiki</i> : um ambiente para incorporar a elaboração discente de perguntas no processo de aprendizagem . . . . .	20
3.3.3	O ambiente do <i>quizWiki</i> . . . . .	20
3.3.4	MediaWiki . . . . .	20
3.4	<i>Semantic MediaWiki</i> . . . . .	23
3.4.1	Conceitos Básicos . . . . .	23
3.4.2	Vantagens do <i>Semantic MediaWiki</i> . . . . .	24
<b>4</b>	<b>Proposta</b>	<b>26</b>
4.1	Metodologia . . . . .	26
4.1.1	Modelo de desenvolvimento . . . . .	26



4.1.2	Construção da Ontologia . . . . .	26
4.1.3	Criação do modelo OWL . . . . .	29
4.1.4	Verificação do modelo OWL . . . . .	29
4.1.5	Verificação da ontologia criada . . . . .	30
4.1.6	Correções . . . . .	30
4.2	Modelo Ontológico . . . . .	30
4.2.1	Passo 1, determinar um domínio e escopo . . . . .	30
4.2.2	Passo 2, Considerar o reuso . . . . .	32
4.2.3	Passo 3, Enumerar todos os termos relevantes . . . . .	32
4.2.4	Passo 4: Definir as classes e a sua hierarquia . . . . .	32
4.2.5	Passo 5, Definir as propriedades das classes . . . . .	36
4.2.6	Passo 6, Definir as propriedades das propriedades . . . . .	39
4.2.7	Passo 7, Criar instâncias . . . . .	40
4.2.8	Ontologia resultante no Protégé . . . . .	40
4.2.9	Revisão da Ontologia e do modelo OWL . . . . .	40
4.2.10	Inserção de perguntas na ontologia e Inferências . . . . .	40
4.3	Integração ao <i>quizWiki</i> . . . . .	41
4.4	Cenários hipotéticos de utilização . . . . .	47
<b>5</b>	<b>Conclusão</b> . . . . .	<b>50</b>
5.1	Considerações finais . . . . .	50
5.2	Trabalhos futuros . . . . .	51
	<b>Referências</b> . . . . .	<b>52</b>
	<b>A Modelo OWL criado</b> . . . . .	<b>55</b>
	<b>B Instalação e configuração do ambiente wiki</b> . . . . .	<b>84</b>
	<b>C Programa Java que transforma de OWL para .txt</b> . . . . .	<b>88</b>

# Lista de Figuras

1.1	Exemplo do funcionamento do <i>quizWiki</i> . . . . .	2
2.1	Representação esquemática de um evento no quizWiki . . . . .	6
2.2	Aparência atual de uma página ordinária do quizWiki . . . . .	6
2.3	Aparência da aba quiz . . . . .	7
3.1	Espectro de ontologias, do informal e menos expressivo ao formal e mais expressivo [31] . . . . .	14
3.2	A hierarquia de ontologias: do mais geral (e reusável) na base, ao mais específico (e menos reusável) no topo [31] . . . . .	15
3.3	Exemplo de ontologia (do tipo <i>é um</i> ) sobre trabalhos de arte [31] . . . . .	16
3.4	Exemplo de ontologia sobre Pizzas . . . . .	17
4.1	Modelo de desenvolvimento em espiral . . . . .	26
4.2	Desenvolvimento de ontologias de Noy and McGuinness (Figura retirada de [31]) . . . . .	28
4.3	Classes organizadas em níveis . . . . .	35
4.4	Organização dos <i>Object Properties</i> . . . . .	36
4.5	Exemplo de pergunta classificada como boa . . . . .	43
4.6	Exemplo de pergunta classificada como ruim . . . . .	43
4.7	Versões instaladas localmente . . . . .	44
4.8	Processo de importação da ontologia . . . . .	45
4.9	Algumas das novas páginas criadas na <i>wiki</i> . . . . .	45
4.10	Anotações semânticas como são vistas pelo usuário . . . . .	46
4.11	Visualização do código das anotações semânticas . . . . .	46
4.12	Categorias criadas a partir das anotações semânticas . . . . .	46
4.13	Propriedades criadas a partir das anotações semânticas . . . . .	47
4.14	Caso de uso . . . . .	48
4.15	Caso de uso . . . . .	48

# Lista de Tabelas

1.1	Situação dos projetos . . . . .	2
4.1	<i>Brainstorm</i> de palavras importantes relacionadas ao questionamento. . . . .	33
4.2	Relações importantes nas classes . . . . .	34
4.3	Formação dos <i>Object Properties</i> . . . . .	37
4.4	Classificação das perguntas inferidas . . . . .	41
4.5	Classificação das perguntas desenvolvidas pelos alunos . . . . .	42

# Capítulo 1

## Introdução

Este trabalho mostra uma proposta de ontologia de "perguntas", cuja construção é mostrada no decorrer do texto, mostra também uma integração da ontologia com o projeto *quizWiki*, na avaliação das perguntas criadas pelos alunos e no auxílio do ensino do questionamento.

### 1.1 Contextualização

Dada a informatização crescente da sociedade, é clara a influência cada vez maior das novas tecnologias na vida dos alunos, professores, instituições de ensino e demais atores que formam toda a cadeia educacional.

Nesse contexto, vemos o desenvolvimento de técnicas, conceitos e ferramentas, que visam auxiliar das mais diversas formas e nos mais amplos aspectos o processo educacional, tanto para o aluno quanto para os demais envolvidos. Nesse âmbito localizamos o projeto *quizWiki*, como sendo uma ferramenta de auxílio ao aprendizado através de questionamentos, envolvendo diretamente os docentes e discentes.

Ressaltamos aqui o papel central do questionamento durante o processo cognitivo de aprendizagem, onde segundo Mackay [19] os alunos se valem da confecção de perguntas, para:

- Organizar suas ideias;
- Testar seus conhecimentos;
- Validar suas ideias;
- Mesclar os novos conteúdos aprendidos com suas bases de conhecimentos pré-existentes.

Podemos caracterizar o projeto *quizWiki*, em linhas gerais, como sendo um projeto de comunidade baseada na plataforma wiki, onde os alunos possuem a possibilidade, através de uma extensão wiki, de formular perguntas sobre determinados conteúdos e assim (devidamente acompanhados) vão desenvolvendo suas habilidades e fixando os conteúdos estudados.

Mais especificamente, um aluno utilizando o *quizWiki*, de tempos em tempos recebe a tarefa de confeccionar algumas perguntas, após fazer isso, tais perguntas são avaliadas

pelos seus colegas e pelo professor, por fim o aluno recebe um *feedback* (na forma de nota de reputação).

O *quizWiki* apresenta uma série de avanços no ensino do questionamento, no entanto apresenta também algumas lacunas, neste trabalho apresentamos soluções para algumas delas, conforme a tabela 1.1, conforme a Figura 1.1.

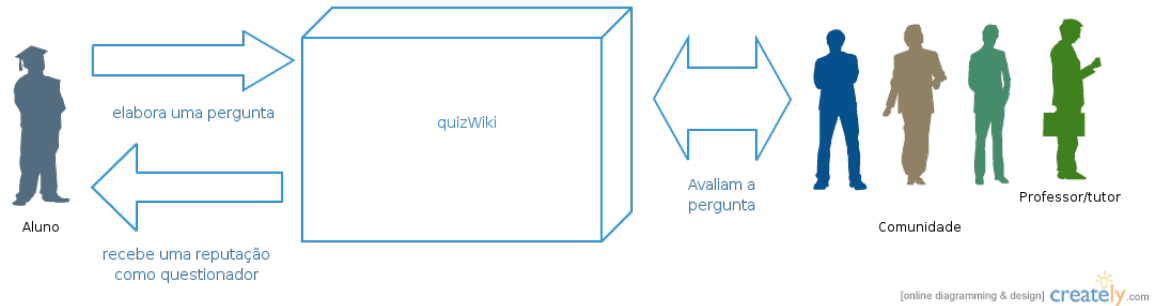


Figura 1.1: Exemplo do funcionamento do *quizWiki*

Tabela 1.1: Situação dos projetos

	<i>quizWiki</i>	Este trabalho	<i>quizWiki2</i>
objetivos	No projeto <i>quizWiki</i> buscamos prover suporte computacional para a aprendizagem significativa crítica. Foi elaborada uma proposta conceitual de uma ferramenta a partir do princípio da interação social e do questionamento, tendo o estudante como (i) elaborador de questões e como (ii) avaliador das questões elaboradas por seus pares.	Apresentar um modelo computacional que seja capaz de: modelar o domínio do questionamento, ser adicionado ao ambiente <i>quizWiki</i> de forma a representar uma fonte interna de conhecimento, que possa auxiliar os usuários a aprender e melhorar suas habilidades de estudante.	Prover uma modelagem computacional de conhecimento que permita ao <i>quizWiki</i> contribuir mais ativamente no ensino do questionamento.
lacuna	Não possui conhecimento embutido, que possa ajudar os alunos de alguma forma.	Peca em não atender temas mais específicos dentro de cada área do conhecimento	Fica restrito ao projeto <i>quizWiki</i> .

### 1.1.1 Problema

Encontramos então o problema que pode ser enunciado como: construir um modelo computacional que classifique as perguntas como boas ou ruins, que possa ser embutido em um software como o *quizWiki*?

### 1.1.2 Suposição

Baseados nos problemas citados, acreditamos que poderia ser construída uma ontologia sobre perguntas (ou mesmo uma taxonomia), que pudesse organizar e classificar as

mesmas. Acreditamos também que tal ontologia pode ser adicionada ao projeto *quizWiki* original e assim possa, na forma de uma melhoria, auxiliar os estudantes tanto a conhecer uma boa pergunta, quanto a desenvolver habilidades que resultem em uma capacidade mais apurada de desenvolver perguntas.

### 1.1.3 Objetivos

#### Objetivo geral

Construir uma ontologia sobre perguntas e integra-la ao projeto *quizWiki*.

#### Justificativa (motivação)

Notou-se durante a utilização do *quizWiki*, que muitos alunos, após serem avaliados como um elaborador de perguntas, não sabem quais são os critérios utilizados para classificar as perguntas, também desconhecem quais são os critérios de uma boa pergunta.

Notou-se também que os alunos após terem as suas perguntas classificadas de maneira negativa, não conseguiam saber a causa e tinham dificuldade em desenvolver a habilidade de construir boas perguntas.

#### Objetivos específicos

- Construir uma ontologia sobre perguntas;
- Testar e validar tal ontologia;
- Transcrever a ontologia criada para uma linguagem web-semântica (OWL);
- Usar a extensão *Semantic MediaWiki* para integrar a ontologia ao *quizWiki*.

#### Resultados Esperados

Com este projeto, pretende-se ao fim do mesmo produzir os seguintes resultados:

1. Modelo (ontologia) de parte do domínio "questionamento";
2. Inserção da ontologia no ambiente Media Wiki;
3. Cenário ilustrativo de utilização do ambiente resultante.

#### Importância

Tais resultados se mostram como sendo muito importantes, pois:

- Além das ontologias serem por si só, referências em seus domínios, pretendemos disponibilizá-la em repositório global, a fim de ser re-utilizada em outros projetos.
- Com a inserção da ontologia no Media Wiki, pode-se inseri-la no software *quizWiki*, ele por sua vez, poderá se mostrar como uma ferramenta mais completa e atrativa, fornecendo aos alunos uma qualidade e profundidade maior em seu aprendizado.

- O cenário ilustrativo torna-se essencial, pois o mesmo é relevante para o desenvolvimento do software, uma vez que um ambiente de testes deste porte pode revelar aos desenvolvedores uma vasta gama de situações que dificilmente seriam enfrentadas ou mesmo idealizadas em um ambiente teórico.

## 1.2 Organização do documento

Este documento está organizado da seguinte forma:

O Capítulo 2 apresenta alguns trabalhos relacionados a educação utilizando o questionamento.

O Capítulo 3 levanta todo o referencial teórico acerca do tema, desde o levantamento acerca do questionamento, passando pelos conceitos relacionados a ontologias, pelas descrições dos ambientes de implantação até as ferramentas, que serão utilizadas tanto para desenvolvimento quanto para implantação dos objetivos.

No Capítulo 4 será descrita nossa proposta, incluindo os procedimentos que utilizamos e culminando com a ontologia criada e a sua implantação no MediaWiki.

No Capítulo 5 temos uma breve conclusão do trabalho, mostrando os objetivos alcançados, as dificuldades enfrentadas, assim como indicações para trabalhos futuros.

## Capítulo 2

# Trabalhos relacionados (revisão de literatura) : suporte computacional ao ensino do questionamento

A elaboração de questões, como veremos mais profundamente na seção 3.1, é uma faceta fundamental no processo de aprendizagem de um aluno. Podemos delinear, dentro da literatura, alguns trabalhos que possuem grande ligação com o trabalho corrente (como o caso do projeto quizWiki) e outros que se relacionam de forma menos intensa, tendo em comum somente a parte relacionada ao suporte computacional relacionado.

### 2.1 O quizWiki

Quando verificamos o tema do trabalho corrente, podemos claramente relacioná-lo ao projeto *quizWiki* [7], que é por sua natureza, uma ferramenta de suporte ao ensino do questionamento.

O *quizWiki* é uma adaptação baseada no *MediaWiki*, através das possibilidades abertas pelo licenciamento do mesmo (conforme será discutido em 3.3.4). Ele é uma plataforma colaborativa, um ambiente computacional de apoio à elaboração de questões por parte dos estudantes.

O *quizWiki* apresenta ao aluno uma plataforma de discussões, a qual o aluno já está familiarizado, visto que utiliza a aparência das wikis mais comuns (como a *Wikipedia* por exemplo e conforme a figura 2.2). No *quizWiki* além de interagir, de forma simplista, podemos dizer que: o aluno quando solicitado, posta na aba quiz, perguntas relacionadas ao tema que é objeto de estudo (figura 2.3), ao fazê-lo ele indica um investimento que deseja fazer em sua própria pergunta (investimento este que deve levar em conta a sua avaliação em relação a qualidade de sua própria pergunta), após um determinado período as perguntas serão avaliadas, tanto pelo professor quanto pela comunidade, e esta avaliação resultará em uma nota, que influenciará na "reputação de questionador" que cada aluno possui perante a comunidade (conforme ilustrado na figura 2.1). Em cima disto, fica a critério do professor a utilização das notas auferidas a cada aluno, dentro do contexto letivo em que se está inserido.

Utilizando este expediente, o *quizWiki* incentiva e induz o aluno a apurar as suas habilidades de questionador, fazendo isso tanto no momento em que o aluno desenvolve



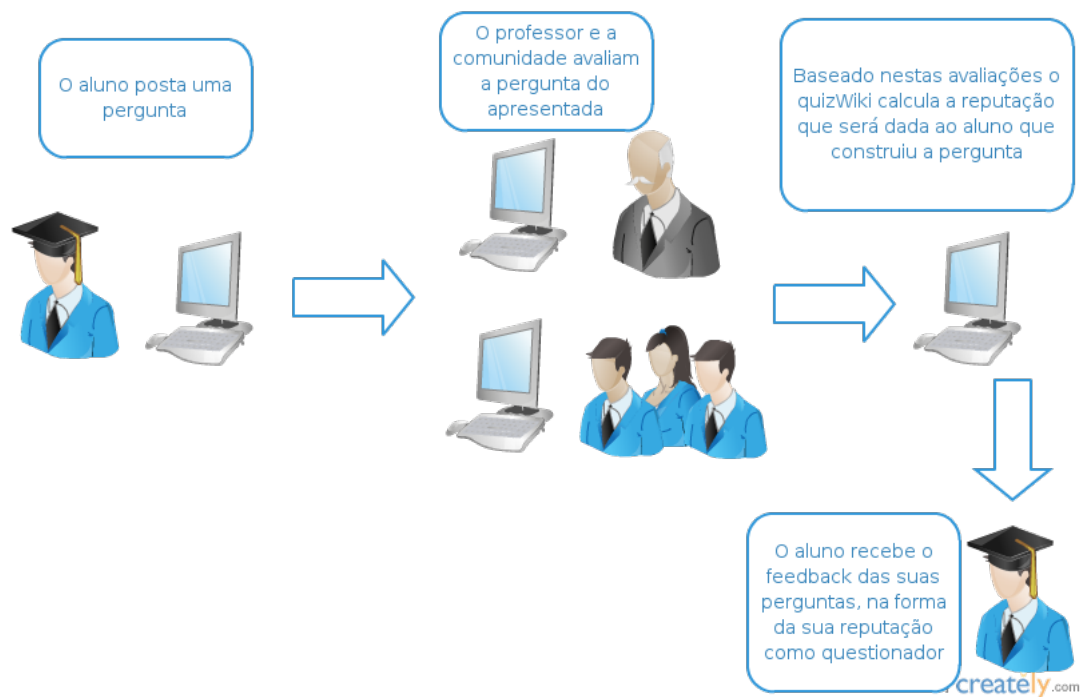


Figura 2.1: Representação esquemática de um evento no quizWiki



Figura 2.2: Aparência atual de uma página ordinária do quizWiki

as questões, quando no momento em que ele dá notas as questões dos seus pares (este processo todo será mais detalhadamente mostrado posteriormente em 3.3).

<a href="#">page</a>	<a href="#">discussion</a>	<a href="#">edit</a>	<a href="#">history</a>	<a href="#">move</a>	<a href="#">watch</a>	<a href="#">quiz</a>	<a href="#">quiz config</a>
----------------------	----------------------------	----------------------	-------------------------	----------------------	-----------------------	----------------------	-----------------------------

## Quiz sobre Árvores

---

Perguntas já postadas pela comunidade	
<a href="#">Ordenar por investimento</a>	
Investimento	Enunciado
20	Como seria o uso de uma árvore para decodificação de código Morse?
20	Qual ganho se tem em implementar uma calculadora usando uma árvore binária em vez de usar uma pilha?
0	Fazendo uma árvore binária para o código morse e usando apenas os símbolos mais básicos, aparecem nós vazios, mas com sub-árvores à direita ou à esquerda. Como implementar a árvore de forma que esses nós vazios desapareçam, mas sem tirar a dinâmica de ponto->direita; traço->esquerda?
0	Para o balanceamento estático (vetor) de uma árvore (slide 32 de árvores) com um número par de elementos, devemos necessariamente usar o termo da direita entre os dois elementos centrais para ser o nó ou simplesmente podemos escolher entre o direito e o esquerdo?
20	Existe alguma aplicação em que realizar travessia na árvore por nível seja a melhor opção?
15	Se quisermos remover um nó com subárvores relativamente grandes, é mais fácil fazer remoção por fusão ou por cópia?

Figura 2.3: Aparência da aba quiz

## 2.2 Outros trabalhos

De forma menos correlata, podemos destacar outros trabalhos, que não possuem um enfoque na questão de ensinar o aluno a produzir questionamentos, mas auxiliam no aprendizado utilizando suportes computacionais e colaborativos. Como o CoWeb (Collaborative Website) [16] que permite a criação e modificação de páginas da web através de um navegador Web, sem qualquer software especializado no lado do cliente.

# Capítulo 3

## Referencial Teórico

### 3.1 Questionamento

#### 3.1.1 O que é uma pergunta

De forma simplificada, uma pergunta é o resultado do ato de perguntar, uma frase interrogativa, uma inquirição a respeito de algo. Na língua portuguesa, uma pergunta (ou questionamento) é uma frase que apresenta uma estrutura sintática própria, com um sinal delimitador (o sinal de interrogação) e uma entonação ascendente.

Se colocada dentro do contexto linguístico, de acordo com Souza-Galvão [8] o par pergunta/resposta é um dos organizadores da interação dentro de uma linguagem qualquer, colocando os interlocutores em contato.

#### 3.1.2 Questionamento dentro do contexto educacional

Segundo Barreiro [2] ato de conhecer, através do questionamento, além de alimentar a curiosidade científica, estimula através da criatividade a inovação, a qualidade formal. Estimula através da criticidade o diálogo inteligente, depara o inquisidor com as suas dificuldades e dá uma ideia bem mais clara quanto as suas limitações.

Segundo Ciardiello [6] "aprender a questionar é aprender a tornar-se literato", outros autores também consideram que o processo mental associado à formulação de uma pergunta ou questão, estimula o desenvolvimento e estruturação do raciocínio crítico e do pensamento criativo. De acordo com Barreiro [2] "não se pode impor verdades ao sujeito questionador. Apenas se pode propiciar a curiosidade para que este sujeito reconstrua o trajeto desta "verdade", tornando-a sua. Fica claro que os sujeitos entrevistados acreditam ter o processo de crescimento e de novas construções, em seu alicerce, o questionamento reconstrutivo".

O ato de questionar implica na absorção do conteúdo de alguma forma e posterior interação pessoal com o mesmo, onde o aluno utiliza-se de todos os conhecimentos pré-adquiridos, para confrontar e adaptar o novo conhecimento a sua concepção pessoal. Fazendo isto, o aluno condensa, relembra e reforça os seus conhecimentos, além de incluir os novos pensamentos em seu panorama. Sendo assim, uma simples elaboração de uma pergunta instiga o aluno e faz com que o mesmo possa vislumbrar, desde um aumento de conhecimento até uma educação, no sentido de comparar conhecimentos.

Um dos principais problemas encontrados nas perguntas feitas dentro de sala de aula, é a sua pequena sofisticação e rara frequência. Tais problemas, podem ser associadas à barreiras em três níveis diferentes, como proposto por Graesser [12]:

- A primeira é a dificuldade por parte dos estudantes de identificar as suas próprias deficiências de conhecimento, a não ser que possuam um grande domínio do conhecimento. Há uma dificuldade dos estudantes em detectar informações contraditórias, em identificar a falta de dados essenciais e em discernir, quais são as informações supérfluas e quais são as importantes;
- Uma segunda barreira é a barreira social, onde por exemplo: perguntar uma questão "ruim" pode ser considerado uma perda de status;
- A terceira barreira seria o déficit em se conseguir adquirir boas habilidades de questionar-responder, já que muitas vezes nem os próprios professores são bons exemplos, pois muitas vezes nem eles mesmos costumam fazer boas perguntas, ou dar boas respostas.

### 3.1.3 O questionamento 1 para 1

Uma pergunta feita em um ambiente onde haja somente duas pessoas pode remover muitas barreiras que os estudantes encontram para construir uma questão. É muito comum estudantes possuírem tutores ou professores particulares, ou os mesmos serem disponibilizados pelas instituições de ensino. Tais pessoas conseguem responder de forma mais direta e auxiliam o aluno no desenvolvimento das suas habilidades de questionador.

### 3.1.4 Ambientes Virtuais de Aprendizagem

A partir dos avanços tecnológicos obtidos nos séculos XX e XXI, principalmente com os avanços da computação, das comunicações e da educação, como descrito em Pereira-Schmitt [25] foram se estabelecendo os ambientes virtuais de Aprendizagem (AVAs). Em um primeiro momento, com a criação de turmas e aulas a distância, utilizando vídeo-aulas, e posteriormente progredindo em direção ao ensino e auxílio-escolar, utilizando a Informática e a Internet, como temos hoje.

Mais especificamente, o ambiente virtual de aprendizagem (AVA) voltado para a Internet, consiste em uma opção de mídia que está sendo utilizada para mediar o processo ensino-aprendizagem a distância. Os ambientes mais comuns são:

- Web-based Learning;
- Online Learning;
- Learning Management Systems;
- Virtual Learning Environments;
- E-Learning.

Os ambientes virtuais de aprendizagem apresentam diversas vantagens:

- Maior capacidade de difusão;
- Pode atender uma grande quantidade de alunos;
- É bastante atrativo, quando comparado aos métodos tradicionais de ensino;
- Possui um menor custo por aluno, após o seu desenvolvimento;
- Atende cada estudante de forma individual, podendo ser mais específico;
- Há a possibilidade de integrar atividades pedagógicas com o meio virtual;
- Possui um menor custo de manutenção da ferramenta;
- Trata os alunos com isonomia.

No entanto ele também traz consigo várias desvantagens como:

- Tem um alto custo de desenvolvimento;
- Demora um longo tempo para ser desenvolvido;
- Depende muito da rede de infra-estrutura a qual será submetido;
- Requer uma manutenção bastante especializada da ferramenta;
- Enfrenta resistência dentro do próprio meio escolar;
- Enfrenta resistência entre alguns alunos;
- Problemas de temporalidade.

Os AVAs podem ser construídos de formas genéricas, a fim de atender um público geral, ou podem ser mais específicos, dependendo de cada aplicação. Vale ressaltar as diferenças que cada um pode ter, como por exemplo em relação a faixa etária do público alvo, que normalmente passa pelo público infantil, jovem, adulto e terceira-idade, podendo variar também quanto a intimidade do mesmo com os meios eletrônicos e de diversas outras formas.

### **Tutores virtuais**

Os tutores virtuais, ou somente tutores, são sujeitos com algum conhecimento, que auxiliam os alunos na execução de tarefas, pequenas lições e esclarecimento de dúvidas, tudo isso de forma remota. Normalmente tais tutores são pessoas que não possuem o grau de instrução de um professor, porém são versadas no objetivo do estudo em questão.

Há ainda o tutor não-humano, ou seja, aquele que é na verdade um sistema informatizado, que auxilia o aluno como se fosse um tutor normal. Tal tutor, apresenta várias vantagens, pois:

- estará 24 horas disponível ao aluno;
- é menos dispendioso do que manter uma pessoa física;
- é mais rápido;

- dentro de suas capacidades, normalmente não comete erros, se bem construído.

O Tutor Virtual é uma tendência muito interessante, pois muitas vezes nem o aluno saberá dizer ao certo se o mesmo é uma máquina ou uma pessoa.

Porém, o tutor não-humano é de difícil concepção, mantendo uma relação de grande custo e grandes benefícios, ficando a cargo de cada implementação verificar a viabilidade do mesmo (já que existem casos em que o mesmo é simplesmente inapropriado).

### 3.1.5 Uso neste contexto

Sendo a linguagem natural, a utilizada para a elaboração de perguntas, as perguntas que são alvo deste estudo são aquelas elaboradas pelos alunos e postadas de forma virtual no ambiente designado, sendo as mesmas classificadas, aferindo ao seu respectivo autor uma "reputação". A reputação de um questionador será construída de forma contínua e incremental, levando em conta a qualidade das perguntas elaboradas.

Através das análises dos dados obtidos, os alunos poderão verificar o seu desempenho e conforme for, modificar seus atos e formas de construir as perguntas, de forma a melhorá-las, ou simplesmente receber um reforço positivo e por consequência serem estimulados a continuarem o bom trabalho. No que tange aos dados auferidos por parte dos professores, os mesmos poderão detectar os vários aspectos que podem estar presentes em uma turma heterogênea, verificando os pontos de dificuldade coletivos e individuais.

## 3.2 Ontologia

Ontologias são usadas nas mais variadas áreas do conhecimento, para organizar a informação, tendo diversas definições, tipos e propostas para aplicações. Uma ontologia é de forma geral, uma forma de organizar, categorizar e inter-relacionar o conhecimento.

### 3.2.1 Ontologia para a Filosofia

Apesar de os filósofos se utilizarem muito das ontologias para descrever o mundo, não se tem um consenso sobre a real semântica do termo [15]. De acordo com Almeida [1], é o ramo da metafísica que estuda os tipos de coisas que existem no mundo. Historicamente o termo ontologia tem origem no grego “*ontos*”, ser, e “*logos*”, palavra. O termo original é a palavra aristotélica “*categoria*”, que pode ser usada para classificar alguma coisa. Aristóteles apresenta categorias que servem de base para classificar qualquer entidade e introduz ainda o termo “*differentia*” para propriedades que distinguem diferentes espécies do mesmo gênero. A conhecida técnica de herança é o processo de mesclar “*differentias*” definindo categorias por gênero.

Segundo Blackburn [5], é também a parte da filosofia que trata da natureza do ser, ou seja, da realidade, da existência dos entes e das questões metafísicas em geral.

### 3.2.2 Ontologia para a Ciência da Computação

Para o contexto das ciências da computação e informação, "as ontologias fornecem um vocabulário para a representação do conhecimento. Sendo assim, caso exista uma

ontologia que modele adequadamente certo domínio de conhecimento, essa pode ser compartilhada e usada por pessoas que desenvolvam aplicações dentro desse domínio." [4]

Como disposto em [14], uma ontologia define um conjunto de primitivas de representação, com os quais modela-se um domínio de conhecimento ou discurso. As primitivas de representação são tipicamente:

- classes (ou conjuntos);
- atributos (ou propriedades);
- relacionamentos (ou as relações entre os membros da classe).

As definições das primitivas de representação incluem informações sobre o seu significado e restrições sobre a sua aplicação logicamente consistente. As ontologias são normalmente especificadas em linguagens, que permitem abstração longe de estruturas de dados e estratégias de implementação. Na prática as línguas de ontologias estão mais próximas do poder expressivo da lógica de primeira ordem, do que as linguagens utilizadas para modelar bancos de dados. Por esta razão, as ontologias estão no nível de "semântica", enquanto esquemas de banco de dados, são modelos de dados na "lógica" ou nível "físico". Devido à sua independência a partir de modelos de dados de nível inferior, ontologias são usadas para integrar bancos de dados heterogêneos, permitindo a interoperabilidade entre sistemas diferentes, e especificando interfaces para serviços baseados no conhecimento independentes.

### Tipos de ontologias

Diferentes tipos de ontologias, de acordo com seu grau de genericidade podem ser delineados [11]:

- Ontologias de representação do conhecimento, definem as primitivas de representação - como frames, axiomas, atributos e outros - de forma declarativa. Essa ideia abstrai os formalismos de representação;
- Ontologias gerais (ou comuns) trazem definições abstratas necessárias para a compreensão de aspectos do mundo, como tempo, processos, papéis, espaço, seres, coisas, etc.
- Ontologias centrais (*core ontologies*) ou genéricas de domínio definem os ramos de estudo de uma área e/ou conceitos mais genéricos e abstratos desta área. Por exemplo, a ontologia central de direito criada por Valente e Breuker inclui conhecimentos normativos, de responsabilidade, reativos, de agências legais, comportamentos permitidos, etc. Esses conceitos e conhecimentos foram agrupados nesta ontologia para que ela sirva de base para a construção de ontologias de ramos mais específicos do direito, como direito tributário, de família e outros;
- Ontologias de domínio tratam de um domínio mais específico de uma área genérica de conhecimento, como direito tributário, microbiologia, etc;
- Ontologias de aplicação procuram solucionar um problema específico de um domínio, como identificar doenças do coração, a partir de uma ontologia de domínio de cardiologia. Normalmente, ela referencia termos de uma ontologia de domínio.

Já de acordo com Wooldridge, em [31] podemos classificar as ontologias em função do seu grau de formalismo e do tipo de informação que ela dispõe:

- Ontologias Superiores: onde se começa por definir as classes mais gerais que se possa imaginar, como por exemplo a classe "coisa" e continua-se em direção as suas especializações, como por exemplo "coisa viva" ou "coisa não-viva";
- Ontologias de Domínio: onde se define desde o princípio, os conceitos apropriados para uma aplicação de domínio específico, por exemplo definindo conceitos relacionados a terminologia médica;
- Ontologia de aplicação: que define os conceitos utilizados por uma aplicação específica.

As ontologias podem variar de acordo com o seu grau de formalidade ou expressionalidade, podemos observar nesse sentido, a classificação informal dada por [18] e ilustrada na Figura 3.1, onde temos:

- Informais

Vocabulário controlado: é construída com o uso de algumas *palavras-chave* para classificar entidades.

Termos/glossário: é uma lista de termos com um vocabulário controlado, com algum esforço no sentido de definir os significados dos termos.

Dicionário de sinônimos: define termos que possuam o mesmo significado.

Taxonomias informais "é um": são vocabulários controlados organizados em uma hierarquia informal.

- Formais

Taxonomias formais "é um": são vocabulários controlados organizados em uma hierarquia formal, onde ficam claras relações de subordinação entre as classes.

Propriedades: aqui se permite que as classes possuam propriedades inerentes a elas, o que somado as subordinações das classes, permite tomar inferências e conclusões sobre as propriedades das classes.

Valores de restrições: estes valores contribuem ao dar as classes, informações adicionais sobre os seus relacionamentos, como por exemplo: "uma pessoa possui apenas um pai".

Restrições de lógica arbitrárias: uma ontologia pode ainda, conter restrições de lógica bastante arbitrárias, o que aumenta a precisão e a complexidade.

## Reuso de ontologias

De acordo com Wooldridge [31] criar uma ontologia desde o seu princípio é um processo complexo, demorado e oneroso. Pensando nisso, é largamente indicado que se faça o reuso de ontologias e suas bases de conhecimento, pois assim, aumenta-se a celeridade do processo, além de se aumentar a confiabilidade da nova ontologia, pois se bem escolhida,



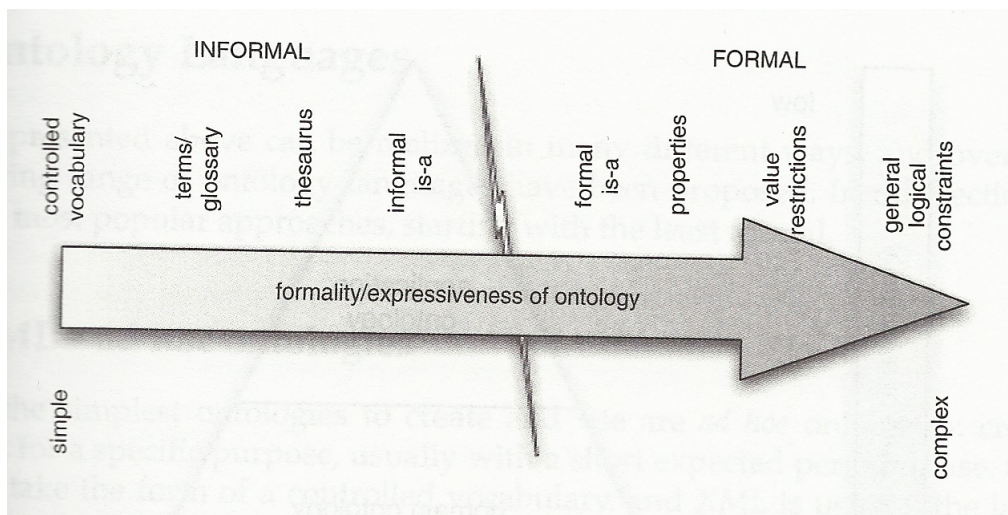


Figura 3.1: Espectro de ontologias, do informal e menos expressivo ao formal e mais expressivo [31]

a ontologia que venha a servir de base já terá sido largamente testada, incrementada e aperfeiçoada.

Porém, fazer o reuso não é uma tarefa trivial, pois grau de reusabilidade varia entre os tipos de ontologia, como podemos perceber na Figura 3.2 e também varia dentro de cada tipo, de acordo com as particularidades que cada uma apresenta.

### Exemplo de ontologias

Um bom exemplo de ontologia (do tipo "é um") pode ser uma ontologia de sobre trabalhos de arte, como mostrado na Figura 3.3.

Ou ainda uma ontologia que classifique pizzas, levando em conta a sua base, e o que ela possui por cima (sua cobertura), como na Figura 3.4.

### 3.2.3 OWL

Para se trazer a ontologia criada de um modelo descritivo, para uma linguagem que possa ser mais facilmente entendida e manipulada por computadores, utilizaremos a linguagem ontológica OWL [3], pois ela é a mais importante e influente linguagem ontológica na atualidade segundo [3].

A Web Ontology Language (OWL) é como o próprio nome sugere, uma linguagem de marcação semântica, desenvolvida para o uso por aplicações, que precisem processar o conteúdo das informações, ao invés de apenas exibí-las. OWL pode ser usada para explicitamente representar o significado de termos em vocabulários e os inter-relacionamentos entre os mesmos.

A OWL caracteriza-se por ter mais recursos para se expressar semanticamente, do que outras linguagens como XML, RDF e RDF-S, e ainda as supera no que toca a habilidade de representar conteúdo Web para a máquina. A OWL caracteriza-se historicamente por ser uma revisão de outra linguagem voltada para a ontologia na Web, a DAML+OIL, incorporando as experiências aprendidas na mesma [20].

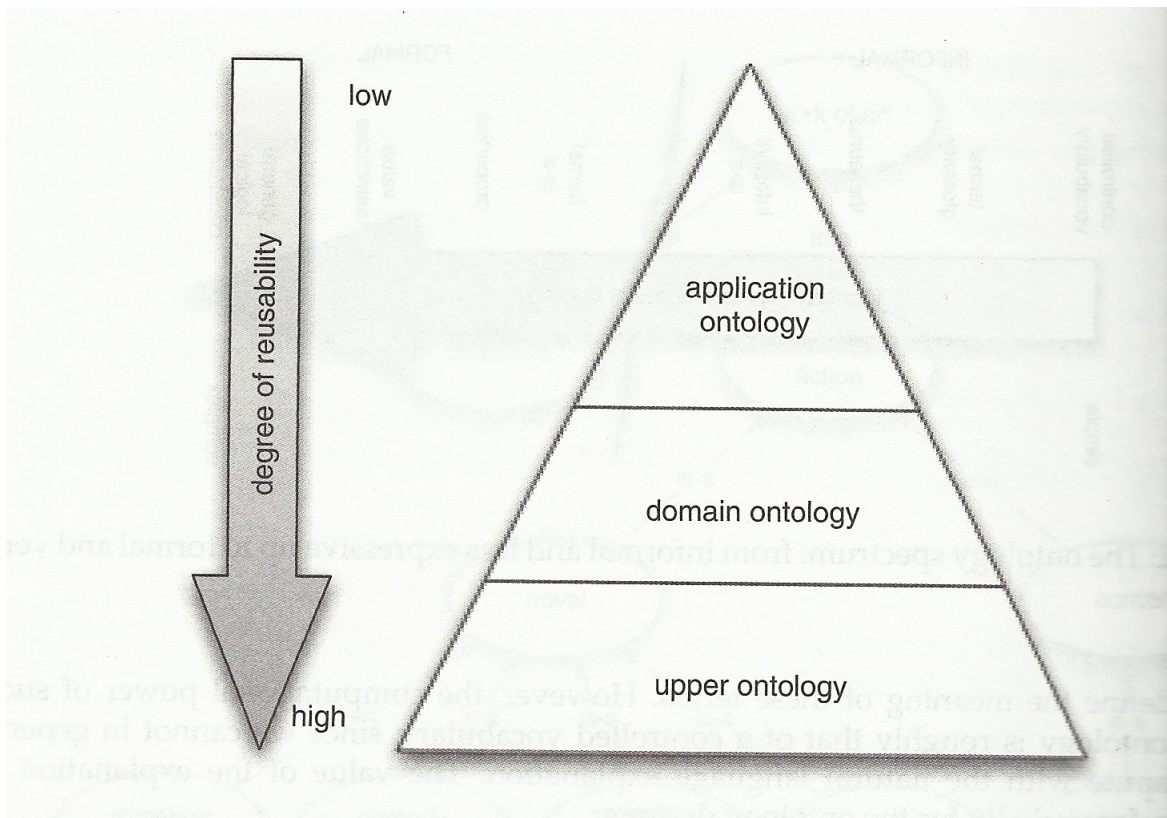


Figura 3.2: A hierarquia de ontologias: do mais geral (e reusável) na base, ao mais específico (e menos reusável) no topo [31]

Especificamente existem 3 *levels* (sub-linguagens) de OWL:

- *OWL Lite*: variante mais simples e menos expressiva, suportando apenas um pequeno número de recursos, mas em compensação é de um trato mais simples, facilitando a confecção e entendimento por parte das pessoas;
- *OWL DL*: é uma versão mais estendida da linguagem *OWL Lite* sendo um pouco mais poderosa;
- *OWL Full*: é uma forma muito expressiva de OWL, provendo inúmeras opções e recursos para definições, porém devido a sua riqueza e complexidade, acaba sendo de difícil trato, levando o criador da ontologia a muitos casos de falta de consistência da ontologia.

### 3.2.4 Ferramentas

Além das ferramentas usuais, utilizamos a ferramenta Protégé [10], tanto em sua forma local quanto a versão para Web. Fizemos isso, nos cadastrando na ferramenta e criando um projeto específico, denominado Projeto *quizWiki2*. Tal projeto foi disponibilizado de forma aberta para todos os colaboradores do projeto e deve ser adicionada a base de ontologias da própria ferramenta.

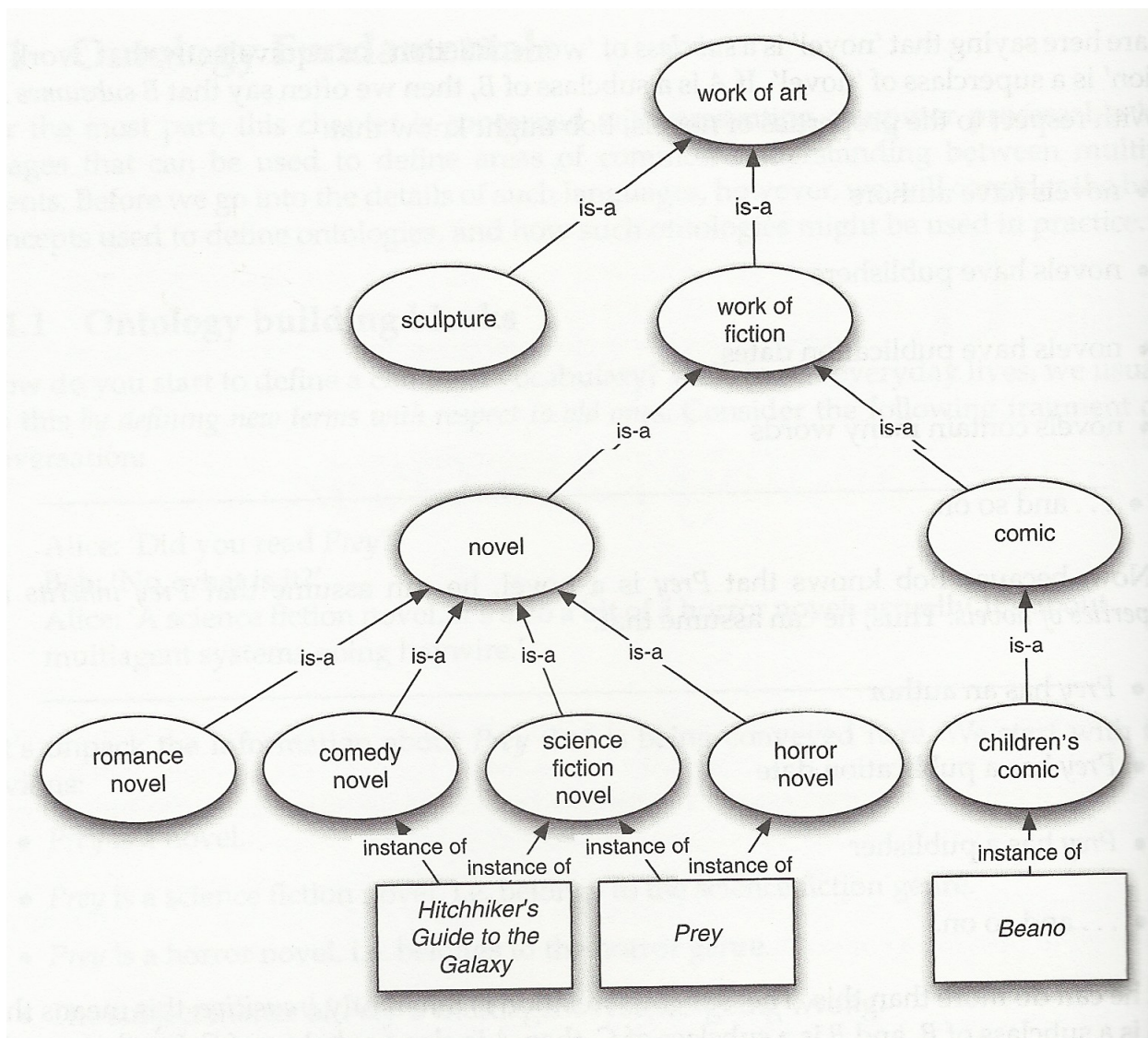


Figura 3.3: Exemplo de ontologia (do tipo *é um*) sobre trabalhos de arte [31]

### O Ambiente Protégé-2000

De acordo com Gennari, em [10], em seu conceito original o Protégé era uma ferramenta de aquisição de conhecimento, voltada a um sistema especialista para oncologia, o Oncocin. No entanto, a ferramenta foi sendo modernizada de forma gradativa e incremental, a fim de acompanhar a evolução das tecnologias e servir para aquisição de conhecimento, diretamente de especialistas de domínios, diminuindo a dependência de engenheiros de conhecimento e de forma a permitir diversos formalismos e estratégias de inferência, integrar tarefas (aquisição de ontologias e instâncias, ambiente de teste com inferência) em um mesmo ambiente, criar automaticamente formulários para entrada de conhecimento, acessar ontologias via OKBC e combinar ontologias.

Os responsáveis pelo Protégé, ao verificarem o potencial de desenvolvimento que tinha em mãos, optaram por abrir o código da ferramenta. Assim, surgiu uma arquitetura integrável a diversas aplicações, via componentes que podem ser adaptados e inseridos ao

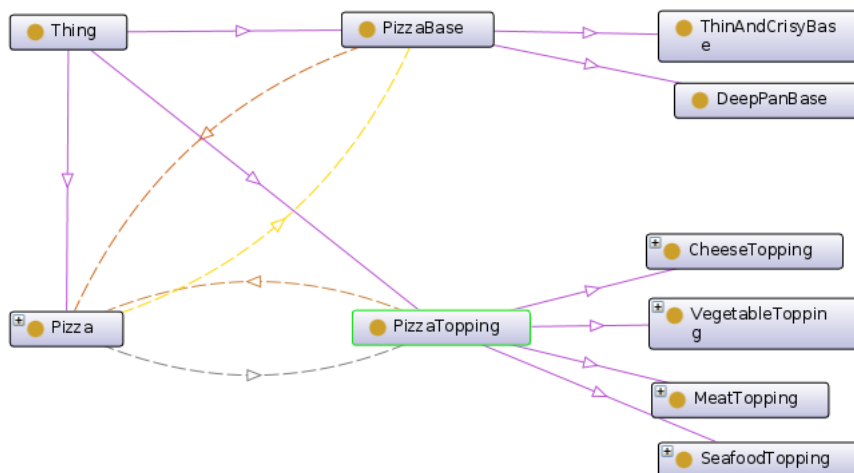


Figura 3.4: Exemplo de ontologia sobre Pizzas

sistema. Conseqüentemente, houve uma grande popularização da ferramenta no seu nicho de desenvolvedores, componentes de várias matizes, foram elaborados por grupos de pesquisa de usuários e puderam ser adicionados ao sistema, sem exigir o re-desenvolvimento.

### WebProtégé

Baseado no Protégé-2000, foi desenvolvido o WebProtégé [28], que é um ambiente colaborativo de desenvolvimento de ontologias, grátis e *open source* desenvolvido pela equipe Protégé no *Stanford Center for Biomedical Informatics Research*, que fornece as seguintes funcionalidades:

- Suporte para edição de ontologias OWL;
- Interface de edição simples padrão, que fornece acesso a construções OWL comumente usados;
- Controle de alterações completa;
- histórico de revisão;
- Ferramentas de colaboração, como: compartilhamento e permissões, anotações compartilhadas e discussões, relógios e notificações de e-mail;
- Interface de usuário personalizável;
- Formulários da Web personalizáveis para edição específica de aplicação;
- Formulários da Web personalizáveis para edição específica de domínio;
- Suporte para edição de ontologias OBO;
- Vários formatos para *upload* e *download* de ontologias;

formatos suportados:

RDF / XML;

Turtle;  
OWL / XML;  
OBO, e outros.

Por uma questão de praticidade, utilizamos o WebProtégé como ferramenta para desenvolvimento de ontologias neste trabalho, pois o mesmo disponibiliza uma série de vantagens além das descritas anteriormente:

- Não carece de instalação prévia;
- É uma ferramenta de fácil manejo e adaptação;
- Disponível 24 horas por dia, contanto que haja uma conexão com a Internet;
- Vasta documentação;
- Possui uma vasta gama de *pluggins* disponíveis;
- Trata-se de uma ferramenta largamente utilizada para estes fins, possuindo uma grande comunidade de usuários, que produzem muito material de interesse (desde novas funcionalidades até documentações).

Há também uma quantidade considerável de desvantagens, como: a instabilidade da rede de Internet na UnB (no que toca a rede wireless UnBWireless), ou a utilização da "nuvem" para armazenamentos, assim como os problemas de privacidade relacionados a ela, porém os benefícios ainda se mostram superiores.

## Protégé Local

Localmente foi utilizada a última versão do Protégé disponibilizada, a versão 5.0 beta [26]. Optamos por utilizar a versão local, pois a mesma oferece suporte a uma ferramenta de extrema importância para o nosso projeto, o *Reasoner*. Por ser uma versão beta, a mesma sofre com alguns problemas de instabilidades, principalmente com uma falta de sincronia entre a aba OntoGraf (que mostra graficamente a ontologia) e a versão mais atual do desenvolvimento, além de apresentar problemas na nomeação das entidades, reconhecendo aleatoriamente sinais gráficos.

## *Reasoner*

Quando trabalhamos com ontologias, é bastante complicado fazer a validação da mesma, pois mesmo uma pequena ontologia, requer modelos de verificação extremamente grandes, indo desde verificações simples, como de nomenclaturas, até questões mais conceituais, como veremos mais detalhadamente nas seções 4.1.5 e 4.1.4. Para tal tarefa foram criadas ferramentas específicas, chamadas *reasoners*, que podem, em poucos segundos, validar uma ontologia ou mostrar suas inconsistências, erros e demais problemas, muitas vezes até, fazendo sugestões de correções e inferências.

Tendo tais facilidades em vista, e levando em conta o desejo de ao final deste trabalho, de fornecer uma forma de inferir se uma pergunta apresentada é boa ou ruim, a utilização de um *reasoner* torna-se o caminho natural. Optamos por utilizar o Hermit [27] em sua

versão 1.3.8.3, que segundo Shearer-Motic-Horrocks é um *reasoner* de descrição lógica, baseado em uma nova arquitetura, onde o mesmo ataca o grande número de diferentes possibilidades de construção de um modelo, usando um algoritmo tabelamento, que deve analisar cada uma das possibilidades antes de concluir que nenhum modelo é possível. Tal escolha foi feita pois o Hermit além de ser o reasoner nativo do Protégé, se mostrou mais amigável e mais estável que os demais.

### 3.3 O *quizWiki*

Como apresentado anteriormente no capítulo 2.1, o *quizWiki* [7] é um ambiente computacional de apoio à elaboração de questões por parte dos estudantes, baseada no *MediaWiki*.

#### 3.3.1 O ambiente wiki no contexto pedagógico

O ambiente wiki foi originalmente concebido de uma forma simples, para permitir trabalho cooperativo em textos e hipertextos, porém suas características motivaram funcionalidades adicionais, nos chamados clones [9].

##### O exemplo da *Wikipédia*

Um exemplo desses clones é o *MediaWiki* desenvolvido para a *Wikipédia* e que tornou-se um dos clones mais populares. A edição de uma página conta com uma caixa de ferramentas contendo botões com a sintaxe das principais marcações. Associada à página, aparece uma aba *discussion* para discussão de seu conteúdo.

##### Contexto educacional

No contexto educacional, podemos destacar vários casos de sucesso na utilização desse ambiente, como o *COWEB* (Collaborative Website) [16]. O *COWEB* permite a criação e modificação de páginas Web através de um navegador Web, sem qualquer software especializado no lado do cliente. Cada página *COWEB* se parece com uma página Web normal, e pode conter qualquer tipo de mídia que os navegadores da Web podem apresentar. O que torna o *COWEB* incomum é que cada página tem um *link Editar*. Ao clicar em "*Editar*" abre uma página com uma área de texto para mudar o nome da página e uma área de texto grande com o conteúdo da página. Ao digitar um novo texto para a grande área de texto, em seguida, clicar em "Salvar", o autor muda o conteúdo da página.

Em [23] utiliza-se um wiki Semântico, no auxílio à disciplina Linguagens Formais e Autômatos, integrando tanto as partes teóricas da disciplina, com mapas conceituais e textos, como a parte prática, na forma do editor dedicado e dos recursos de desenho gráfico do wiki.

### 3.3.2 *QuizWiki*: um ambiente para incorporar a elaboração discente de perguntas no processo de aprendizagem

A proposta encontrada no *quizWiki* é adaptar e integrar trabalhos anteriores da literatura com a finalidade de promover a aprendizagem a partir da interação e do questionamento.

O *quizWiki* contribui para o contexto educacional, promovendo:

- Elaboração de perguntas;
- Avaliação de perguntas.

A provisão de respostas está prevista para a próxima versão do software [7].

### 3.3.3 O ambiente do *quizWiki*

O ambiente *quizWiki* é proposto para utilização em apoio ao ensino presencial ou em contexto de Educação à Distância. Se o curso é organizado em uma estrutura de tópicos, um *quiz* é associado a cada tópico, aberto à medida em que é estudado.

O *quizWiki* funciona utilizando o conceito de reputação, conforme Yao-Vassileva [30], onde o usuário recebe uma determinada reputação que reflete a sua participação no ambiente.

De forma geral, o processo se dá da seguinte forma (já previamente indicado na Figura 2.1):

1. O professor configura o *quiz* de tópico, o professor informa quantas perguntas deseja para o tópico e informa a data para o fechamento do tópico ;
2. Em seguida o estudante participa do *quiz*, elaborando suas perguntas e adicionando a elas um valor de "investimento", que representa o quanto o aluno confia naquela pergunta;
3. Após isto, a comunidade e o professor avaliam as perguntas que foram desenvolvidas;
4. O sistema calcula a nota que o aluno obteve com as suas perguntas e retorna esta nota como uma nota de reputação.

Podem-se destacar ainda, algumas particularidades, como o fato dos tópicos nunca serem "fechados", mas apenas irem diminuindo a sua recompensa (impacto positivo na reputação) conforme o tempo vai passando. Há também a possibilidade de o estudante visualizar as informações sintetizadas de sua participação ao longo dos *quizzes* no ambiente: suas reputações, a classificação refletindo seu grau de adesão à comunidade, a(s) mensagem(ns) personalizada(s) com a(s) recomendação(ões) do *quizWiki* para ele no momento e a lista de tópicos abertos.

### 3.3.4 MediaWiki

*MediaWiki* [22] é um software livre, de código *wiki* aberto, usado para sustentar sites *wiki*, como a *Wikipédia*, *Wikcionário*, *Commons* e por milhares de outros sítios, sendo

desenvolvido pela *Wikimedia Foundation* e alguns parceiros. Ele é escrito na linguagem de programação *PHP* e usa um banco de dados *back-end*.

O software tem mais de 700 definições de configuração e mais de 1.800 extensões disponíveis para permitir que vários recursos sejam adicionados ou alterados. Sozinha, a *Wikipédia* possui mais de 1.000 *bots* automatizados, semi-automatizados e outras ferramentas que foram desenvolvidas para ajudar na sua própria edição.

## Licenciamento

O *MediaWiki* é gratuito e de código aberto, sendo distribuído sob os termos da *GNU General Public License versão 2* ou qualquer versão posterior, enquanto a sua documentação é liberada sob a licença *Creative Commons by-sa 3.0* e em parte como domínio público. Especificamente, os manuais e outros conteúdos em *MediaWiki.org* são licenciados como *Creative Commons*, enquanto o conjunto de páginas de ajuda destinados a ser copiado livremente em instalações novas e/ou distribuído com o software *MediaWiki* é de domínio público. O desenvolvimento do *MediaWiki* geralmente favorece o uso de formatos de mídia de código aberto.

## Características

O *MediaWiki* possui uma série de características e mecanismos que provêem uma gama de vantagens e funcionalidades, sendo estes um dos seus grandes diferenciais e pontos-fortes. Podemos destacar:

- **Internacionalização e localização:** Devido à forte ênfase no multilinguismo nos projetos *Wikimedia*, internacionalização e localização tem recebido atenção significativa por parte dos desenvolvedores. A interface do usuário foi total ou parcialmente traduzidas para mais de 300 idiomas através do *translatewiki.net*, e pode ser personalizada pelos administradores (toda a interface é editável através do *wiki*);
- **Instalação e configuração:** O *MediaWiki* possui uma instalação e configuração relativamente simples, sendo necessário que o usuário possua privilégios de administrador, em um servidor que rode tanto *PHP* quanto utilize *SQL*. Ele pode ainda ser hospedado em uma "nuvem" ou em uma máquina virtual;
- **Remarcação:** Uma das primeiras diferenças entre *MediaWiki* (e seu antecessor, *UseModWiki*) e de outros mecanismos *wiki* foi o uso de "*links livres*" em vez de *CamelCase*. Quando *MediaWiki* foi criado, era comum que *wikis* exigissem o texto, como "*WorldWideWeb*", para criar um *link* para uma página sobre a *World Wide Web*. Por outro lado, links no *MediaWiki*, são criados por palavras que foram colocadas entre colchetes duplos, e quaisquer espaços entre elas são deixadas intactas, por exemplo, [[World Wide Web]]. Esta mudança era lógica para o propósito de criar uma enciclopédia, onde a precisão nos títulos é importante;
- **Interface de edição:** Existe toda uma interface de edição das páginas, bastante simples e intuitiva, que conta com auto-corretores gramaticais e esquemas que fazem "merges" de informações editadas no mesmo instante;



- ***Application Programming Interface:*** É utilizada uma *API* extensível para Web, que fornece acesso direto de alto nível, para os dados contidos nas bases de dados do *MediaWiki*. Os programas clientes podem usar *API* para fazer *log-in*, obter dados e postar as mudanças. A *API* suporta clientes *JavaScript* baseados na Web e aplicativos voltados ao usuário final (como ferramentas de combate a vandalismo). Uma das vantagens percebidas da *API* é a sua independência da linguagem, ela escuta para conexões *HTTP* de clientes e pode enviar uma resposta em uma variedade de formatos, tais como *XML*, *PHP* serializado, *YAML* ou *JSON*. O código Cliente foi desenvolvido para fornecer camadas de abstração para a *API*;
- **Riqueza de conteúdo:** É suportada uma vasta gama de conteúdos, desde que eles sejam gerados através de sintaxes especializadas. Por exemplo, o software vem com suporte opcional para renderizar fórmulas matemáticas usando *LaTeX* e um analisador especial escrito em *OCaml*. Existem funcionalidades semelhantes para outros conteúdos, que vão desde linhas temporais, plotagens matemáticas e partituras musicais até hieróglifos egípcios;
- **Edições rastreáveis:** Pode-se fazer um rastreamento das páginas alteradas, resultando em uma lista, que contém informações básicas sobre essas edições, como o usuário a edição, o sumário de edição, a página editada, bem como existem filtros e *bots anti-vandalismo* que inibem e corrigem alterações indevidas/abusivas;
- **Navegação:** A navegação pelos sítios criados é bem simples, se dando através de pesquisas e links externos, conta-se também com a familiaridade que os usuários já possuem, por navegar na *Wikipédia*;
- **Organização do conteúdo:** O software organiza o conteúdo de forma intuitiva:
  - Guias de páginas e páginas associadas;
  - Namespaces*;
  - Tags* de categorias;
  - Sub-páginas.
- **Customizações:** Se este recurso for ativado, os usuários podem personalizar suas folhas de estilo e configurar do lado do cliente o *JavaScript* para ser executado a cada *pageview*. Na *Wikipédia*, isto levou a um grande número de ferramentas e auxiliares adicionais, que foram desenvolvidos por meio da *wiki* e compartilhados entre os usuários.

Toda a interface de usuário *MediaWiki* pode ser editada através da própria *wiki*, por usuários com as permissões necessárias (normalmente permissões de "*Administradores*"). Isso é feito através de um *namespace* especial com o prefixo "*MediaWiki:*", onde cada título da página identifica uma mensagem de interface de usuário particular. Usando uma extensão, também é possível para um usuário para criar *scripts* pessoais e escolher se certos parâmetros devem aplicar-se a ele, alternando as opções apropriadas na página de preferências do usuário;
- **Templates:** *Templates* têm encontrado muitos usos diferentes, eles permitem aos usuários criar *layouts* diversos, que são utilizados de forma consistente em várias páginas, onde apenas o conteúdo das tabelas é inserido através de parâmetros do

modelo. Os *templates* são muitas vezes utilizados para identificar problemas com um artigo da *Wikipédia*, colocando um modelo no artigo. Este modelo, em seguida, envia uma caixa gráfica afirmando que o conteúdo do artigo é contestada ou que necessitam de alguma outra atenção, e também categoriza-o, de modo que os artigos desta natureza possam ser localizados;

- **Grupos e restrições de acesso:** É oferecida flexibilidade na criação e definição de grupos de usuários. Por exemplo, seria possível a criação de um grupo arbitrário que pode bloquear usuários e excluir páginas, e cuja edições são ocultas por padrão nas recentes mudanças de *log*. Também é possível a criação de um grupo de usuários "*autoconfirmed*" em que uma pessoa se torna membro depois de fazer um certo número de edições e à espera de um certo número de dias. Alguns grupos que são habilitados por padrão são: *burocratas* e *sysops*. Os *burocratas* têm poder para alterar os direitos de outros usuários. Os *Sysops* tem poder sobre a proteção da página e exclusão e bloqueio de usuários de edição. Controles disponíveis do *MediaWiki* sobre direitos de edição foram consideradas suficientes para a publicação e manutenção de documentos importantes, como um manual de procedimentos operacionais padronizados em um hospital.

O *MediaWiki* vem com um conjunto básico de características relacionadas à restrição de acesso, mas o seu *design* original e em curso é orientado por funções que, são em grande parte relacionadas ao conteúdo, e não a segregação de conteúdo. Como resultado, com exceções mínimas (relacionados com ferramentas específicas e seus afins, as páginas "*Especiais*" ), controle de acesso a página nunca foi uma prioridade no desenvolvimento do núcleo e os desenvolvedores afirmaram que os usuários que necessitam de acesso seguro ao usuário e controles de autorização não deve confiar no *MediaWiki*, uma vez que, o mesmo nunca foi projetado para esses tipos de situações. Por exemplo, é extremamente difícil criar uma *wiki* onde apenas alguns usuários possam ler e acessar determinadas páginas páginas;

- **Extensibilidade:** A base de código *MediaWiki* contém vários "ganchos", usando as funções de retorno de chamada para adicionar código adicional *PHP* de uma maneira extensível. Isso permite que os desenvolvedores escrevam extensões, sem necessariamente precisar modificar o núcleo ou de ter de submeter o seu código para revisão. Instalar uma extensão, normalmente consiste em adicionar uma linha ao arquivo de configuração, que em alguns casos requer mudanças adicionais, tais como atualizações de bancos de dados ou correções básicas.

## 3.4 *Semantic Media Wiki*

### 3.4.1 Conceitos Básicos

O *Semantic MediaWiki* (ou simplesmente *SMW*) [17] é uma extensão do software *MediaWiki*, aplicação *wiki* mais conhecida para alimentar a *Wikipédia*, que ajuda a pesquisar, organizar, relacionar *tags*, navegar, avaliar e compartilhar o conteúdo da *wiki*. Enquanto *wikis* tradicionais contêm apenas textos, que os computadores não conseguem entender ou avaliar, o *SMW* acrescenta anotações semânticas, que permitem que uma

*wiki* funcione como um banco de dados colaborativo. O *Semantic MediaWiki* foi lançado em 2005 e está em uso em centenas de sites.

Para estender a capacidade de editar, visualizar e navegar através dos dados armazenados, um grande número de extensões foi criada, tanto pela equipe responsável pelo projeto do SMW, quanto por desenvolvedores independentes. Vale ressaltar que o termo "*Semantic MediaWiki*" as vezes é utilizado para se referir a toda esta família de extensões.

### 3.4.2 Vantagens do *Semantic MediaWiki*

O *Semantic MediaWiki* apresenta algumas marcações adicionais no *wiki-texto*, o que por sua vez permite aos usuários adicionar "*anotações semânticas*" para o wiki. Enquanto isso parece tornar as coisas mais complexas, também pode simplificar muito a estrutura da wiki, ajudar os usuários a encontrar mais informação em menos tempo, e melhorar a qualidade geral e a consistência da wiki. Podemos listar alguns dos benefícios do uso do SMW:

- **Listas geradas automaticamente:** Listas e tabelas são uma maneira de exibir informações de forma rápida e fácil. Em alguns casos, wikis não semânticas contêm listas geradas por humanos, a própria *Wikipédia* possui várias delas, como por exemplo: a lista de áreas metropolitanas de determinado país, indexada pela sua população. Estas listas são propensas a erros, uma vez que têm que ser atualizadas manualmente, o que não ocorre nas listas geradas automaticamente pelo SMW. Além disso, o número de listas potencialmente interessantes é enorme, e é impossível fornecer todas elas em qualidade aceitável. Com o SMW, as listas são geradas automaticamente, assim elas estão sempre atualizadas e podendo ser facilmente personalizadas para obter informações adicionais, abrem-se as possibilidades de manipulações tais como uniões, intersecções de listas, assim como diversas outras operações, que feitas de forma automatizada podem apresentar resultados bastante interessantes;
- **Apresentação visual de informações:** Ficam estabelecidas várias formas de se apresentar as informações, apoiado por vários formatos de exibição definidos por extensões adicionais, como Formatos de Resultados Semânticos, permitem a exibição de informações em formatos visuais, tais como calendários, gráficos, matrizes, mapas, dentre outros, proporcionando uma visão muito mais interessante dos dados;
- **Estrutura de dados melhorada:** *Wikis MediaWiki* tendem a fazer uso intensivo de categorias guardar os seus dados de forma estruturada. Porém, com o uso extensivo do SMW as categorias podem ser substituídas por valores semânticos simples, reduzindo a necessidade de um sistema de classificação complexo. Além disso, se uma marcação semântica dentro da *wiki* é armazenada dentro de *templates*, dessa forma uma *wiki* pode facilmente ganhar uma estrutura de dados sólida. A extensão *Semantic Forms* permite aos administradores criarem formulários para adicionar e editar os dados dentro de modelos semânticos, tornando assim a adição de informação semântica ainda mais fáceis e mais simples do que usando o formato de um texto *wiki* regular;

- **Busca de informações:** Os usuários podem procurar informações específicas, criando suas próprias consultas, apoiados por extensões como o *Semantic Drilldown* e o *Semantic Forms*;
- **Re-uso externo:** Os dados criados em uma *wiki* utilizando o *SMW* não tem que permanecer dentro da *wiki*; eles podem ser facilmente exportados via formatos como *CSV* e *JSON*. Isso permite que uma *wiki* utilizando o *SMW* sirva como fonte de dados para outras aplicações, ou, no caso de usos empresariais, para assumir o papel que um banco de dados relacional normalmente assumiria. Através do uso da extensão *External Data*, *SPARQL*, e outras ferramentas, uma *wiki* baseada em *SMW* ainda pode usar os dados de outra *wiki* baseada em *SMW*, eliminando assim a necessidade de redundância entre *wikis*. Pode-se também consultar os dados do *SMW* de fora da *wiki*, por meio da *API* ou de um *RDF triplestore*;
- **Integrar dados e fazer *mash-up*'s:** Extensões como *Data Transfer* e *External Data* permitem ao usuário integrar dados externos (provenientes, por exemplo de sistemas legados, serviços Web ou fontes de dados vinculados) e relacioná-los a dados semânticos existentes na wiki, podendo fazer da sua wiki uma central de informações e um ambiente de mistura entre vários atores de conhecimento.

# Capítulo 4

## Proposta

### 4.1 Metodologia

#### 4.1.1 Modelo de desenvolvimento

Foi utilizado para o desenvolvimento deste trabalho, o modelo em espiral, conforme a Figura 4.1.

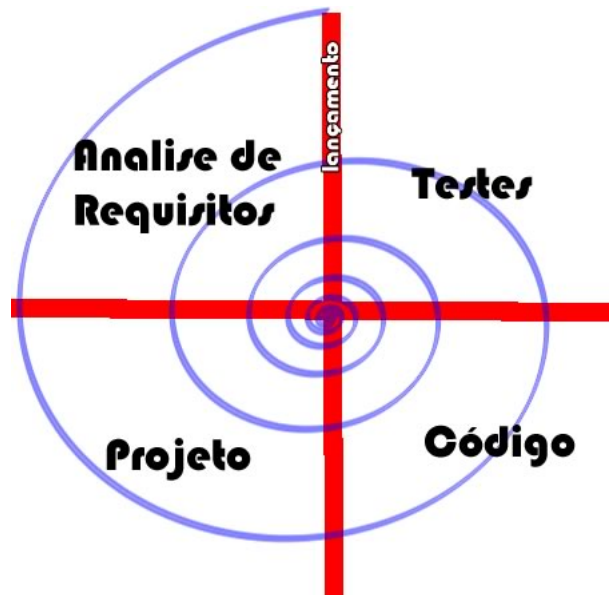


Figura 4.1: Modelo de desenvolvimento em espiral

#### 4.1.2 Construção da Ontologia

Para construir uma ontologia, existem diversas metodologias, com várias alternativas viáveis, dependendo sempre da aplicação. O desenvolvimento de uma ontologia é caracterizado por ser um processo necessariamente iterativo. Utilizamos aqui uma metodologia híbrida, extraíndo uma ontologia (ou pelo menos uma taxonomia) a partir de material teórico sobre o questionamento. No nosso caso, o material que serviu de base para a extração foi o livro "Aprendendo a Perguntar" [19] e fizemos um processo muito próximo

do proposto por [24], adotando pequenas alterações quando necessário, chegando a uma ontologia intermediária entre o proposto pelo especialista (no caso o autor do livro) e a nossa visão conseguida através da metodologia científica.

Após a extração da ontologia, em linhas gerais, como mostrado na Figura 4.2, seguimos os seguintes passos, propostos por Noy-McGuinness [24]:

- Passo 1: Determinar um domínio e escopo; podemos fazer isso levantando os requisitos da ontologia, respondendo por exemplo as seguintes perguntas:

Que domínio a ontologia vai cobrir?

Que questões esta ontologia será usada para responder?

Como esta ontologia será usada?

- Passo 2: Considerar o reuso; levar em conta durante o processo a possibilidade de criar uma ontologia que possa ser usada por outras pessoas ou organizações, em outras aplicações e situações;
- Passo 3: Enumerar todos os termos relevantes; fazer um *brainstorming* de todos os termos relevantes, associados ao domínio escolhido, listando-os;
- Passo 4: Definir as classes e a sua hierarquia; começaremos a organizar o nosso entendimento sobre o domínio, começando por identificar as classes e organizá-las. A organização das mesmas pode ser *top-down* das classes mais gerais em direção as mais específicas ou pelo contrário fazendo-a *bottom-up*;
- Passo 5: Definir as propriedades; identificaremos as propriedades inerentes a cada classe levantada. Tais propriedades podem ser dos mais diferentes tipos [20]:

*Intrinsic properties* relacionadas à natureza do objeto, como peso, tamanho, cor, estocamento.

*Extrinsic properties* são as relacionadas de forma abstrata, como: nome, identificador, sequência de fabricação.

*Components of an object* qualquer outro tipo de estrutura a qual o objeto esteja sujeito.

*Relationships* podem haver relacionamentos entre as propriedades e objetos, por exemplo: a propriedade "dono" está relacionada a um objeto do tipo pessoa.

- Passo 6: Definir as propriedades das propriedades, pois cada propriedade pode ter suas próprias propriedades intrínsecas, por exemplo:

*Cardinality constrains*

*Type constrains*

*Range constrains*

*Domain constrains*

- Passo 7: Criar instâncias, que é a etapa onde a ontologia é populada com as instâncias.

Alguns princípios, se usados com precisão, garantem a qualidade de uma ontologia [13]:

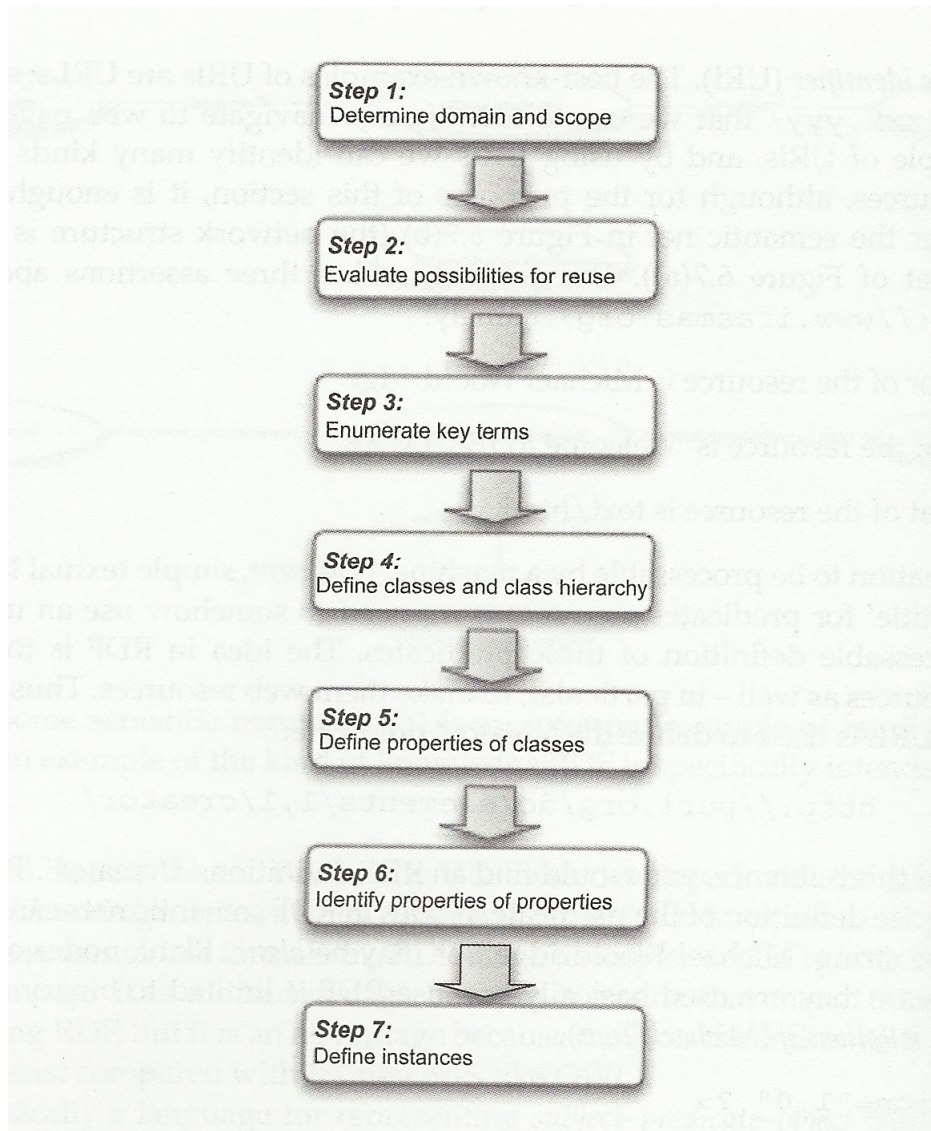


Figura 4.2: Desenvolvimento de ontologias de Noy and McGuinness (Figura retirada de [31])

- Clareza: Na definição do conhecimento, deve-se definir apenas o que se supõe ser útil, para se resolver os problemas a serem atingidos. Definições completas, com condições necessárias e suficientes, são preferíveis ao invés de definições parciais;
- Legibilidade: As classes, atributos e relacionamentos devem guardar certa correspondência para com as definições formais e informais. A ontologia deve usar um vocabulário comum, normalmente o jargão e terminologia usados por especialistas do domínio;
- Coerência: As inferências provenientes da ontologia que foi definida devem estar corretas, tanto do ponto de vista formal quanto informal, de acordo com as definições estabelecidas;

- Extensibilidade: A ontologia deve permitir a sua extensão e aprofundamento, sem a necessidade de revisão da teoria, de qualquer natureza.
- Mínima codificação: Devem ser especificados conceitos genéricos, de forma a diminuir a quantidade de codificação a ser feita (essa generalidade é limitada pela clareza);
- Mínimo compromisso ontológico: Para maximizar o reuso, apenas o conhecimento essencial deve ser incluído, gerando a menor teoria possível acerca de cada conceito, e permitindo a criação de conceitos novos, mais especializados ou estendidos.

Este projeto busca observar tais princípios.

### 4.1.3 Criação do modelo OWL

Ao criar a ontologia, esperamos ao utilizar o Protégé, obter um modelo OWL resultante.

### 4.1.4 Verificação do modelo OWL

Uma ontologia tem associada a si, uma infinidade de desafios relacionados a sua criação, principalmente no que se refere às dificuldades relacionadas ao seu *design* e ao seu uso, sendo os últimos, chamados de "desafios de raciocínio" [31]. Tais desafios podem ser classificados em *TBox* quando estamos tratando apenas das classes e seus relacionamentos e *TBox* e *ABox* quando as assertivas são incluídas também.

No que diz respeito ao raciocínio sobre as classes, podemos destacar:

- *Consistency Checking*: é um desafio de responsabilidade do *designer* da ontologia, onde em linhas gerais, deve-se verificar se a ontologia possui contradições explícitas ou implícitas, por exemplo, quando uma classe é subclasse dela mesma, ou ainda, de uma de suas subclasses. A complexidade desta verificação é proporcional à sublinguagem de OWL que esteja sendo usada, devido a suas diferentes complexidades;
- *Concept satisfiability*: esta é uma noção um pouco mais delicada de consistência, onde verificamos se a ontologia satisfaz os conceitos impostos previamente à ela. Por exemplo: seja uma ontologia  $X$  e um conjunto de classes  $A, B$  e  $C$ , onde  $A$  é uma classe não vazia,  $A$  e  $B$  são disjuntos e  $C$  é a intersecção entre  $A$  e  $B$ ,  $C$  só pode ser vazia, se não o for, a ontologia contém erros estruturais;
- *Computing the subsumption hierarchy*: aqui se verifica se as relações de subordinação estão corretamente estabelecidas, ou seja, para cada classe se verificará se as suas classes superiores deveriam realmente ser superiores, e o mesmo para as subordinadas e não-relacionadas, respectivamente. O processo se dá para todas as classes existentes, logo teremos  $(QxQ)$  verificações, onde  $Q$  é o conjunto que representa todas as classes da ontologia;
- *Least common subsumer*: aqui escolhe-se um conjunto de classes  $A, B, C, \dots, X$  e tenta-se encontrar a classe que contém especificamente todo este conjunto como sub-classe;

Quando tratamos conceitos e instâncias, temos um raciocínio adicional:



- *Instance classification*: munidos de algumas informações sobre um novo objeto  $Y$ , devemos classificá-lo na classe mais específica possível, mesmo não conhecendo todas as propriedades de  $Y$ .

#### 4.1.5 Verificação da ontologia criada

Existem na literatura, diversas propostas para a avaliação de ontologias, no entanto parecem existir poucas metodologias formais, não há um consenso. A construção de ontologias é ainda mais artesanal do que científica. Segundo Almeida [1] "Algumas questões básicas para a avaliação de ontologias são: Quais são os mecanismos para interagir com as ontologias? Qual é o formalismo de representação do conhecimento utilizado? A ontologia é bem documentada? A ontologia foi avaliada sob o ponto de vista técnico?".

Gómez-Perez [11] apresenta critérios que podem ser utilizados para avaliar ontologias. Os passos apresentados para a avaliação discorrem-se sobre os conceitos e definições que compõem a ontologia:

- Verificar a estrutura ou arquitetura da ontologia: as definições são construídas seguindo os critérios de projeto?
- Verificar a sintaxe das definições: existem estruturas ou palavras-chave sintaticamente incorretas nas definições?
- Verificar o conteúdo das definições: o que a ontologia define ou não? O que define incorretamente? O que pode ser inferido e o que não pode?

#### 4.1.6 Correções

As correções foram efetuadas conforme o modelo em espiral descrito na Figura 4.1 e as observações da professora orientadora.

## 4.2 Modelo Ontológico

Seguindo os passos sugeridos na Figura 4.2, foi desenvolvida a ontologia.

### 4.2.1 Passo 1, determinar um domínio e escopo

**Determinamos o escopo do projeto:**

O "produto" entregue é inicialmente uma ontologia, escrita em OWL sobre "questionamentos", o projeto está dividido em duas partes:

1. Criar a ontologia.
2. Implementar a integração entre a ontologia e o *quizWiki*.

## Requisitos Funcionais

1. Identificar uma pergunta.
2. Classificar uma pergunta.
3. Indicar se uma pergunta é boa ou não.

## Requisitos Não-Funcionais

1. Clareza na implantação.
2. Homogeneidade nos princípios norteadores das classificações.
3. Poucas ambiguidades.

## Determinamos o domínio, respondendo as perguntas levantadas:

1. Que domínio esta ontologia vai cobrir?
  - Questionamento;
  - Aprendizagem do questionamento;
  - Avaliação do questionamento.
2. Que questões esta ontologia será usada para responder?
  - Dada uma questão, trata-se de uma boa questão (no contexto)? (exemplo da questão que não é boa no absoluto, e.g. uma repetida);
  - O que é uma boa questão?
  - Que tipos de questões existem?
  - Em que situação usar cada tipo?
  - Em que situação não usar um certo tipo?
  - Que consequências podem advir do uso de certo tipo de questão?
  - Um tipo de questão pode indicar um comportamento de um questionador?
  - Um tipo de questão pode indicar um estado de um questionador?
3. Quem usará a ontologia?
  - Inicialmente, a aplicação *quizWiki*, via interface *MediaWiki*;
  - Os estudantes de forma indireta;
  - Qualquer um que deseje fazer re-uso da mesma.
4. Como esta ontologia será usada?
  - Antes de o estudante postar uma questão;
  - Antes de o estudante avaliar uma questão;
  - Após o estudante verificar sua reputação de elaborador;
  - Após o estudante verificar sua reputação de avaliador.

### 4.2.2 Passo 2, Considerar o reuso

Ontologias podem ser reutilizadas de diversas maneiras, que ora resultam na criação de uma ontologia independente a partir dos conceitos de outras (podendo ser estendidos e adaptados), ora preservam as ontologias originais.

Procuramos na literatura, ontologias que se adequassem ao desenvolvimento neste projeto, porém não fomos capazes de encontrar uma que fosse satisfatória aos nossos interesses. Logo, reutilizar uma ontologia previamente desenvolvida, acabou por não se tornar uma opção, levando-nos ao desenvolvimento de uma própria ontologia.

Como não fomos capazes de encontrar trabalhos semelhantes, optamos por tentar desenvolver uma ontologia bastante generalizada, a fim de propiciar e incentivar o reuso da mesma. Para isso, procuramos não nos ater a questões muito intrínsecas das ferramentas e da aplicação *quizWiki*.

### 4.2.3 Passo 3, Enumerar todos os termos relevantes

Através da técnica do *brainstorm*, e da extração de termos da literatura [19], levantamos os seguintes termos, como indicados na tabela 4.1.

### 4.2.4 Passo 4: Definir as classes e a sua hierarquia

Escolhemos a organização *top-down*, partimos então de nosso objetivo que é diferenciar as perguntas boas das ruins. A partir do *Brainstorm* contido na tabela 4.1, separamos alguns itens que julgamos mais importantes e os tomamos como classes, a saber:

- Frase
- Pergunta
- Contraproducente
- Ambígua
- Ardilosa
- Discriminatória
- Indutiva
- Retórica
- Boa
- Ruim
- Aberta
- De contato
- Investigadora
- Comparativa
- Conclusiva

Tabela 4.1: *Brainstorm* de palavras importantes relacionadas ao questionamento.

Palavra	Sinônimos e associados
Pergunta	questão, questionamento
Boa	ótima, perfeita
Ruim	péssima, mal feita
Indefinida	indistinguível, inclassificável
Aberta	solta,
Fechada	direta, sem rodeios, inflexível, fatos
Contraproducente	atraso, contra-mão
Investigadora	procura, curiosidade
Temporal	tempo, período
Regional	região, lugar
Instigante	estimula, constrange
Provocativa	insolente, constrangedora
Direta	clara, direcionada
Indireta	obtusa, oculta
Contato	ligação, ponte, começo
Pessoais	íntimas, secretas
Impessoais	públicas, qualquer pessoa
Fundamentadas	base, sustentadas, ordem
Simples	pequenas, fáceis, curtas
Comparativas	associativas, comparação
Extensivas	longas, cansativas
Precisas	certeiras, alvo, sem dúvidas
Hipotéticas	situações, suposição, inverdades
Reflexivas	espelhadas, se repetem, ligada
Conclusivas	finalidade, conclusão, fechamento
Fechada	dura, final, sem aberturas
Sim	concordância, anuência
Não	negativa, inverdade
Identificação	catalogação, esclarecimento
Indutiva	conduz, leva
Ardilosa	ruim, mal intencionada
Ambígua	duplo-sentido, imprecisa, incerta
Retórica	
Discriminatória	étnica, separadora, ruim

- Extensiva
- Fundamentada
- Hipotética
- Precisa
- Simples
- Reflexiva
- Fechada
- Identificação
- Sim ou Não
- Avaliação
- Resposta

A partir delas montamos uma hierarquia primitiva, separando as classes em níveis:

- Primeiro nível: Avaliação, Frase;
- Segundo nível: Boa, Ruim, Pergunta, Resposta;
- Terceiro nível: Aberta, Fechada, Contraproducente;
- Quarto nível: De contato, Investigadora, Identificação, Sim ou Não, Ambígua, Arditosa, Discriminatória, Indutiva, Retórica;
- Quinto Nível: Comparativa, Conclusiva, Extensiva, Fundamentada, Hipotética, Precisa, Simples e Reflexiva.

Visualmente temos algo como sugere a Figura 4.3.

Cada uma das classes foi criada, destacando-se algumas relações de equivalência e disjunções, conforme indicamos na tabela 4.2.

Tabela 4.2: Relações importantes nas classes

Classe	Equivalent To	Disjoint With
Boa	not (Ruim)	Ruim
Ruim	not (Boa), é_uma_pergunta_do_tipo some Contraproducente	Boa
Aberta	not (Fechada)	Fechada
Fechada	not (Aberta)	Aberta

Criamos então os *Object Properties*, que são a forma como as classes se relacionam entre si, e que são organizados conforme a Figura 4.4, sendo:

- *sofre\_uma*: Uma pergunta sofre uma avaliação;

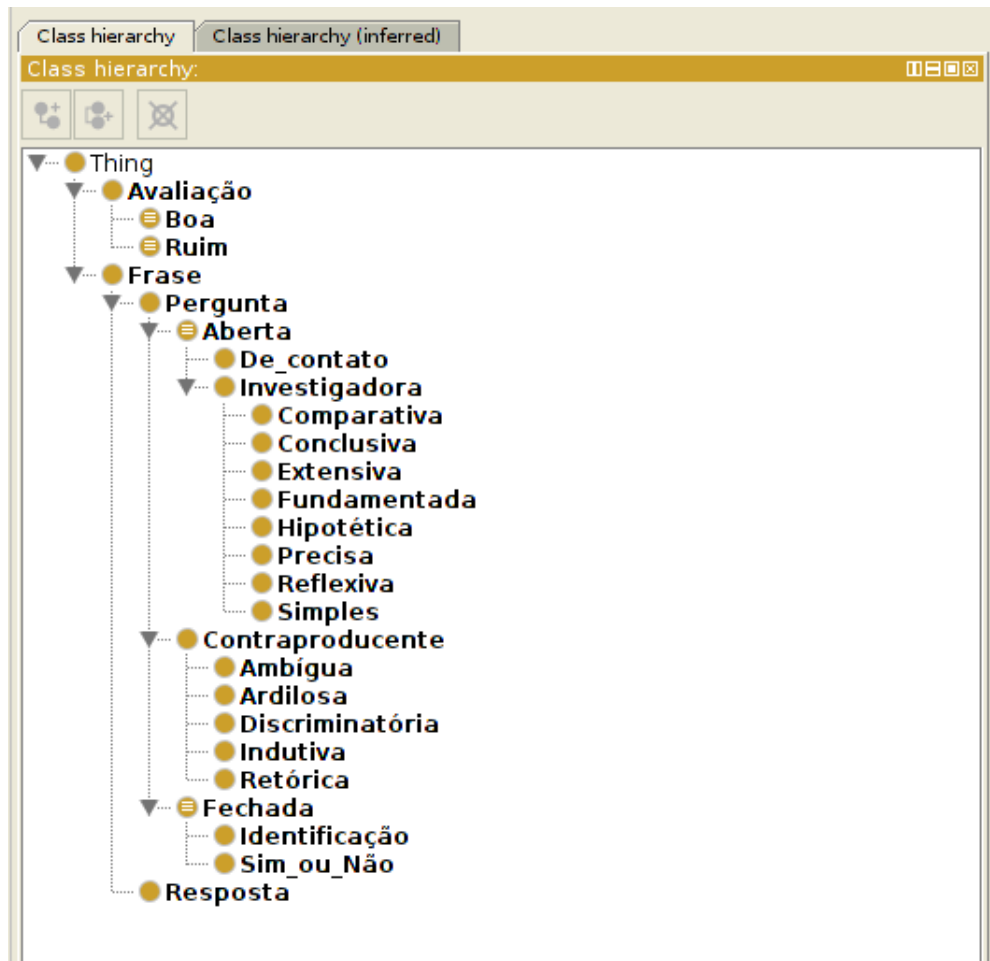


Figura 4.3: Classes organizadas em níveis

- *é\_dada\_como*: Uma avaliação é dada como boa ou ruim;
- *é\_aplicada\_sobre\_uma*: Uma avaliação é aplicada sobre uma pergunta;
- *é\_fruto\_de\_uma*: Uma pergunta boa ou ruim é fruto de uma avaliação;
- *é\_uma\_pergunta\_do\_tipo*: Uma pergunta é do seguinte tipo (podendo ser mais de um tipo) ;
- *remonta\_uma*: Uma pergunta fechada remonta uma identificação ou uma resposta de Sim ou Não;
- *tem\_intencao*: Uma pergunta aberta tem uma intenção de ser investigadora ou de estabelecer contato;
- *pode\_ser*: Uma pergunta com intenção investigadora pode ser.
- *é\_contraproducente\_pois\_é*: Uma pergunta é contraproducente pois apresenta alguns comportamentos;
- *é\_um\_tipo\_de*: Uma pergunta aberta, fechada e/ou contraproducente é um tipo de pergunta ;

- *também\_é\_uma*: Uma pergunta de identificação e ou Sim ou Não também é uma pergunta fechada;
- *é\_contraproducente*: uma pergunta que apresenta certas características é contra-producente;
- *é\_uma*: Uma pergunta Investigadora ou de Contato é uma pergunta aberta;
- *possui\_uma\_natureza*: Uma pergunta destas possui uma natureza investigadora;

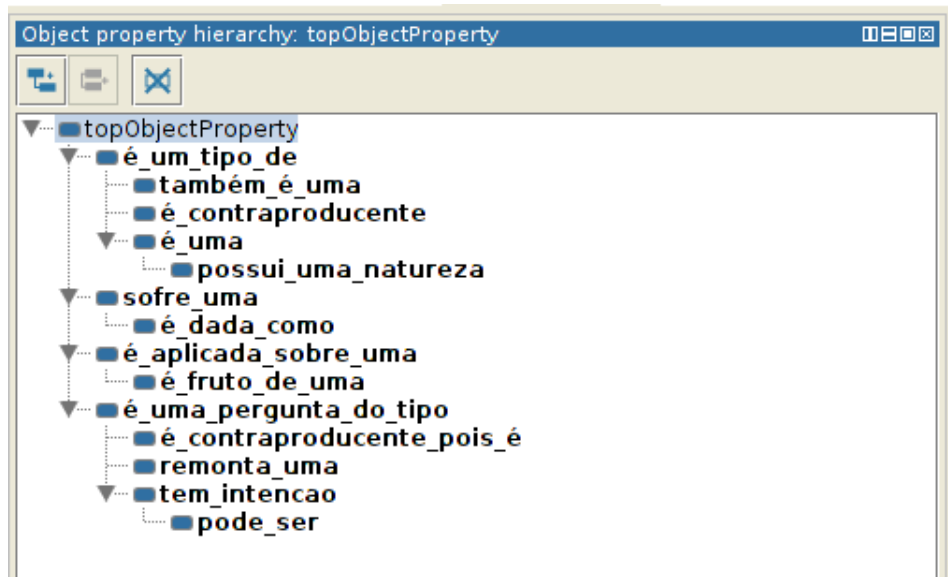


Figura 4.4: Organização dos *Object Properties*

Cada um desses *Object Properties* é constituído conforme a tabela 4.3.

#### 4.2.5 Passo 5, Definir as propriedades das classes

Para cada classe escolhida, podemos destacar propriedades inerentes (ou não):

- Pergunta

*Intrinsic properties:*

Formato interrogativo

*Extrinsic properties:*

Língua comum

- Boa

*Intrinsic properties:*

Tamanho: pequena, média ou grande

- Ruim

*Intrinsic properties:*

Tabela 4.3: Formação dos *Object Properties*

Object Property	Domain	Range	Inverse of
sofre_uma	Pergunta	Classificação	é_aplicada_sobre_uma
é_dada_como	Classificação	Boa or Ruim	é_fruto_de_uma
é_aplicada_sobre_uma	Classificação	Pergunta	sofre_uma
é_fruto_de_uma	Boa or Ruim	Classificação	é_dada_como
é_uma_pergunta_do_tipo	Pergunta	Aberta or Fechada or Contraproducente	é_um_tipo_de
é_contraproducente_pois_é	Contraproducente	Indutiva, Ambígua, Discriminatória, Ardilosa, Retórica	é_contraproducente
remonta_uma	Fechada	Sim_ou_Não , Identificação	também_é_uma
tem_intencao	Aberta	De_Contato, Investigadora	é_uma
pode_ser	Investigadora	Reflexiva, Hipotética, Precisa, Fundamentada, Comparativa, Conclusiva, Extensiva, Simples	possui_uma_natureza
é_um_tipo_de	Aberta, Fechada, Contraproducente	Pergunta	é_uma_pergunta_do_tipo
também_é_uma	Sim_ou_Não , Identificação	Fechada	remonta_uma
é_contraproducente	Indutiva, Ambígua, Discriminatória, Ardilosa, Retórica	Contraproducente	é_contraproducente_pois_é
é_uma	De_Contato, Investigadora	Aberta	tem_intencao
possui_uma_natureza	Reflexiva, Hipotética, Precisa, Fundamentada, Comparativa, Conclusiva, Extensiva, Simples	Investigadora	pode_ser



Tamanho: pequena, média, grande ou muito grande

Contém erros gramaticais

Contém erros sintáticos

Contém erros semânticos

- Aberta

*Extrinsic properties:*

Permite divagações

- Fechada

*Extrinsic properties:*

Não Permite divagações

- De contato

*Extrinsic properties:*

Sua posição na ordem das perguntas

- Sim/Não

*Intrinsic properties:*

Aceita respostas SIM/NÃO

- Identificação

*Intrinsic properties:*

Testando o canal

Testando a linguagem

- Indutiva

*Intrinsic properties:*

Apresenta afirmações

Pede concordâncias

- Artilosa

*Intrinsic properties:*

Induz ao erro

Usa palavras ambíguas

Usa palavras rebuscadas

- Ambígua

*Intrinsic properties:*

Não utilização de pontuação

- Discriminatória

*Intrinsic properties:*

Utilização de palavras chave discriminatórias

- Fundamentadas

*Intrinsic properties:*

Uso de expressões que entonem pessoalidade, "como você", "seu ponto de vista",...

- Simples

*Intrinsic properties:*

Tamanho pequeno

Ocorrência de palavras chaves "Por que", "Como"

- Comparativas

*Intrinsic properties:*

Número de comparações

Comparações relacionadas ao tema

- Extensivas

*Intrinsic properties:*

Uso de perguntas anteriores para estender uma pergunta

- Precisas

*Intrinsic properties:*

Uso de perguntas anteriores para requisitar informações específicas

Uso de palavras chaves "precisamente", "especificamente", "francamente",...

- Hipotéticas

*Intrinsic properties:*

Uso de palavras chave "se", "hipoteticamente", "suponhamos"...

- Conclusivas

*Intrinsic properties:*

Usa um fato citado

#### 4.2.6 Passo 6, Definir as propriedades das propriedades

Para procurar manter a simplicidade do projeto e aumentar o re-uso, optamos por evitar definir as propriedades das propriedades.

## 4.2.7 Passo 7, Criar instâncias

As instâncias criadas serão introduzidas na seção 4.2.9, ao fazermos a revisão da ontologia.

## 4.2.8 Ontologia resultante no Protégé

Com tais aspectos definidos, usamos a ferramenta Protégé, para construir a ontologia e gerar um código OWL correspondente.

## 4.2.9 Revisão da Ontologia e do modelo OWL

Para fazer a revisão da ontologia, utilizamos os critérios de Gómez-Perez [11] anteriormente citados, fazendo verificações manuais e visuais. Como temos em mãos a ontologia criada usando o Protégé, utilizamos o *reasoner* padrão que vem instalado no Protégé, o HermiT 1.3.8.3, que efetua uma série de verificações (como foi descrito anteriormente no capítulo 3.2.4). O *reasoner* não indicou nenhum erro ou inconsistência.

Dessa forma revisamos a ontologia, tanto manualmente quanto automaticamente, o que nos indica uma ontologia consistente com o que propusemos.

## 4.2.10 Inserção de perguntas na ontologia e Inferências

Como temos por intenção conseguir classificar uma pergunta em boa ou ruim, nos utilizamos de uma possibilidade oferecida pelo nosso *reasoner*, no caso o HermiT, que a partir da ontologia faz inferências e é capaz de através dessas inferências, apontar classes as quais um indivíduo pertença, mesmo que elas não tenham sido explicitamente declaradas. No nosso caso, ao inserirmos um indivíduo (no caso uma pergunta) na ontologia, e declarando a quais classes este indivíduo pertence (ou seja, indicando se a pergunta é investigadora, simples, ardilosa e etc.), o HermiT inferirá se o indivíduo pertence a classe Boa ou a classe Ruim (ou seja, inferirá se a pergunta é boa ou ruim).

Estabelecemos então algumas perguntas e verificamos como elas se comportam no modelo criado. Utilizamos dois tipos de perguntas, as idealizadas propriamente para os nossos testes e as perguntas criadas pelos alunos, no decorrer da disciplina de Estrutura de Dados ministrada na Universidade de Brasília no segundo semestre de 2015. As idealizadas propriamente para os nossos testes foram:

- Pergunta 1: Como você vê a atual situação do Oriente Médio?
- Pergunta 2: Você viu o ponto quando entrou na sala?
- Pergunta 3: Olá, tudo bem?
- Pergunta 4: Hipoteticamente, como você reagiria?
- Pergunta 5: Qual sistema operacional é melhor, IOS ou Android?
- Pergunta 6: O que você quis dizer com a sua última resposta?
- Pergunta 7: O que você quis dizer com isso em particular?
- Pergunta 8: Para você, como andam os rumos da economia?

- Pergunta 9: Por quê?
- Pergunta 10: Por quanto tempo você trabalhou lá?
- Pergunta 11: Você é canhoto?
- Pergunta 12: Quais são os seus principais defeitos?
- Pergunta 13: Você possui problemas com álcool?
- Pergunta 14: Você não gosta de computadores, certo?
- Pergunta 15: Nós queremos mesmo pintar a parede?
- Pergunta 16: Se eu entendi direito, tudo seria diferente, não?

Tais perguntas podem ser classificadas (levando em conta a sua principal característica) como indica a tabela 4.4.

Tabela 4.4: Classificação das perguntas inseridas

Classificação como Boa ou Ruim	Pergunta	Identificada como
Boa	Como você vê a atual situação do Oriente Médio?	Fundamentada
Ruim	Você viu o ponto quando entrou na sala?	Ambígua
Boa	Olá, tudo bem?	De contato
Boa	Hipoteticamente, como você reagiria?	Hipotética
Boa	Qual sistema operacional é melhor, IOS ou Android?	Comparativa
Boa	O que você quis dizer com a sua última resposta?	Extensiva
Boa	O que você quis dizer com isso em particular?	Precisa
Boa	Para você, como andam os rumos da economia?	Reflexiva
Boa	Por quê?	Simple
Boa	Por quanto tempo você trabalhou lá?	Identificação
Boa	Você é canhoto?	Sim ou Não
Ruim	Quais são os seus principais defeitos?	Ardilosa
Ruim	Você possui problemas com álcool?	Discriminatória
Ruim	Você não gosta de computadores, certo?	Indutiva
Ruim	Nós queremos mesmo pintar a parede?	Retórica
Boa	Se eu entendi direito, tudo seria diferente, não?	Conclusiva

E as perguntas criadas pelos alunos no decorrer da disciplina de Estrutura de Dados, conforme a tabela 4.5.

A ontologia proposta inferiu de maneira correta todos os resultados desejados em todas as perguntas inseridas, apresentando resultados como ilustrados nas figuras 4.6 e 4.5, onde temos um exemplo de pergunta classificada pelo *reasoner* como boa e outra como ruim.

### 4.3 Integração ao *quizWiki*

Atualmente o *quizWiki* usa a versão 1.16.4 do *MediaWiki*, porém, existe a intenção (e a necessidade) de migrá-lo para uma versão mais recente. Por estes motivos escolhemos e instalamos a versão 1.23.5 do mesmo, para permitir a inserção da ontologia.

Tabela 4.5: Classificação das perguntas desenvolvidas pelos alunos

Classificação como Boa ou Ruim	Pergunta	Identificada como
Boa	Como seria o uso de uma árvore para decodificação de código Morse?	Fundamentada
Ruim	A calculadora polonesa seria melhor implementada com árvore?	Ambígua
Boa	Não ocorreu	De contato a
Boa	É possível criar 2 árvores podendo haver um nó em comum entre eles?	Hipotética
Boa	Qual ganho se tem em implementar uma calculadora usando uma árvore binária em vez de usar uma pilha?	Comparativa
Boa	Se não, quais problemas possíveis que possam ocorrer?	Extensiva
Boa	Como usar uma árvore para implementar a função <i>hash</i> (mapeamento de dados)?	Precisa
Boa	Seria possível fazer a função <i>insere nó</i> de uma maneira não recursiva?	Reflexiva
Boa	Como faço uma calculadora em C usando árvore binária?	Simples
Boa	Não ocorreu	Identificação
Boa	Existe alguma aplicação em que realizar travessia na árvore por nível seja a melhor opção?	Sim ou Não
Ruim	A árvore binária realmente é o modelo de árvore de melhor rendimento computacional ?	Ardilosa
Ruim	Não ocorreu	Discriminatória
Ruim	Ao executar a remoção de um elemento estaremos comprometendo todos os seus filhos e os respectivos filhos desses filhos? Se sim, qual seria a perda	Indutiva
Ruim	Não ocorreu	Retórica
Boa	Não ocorreu	Conclusiva

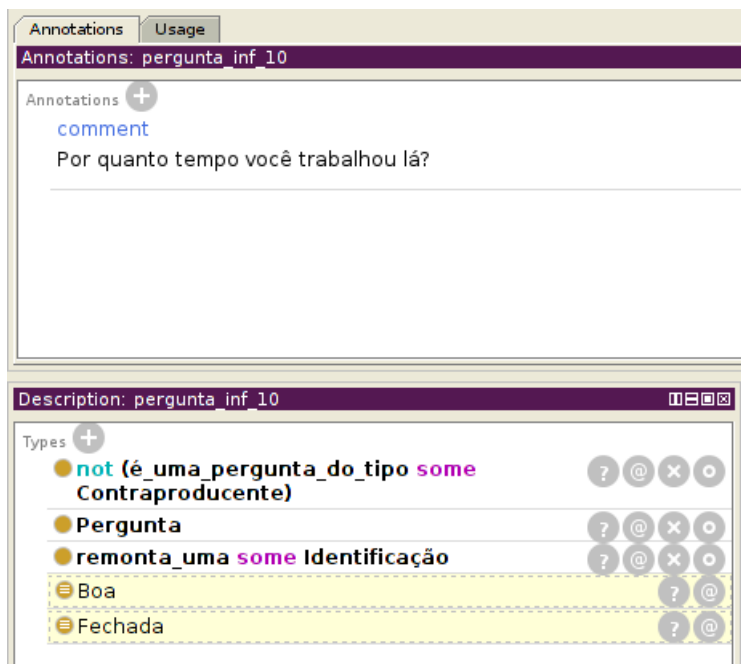


Figura 4.5: Exemplo de pergunta classificada como boa



Figura 4.6: Exemplo de pergunta classificada como ruim

Utilizamos também o *PHP* na versão *5.5.9-1ubuntu4.5 (apache2handler)* e o *MySQL* *5.5.41-0ubuntu0.14.04.1*. A partir dele, com o *MediaWiki* funcionando, instalamos a versão 2.2 da extensão *Semantic MediaWiki*, como mostra a Figura 4.7.

Com o *MediaWiki* funcionando, apoiado pela extensão semântica, passamos a ontologia gerada em *OWL* para a instalação local do *MediaWiki*. Infelizmente, o *Semantic MediaWiki* não dispõe de ferramentas nativas para a importação de ontologias [29], isso

Software instalado	
Produto	Versão
MediaWiki	1.23.5
PHP	5.5.9-1ubuntu4.5 (apache2handler)
MySQL	5.5.41-0ubuntu0.14.04.1

URLs dos pontos de entrada	
Ponto de entrada	URL
Article path	/wiki/index.php/\$1
Script path	/wiki
index.php	/wiki/index.php
api.php	/wiki/api.php
load.php	/wiki/load.php

Extensões instaladas	
Extensão	Versão
Semantic MediaWiki	2.2 alpha (7d1792c) 20h45min de 22 de janeiro de 2015

Figura 4.7: Versões instaladas localmente

poderia ser feito utilizando a extensão *SMW+*, mas a mesma está disponível apenas para o ambiente *Windows* (o que não é o caso do projeto *quizWiki*) e teve o seu projeto descontinuado (implicando em falta de material de apoio) o que nos impediu de utilizá-la.

Poderíamos fazer a transposição manual, mas decidimos criar uma forma mais inteligente e automatizada de, fazer pelo menos parte do processo. Para isso, utilizamos uma ferramenta para criar automaticamente as páginas na nossa *wiki* local, no caso o *Pywikibot* [21]. Primeiramente, tendo em mãos o código *OWL* gerado pela ontologia no *Protégé*, desenvolvemos um programa em *Java*, que retira as *tags* provenientes do *OWL* e formata todo o conteúdo, na forma de páginas *wiki* e as salva em um arquivo *.txt*. Feito isto, instalamos a versão core do *Pywikibot* e a configuramos para o nosso ambiente (apontando-o para a nossa versão local da *wiki* e fornecendo credenciais que tenham permissões para criar páginas), depois utilizamos um *script* nativo, chamado *pagefromfile.py*, que a partir de um arquivo *.txt*, cria automaticamente, as páginas discriminadas, com seus respectivos conteúdos.

Por fim, invocamos o *Pywikibot* com o seguinte comando:

```
1 python pagefromfile.py [global-arguments] -start:xxxx -end:yyyy -file :
   saidaJava.txt
```

Dessa forma, encontramos na *wiki* local, a inserção de diversas novas páginas (Figura 4.9), contendo a ontologia transposta, e contendo (ou não) anotações semânticas em

suas páginas (conforme as figuras 4.10 e 4.11). Todo este processo pode ser resumido visualmente conforme indica a Figura 4.8.

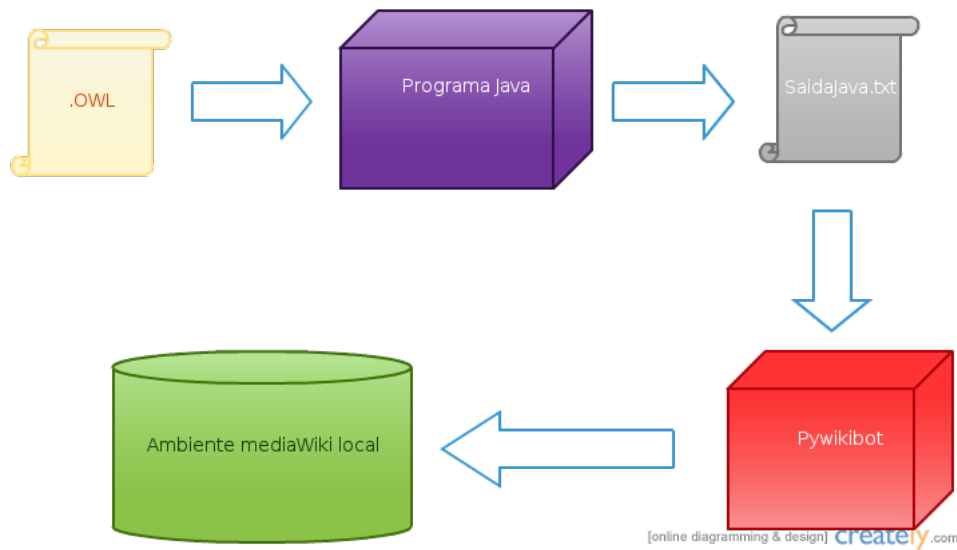


Figura 4.8: Processo de importação da ontologia

Foram criadas também, por parte do Semantic MediaWiki, as páginas referentes às categorias (Figura 4.12) e as propriedades criadas (Figura 4.13).

Página especial Pesquis

## Mudanças recentes

Acompanhe nesta página as mudanças mais recentes deste wiki.

Opções das mudanças recentes

Exibir as **50** | 100 | 250 | 500 alterações recentes feitas nos últimos 1 | 3 | 7 | 14 | 30 dias  
 Ocultar edições menores | Exibir robôs | Ocultar usuários anônimos | Ocultar usuários registrados | Ocultar minhas edições  
 Mostrar as novas alterações a partir das 22h49min de 3 de março de 2015

Espaço **todos**  Inverter seleção  Espaço nominal  
 nominal: associado

**Legenda:**  
**N** Esta edição criou uma nova página (veja també  
**m** Esta é uma edição menor  
**b** Esta edição foi feita por um robô  
**(±123)** Alteração no tamanho da página, em bytes

**3 de março de 2015**

- (dif | his) .. **N** Property:É um tipo de; 22h42min .. (+30) .. **Victor** (Discussão | contribs) (Bot: Automated import of articles)
- (dif | his) .. **N** Property:É uma pergunta do tipo; 22h41min .. (+40) .. **Victor** (Discussão | contribs) (Bot: Automated import of articles)
- (dif | his) .. **N** Property:É contraproducente pois é; 22h41min .. (+44) .. **Victor** (Discussão | contribs) (Bot: Automated import of articles)
- (dif | his) .. **N** Simples; 22h41min .. (+441) .. **Victor** (Discussão | contribs) (Bot: Automated import of articles)
- (dif | his) .. **N** Sim ou Não; 22h41min .. (+416) .. **Victor** (Discussão | contribs) (Bot: Automated import of articles)
- (dif | his) .. **N** Ruim; 22h41min .. (+390) .. **Victor** (Discussão | contribs) (Bot: Automated import of articles)
- (dif | his) .. **N** Retórica; 22h40min .. (+431) .. **Victor** (Discussão | contribs) (Bot: Automated import of articles)
- (dif | his) .. **N** Baqueta; 22h40min .. (+262) .. **Victor** (Discussão | contribs) (Bot: Automated import of articles)

Figura 4.9: Algumas das novas páginas criadas na *wiki*



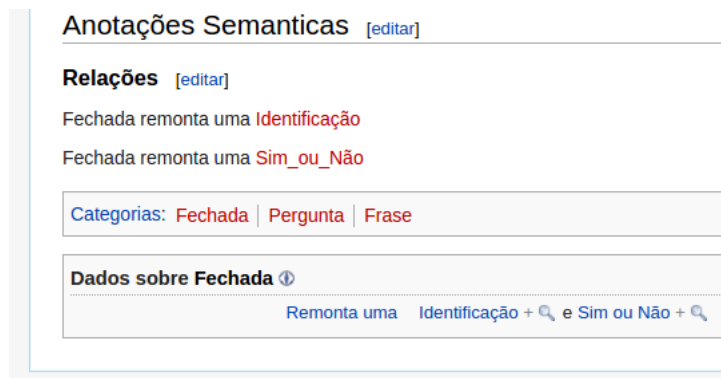


Figura 4.10: Anotações semânticas como são vistas pelo usuário

```

== Anotações Semânticas ==
=== Relações ===
Fechada remonta uma [[remonta_uma::Identificação]]
Fechada remonta uma [[remonta_uma::Sim_ou_Não]]
[[Category:Fechada]]
[[Category:Pergunta]]
[[Category:Frase]]

```

Figura 4.11: Visualização do código das anotações semânticas

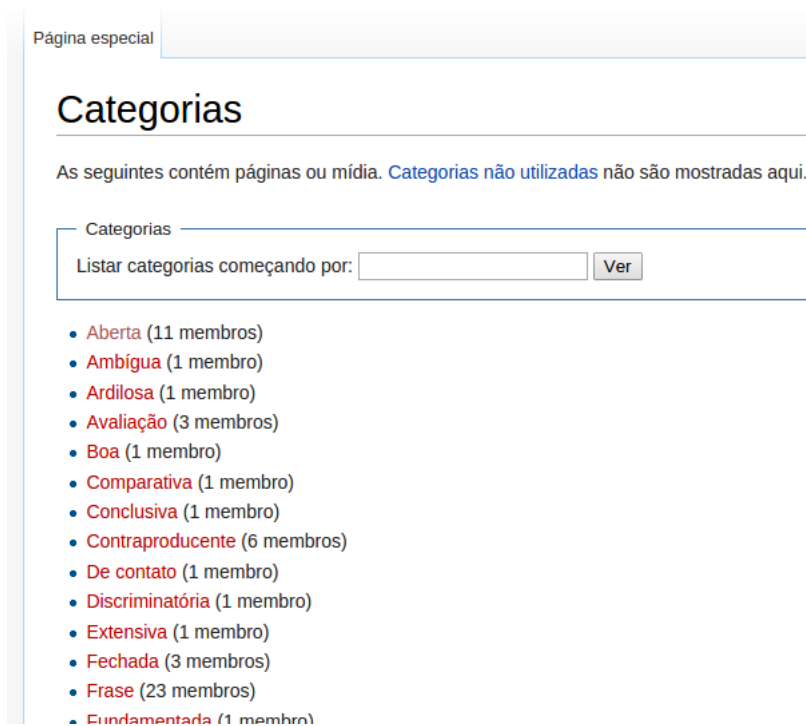


Figura 4.12: Categorias criadas a partir das anotações semânticas



Figura 4.13: Propriedades criadas a partir das anotações semânticas

## 4.4 Cenários hipotéticos de utilização

Primeiramente podemos estabelecer alguns casos de uso onde o trabalho desenvolvido pode ser inserido:

- Ao se começar o curso, o professor pede aos alunos que façam uma pesquisa sobre: "O que é uma boa pergunta" (ou algo do gênero), e pede que eles respondam utilizando o *quizWiki* como fonte de pesquisa. Com isso os alunos fazem a pesquisa e a partir deste momento, sempre que forem desenvolver perguntas no *quiz* ou avaliar as perguntas dos colegas, saberão onde encontrar uma fonte de conhecimento sobre o assunto, conforme ilustra a Figura 4.14.
- Ao receber o *feedback* do sistema através de sua "reputação", o aluno pode pesquisar dentro do *quizWiki* a fim de entender a causa do seu desempenho (seja ele bom ou ruim) e assim melhorar suas capacidades como estudante, conforme ilustra a Figura 4.15.

Desenvolvemos também, algum conhecimento tanto na forma de discussões, quanto na de objetos (ontologia, representação em OWL e diretamente na *wiki*). Quanto as discussões, fica estabelecida uma base de conhecimentos para utilização em cenários educacionais, levantando a bandeira do aprendizado educacional através da elaboração de questões e a possibilidade de modelar tais questões na forma de uma ontologia.

No que diz respeito a ontologia propriamente dita, ela pode ser utilizada como fonte para um motor de inferências, para classificar perguntas. A sua versão no formato OWL

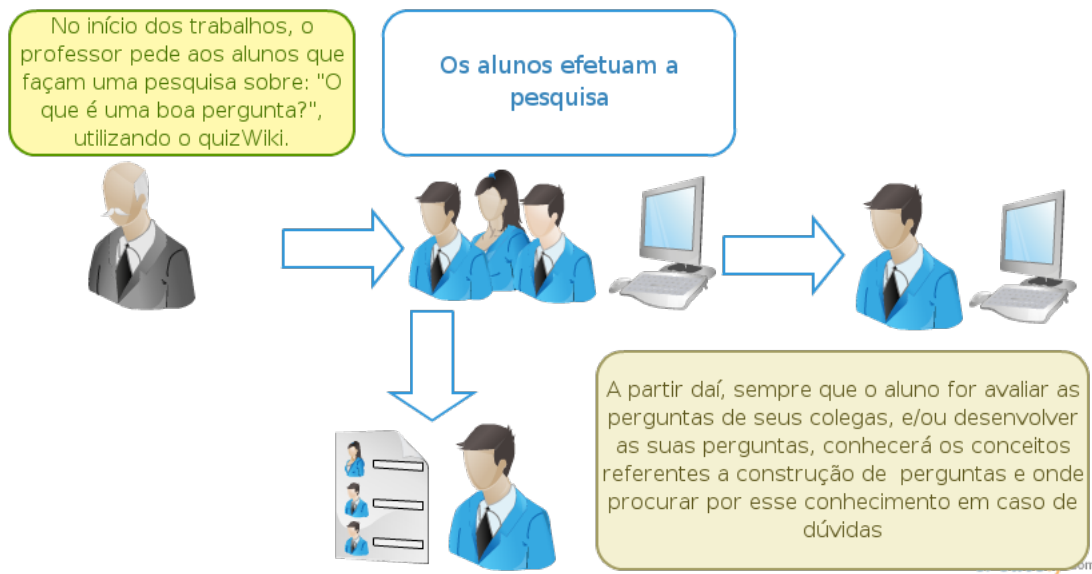


Figura 4.14: Caso de uso

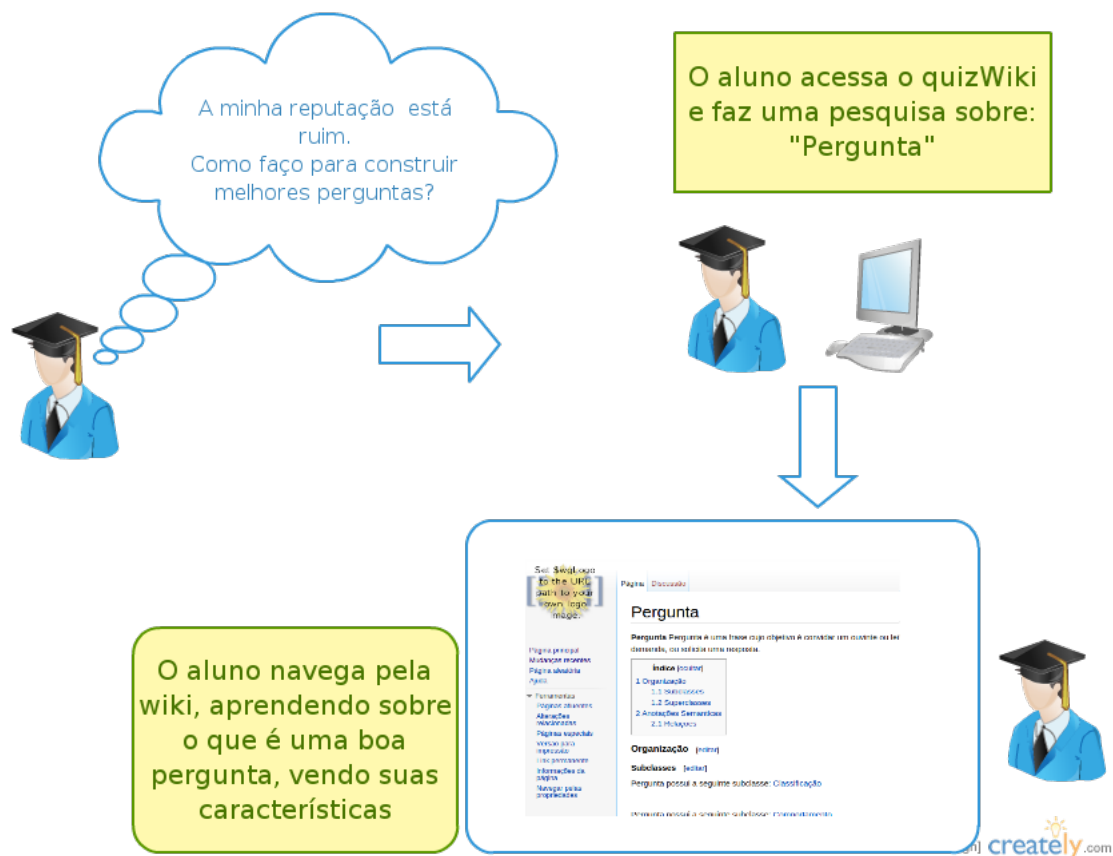


Figura 4.15: Caso de uso

fica como uma fonte para conversões para os diversas outros formatos disponíveis (Manchester OWL Syntax, RDF/XML e etc.) e suas respectivas ferramentas. A sua versão

exportada para o ambiente *wiki* pode ser utilizada no contexto educacional como um suporte ao aluno, servindo de material de consulta no *quizWiki*, auxiliando o mesmo tanto a entender em parte as notas que o mesmo tem recebido por suas participações no *quiz*, quanto ajudá-lo a verificar e aprender as formas corretas de construir uma pergunta.

# Capítulo 5

## Conclusão

### 5.1 Considerações finais

Neste trabalho, mostramos que a construção de uma ontologia sobre o questionamento, que seja capaz de classificar uma pergunta como boa ou ruim, é possível, como mostramos nos resultados. Notamos ainda que a construção da ontologia, segue critérios e toma caminhos que variam de acordo com a pessoa que efetua a construção, logo ela pode não ser uma unanimidade (quanto mais complexo o tema, maior a probabilidade de isso acontecer), mas apresenta uma representação válida dentro do que foi proposto.

As principais dificuldades encontradas nessa construção foram de cunho teórico, pois há dificuldades em trabalhar com um tema tão abstrato quanto a avaliação de questões, assim como também é complexo encontrar um vocabulário que expresse com exatidão as relações envolvendo as classes encontradas. A extração do conhecimento feita a partir de uma fonte teórica foi essencial, mesmo sendo uma fonte voltada para o questionamento visando entrevistas, conseguimos extrair o conhecimento de forma bastante satisfatória.

O objetivo de inserir a ontologia criada no ambiente do *MediaWiki*, também foi alcançado, mostrando uma real aplicação do desenvolvimento da ontologia. Tal inserção contou com a instalação do ambiente *MediaWiki* mais atual (que é um desafio de migração dentro do projeto *quizWiki*). Para isso temos como legado uma série de *scripts* e experiências nesse sentido. Desenvolvemos também uma forma de importar ontologias para o *MediaWiki*, utilizando um programa *Java* aliado ao *PyWikibot*.

Notamos que a classificação feita neste trabalho leva em conta principalmente as questões estruturais das perguntas, muitas vezes não sendo capaz de inferir sobre o conteúdo das mesmas. Tal problema incorre na classificação errônea de perguntas ruins (por conta de diversos fatores, como grau de dificuldade, respostas muito grandes, e etc.) como boas perguntas (o inverso não ocorre). Para resolver tal problema levantamos algumas hipóteses, como a de passar a considerar também as respostas relativas a cada pergunta, assim como a possibilidade de em ontologias mais específicas, ou esquemas que incluam algum tipo de conhecimento extra ao sistema. Tais aspectos tão somente nos deixam a certeza da necessidade de aprofundar em pesquisas como esta, procurando tanto melhorar o ensino através da utilização de suportes computacionais, quanto da criação de modelos teóricos que possam representar e inferir sobre temas ligados a área educacional.

## 5.2 Trabalhos futuros

A partir deste trabalho podemos sugerir uma série de projetos que podem ser desenvolvidos futuramente, tendo o mesmo como ponto de partida e/ou referência:

- Refinamento da ontologia: a partir da verificação do comportamento da ontologia, perante as questões levantadas com o decorrer do tempo, acrescentar classes, relações e/ou se alterar sua estrutura, aumentando sua acurácia em direção a uma ontologia que represente melhor as perguntas;
- Expansão da ontologia para tratar as respostas, e posteriormente os diferentes tipos de frases.
- Procurar outras fontes de extração da ontologia, e modelar ontologias alternativas, possivelmente mesclando-as com a que é fruto deste trabalho;
- Utilizar a ontologia como matriz de um motor de inferências, podendo assim classificar as perguntas e utilizar tal objeto dentro do *quizWiki*. Tal programa funcionaria como uma *persona* dentro do *quizWiki*, fornecendo uma nota a ser considerada para cada pergunta submetida pelo aluno. Essa nota entraria na equação que calcula a reputação do aluno e ainda seria fonte de aperfeiçoamento, tanto para os avaliadores quanto para os alunos (fornecendo uma explicação lógica, baseada na ontologia para a nota auferida);
- Verificar a validade da ontologia para perguntas efetuadas em outras línguas: tendo em vista as diferenças entre as diversas línguas (Português, Inglês, Espanhol, Chinês e etc.) e as diversas famílias linguísticas (índo-europeia, sino-tibetana, afro-asiática e etc.) verificamos que as estruturas usadas nas perguntas podem variar, assim como significados e formas. Logo a ontologia criada, que funciona bem para a língua portuguesa, provavelmente vai se comportar bem ao ser testada com a língua inglesa (visto que, a ontologia foi baseada em uma extração feita a partir de um livro escrito na língua inglesa [19]) mas nada garante que ela seja precisa ao tratar outras línguas pouco semelhantes;
- Criar ontologias que sejam capazes de classificar perguntas dentro de áreas do conhecimento mais específicas, como computação, artes, geografia e etc..

# Referências

- [1] Mauricio B Almeida and Marcello P Bax. Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção. *Ciência da Informação, Brasília*, 32(3):7–20, 2003. 11, 30
- [2] Cristhianny Barreiro. Questionamento sistemático: alicerce na reconstrução dos conhecimentos. *Pesquisa em sala de aula: tendências para a educação em novos tempos. Porto Alegre: EDIPUCRS*, pages 171–188, 2002. 8
- [3] Sean Bechhofer, F v Harmelen, Jim Hendler, Ian Horrocks, Deborah McGuinness, Peter Patel-Schneider, and L Stein. Owl web ontology language reference, feb. 2004. *World Wide Web Consortium*, 2005. 14
- [4] Ig Bittencourt, Camila Bezerra, Camila Nunes, Evandro Costa, Marcos Tadeu, Rômulo Nunes, Marcos Costa, and Alan Silva. Ontologia para construção de ambientes interativos de aprendizagem. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 1, pages 567–576, 2006. 12
- [5] Simon Blackburn and Desidério Murcho. *Dicionário Oxford de filosofia*. Zahar, 1997. 11
- [6] Angelo V Ciardiello. Did you ask a good question today? alternative cognitive and metacognitive strategies. *Journal of adolescent & adult literacy*, pages 210–219, 1998. 8
- [7] Germana M da Nóbrega and Fernanda Lima. Incorporando o questionamento discente no processo de aprendizagem em comunidades on-line sustentáveis. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 1, 2011. 5, 19, 20
- [8] Maria Betânia Dantas de Souza and Marise Adriana Mamede Galvão. Considerações acerca de funções de perguntas e respostas em sala de aula. 8
- [9] Anja Ebersbach, Andrea Adelung, Gunter Dueck, Markus Glaser, Richard Heigl, and Alexander Warta. *Wiki: web collaboration*. Springer, 2008. 19
- [10] John H Gennari, Mark A Musen, Ray W Ferguson, William E Grosso, Monica Crubézy, Henrik Eriksson, Natalya F Noy, and Samson W Tu. The evolution of protégé: an environment for knowledge-based systems development. *International Journal of Human-computer studies*, 58(1):89–123, 2003. 15, 16
- [11] Asunción Gómez-Pérez. Evaluation of taxonomic knowledge in ontologies and knowledge bases. 1999. 12, 30, 40

- [12] Arthur Graesser and Natalie Person. Question asking during tutoring. *American educational research journal*, 31(1):104–137, 1994. 9
- [13] Thomas R Gruber. *Ontolingua: A mechanism to support portable ontologies*. Stanford University, Knowledge Systems Laboratory, 1992. 27
- [14] Tom Gruber. Definição de ontologia. <http://tomgruber.org/writing/ontology-definition-2007.htm>, 2014. [Online; acessado 25-Agosto-2014]. 12
- [15] Giancarlo Guizzardi. Desenvolvimento para e com reuso: Um estudo de caso no domínio de vídeo sob demanda. *Universidade Federal do Espírito Santo, Centro Tecnológico*, 2000. 11
- [16] Mark Guzdial. Use of collaborative multimedia in computer science classes. In *ACM SIGCSE Bulletin*, volume 33, pages 17–20. ACM, 2001. 7, 19
- [17] Markus Krötzsch, Denny Vrandečić, and Max Völkel. Semantic mediawiki. 23
- [18] Ora Lassila and Deborah McGuinness. The role of frame-based representation on the semantic web. *Linköping Electronic Articles in Computer and Information Science*, 6(5):2001, 2001. 13
- [19] Ian Mackay. *Aprendendo a perguntar*. NBL Editora, 2001. 1, 26, 32, 51
- [20] Deborah L McGuinness, Frank Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(2004-03):10, 2004. 14, 27
- [21] MediaWiki. Manual:pywikibot/pt-br — mediawiki, the free wiki engine. <http://www.mediawiki.org/w/index.php?title=Manual:Pywikibot/pt-br&oldid=1273517>, 2014. [Online; acessado 25-Janeiro-2015]. 44
- [22] MediaWiki. Mediawiki — mediawiki, the free wiki engine. <http://www.mediawiki.org/w/index.php?title=MediaWiki&oldid=1083211>, 2014. [Online; acessado 14-Fevereiro-2015]. 20
- [23] Germana Nóbrega, Fernanda Lima, and Dayane Freire. Integrating the semantic wiki approach to face to face courses. In *World Conference on Computers in Education*, 2009. 19
- [24] Natalya F Noy, Deborah L McGuinness, et al. Ontology development 101: A guide to creating your first ontology, 2001. 27
- [25] Alice Theresinha Cybis Pereira, Valdenise Schmitt, and MRAC Dias. Ambientes virtuais de aprendizagem. *Ambientes Virtuais de Aprendizagem: em diferentes contextos*. Rio de Janeiro: Ciência Moderna Ltda, pages 2–22, 2007. 9
- [26] Protégé. Protegedesktopuserdocs - protege wiki. <http://protegewiki.stanford.edu/wiki/Protege4UserDocs>, 2014. [Online; acessado 14-Fevereiro-2015]. 18
- [27] Rob Shearer, Boris Motik, and Ian Horrocks. Hermit: A highly-efficient owl reasoner. In *OWLED*, volume 432, 2008. 18



- [28] Tania Tudorache, Csongor Nyulas, Natalya F Noy, and Mark A Musen. Webprotégé: A collaborative ontology editor and knowledge acquisition tool for the web. *Semantic web*, 4(1):89–99, 2013. 17
- [29] Denny Vrandečić and Markus Krötzsch. Reusing ontological background knowledge in semantic wikis. *SemWiki*, 206, 2006. 43
- [30] Yao Wang and Julita Vassileva. Trust and reputation model in peer-to-peer networks. 20
- [31] Michael Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009. vii, 13, 14, 15, 16, 28, 29

# Apêndice A

## Modelo OWL criado

Aqui temos o modelo da ontologia em owl:

```
1 <?xml version="1.0" ?>
2
3
4 <!DOCTYPE Ontology [
5   <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
6   <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
7   <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
8   <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
9 ]>
10
11
12 <Ontology xmlns="http://www.w3.org/2002/07/owl#"
13   xml:base="http://www.semanticweb.org/victor/ontologies/2015/0/untitled
14   -ontology-9"
15   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
16   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
17   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
18   xmlns:xml="http://www.w3.org/XML/1998/namespace"
19   ontologyIRI="http://www.semanticweb.org/victor/ontologies/2015/0/
20   untitled-ontology-9">
21   <Prefix name="" IRI="http://www.semanticweb.org/victor/ontologies
22   /2015/0/untitled-ontology-9#" />
23   <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#" />
24   <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
25   <Prefix name="xml" IRI="http://www.w3.org/XML/1998/namespace" />
26   <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#" />
27   <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#" />
28   <Declaration>
29     <Class IRI="#Aberta" />
30   </Declaration>
31   <Declaration>
32     <Class IRI="#Ambígua" />
33   </Declaration>
34   <Declaration>
35     <Class IRI="#Ardilosa" />
36   </Declaration>
37   <Declaration>
38     <Class IRI="#Avaliação" />
39   </Declaration>
```

```
37 <Declaration>
38   <Class IRI="#Boa" />
39 </Declaration>
40 <Declaration>
41   <Class IRI="#Comparativa" />
42 </Declaration>
43 <Declaration>
44   <Class IRI="#Conclusiva" />
45 </Declaration>
46 <Declaration>
47   <Class IRI="#Contraproducente" />
48 </Declaration>
49 <Declaration>
50   <Class IRI="#De_contato" />
51 </Declaration>
52 <Declaration>
53   <Class IRI="#Discriminatória" />
54 </Declaration>
55 <Declaration>
56   <Class IRI="#Extensiva" />
57 </Declaration>
58 <Declaration>
59   <Class IRI="#Fechada" />
60 </Declaration>
61 <Declaration>
62   <Class IRI="#Frase" />
63 </Declaration>
64 <Declaration>
65   <Class IRI="#Fundamentada" />
66 </Declaration>
67 <Declaration>
68   <Class IRI="#Hipotética" />
69 </Declaration>
70 <Declaration>
71   <Class IRI="#Identificação" />
72 </Declaration>
73 <Declaration>
74   <Class IRI="#Indutiva" />
75 </Declaration>
76 <Declaration>
77   <Class IRI="#Investigadora" />
78 </Declaration>
79 <Declaration>
80   <Class IRI="#Pergunta" />
81 </Declaration>
82 <Declaration>
83   <Class IRI="#Precisa" />
84 </Declaration>
85 <Declaration>
86   <Class IRI="#Reflexiva" />
87 </Declaration>
88 <Declaration>
89   <Class IRI="#Resposta" />
90 </Declaration>
91 <Declaration>
92   <Class IRI="#Retórica" />
```

```

93 </Declaration>
94 <Declaration>
95   <Class IRI="#Ruim" />
96 </Declaration>
97 <Declaration>
98   <Class IRI="#Sim_ou_Não" />
99 </Declaration>
100 <Declaration>
101   <Class IRI="#Simples" />
102 </Declaration>
103 <Declaration>
104   <ObjectProperty IRI="#possui_uma_natureza" />
105 </Declaration>
106 <Declaration>
107   <ObjectProperty IRI="#remonta_uma" />
108 </Declaration>
109 <Declaration>
110   <ObjectProperty IRI="#sofre_uma" />
111 </Declaration>
112 <Declaration>
113   <ObjectProperty IRI="#também_é_uma" />
114 </Declaration>
115 <Declaration>
116   <ObjectProperty IRI="#tem_intencao" />
117 </Declaration>
118 <Declaration>
119   <ObjectProperty IRI="#é_aplicada_sobre_uma" />
120 </Declaration>
121 <Declaration>
122   <ObjectProperty IRI="#é_contraproducente" />
123 </Declaration>
124 <Declaration>
125   <ObjectProperty IRI="#é_contraproducente_pois_é" />
126 </Declaration>
127 <Declaration>
128   <ObjectProperty IRI="#é_dada_como" />
129 </Declaration>
130 <Declaration>
131   <ObjectProperty IRI="#é_fruto_de_uma" />
132 </Declaration>
133 <Declaration>
134   <ObjectProperty IRI="#é_um_tipo_de" />
135 </Declaration>
136 <Declaration>
137   <ObjectProperty IRI="#é_uma" />
138 </Declaration>
139 <Declaration>
140   <ObjectProperty IRI="#é_uma_pergunta_do_tipo" />
141 </Declaration>
142 <Declaration>
143   <ObjectProperty IRI="#estrutura_aberta ,pode_ser" />
144 </Declaration>
145 <Declaration>
146   <NamedIndividual IRI="#pergunta_col_01" />
147 </Declaration>
148 <Declaration>

```

```

149     <NamedIndividual IRI="#pergunta_col_02"/>
150 </Declaration>
151 <Declaration>
152     <NamedIndividual IRI="#pergunta_col_04"/>
153 </Declaration>
154 <Declaration>
155     <NamedIndividual IRI="#pergunta_col_05"/>
156 </Declaration>
157 <Declaration>
158     <NamedIndividual IRI="#pergunta_col_06"/>
159 </Declaration>
160 <Declaration>
161     <NamedIndividual IRI="#pergunta_col_07"/>
162 </Declaration>
163 <Declaration>
164     <NamedIndividual IRI="#pergunta_col_08"/>
165 </Declaration>
166 <Declaration>
167     <NamedIndividual IRI="#pergunta_col_09"/>
168 </Declaration>
169 <Declaration>
170     <NamedIndividual IRI="#pergunta_col_11"/>
171 </Declaration>
172 <Declaration>
173     <NamedIndividual IRI="#pergunta_col_12"/>
174 </Declaration>
175 <Declaration>
176     <NamedIndividual IRI="#pergunta_col_14"/>
177 </Declaration>
178 <Declaration>
179     <NamedIndividual IRI="#pergunta_inf_01"/>
180 </Declaration>
181 <Declaration>
182     <NamedIndividual IRI="#pergunta_inf_02"/>
183 </Declaration>
184 <Declaration>
185     <NamedIndividual IRI="#pergunta_inf_03"/>
186 </Declaration>
187 <Declaration>
188     <NamedIndividual IRI="#pergunta_inf_04"/>
189 </Declaration>
190 <Declaration>
191     <NamedIndividual IRI="#pergunta_inf_05"/>
192 </Declaration>
193 <Declaration>
194     <NamedIndividual IRI="#pergunta_inf_06"/>
195 </Declaration>
196 <Declaration>
197     <NamedIndividual IRI="#pergunta_inf_07"/>
198 </Declaration>
199 <Declaration>
200     <NamedIndividual IRI="#pergunta_inf_08"/>
201 </Declaration>
202 <Declaration>
203     <NamedIndividual IRI="#pergunta_inf_09"/>
204 </Declaration>

```

```

205 <Declaration>
206   <NamedIndividual IRI="#pergunta_inf_10" />
207 </Declaration>
208 <Declaration>
209   <NamedIndividual IRI="#pergunta_inf_11" />
210 </Declaration>
211 <Declaration>
212   <NamedIndividual IRI="#pergunta_inf_12" />
213 </Declaration>
214 <Declaration>
215   <NamedIndividual IRI="#pergunta_inf_13" />
216 </Declaration>
217 <Declaration>
218   <NamedIndividual IRI="#pergunta_inf_14" />
219 </Declaration>
220 <Declaration>
221   <NamedIndividual IRI="#pergunta_inf_15" />
222 </Declaration>
223 <Declaration>
224   <NamedIndividual IRI="#pergunta_inf_16" />
225 </Declaration>
226 <EquivalentClasses>
227   <Class IRI="#Aberta" />
228   <ObjectComplementOf>
229     <Class IRI="#Fechada" />
230   </ObjectComplementOf>
231 </EquivalentClasses>
232 <EquivalentClasses>
233   <Class IRI="#Boa" />
234   <ObjectComplementOf>
235     <Class IRI="#Ruim" />
236   </ObjectComplementOf>
237 </EquivalentClasses>
238 <EquivalentClasses>
239   <Class IRI="#Fechada" />
240   <ObjectComplementOf>
241     <Class IRI="#Aberta" />
242   </ObjectComplementOf>
243 </EquivalentClasses>
244 <EquivalentClasses>
245   <Class IRI="#Ruim" />
246   <ObjectComplementOf>
247     <Class IRI="#Boa" />
248   </ObjectComplementOf>
249 </EquivalentClasses>
250 <EquivalentClasses>
251   <Class IRI="#Ruim" />
252   <ObjectSomeValuesFrom>
253     <ObjectProperty IRI="#é_uma_pergunta_do_tipo" />
254     <Class IRI="#Contraproducente" />
255   </ObjectSomeValuesFrom>
256 </EquivalentClasses>
257 <SubClassOf>
258   <Class IRI="#Aberta" />
259   <Class IRI="#Pergunta" />
260 </SubClassOf>

```

```

261 <SubClassOf>
262   <Class IRI="#Ambígua" />
263   <Class IRI="#Contraproducente" />
264 </SubClassOf>
265 <SubClassOf>
266   <Class IRI="#Ardilosa" />
267   <Class IRI="#Contraproducente" />
268 </SubClassOf>
269 <SubClassOf>
270   <Class IRI="#Boa" />
271   <Class IRI="#Avaliação" />
272 </SubClassOf>
273 <SubClassOf>
274   <Class IRI="#Comparativa" />
275   <Class IRI="#Investigadora" />
276 </SubClassOf>
277 <SubClassOf>
278   <Class IRI="#Conclusiva" />
279   <Class IRI="#Investigadora" />
280 </SubClassOf>
281 <SubClassOf>
282   <Class IRI="#Contraproducente" />
283   <Class IRI="#Pergunta" />
284 </SubClassOf>
285 <SubClassOf>
286   <Class IRI="#De_contato" />
287   <Class IRI="#Aberta" />
288 </SubClassOf>
289 <SubClassOf>
290   <Class IRI="#Discriminatória" />
291   <Class IRI="#Contraproducente" />
292 </SubClassOf>
293 <SubClassOf>
294   <Class IRI="#Extensiva" />
295   <Class IRI="#Investigadora" />
296 </SubClassOf>
297 <SubClassOf>
298   <Class IRI="#Fechada" />
299   <Class IRI="#Pergunta" />
300 </SubClassOf>
301 <SubClassOf>
302   <Class IRI="#Fundamentada" />
303   <Class IRI="#Investigadora" />
304 </SubClassOf>
305 <SubClassOf>
306   <Class IRI="#Hipotética" />
307   <Class IRI="#Investigadora" />
308 </SubClassOf>
309 <SubClassOf>
310   <Class IRI="#Identificação" />
311   <Class IRI="#Fechada" />
312 </SubClassOf>
313 <SubClassOf>
314   <Class IRI="#Indutiva" />
315   <Class IRI="#Contraproducente" />
316 </SubClassOf>

```

```

317 <SubClassOf>
318   <Class IRI="#Investigadora" />
319   <Class IRI="#Aberta" />
320 </SubClassOf>
321 <SubClassOf>
322   <Class IRI="#Pergunta" />
323   <Class IRI="#Frase" />
324 </SubClassOf>
325 <SubClassOf>
326   <Class IRI="#Precisa" />
327   <Class IRI="#Investigadora" />
328 </SubClassOf>
329 <SubClassOf>
330   <Class IRI="#Reflexiva" />
331   <Class IRI="#Investigadora" />
332 </SubClassOf>
333 <SubClassOf>
334   <Class IRI="#Resposta" />
335   <Class IRI="#Frase" />
336 </SubClassOf>
337 <SubClassOf>
338   <Class IRI="#Retórica" />
339   <Class IRI="#Contraproducente" />
340 </SubClassOf>
341 <SubClassOf>
342   <Class IRI="#Ruim" />
343   <Class IRI="#Avaliação" />
344 </SubClassOf>
345 <SubClassOf>
346   <Class IRI="#Sim_ou_Não" />
347   <Class IRI="#Fechada" />
348 </SubClassOf>
349 <SubClassOf>
350   <Class IRI="#Simples" />
351   <Class IRI="#Investigadora" />
352 </SubClassOf>
353 <DisjointClasses>
354   <Class IRI="#Aberta" />
355   <Class IRI="#Fechada" />
356 </DisjointClasses>
357 <DisjointClasses>
358   <Class IRI="#Boa" />
359   <Class IRI="#Ruim" />
360 </DisjointClasses>
361 <ClassAssertion>
362   <Class IRI="#Pergunta" />
363   <NamedIndividual IRI="#pergunta_col_01" />
364 </ClassAssertion>
365 <ClassAssertion>
366   <ObjectComplementOf>
367     <ObjectSomeValuesFrom>
368       <ObjectProperty IRI="#é_uma_pergunta_do_tipo" />
369       <Class IRI="#Contraproducente" />
370     </ObjectSomeValuesFrom>
371   </ObjectComplementOf>
372   <NamedIndividual IRI="#pergunta_col_01" />

```



```

373 </ClassAssertion>
374 <ClassAssertion>
375   <ObjectSomeValuesFrom>
376     <ObjectProperty IRI="#estrutura_aberta ,pode_ser"/>
377     <Class IRI="#Fundamentada"/>
378   </ObjectSomeValuesFrom>
379   <NamedIndividual IRI="#pergunta_col_01"/>
380 </ClassAssertion>
381 <ClassAssertion>
382   <Class IRI="#Pergunta"/>
383   <NamedIndividual IRI="#pergunta_col_02"/>
384 </ClassAssertion>
385 <ClassAssertion>
386   <ObjectSomeValuesFrom>
387     <ObjectProperty IRI="#é_contraproducente_pois_é"/>
388     <Class IRI="#Ambígua"/>
389   </ObjectSomeValuesFrom>
390   <NamedIndividual IRI="#pergunta_col_02"/>
391 </ClassAssertion>
392 <ClassAssertion>
393   <Class IRI="#Pergunta"/>
394   <NamedIndividual IRI="#pergunta_col_04"/>
395 </ClassAssertion>
396 <ClassAssertion>
397   <ObjectComplementOf>
398     <ObjectSomeValuesFrom>
399       <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
400       <Class IRI="#Contraproducente"/>
401     </ObjectSomeValuesFrom>
402   </ObjectComplementOf>
403   <NamedIndividual IRI="#pergunta_col_04"/>
404 </ClassAssertion>
405 <ClassAssertion>
406   <ObjectSomeValuesFrom>
407     <ObjectProperty IRI="#estrutura_aberta ,pode_ser"/>
408     <Class IRI="#Hipotética"/>
409   </ObjectSomeValuesFrom>
410   <NamedIndividual IRI="#pergunta_col_04"/>
411 </ClassAssertion>
412 <ClassAssertion>
413   <Class IRI="#Pergunta"/>
414   <NamedIndividual IRI="#pergunta_col_05"/>
415 </ClassAssertion>
416 <ClassAssertion>
417   <ObjectComplementOf>
418     <ObjectSomeValuesFrom>
419       <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
420       <Class IRI="#Contraproducente"/>
421     </ObjectSomeValuesFrom>
422   </ObjectComplementOf>
423   <NamedIndividual IRI="#pergunta_col_05"/>
424 </ClassAssertion>
425 <ClassAssertion>
426   <ObjectSomeValuesFrom>
427     <ObjectProperty IRI="#estrutura_aberta ,pode_ser"/>
428     <Class IRI="#Comparativa"/>

```

```

429     </ObjectSomeValuesFrom>
430     <NamedIndividual IRI="#pergunta_col_05"/>
431 </ClassAssertion>
432 <ClassAssertion>
433     <Class IRI="#Pergunta"/>
434     <NamedIndividual IRI="#pergunta_col_06"/>
435 </ClassAssertion>
436 <ClassAssertion>
437     <ObjectComplementOf>
438         <ObjectSomeValuesFrom>
439             <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
440             <Class IRI="#Contraproducente"/>
441         </ObjectSomeValuesFrom>
442     </ObjectComplementOf>
443     <NamedIndividual IRI="#pergunta_col_06"/>
444 </ClassAssertion>
445 <ClassAssertion>
446     <ObjectSomeValuesFrom>
447         <ObjectProperty IRI="#estrutura_aberta ,pode_ser"/>
448         <Class IRI="#Extensiva"/>
449     </ObjectSomeValuesFrom>
450     <NamedIndividual IRI="#pergunta_col_06"/>
451 </ClassAssertion>
452 <ClassAssertion>
453     <Class IRI="#Pergunta"/>
454     <NamedIndividual IRI="#pergunta_col_07"/>
455 </ClassAssertion>
456 <ClassAssertion>
457     <ObjectComplementOf>
458         <ObjectSomeValuesFrom>
459             <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
460             <Class IRI="#Contraproducente"/>
461         </ObjectSomeValuesFrom>
462     </ObjectComplementOf>
463     <NamedIndividual IRI="#pergunta_col_07"/>
464 </ClassAssertion>
465 <ClassAssertion>
466     <ObjectSomeValuesFrom>
467         <ObjectProperty IRI="#estrutura_aberta ,pode_ser"/>
468         <Class IRI="#Precisa"/>
469     </ObjectSomeValuesFrom>
470     <NamedIndividual IRI="#pergunta_col_07"/>
471 </ClassAssertion>
472 <ClassAssertion>
473     <Class IRI="#Pergunta"/>
474     <NamedIndividual IRI="#pergunta_col_08"/>
475 </ClassAssertion>
476 <ClassAssertion>
477     <ObjectComplementOf>
478         <ObjectSomeValuesFrom>
479             <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
480             <Class IRI="#Contraproducente"/>
481         </ObjectSomeValuesFrom>
482     </ObjectComplementOf>
483     <NamedIndividual IRI="#pergunta_col_08"/>
484 </ClassAssertion>

```

```

485 <ClassAssertion>
486   <ObjectSomeValuesFrom>
487     <ObjectProperty IRI="#estrutura_aberta_pode_ser"/>
488     <Class IRI="#Reflexiva"/>
489   </ObjectSomeValuesFrom>
490   <NamedIndividual IRI="#pergunta_col_08"/>
491 </ClassAssertion>
492 <ClassAssertion>
493   <Class IRI="#Pergunta"/>
494   <NamedIndividual IRI="#pergunta_col_09"/>
495 </ClassAssertion>
496 <ClassAssertion>
497   <ObjectComplementOf>
498     <ObjectSomeValuesFrom>
499       <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
500       <Class IRI="#Contraproducente"/>
501     </ObjectSomeValuesFrom>
502   </ObjectComplementOf>
503   <NamedIndividual IRI="#pergunta_col_09"/>
504 </ClassAssertion>
505 <ClassAssertion>
506   <ObjectSomeValuesFrom>
507     <ObjectProperty IRI="#estrutura_aberta_pode_ser"/>
508     <Class IRI="#Simples"/>
509   </ObjectSomeValuesFrom>
510   <NamedIndividual IRI="#pergunta_col_09"/>
511 </ClassAssertion>
512 <ClassAssertion>
513   <Class IRI="#Pergunta"/>
514   <NamedIndividual IRI="#pergunta_col_11"/>
515 </ClassAssertion>
516 <ClassAssertion>
517   <ObjectComplementOf>
518     <ObjectSomeValuesFrom>
519       <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
520       <Class IRI="#Contraproducente"/>
521     </ObjectSomeValuesFrom>
522   </ObjectComplementOf>
523   <NamedIndividual IRI="#pergunta_col_11"/>
524 </ClassAssertion>
525 <ClassAssertion>
526   <ObjectSomeValuesFrom>
527     <ObjectProperty IRI="#remonta_uma"/>
528     <Class IRI="#Sim_ou_Não"/>
529   </ObjectSomeValuesFrom>
530   <NamedIndividual IRI="#pergunta_col_11"/>
531 </ClassAssertion>
532 <ClassAssertion>
533   <Class IRI="#Pergunta"/>
534   <NamedIndividual IRI="#pergunta_col_12"/>
535 </ClassAssertion>
536 <ClassAssertion>
537   <ObjectSomeValuesFrom>
538     <ObjectProperty IRI="#é_contraproducente_pois_é"/>
539     <Class IRI="#Ardilosa"/>
540   </ObjectSomeValuesFrom>

```

```

541     <NamedIndividual IRI="#pergunta_col_12"/>
542 </ClassAssertion>
543 <ClassAssertion>
544   <Class IRI="#Pergunta"/>
545   <NamedIndividual IRI="#pergunta_col_14"/>
546 </ClassAssertion>
547 <ClassAssertion>
548   <ObjectSomeValuesFrom>
549     <ObjectProperty IRI="#é_contraproducente_pois_é"/>
550     <Class IRI="#Indutiva"/>
551   </ObjectSomeValuesFrom>
552   <NamedIndividual IRI="#pergunta_col_14"/>
553 </ClassAssertion>
554 <ClassAssertion>
555   <Class IRI="#Pergunta"/>
556   <NamedIndividual IRI="#pergunta_inf_01"/>
557 </ClassAssertion>
558 <ClassAssertion>
559   <ObjectComplementOf>
560     <ObjectSomeValuesFrom>
561       <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
562       <Class IRI="#Contraproducente"/>
563     </ObjectSomeValuesFrom>
564   </ObjectComplementOf>
565   <NamedIndividual IRI="#pergunta_inf_01"/>
566 </ClassAssertion>
567 <ClassAssertion>
568   <ObjectSomeValuesFrom>
569     <ObjectProperty IRI="#estrutura_aberta ,pode_ser"/>
570     <Class IRI="#Fundamentada"/>
571   </ObjectSomeValuesFrom>
572   <NamedIndividual IRI="#pergunta_inf_01"/>
573 </ClassAssertion>
574 <ClassAssertion>
575   <Class IRI="#Aberta"/>
576   <NamedIndividual IRI="#pergunta_inf_02"/>
577 </ClassAssertion>
578 <ClassAssertion>
579   <Class IRI="#Pergunta"/>
580   <NamedIndividual IRI="#pergunta_inf_02"/>
581 </ClassAssertion>
582 <ClassAssertion>
583   <ObjectSomeValuesFrom>
584     <ObjectProperty IRI="#é_contraproducente_pois_é"/>
585     <Class IRI="#Ambígua"/>
586   </ObjectSomeValuesFrom>
587   <NamedIndividual IRI="#pergunta_inf_02"/>
588 </ClassAssertion>
589 <ClassAssertion>
590   <Class IRI="#De_contato"/>
591   <NamedIndividual IRI="#pergunta_inf_03"/>
592 </ClassAssertion>
593 <ClassAssertion>
594   <Class IRI="#Pergunta"/>
595   <NamedIndividual IRI="#pergunta_inf_03"/>
596 </ClassAssertion>

```

```

597 <ClassAssertion>
598   <ObjectComplementOf>
599     <ObjectSomeValuesFrom>
600       <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
601       <Class IRI="#Contraproducente"/>
602     </ObjectSomeValuesFrom>
603   </ObjectComplementOf>
604   <NamedIndividual IRI="#pergunta_inf_03"/>
605 </ClassAssertion>
606 <ClassAssertion>
607   <Class IRI="#Hipotética"/>
608   <NamedIndividual IRI="#pergunta_inf_04"/>
609 </ClassAssertion>
610 <ClassAssertion>
611   <Class IRI="#Pergunta"/>
612   <NamedIndividual IRI="#pergunta_inf_04"/>
613 </ClassAssertion>
614 <ClassAssertion>
615   <ObjectComplementOf>
616     <ObjectSomeValuesFrom>
617       <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
618       <Class IRI="#Contraproducente"/>
619     </ObjectSomeValuesFrom>
620   </ObjectComplementOf>
621   <NamedIndividual IRI="#pergunta_inf_04"/>
622 </ClassAssertion>
623 <ClassAssertion>
624   <Class IRI="#Pergunta"/>
625   <NamedIndividual IRI="#pergunta_inf_05"/>
626 </ClassAssertion>
627 <ClassAssertion>
628   <ObjectComplementOf>
629     <ObjectSomeValuesFrom>
630       <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
631       <Class IRI="#Contraproducente"/>
632     </ObjectSomeValuesFrom>
633   </ObjectComplementOf>
634   <NamedIndividual IRI="#pergunta_inf_05"/>
635 </ClassAssertion>
636 <ClassAssertion>
637   <ObjectSomeValuesFrom>
638     <ObjectProperty IRI="#estrutura_aberta ,pode_ser"/>
639     <Class IRI="#Comparativa"/>
640   </ObjectSomeValuesFrom>
641   <NamedIndividual IRI="#pergunta_inf_05"/>
642 </ClassAssertion>
643 <ClassAssertion>
644   <Class IRI="#Pergunta"/>
645   <NamedIndividual IRI="#pergunta_inf_06"/>
646 </ClassAssertion>
647 <ClassAssertion>
648   <ObjectComplementOf>
649     <ObjectSomeValuesFrom>
650       <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
651       <Class IRI="#Contraproducente"/>
652     </ObjectSomeValuesFrom>

```

```

653     </ObjectComplementOf>
654     <NamedIndividual IRI="#pergunta_inf_06"/>
655 </ClassAssertion>
656 <ClassAssertion>
657     <ObjectSomeValuesFrom>
658         <ObjectProperty IRI="#estrutura_aberta_pode_ser"/>
659         <Class IRI="#Extensiva"/>
660     </ObjectSomeValuesFrom>
661     <NamedIndividual IRI="#pergunta_inf_06"/>
662 </ClassAssertion>
663 <ClassAssertion>
664     <Class IRI="#Pergunta"/>
665     <NamedIndividual IRI="#pergunta_inf_07"/>
666 </ClassAssertion>
667 <ClassAssertion>
668     <ObjectComplementOf>
669         <ObjectSomeValuesFrom>
670             <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
671             <Class IRI="#Contraproducente"/>
672         </ObjectSomeValuesFrom>
673     </ObjectComplementOf>
674     <NamedIndividual IRI="#pergunta_inf_07"/>
675 </ClassAssertion>
676 <ClassAssertion>
677     <ObjectSomeValuesFrom>
678         <ObjectProperty IRI="#estrutura_aberta_pode_ser"/>
679         <Class IRI="#Precisa"/>
680     </ObjectSomeValuesFrom>
681     <NamedIndividual IRI="#pergunta_inf_07"/>
682 </ClassAssertion>
683 <ClassAssertion>
684     <Class IRI="#Pergunta"/>
685     <NamedIndividual IRI="#pergunta_inf_08"/>
686 </ClassAssertion>
687 <ClassAssertion>
688     <ObjectComplementOf>
689         <ObjectSomeValuesFrom>
690             <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
691             <Class IRI="#Contraproducente"/>
692         </ObjectSomeValuesFrom>
693     </ObjectComplementOf>
694     <NamedIndividual IRI="#pergunta_inf_08"/>
695 </ClassAssertion>
696 <ClassAssertion>
697     <ObjectSomeValuesFrom>
698         <ObjectProperty IRI="#estrutura_aberta_pode_ser"/>
699         <Class IRI="#Reflexiva"/>
700     </ObjectSomeValuesFrom>
701     <NamedIndividual IRI="#pergunta_inf_08"/>
702 </ClassAssertion>
703 <ClassAssertion>
704     <Class IRI="#Pergunta"/>
705     <NamedIndividual IRI="#pergunta_inf_09"/>
706 </ClassAssertion>
707 <ClassAssertion>
708     <ObjectComplementOf>

```

```

709     <ObjectSomeValuesFrom>
710         <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
711         <Class IRI="#Contraproducente"/>
712     </ObjectSomeValuesFrom>
713 </ObjectComplementOf>
714     <NamedIndividual IRI="#pergunta_inf_09"/>
715 </ClassAssertion>
716 <ClassAssertion>
717     <ObjectSomeValuesFrom>
718         <ObjectProperty IRI="#estrutura_aberta_pode_ser"/>
719         <Class IRI="#Simples"/>
720     </ObjectSomeValuesFrom>
721     <NamedIndividual IRI="#pergunta_inf_09"/>
722 </ClassAssertion>
723 <ClassAssertion>
724     <Class IRI="#Pergunta"/>
725     <NamedIndividual IRI="#pergunta_inf_10"/>
726 </ClassAssertion>
727 <ClassAssertion>
728     <ObjectComplementOf>
729         <ObjectSomeValuesFrom>
730             <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
731             <Class IRI="#Contraproducente"/>
732         </ObjectSomeValuesFrom>
733     </ObjectComplementOf>
734     <NamedIndividual IRI="#pergunta_inf_10"/>
735 </ClassAssertion>
736 <ClassAssertion>
737     <ObjectSomeValuesFrom>
738         <ObjectProperty IRI="#remonta_uma"/>
739         <Class IRI="#Identificação"/>
740     </ObjectSomeValuesFrom>
741     <NamedIndividual IRI="#pergunta_inf_10"/>
742 </ClassAssertion>
743 <ClassAssertion>
744     <Class IRI="#Pergunta"/>
745     <NamedIndividual IRI="#pergunta_inf_11"/>
746 </ClassAssertion>
747 <ClassAssertion>
748     <ObjectComplementOf>
749         <ObjectSomeValuesFrom>
750             <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
751             <Class IRI="#Contraproducente"/>
752         </ObjectSomeValuesFrom>
753     </ObjectComplementOf>
754     <NamedIndividual IRI="#pergunta_inf_11"/>
755 </ClassAssertion>
756 <ClassAssertion>
757     <ObjectSomeValuesFrom>
758         <ObjectProperty IRI="#remonta_uma"/>
759         <Class IRI="#Sim_ou_Não"/>
760     </ObjectSomeValuesFrom>
761     <NamedIndividual IRI="#pergunta_inf_11"/>
762 </ClassAssertion>
763 <ClassAssertion>
764     <Class IRI="#Pergunta"/>

```

```

765     <NamedIndividual IRI="#pergunta_inf_12"/>
766 </ClassAssertion>
767 <ClassAssertion>
768   <ObjectSomeValuesFrom>
769     <ObjectProperty IRI="#é_contraproducente_pois_é"/>
770     <Class IRI="#Ardilosa"/>
771   </ObjectSomeValuesFrom>
772   <NamedIndividual IRI="#pergunta_inf_12"/>
773 </ClassAssertion>
774 <ClassAssertion>
775   <Class IRI="#Pergunta"/>
776   <NamedIndividual IRI="#pergunta_inf_13"/>
777 </ClassAssertion>
778 <ClassAssertion>
779   <ObjectSomeValuesFrom>
780     <ObjectProperty IRI="#é_contraproducente_pois_é"/>
781     <Class IRI="#Discriminatória"/>
782   </ObjectSomeValuesFrom>
783   <NamedIndividual IRI="#pergunta_inf_13"/>
784 </ClassAssertion>
785 <ClassAssertion>
786   <Class IRI="#Pergunta"/>
787   <NamedIndividual IRI="#pergunta_inf_14"/>
788 </ClassAssertion>
789 <ClassAssertion>
790   <ObjectSomeValuesFrom>
791     <ObjectProperty IRI="#é_contraproducente_pois_é"/>
792     <Class IRI="#Indutiva"/>
793   </ObjectSomeValuesFrom>
794   <NamedIndividual IRI="#pergunta_inf_14"/>
795 </ClassAssertion>
796 <ClassAssertion>
797   <Class IRI="#Pergunta"/>
798   <NamedIndividual IRI="#pergunta_inf_15"/>
799 </ClassAssertion>
800 <ClassAssertion>
801   <ObjectSomeValuesFrom>
802     <ObjectProperty IRI="#é_contraproducente_pois_é"/>
803     <Class IRI="#Retórica"/>
804   </ObjectSomeValuesFrom>
805   <NamedIndividual IRI="#pergunta_inf_15"/>
806 </ClassAssertion>
807 <ClassAssertion>
808   <Class IRI="#Pergunta"/>
809   <NamedIndividual IRI="#pergunta_inf_16"/>
810 </ClassAssertion>
811 <ClassAssertion>
812   <ObjectComplementOf>
813     <ObjectSomeValuesFrom>
814       <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
815       <Class IRI="#Contraproducente"/>
816     </ObjectSomeValuesFrom>
817   </ObjectComplementOf>
818   <NamedIndividual IRI="#pergunta_inf_16"/>
819 </ClassAssertion>
820 <ClassAssertion>

```



```

821     <ObjectSomeValuesFrom>
822       <ObjectProperty IRI="#estrutura_aberta ,pode_ser" />
823       <Class IRI="#Conclusiva" />
824     </ObjectSomeValuesFrom>
825     <NamedIndividual IRI="#pergunta_inf_16" />
826   </ClassAssertion>
827   <SubObjectPropertyOf>
828     <ObjectProperty IRI="#possui_uma_natureza" />
829     <ObjectProperty IRI="#é_uma" />
830   </SubObjectPropertyOf>
831   <SubObjectPropertyOf>
832     <ObjectProperty IRI="#remonta_uma" />
833     <ObjectProperty IRI="#é_uma_pergunta_do_tipo" />
834   </SubObjectPropertyOf>
835   <SubObjectPropertyOf>
836     <ObjectProperty IRI="#também_é_uma" />
837     <ObjectProperty IRI="#é_um_tipo_de" />
838   </SubObjectPropertyOf>
839   <SubObjectPropertyOf>
840     <ObjectProperty IRI="#tem_intencao" />
841     <ObjectProperty IRI="#é_uma_pergunta_do_tipo" />
842   </SubObjectPropertyOf>
843   <SubObjectPropertyOf>
844     <ObjectProperty IRI="#é_contraproducente" />
845     <ObjectProperty IRI="#é_um_tipo_de" />
846   </SubObjectPropertyOf>
847   <SubObjectPropertyOf>
848     <ObjectProperty IRI="#é_contraproducente_pois_é" />
849     <ObjectProperty IRI="#é_uma_pergunta_do_tipo" />
850   </SubObjectPropertyOf>
851   <SubObjectPropertyOf>
852     <ObjectProperty IRI="#é_dada_como" />
853     <ObjectProperty IRI="#sofre_uma" />
854   </SubObjectPropertyOf>
855   <SubObjectPropertyOf>
856     <ObjectProperty IRI="#é_fruto_de_uma" />
857     <ObjectProperty IRI="#é_aplicada_sobre_uma" />
858   </SubObjectPropertyOf>
859   <SubObjectPropertyOf>
860     <ObjectProperty IRI="#é_uma" />
861     <ObjectProperty IRI="#é_um_tipo_de" />
862   </SubObjectPropertyOf>
863   <SubObjectPropertyOf>
864     <ObjectProperty IRI="#estrutura_aberta ,pode_ser" />
865     <ObjectProperty IRI="#tem_intencao" />
866   </SubObjectPropertyOf>
867   <InverseObjectProperties>
868     <ObjectProperty IRI="#estrutura_aberta ,pode_ser" />
869     <ObjectProperty IRI="#possui_uma_natureza" />
870   </InverseObjectProperties>
871   <InverseObjectProperties>
872     <ObjectProperty IRI="#remonta_uma" />
873     <ObjectProperty IRI="#também_é_uma" />
874   </InverseObjectProperties>
875   <InverseObjectProperties>
876     <ObjectProperty IRI="#sofre_uma" />

```

```

877     <ObjectProperty IRI="#é_aplicada_sobre_uma"/>
878 </InverseObjectProperty>
879 <InverseObjectProperties>
880     <ObjectProperty IRI="#é_uma"/>
881     <ObjectProperty IRI="#tem_intencao"/>
882 </InverseObjectProperties>
883 <InverseObjectProperties>
884     <ObjectProperty IRI="#é_contraproducente"/>
885     <ObjectProperty IRI="#é_contraproducente_pois_é"/>
886 </InverseObjectProperties>
887 <InverseObjectProperties>
888     <ObjectProperty IRI="#é_fruto_de_uma"/>
889     <ObjectProperty IRI="#é_dada_como"/>
890 </InverseObjectProperties>
891 <InverseObjectProperties>
892     <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
893     <ObjectProperty IRI="#é_um_tipo_de"/>
894 </InverseObjectProperties>
895 <FunctionalObjectProperty>
896     <ObjectProperty IRI="#é_dada_como"/>
897 </FunctionalObjectProperty>
898 <FunctionalObjectProperty>
899     <ObjectProperty IRI="#é_fruto_de_uma"/>
900 </FunctionalObjectProperty>
901 <InverseFunctionalObjectProperty>
902     <ObjectProperty IRI="#possui_uma_natureza"/>
903 </InverseFunctionalObjectProperty>
904 <InverseFunctionalObjectProperty>
905     <ObjectProperty IRI="#remonta_uma"/>
906 </InverseFunctionalObjectProperty>
907 <InverseFunctionalObjectProperty>
908     <ObjectProperty IRI="#também_é_uma"/>
909 </InverseFunctionalObjectProperty>
910 <InverseFunctionalObjectProperty>
911     <ObjectProperty IRI="#é_contraproducente"/>
912 </InverseFunctionalObjectProperty>
913 <InverseFunctionalObjectProperty>
914     <ObjectProperty IRI="#é_contraproducente_pois_é"/>
915 </InverseFunctionalObjectProperty>
916 <InverseFunctionalObjectProperty>
917     <ObjectProperty IRI="#é_dada_como"/>
918 </InverseFunctionalObjectProperty>
919 <InverseFunctionalObjectProperty>
920     <ObjectProperty IRI="#é_fruto_de_uma"/>
921 </InverseFunctionalObjectProperty>
922 <InverseFunctionalObjectProperty>
923     <ObjectProperty IRI="#estrutura_aberta ,pode_ser"/>
924 </InverseFunctionalObjectProperty>
925 <TransitiveObjectProperty>
926     <ObjectProperty IRI="#sofre_uma"/>
927 </TransitiveObjectProperty>
928 <TransitiveObjectProperty>
929     <ObjectProperty IRI="#tem_intencao"/>
930 </TransitiveObjectProperty>
931 <TransitiveObjectProperty>
932     <ObjectProperty IRI="#é_aplicada_sobre_uma"/>

```

```

933 </TransitiveObjectProperty>
934 <TransitiveObjectProperty>
935   <ObjectProperty IRI="#é_um_tipo_de"/>
936 </TransitiveObjectProperty>
937 <TransitiveObjectProperty>
938   <ObjectProperty IRI="#é_uma"/>
939 </TransitiveObjectProperty>
940 <TransitiveObjectProperty>
941   <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
942 </TransitiveObjectProperty>
943 <ObjectPropertyDomain>
944   <ObjectProperty IRI="#possui_uma_natureza"/>
945   <Class IRI="#Comparativa"/>
946 </ObjectPropertyDomain>
947 <ObjectPropertyDomain>
948   <ObjectProperty IRI="#possui_uma_natureza"/>
949   <Class IRI="#Conclusiva"/>
950 </ObjectPropertyDomain>
951 <ObjectPropertyDomain>
952   <ObjectProperty IRI="#possui_uma_natureza"/>
953   <Class IRI="#Extensiva"/>
954 </ObjectPropertyDomain>
955 <ObjectPropertyDomain>
956   <ObjectProperty IRI="#possui_uma_natureza"/>
957   <Class IRI="#Fundamentada"/>
958 </ObjectPropertyDomain>
959 <ObjectPropertyDomain>
960   <ObjectProperty IRI="#possui_uma_natureza"/>
961   <Class IRI="#Hipotética"/>
962 </ObjectPropertyDomain>
963 <ObjectPropertyDomain>
964   <ObjectProperty IRI="#possui_uma_natureza"/>
965   <Class IRI="#Precisa"/>
966 </ObjectPropertyDomain>
967 <ObjectPropertyDomain>
968   <ObjectProperty IRI="#possui_uma_natureza"/>
969   <Class IRI="#Reflexiva"/>
970 </ObjectPropertyDomain>
971 <ObjectPropertyDomain>
972   <ObjectProperty IRI="#possui_uma_natureza"/>
973   <Class IRI="#Simples"/>
974 </ObjectPropertyDomain>
975 <ObjectPropertyDomain>
976   <ObjectProperty IRI="#remonta_uma"/>
977   <Class IRI="#Fechada"/>
978 </ObjectPropertyDomain>
979 <ObjectPropertyDomain>
980   <ObjectProperty IRI="#sofre_uma"/>
981   <Class IRI="#Pergunta"/>
982 </ObjectPropertyDomain>
983 <ObjectPropertyDomain>
984   <ObjectProperty IRI="#também_é_uma"/>
985   <Class IRI="#Identificação"/>
986 </ObjectPropertyDomain>
987 <ObjectPropertyDomain>
988   <ObjectProperty IRI="#também_é_uma"/>

```

```

989     <Class IRI="#Sim_ou_Não" />
990 </ObjectPropertyDomain>
991 <ObjectPropertyDomain>
992     <ObjectProperty IRI="#tem_intencao" />
993     <Class IRI="#Aberta" />
994 </ObjectPropertyDomain>
995 <ObjectPropertyDomain>
996     <ObjectProperty IRI="#é_aplicada_sobre_uma" />
997     <Class IRI="#Avaliação" />
998 </ObjectPropertyDomain>
999 <ObjectPropertyDomain>
1000    <ObjectProperty IRI="#é_contraproducente" />
1001    <Class IRI="#Ambígua" />
1002 </ObjectPropertyDomain>
1003 <ObjectPropertyDomain>
1004    <ObjectProperty IRI="#é_contraproducente" />
1005    <Class IRI="#Ardilosa" />
1006 </ObjectPropertyDomain>
1007 <ObjectPropertyDomain>
1008    <ObjectProperty IRI="#é_contraproducente" />
1009    <Class IRI="#Discriminatória" />
1010 </ObjectPropertyDomain>
1011 <ObjectPropertyDomain>
1012    <ObjectProperty IRI="#é_contraproducente" />
1013    <Class IRI="#Indutiva" />
1014 </ObjectPropertyDomain>
1015 <ObjectPropertyDomain>
1016    <ObjectProperty IRI="#é_contraproducente" />
1017    <Class IRI="#Retórica" />
1018 </ObjectPropertyDomain>
1019 <ObjectPropertyDomain>
1020    <ObjectProperty IRI="#é_contraproducente_pois_é" />
1021    <Class IRI="#Contraproducente" />
1022 </ObjectPropertyDomain>
1023 <ObjectPropertyDomain>
1024    <ObjectProperty IRI="#é_dada_como" />
1025    <Class IRI="#Avaliação" />
1026 </ObjectPropertyDomain>
1027 <ObjectPropertyDomain>
1028    <ObjectProperty IRI="#é_fruto_de_uma" />
1029    <ObjectUnionOf>
1030        <Class IRI="#Boa" />
1031        <Class IRI="#Ruim" />
1032    </ObjectUnionOf>
1033 </ObjectPropertyDomain>
1034 <ObjectPropertyDomain>
1035    <ObjectProperty IRI="#é_um_tipo_de" />
1036    <ObjectUnionOf>
1037        <Class IRI="#Aberta" />
1038        <Class IRI="#Contraproducente" />
1039        <Class IRI="#Fechada" />
1040    </ObjectUnionOf>
1041 </ObjectPropertyDomain>
1042 <ObjectPropertyDomain>
1043    <ObjectProperty IRI="#é_uma" />
1044    <Class IRI="#De_contato" />

```

```

1045 </ObjectPropertyDomain>
1046 <ObjectPropertyDomain>
1047   <ObjectProperty IRI="#é_uma"/>
1048   <Class IRI="#Investigadora"/>
1049 </ObjectPropertyDomain>
1050 <ObjectPropertyDomain>
1051   <ObjectProperty IRI="#é_uma_pergunta_do_tipo"/>
1052   <Class IRI="#Pergunta"/>
1053 </ObjectPropertyDomain>
1054 <ObjectPropertyDomain>
1055   <ObjectProperty IRI="#estrutura_aberta_pode_ser"/>
1056   <Class IRI="#Investigadora"/>
1057 </ObjectPropertyDomain>
1058 <ObjectPropertyRange>
1059   <ObjectProperty IRI="#possui_uma_natureza"/>
1060   <Class IRI="#Investigadora"/>
1061 </ObjectPropertyRange>
1062 <ObjectPropertyRange>
1063   <ObjectProperty IRI="#remonta_uma"/>
1064   <Class IRI="#Identificação"/>
1065 </ObjectPropertyRange>
1066 <ObjectPropertyRange>
1067   <ObjectProperty IRI="#remonta_uma"/>
1068   <Class IRI="#Sim_ou_Não"/>
1069 </ObjectPropertyRange>
1070 <ObjectPropertyRange>
1071   <ObjectProperty IRI="#sofre_uma"/>
1072   <Class IRI="#Avaliação"/>
1073 </ObjectPropertyRange>
1074 <ObjectPropertyRange>
1075   <ObjectProperty IRI="#também_é_uma"/>
1076   <Class IRI="#Fechada"/>
1077 </ObjectPropertyRange>
1078 <ObjectPropertyRange>
1079   <ObjectProperty IRI="#tem_intencao"/>
1080   <Class IRI="#De_contato"/>
1081 </ObjectPropertyRange>
1082 <ObjectPropertyRange>
1083   <ObjectProperty IRI="#tem_intencao"/>
1084   <Class IRI="#Investigadora"/>
1085 </ObjectPropertyRange>
1086 <ObjectPropertyRange>
1087   <ObjectProperty IRI="#é_aplicada_sobre_uma"/>
1088   <Class IRI="#Pergunta"/>
1089 </ObjectPropertyRange>
1090 <ObjectPropertyRange>
1091   <ObjectProperty IRI="#é_contraproducente"/>
1092   <Class IRI="#Contraproducente"/>
1093 </ObjectPropertyRange>
1094 <ObjectPropertyRange>
1095   <ObjectProperty IRI="#é_contraproducente_pois_é"/>
1096   <Class IRI="#Ambígua"/>
1097 </ObjectPropertyRange>
1098 <ObjectPropertyRange>
1099   <ObjectProperty IRI="#é_contraproducente_pois_é"/>
1100   <Class IRI="#Ardilosa"/>

```

```

1101 </ObjectPropertyRange>
1102 <ObjectPropertyRange>
1103   <ObjectProperty IRI="#é_contraproducente_pois_é" />
1104   <Class IRI="#Discriminatória" />
1105 </ObjectPropertyRange>
1106 <ObjectPropertyRange>
1107   <ObjectProperty IRI="#é_contraproducente_pois_é" />
1108   <Class IRI="#Indutiva" />
1109 </ObjectPropertyRange>
1110 <ObjectPropertyRange>
1111   <ObjectProperty IRI="#é_contraproducente_pois_é" />
1112   <Class IRI="#Retórica" />
1113 </ObjectPropertyRange>
1114 <ObjectPropertyRange>
1115   <ObjectProperty IRI="#é_dada_como" />
1116   <ObjectUnionOf>
1117     <Class IRI="#Boa" />
1118     <Class IRI="#Ruim" />
1119   </ObjectUnionOf>
1120 </ObjectPropertyRange>
1121 <ObjectPropertyRange>
1122   <ObjectProperty IRI="#é_fruto_de_uma" />
1123   <Class IRI="#Avaliação" />
1124 </ObjectPropertyRange>
1125 <ObjectPropertyRange>
1126   <ObjectProperty IRI="#é_um_tipo_de" />
1127   <Class IRI="#Pergunta" />
1128 </ObjectPropertyRange>
1129 <ObjectPropertyRange>
1130   <ObjectProperty IRI="#é_uma" />
1131   <Class IRI="#Aberta" />
1132 </ObjectPropertyRange>
1133 <ObjectPropertyRange>
1134   <ObjectProperty IRI="#é_uma_pergunta_do_tipo" />
1135   <ObjectUnionOf>
1136     <Class IRI="#Aberta" />
1137     <Class IRI="#Contraproducente" />
1138     <Class IRI="#Fechada" />
1139   </ObjectUnionOf>
1140 </ObjectPropertyRange>
1141 <ObjectPropertyRange>
1142   <ObjectProperty IRI="#estrutura_aberta_pode_ser" />
1143   <Class IRI="#Comparativa" />
1144 </ObjectPropertyRange>
1145 <ObjectPropertyRange>
1146   <ObjectProperty IRI="#estrutura_aberta_pode_ser" />
1147   <Class IRI="#Conclusiva" />
1148 </ObjectPropertyRange>
1149 <ObjectPropertyRange>
1150   <ObjectProperty IRI="#estrutura_aberta_pode_ser" />
1151   <Class IRI="#Extensiva" />
1152 </ObjectPropertyRange>
1153 <ObjectPropertyRange>
1154   <ObjectProperty IRI="#estrutura_aberta_pode_ser" />
1155   <Class IRI="#Fundamentada" />
1156 </ObjectPropertyRange>

```

```

1157 <ObjectPropertyRange>
1158   <ObjectProperty IRI="#estrutura_aberta_pode_ser"/>
1159   <Class IRI="#Hipotética"/>
1160 </ObjectPropertyRange>
1161 <ObjectPropertyRange>
1162   <ObjectProperty IRI="#estrutura_aberta_pode_ser"/>
1163   <Class IRI="#Precisa"/>
1164 </ObjectPropertyRange>
1165 <ObjectPropertyRange>
1166   <ObjectProperty IRI="#estrutura_aberta_pode_ser"/>
1167   <Class IRI="#Reflexiva"/>
1168 </ObjectPropertyRange>
1169 <ObjectPropertyRange>
1170   <ObjectProperty IRI="#estrutura_aberta_pode_ser"/>
1171   <Class IRI="#Simples"/>
1172 </ObjectPropertyRange>
1173 <AnnotationAssertion>
1174   <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
1175   <IRI##Aberta</IRI>
1176   <Literal datatypeIRI="&rdf;PlainLiteral">Perguntas abertas , são
aquelas que incentivam o respondente a se expressar , levando a respostas
mais lingas e elaboradas.</Literal>
1177 </AnnotationAssertion>
1178 <AnnotationAssertion>
1179   <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
1180   <IRI##Ambígua</IRI>
1181   <Literal datatypeIRI="&rdf;PlainLiteral">Uma pergunta é ambigua ,
quando é feita de forma a permitir várias interpretações diferentes do
seu conteúdo, se utilizando tanto de construções ambíguas, quanto de
palavras com mais de um significado.</Literal>
1182 </AnnotationAssertion>
1183 <AnnotationAssertion>
1184   <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
1185   <IRI##Ardilosa</IRI>
1186   <Literal datatypeIRI="&rdf;PlainLiteral">Uma pergunta é ardilosa ,
quando tem o objetivo de enganar e/ou confundir o seu alvo.</Literal>
1187 </AnnotationAssertion>
1188 <AnnotationAssertion>
1189   <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
1190   <IRI##Avaliação</IRI>
1191   <Literal datatypeIRI="&rdf;PlainLiteral">Uma pergunta pode ser
classificada , conforme parâmetros pré-estabelecidos , em uma boa ou má
pergunta.</Literal>
1192 </AnnotationAssertion>
1193 <AnnotationAssertion>
1194   <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
1195   <IRI##Boa</IRI>
1196   <Literal datatypeIRI="&rdf;PlainLiteral">Uma boa pergunta</Literal>
1197 </AnnotationAssertion>
1198 <AnnotationAssertion>
1199   <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
1200   <IRI##Comparativa</IRI>
1201   <Literal datatypeIRI="&rdf;PlainLiteral">Perguntas que buscam
informações através do uso de comparações.</Literal>
1202 </AnnotationAssertion>
1203 <AnnotationAssertion>

```

```

1204     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1205     <IRI##Conclusiva</IRI>
1206     <Literal datatypeIRI="&rdf;PlainLiteral">Perguntas que utilizam
conclusões sobre determinados assuntos, para se construírem.</Literal>
1207 </AnnotationAssertion>
1208 <AnnotationAssertion>
1209     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1210     <IRI##Contraproducente</IRI>
1211     <Literal datatypeIRI="&rdf;PlainLiteral">Contraproducente é um
comportamento que influencia negativamente.</Literal>
1212 </AnnotationAssertion>
1213 <AnnotationAssertion>
1214     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1215     <IRI##De_contato</IRI>
1216     <Literal datatypeIRI="&rdf;PlainLiteral">São perguntas que tem como
objetivo, estabelecer uma espécie de contato, tanto com o respondente
quanto com o tema.</Literal>
1217 </AnnotationAssertion>
1218 <AnnotationAssertion>
1219     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1220     <IRI##Discriminatória</IRI>
1221     <Literal datatypeIRI="&rdf;PlainLiteral">Uma pergunta é discrimnató
ria, quando desencoraja o respondente (ou intimida-o) de forma a indicar
uma parcialidade sobre algum assunto.</Literal>
1222 </AnnotationAssertion>
1223 <AnnotationAssertion>
1224     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1225     <IRI##Extensiva</IRI>
1226     <Literal datatypeIRI="&rdf;PlainLiteral">Perguntas extensivas são
aquelas que utilizam como base perguntas anteriores(ou fatos) para
investigar.</Literal>
1227 </AnnotationAssertion>
1228 <AnnotationAssertion>
1229     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1230     <IRI##Fechada</IRI>
1231     <Literal datatypeIRI="&rdf;PlainLiteral">Uma pergunta fechada,
oferece ao inquisidor, itens específicos de informação.</Literal>
1232 </AnnotationAssertion>
1233 <AnnotationAssertion>
1234     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1235     <IRI##Frase</IRI>
1236     <Literal datatypeIRI="&rdf;PlainLiteral">Frase é todo enunciado
linguístico capaz de transmitir uma ideia. A frase é uma palavra ou
conjunto de palavras que constitui um enunciado de sentido completo.
1237 A frase se define pelo propósito de comunicação, e não pela sua extensão. O
conceito de frase, portanto, abrange desde estruturas linguísticas
muito simples até enunciados bastante complexos.</Literal>
1238 </AnnotationAssertion>
1239 <AnnotationAssertion>
1240     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1241     <IRI##Fundamentada</IRI>
1242     <Literal datatypeIRI="&rdf;PlainLiteral">São perguntas, que
fundamentadas em algum fato, circunstância, ou etc. buscam investigar um
assunto.</Literal>
1243 </AnnotationAssertion>
1244 <AnnotationAssertion>

```



```

1245     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1246     <IRI>#Hipotética</IRI>
1247     <Literal datatypeIRI="&rdf;PlainLiteral">São perguntas que usam hip
1248     óteses para construir sua linha de raciocínio.</Literal>
1248 </AnnotationAssertion>
1249 <AnnotationAssertion>
1250     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1251     <IRI>#Identificação</IRI>
1252     <Literal datatypeIRI="&rdf;PlainLiteral">Perguntas de identificação
1253     , são aquelas que procuram informações específicas e precisas , buscando
1254     identificar um lugar , pessoa , quantidade ... sendo portanto perguntas
1255     bastante objetivas.</Literal>
1253 </AnnotationAssertion>
1254 <AnnotationAssertion>
1255     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1256     <IRI>#Indutiva</IRI>
1257     <Literal datatypeIRI="&rdf;PlainLiteral">Uma pergunta é indutiva ,
1258     quando induz o respondente a uma resposta (podendo ou não, ser uma induç
1259     ão ao erro ).</Literal>
1258 </AnnotationAssertion>
1259 <AnnotationAssertion>
1260     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1261     <IRI>#Investigadora</IRI>
1262     <Literal datatypeIRI="&rdf;PlainLiteral">São perguntas , como o pró
1263     prio nome sugere , que possuem o objetivo de investigar determinado tema
1264     mais profundamente.</Literal>
1263 </AnnotationAssertion>
1264 <AnnotationAssertion>
1265     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1266     <IRI>#Pergunta</IRI>
1267     <Literal datatypeIRI="&rdf;PlainLiteral">Pergunta é uma frase cujo
1268     objetivo é convidar um ouvinte ou leitor a dar uma explicação, uma
1269     informação, ou, em qualquer caso, uma interrogação que demanda, ou
1270     solicita uma resposta.</Literal>
1268 </AnnotationAssertion>
1269 <AnnotationAssertion>
1270     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1271     <IRI>#Precisa</IRI>
1272     <Literal datatypeIRI="&rdf;PlainLiteral">Perguntas que especificam
1273     trechos precisos de atuação.</Literal>
1273 </AnnotationAssertion>
1274 <AnnotationAssertion>
1275     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1276     <IRI>#Reflexiva</IRI>
1277     <Literal datatypeIRI="&rdf;PlainLiteral">Uma pergunta que explora
1278     especificamente a reflexão na visão do respondente.</Literal>
1278 </AnnotationAssertion>
1279 <AnnotationAssertion>
1280     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1281     <IRI>#Resposta</IRI>
1282     <Literal datatypeIRI="&rdf;PlainLiteral">As partes referentes as
1283     respostas serão tratadas em trabalhos futuros</Literal>
1283 </AnnotationAssertion>
1284 <AnnotationAssertion>
1285     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1286     <IRI>#Retórica</IRI>

```

```

1287     <Literal datatypeIRI="&rdf;PlainLiteral">Um tipo de pergunta , onde
1288     o questionador responde a sua própria pergunta.</Literal>
1289   </AnnotationAssertion>
1290   <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1291   <IRI##Ruim</IRI>
1292   <Literal datatypeIRI="&rdf;PlainLiteral">Uma pergunta ruim</Literal
1293 >
1294 </AnnotationAssertion>
1295 <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1296 <IRI##Sim_ou_Não</IRI>
1297 <Literal datatypeIRI="&rdf;PlainLiteral">São perguntas que tem como
1298 propósito , gerar respostas que se restrinjam a sim ou não.</Literal>
1299 </AnnotationAssertion>
1300 <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1301 <IRI##Simples</IRI>
1302 <Literal datatypeIRI="&rdf;PlainLiteral">Perguntas que de forma
1303 descomplicada fazem com que o respondente se manifeste.</Literal>
1304 </AnnotationAssertion>
1305 <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1306 <IRI##pergunta_col_01</IRI>
1307 <Literal datatypeIRI="&rdf;PlainLiteral">Como seria o uso de uma á
1308 rvore para de-codificação de código Morse?</Literal>
1309 </AnnotationAssertion>
1310 <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1311 <IRI##pergunta_col_02</IRI>
1312 <Literal datatypeIRI="&rdf;PlainLiteral">A calculadora polonesa
1313 seria melhor im-plementada com arvore?</Literal>
1314 </AnnotationAssertion>
1315 <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1316 <IRI##pergunta_col_04</IRI>
1317 <Literal datatypeIRI="&rdf;PlainLiteral">É possível criar 2 árvores
1318 podendo haverum nó em comum entre eles?</Literal>
1319 </AnnotationAssertion>
1320 <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1321 <IRI##pergunta_col_05</IRI>
1322 <Literal datatypeIRI="&rdf;PlainLiteral">Qual ganho se tem em
1323 implementar umacalculadora usando uma árvore binária emvez de usar uma
1324 pilha?</Literal>
1325 </AnnotationAssertion>
1326 <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1327 <IRI##pergunta_col_06</IRI>
1328 <Literal datatypeIRI="&rdf;PlainLiteral">Se não, quais problemas
1329 possíveis que pos-sam ocorrer?</Literal>
1330 </AnnotationAssertion>
1331 <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1332 <IRI##pergunta_col_07</IRI>

```

```

1332     <Literal datatypeIRI="&rdf;PlainLiteral">Como usar uma árvore para
1333     implementara função hash(mapeamento de dados)?</Literal>
1334     </AnnotationAssertion>
1335     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1336     <IRI#pergunta_col_08</IRI>
1337     <Literal datatypeIRI="&rdf;PlainLiteral">Seria possível fazer a fun
1338     ção insere nó deuma maneira não recursiva?</Literal>
1339     </AnnotationAssertion>
1340     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1341     <IRI#pergunta_col_09</IRI>
1342     <Literal datatypeIRI="&rdf;PlainLiteral">como faço uma calculadora
1343     em C usandoárvore binária?</Literal>
1344     </AnnotationAssertion>
1345     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1346     <IRI#pergunta_col_11</IRI>
1347     <Literal datatypeIRI="&rdf;PlainLiteral">Existe alguma aplicação em
1348     que realizartravessia na árvore por nível seja a melhoropção?</Literal>
1349     </AnnotationAssertion>
1350     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1351     <IRI#pergunta_col_12</IRI>
1352     <Literal datatypeIRI="&rdf;PlainLiteral">A árvore binária realmente
1353     é o modelo deárvore de melhor rendimento computacional ?</Literal>
1354     </AnnotationAssertion>
1355     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1356     <IRI#pergunta_col_14</IRI>
1357     <Literal datatypeIRI="&rdf;PlainLiteral">Ao executar a remoção de
1358     um elemento es–taremos comprometendo todos os seus fi–lhos e os
1359     respectivos filhos desses filhos?</Literal>
1360     </AnnotationAssertion>
1361     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1362     <IRI#pergunta_inf_01</IRI>
1363     <Literal datatypeIRI="&rdf;PlainLiteral">Como você vê a atual situa
1364     ção do oriente médio?</Literal>
1365     </AnnotationAssertion>
1366     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1367     <IRI#pergunta_inf_02</IRI>
1368     <Literal datatypeIRI="&rdf;PlainLiteral">Você viu o ponto quando
1369     entrou na sala?</Literal>
1370     </AnnotationAssertion>
1371     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1372     <IRI#pergunta_inf_03</IRI>
1373     <Literal datatypeIRI="&rdf;PlainLiteral">Olá, tudo bem?</Literal>
1374     </AnnotationAssertion>
1375     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1376     <IRI#pergunta_inf_04</IRI>
1377     <Literal datatypeIRI="&rdf;PlainLiteral">Hipoteticamente , como você
1378     reagiria?</Literal>

```

```

1378 </AnnotationAssertion>
1379 <AnnotationAssertion>
1380   <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1381   <IRI##pergunta_inf_05</IRI>
1382   <Literal datatypeIRI="&rdfs:PlainLiteral">Qual sistema operacional é
      melhor , IOS ou Android?</Literal>
1383 </AnnotationAssertion>
1384 <AnnotationAssertion>
1385   <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1386   <IRI##pergunta_inf_06</IRI>
1387   <Literal datatypeIRI="&rdfs:PlainLiteral">O que você quis dizer com
      a sua última resposta?</Literal>
1388 </AnnotationAssertion>
1389 <AnnotationAssertion>
1390   <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1391   <IRI##pergunta_inf_07</IRI>
1392   <Literal datatypeIRI="&rdfs:PlainLiteral">O que você quis dizer com
      isso em particular?</Literal>
1393 </AnnotationAssertion>
1394 <AnnotationAssertion>
1395   <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1396   <IRI##pergunta_inf_08</IRI>
1397   <Literal datatypeIRI="&rdfs:PlainLiteral">Para você , como andam os
      rumos da economia?</Literal>
1398 </AnnotationAssertion>
1399 <AnnotationAssertion>
1400   <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1401   <IRI##pergunta_inf_09</IRI>
1402   <Literal datatypeIRI="&rdfs:PlainLiteral">Por quê?</Literal>
1403 </AnnotationAssertion>
1404 <AnnotationAssertion>
1405   <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1406   <IRI##pergunta_inf_10</IRI>
1407   <Literal datatypeIRI="&rdfs:PlainLiteral">Por quanto tempo você
      trabalhou lá?</Literal>
1408 </AnnotationAssertion>
1409 <AnnotationAssertion>
1410   <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1411   <IRI##pergunta_inf_11</IRI>
1412   <Literal datatypeIRI="&rdfs:PlainLiteral">Você é canhoto?</Literal>
1413 </AnnotationAssertion>
1414 <AnnotationAssertion>
1415   <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1416   <IRI##pergunta_inf_12</IRI>
1417   <Literal datatypeIRI="&rdfs:PlainLiteral">Quais são os seus
      principais defeitos?</Literal>
1418 </AnnotationAssertion>
1419 <AnnotationAssertion>
1420   <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1421   <IRI##pergunta_inf_13</IRI>
1422   <Literal datatypeIRI="&rdfs:PlainLiteral">Você possui problemas com
      álcool?</Literal>
1423 </AnnotationAssertion>
1424 <AnnotationAssertion>
1425   <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1426   <IRI##pergunta_inf_14</IRI>

```

```

1427     <Literal datatypeIRI="&rdf;PlainLiteral">Você não gosta de
computadores , certo?</Literal>
1428 </AnnotationAssertion>
1429 <AnnotationAssertion>
1430     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1431     <IRI##pergunta_inf_15</IRI>
1432     <Literal datatypeIRI="&rdf;PlainLiteral">Nós queremos mesmo pintar
a parede?</Literal>
1433 </AnnotationAssertion>
1434 <AnnotationAssertion>
1435     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1436     <IRI##pergunta_inf_16</IRI>
1437     <Literal datatypeIRI="&rdf;PlainLiteral">Se eu entendi direito ,
tudo seria diferente , não?</Literal>
1438 </AnnotationAssertion>
1439 <AnnotationAssertion>
1440     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1441     <IRI##possui_uma_natureza</IRI>
1442     <Literal datatypeIRI="&rdf;PlainLiteral">Uma pergunta _____
possui uma natureza investigadora</Literal>
1443 </AnnotationAssertion>
1444 <AnnotationAssertion>
1445     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1446     <IRI##remonta_uma</IRI>
1447     <Literal datatypeIRI="&rdf;PlainLiteral">Uma pergunta fechada
remonta uma identificação ou uma resposta de Sim ou Não</Literal>
1448 </AnnotationAssertion>
1449 <AnnotationAssertion>
1450     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1451     <IRI##sofre_uma</IRI>
1452     <Literal datatypeIRI="&rdf;PlainLiteral">Uma pergunta sofre uma
avaliação</Literal>
1453 </AnnotationAssertion>
1454 <AnnotationAssertion>
1455     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1456     <IRI##também_é_uma</IRI>
1457     <Literal datatypeIRI="&rdf;PlainLiteral">Uma pergunta de identifica
ção e ou Sim ou Não também é uma pergunta fechada</Literal>
1458 </AnnotationAssertion>
1459 <AnnotationAssertion>
1460     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1461     <IRI##tem_intencao</IRI>
1462     <Literal datatypeIRI="&rdf;PlainLiteral">Uma pergunta aberta tem
uma intenção de ser investigadora ou de estabelecer contato</Literal>
1463 </AnnotationAssertion>
1464 <AnnotationAssertion>
1465     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1466     <IRI##é_aplicada_sobre_uma</IRI>
1467     <Literal datatypeIRI="&rdf;PlainLiteral">Uma avaliação é aplicada
sobre uma pergunta</Literal>
1468 </AnnotationAssertion>
1469 <AnnotationAssertion>
1470     <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1471     <IRI##é_contraproducente</IRI>
1472     <Literal datatypeIRI="&rdf;PlainLiteral">uma pergunta _____ é
contraproducente</Literal>

```

```

1473 </AnnotationAssertion>
1474 <AnnotationAssertion>
1475   <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1476   <IRI##é_dada_como</IRI>
1477   <Literal datatypeIRI="&rdf;PlainLiteral">Uma avaliação é dada como
boa ou ruim</Literal>
1478 </AnnotationAssertion>
1479 <AnnotationAssertion>
1480   <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1481   <IRI##é_fruto_de_uma</IRI>
1482   <Literal datatypeIRI="&rdf;PlainLiteral">Uma pergunta boa ou ruim é
fruto de uma avaliação</Literal>
1483 </AnnotationAssertion>
1484 <AnnotationAssertion>
1485   <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1486   <IRI##é_uma</IRI>
1487   <Literal datatypeIRI="&rdf;PlainLiteral">Uma perguntas
Investigadora ou de Contato é uma pergunta aberta</Literal>
1488 </AnnotationAssertion>
1489 <AnnotationAssertion>
1490   <AnnotationProperty abbreviatedIRI="rdfs:comment" />
1491   <IRI##estrutura_aberta ,pode_ser</IRI>
1492   <Literal datatypeIRI="&rdf;PlainLiteral">Uma pergunta com intenção
investigadora pode ser</Literal>
1493 </AnnotationAssertion>
1494 </Ontology>
1495
1496
1497
1498 <!-- Generated by the OWL API (version 3.5.0) http://owlapi.sourceforge.net
-->

```

# Apêndice B

## Instalação e configuração do ambiente wiki

Aqui indicamos os *scripts* utilizados para a instalação e configuração do ambiente *MediaWiki*:

```
1 #!/bin/bash
2 #Autor: Victor Guimarães
3 #Este script deve ser executado como root ,
4
5 #Script que instala o Media Wiki
6
7 clear
8 clear
9 clear
10 clear
11
12
13
14 #Instalamos/atualizamos o LAMP (Linux, Apache, MySQL, PHP) (tutorial
    completo em: https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu)
15
16 sudo apt-get update
17 #instala o apache
18 sudo apt-get install apache2
19 sudo apt-get install mysql-server libapache2-mod-auth-mysql php5-mysql
20 #instala o mysql
21 sudo mysql_install_db
22 #Instale o mysql, como eu já tenho, comentei a linha
23 #sudo /usr/bin/mysql_secure_installation
24
25 #instala o php
26 sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt
27
28 #reinicio o servidor
29 sudo service apache2 restart
30
31
32
```

```

33 #faço a instalação (fontes: https://www.digitalocean.com/community/
    tutorials/how-to-install-mediawiki-on-ubuntu-12-04 e http://www.youtube.
    com/watch?v=Ku-vuy0d5po)
34 #baixamos o arquivo fonte
35 wget http://releases.wikimedia.org/mediawiki/1.23/mediawiki-1.23.5.tar.gz
36
37 if ! [ $? -eq 0 ] ; then
38     echo -e "\e[1;31m(EE) Erro ao baixar o arquivo \e[0m"
39     exit 1
40 fi
41
42 echo -e "\e[32m===== MediaWiki baixado =====\e[0m"
43
44
45 tar -xvzf mediawiki-1.23.5.tar.gz
46
47
48 mv mediawiki-1.23.5 /etc/mediawiki
49
50 ln -s /etc/mediawiki/ /var/www/html
51
52
53
54 if ! [ $? -eq 0 ] ; then
55     echo -e "\e[1;31m(EE) Erro ao mover o arquivo mediawiki-1.23.5.tar.gz
    para /var/www/html \e[0m"
56     exit 1
57 fi
58
59 echo -e "\e[32m===== Arquivo mediawiki-1.23.5.tar.gz movido para /var/www/
    html com sucesso =====\e[0m"
60
61 rm -R mediawiki-1.23.5
62 rm -R mediawiki-1.23.5.tar.gz
63 cd ..
64 cd ..
65 cd ..
66 cd ..
67 cd ..
68 cd ..
69 cd ..
70 pwd
71 cd var/www/html
72
73
74
75 echo -e "\e[32m===== Arquivo mediawiki-1.23.5.tar.gz descompactado com
    sucesso =====\e[0m"
76
77 mv mediawiki wiki
78
79 chmod 777 wiki
80
81 echo -e "\e[32m===== Pasta mediawiki-1.23.5 renomeada para wiki \n Não se
    esqueça de configurar o BAnco de dados(como no tutorial indicado) \n
    agora o wiki deve ser configurado em http://localhost/wiki =====\e[0m"

```



Fazemos a configuração do *MediaWiki* via *browser* e depois usamos o seguinte *script*:

```
1 #!/ bin/bash
2 #Autor: Victor Guimarães
3 #Este script deve ser executado como root ,
4
5
6 #Segundo passo da instalação do Media Wiki, antes de usa-lo configure o MW
  no endereço: localhost/wiki
7 clear
8 clear
9 clear
10 clear
11
12
13
14 clear
15 clear
16 clear
17 clear
18
19
20 #mova o LocalSettings pra pasta var/www/html/wiki/
21 mv LocalSettings.php ../../../../var/www/html/wiki/
22
23
24
25 cd
26 cd ..
27 cd ..
28 cd ..
29 cd var/www/html/wiki/
30 # conceda permissão de leitura
31
32 chmod 777 LocalSettings.php
33
34 # Na instalação corrente usei o nome de Root Victor e a senha Baleia
```

Em seguida instalamos o *Semantic MediaWiki*

```
1 #!/ bin/bash
2 #Autor: Victor Guimarães
3 #Este script deve ser executado como root , ele instala o Semantic Media
  Wiki
4
5 #Scrpit que instala o Semantic Media Wiki
6 clear
7 clear
8 clear
9 clear
10
11
12
13 #Nos baseamos no seguinte tutorial: http://semantic-mediawiki.org/wiki/Help
  :Installation
14
15 cd ..
```

```
16 cd ..
17 cd ..
18 cd ..
19 cd ..
20 cd ..
21 cd ..
22 cd /etc/mediawiki/
23
24
25
26 sudo curl -sS https://getcomposer.org/installer | php
27
28
29
30 php composer.phar require mediawiki/semantic-media-wiki @dev
31
32 cd maintenance/
33
34 php update.php
```

# Apêndice C

## Programa Java que transforma de OWL para .txt

Aqui indicamos o programa *java* utilizado para transformar o arquivo *owl* em um arquivo de texto entendível pelo *PyWikibot*:

Temos um controlador, que atua como elemento central:

```
1 import java.io.BufferedReader;
2 import java.io.File;
3 import java.io.FileNotFoundException;
4 import java.io.FileReader;
5 import java.io.FileWriter;
6 import java.io.IOException;
7 import java.io.PrintWriter;
8 import java.util.ArrayList;
9
10
11 public class controle {
12
13     /**
14     * @param args
15     * @throws IOException
16     */
17     public static void main(String [] args) throws IOException {
18         // TODO Auto-generated method stub
19
20         File file = new File("/home/victor/Downloads/UnB/TG2/core/modeloTeste.
21 owl");
22         //lista que guarda as páginas
23         ArrayList<dtoPagina> lista = new ArrayList<dtoPagina>();
24         //lista que guarda os ObjectsProperties
25         ArrayList<dtoObjectProperty> listaDeObjectsProperties = new ArrayList<
26 dtoObjectProperty>();
27         //lista que guarda as perguntas
28         ArrayList<dtoPergunta> listaPerguntas = new ArrayList<dtoPergunta>();
29
30         criaListaDePaginas(file , lista);
31
32         colocaAnotacoesNaPagina(file , lista);
33
34         encontraObjectPropertyDomain(file , listaDeObjectsProperties);
```

```

33 encontraObjectPropertyRange(file , listaDeObjectsProperties);
34
35
36
37
38 //Adicionar as relações com objects domains a cada pagina
39 adicionarRelacoes(lista , listaDeObjectsProperties);
40
41 //adiciona as descrições a um Object property
42 adicionadescricoes(file , listaDeObjectsProperties);
43
44 //Adiciona as superClasses e subclasses referentes a cada página
45 adicionaSuperclasses(file , lista);
46 adicionaSubclasses(file , lista);
47
48 //Adicionar as categorias a cada pagina
49 adicionaCategorias(file , lista);
50 adicionaEquivalencias(file , lista);
51
52
53 //parte que trata das perguntas
54 criaListaDePerguntas(file , listaPerguntas);
55 colocaAnotacoesNaPergunta(file , listaPerguntas);
56
57 // escreve em um arquivo .txt para criar as páginas
58 escreveArquivo(lista , listaDeObjectsProperties);
59
60 //escreve um arquivo .txt para eventualmente deletar as páginas
61 criadas
62 escreveArquivoDelete(lista , listaDeObjectsProperties);
63 }
64
65
66 /*
67 * Métodos que encontram as tags
68 * @throws IOException
69 * @throws IOException
70 *
71 *
72 *
73
74 *
75
76
77 private static void criaListaDePerguntas(File file ,
78 ArrayList<dtoPergunta> listaPerguntas) throws IOException {
79
80 int flagDeclaration = 0;//ajuda a encontrar uma Declaration
81
82 FileReader reader = new FileReader(file);
83 BufferedReader input = new BufferedReader(reader);

```

```

84     String linha;
85     while ((linha = input.readLine()) != null) {
86
87         //trecho que encontra uma pergunta a ser criada
88         if (flagDeclaration == 1) {
89             criaPergunta(linha , listaPerguntas);
90             flagDeclaration =0;
91         }
92         if (linha.equalsIgnoreCase("    <Declaration>")) {
93             flagDeclaration =1;
94         }
95
96
97     }
98     input.close();
99
100 }
101
102
103
104 private static void adicionaEquivalencias(File file ,
105     ArrayList<dtoPagina> lista) throws IOException {
106
107
108
109     FileReader reader = new FileReader(file);
110     BufferedReader input = new BufferedReader(reader);
111     String linha;
112     String nomeClasseBase = null; // classe base é equivalente a classe
113     alvo
114     String nomeClasseAlvo = " "; // classe base é equivalente a classe
115     alvo
116
117     int flagEquivalentClasses=0;
118
119     int apontador=0;
120
121     ArrayList<String> pilha = new ArrayList<>();
122
123     while ((linha = input.readLine()) != null) {
124
125
126
127         if (flagEquivalentClasses == 2) {
128
129             if (linha.contains("    <ObjectComplementOf>")) {
130                 nomeClasseAlvo = nomeClasseAlvo.concat("Complemento de ( ");
131                 pilha.add("ObjectComplementOf");
132             } else if (linha.contains("    <ObjectIntersectionOf>")) {
133                 nomeClasseAlvo = nomeClasseAlvo.concat("Intersecção de ( ");
134                 pilha.add("ObjectIntersectionOf");
135             } else if (linha.contains("    <ObjectSomeValuesFrom>")) {
136                 nomeClasseAlvo = nomeClasseAlvo.concat("Alguns valores de ( ");
137                 pilha.add("ObjectSomeValuesFrom");

```

```

138
139     }else if (linha.contains("                </ObjectComplementOf>")) {
140         nomeClasseAlvo = nomeClasseAlvo.concat(" ) ");
141         pilha.add("ObjectComplementOf");
142     }else if (linha.contains("                </ObjectIntersectionOf>")) {
143         nomeClasseAlvo = nomeClasseAlvo.concat(" ) ");
144         pilha.add("ObjectIntersectionOf");
145     }else if (linha.contains("                </ObjectSomeValuesFrom>")) {
146         nomeClasseAlvo = nomeClasseAlvo.concat(" ) ");
147         pilha.add("ObjectSomeValuesFrom");
148
149     }else if (linha.contains("<Class IRI=")) {
150         apontador = linha.indexOf("#");
151         nomeClasseAlvo = nomeClasseAlvo.concat("[[" );
152         nomeClasseAlvo = nomeClasseAlvo.concat(linha.substring(apontador
+1, linha.length()-3));
153         nomeClasseAlvo = nomeClasseAlvo.concat("]] ");
154     }else if (linha.contains("<ObjectProperty IRI")) {
155         apontador = linha.indexOf("#");
156
157         nomeClasseAlvo = nomeClasseAlvo.concat(linha.substring(apontador
+1, linha.length()-3));
158         nomeClasseAlvo = nomeClasseAlvo.concat(" ");
159     }
160
161
162     }else if (flagEquivalentClasses == 1) {
163         nomeClasseBase = linha.substring(21, linha.length()-3);
164         flagEquivalentClasses = flagEquivalentClasses +1;
165     }
166
167
168     if (linha.equalsIgnoreCase("                <EquivalentClasses>")) {
169         flagEquivalentClasses = 1;
170     }else if (linha.equalsIgnoreCase("                </EquivalentClasses>")) {
171         insereEquivalencia(lista, nomeClasseBase, nomeClasseAlvo);
172         pilha = new ArrayList<>();
173         nomeClasseAlvo = " ";
174
175     }
176
177
178     }
179
180 }
181 /**
182  * Insere em uma classe Base(dtoPagina) a sua equivalencia
183  * @param lista
184  * @param nomeClasseBase
185  * @param nomeClasseAlvo
186  */
187 private static void insereEquivalencia(ArrayList<dtoPagina> lista ,
188     String nomeClasseBase, String nomeClasseAlvo) {
189
190     dtoPagina dto = encontraPagina(lista, nomeClasseBase);
191     if (dto.getEquivalente() == null) {

```

```

192     ArrayList<String> listaEquivalencia = new ArrayList<>();
193     listaEquivalencia.add(nomeClasseAlvo);
194     dto.setEquivalente(listaEquivalencia);
195 } else {
196     dto.getEquivalente().add(nomeClasseAlvo);
197 }
198
199 }
200
201
202 /**
203  * Método que adiciona encontra a tag SubClassOf e adiciona a cada classe(
204     dtoPagina) as suas respectivas subclasses
205  * @param file
206  * @param lista
207  * @throws IOException
208  */
209 private static void adicionaSubclasses(File file , ArrayList<dtoPagina>
210     lista) throws IOException {
211     int flagSubClassOf = 0; //ajuda a encontrar uma SubClassOf
212     String nomeClasseFilha=null;
213     String nomeClasseMae=null;
214
215     FileReader reader = new FileReader(file);
216     BufferedReader input = new BufferedReader(reader);
217     String linha;
218
219     while ((linha = input.readLine()) != null) {
220
221
222         if (flagSubClassOf == 2) {
223             nomeClasseMae=linha.substring(21, linha.length()-3);
224             flagSubClassOf=0;
225             adicionaClasseFilha(lista , nomeClasseMae , nomeClasseFilha);
226         } else if (flagSubClassOf == 1) {
227             nomeClasseFilha=linha.substring(21, linha.length()-3);
228             flagSubClassOf=flagSubClassOf+1;
229         }
230         if (linha.equalsIgnoreCase("<SubClassOf>")) {
231             flagSubClassOf =1;
232         }
233
234
235     }
236
237 }
238
239
240
241 /**
242  * Método que adiciona uma classe filha em uma determinada classe mãe(
243     dtoPagina)
244  * @param lista
245  * @param nomeClasseMae

```

```

245 * @param nomeCLasseFilha
246 */
247 private static void adicionaClasseFilha(ArrayList<dtoPagina> lista ,
248     String nomeClasseMae, String nomeCLasseFilha) {
249
250     for (dtoPagina dtoPagina : lista) {
251         if (dtoPagina.getNome().equalsIgnoreCase(nomeClasseMae)) {
252             if (dtoPagina.getSubClasse() == null) {
253                 ArrayList<String> listaDeSubclasses = new ArrayList<>();
254                 listaDeSubclasses.add(nomeCLasseFilha);
255                 dtoPagina.setSubClasse(listaDeSubclasses);
256             } else {
257                 dtoPagina.getSubClasse().add(nomeCLasseFilha);
258             }
259
260         }
261     }
262 }
263
264 }
265
266
267 /**
268 * Adiciona as classes mãess como superclasses das classes filhas
269 * @param file
270 * @param lista
271 * @throws IOException
272 */
273 private static void adicionaSuperclasses(File file ,
274     ArrayList<dtoPagina> lista) throws IOException {
275     int flagSubClassOf = 0; //ajuda a encontrar uma SubClassOf
276     String nomeCLasseFilha=null;
277     String nomeClasseMae=null;
278
279
280
281     FileReader reader = new FileReader(file);
282     BufferedReader input = new BufferedReader(reader);
283     String linha;
284
285     while ((linha = input.readLine()) != null) {
286
287
288         if (flagSubClassOf == 2) {
289             nomeClasseMae=linha.substring(21, linha.length()-3);
290             flagSubClassOf=0;
291             adicionaClasseMae(lista , nomeClasseMae , nomeCLasseFilha);
292         } else if (flagSubClassOf == 1) {
293             nomeCLasseFilha=linha.substring(21, linha.length()-3);
294             flagSubClassOf=flagSubClassOf+1;
295         }
296         if (linha.equalsIgnoreCase("<SubClassOf>")) {
297             flagSubClassOf =1;
298         }
299
300

```



```

301     }
302
303
304 }
305 /**
306  * Método que adiciona a cada classe filha(dtoPagina) suas respectivas
307  * classes mães
308  * @param lista
309  * @param nomeClasseMae
310  * @param nomeCLasseFilha
311  */
312 private static void adicionaClasseMae( ArrayList<dtoPagina> lista ,
313     String nomeClasseMae, String nomeClasseFilha) {
314
315     for (dtoPagina dtoPagina : lista) {
316         if (dtoPagina.getNome().equalsIgnoreCase(nomeClasseFilha)) {
317             if (dtoPagina.getSuperClasse() == null) {
318                 ArrayList<String> listaDeSuperclasses = new ArrayList<>();
319                 listaDeSuperclasses.add(nomeClasseMae);
320                 dtoPagina.setSuperClasse(listaDeSuperclasses);
321             } else {
322                 dtoPagina.getSuperClasse().add(nomeClasseMae);
323             }
324         }
325     }
326 }
327
328 }
329
330 /**
331  * Método que encontra os ranges de um ObjectProperties, adicionando o
332  * range ao seu OP correspondente na lista de OP's
333  *
334  * @param file
335  * @param listaDeObjectsProperties
336  * @throws IOException
337  */
338 private static void encontraObjectPropertyRange(File file ,
339     ArrayList<dtoObjectProperty> listaDeObjectsProperties) throws
340     IOException {
341
342     String nomeClasse = null;
343     int tamanhoNomeClasse=0;
344
345     int flagObjectPropertyRange = 0;// ajuda a encontrar um
346     ObjectPropertyRange
347     int tamanhoNomeObjectPropertyRange=0;
348     String nomeObjectPropertyRange = null;
349
350     FileReader reader = new FileReader(file);
351     BufferedReader input = new BufferedReader(reader);
352     String linha;
353     while ((linha = input.readLine()) != null) {
354
355         //trecho que encontra os ObjectPropertyRange

```

```

353
354         if (flagObjectPropertyRange > 2) {
355             if(linha.contains("<Class IRI=")) {
356                 //continua adicionando
357                 tamanhoNomeClasse = linha.length();
358                 nomeClasse = linha.substring(25, tamanhoNomeClasse-3);
359                 populaRange(listaDeObjectsProperties, nomeClasse,
nomeObjectPropertyRange);
360             } else {
361                 flagObjectPropertyRange = 0;
362             }
363
364             } else if (flagObjectPropertyRange == 2) {
365                 if (linha.equalsIgnoreCase("<ObjectUnionOf>")) { //
quando for uma união
366                     flagObjectPropertyRange = 3;
367
368                 } else {
369                     flagObjectPropertyRange = 0;
370                     tamanhoNomeClasse = linha.length();
371                     nomeClasse = linha.substring(21, tamanhoNomeClasse-3);
372                     populaRange(listaDeObjectsProperties, nomeClasse,
nomeObjectPropertyRange);
373                     /*
374                     * Método que encontra os ObjectProperties, e aproveita e acha
o seu respectivo domínio, adicionando o OP a uma lista
375                     * */
376                 } else if (flagObjectPropertyRange == 1) {
377                     flagObjectPropertyRange = flagObjectPropertyRange+1;
378                     tamanhoNomeObjectPropertyRange = linha.length();
379                     nomeObjectPropertyRange = linha.substring(30,
tamanhoNomeObjectPropertyRange-3);
380                 }
381                 if (linha.equalsIgnoreCase("<ObjectPropertyRange>")) {
382                     flagObjectPropertyRange = 1;
383                 }
384
385             }
386             input.close();
387
388     }
389
390     /**
391     * Método que encontra os ObjectProperties, e aproveita e acha o seu
respectivo domínio, adicionando o OP a uma lista
392     *
393     * @param file
394     * @param listaDeObjectsProperties
395     * @throws IOException
396     */
397     private static void encontraObjectPropertyDomain(File file,
398     ArrayList<dtoObjectProperty> listaDeObjectsProperties) throws
IOException {
399
400         String nomeClasse = null;
401         int tamanhoNomeClasse=0;

```

```

402
403     int flagObjectPropertyDomain = 0; // ajuda a encontrar um
ObjectPropertyDomain
404     int tamanhoNomeObjectPropertyDomain=0;
405     String nomeObjectPropertyDomain = null;
406
407     FileReader reader = new FileReader(file);
408     BufferedReader input = new BufferedReader(reader);
409     String linha;
410     while ((linha = input.readLine()) != null) {
411
412         //trecho que encontra os ObjectPropertyDomain
413
414
415         if (flagObjectPropertyDomain == 2) {
416             if (linha.equalsIgnoreCase("<ObjectUnionOf>")) { //
quando for uma união
417                 flagObjectPropertyDomain = 0;
418             } else {
419                 flagObjectPropertyDomain = 0;
420                 tamanhoNomeClasse = linha.length();
421                 nomeClasse = linha.substring(21, tamanhoNomeClasse-3);
422                 populaDominio(listaDeObjectsProperties, nomeClasse,
nomeObjectPropertyDomain);
423             }
424             } else if (flagObjectPropertyDomain == 1) {
425                 flagObjectPropertyDomain = flagObjectPropertyDomain+1;
426                 tamanhoNomeObjectPropertyDomain = linha.length();
427                 nomeObjectPropertyDomain = linha.substring(30,
tamanhoNomeObjectPropertyDomain-3);
428             }
429                 if (linha.equalsIgnoreCase("<ObjectPropertyDomain>")) {
430                     flagObjectPropertyDomain =1;
431                 }
432             }
433         }
434
435         input.close();
436
437     }
438     /**
439     * Método que adiciona as anotações referentes a uma classe no OWL, na
sua respectiva página da wiki
440     *
441     * @param file
442     * @param lista
443     * @throws IOException
444     */
445     private static void colocaAnotacoesNaPagina(File file ,
446         ArrayList<dtoPagina> lista) throws IOException {
447
448         int flagAnnotationAssertion =0; //ajuda a encontrar uma
AnnotationAssertion
449         String nomeClasse = null;
450         int tamanhoNomeClasse=0;
451         FileReader reader = new FileReader(file);

```

```

452     BufferedReader input = new BufferedReader(reader);
453     String linha;
454     while ((linha = input.readLine()) != null) {
455
456         //trecho que encontra as anotações de uma respectiva classe
457         if (flagAnnotationAssertion == 3) {
458             flagAnnotationAssertion=flagAnnotationAssertion+1;
459             populaPagina(linha , lista , nomeClasse);
460             flagAnnotationAssertion=0;
461         }else if (flagAnnotationAssertion == 2) {
462             flagAnnotationAssertion=flagAnnotationAssertion+1;
463             tamanhoNomeClasse = linha.length();
464             nomeClasse = linha.substring(14, tamanhoNomeClasse-6);
465
466         }else if (flagAnnotationAssertion == 1) {
467             flagAnnotationAssertion=flagAnnotationAssertion+1;
468
469         }
470
471         if (linha.equalsIgnoreCase(" <AnnotationAssertion>")) {
472             flagAnnotationAssertion = 1;
473         }
474
475     }
476     input.close();
477
478
479 }
480
481 /**
482  * Método que adiciona as anotações referentes a uma pergunta no OWL, na
483  * sua respectiva pergunta
484  *
485  * @param file
486  * @param listaPerguntas
487  * @throws IOException
488  */
489 private static void colocaAnotacoesNaPergunta(File file ,
490     ArrayList<dtoPergunta> listaPerguntas) throws IOException {
491
492     int flagAnnotationAssertion = 0;//ajuda a encontrar uma
493     AnnotationAssertion
494     String nomeClasse = null;
495     int tamanhoNomeClasse=0;
496     FileReader reader = new FileReader(file);
497     BufferedReader input = new BufferedReader(reader);
498     String linha;
499     while ((linha = input.readLine()) != null) {
500
501         //trecho que encontra as anotações de uma respectiva classe
502         if (flagAnnotationAssertion == 3) {
503             flagAnnotationAssertion=flagAnnotationAssertion+1;
504             populaPergunta(linha , listaPerguntas , nomeClasse);
505             flagAnnotationAssertion=0;
506         }else if (flagAnnotationAssertion == 2) {
507             flagAnnotationAssertion=flagAnnotationAssertion+1;

```

```

506         tamanhoNomeClasse = linha.length();
507         nomeClasse = linha.substring(14, tamanhoNomeClasse-6);
508
509     }else if (flagAnnotationAssertion == 1) {
510         flagAnnotationAssertion=flagAnnotationAssertion+1;
511     }
512
513
514         if (linha.equalsIgnoreCase("    <AnnotationAssertion>")) {
515             flagAnnotationAssertion =1;
516         }
517
518     }
519     input.close();
520
521
522 }
523
524 /**
525  * Método que adiciona uma descrição a um determinado object property
526  * equivalente
527  * @param file
528  * @param listaDeObjectsProperties
529  * @throws IOException
530 */
531 private static void adicionadescricoes(File file ,
532     ArrayList<dtoObjectProperty> listaDeObjectsProperties) throws
533     IOException {
534
535     int flagAnnotationAssertion =0;//ajuda a encontrar uma
536     AnnotationAssertion
537     String nomeClasse = null;
538     int tamanhoNomeClasse=0;
539     FileReader reader = new FileReader(file);
540     BufferedReader input = new BufferedReader(reader);
541     String linha;
542     while ((linha = input.readLine()) != null) {
543
544         //trecho que encontra as anotações de uma respectiva classe
545         if (flagAnnotationAssertion == 3) {
546             flagAnnotationAssertion=flagAnnotationAssertion+1;
547             populaObjectProperty(linha , listaDeObjectsProperties , nomeClasse);
548             flagAnnotationAssertion=0;
549         }else if (flagAnnotationAssertion == 2) {
550             flagAnnotationAssertion=flagAnnotationAssertion+1;
551             tamanhoNomeClasse = linha.length();
552             nomeClasse = linha.substring(14, tamanhoNomeClasse-6);
553
554         }else if (flagAnnotationAssertion == 1) {
555             flagAnnotationAssertion=flagAnnotationAssertion+1;
556
557         }
558
559         if (linha.equalsIgnoreCase("    <AnnotationAssertion>")) {
560             flagAnnotationAssertion =1;
561         }
562     }
563 }

```

```

559     }
560     input.close();
561
562 }
563
564
565
566
567 /**
568  * Método que encontra as páginas(classes) e as cloca em uma lista de pá
569  *
570  * @param file
571  * @param lista
572  * @throws IOException
573  */
574
575 private static void criaListaDePaginas(File file , ArrayList<dtoPagina>
576     lista) throws IOException {
577
578     int flagDeclaration = 0;//ajuda a encontrar uma Declaration
579
580     FileReader reader = new FileReader(file);
581     BufferedReader input = new BufferedReader(reader);
582     String linha;
583     while ((linha = input.readLine()) != null) {
584
585         //trecho que encontra uma classe a ser criada
586         if (flagDeclaration == 1) {
587             criaPagina(linha , lista);
588             flagDeclaration =0;
589         }
590         if (linha.equalsIgnoreCase("<Declaration>")) {
591             flagDeclaration =1;
592         }
593
594     }
595     input.close();
596
597 }
598
599 /**
600  * Métodos auxiliares
601  *
602  *
603  */
604
605 /**
606  * Método que adiciona as categorias que serão adicionadas a cada página
607  *
608  * @param file
609  * @param lista
610  */
611 private static void adicionaCategorias(File file , ArrayList<dtoPagina>
612     lista) {

```

```

611
612 //adiciono na lista de categorias , a categoria da página que tem seu pró
        prio nome
613 for (dtoPagina dtoPagina : lista) {
614     ArrayList<dtoCategoria> listaDeCategorias = new ArrayList<>();
615     dtoCategoria categoria = new dtoCategoria();
616     categoria.setNome(dtoPagina.getNome());
617     listaDeCategorias.add(categoria);
618     dtoPagina.setCategoria(listaDeCategorias);
619 }
620
621
622
623 //adiciono as categorias referentes as suas superClasses diretas e
        indiretas
624 for (dtoPagina dtoPagina : lista) {
625     adicionoCategoriasPais(lista ,dtoPagina ,dtoPagina);
626 }
627
628 }
629 /**
630  * Adiciono adicionoCategoriasPais
631  * @param lista
632  * @param dtoPagina
633  * @param dtoPaginaRaiz
634  */
635 private static void adicionoCategoriasPais(ArrayList<dtoPagina> lista ,
        dtoPagina dtoPagina , dtoPagina dtoPaginaRaiz) {
636     if (dtoPagina.getSuperClasse()!=null) {
637
638         for (int i = 0; i < dtoPagina.getSuperClasse().size(); i++) {
639             dtoCategoria categoria = new dtoCategoria();
640             categoria.setNome(dtoPagina.getSuperClasse().get(i));
641             dtoPaginaRaiz.getCategoria().add(categoria);
642             dtoPagina a = encontraPagina(lista ,categoria.getNome());
643             adicionoCategoriasPais(lista ,a ,dtoPaginaRaiz);
644
645         }
646     }else{
647         return;
648     }
649 }
650
651
652 }
653
654
655 /**
656  * Método que organiza uma lista de relações pertencente a cada página
657  * @param lista
658  * @param listaDeObjectsProperties
659  */
660 private static void adicionarRelacoes(ArrayList<dtoPagina> lista ,
        ArrayList<dtoObjectProperty> listaDeObjectsProperties) {
661
662     for (dtoPagina dtoPagina : lista) {
663

```

```

664
665     for (dtoObjectProperty dtoObjectProperty : listaDeObjectsProperties)
666     {
667         if (dtoObjectProperty.getDominio()!=null) {
668             for (int j = 0; j < dtoObjectProperty.getDominio().size(); j++) {
669                 if (!seRelacaoJaExiste(dtoPagina ,dtoPagina.getNome() ,
670                 dtoObjectProperty.getNome() ,dtoObjectProperty.getRange().get(0)) {
671                     if (dtoPagina.getNome().equals(dtoObjectProperty.getDominio()
672                     .get(j))) {
673                         if (dtoPagina.getRelacoes()==null) {
674                             ArrayList<dtoRelacoes> relacoes = new ArrayList<>();
675                             dtoRelacoes relacao = new dtoRelacoes();
676                             relacao.setObjectProperty(dtoObjectProperty.getNome());
677                             relacao.setDominio(dtoPagina.getNome());
678                             relacao.setRange(dtoObjectProperty.getRange().get(0));
679                             relacoes.add(relacao);
680                             dtoPagina.setRelacoes(relacoes);
681                             for (int i = 1; i < dtoObjectProperty.getRange().size();
682                             i++) {
683                                 relacao = new dtoRelacoes();
684                                 relacao.setObjectProperty(dtoObjectProperty.getNome());
685                                 relacao.setDominio(dtoPagina.getNome());
686                                 relacao.setRange(dtoObjectProperty.getRange().get(i));
687
688                                 //relacoes.add(relacao);
689                                 dtoPagina.getRelacoes().add(relacao);
690                             }
691                         }else{
692                             dtoRelacoes relacao = new dtoRelacoes();
693                             for (int i = 0; i < dtoObjectProperty.getRange().size();
694                             i++) {
695                                 relacao = new dtoRelacoes();
696                                 relacao.setObjectProperty(dtoObjectProperty.getNome());
697                                 relacao.setDominio(dtoPagina.getNome());
698                                 relacao.setRange(dtoObjectProperty.getRange().get(i));
699                                 dtoPagina.getRelacoes().add(relacao);
700                             }
701                         }
702                     }
703                 }
704             }
705         }
706     }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }

```



```

715
716 /**
717  * Método que verifica se uma relação já existe
718  * @param dtoPagina
719  * @param dominio
720  * @param objectProperty
721  * @param range
722  * @return
723  */
724 private static boolean seRelacaoJaExiste(dtoPagina dtoPagina, String
725     dominio, String objectProperty,
726     String range) {
727     // TODO Auto-generated method stub
728
729     if (dtoPagina.getRelacoes() != null) {
730         ArrayList<dtoRelacoes> listaRelacoes = dtoPagina.getRelacoes();
731         for (dtoRelacoes dtoRelacoes : listaRelacoes) {
732             if (dtoRelacoes.getDominio().equalsIgnoreCase(dominio)) {
733                 if (dtoRelacoes.getObjectProperty().equalsIgnoreCase(
734                     objectProperty)) {
735                     if (dtoRelacoes.getRange().equalsIgnoreCase(range)) {
736                         //A RELAÇÃO JÁ ESTÁ LISTADA NESSA PÁGINA
737                         return true;
738                     }
739                 }
740             }
741         }
742     } else {
743         return false;
744     }
745 }
746
747
748
749 }
750
751
752 /**
753  * A partir de uma lista de dto's ele escreve um arquivo .txt contendo as
754  * páginas que serão criadas
755  * @param lista
756  * @param listaDeObjectsProperties
757  * @throws IOException
758  */
759 private static void escreveArquivo(ArrayList<dtoPagina> lista, ArrayList<
760     dtoObjectProperty> listaDeObjectsProperties) throws IOException {
761     // TODO Auto-generated method stub
762     System.out.printf("\n***** Escrevendo no arquivo de saída *****\n");
763
764     FileWriter arq = new FileWriter("/home/victor/Downloads/UnB/TG2/
765     core/saidaJava.txt");
766     PrintWriter gravarArq = new PrintWriter(arq);

```

```

765     escreveClassesNoArquivo(arq, lista, gravarArq);
766     escreveObjectPropertiesNoArquivo(arq, listaDeObjectsProperties,
gravarArq);
767
768     arq.close();
769     System.out.printf("\n***** Arquivo de saída gerado, cheque
procurando por: saidaJava.txt *****\n");
770 }
771
772
773
774
775
776
777
778 private static void escreveObjectPropertiesNoArquivo(FileWriter arq,
779     ArrayList<dtoObjectProperty> listaDeObjectsProperties,
780     PrintWriter gravarArq) {
781     for (dtoObjectProperty dtoObjectProperty : listaDeObjectsProperties) {
782         gravarArq.printf("xxxx \n"); //o script Python necessita que as p
783         áginas sejam separadas por marcadores (xxxx e yyyy) e o nome da página
784         deve vir separado por 's (ex: '''NovaPagina''')
785         gravarArq.printf("'''Property:");
786         gravarArq.printf(dtoObjectProperty.getNome());
787         gravarArq.printf("'''");
788         gravarArq.printf("\n");
789         if (dtoObjectProperty.getDescricao() != null) {
790             gravarArq.printf("\n\n");
791             gravarArq.printf(dtoObjectProperty.getDescricao());
792             gravarArq.printf("\n");
793         }
794
795         gravarArq.printf("\n");
796         gravarArq.printf("\n");
797
798
799
800         gravarArq.printf("yyyy \n");
801
802     }
803 }
804
805
806
807
808
809
810 /**
811  * Método que grava as classes no arquivo de saída, como se fossem pá
812  * ginas
813  * @param arq
814  * @param lista
815  * @param gravarArq
816  */

```

```

816 private static void escreveClassesNoArquivo(FileWriter arq, ArrayList<
      dtoPagina> lista, PrintWriter gravarArq) {
817
818     for (dtoPagina dtoPagina : lista) {
819         gravarArq.printf("xxxx \n"); //o script Python necessita que as p
      áginas sejam separadas por marcadores (xxxx e yyyy) e o nome da página
      deve vir separado por 's (ex: '''NovaPagina''')
820         gravarArq.printf("'''");
821         gravarArq.printf(dtoPagina.getNome());
822         gravarArq.printf("'''");
823         gravarArq.printf("\n");
824         if (dtoPagina.getConteudo()!=null) {
825             gravarArq.printf(dtoPagina.getConteudo());
826             gravarArq.printf("\n");
827         }
828
829
830         //escreve as subclasses
831         gravarArq.printf("== Organização ==\n");
832         if (dtoPagina.getSubClasse()!=null) {
833             gravarArq.printf("== Subclasses ==\n");
834             for (int i = 0; i < dtoPagina.getSubClasse().size(); i++) {
835                 gravarArq.printf(" \n\n");
836                 gravarArq.printf(dtoPagina.getNome());
837                 gravarArq.printf(" possui a seguinte subclasse: [[");
838                 gravarArq.printf(dtoPagina.getSubClasse().get(i));
839                 gravarArq.printf("]] \n\n");
840             }
841         }
842         //escreve as superclasses
843         if (dtoPagina.getSuperClasse()!=null) {
844             gravarArq.printf("== Superclasses ==\n");
845             for (int i = 0; i < dtoPagina.getSuperClasse().size(); i++) {
846                 gravarArq.printf(" \n\n");
847                 gravarArq.printf(dtoPagina.getNome());
848                 gravarArq.printf(" possui a seguinte superclasse: [[");
849                 gravarArq.printf(dtoPagina.getSuperClasse().get(i));
850                 gravarArq.printf("]] \n\n");
851             }
852         }
853
854         //escreve as equivalências
855         if (dtoPagina.getEquivalente()!=null) {
856             gravarArq.printf("== Equivalências ==\n");
857             for (int i = 0; i < dtoPagina.getEquivalente().size(); i++) {
858                 gravarArq.printf(" \n\n");
859                 gravarArq.printf(dtoPagina.getNome());
860                 gravarArq.printf(" é equivalente a: ");
861                 gravarArq.printf(dtoPagina.getEquivalente().get(i));
862                 gravarArq.printf(" \n\n");
863             }
864         }
865
866         gravarArq.printf("== Anotações Semanticas ==");
867         gravarArq.printf("\n\n");
868         if (dtoPagina.getRelacoes()!=null) {

```

```

869     gravarArq.printf("== Relações ==\n");
870     for (int i = 0; i < dtoPagina.getRelacoes().size(); i++) {
871         gravarArq.printf(dtoPagina.getRelacoes().get(i).getDominio())
;
872         gravarArq.printf(" ");
873         gravarArq.printf(retiraUnderline(dtoPagina.getRelacoes().get(
i).getObjectProperty()));
874         gravarArq.printf(" []");
875         gravarArq.printf(dtoPagina.getRelacoes().get(i).
getObjectProperty());
876         gravarArq.printf("::");
877         gravarArq.printf(dtoPagina.getRelacoes().get(i).getRange());
878         gravarArq.printf("]] \n\n");
879     }
880 }
881 //escreve as categorias
882 if (dtoPagina.getCategoria()!=null) {
883     for (int i = 0; i < dtoPagina.getCategoria().size(); i++) {
884         gravarArq.printf(" [Category:");
885         gravarArq.printf(dtoPagina.getCategoria().get(i).getNome());
886         gravarArq.printf("]] \n\n");
887     }
888 }
889 gravarArq.printf("\n");
890 gravarArq.printf("\n");
891
892
893
894     gravarArq.printf("yyyy \n");
895
896 }
897
898 }
899
900
901
902 private static String retiraUnderline(String objectProperty) {
903
904
905
906     objectProperty = objectProperty.replace("_", " ");
907
908
909
910     return objectProperty;
911 }
912
913
914
915
916 /**
917  * Método que escreve um arquivo .txt onde ficam os nomes das páginas
    criadas, para que elas possam,
918  * futuramente ser deletadas se necessário for
919  * @param lista
920  * @param listaDeObjectsProperties

```

```

921 * @throws IOException
922 */
923 private static void escreveArquivoDelete( ArrayList<dtoPagina> lista ,
924 ArrayList<dtoObjectProperty> listaDeObjectsProperties) throws
925 IOException {
926
927 System.out.printf("\n***** Escrevendo no arquivo de saída para exclusão de
928 páginas*****");
929
930 FileWriter arq = new FileWriter("/home/victor/Downloads/UnB/TG2/
931 core/deletarPaginas.txt");
932 PrintWriter gravarArq = new PrintWriter(arq);
933
934 for (dtoPagina dtoPagina : lista) {
935     gravarArq.printf(dtoPagina.getNome());
936     gravarArq.printf("\n");
937 }
938
939 for (dtoObjectProperty dtoObjectProperty : listaDeObjectsProperties
940 ) {
941     gravarArq.printf(dtoObjectProperty.getNome());
942     gravarArq.printf("\n");
943 }
944 arq.close();
945 System.out.printf("\n***** Arquivo de saída gerado, cheque
946 procurando por: deletarPaginas.txt *****");
947 }
948
949 /**
950 * Método que popula os dados relativos ao range encontrado
951 *
952 * @param listaDeObjectsProperties
953 * @param nomeClasse
954 * @param nomeObjectPropertyRange
955 */
956 private static void populaRange( ArrayList<dtoObjectProperty>
957 listaDeObjectsProperties ,
958 String nomeClasse, String nomeObjectPropertyRange) {
959     dtoObjectProperty dto = encontraObjectsProperty(
960 listaDeObjectsProperties , nomeObjectPropertyRange);
961
962 if (dto.getRange() == null) {
963     ArrayList<String> range = new ArrayList<>();
964     range.add(nomeClasse);
965     dto.setRange(range);
966     dto.setNome(nomeObjectPropertyRange);
967     listaDeObjectsProperties.add(dto);
968 } else {
969     dto.getRange().add(nomeClasse);
970 }

```

```

969 }
970
971 /**
972  * Método que popula os dados relativos ao ObjectProperty encontrado, e
973  * também ao seu domínio
974  * @param listaDeObjectsProperties
975  * @param nomeClasse
976  * @param nomeObjectProperty
977  */
978 private static void populaDominio( ArrayList<dtoObjectProperty>
979 listaDeObjectsProperties ,
980 String nomeClasse, String nomeObjectProperty) {
981 // encontro a pagina correspondente
982 dtoObjectProperty dto = encontraObjectsProperty(
983 listaDeObjectsProperties , nomeObjectProperty);
984
985 if (dto.getDominio() == null) {
986 ArrayList<String> dominio = new ArrayList<>();
987 dominio.add(nomeClasse);
988 dto.setDominio(dominio);
989 dto.setNome(nomeObjectProperty);
990 listaDeObjectsProperties.add(dto);
991 } else {
992 dto.getDominio().add(nomeClasse);
993 }
994 }
995
996 /**
997  * Método que encontra um ObjectsProperty em uma lista
998  * @param listaDeObjectsProperties
999  * @param nomeObjectProperty
1000  * @return
1001  */
1002 private static dtoObjectProperty encontraObjectsProperty(
1003 ArrayList<dtoObjectProperty> listaDeObjectsProperties ,
1004 String nomeObjectProperty) {
1005 dtoObjectProperty dto = new dtoObjectProperty();
1006
1007 for (dtoObjectProperty dtoObjectProperty : listaDeObjectsProperties) {
1008 if (dtoObjectProperty.getNome().equalsIgnoreCase(nomeObjectProperty))
1009 {
1010 dto=dtoObjectProperty;
1011 }
1012 }
1013 }
1014 return dto;
1015 }
1016 }
1017
1018
1019
1020 /**

```

```

1021 * Método que pega uma AnnotationAssertion e a coloca como conteúdo de
1022 * uma determinada página criada
1023 * @param linha
1024 * @param lista
1025 * @param nomeClasse
1026 */
1027 private static void populaPagina(String linha , ArrayList<dtoPagina> lista
1028 , String nomeClasse) {
1029 // TODO Auto-generated method stub
1030 int tamanho;
1031 tamanho= linha.length();
1032
1033 dtoPagina dto = encontraPagina(lista , nomeClasse);
1034
1035 linha = linha.substring(49, tamanho-10);
1036 dto.setConteudo(linha);
1037
1038
1039
1040 }
1041
1042
1043 /**
1044 * Método que pega uma AnnotationAssertion e a coloca como conteúdo de
1045 * uma determinada pergunta criada
1046 * @param linha
1047 * @param lista
1048 * @param nomePergunta
1049 */
1050 private static void populaPergunta(String linha , ArrayList<dtoPergunta>
1051 listaPerguntas , String nomePergunta) {
1052 // TODO Auto-generated method stub
1053 int tamanho;
1054 tamanho= linha.length();
1055
1056 dtoPergunta dto = encontraPergunta(listaPerguntas , nomePergunta);
1057
1058 linha = linha.substring(49, tamanho-10);
1059 dto.setConteudo(linha);
1060
1061
1062
1063 }
1064 /**
1065 * Método que pega uma AnnotationAssertion e a coloca como descrição de
1066 * uma determinada object property criada
1067 * @param linha
1068 * @param lista
1069 * @param nomeClasse
1070 */

```

```

1071 private static void populaObjectProperty(String linha , ArrayList<
1072     dtoObjectProperty> listaDeObjectsProperties , String nomeClasse) {
1073     // TODO Auto-generated method stub
1074     int tamanho;
1075     tamanho= linha.length();
1076
1077     dtoObjectProperty dto = encontraObjectProperty(listaDeObjectsProperties
1078     , nomeClasse);
1079
1080     linha = linha.substring(49, tamanho-10);
1081     dto.setDescricao(linha);
1082
1083
1084 }
1085 /**
1086  * retorna o dto relativo a uma pagina com um determinado nome
1087  * @param lista
1088  * @param nomeClasse
1089  * @return
1090  */
1091 private static dtoPagina encontraPagina(ArrayList<dtoPagina> lista ,
1092     String nomeClasse) {
1093     // TODO Auto-generated method stub
1094     dtoPagina dto = new dtoPagina();
1095
1096
1097     for (dtoPagina dtoPagina : lista) {
1098
1099         if (dtoPagina.getNome().equalsIgnoreCase(nomeClasse)) {
1100             dto=dtoPagina;
1101         }
1102     }
1103     return dto;
1104 }
1105
1106 /**
1107  * retorna o dto relativo a uma pergunta com um determinado nome
1108  * @param listaPerguntas
1109  * @param nomeClasse
1110  * @return
1111  */
1112 private static dtoPergunta encontraPergunta(ArrayList<dtoPergunta>
1113     listaPerguntas , String nomePergunta) {
1114     // TODO Auto-generated method stub
1115     dtoPergunta dto = new dtoPergunta();
1116
1117
1118     for (dtoPergunta dtoPergunta : listaPerguntas) {
1119
1120         if (dtoPergunta.getNome().equalsIgnoreCase(nomePergunta)) {
1121             dto=dtoPergunta;
1122         }

```



```

1123     }
1124     return dto;
1125 }
1126
1127 /**
1128  * retorna o dto relativo a uma object property com um determinado nome
1129  * @param lista
1130  * @param nomeClasse
1131  * @return
1132  */
1133 private static dtoObjectProperty encontraObjectProperty(ArrayList<
1134     dtoObjectProperty> listaDeObjectsProperties , String nomeClasse) {
1135     // TODO Auto-generated method stub
1136     dtoObjectProperty dto = new dtoObjectProperty();
1137
1138
1139     for (dtoObjectProperty dtoObjectProperty : listaDeObjectsProperties) {
1140
1141         if (dtoObjectProperty.getNome().equalsIgnoreCase(nomeClasse)) {
1142             dto=dtoObjectProperty;
1143         }
1144     }
1145     return dto;
1146 }
1147 /**
1148  * Crio uma página na lista de páginas
1149  *
1150  * @param linha
1151  * @param lista
1152  */
1153 private static void criaPagina(String linha , ArrayList<dtoPagina> lista)
1154 {
1155     // TODO Auto-generated method stub
1156
1157     int tamanho;
1158     tamanho= linha.length();
1159
1160     if (linha.contains("Class IRI")) { // é uma classe , portanto será
1161     transformada em uma página
1162         dtoPagina dto = new dtoPagina();
1163         linha = linha.substring(21, tamanho-3);
1164         dto.setNome(linha);
1165         System.out.println(linha);
1166         lista.add(dto);
1167     }
1168 }
1169 /**
1170  * Crio uma pergunta na lista de perguntas
1171  *
1172  * @param linha
1173  * @param listaPerguntas
1174  */

```

```

1175 private static void criaPergunta(String linha , ArrayList<dtoPergunta>
1176     listaPerguntas) {
1177     // TODO Auto-generated method stub
1178     int tamanho;
1179     tamanho= linha.length();
1180
1181     if (linha.contains("NamedIndividual IRI")) { // é um indivíduo,
1182         portanto será transformada em uma pergunta
1183         dtoPergunta dto = new dtoPergunta();
1184         linha = linha.substring(31, tamanho-3);
1185         dto.setNome(linha);
1186         System.out.println(linha);
1187         listaPerguntas.add(dto);
1188     }
1189 }
1190
1191 }

```

Temos uma série de *dto's* (*data transfer objects*)

```

1
2 public class dtoCategoria {
3
4     String nome;
5
6     public String getNome() {
7         return nome;
8     }
9
10    public void setNome(String nome) {
11        this.nome = nome;
12    }
13
14
15 }

```

```

1 import java.util.ArrayList;
2
3
4 public class dtoObjectProperty {
5
6     String nome;
7     ArrayList<String> range;
8     ArrayList<String> dominio;
9     String descricao;
10
11    public String getDescricao() {
12        return descricao;
13    }
14    public void setDescricao(String descricao) {
15        this.descricao = descricao;
16    }
17    public ArrayList<String> getDominio() {
18        return dominio;
19    }

```

```

20 public void setDominio(ArrayList<String> dominio) {
21     this.dominio = dominio;
22 }
23 public String getNome() {
24     return nome;
25 }
26 public void setNome(String nome) {
27     this.nome = nome;
28 }
29 public ArrayList<String> getRange() {
30     return range;
31 }
32 public void setRange(ArrayList<String> range) {
33     this.range = range;
34 }
35 }
36 }

```

```

1 import java.util.ArrayList;
2
3
4 public class dtoPagina {
5
6     String nome;
7     String conteudo;
8     ArrayList<dtoRelacoes> relacoes;
9     ArrayList<dtoCategoria> categoria;
10    ArrayList<String> superClasse;
11    ArrayList<String> subClasse;
12    ArrayList<String> equivalente;
13
14    public ArrayList<String> getEquivalente() {
15        return equivalente;
16    }
17    public void setEquivalente(ArrayList<String> equivalente) {
18        this.equivalente = equivalente;
19    }
20    public ArrayList<String> getSuperClasse() {
21        return superClasse;
22    }
23    public void setSuperClasse(ArrayList<String> superClasse) {
24        this.superClasse = superClasse;
25    }
26    public ArrayList<String> getSubClasse() {
27        return subClasse;
28    }
29    public void setSubClasse(ArrayList<String> subClasse) {
30        this.subClasse = subClasse;
31    }
32    public ArrayList<dtoCategoria> getCategoria() {
33        return categoria;
34    }
35    public void setCategoria(ArrayList<dtoCategoria> categoria) {
36        this.categoria = categoria;
37    }
38    public ArrayList<dtoRelacoes> getRelacoes() {

```

```

39     return relacoes;
40 }
41 public void setRelacoes(ArrayList<dtoRelacoes> relacoes) {
42     this.relacoes = relacoes;
43 }
44 public String getNome() {
45     return nome;
46 }
47 public void setNome(String nome) {
48     this.nome = nome;
49 }
50
51 public String getConteudo() {
52     return conteudo;
53 }
54 public void setConteudo(String conteudo) {
55     this.conteudo = conteudo;
56 }
57
58
59
60
61
62
63 }

```

```

1 import java.util.ArrayList;
2
3
4 public class dtoRelacoes {
5
6     String range;
7     String dominio;
8     String objectProperty;
9
10    public String getObjectProperty() {
11        return objectProperty;
12    }
13    public void setObjectProperty(String objectProperty) {
14        this.objectProperty = objectProperty;
15    }
16    public String getRange() {
17        return range;
18    }
19    public void setRange(String range) {
20        this.range = range;
21    }
22    public String getDominio() {
23        return dominio;
24    }
25    public void setDominio(String dominio) {
26        this.dominio = dominio;
27    }
28
29
30 }

```

```
1
2 public class dtoPergunta {
3
4     String nome;
5     String conteudo;
6
7
8     public String getNome() {
9         return nome;
10    }
11    public void setNome(String nome) {
12        this.nome = nome;
13    }
14    public String getConteudo() {
15        return conteudo;
16    }
17    public void setConteudo(String conteudo) {
18        this.conteudo = conteudo;
19    }
20
21 }
```